

Schätzung betrieblicher Kostenfunktionen mit künstlichen neuronalen Netzen

Dissertation zur Erlangung des Grades eines
Doktors der Wirtschaftswissenschaften (Dr. oec.)
an der Fakultät Wirtschafts- und Sozialwissenschaften
der Universität Hohenheim

Eingereicht von
Jan-Philipp Simen
aus Karlsruhe

Dekan: Professor Dr. Dirk Hachmeister

Erstgutachter und Betreuer: Professor Dr. Ernst Troßmann

Zweitgutachter: Professor Dr. Andreas Kleine

Datum der mündlichen Prüfung: 21. Mai 2015



Jan-Philipp Simen, Hohenheim 2015

© 2015 by Jan-Philipp Simen. This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl Controlling der Universität Hohenheim. Sie wurde von der Fakultät für Wirtschafts- und Sozialwissenschaften im Dezember 2014 als Promotionsschrift angenommen. Sie richtet sich jedoch nicht nur an ein akademisches Fachpublikum, sondern dient gleichzeitig dem interessierten Nutzer als Handbuch und Dokumentation der von mir entwickelten Anwendungssoftware *Cenobi*, die unter www.cenobi.org zum kostenfreien Download zur Verfügung steht. In *Cenobi* werden Machine-Learning-Algorithmen in einer übersichtlichen und einfach zu bedienenden Anwendung so verbunden, dass typische Probleme der Kostenrechnung mit geringem Aufwand und hoher Prognosegüte gelöst werden können. *Cenobi* ist freie Software und darf unter den Bedingungen der GNU General Public License, Version 3, genutzt, weitergegeben und an eigene Bedürfnisse angepasst werden. Der Programmcode ist unter <https://sourceforge.net/p/cenobi/code> öffentlich zugänglich. Auf beiden Webseiten sind Beiträge in Form von Kommentaren, Verbesserungen oder Weiterentwicklungen sehr willkommen!

Herzlicher Dank gebührt an dieser Stelle meinem Doktorvater Herrn Prof. Dr. Ernst Troßmann, der durch intensive Diskussionen und wertvolle Hinweise mein Dissertationsvorhaben stets auf dem richtigen Weg gehalten hat. Zudem danke ich meinen Industriepartnern für die zur Verfügung gestellten Kostenrechnungsdaten und die Rückmeldungen zur Softwarelösung. Mein Dank gilt weiterhin Herrn Prof. Dr. Andreas Kleine für die Übernahme des Zweitgutachtens sowie Frau Prof. Dr. Katja Schimmelpfeng für die Leitung des Kolloquiums. Meinen Kolleginnen und Kollegen am Lehrstuhl Controlling danke ich für die hervorragende Zusammenarbeit in den vergangenen Jahren. Besonders danke ich Frau Mariya Antoniou für die gewissenhafte Durchsicht des Manuskripts. Meiner lieben Frau danke ich für ihre fortwährende fachliche und moralische Unterstützung.

Hohenheim, im Juli 2015

Jan-Philipp Simen

Inhaltsübersicht

Abbildungsverzeichnis	XI
Algorithmenverzeichnis	XVIII
Symbolverzeichnis.....	XIX
A. Bessere Entscheidungen durch intelligente Kostenschätzung	1
I. Problemstellung der Dissertation	1
II. Gang der Untersuchung.....	3
B. Zur Bedeutung von Kostenfunktionen in der Kostenrechnung.....	6
I. Grundlegende Charakterisierung der Kostenrechnung.....	6
II. Zwecke der Kostenrechnung.....	11
III. Kostenfunktionen in Kostentheorie und Kostenrechnung	20
IV. Anwendungsfälle von Kostenfunktionen für Kostenrechnungs- zwecke.....	55
C. Künstliche neuronale Netze als Instrument zur Schätzung von Kostenfunktionen	83
I. Methodische Grundlagen künstlicher neuronaler Netze.....	83
II. Anwendungsbereiche künstlicher neuronaler Netze	123
III. Zweckmäßigkeit der Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen	130
D. Objektorientierte Umsetzung der Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen	161
I. Java-Programm „Cenobi“	161
II. Beispieldatensatz eines Herstellers von Kunststoffprodukten im Spritzgussverfahren.....	163
III. Schätzung von Kostenfunktionen als Teil eines Knowledge- Discovery-in-Databases-Prozesses.....	169
IV. Entwurf einer Software zur Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen	173

V. Auswahl der unabhängigen Variablen	192
VI. Vorverarbeitung der Daten	200
VII. Transformation der Daten	230
VIII. Objektorientierte Implementierung eines künstlichen neuronalen Netzes.....	237
IX. Objektorientierte Umsetzung der Trainingsalgorithmen	245
X. Bewertung der Modellgüte.....	253
XI. Techniken zur Vermeidung von overfitting.....	258
XII. Modelloptimierung mit Metaheuristiken	267
XIII. Bessere Generalisierungsfähigkeit durch Neuronale-Netze- Ensembles	284
E. Anwendungen künstlicher neuronaler Netze in der Kostenrechnung	300
I. Problemstrukturierung durch transparente Darstellung der Kostenstruktur	300
II. Kostenprognosen mit künstlichen neuronalen Netzen.....	314
III. Alternativenbewertung mit künstlichen neuronalen Netzen.....	317
IV. Kumulative Abweichungsanalysen mit künstlichen neuronalen Netzen.....	327
F. Grenzen der statistischen Schätzung betrieblicher Kostenfunktionen.....	330
Anhang	332
Literaturverzeichnis	342

Inhaltsverzeichnis

Abbildungsverzeichnis	XI
Algorithmenverzeichnis	XVIII
Symbolverzeichnis.....	XIX
A. Bessere Entscheidungen durch intelligente Kostenschätzung	1
I. Problemstellung der Dissertation	1
II. Gang der Untersuchung.....	3
B. Zur Bedeutung von Kostenfunktionen in der Kostenrechnung.....	6
I. Grundlegende Charakterisierung der Kostenrechnung.....	6
II. Zwecke der Kostenrechnung.....	11
1. Entscheidungsunterstützung im Führungsprozess	11
2. Unterstützung der verhaltensorientierten Steuerung.....	14
3. Dokumentation vergangener Kosten.....	18
III. Kostenfunktionen in Kostentheorie und Kostenrechnung	20
1. Grundlegende Charakterisierung von Kostenfunktionen	20
2. Kostenfunktionen als theoretische Grundlage der Erklärung und Prognose in der Kostenrechnung	25
3. Grenzkosten als Entscheidungsgrundlage.....	29
4. Typische Kostenfunktionen in der Kostenrechnung	31
a) Kostenspaltung in fixe und variable Kosten als implizite Kostenfunktion	31
b) Kostenfunktionen der Grenzplankostenrechnung.....	33
c) Multivariate Kostenfunktionen der Prozesskostenrechnung ..	37
5. Übersicht relevanter Kosteneinflussgrößen.....	40
a) Produktionstheoretisch begründete Kosteneinflussgrößen	40
b) Kosteneinflussgrößen in verschiedenen Funktionsbereichen.	45
c) Spezielle Kosteneinflussgrößen für Dienstleistungs- unternehmen.....	50
6. Entscheidungsabhängige Auswahl der Kosteneinflussgrößen	52

IV. Anwendungsfälle von Kostenfunktionen für Kostenrechnungszwecke.....	55
1. Einordnung der Anwendungsfälle in den Führungsprozess	55
2. Problemerkennntnis durch Kostenfunktionen	57
3. Alternativensuche mit Kostenfunktionen.....	59
4. Prognose mit Kostenfunktionen.....	60
5. Alternativenbewertung mit Kostenfunktionen	62
6. Outputorientierte Budgetierung mit Kostenfunktionen	64
7. Kontrolle und Abweichungsanalyse mit Kostenfunktionen	68
a) Unterscheidung von Durchführungs- und Dispositionskontrolle	68
b) Unterscheidung von Abweichungsursachen.....	69
c) Differenzierte Abweichungsanalyse als allgemeines Verfahren	72
d) Alternative und kumulative Abweichungsanalysen als Sonderformen	78
C. Künstliche neuronale Netze als Instrument zur Schätzung von Kostenfunktionen	83
I. Methodische Grundlagen künstlicher neuronaler Netze.....	83
1. Funktionsweise von Feedforward-Netzen.....	83
2. Universelle Approximation und Überanpassung.....	88
3. Parametrisierung von künstlichen neuronalen Netzen.....	92
a) Überwachtes Lernen mit einer Fehlerfunktion.....	92
b) Optimierung der Fehlerfunktion durch Gradientenabstieg....	94
c) Fehler-Rückpropagation zur Berechnung des Gradienten der Fehlerfunktion	96
d) Der Backpropagation-Algorithmus: Gradientenabstieg mit Fehler-Rückpropagation.....	99
e) Nachteile des Backpropagation-Algorithmus	100
f) Alternative Fehlerfunktionen mit Minkowski-Metrik.....	106
g) Erweiterungen und Alternativen zum Backpropagation-Algorithmus.....	109
α) Backpropagation mit Impulsterm.....	109

β) Medianschätzung durch <i>resilient propagation</i>	111
γ) Berechnung der Newton-Richtung mit Gradienten- verfahren zweiter Ordnung.....	113
h) Verbindung von lokalen Lernverfahren mit Meta- heuristiken zu hybriden Algorithmen.....	116
i) Schnelles Lernen mit <i>extreme learning machine</i>	118
II. Anwendungsbereiche künstlicher neuronaler Netze	123
1. Historische Entwicklung der Anwendungsbereiche	123
2. Betriebswirtschaftliche Anwendungen	126
III. Zweckmäßigkeit der Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen	130
1. Analytische Bestimmung von Kostenfunktionen als Mittel der Wahl.....	130
2. Statistische Schätzung von Kostenfunktionen als praktische Vereinfachung	133
3. Allgemeine Probleme beim Einsatz statistischer Verfahren in der Kostenrechnung.....	134
a) Dynamik der Modellumwelt	134
b) Wertebereich der unabhängigen Variablen	137
c) Korrelation von Entscheidungsvariablen.....	138
d) Vergleich von Schlendrian mit Schlendrian? Eignung von Istkostenfunktionen zur Vorgabe und Kontrolle	141
e) Sind statistische Verfahren „theorielos“?	143
4. Vorteile künstlicher neuronaler Netze	147
a) Abbildung nichtlinearer Zusammenhänge unter Berück- sichtigung multipler Interdependenzen.....	147
b) Gute Prognosefähigkeit auch bei unbekanntem Funktions- verläufen.....	149
c) Einfache Modellbildung und Anwendung	153
5. Besondere Schwierigkeiten beim Einsatz künstlicher neuronaler Netze.....	155
a) Vermeidung von <i>overfitting</i>	155
b) Interpretation der geschätzten Parameter	156

6. Zusammenfassende Betrachtung der Zweckmäßigkeit künstlicher neuronaler Netze zur Schätzung von Kostenfunktionen.....	158
D. Objektorientierte Umsetzung der Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen.....	161
I. Java-Programm „Cenobi“	161
II. Beispieldatensatz eines Herstellers von Kunststoffprodukten im Spritzgussverfahren	163
III. Schätzung von Kostenfunktionen als Teil eines Knowledge-Discovery-in-Databases-Prozesses	169
IV. Entwurf einer Software zur Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen	173
1. Notwendigkeit einer eigenständigen Software zur Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen.....	173
2. Grundlagen des objektorientierten Softwareentwurfs	174
a) Prinzipien der Objektorientierung.....	174
b) Objektorientierte Modellierung mit der <i>unified modeling language</i>	177
3. Erfassung der Systemanforderungen in einem Anwendungsfalldiagramm	178
4. Ableitung der Programmstruktur aus dem Anwendungsfalldiagramm.....	184
5. Gestaltung der Benutzeroberfläche	188
V. Auswahl der unabhängigen Variablen	192
1. Kriterien zur Auswahl von Kosteneinflussgrößen	192
2. Prüfung auf Korrelation zwischen den unabhängigen Variablen.....	193
VI. Vorverarbeitung der Daten	200
1. Umgang mit Ausreißern	200
a) Problematik der Ausreißerererkennung.....	200
b) Erkennen von Ausreißern mit autoassoziativen künstlichen neuronalen Netzen.....	203
c) Korrektur der Ausreißer.....	209

2. Umgang mit fehlenden Werten	211
a) Gründe für fehlende Werte	211
b) Nachteile einfacher Verfahren im Umgang mit fehlenden Werten	212
c) Multiple Imputation zur Berücksichtigung der Unsicherheit über die fehlenden Werte	214
α) Grundidee der multiplen Imputation	214
β) Bayessche Regressionsimputation zur Erzeugung repräsentativer Stichproben	216
γ) Robuste Imputation durch <i>predictive mean matching</i>	219
δ) Objektorientierte Implementierung der multiplen Imputation	223
VII. Transformation der Daten	230
1. Normierung metrischer Daten	230
2. Kodierung nominaler und ordinaler Daten	233
VIII. Objektorientierte Implementierung eines künstlichen neuronalen Netzes	237
1. Klassenstruktur der Netzelemente	237
2. Vorwärtsgerichtete Signalverarbeitung im künstlichen neuronalen Netz	242
IX. Objektorientierte Umsetzung der Trainingsalgorithmen	245
1. Struktur der Trainingsklassen	245
2. Objektorientierte Umsetzung des Backpropagation-Algorithmus	248
3. <i>Resilient propagation</i> als Unterklasse der <i>backpropagation</i>	251
4. Implementierung der <i>extreme learning machine</i>	252
X. Bewertung der Modellgüte	253
1. Fehlermaße zur Beurteilung der Prognosegenauigkeit	253
2. Empirische Prognosegenauigkeit der implementierten Trainingsalgorithmen	254
XI. Techniken zur Vermeidung von <i>overfitting</i>	258
1. Modellselektion durch Kreuzvalidierung	258
2. <i>Early stopping</i> der Parametrisierung	261

XII. Modelloptimierung mit Metaheuristiken	267
1. Möglichkeiten der metaheuristischen Optimierung künstlicher neuronaler Netze.....	267
2. Implementierung eines genetischen Algorithmus zur Optimierung der Netzarchitektur.....	269
a) Ziele der Optimierung.....	269
b) Grundlagen evolutionärer Algorithmen	271
c) Kodierung der Netzarchitektur.....	272
d) Ablauf des genetischen Algorithmus	275
e) Struktur der Heuristikklassen	280
3. Empirische Ergebnisse der metaheuristischen Optimierung	282
XIII. Bessere Generalisierungsfähigkeit durch Neuronale-Netze-Ensembles	284
1. Vorteilhaftigkeit einer Ensembleschätzung.....	284
2. Auswahl guter Ensemblemitglieder	286
3. Sicherstellung der Vielfalt im Ensemble	289
4. Implementierung der Ensemblebildung in Cenobi	293
5. Kombination der einzelnen Schätzungen	297
6. Empirische Ergebnisse der Ensembleschätzung	298
E. Anwendungen künstlicher neuronaler Netze in der Kostenrechnung	300
I. Problemstrukturierung durch transparente Darstellung der Kostenstruktur	300
1. Identifikation wichtiger Kosteneinflussgrößen.....	300
2. Graphische Darstellung mehrdimensionaler nichtlinearer Kostenfunktionen.....	307
II. Kostenprognosen mit künstlichen neuronalen Netzen.....	314
III. Alternativenbewertung mit künstlichen neuronalen Netzen.....	317
1. Berechnung der Grenzkosten einer Entscheidung.....	317
2. Kalkulation mit exakten Grenzkosten	319
3. Berechnung von Kostensätzen für Bezugsgrößenkalkulation und Periodenerfolgsrechnung.....	322

Inhaltsverzeichnis	X
IV. Kumulative Abweichungsanalysen mit künstlichen neuronalen Netzen	327
F. Grenzen der statistischen Schätzung betrieblicher Kostenfunktionen.....	330
Anhang	332
Literaturverzeichnis	342

Abbildungsverzeichnis

Abb. B-1: Unterscheidungsmerkmale des internen Rechnungswesens	6
Abb. B-2: Phasenstruktur von Entscheidungsprozessen.....	12
Abb. B-3: Gegenüberstellung von Kostenermittlung (a) und theorie- gestützter Kostenerklärung und -prognose (b).....	27
Abb. B-4: Grenzbetrachtungen bei einer nichtlinearen Kostenfunktion	30
Abb. B-5: Produktionstheoretisch fundierte Kosteneinflussgrößen	43
Abb. B-6: Kosteneinflussgrößen in verschiedenen Funktionsbereichen	48
Abb. B-7: Kosteneinflussgrößen zur Abbildung betrieblicher Komplexität („Komplexitätstreiber“)	49
Abb. B-8: Kosteneinflussgrößen für Dienstleistungsunternehmen.....	52
Abb. B-9: Anwendungsfälle von Kostenfunktionen in der Kostenrechnung ..	56
Abb. B-10: Beispiel einer Kostenprognose	61
Abb. B-11: Prognose- und Sollkostenfunktionen einer flexiblen Plankosten- rechnung	67
Abb. B-12: Iterative Berechnung der Teilabweichungen einer differenzierten Abweichungsanalyse.....	76
Abb. B-13: Teilabweichungen einer alternativen Abweichungsanalyse	78
Abb. B-14: Teilabweichungen einer kumulativen Abweichungsanalyse am Beispiel einer Kostenfunktion mit vier unabhängigen Variablen..	80
Abb. B-15: Kumulative Abweichungsanalyse in der flexiblen Plankosten- rechnung	82

Abb. C-1: Signalverarbeitung im künstlichen Neuron.....	84
Abb. C-2: Verlauf der logistischen Aktivierungsfunktion für unterschiedliche Schwellenwerte θ	85
Abb. C-3: Aufbau eines <i>multilayer perceptron</i>	86
Abb. C-4: <i>Multilayer perceptron</i> mit Schwellenwertneuron	88
Abb. C-5: Approximation einer verrauschten Sinusfunktion durch neuronale Netze mit zwei, fünf und elf verdeckten Neuronen.....	90
Abb. C-6: Geometrische Interpretation des Gradienten einer Fehlerfunktion	95
Abb. C-7: Nachteile des Backpropagation-Algorithmus.....	101
Abb. C-8: Schätzung einer Regressionsgeraden nach der Methode der kleinsten Fehlerquadrate (a) und mit der Manhattan-Metrik (b)...	105
Abb. C-9: Generalisierte Fehlerfunktionen mit $p = 1$, $p = 2$ und $p = 20$	107
Abb. C-10: Newton-Richtung im Vergleich zum negativen Gradienten.....	114
Abb. C-11: Übersicht betriebswirtschaftlicher Anwendungen künstlicher neuronaler Netze	127
Abb. C-12: Literaturüberblick zur Kostenschätzung mit künstlichen neuronalen Netzen	129
Abb. C-13: Extrapolation einer Regressionsgeraden über den Beobachtungsbereich hinaus	137
Abb. C-14: Kostenfunktion mit interdependenten Einflussgrößen.....	148

Abbildungsverzeichnis	XIII
Abb. D-1: Kostenstellen des Beispieldatensatzes (Ausschnitt).....	164
Abb. D-2: Unabhängige Variable des Beispieldatensatzes	166
Abb. D-3: Stufen des Knowledge-Discovery-in-Databases-Prozesses.....	171
Abb. D-4: Elemente eines Anwendungsfalldiagramms	179
Abb. D-5: Anwendungsfalldiagramm für ein System zur Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen.....	181
Abb. D-6: Paketdiagramm für ein objektorientiertes Programm zur Schät- zung von Kostenfunktionen mit künstlichen neuronalen Netzen	187
Abb. D-7: Cenobi-Benutzeroberfläche.....	189
Abb. D-8: Benutzerführung in Cenobi – Datenimport	189
Abb. D-9: Benutzerführung in Cenobi – Datenvorverarbeitung.....	190
Abb. D-10: Benutzerführung in Cenobi – Kostenrechnung	191
Abb. D-11: Zuordnung von Rangzahlen zu kardinalen Daten für die Berechnung des Rangkorrelationskoeffizienten	196
Abb. D-12: Matrix mit Rangkorrelationskoeffizienten für die unabhängigen Variablen des Beispieldatensatzes.....	197
Abb. D-13: Zweidimensionaler Datensatz mit Ausreißer	201
Abb. D-14: Aufbau eines autoassoziativen künstlichen neuronalen Netzes .	203
Abb. D-15: Artificieller Testdatensatz zur Ausreißererkennung	205
Abb. D-16: Absolute Rekonstruktionsfehler eines trainierten autoassozia- tiven Netzes für die normalisierten Werte aus Abb. D-15	206
Abb. D-17: Ergebnis der Ausreißererkennung für die Einflussgrößen des Beispieldatensatzes (Ausschnitt)	208

Abb. D–18: Ergebnis der Ausreißerererkennung für die Kostenstellen des Beispieldatensatzes (Ausschnitt).....	209
Abb. D–19: Ablauf einer multiplen Imputation.....	215
Abb. D–20: <i>Predictive mean matching</i> mit einem Prädiktor x und einer unvollständigen Variablen y	222
Abb. D–21: Dialogfenster zur Kennzeichnung fehlender Werte.....	223
Abb. D–22: Klassendiagramm zur multiplen Imputation	225
Abb. D–23: Sequenzdiagramm zur Initialisierung des <i>predictive mean matching</i> bei der Erzeugung eines neuen Ensemble-Objekts	227
Abb. D–24: Sequenzdiagramm zur multiplen Imputation bei der Ensemblebildung	228
Abb. D–25: Multiple Imputation im Beispieldatensatz.....	230
Abb. D–26: Binärkodierung nominaler Ausprägungen.....	233
Abb. D–27: Binärkodierung ordinaler Ausprägungen.....	234
Abb. D–28: Kennzeichnung nicht-metrischer Variablen im Cenobi-Importdialog.....	235
Abb. D–29: Kodiertabellen (<i>mappings</i>) für nicht-metrische Variable.....	236
Abb. D–30: Einflussgrößen des Spritzguss-Datensatzes mit Kalendermonat in Binärkode	236
Abb. D–31: Klassendiagramm der Netzelemente.....	239
Abb. D–32: Klassendiagramm der Aktivierungsfunktionen.....	241
Abb. D–33: Klassendiagramm der Netzarchitekturen	241

Abb. D-34: Sequenzdiagramm der Vorwärtspropagation eines Eingabevektors im neuronalen Netz	243
Abb. D-35: Sequenzdiagramm der Vorwärtspropagation einer Synapse	244
Abb. D-36: Klassendiagramm der Trainingsalgorithmen	246
Abb. D-37: Einstellungsmenü der Trainingsalgorithmen	247
Abb. D-38: Sequenzdiagramm einer Iteration des Backpropagation-Algorithmus	249
Abb. D-39: Sequenzdiagramm der Fehlerrückpropagation im künstlichen neuronalen Netz	250
Abb. D-40: Prognosefehler der implementierten Trainingsalgorithmen	255
Abb. D-41: Rechendauer der implementierten Trainingsalgorithmen	257
Abb. D-42: Schema einer vierfachen Kreuzvalidierung	259
Abb. D-43: Illustration der Early-Stopping-Idee	263
Abb. D-44: Tatsächlicher Verlauf der Trainings- und Validierungsfehler eines Resilient-Propagation-Algorithmus	263
Abb. D-45: Klassendiagramm der EarlyStop-Klasse in Cenobi	265
Abb. D-46: Struktur der evolutionären Optimierung von Netzarchitektur, Lernverfahren und Gewichten	268
Abb. D-47: Binärcodierung der Netzarchitektur auf einem Chromosom	272
Abb. D-48: Gray-Kode für natürliche Zahlen	273
Abb. D-49: Einstellungsmenü der Netzarchitektur	274
Abb. D-50: Einstellungsmenü des genetischen Algorithmus	277

Abbildungsverzeichnis	XVI
Abb. D–51: Beispiel einer K -Punkt-Kreuzung mit $K = 2$	279
Abb. D–52: Klassendiagramm des Heuristikpakets	281
Abb. D–53: Leistung des genetischen Algorithmus mit verschiedenen Trainingsalgorithmen	283
Abb. D–54: Funktionsapproximationen mit hoher Varianz (a), die im Durchschnitt den wahren Zusammenhang fast unverzerrt wiedergeben (b).....	285
Abb. D–55: Evolutionäre Ensemblebildung mit einer mitlaufenden Ensembleliste	287
Abb. D–56: Finale Generation des genetischen Algorithmus mit <i>extreme</i> <i>learning machine</i> aus Abb. D–53	288
Abb. D–57: Klassendiagramm des Ensemblepakets.....	296
Abb. D–58: Testergebnisse der Ensembleschätzung	299
Abb. E–1: Wichtige Einflussgrößen im Fertigungsbereich	304
Abb. E–2: Wichtige Einflussgrößen in der Kostenstelle Einkauf	305
Abb. E–3: Wichtige Einflussgrößen in der Kostenstelle Hochregallager	305
Abb. E–4: Wichtige Einflussgrößen in der Kostenstelle Verfahrenstechnik ...	306
Abb. E–5: Wichtige Einflussgrößen in der Kostenstelle Auftragsbereit- stellung.....	306
Abb. E–6: Kostenverläufe im Fertigungsbereich in Abhängigkeit von der Zuhaltekraft.....	308
Abb. E–7: Kostenverläufe im Fertigungsbereich nach Ausreißerbeseitigung	309

Abb. E-8: Kostenverläufe im Fertigungsbereich in Abhängigkeit von den geleisteten Spritzgussstunden.....	309
Abb. E-9: Kostenverläufe im Fertigungsbereich in Abhängigkeit von den geleisteten Spritzgussstunden(Fortsetzung)	310
Abb. E-10: Kostenverlauf im Einkauf in Abhängigkeit von der Anzahl bearbeiteter Einkaufsrechnungen.....	310
Abb. E-11: Kostenverläufe im Einkauf in Abhängigkeit von Handelswarenumsatz und Anzahl verkaufter Produktarten.....	311
Abb. E-12: Kostenverlauf im Hochregallager.....	311
Abb. E-13: Kostenverläufe in der Kostenstelle Verfahrenstechnik	312
Abb. E-14: Kostenverläufe in der Auftragsbereitstellung.....	313
Abb. E-15: Kostenprognose in Cenobi.....	315
Abb. E-16: Prognosegenauigkeit des Neuronale-Netze-Ensembles im Vergleich zur linearen Regression für Out-of-Sample-Testdaten ausgewählter Kostenstellen des Beispieldatensatzes.....	316
Abb. E-17: Grenzkostenrechnung in Cenobi	319
Abb. E-18: Kalkulation mit Differenzkosten.....	320
Abb. E-19: Kalkulation in Cenobi	321
Abb. E-20: Berechnung eines Grenzkostensatzes.....	323
Abb. E-21: Berechnung von Kostensätzen in Cenobi	324
Abb. E-22: Bezugsgrößenkalkulation für einen Auftrag.....	324
Abb. E-23: Berechnung eines mittleren Kostensatzes.....	326
Abb. E-24: Abweichungsanalyse in Cenobi.....	329

Algorithmenverzeichnis

Algorithmus C-1: Gradientenabstieg.....	96
Algorithmus C-2: <i>Extreme learning machine</i>	122
Algorithmus D-1: Multiple Imputation mit bayesscher Regressions- imputation	217
Algorithmus D-2: Multiple Imputation mit bayesschem <i>predictive mean</i> <i>matching</i>	221
Algorithmus D-3: Genetische Optimierung der Netzarchitektur	277
Algorithmus D-4: Ensemblebildung in Cenobi	295
Algorithmus E-1: Input-Randomisierung	302
Algorithmus E-2: Kumulative Abweichungsanalyse	327

Symbolverzeichnis

a	Produktionskoeffizient (Gütereinheiten je Einflussgrößeneinheit)
b	Bezugsgrößenkoeffizient (Bezugsgrößeneinheiten je Ausbringungseinheit)
Δ	Änderung, Abweichung
$\Delta K_i^{\text{alt(Ist)}}$	Teilabweichung einer alternativen Abweichungsanalyse auf Istbezugsbasis, die der Einflussgröße x_i zugerechnet wird
$\Delta K_i^{\text{alt(Plan)}}$	Teilabweichung einer alternativen Abweichungsanalyse auf Planbezugsbasis, die der Einflussgröße x_i zugerechnet wird
$\Delta K_{i,\dots,j}^{\text{diff}}$	Teilabweichung einer differenzierten Abweichungsanalyse, die von den Einflussgrößen x_i bis x_j gemeinsam verursacht wurde
ΔK_i^{kum}	Teilabweichung einer kumulativen Abweichungsanalyse, die der Einflussgröße x_i zugerechnet wird
$\Delta w^{(\tau)}$	Gewichtsänderung in Iterationsschritt τ
E	Schätzfehler über alle Beobachtungen
$E^{(n)}$	Schätzfehler der n . Beobachtung
η	Lernrate in einem Gradientenverfahren
f_{act}	Aktivierungsfunktion eines Neurons
G	Gesamtbedarfsmatrix
H	Anzahl der verdeckten Neuronen (<i>hidden units</i>)
I	Anzahl der unabhängigen Variablen (<i>inputs</i>)
k_i^v	variable Kosten einer Einheit von Einflussgröße x_i
K	Kostenhöhe
\widehat{K}	prognostizierte (geschätzte) Kostenhöhe
K^{fix}	fixe Kosten, Absolutglied einer Kostenfunktion
N	Anzahl der Beobachtungen
net_i	Nettoeingangssignal am i . Neuron
o_i	Ausgangssignal am i . Neuron

O	Anzahl der abhängigen Variablen (<i>outputs</i>)
q	Güterpreis (Faktorpreis)
r	Güterverbrauchsmenge (Faktormenge),
r_{ij}	Korrelationskoeffizient
s_{x_i}	empirische Standardabweichung
$s_{x_i x_j}$	empirische Kovarianz
θ	Schwellenwert
w_{ij}	Gewicht der Verbindung vom i . zum j . Neuron
x	unabhängige Variable, Kosteneinflussgröße, Modell-Input
$x^{(n)}$	n . Beobachtung von x
\bar{x}	Mittelwert oder Median von x
$x_{A,j}$	Ausbringungsmenge (Beschäftigung) der j . Leistungsart
y	abhängige Variable
$y^{(n)}$	n . Beobachtung von y
\hat{y}	Schätzer einer abhängigen Variablen
$z^{(n)}$	standardisierter Wert einer Beobachtung $x^{(n)}$.

Fettgedruckte lateinische Kleinbuchstaben kennzeichnen Vektoren (z. B. x),
fettgedruckte lateinische Großbuchstaben Matrizen (z. B. G).

A. Bessere Entscheidungen durch intelligente Kostenschätzung

1. *Problemstellung der Dissertation*

Das Controlling ist im Wandel: *business intelligence* verändert die Erscheinung des Berichtswesens und unter den Stichworten *advanced analytics* und *big data* kehrt das *data mining* zurück – benutzerfreundlicher und mit einer breiteren Datenbasis. Aber gerade in einem Kernbereich des Controlling – der Kostenrechnung – werden die Möglichkeiten moderner Informationstechnik bislang kaum genutzt. Insbesondere werden dort Kostenfunktionen noch immer fast ausnahmslos unter der vereinfachenden Annahme aufgestellt, dass sie linear verlaufen. Eine treffsichere Kostenprognose ist so oft nicht möglich. Außerdem bleiben entscheidungsrelevante Merkmale der Kostenstruktur wie etwa abnehmende oder zunehmende Grenzkosten unentdeckt. Dabei ist eine problemadäquate, d. h. multivariate und ggf. nicht-lineare Funktionsschätzung mit den in heutigen ERP-Systemen und *data warehouses* verfügbaren Vergangenheitsdaten ohne weiteres möglich.

Ein Grund, warum dieser „Datenschatz“ nicht gehoben und betriebliche Kostenfunktionen in der Praxis nur vereinzelt mit Modellen geschätzt werden, die der Komplexität der Entscheidungssituation angemessen sind, liegt vermutlich darin, dass viele Schätzmodelle beim Aufstellen, Parametrisieren und Interpretieren ein statistisches Expertenwissen verlangen, das in Kostenrechnungs-Abteilungen häufig nicht vorhanden ist. Deshalb ist die Einbettung fortgeschrittener statistischer Methoden in übersichtliche und einfach zu bedienende Softwareanwendungen der Schlüssel zu ihrem erfolgreichen Einsatz in der Kostenrechnungspraxis. Wenn der Kostenrech-

ner von einem intelligenten System durch den Analyseprozess geführt wird, wird er eher bereit sein, Modelle mit problemgerechter Komplexität einzusetzen.

Eine Methode, die sich als Basis für eine solche Anwendung anbietet, sind künstliche neuronale Netze. Sie haben gegenüber anderen statistischen Verfahren den Vorteil, dass sie auch unbekannte Funktionstypen zuverlässig schätzen sowie beliebige Funktionsverläufe und Interaktionseffekte unter den unabhängigen Variablen abbilden können. Da die Art des gesuchten funktionalen Zusammenhangs nicht vorgegeben werden muss, lässt sich die Schätzung betrieblicher Kostenfunktionen mit künstlichen neuronalen Netzen leicht automatisieren.

In dieser Arbeit wird ein Konzept entwickelt und in einer Softwareanwendung umgesetzt, das künstliche neuronale Netze so in die Kostenrechnung integriert, dass sie traditionelle, lineare Kostenfunktionen bei entscheidungsorientierten Kostenrechnungsaufgaben ersetzen können. Zu diesen Aufgaben gehören vor allem die Identifikation zentraler Kosteneinflussgrößen, die transparente Darstellung der Kostenstruktur in einem Funktionsgraphen, die Kostenprognose, die Alternativenbewertung sowie die mehrdimensionale Abweichungsanalyse. Eine derart umfassende Integration künstlicher neuronaler Netze in die Kostenrechnung wurde bislang noch nicht vorgenommen.

II. Gang der Untersuchung

In den Kapiteln B und C wird untersucht, unter welchen Bedingungen es zweckmäßig ist, künstliche neuronale Netze zur Schätzung von betrieblichen Kostenfunktionen zu verwenden. Die Untersuchung beschränkt sich auf Kostenfunktionen, die Zwecken der Kostenrechnung dienen. Kostentheoretische Überlegungen spielen nur insoweit eine Rolle, als sie grundlegend für planungsorientierte Systeme der Kosten- und Erlösrechnung sind.

Um die Zweckmäßigkeit einer bestimmten Herangehensweise zur Bestimmung von Kostenfunktionen beurteilen zu können, muss klar sein, welche Zwecke Kostenfunktionen überhaupt erfüllen sollen. Deshalb enthält **Kapitel B** zunächst einen allgemeinen Überblick über typische Zwecksetzungen der Kostenrechnung. Anschließend wird gezeigt, dass Kostenfunktionen wesentlich zur Entscheidungsunterstützung im Führungsprozess beitragen, aber auch bei der hierarchischen Steuerung mit Budgets eingesetzt werden. Demnach bemisst sich die Zweckmäßigkeit einer Kostenfunktion vor allem daran, wie gut sie für diese Anwendungsfälle im Führungsprozess geeignet ist.

Kapitel C befasst sich mit den Grundlagen künstlicher neuronaler Netze. Künstliche neuronale Netze sind mathematische Modelle, die ursprünglich im Bereich der Kognitionswissenschaft¹⁾ entwickelt wurden, um geistige Prozesse zu erklären. Heute werden künstliche neuronale Netze in zahlreichen Anwendungen des maschinellen Lernens und der Mustererkennung eingesetzt. Biologische oder psychologische Plausibilitätsüberlegungen treten dabei in den Hintergrund. Unter anderem können künstliche neuronale Netze für nichtparametrische oder semiparametrische Regressionsanalysen

¹⁾ Vgl. Thagard [Cognitive Science].

verwendet werden.¹⁾ Demnach sind sie grundsätzlich zur statistischen Schätzung von Kostenfunktionen geeignet.

Schließlich wird in Teilkapitel C.III diskutiert, ob Kostenfunktionen auf Basis künstlicher neuronaler Netze zweckmäßiger sind als Kostenfunktionen, die auf andere Weise ermittelt werden. Dabei zeigt sich, dass analytische Kostenfunktionen, die vor allem auf physikalisch-technischen Zusammenhängen begründet sind, jeder statistisch geschätzten Kostenfunktion überlegen sind. Wo die analytische Bestimmung von Kostenfunktionen allerdings nicht möglich oder zu aufwendig ist, haben künstliche neuronale Netze einige Vorteile gegenüber anderen statistischen Verfahren.

In den Kapiteln D und E werden zwei zentrale Gestaltungsfragen untersucht:

- Wie kann die Schätzung betrieblicher Kostenfunktionen mit künstlichen neuronalen Netzen sinnvoll umgesetzt werden (Kapitel D)?
- Wie lassen sich die so geschätzten Kostenfunktionen für verschiedene Kostenrechnungszwecke auswerten (Kapitel E)?

In **Kapitel D** wird zunächst festgestellt, dass es, um Kostenrechnern und betrieblichen Entscheidungsträgern das volle Potential künstlicher neuronaler Netze zur Schätzung von Kostenfunktionen zugänglich zu machen, einer einfach zu bedienenden Software bedarf, die alle notwendigen Schritte des Knowledge-Discovery-in-Databases-Prozesses in einer übersichtlichen Anwendungsumgebung integriert. Da derzeit keine verfügbare Software diesen Anforderungen genügt, wird eine eigene Lösung – das Java-Programm „Cenobi“ – entwickelt und in den Kapiteln D und E beschrieben.

¹⁾ Vgl. White [Nonparametric Regression]; Geman/Bienenstock/Doursat [Dilemma]; Cheng/Titterton [Statistical Perspective] 19; Crone [Neuronale Netze] 208.

Die Grundlage bildet ein objektorientiert implementiertes *multilayer perceptron* ohne Rückkopplungen. Als Trainingsverfahren dienen ein klassischer Backpropagation-Algorithmus sowie *resilient propagation* und *extreme learning machine*. Ein genetischer Algorithmus optimiert die Netzarchitektur, d. h. die Anzahl der Neuronen in Eingabeschicht und verdeckter Schicht. Um eine hohe Generalisierungsfähigkeit zu gewährleisten, wird ein Ensemble aus mehreren trainierten neuronalen Netzen gebildet.

In **Kapitel E** wird beschrieben, wie ein mit Vergangenheitsdaten trainiertes Neuronale-Netze-Ensemble für verschiedene kostenrechnerische Anwendungen ausgewertet werden kann. Mit einer Sensitivitätsanalyse, bei der die Inputvariablen nacheinander randomisiert werden, wird bestimmt, wie groß der Einfluss der einzelnen Inputs auf die Kostenhöhe ist. Zur Visualisierung der Kostenzusammenhänge werden aus dem neuronalen Netz zweidimensionale Funktionsgraphen abgeleitet, indem jeweils eine Inputvariable variiert wird, während die übrigen Inputvariablen konstant bleiben. Für die Alternativenbewertung werden die diskreten Grenzkosten einer Alternative berechnet. Kalkulationen sowie die Berechnung variabler Kostensätze werden als besondere Grenzkostenrechnungen behandelt. Außerdem wird eine kumulative Abweichungsanalyse mit künstlichen neuronalen Netzen umgesetzt.

Kapitel F beschließt die Arbeit mit einem Ausblick zu den Grenzen der entwickelten Konzeption, die vor allem in der Datenverfügbarkeit liegen.

B. Zur Bedeutung von Kostenfunktionen in der Kostenrechnung

I. Grundlegende Charakterisierung der Kostenrechnung

Die Kostenrechnung ist Teil des betrieblichen Informationssystems. Sie richtet sich an betriebsinterne Informationsempfänger und gehört damit zum internen Rechnungswesen. Teilgebiete des internen Rechnungswesens unterscheiden sich in den in Abb. B-1 dargestellten Merkmalen. Im Folgenden werden die charakteristischen Merkmale der Kostenrechnung näher betrachtet.

(1) Zielbezug	ökonomische Ziele Sachziele	Formalziele	soziale Ziele	ökologische Ziele
(2) Zeitbezug	Zeitpunkt	Zeitraum eine Periode mehrere Perioden		
(3) Basisgrößen	Ein- und Auszahlungen	Vermögen und Schulden	Erträge und Aufwendungen	Erlöse und Kosten
(4) zeitliche Richtung	Vergangenheit (Nachrechnungen)		Zukunft (Vorrechnungen)	
(5) Rechnungszweck	Dokumentation	Entscheidungsunterstützung	verhaltensorientierte Steuerung	

Abb. B-1: Unterscheidungsmerkmale des internen Rechnungswesens¹⁾

¹⁾ Vgl. Schweitzer/Küpper [Systeme] 7, abgeändert. Merkmale der Kostenrechnung sind hervorgehoben. Für eine Charakterisierung weiterer Teilgebiete des internen Rechnungswesens vgl. Schweitzer/Küpper [Systeme] 8 ff.

(1) *Zielbezug*: Das interne Rechnungswesen bildet die Zielwirkungen betrieblicher Vorgänge ab. Im betrieblichen Zielsystem können ökonomische, soziale und ökologische Ziele unterschieden werden. Die ökonomischen Ziele eines Betriebs ergeben sich unmittelbar daraus, dass ein Betrieb eine *wirtschaftende* Organisation ist.¹⁾ Wirtschaften als *rationales* Verfügen über knappe Mittel zur planmäßigen *Befriedigung von Bedürfnissen*²⁾ umfasst den *sachlichen* Zweck der Bedürfnisbefriedigung und die *formale* Forderung nach Rationalität.³⁾

Sachziele betreffen demnach vor allem Menge, Beschaffenheit und zeitlichen Anfall der für die Bedürfnisbefriedigung bereitgestellten Güter (Leistungen);⁴⁾ aber auch die Ausgestaltung des gesamten Leistungsprozesses vom Einkauf über die Produktion bis zum Absatz.⁵⁾

Formalziele geben Auskunft über das Verhältnis der Sachzielerreichung zu den dafür aufgewendeten knappen Mitteln (materiellen und immateriellen Gütern).⁶⁾ Eine mengenmäßige Formalzielrechnung – z. B. als Vergleich der Menge der abgesetzten Güter mit der Menge an eingesetzter Arbeit, Rohstoffen, Kapital und anderen Gütern – scheidet an der Verschiedenartigkeit der Vergleichsobjekte. Sie können nur anhand ihres Wertes verglichen werden.⁷⁾ Während also Sachzielwirkungen in der Regel direkt beobachtbar und messbar sind (indem man z. B. die in einer Periode hergestellten Güter

¹⁾ Vgl. Schierenbeck/Wöhle [Grundzüge] 77 f.

²⁾ Vgl. zu dieser Definition u. a. Weber [Wirtschaft] 2. Teil, Kap. I, § 1, S. 181.

³⁾ Vgl. Kosiol [Erkenntnisgegenstand] 130; Kosiol [Aktionszentrum] 54. Schon Max Weber unterscheidet zwischen formaler und materialer Rationalität des Wirtschaftens (vgl. Weber [Wirtschaft] 1. Teil, Kap. 2, § 9, S. 44 f., und § 13, S. 58 f.

⁴⁾ Vgl. Kosiol [Aktionszentrum] 223.

⁵⁾ Vgl. Heinen [Kostenlehre] 26; Schweitzer [Gegenstand] 28 f.; Bea [Entscheidungen] 319 f.

⁶⁾ Vgl. Weber [Wirtschaft] 1. Teil, Kap. 2, § 1.5, S. 32 f.; Kosiol [Erkenntnisgegenstand] 130.

⁷⁾ Vgl. auch im Weiteren Adam [Kostenbewertung] 14 f.

abzählt), erfordern Formalziele zusätzliche bewertende Rechnungen. Diese Bewertung erfolgt im marktwirtschaftlichen System über das Vergleichsgut Geld. Dadurch werden dimensionsverschiedene Güter auf eine einheitliche Skala abgebildet und so vergleichbar gemacht. Als zentrales Formalziel gilt der finanzielle Erfolg (Gewinn/Verlust), der die Veränderung des Geldvermögens zwischen zwei Zeitpunkten widerspiegelt. In der Kostenrechnung werden die Erfolgswirkungen sachzielbezogener Güterverbräuche abgebildet.

(2) *Zeitbezug*: Die Kostenrechnung bezieht sich auf eine Periode (z. B. einen Monat oder eine Woche).

(3) *Basisgrößen*: Aus den Merkmalen (1) und (2) ergibt sich die Basisgröße der Kostenrechnung: *Kosten* sind der *sachzielbezogene, bewertete Güterverbrauch einer Periode*.¹⁾ Unter diese allgemeine Definition fallen sowohl der wertmäßige,²⁾ als auch der pagatorische³⁾ und der ausgabenbezogen entscheidungsorientierte⁴⁾ Kostenbegriff. Den Begriffen liegen zwar unterschiedliche Argumentationen zugrunde, wie die sachzielbezogenen Güterverbräuche zu bewerten sind.⁵⁾ In einer entscheidungslogisch korrekten Rechnung, in der auch Opportunitätskosten berücksichtigt werden, führen aber alle drei Ansätze trotz unterschiedlicher Definitionen zum selben Ergebnis.⁶⁾

¹⁾ Vgl. Schweitzer/Küpper [Systeme] 13; zurückgehend auf Schmalenbach [Kostenrechnung] 6.

²⁾ Vgl. z. B. Fandel [Kostentheorie] 293 f.

³⁾ Vgl. vor allem Koch [Kostenbegriff].

⁴⁾ Vgl. Riebel [Einzelkostenrechnung] 409 ff.

⁵⁾ Vgl. Schweitzer/Küpper [Systeme] 15 f.

⁶⁾ Vgl. Adam [Kostenbewertung] 15 ff., 164 f.; Troßmann [Kostentheorie] 2388; Troßmann/Baumeister [Rechnungswesen] 18 f.

Das Definitionsmerkmal des Güterverbrauchs wird weit ausgelegt und ist nicht auf physische Stoffe beschränkt – so werden auch der Arbeitseinsatz und der Verbrauch von Nominalgütern berücksichtigt.¹⁾ Letzterer geht vor allem in Form von kalkulatorischen Zinsen in die Kostenrechnung ein.²⁾

Charakteristisch für die Berechnung von Kosten ist die monetäre Bewertung einer *mengenmäßigen Bemessungsgrundlage*. Typische Bemessungsgrundlagen sind die Verbrauchsmengen von Gütern. Einige Kosten wie z. B. Steuern, Lizenzgebühren, Mieten und Provisionen basieren hingegen auf anderen Bemessungsgrundlagen, die nur indirekt mit mengenmäßigen Güterverbräuchen zusammenhängen.³⁾

Den Kosten stehen Erlöse als sachzielbezogene, bewertete Güterentstehung gegenüber.⁴⁾ Aus der Differenz von Erlösen und Kosten lassen sich Erfolge für organisatorische Einheiten und sachzielbezogene Handlungsalternativen berechnen. Organisatorische Einheiten sind z. B. Betriebe, Abteilungen, Prozesse oder Einzelpersonen. Für sie werden Periodenerfolge ausgewiesen. Eine typische sachzielbezogene Handlungsalternative ist die Herstellung und der Absatz einer weiteren Produkteinheit einer bestimmten Produktart. Der Erfolg dieser Alternative gegenüber ihrem Unterlassen wird als „Stückerfolg“ bezeichnet.

(4) *Zeitliche Richtung*: Eine grundsätzliche Funktion der Kostenrechnung ist die periodische Erfassung der angefallenen Kosten in Form einer Nach-

¹⁾ Vgl. Kosiol [Wesensmerkmale] 12 f.; Schweitzer/Küpper [Systeme] 13.

²⁾ Vgl. Kosiol [Wesensmerkmale] 17 ff.

³⁾ Vgl. Riebel [Einzelkostenrechnung] 414.

⁴⁾ Vgl. Schweitzer/Küpper [Systeme] 21.

rechnung (Ist-Rechnung).¹⁾ Schon bei ihrer Erfassung werden die Kosten üblicherweise nach Kostenarten geordnet, um sie anschließend auf Kostenstellen und Kostenträger zu verteilen. In einer Vorrechnung (Plan-Rechnung) werden zukünftige Kosten prognostiziert und organisatorischen Einheiten vorgegeben.

(5) *Rechnungszweck*: Die Kostenrechnung ist nicht gesetzlich geregelt und damit frei an gewünschte Zwecksetzungen anpassbar. Für welche Zwecke eine periodische Kostenrechnung aber am besten geeignet ist, wird mitunter kontrovers diskutiert.²⁾ Dabei hat sich eine grobe Einteilung in Zwecke der Entscheidungsunterstützung, der verhaltensorientierten Steuerung und der Dokumentation etabliert.³⁾ Diese Kategorisierung ergibt sich zum Teil aus der zeitlichen Richtung der Rechnung (Merkmal 4). Vorrechnungen dienen der Entscheidungsunterstützung oder der verhaltensorientierten Steuerung. Nachrechnungen dienen grundsätzlich der Dokumentation, können darüber hinaus aber auch zur Fundierung einer Prognose- oder Vorgaberechnung sowie zur Kontrolle früher erstellter Vorrechnungen herangezogen werden.⁴⁾ Dann sind auch sie Teil einer Kostenrechnung zur Entscheidungsunterstützung oder zur verhaltensorientierten Steuerung. Alle drei Rechnungszwecke werden im folgenden Abschnitt näher bestimmt.

¹⁾ Vgl. auch im Weiteren Schweitzer/Küpper [Systeme] 12; Troßmann/Baumeister [Rechnungswesen] 234.

²⁾ Vgl. vor allem den „DBW-Dialog“ zwischen Pfaff/Weber [Zweck], Wagenhofer [Zwecke], Kloock/Schiller [Zwecke] und Pfaff/Weber/Weißberger [Relevance].

³⁾ Eine ausführliche Literaturübersicht zu den Zwecken der Kostenrechnung bietet Selke [Gestaltung] 48 ff.

⁴⁾ Vgl. Troßmann [Controlling] 41.

II. Zwecke der Kostenrechnung

1. Entscheidungsunterstützung im Führungsprozess

Die Funktion der Entscheidungsunterstützung der Kostenrechnung ergibt sich unmittelbar aus ihrer Eigenschaft als formalzielorientierte Rechnung. Formalzielgrößen messen die ökonomische Rationalität einer Handlung und sind damit ein zentraler Orientierungspunkt wirtschaftlicher Entscheidungen. Eine entscheidungsorientierte Kostenrechnung enthält Informationen über die Auswirkungen von Handlungsalternativen auf die Formalzielgröße „Kosten“, so dass ein Entscheidungsträger die im Hinblick auf dieses Ziele optimale Alternative auswählen kann.¹⁾

Die Entscheidungsunterstützungsfunktion der Kostenrechnung beschränkt sich nicht auf die Bewertung von Handlungsalternativen, sondern betrifft den gesamten Prozess, der zur Entscheidung führt und auf die Entscheidung folgt. Dieser besteht aus den untereinander vor- und rückgekoppelten Teilprozessen der Planung, Durchsetzung und Kontrolle, die sich weiter untergliedern lassen (siehe Abb. B-2).²⁾ Ein Entscheidungsprozess, der die Gestaltung von sozialen Systemen betrifft, wird als *Führungsprozess* bezeichnet.³⁾

¹⁾ Zur Problematik, dass einige Kostenrechnungssysteme, zwar als „entscheidungsorientiert“ bezeichnet, dieser Anforderung aber nicht durchweg gerecht werden, vgl. Bungenstock [Kostenrechnungssysteme] und Obermaier [Entscheidungsrelevanz].

²⁾ Vgl. Witte [Phasen-Theorem], mit weiteren Nachweisen; Heinen [Einführung] 22 f.; Wild [Unternehmensplanung] 37; Troßmann [Controlling] 74.

³⁾ Vgl. die entsprechende Definition von „Führung“ bei Wild [Unternehmensplanung] 32; Bea [Führung] 23.

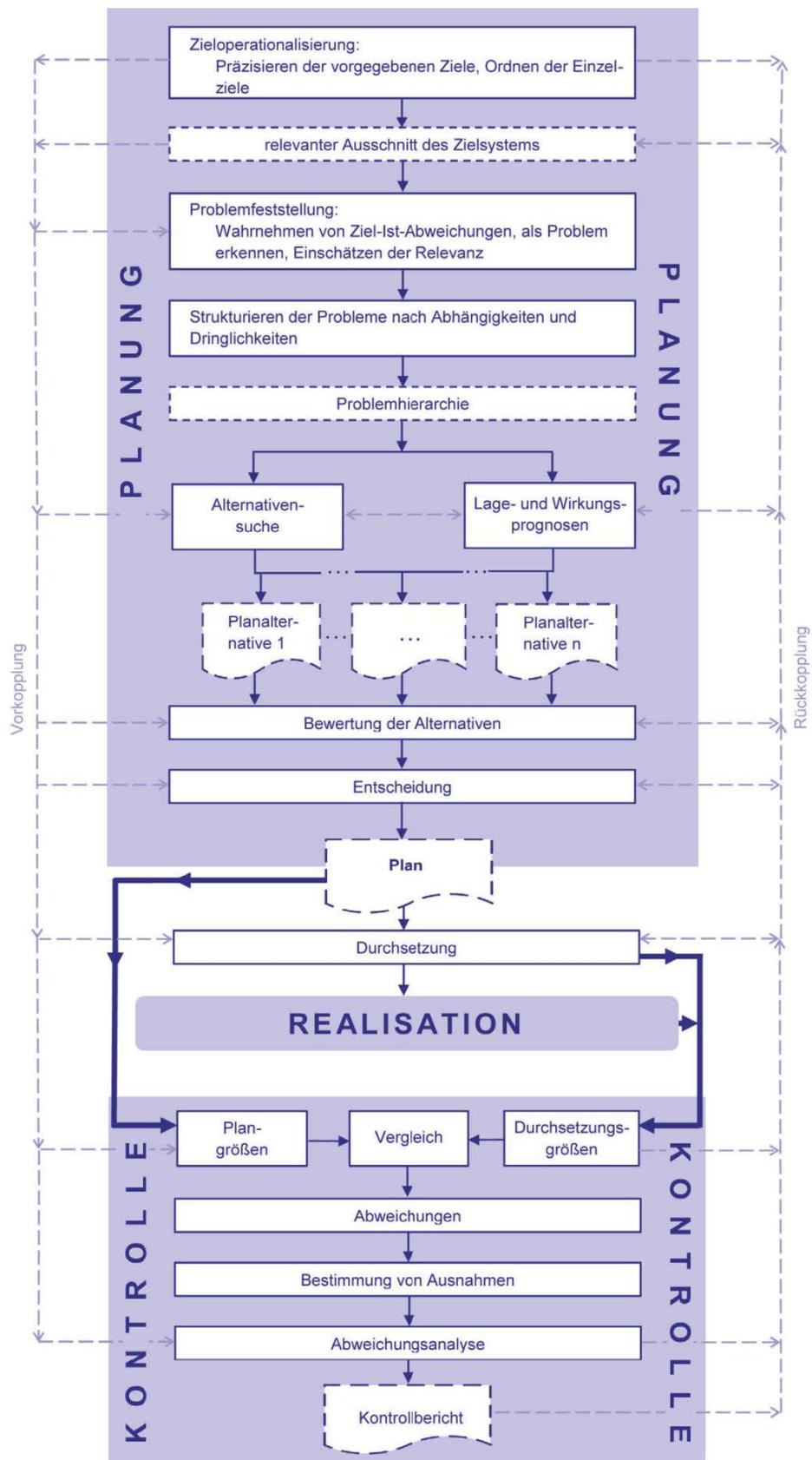


Abb. B-2: Phasenstruktur von Entscheidungsprozessen¹⁾

¹⁾ Troßmann [Controlling] 74.

Die Kostenrechnung eignet sich besonders als Führungsunterstützungsinstrument, weil ihre Basisgröße – periodisch ermittelte Kosten – für viele Entscheidungsträger eine *operable Zielgröße* ist. Im Gegensatz zu anderen Formalzielgrößen, wie z. B. dem Gewinn oder dem Shareholder Value, ist das Zustandekommen von Kostengrößen auch unterhalb der obersten Führungsebene i. d. R. gut nachvollziehbar und beeinflussbar.¹⁾ Dazu tragen die grundlegenden Eigenschaften des Kostenbegriffs bei, der sich immer auf einen begrenzten Zeitraum (eine Periode) und oft auf mengenmäßige Bemessungsgrundlagen (nicht ausschließlich auf Zahlungsströme) bezieht.²⁾ Damit lassen sich die Kostenwirkungen operativer Handlungsalternativen, insbesondere wenn ihr Planungshorizont innerhalb einer Periode liegt und ihr mengenmäßiger Güterverbrauch bekannt ist, leicht abschätzen.³⁾ Dadurch sind Kostenziele auch für Nicht-Ökonomen, z. B. in technischen Unternehmensbereichen, und auch in nicht standardisierten Situationen operabel.

Eine entscheidungsorientierte Kostenrechnung soll einen Entscheidungsträger, der u. a. Kostenziele verfolgt, in den einzelnen Phasen des Führungsprozesses durch geeignete Informationen dabei unterstützen, Entscheidungen zu treffen, die im Hinblick auf seine Zielsetzungen optimal sind.⁴⁾ Da Entscheidungen immer zukunftsgerichtet sind, muss es sich dabei um Prognoseinformationen handeln. Prognosen basieren auf der Beobachtung regelmäßiger Zusammenhänge, von denen man annimmt, dass sie auch in Zukunft gelten. Mathematische Funktionen dienen der exakten Beschreibung solcher Regelmäßigkeiten. In der Kostenrechnung bilden Kos-

¹⁾ Vgl. Heinen [Kostenlehre] 34 f.

²⁾ Siehe Abschnitt I, S. 6 ff.

³⁾ Vgl. auch im Weiteren Pfaff/Weber [Zweck] 160 f.

⁴⁾ Vgl. Selke [Gestaltung] 38 ff.

tenfunktionen Regelmäßigkeiten zwischen der Kostenhöhe und verschiedenen Einflussgrößen ab. Sie tragen dadurch in vielen Phasen des Führungsprozesses wesentlich zur Erfüllung der Kostenrechnungszwecke bei. Teilkapitel IV (S. 55 ff.) zeigt die Anwendungsfälle von Kostenfunktionen zur Führungsunterstützung im Detail.

2. Unterstützung der verhaltensorientierten Steuerung

Damit eine Entscheidung realisiert wird, muss sie in der Organisation durchgesetzt werden (siehe Abb. B-2). Das ist unproblematisch, wenn der Entscheidungsträger seine Entscheidung selbst realisieren kann. Andernfalls ist zu beachten, dass die ausführenden Mitarbeiter nicht immer so handeln *wollen* oder *können*, wie er es wünscht. Verbleibende Handlungsspielräume werden sie entsprechend ihrer individuellen Zielsetzungen und Fähigkeiten ausfüllen. Weil eine *Informationsasymmetrie* zwischen den Ausführenden und dem Auftraggeber besteht, kann auch ein hierarchisch Vorgesetzter diese Spielräume nie ganz beseitigen. Es ist eine Aufgabe der organisatorisch-hierarchischen Führung, ausführende Organisationseinheiten trotzdem zur gewünschten Aufgabenerfüllung zu motivieren und zu befähigen. Möchte man diesen Teilprozess von den übrigen Phasen des Führungsprozesses abgrenzen und dabei besonders betonen, dass psychologische und soziale Eigenschaften, vor allem Fähigkeits- und Präferenzdefizite, sowie Informationsasymmetrien berücksichtigt werden, spricht man von *verhaltensorientierter Steuerung* (Verhaltenssteuerung, Entscheidungssteue-

rung, hierarchischer Lenkung).¹⁾ Sie lässt sich in die Durchsetzungs- und Kontrollphasen des Führungsprozesses einordnen.

Im Gegensatz zur entscheidungsorientierten Kostenrechnung unterstützt die verhaltensorientierte Kostenrechnung einen Manager nicht bei seinen eigenen Entscheidungen, sondern dabei, die Entscheidungen anderer Mitarbeiter in seinem Sinn bzw. im Sinn des Gesamtbetriebs zu beeinflussen.²⁾

Kostenrechnungsinformationen dienen hier vor allem der Vorgabe und Kontrolle von Kostenzielen (Budgets), der Berechnung des Erfolgs einer Verantwortungseinheit sowie der Begründung von Verrechnungspreisen.³⁾ Kostenziele eignen sich gut als Zielvorgaben in der organisatorisch-hierarchischen Steuerung und insbesondere als Grundlage von Anreizsystemen, weil sie für viele Verantwortungseinheiten operabel⁴⁾ sind und in einer definitionslogischen Zweck-Mittel-Beziehung zum Betriebserfolg stehen.⁵⁾ Demnach sind die beiden Hauptanforderungen an ein Belohnungssystem – Controllability (Beeinflussbarkeit) und Anreizkompatibilität (Kongruenz der Bemessungsgrundlage mit den Unternehmenszielen)⁶⁾ – mit Kostengrößen als Bemessungsgrundlagen erfüllbar.

¹⁾ Für ausführliche Überblicksdarstellungen mit weiteren Nachweisen vgl. Wagenhofer [Agency]; Ewert/Wagenhofer [Unternehmensrechnung] 6 ff.; Schweitzer/Küpper [Systeme] 609 ff.

²⁾ Vgl. die Unterscheidung zwischen *decision-facilitating* und *decision-influencing information* bei Demski/Feltham [Cost Determination] 8 f.; vgl. auch Ewert/Wagenhofer [Unternehmensrechnung] 6, mit weiteren Nachweisen; Troßmann/Baumeister [Rechnungswesen].

³⁾ Siehe Abschnitt IV.1, S. 55 ff.

⁴⁾ Siehe S. 13.

⁵⁾ Vgl. Heinen [Kostenlehre] 34 f.; Wild [Unternehmensplanung] 59. Der kalkulatorische Erfolg ist definiert als Differenz aus Erlösen und Kosten (vgl. Schweitzer/Küpper [Systeme] 21).

⁶⁾ Vgl. Troßmann [Controlling] 234 ff.

Für die übergeordnete Einheit ist es in einigen Fällen sinnvoll, zur verhaltensorientierten Steuerung andere Informationen weiterzugeben als die, die sie selbst zur Entscheidungsfindung verwendet.¹⁾ Beispielsweise kann ein Bereichsleiter ein Kostenbudget von 2.000.000 € für Abteilung XY *einplanen*, der Abteilung aber ein Budget von 1.800.000 € *vorgeben* – unter der Annahme, dass eine schwierigere Zielvorgabe stärker zu zielgerichtetem Verhalten motiviert,²⁾ diese Vorgabe aber nicht erreicht wird (die Abteilung sollte natürlich nicht wissen, dass die Bereichsleitung von vornherein damit rechnet, dass die Budgetvorgabe um 200.000 € überschritten wird).

Die Idee, Fähigkeits- und Präferenzdefizite in der Kostenrechnung zu berücksichtigen, ist keineswegs neu. Bereits 1959 begegnete Riebel der Befürchtung, dass das von ihm propagierte Rechnen mit Einzelkosten „zu einer nachgiebigen Preispolitik verleitet und dann keine Vollkostendeckung erreicht wird“, mit der Empfehlung, die für die Preisgestaltung Verantwortlichen nicht nur über die Produkt-Einzelkosten, sondern auch über erwartete Soll-Deckungsbeiträge zu informieren, aus denen sich Preisuntergrenzen oberhalb der Einzelkosten ergeben.³⁾

Psychologische, soziale und soziopsychologische Faktoren der Informationsweitergabe und -verarbeitung wurden in der deutschsprachigen Kostenrechnungsliteratur allerdings lange Zeit nur unsystematisch und ohne

¹⁾ Vgl. Weber [Verhaltensorientierung] 103; Ewert/Wagenhofer [Unternehmensrechnung] 10 f.

²⁾ Zur Motivationswirkung anspruchsvoller Ziele vgl. Ewert/Wagenhofer [Unternehmensrechnung] 309; Frese/Graumann/Theuvsen [Organisation] 139; Weber/Schäffer [Controlling] 70 ff.; jeweils mit weiteren Nachweisen.

³⁾ Riebel [Rechnen] 234; Nachdruck in Riebel [Einzelkostenrechnung] 54 f. Weitere praktische Beispiele, wo entscheidungs- und verhaltensorientierte Kostenansätze auseinanderfallen, finden sich bei Pfaff [Entscheidungssteuerung] 152 f. und Ewert/Wagenhofer [Unternehmensrechnung] 10 f.; Ewert [Entscheidungssteuerung] ordnet die Beiträge von Riebel und Pfaff in die Terminologie der Agency-Theorie ein.

Bezug auf verhaltenswissenschaftliche Forschungsergebnisse behandelt.¹⁾ Dagegen hat sich im angelsächsischen Raum seit den 1960er Jahren die *Behavioral-Accounting-Forschung* als eigenständige Forschungsrichtung etabliert.²⁾ Sie untersucht einerseits die Wirkung von Rechnungswesen-Informationen auf Einzelpersonen, Gruppen, Organisationen und das gesellschaftliche Umfeld, aber auch, wie diese Akteure das Rechnungswesen prägen.³⁾ Neben der weit gefächerten Fundierung durch verhaltenswissenschaftliche Theorien sind experimentelle Studien von herausragender Bedeutung für das *behavioral accounting*.⁴⁾ Nach 1990 sind Methoden und Erkenntnisse des *behavioral accounting*, darunter vor allem Principal-Agent-Modelle, auch in der deutschsprachigen Literatur umfassend rezipiert worden.⁵⁾

Auch wenn entscheidungsorientierte Rechnungen zum Teil andere Informationen enthalten als solche zur verhaltensorientierten Steuerung, können sie in der Regel auf dieselbe Grundrechnung, d. h. dieselbe Datenbasis (z. B. eine Datenbank),⁶⁾ zurückgreifen. Sie unterscheiden sich dann nur in der Auswertung der dort erfassten Kosteninformationen.⁷⁾ Im Übrigen verlangt aber auch jede Entscheidungssituation ihre eigene Auswertungsrechnung.

¹⁾ Vgl. auch im Weiteren Weber [Gestaltung] 80 f.

²⁾ Vgl. auch im Weiteren Schweitzer/Küpper [Systeme] 610 ff.

³⁾ Vgl. Birnberg [Framework] 3 ff.

⁴⁾ Für einen sehr umfassenden Literaturüberblick zur aktuellen Behavioral-Accounting-Forschung vgl. Birnberg [Framework]; für eine einführende Übersicht vgl. Schweitzer/Küpper [Systeme] 610 ff. und Fehrenbacher [Behavioral Accounting].

⁵⁾ Vor allem durch Ewert/Wagenhofer [Unternehmensrechnung] und Schweitzer/Küpper [Systeme] 609 ff. Vgl. auch Wagenhofer [Agency]; Pfaff/Weber [Zweck] 156 ff.; Weber [Gestaltung] 52 ff. Neuere Ergebnisse präsentieren u. a. Hofmann/Küpper [Neurobiologie].

⁶⁾ Zur Realisierung einer Grundrechnung als Datenbank vgl. Riebel/Sinzig [Datenbank].

⁷⁾ Zur Trennung von Grund- und Auswertungsrechnungen vgl. Schmalenbach [Kostenrechnung] 268 ff. (Schmalenbach spricht von *Grundrechnung* und *Sonderrechnungen*); sowie ausführlich Riebel [Einzelkostenrechnung] 149 ff.

Schließlich soll eine entscheidungsorientierte Kostenrechnung darüber Auskunft geben, welchen Einfluss eine Handlungsalternative unter gegebenen Umständen auf die in der Entscheidungssituation maßgeblichen Formalziele hat.¹⁾ Das lässt sich nur selten direkt aus der Datenbasis der Kostenrechnung erkennen. Eine verhaltensorientierte Kostenrechnung steht also nicht im Widerspruch zur entscheidungsorientierten Kostenrechnung, verlangt aber zusätzliche Auswertungsrechnungen.²⁾

Die Einsatzmöglichkeiten von *Kostenfunktionen* zur verhaltensorientierten Steuerung beschränken sich weitgehend auf die Fundierung und Kontrolle von outputorientierten (leistungsorientierten) Kostenbudgets.³⁾ Für andere Kostenrechnungszwecke in der Durchsetzungsphase werden Kostenfunktionen höchstens indirekt gebraucht.

3. Dokumentation vergangener Kosten

Die periodische Dokumentation der angefallenen Kosten ist kein Selbstzweck. Vergangenheitsdaten werden für übergeordnete interne oder externe Zwecke gebraucht. Intern sind sie die Grundlage für Prognoserechnungen, Kontrollen und weitere Unterstützungsrechnungen im Führungspro-

¹⁾ Vgl. Riebel [Einzelkostenrechnung] 602; Troßmann [Rechnungswesen] 106; Troßmann/Baumeister [Rechnungswesen].

²⁾ Vgl. Weber [Verhaltensorientierung] 104; Wagenhofer [Zwecke] 133 f.; Kloock/Schiller [Zwecke] 136. Demhingegen behauptet Pfaff ([Entscheidungssteuerung] 154 f.), der Relativitätsgrundsatz „different costs for different purposes“ gelte nicht für das Verhältnis von entscheidungs- und verhaltensorientierter Kostenrechnung. Gegen Pfaff argumentiert wiederum Wagenhofer, durch die geschickte Gestaltung von Anreizsystemen und Verrechnungspreisen sei durchaus eine Kostenrechnung realisierbar, die sowohl entscheidungsorientierten als auch verhaltensorientierten Zwecken genügt (vgl. [Entscheidungssteuerung] 103).

³⁾ Siehe dazu die Abschnitte IV.6 und IV.7, S. 64 ff. Siehe auch Abb. B-9, S. 56.

zess.¹⁾ Insbesondere gehen sie in statistische Schätzverfahren zur Parametrisierung von Kostenfunktionen ein. Die Kostenfunktionen selbst werden für rein dokumentarische Zwecke allerdings nicht benötigt.

Extern werden Vergangenheitsdaten der Kostenrechnung vor allem für die Kalkulation öffentlicher Aufträge,²⁾ zur Bewertung selbsterstellter Erzeugnisse in der Bilanz³⁾ sowie zur Preisrechtfertigung gegenüber Kunden und staatlichen Institutionen herangezogen. Der Gedanke, die eigene Kalkulation offenzulegen, um damit den Produktpreis gegenüber den Kunden zu rechtfertigen, wurde zuletzt im Zusammenhang des *open-book accounting* verstärkt thematisiert.⁴⁾ Die Rechtfertigung gegenüber staatlichen Institutionen betrifft vor allem die Betreiber großer Versorgungsnetze (u. a. Post-, Telekommunikations-, Energieversorgungs- und Eisenbahnnetze). Als Inhaber eines natürlichen Monopols müssen sie ihre Preise regelmäßig gegenüber staatlichen Regulierungsbehörden begründen.⁵⁾

Den angeführten Dokumentationsrechnungen für externe Zwecke ist gemeinsam, dass sie stark von externen Anforderungen geprägt sind und wenig Gestaltungsspielraum zulassen.⁶⁾ Für interne Entscheidungszwecke

¹⁾ Siehe Abschnitt 1, S. 11 ff.

²⁾ Vgl. Budäus [Aufträge].

³⁾ Vgl. Eisele/Knobloch [Technik] 401 ff.

⁴⁾ Vgl. Hoffjan/Lührs/Kolburg [Transparency]; Windolph/Moeller [Open-book].

⁵⁾ Vgl. Schweitzer/Küpper [Systeme] 794.

⁶⁾ Vgl. Weber [Gestaltung] 39 f.

sind sie nicht geeignet.¹⁾ Sie zählen deshalb zum externen Rechnungswesen²⁾ und werden in dieser Arbeit nicht weiter berücksichtigt.

III. Kostenfunktionen in Kostentheorie und Kostenrechnung

1. Grundlegende Charakterisierung von Kostenfunktionen

Kostenfunktionen zeigen, wie die Kostenhöhe von verschiedenen *Kosteneinflussgrößen* (Kostenbestimmungsgrößen, *cost drivers*) abhängt. Eine Kostenfunktion der Form

$$K = f(x_1, x_2, \dots, x_I) = f(\mathbf{x}) \quad (\text{B.1})$$

mit $\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_I \end{pmatrix}$

gibt darüber Auskunft, wie sich die Kostenhöhe K ändert, wenn die Einflussgrößen \mathbf{x} variieren. Formal gesprochen bildet sie den Vektor der Einflussgrößen \mathbf{x} auf einen Geldbetrag $f(\mathbf{x})$ ab.³⁾ Je nach dem, wie gut sich der

¹⁾ Selbst wenn eine Kalkulation freiwillig offengelegt wird, steht die Beeinflussung der externen Informationsempfänger im Vordergrund – nicht die eigene Entscheidungsfundierung (vgl. Troßmann/Baumeister [Rechnungswesen]). Beispielsweise kann ein Verkäufer den hohen Preis seines Produkts besser auf einer Vollkostenrechnung als auf einer entscheidungsorientierten Teilkostenrechnung begründen (vgl. Pfaff/Weber [Zweck] 162, Anmerkung 9).

²⁾ Vgl. Troßmann [Controlling] 40 f.; Troßmann/Baumeister [Rechnungswesen] 8.

³⁾ Vgl. auch im Weiteren Troßmann [Kostenfunktionen] 1195.

abgebildete Zusammenhang empirisch bewährt hat, bezeichnet man ihn als *Hypothese, Regelmäßigkeit* oder *gesetzesartige Aussage*.¹⁾

Die Anwendung von Kostenfunktionen ist nicht auf bestimmte betriebliche Funktionsbereiche oder einzelne Wirtschaftszweige beschränkt. Die Funktion f kann eine beliebige Struktur und beliebige Einflussgrößen enthalten.²⁾ Der Schwerpunkt kostentheoretischer Überlegungen lag in der Vergangenheit allerdings auf der industriellen Produktion. Auf Grundlage der Produktionstheorie wurden einige sehr differenzierte Kostenfunktionen entwickelt.³⁾ Sie sind zur Anwendung in anderen Unternehmensbereichen und in Dienstleistungsbetrieben nur bedingt geeignet.⁴⁾ Deswegen wird in diesem Abschnitt lediglich ihre gemeinsame Grundstruktur, die sich ohne weiteres auch auf Zusammenhänge außerhalb der Produktion anwenden lässt, so weit referiert, wie es zum Verständnis wichtiger Argumentationslinien und Methoden der Kostenrechnung notwendig ist.

Entsprechend der Definition von Kosten als sachzielbezogenem, *bewertetem Güterverbrauch*⁵⁾ unterscheiden produktionstheoretisch fundierte Ansätze einen Bewertungsvektor q und einen Verbrauchsvektor r . Aus dem Skalarprodukt dieser Vektoren resultiert eine allgemeine Kostenfunktion

¹⁾ Vgl. auch Zelewski [Forschungsprogramme] 492 f. Dagegen lehnt es Dyckhoff ([Neukonzeption] 712 f.) ab, diese naturwissenschaftlich geprägte Terminologie in der Betriebswirtschaftslehre zu verwenden.

²⁾ Zu möglichen Kosteneinflussgrößen in verschiedenen Funktionsbereichen siehe Abb. B-6 (S. 48); für Einflussgrößen in unterschiedlichen Branchen siehe Abb. B-8 (S. 52).

³⁾ Für einen Überblick vgl. z. B. Corsten/Gössinger [Produktionswirtschaft] 141 ff.; Fandel [Kostentheorie] 335 ff.

⁴⁾ Vgl. Neugebauer [Versicherungsunternehmen] 34 ff.; Fandel [Kostentheorie] 2 ff. Zu den Problemen einer Produktionstheorie für Dienstleistungen vgl. Dyckhoff [Neukonzeption] 721 f.; Corsten/Gössinger [Überlegungen] 516 ff. Konzeptionen einer Produktionstheorie für Dienstleistungen finden sich bei Altenburger [Dienstleistungen]; Schneeweiß [Dienstleistungsproduktion]; Fandel [Kostentheorie] 106 ff. Eine institutionenökonomisch fundierte Kostentheorie skizziert Neugebauer [Versicherungsunternehmen] 83 ff.

⁵⁾ Siehe S. 8.

$$K = \mathbf{q} \cdot \mathbf{r} = (q_1, \dots, q_M) \cdot \begin{pmatrix} r_1 \\ \vdots \\ r_M \end{pmatrix} = q_1 r_1 + \dots + q_M r_M = \sum_{j=1}^M q_j r_j \quad (\text{B.2})$$

mit Güterpreisen („Faktorpreisen“) \mathbf{q} und Gütermengen („Faktormengen“) \mathbf{r} .¹⁾ Der Güterpreis q_j ist der Wert, den eine Mengeneinheit r_j für den Betrieb hat. Dieser Wert hängt von den zugrundeliegenden Zielvorstellungen ab.²⁾ Er muss nicht dem Marktpreis entsprechen (soweit dieser überhaupt feststellbar ist), sondern kann u. a. an den Opportunitätskosten („Schattenpreisen“) des Güterverbrauchs orientiert sein. In der produktionstheoretischen Literatur werden auch Bewertungsansätze vorgeschlagen, die neben dem ökonomischen Wert im engeren Sinn³⁾ zusätzlich soziale und ökologische Bewertungen des Güterverbrauchs berücksichtigen.⁴⁾

Der Güterverbrauch \mathbf{r} hängt über eine Verbrauchsfunktion $\mathbf{r} = g(\mathbf{x})$ vom Einflussgrößenvektor \mathbf{x} ab. Gleichmaßen sind die Güterpreise \mathbf{q} über eine Preisfunktion $\mathbf{q} = h(\mathbf{x})$ einflussgrößenabhängig. Die Preise können je nach Güterart z. B. mit der gewährten Rabattstaffelung in Abhängigkeit von der Beschaffungsmenge oder wegen Überstundenzuschlägen und leistungsabhängigen Prämienzahlungen variieren.⁵⁾ Insgesamt lässt sich (B.2) zu

$$K = \mathbf{q} \cdot \mathbf{r} = h(\mathbf{x}) \cdot g(\mathbf{x}) \quad (\text{B.3})$$

umformen. Dabei müssen in h und g nicht dieselben Elemente des Vektors \mathbf{x} wirksam sein; es kann aber durchaus Überschneidungen geben. Durch Mengenrabatte könnte z. B. der Preis q_j mit zunehmender Gütermenge r_j

¹⁾ Vgl. auch im Weiteren Schweitzer/Küpper [Kostentheorie] 220 ff., 326.

²⁾ Vgl. auch im Weiteren Schweitzer/Küpper [Systeme] 15 f.

³⁾ Siehe Abschnitt I, S. 6 ff.

⁴⁾ Vgl. Esser [Produktionstheorie] 82 ff., mit weiteren Nachweisen; Dyckhoff/Spengler [Produktionswirtschaft] 102 f.

⁵⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 170 ff.; Riebel [Einzelkostenrechnung] 415 f.

sinken. Für Preise, die auf diese Weise ausschließlich von der Gütermenge abhängen, gilt $q = h(g(x))$.¹⁾

Als Verbrauchsfunktionen $g(x)$ sind verschiedene Produktionsfunktionen geeignet. Sehr breite Anwendungsmöglichkeiten bietet Kloocks Produktionsfunktion vom Typ D.²⁾ Sie enthält u. a. die Produktionsfunktionen von Leontief, Gutenberg (Typ B) und Heinen (Typ C) als Sonderfälle.³⁾ Einem allgemeinen Input-Output-Ansatz folgend werden hier die Verbrauchsmengen über eine Gesamtbedarfsmatrix G berechnet. Damit ergibt sich eine Kostenfunktion

$$K = h(x) \cdot g(x) = h(x) \cdot G \cdot x \quad (\text{B.4})$$

Mit diesem Ansatz lassen sich auch mehrstufige Produktionsprozesse abbilden, bei denen der Output der einen Stelle von einer anderen Stelle als Input weiterverarbeitet wird.⁴⁾ Auch mehrstufige Güterverflechtungen über den Produktionsprozess hinaus können so erfasst werden. Beispielsweise bilden Mengen und Werte von Beschaffungs- und Ausbringungsgütern die Bemessungsgrundlagen von Steuern, Lizenzen und Provisionen.⁵⁾ Über geeignete Einflussgrößen und eine entsprechende Erweiterung der Gesamtbedarfsmatrix und des Bewertungsvektors können diese Kostenarten in (B.4) integriert werden. Dasselbe gilt für die Kosten des Kapitalbedarfs. Die finanz- und kapitaltheoretisch fundierte Berechnung von Zinsen⁶⁾ und an-

¹⁾ Vgl. Schweitzer/Küpper [Kostentheorie] 225.

²⁾ Vgl. auch im Weiteren Fandel [Kostentheorie] 215 ff.; Schweitzer/Küpper [Kostentheorie] 176 f., 209 f.

³⁾ Vgl. Schweitzer/Küpper [Kostentheorie] 163.

⁴⁾ Vgl. Schweitzer/Küpper [Kostentheorie] 167 ff.

⁵⁾ Vgl. Riebel [Einzelkostenrechnung] 414.

⁶⁾ Zur Berechnung kalkulatorischer Zinsen für die Kostenrechnung vgl. Küpper [Berechnung] und Küpper [Bedeutung].

derem Nominalgüterverbrauch¹⁾ ist ohne größere Schwierigkeiten in (B.4) möglich.²⁾ In der Betriebsplankostenrechnung (auch „Einflussgrößenrechnung“) nach Laßmann werden mehrstufige Kostenzusammenhänge wie in (B.4) modelliert.³⁾

In anderen Kostenrechnungssystemen kommen jedoch häufig einfachere Kostenfunktionen zum Einsatz. Sie setzen eine *additive Separabilität* der Kosteneinflussgrößen voraus, so dass sich Kostenfunktionen der Form

$$\begin{aligned} K &= \mathbf{q} \cdot \mathbf{g}(\mathbf{x}) = \mathbf{q} \cdot \begin{pmatrix} a_{11}x_1 + \cdots + a_{1I}x_I \\ \vdots \\ a_{M1}x_1 + \cdots + a_{MI}x_I \end{pmatrix} \\ &= \sum_{j=1}^M \left(q_j \cdot \sum_{i=1}^I a_{ji}x_i \right) = \sum_{i=1}^I \left(\sum_{j=1}^M q_j a_{ji} \right) x_i \end{aligned} \quad (\text{B.5})$$

mit den Produktionskoeffizienten a_{ji} aufstellen lassen.⁴⁾ Die hier verwendete Verbrauchsfunktion g entspricht einer Produktionsfunktion vom Typ B nach Gutenberg.

Die Summe aus Güterpreisen und Produktionskoeffizienten $\sum_{j=1}^M q_j a_{ji}$ definieren wir als variable Kosten k_i^y einer Einflussgrößeneinheit, so dass sich (B.5) zu

$$K = \sum_{i=1}^I \left(\sum_{j=1}^M q_j a_{ji} \right) x_i = \sum_{i=1}^I k_i^y x_i \quad (\text{B.6})$$

vereinfacht. Die in der Kostenrechnung üblicherweise gesondert ausgewiesenen fixen Kosten werden in (B.6) als (variable) Kosten der Einflussgröße

¹⁾ Vgl. grundlegend Kosiol [Wesensmerkmale] 33 f.

²⁾ Vgl. ähnlich Troßmann [Kostenfunktionen] 1203; vgl. auch Rieper [Fundierung].

³⁾ Vgl. Laßmann [Kostenrechnung] 106 ff.; Schweitzer/Küpper [Systeme] 392 ff.

⁴⁾ Vgl. z. B. Fandel [Kostentheorie] 357 ff.

„Aufrechterhalten der Betriebsbereitschaft“ interpretiert. Über die Definition $K^{\text{fix}} = k_{\text{Bereitschaft}}^{\text{v}} x_{\text{Bereitschaft}}$ gelangt man zur bekannten Formulierung

$$K = K^{\text{fix}} + \sum_{i=1}^I k_i^{\text{v}} x_i. \quad (\text{B.7})$$

Kostenfunktionen der Form (B.6) bzw. (B.7) werden u. a. in der Grenzplankostenrechnung und in der Prozesskostenrechnung verwendet.¹⁾ Der Vorteil dieser Kostenfunktionen liegt darin, dass ihre Einflussgrößen isoliert betrachtet werden können. Die Mehrdimensionalität einer Kostenfunktion stellt keine Schwierigkeit mehr dar, weil die Summanden als einzelne Kostenfunktionen mit nur einer unabhängigen Variablen analysiert werden können.²⁾ Allerdings lassen sich mit der Kostenfunktion (B.7) mehrstufige Leistungsverflechtungen nicht unmittelbar abbilden. Zudem können Interaktionseffekte zwischen den Einflussgrößen (z. B. multiplikative Beziehungen) nicht berücksichtigt werden.³⁾ Wenn außerdem – wie in der Grenzplan- und der Prozesskostenrechnung – die variablen Kostensätze k_i^{v} und damit auch Preise und Produktionskoeffizienten als konstant angenommen werden (siehe (B.6)), enthält die Kostenfunktion nur lineare Zusammenhänge.

2. Kostenfunktionen als theoretische Grundlage der Erklärung und Prognose in der Kostenrechnung

Sowohl Kostentheorie als auch Kostenrechnung untersuchen funktionale Beziehungen zwischen der Kostenhöhe und ihren Einflussgrößen. Dabei

¹⁾ Siehe Abschnitt 4, S. 31 ff.

²⁾ Vgl. Troßmann [Kostenfunktionen] 1202.

³⁾ Vgl. Troßmann [Plankostenrechnung] 234 f. Siehe auch Abschnitt C.III.4.a, S. 147 ff.

fallen ihnen unterschiedliche Aufgaben zu. In der *Kostentheorie* wird allgemein untersucht, wie reale Leistungsprozesse modelliert werden können, um die Kostenentstehung in Betrieben zu *erklären* und mithin zu *prognostizieren*.¹⁾ Die vorgeschlagenen Modelle unterscheiden sich vor allem darin, welche Kosteneinflussgrößen sie berücksichtigen, wie sie die Variablen miteinander verknüpfen und ob sie feste Funktionsverläufe unterstellen. Kostentheoretische Modelle lassen sich in der Regel mit Kostenfunktionen formal darstellen. Eine Kostenfunktion erklärt und prognostiziert die Kostenhöhe durch die Ausprägungen der unabhängigen Variablen.

Während in der Kostentheorie der Anspruch besteht, Zusammenhänge aufzudecken, die universell, also unabhängig von Zeit und Raum gelten,²⁾ hat die *Kostenrechnung* die komplementäre Aufgabe, die konkret anfallenden Kosten in einer spezifischen raum-zeitlichen Situation zu *berechnen*.³⁾ Dadurch ermöglicht sie die Erklärung und Prognose der Kostenhöhe im Einzelfall.

Damit die Modelle der Kostentheorie ihrem universellen Geltungsanspruch gerecht werden können, enthalten sie keine konkreten numerischen Aussagen – beispielsweise über die genauen Kostenwirkungen einzelner Einflussgrößen. Vielmehr zeichnen sie ein abstraktes Bild genereller Kostenstrukturen.⁴⁾ Die zahlenmäßige Bestimmung der Kosten findet erst in der Kostenrechnung statt. Häufig werden dabei Vergangenheitsdaten der Kos-

¹⁾ Vgl. Schweitzer/Küpper [Kostentheorie] 15 f.

²⁾ Das ist bei diesem Betrachtungsgegenstand naturgemäß nur auf einem sehr abstrakten Niveau möglich.

³⁾ Vgl. auch im Weiteren Heinen [Kostenlehre] 38, 144; Pressmar [Kosten- und Leistungsanalyse] 21; Schweitzer/Küpper [Kostentheorie] 16 f.

⁴⁾ Vgl. auch im Weiteren Heinen [Kostenlehre] 144.

tenrechnung genutzt, um die Parameter einer Kostenfunktion zahlenmäßig zu bestimmen.¹⁾

Zwischen den abstrakten, generellen Aussagen der Kostentheorie und den konkreten, situativen Aussagen der Kostenrechnung bestehen wechselseitige Abhängigkeiten. Abb. B-3 veranschaulicht diese Beziehung: In der Kostenrechnung werden in drei Perioden die Kostenhöhen $\{K^{(1)}, K^{(2)}, K^{(3)}\}$ sowie die Ausprägungen $\{x_1^{(1)}, x_1^{(2)}, x_1^{(3)}\}$ einer Größe x_1 ermittelt (Abb. B-3 a). Um die vergangenen Kosten zu erklären und zukünftige zu prognostizieren, bedarf es zusätzlich zu diesen Daten zumindest noch der kostentheoretischen Hypothese, dass K mit x_1 zusammenhängt. Dann können die ermittelten Daten dazu genutzt werden, den Verlauf der Funktion f zu bestimmen und für eine geplante Größe $x_1^{(4)}$ die Kostenprognose $\widehat{K}^{(4)}$ abzuleiten (Abb. B-3 b).

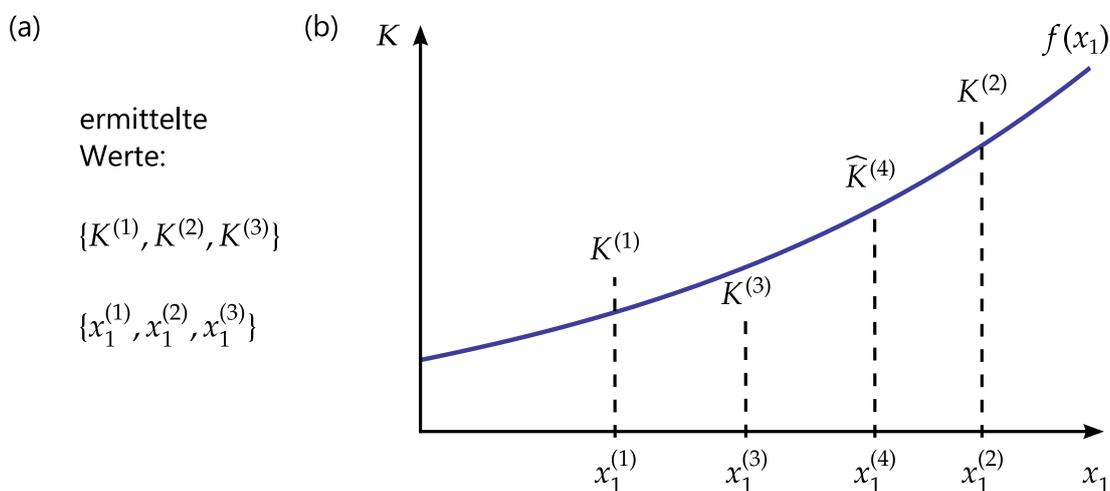


Abb. B-3: Gegenüberstellung von Kostenermittlung (a) und theoriegestützter Kostenerklärung und -prognose (b)²⁾

¹⁾ Siehe Abschnitt C.III.2, S. 133 ff.

²⁾ In Anlehnung an Schweitzer/Küpper [Kostentheorie] 18.

Die Kostenrechnung kann ihre über die reine Dokumentation hinausgehenden Zwecke also nur erfüllen, wenn sie auf eine theoretische Fundierung in Form von Kostenfunktionen zurückgreift.¹⁾ Im besten Fall wird in der Kostenrechnung eine wissenschaftlich begründete Kostentheorie umgesetzt, regelmäßig handelt es sich aber um pragmatische „Ad-hoc-Kostentheorien“. Sie werden vor allem in Anwendungsbereichen gebraucht, die von einer besser begründeten Kostentheorie nicht abgedeckt werden. Da die wissenschaftliche Auseinandersetzung mit der Kostentheorie lange Zeit fast ausschließlich auf die industrielle Produktion ausgerichtet war, gibt es einige solcher Bereiche. Obwohl in der Kostenrechnung auch Zusammenhänge außerhalb des Produktionsbereichs abgebildet werden, gibt es bisher keine einheitliche Kostentheorie für diese Anwendungsfelder.²⁾ Kosteneinflussgrößen außerhalb der Sachgüterproduktion (siehe Abb. B-6, S. 48, und Abb. B-8, S. 52), die seit dem Aufkommen der Prozesskostenrechnung vermehrt in der Kostenrechnungsliteratur vorgeschlagen werden, haben keinen Eingang in eine systematische Kostentheorie gefunden.

Je weniger es möglich ist, Kostenfunktionen aus einer wohlbegründeten Kostentheorie abzuleiten, desto wichtiger wird die statistische Fundierung. Mit statistischen Methoden können Kostenzusammenhänge entdeckt, bestätigt und präzisiert werden.³⁾ Künstliche neuronale Netze erlauben hier in der Regel eine bessere Fundierung als einfachere statistische Verfahren wie z. B. lineare Regressionsanalysen.⁴⁾

¹⁾ Das wird insbesondere bei der genaueren Betrachtung der einzelnen Kostenrechnungszwecke im Führungsprozess in Abschnitt IV (S. 55 ff.) deutlich.

²⁾ Vgl. Schweitzer/Küpper [Kostentheorie] 219 f.

³⁾ Siehe Abschnitt C.III.3.e, S. 134 ff.

⁴⁾ Siehe Abschnitt C.III.4, S. 147 ff.

3. Grenzkosten als Entscheidungsgrundlage

Zur Unterstützung von Entscheidungen reicht es nicht aus, die Kosten der Vergangenheit zu erklären oder einzelne Kostenausprägungen für zukünftige Perioden zu prognostizieren. Im Entscheidungsprozess interessiert vielmehr, wie sich die (prognostizierten) Kosten *ändern*, wenn eine bestimmte Handlungsalternative statt einer anderen realisiert wird. Dieses „Denken in Änderungen“ ist auch in anderen Bereichen der Ökonomie anzutreffen und wird als *Grenzprinzip* oder *Marginalprinzip* bezeichnet.¹⁾

Der Begriff der *Grenzkosten* wird in der Literatur unterschiedlich ausgelegt. *Grenzkosten im weiteren Sinn* beschreiben die absolute Kostenänderung, die infolge einer bestimmten Entscheidung eintritt.²⁾ Sie umfassen die *Differenzkosten* (Schichtkosten, *incremental costs*)³⁾

$$DK = f(\mathbf{x}^{(0)} + \Delta\mathbf{x}) - f(\mathbf{x}^{(0)}) \quad (\text{B.8})$$

mit $\Delta\mathbf{x} = (\Delta x_1, \dots, \Delta x_l)^T$ als Änderung der unabhängigen Variablen, ausgehend vom Punkt $\mathbf{x}^{(0)} = (x_1^{(0)}, \dots, x_l^{(0)})^T$.

Die Grenzkosten i. w. S. sind die disponiblen Kosten einer Entscheidung und deshalb die *entscheidungsrelevanten* Kosten.⁴⁾ In der englischsprachigen Literatur werden sie entsprechend als *relevant costs* bezeichnet.⁵⁾

Grenzkosten im engeren Sinn beziehen sich auf die Kostenänderung, die durch die infinitesimale Variation einer Einflussgröße entsteht, während

¹⁾ Vgl. Loitsberger [Grenzbetrachtungen]; Lücke [Grenzdenken].

²⁾ Vgl. Adam [Grenzkostenrechnung] 824 f.; vgl. auch im Weiteren Männel [Grenz- und Residualkosten] 819.

³⁾ Vgl. Mellerowicz [Theorie der Kosten] 356 f.; Demski [Managerial Uses] 16 f.

⁴⁾ Vgl. auch im Weiteren Adam [Grenzkostenrechnung] 824 f.; Hummel [Relevante Kosten].

⁵⁾ Vgl. Horngren/Datar/Rajan [Cost Accounting] 415 f.

alle anderen Einflussgrößen konstant bleiben. Die Grenzkosten i. e. S einer Einflussgröße x_i an der Stelle $\mathbf{x}^{(0)}$ entsprechen dem Differentialquotienten an dieser Stelle

$$\frac{\partial f}{\partial x_i}(\mathbf{x}^{(0)}). \quad (\text{B.9})$$

Grenzkosten i. e. S und ihre Funktionsverläufe werden in Entscheidungsmodellen analysiert, um Optimalpunkte zu finden. Da in der Realität nicht über infinitesimale Änderungen entschieden wird, sind sie aber keine entscheidungsrelevanten Kosten im oben angeführten Sinn.¹⁾ Näherungsweise relevante Kosten erhält man durch Multiplikation der Differentialquotienten (B.9) mit der absoluten Einflussgrößenänderung Δx :

$$\left(\frac{\partial f}{\partial x_1}(\mathbf{x}^{(0)}), \dots, \frac{\partial f}{\partial x_l}(\mathbf{x}^{(0)}) \right) \begin{pmatrix} \Delta x_1 \\ \vdots \\ \Delta x_l \end{pmatrix} = \left(\nabla f(\mathbf{x}^{(0)}) \right)^T \cdot \Delta \mathbf{x} \quad (\text{B.10})$$

mit $\nabla f(\mathbf{x}^{(0)})$ als Gradient der Funktion f an der Stelle $\mathbf{x}^{(0)}$.

Abb. B-4 illustriert beide Interpretationen des Grenzkostenbegriffs am Beispiel einer Kostenfunktion mit einer unabhängigen Variablen.

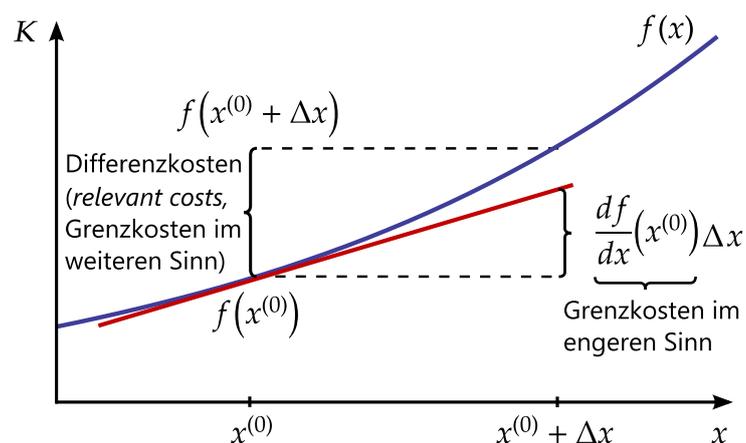


Abb. B-4: Grenzbetrachtungen bei einer nichtlinearen Kostenfunktion

¹⁾ Vgl. Lücke [Grenzdenken] 592 f.

Für eine lineare, additiv separable Kostenfunktion (siehe S. 24) entspricht der Differentialquotient $\partial f/\partial x_i$ an jeder Stelle $\mathbf{x}^{(0)}$ den *variablen Kosten* k_i^v einer Mengeneinheit von Einflussgröße x_i (siehe (B.7), S. 25). Dadurch vereinfacht sich (B.10) zu

$$\left(\nabla f(\mathbf{x}^{(0)})\right)^T \cdot \Delta \mathbf{x} \stackrel{\text{(B.7)}}{=} \sum_{i=1}^I k_i^v \Delta x_i. \quad (\text{B.11})$$

Wie Abb. B-4 zeigt, unterscheiden sich die Differenzkosten (B.8) und die aus Grenzkosten i. e. S. abgeleitete Kostenänderung (B.10), wenn die Kostenfunktion nichtlinear verläuft.¹⁾ Nur bei einer linearen, additiv separablen Funktion f stimmen sie überein. Hinzu kommt, dass Differenzkosten auch bei sprunghaften, diskontinuierlichen Kostenänderungen berechnet werden können, Grenzkosten i. e. S. hingegen nur bei differenzierbaren Kostenfunktionen.²⁾ Grenzkosten nach Formel (B.10) bzw. (B.11) sind also nur im Spezialfall einer linearen Kostenfunktion als Entscheidungsgrundlage geeignet, Grenzkosten nach Formel (B.8) hingegen in jedem Fall.

4. Typische Kostenfunktionen in der Kostenrechnung

a) Kostenspaltung in fixe und variable Kosten als implizite Kostenfunktion

Kosten sind der sachzielbezogene, bewertete Güterverbrauch.³⁾ Dort wo es auf die exakte Kostenerfassung ankommt, ermittelt man zunächst Mengen- und Wertkomponenten des Güterverbrauchs getrennt, um daraus die Kos-

¹⁾ Vgl. Demski [Managerial Uses] 17.

²⁾ Vgl. Adam [Grenzkostenrechnung] 825.

³⁾ Siehe S. 8.

tenhöhe zu berechnen.¹⁾ Diese ausführliche Rechnung findet innerhalb der Kostenrechnung überwiegend in der Kostenartenrechnung statt. In der Kostenstellen- und Kostenträgerrechnung ist eine Aufspaltung der Kostenhöhe in Verbrauchsmengen und Preise zwar auch möglich (siehe (B.5), S. 24); da in einer Kostenstelle oder auf einem Kostenträger aber viele Kostenarten zusammentreffen,²⁾ werden solche differenzierten Analysen schnell unübersichtlich. Deshalb interessiert hier in erster Linie eine aggregierte Kostenfunktion, die nur die Abhängigkeit der Kostenhöhe von den wichtigsten Kosteneinflussgrößen zeigt, wie etwa in Formel (B.7) (S. 25).

Die Abhängigkeit der Kostenhöhe von einer Einflussgröße wird in der Kostenrechnung traditionell nicht in Form von Kostenfunktionen angegeben, sondern durch eine Kategorisierung in fixe und variable Kosten. Variable Kosten ändern sich in Abhängigkeit von einer Einflussgröße, fixe Kosten nicht. Die Kategorisierung in fixe und variable Kosten ist immer relativ zu einer bestimmten Einflussgröße. Wenn die Einflussgröße nicht explizit genannt wird, sind in der Regel beschäftigungsfixe und -variable Kosten gemeint. Die Kostenspaltung (Kostenauflösung) in fixe und variable Bestandteile findet i. A. für jede Kostenart einer Kostenstelle statt.³⁾ So können bei der Weiterverteilung der Kosten zwischen den Kostenstellen und auf Kostenträger die Einflussgrößen als Verrechnungsgrundlage genutzt werden.⁴⁾

Wird die Form der Abhängigkeit nicht näher bestimmt, unterstellt man eine proportionale Änderung von variablen Kosten und Einflussgröße. Dies ä-

¹⁾ Siehe Abschnitt 1, S. 20 ff.

²⁾ Die befragten Unternehmen einer Studie von Schäffer/Weber/Mahlendorf ([Trends] 13) arbeiteten im Durchschnitt mit 213 Kostenarten (100 Kostenstellen im Median).

³⁾ Vgl. Schweitzer/Küpper [Systeme] 405. Eine Übersicht verschiedener Verfahren der Kostenspaltung bieten Kilger/Pampel/Vikas [Plankostenrechnung] 290 ff.; Schweitzer/Küpper [Systeme] 405 ff.; jeweils mit weiteren Nachweisen.

⁴⁾ Vgl. Schweitzer/Küpper [Systeme] 27 f.

ßert sich in einem konstanten Kostensatz k^y je Einflussgrößeneinheit. Wenn in einer Kostenstelle mit variablen Kosten k_1^y je Einheit von x_1 , variablen Kosten k_2^y je Einheit von x_2 und einflussgrößenunabhängigen Kosten K^{fix} gerechnet wird, ist dies äquivalent zur linearen Kostenfunktion $K = K^{\text{fix}} + k_1^y x_1 + k_2^y x_2$. Die Übersetzung von nichtlinearen Funktionen in fixe und variable Kosten gelingt allerdings nur über Umwege: etwa über die Angabe von Kostenelastizitäten¹⁾ oder die stückweise Linearisierung von Kostenfunktionen.²⁾ Unproblematisch ist hingegen die Formulierung multivariater Kostenfunktionen. Die Trennung in fixe und variable Kosten ist für beliebige Einflussgrößen möglich, auch wenn in manchen Kostenrechnungssystemen die Beschäftigung (z. B. Ausbringungsmengen oder Produktionsstunden) die einzige Einflussgröße ist. Zur Entscheidungsfundierung besser geeignet sind multivariate Kostenfunktionen, da sie die verschiedenen Handlungsmöglichkeiten im Betrieb besser abbilden.³⁾

b) Kostenfunktionen der Grenzplankostenrechnung

Ein Schwerpunkt der Grenzplankostenrechnung nach Kilger ist die Planung von Kostenstellenkosten. Dabei kommen additiv separable Kostenfunktionen der Form $K = K^{\text{fix}} + \sum_{i=1}^I k_i^y x_i$ zum Einsatz (siehe (B.7), S. 25).⁴⁾ Die Kosteneinflussgrößen x nennt Kilger „Bezugsgrößen“.⁵⁾ In (B.7) ist der allgemeine Fall dargestellt, dass mehrere Bezugsgrößen notwendig sind, um die Kostenentstehung in einer Kostenstelle zu erklären („heterogene

¹⁾ Vgl. Troßmann [Rechnungswesen] 118 f.

²⁾ Siehe Abschnitt E.III.3, S. 322 ff.; vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 136 f.

³⁾ Siehe Abschnitt 6, S. 52 f. Vgl. auch Schweitzer/Küpper [Kostentheorie] 324, 336.

⁴⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 387.

⁵⁾ Die verwendeten Symbole wurden an die einheitlichen Bezeichnungen in dieser Arbeit angepasst. Kilger bezeichnet Bezugsgrößen hingegen mit B und Ausbringungsmengen mit x .

Kostenverursachung“).¹⁾ Die variablen Kosten je Einflussgrößeneinheit k_i^v sind in der Grenzplankostenrechnung konstant, so dass sich lineare Kostenfunktionen ergeben. Wegen der unterstellten additiven Separabilität können die Kostenfunktionen einfach aufgestellt werden: in allen Kostenstellen wird für jede Bezugsgröße x_i ein fester Kostensatz k_i^v bestimmt. Die Gesamtkosten auf verschiedenen Aggregationsebenen (Kostenstellen, Bereichen, Betrieben) ergeben sich als Summe daraus. Interdependenzen zwischen den Bezugsgrößen bleiben unberücksichtigt.

Die Bezugsgrößen x sollen in der Grenzplankostenrechnung zwei Funktionen erfüllen. Einerseits bestimmen sie in der *Kostenstellenrechnung* den variablen Kostenanteil $k^v x$ in (B.7). Andererseits fungieren sie in der *Kostenträgerrechnung* als Verteilungsschlüssel zur Weiterverrechnung der Kostenstellenkosten auf Kostenträger. In der Terminologie der Grenzplankostenrechnung sind sie „Maßgrößen für die Beschäftigung“.²⁾ Die „Beschäftigung“ ist dabei konkret als Ausbringungsmenge („Leistungsmenge“) einer Kostenstelle zu interpretieren. Wenn eine Kostenstelle verschiedenartige Leistungen erstellt, müssen diese durch geeignete „Maßgrößen der Kostenverursachung gleichnamig gemacht werden.“ Dafür kommen z. B. „Fertigungszeiten, Maschinenlaufzeiten, Durchsatzgewichte, sowie Längen-, Flächen- oder Kubikmaße in Frage.“³⁾ Zwischen den Ausbringungsmengen $x_{A,j}$ verschiedener Leistungsarten j und einer Bezugsgröße x_i besteht der Zusammenhang

¹⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 123 ff.

²⁾ Vgl. auch im Weiteren Kilger/Pampel/Vikas [Plankostenrechnung] 121 f.

³⁾ Kilger/Pampel/Vikas [Plankostenrechnung] 122.

$$x_i = \sum_{j=1}^J b_{ij} x_{A,j} \quad (\text{B.12})$$

mit b_{ij} als Bezugsgrößeneinheiten je Ausbringungseinheit (z. B. Fertigungsminuten je Produktstück).¹⁾

Die Koeffizienten b_{ij} bilden die Grundlage der Bezugsgrößenkalkulation. Die Stückkosten SK_j einer Einheit von Leistung j berechnen sich nach diesem Kalkulationsansatz als Summe aus Einzelkosten und bezugsgrößenabhängigen Kosten

$$SK_j = EK_j + \sum_{i=1}^I k_i^y b_{ij} \quad (\text{B.13})$$

mit Einzelkosten EK_j (Summe aus Materialeinzelkosten, Fertigungseinzelkosten, Sondereinzelkosten).²⁾ Die Kostensätze k_i^y je Bezugsgrößeneinheit sind dieselben wie in der Kostenstellenrechnung (s. o.).

Wenn in Kostenstellen- und Kostenträgerrechnung mit denselben Mengen gerechnet wird, ist die Summe der variablen Kosten einer Kostenstelle identisch mit der Summe der variablen Kosten, die über die Bezugsgrößenkalkulation auf die Ausbringungsmenge verrechnet werden:³⁾

$$\sum_{i=1}^I k_i^y x_i = \sum_{i=1}^I \sum_{j=1}^J k_i^y b_{ij} x_{A,j} \quad (\text{B.14})$$

¹⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 122.

²⁾ Vgl. auch im Weiteren Kilger/Pampel/Vikas [Plankostenrechnung] 520 ff.

³⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 366 f.

In (B.14) werden sämtliche variable Kosten über Bezugsgrößen auf die Beschäftigung in Form der Ausbringungsmengen zurückgeführt. Das wird als *Kostenidentität* bezeichnet.¹⁾

Kostenidentität lässt sich indes nicht für alle Kosteneinflussgrößen verwirklichen. Eine vollständige Kostenidentität, wie sie von Kilger ursprünglich verfolgt wurde, ist aber aus entscheidungsorientierter Sicht auch nicht erstrebenswert. Das betrifft insbesondere Sekundärleistungsbereiche wie z. B. Forschung und Entwicklung, Beschaffung, Absatz, Materialwirtschaft und Logistik oder die kaufmännische Verwaltung;²⁾ Bereiche also, die nicht direkt an der Erstellung von Ausbringungsgütern für einen externen Markt beteiligt sind. Für sie lassen sich zwar durchaus leistungsorientierte Bezugsgrößen – in einer Beschaffungskostenstelle z. B. die Anzahl der bearbeiteten Bestellungen – finden,³⁾ so dass eine outputorientierte Budgetierung⁴⁾ auch für diese Kostenstellen möglich ist. Zur kalkulatorischen Weiterverrechnung auf Erzeugniseinheiten eignen sich diese Bezugsgrößen hingegen oft nicht.⁵⁾

Allerdings können Bezugsgrößen, die sich nicht proportional zur Beschäftigung verhalten, *anderen Bezugsobjekten*, z. B. einem Kundenauftrag, zugerechnet werden. Die Kostenidentität nach Gleichung (B.14) lässt sich wieder herstellen, wenn dann u. a. die Anzahl der Kundenaufträge auf der rechten Seite berücksichtigt wird. Die Kosten eines Kundenauftrags enthalten aber auch die Stückkosten verschiedener Produkte, so dass eine verkettete Funktion entsteht. Je nach dem, welche Bezugsobjekte – beispielsweise auch eine

¹⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 366 f., mit weiteren Nachweisen.

²⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 267 ff.

³⁾ Siehe Abb. B-6 (S. 48). Vgl. auch Kilger/Pampel/Vikas [Plankostenrechnung] 269.

⁴⁾ Siehe Abschnitt IV.6, S. 64.

⁵⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 270.

neue Produktvariante oder beliebige andere Entscheidungsalternativen¹⁾ – betrachtet werden, entsteht eine andere Hierarchie ineinander geschachtelter Bezugsobjekte und damit eine andere verkettete Funktion. Solche *Bezugsobjekthierarchien* hat vor allem Riebel systematisch in die Kostenrechnung eingeführt.²⁾ In der Periodenerfolgsrechnung werden Bezugsobjekthierarchien mit mehrstufigen Deckungsbeitragsrechnungen analysiert. Die Struktur einer Bezugsobjekthierarchie hängt von der interessierenden Fragestellung ab.³⁾

c) Multivariate Kostenfunktionen der Prozesskostenrechnung

Die Prozesskostenrechnung (*activity-based costing*) ist weitgehend äquivalent zur Grenzplankostenrechnung.⁴⁾ Beiden Systemen liegen lineare, additiv separable Kostenfunktionen der Form $K = K^{\text{fix}} + \sum_{i=1}^I k_i^v x_i$ zugrunde (siehe (B.7), S. 25). Zwar werden in der Prozesskostenrechnung Kostenfunktionen für Prozesse und nicht für Kostenstellen aufgestellt, die Struktur der Kostenrechnung ist jedoch dieselbe wie in der Grenzplankostenrechnung, so dass eine Prozesskostenrechnung auch mit Standardsoftware umgesetzt werden kann, die ursprünglich für eine Grenzplankostenrechnung konzipiert ist.⁵⁾ Die Kosteneinflussgrößen x werden in der Prozesskostenrechnung *cost driver* („Kostentreiber“) genannt. Die Kostensätze je Einflussgrößeneinheit k^v heißen *cost driver rates* oder *Prozesskostensätze*.

¹⁾ Siehe Abschnitt IV.3, S. 59.

²⁾ Vgl. Riebel [Einzelkostenrechnung] 94 ff., 160 ff., 178 ff., 406 f.; Schweitzer/Küpper [Systeme] 541 ff.; Fischer [Kostenmanagement] 233 ff., mit weiteren Nachweisen.

³⁾ Vgl. Riebel [Einzelkostenrechnung] 166. Für ein ausführliches Fallbeispiel vgl. Troßmann/Baumeister/Werkmeister [Fallstudien] 50 ff.

⁴⁾ Vgl. auch im Weiteren Kloock [Rückschritt] 183 ff.; Schweitzer/Küpper [Systeme] 479 f.

⁵⁾ Vgl. Kagermann [Standardsoftware]; Kagermann [Einbindung]; Sahl [Integration].

In den historisch grundlegenden Ansätzen von Cooper/Kaplan und Horváth/Mayer wird keine Kostenspaltung¹⁾ in fixe und variable Kosten einer Kostenstelle bzw. eines Prozesses vorgenommen. Stattdessen werden alle Kosten über die Prozesskostensätze k^v auf die *cost driver* verteilt. Zur Berechnung eines Kostensatzes je bearbeiteter Bestellung wird beispielsweise das Jahresgehalt der Mitarbeiter, die im Bestellprozess arbeiten, durch die Anzahl der Bestellungen im Jahr geteilt.²⁾ Fixe Kosten, die von der Anzahl der Bestellungen unberührt bleiben, sind nicht vorgesehen; in den entsprechenden Kostenfunktion fehlt der Summand K^{fix} . Nach Horváth/Mayer sollen daneben noch weitere Fixkosten – so genannte „leistungsmengenun-
trale“ Kosten wie z. B. Kosten der Abteilungsleitung – auf die Prozesskostensätze der *cost driver* umgelegt werden.³⁾ Diese Ansätze einer Prozesskostenrechnung als Vollkostenrechnung erlauben keine verursachungsgerechte Zurechnung der anfallenden Kosten und sind deshalb als Entscheidungsgrundlage ungeeignet.⁴⁾

Die Erweiterung der grundlegenden Ansätze hin zu einer „flexiblen Prozesskostenrechnung“⁵⁾ auf Teilkostenbasis erfordert eine Kostenspaltung in fixe und variable Kosten und einen getrennten Ausweis der fixen Kosten in der Kostenfunktion.⁶⁾ Der einzige Unterschied zur Grenzplankostenrech-

¹⁾ Siehe Abschnitt a, S. 31.

²⁾ Vgl. Cooper/Kaplan [Activity-Based Systems] 2; genauso Cooper/Kaplan [Measure] 99; Horváth/Mayer [Prozeßkostenrechnung] 217; Cooper [Activity-Based Costing 1] 211 ff.

³⁾ Vgl. Horváth/Mayer [Prozeßkostenrechnung] 217.

⁴⁾ Vgl. Franz [Vollkostenidee]; Kloock [Rückschritt] 237 ff.; Riebel [Einzelkostenrechnung] 704 ff. Dagegen Horváth/Mayer [Anmerkungen]; Horváth/Mayer [Prozesskostenrechnung?] 5 f.

⁵⁾ Vgl. Kloock [Rückschritt] 240 ff.; Kloock [Flexible Prozesskostenrechnung].

⁶⁾ Zur Konzeption einer flexiblen Prozesskostenrechnung vgl. Troßmann/Baumeister [Rechnungswesen] 235 ff.; vgl. auch Friedl/Küpper/Pedell [Relevance]; Troßmann/Baumeister/Werkmeister [Fallstudien] 47 ff.

nung besteht dann neben der unterschiedlichen Terminologie nur noch darin, dass grundsätzlich keine Kostenidentität nach Gleichung (B.14) (S. 35) angestrebt wird. *Cost driver* müssen sich nicht proportional zur Beschäftigung verhalten.¹⁾ Die Forderung nach Kostenidentität wurde allerdings auch in der Grenzplankostenrechnung mit der Einführung von Bezugsgrößen für Kostenstellen außerhalb des Fertigungsbereichs abgeschwächt (siehe S. 36).

Problematisch ist allerdings, wenn *cost driver*, die in keinem direkten Zusammenhang zur Beschäftigung stehen, dennoch in die Stück-Kalkulation eingehen.²⁾ Wenn z. B. die Anzahl der Rüstvorgänge als ein *cost driver* gewählt und die Kosten je Rüstvorgang als *cost driver rate* bestimmt wurden, können jeweils einem Fertigungslos Kosten in Höhe der *cost driver rate* zugerechnet werden. Denn die Rüstkosten wären nicht entstanden, wenn das Los nicht produziert worden wäre. Sie können allerdings nicht einer einzelnen Produkteinheit verursachungsgerecht zugerechnet werden.³⁾ Schließlich ändern sich die Rüstkosten nicht, wenn die Losgröße um eine Einheit zu oder abnimmt.⁴⁾

Auch in der Literatur zur Prozesskostenrechnung bzw. zum *activity-based costing* hat sich deshalb die Erkenntnis durchgesetzt, dass Kosten nur in einer Bezugsobjekthierarchie (*cost hierarchy*) korrekt verrechnet werden

¹⁾ Vgl. Cooper [Activity-Based Costing 2] 271.

²⁾ Vgl. Kloock [Rückschritt] 190; Glaser [Entscheidungsrelevanz].

³⁾ Die Verrechnung von Rüstkosten auf Produkteinheiten wird z. B. von Cooper ([Activity-Based Costing 1] 211) vorgeschlagen; genauso Cooper/Kaplan [Measure] 99, Horváth/Mayer [Prozeßkostenrechnung] 219.

⁴⁾ Der Sonderfall der Chargenfertigung sei hier vernachlässigt.

können.¹⁾ Innerhalb einer *cost hierarchy* wird häufig zwischen *unit volume-related* (stückzahlbezogenen), *batch-related* (losbezogenen), *product-related* (produktartbezogenen) und *facility sustaining costs* (Kosten der Betriebsbereitschaft) unterschieden.²⁾ Auf Basis dieser und anderer Bezugsobjekthierarchien sind im System der Prozesskostenrechnung sehr differenzierte mehrstufige Deckungsbeitragsrechnungen für verschiedene Fragestellungen möglich – z. B. Absatzsegmentrechnungen³⁾ mit hierarchisch gegliederten Kundengruppen, Produktlinien oder Absatzregionen.⁴⁾

5. Übersicht relevanter Kosteneinflussgrößen

a) Produktionstheoretisch begründete Kosteneinflussgrößen

Grundsätzlich können in eine Kostenfunktion beliebig viele Einflussgrößen aufgenommen werden. So wird z. B. mit *engineering production functions* und daraus abgeleiteten Kostenfunktionen versucht, technische Zusammenhänge durch eine Vielzahl an Variablen möglichst genau abzubilden.⁵⁾ Häufig ist es jedoch sinnvoll, sich auf die wichtigsten Einflussgrößen zu beschrän-

¹⁾ Vgl. Cooper/Kaplan [Priorities]; Horngren/Datar/Rajan [Cost Accounting] 379; Knöbel [Prozessmanagement]; Anderson/Sedatole [Cost Hierarchy], mit weiteren Nachweisen. Vgl. auch Fischer [Kostenmanagement] 208 ff., mit weiteren Nachweisen. Siehe auch S. 37.

²⁾ Vgl. z. B. Cooper/Kaplan [Priorities] 132. Für eine aktuelle Übersicht empirischer Untersuchungen zu dieser *cost hierarchy* vgl. Anderson/Sedatole [Cost Hierarchy].

³⁾ Vgl. Hoffjan/Reinemann [Absatzsegmentrechnung].

⁴⁾ Vgl. Cooper/Kaplan [Priorities] 134 f.; Fischer [Kostenmanagement] 210 ff., mit weiteren Nachweisen; ausführlich Glaser [Deckungsbeitragsrechnung] 56 ff., 112 ff., 152 ff., mit weiteren Nachweisen; Troßmann/Baumeister/Werkmeister [Fallstudien] 86 f.

⁵⁾ Vgl. z. B. Fandel [Kostentheorie] 195 ff. Ein Beispiel dafür, wie viele Variable bereits zur technisch korrekten Beschreibung relativ einfacher Prozesse notwendig sind, gibt Heinen [Kostenlehre] 569. Vgl. auch Laßmann [Kostenrechnung] 78 ff.

ken. Welche Einflussgrößen im Vordergrund stehen, hängt von der konkreten Entscheidung ab, die fundiert werden soll.¹⁾

Systematische Zusammenstellungen von Kosteneinflussgrößen gibt es vor allem in der Produktions- und Kostentheorie. Kosten werden dort immer als Produkt aus Güterpreisen und Verbrauchsmengen betrachtet (siehe (B.2), S. 22). Dementsprechend sind zentrale Einflussgrößen zum einen die Preise der Einsatzgüter („Faktorpreise“) und zum andern die Einflussgrößen der Produktionsfunktion, die den Güterverbrauch („Faktoreinsatz“) bestimmt (siehe (B.3), S. 22). Abb. B-5 enthält eine Übersicht über Kosteneinflussgrößen, die Gutenberg und Heinen jeweils für eine auf ihren Produktionsfunktionen basierende Kostenfunktion identifiziert haben.²⁾ Zudem sind Kosteneinflussgrößen aufgeführt, die Kilger in Anlehnung an Gutenberg für die Grenzplankostenrechnung vorschlägt.

¹⁾ Siehe Abschnitt 6, S. 52 ff.

²⁾ Für eine ausführliche Diskussion dieser Einflussgrößensysteme vgl. Schweitzer/Küpper [Kostentheorie] 239 ff.

Produktionstheoretisch fundierte Kosteneinflussgrößen

Kosteneinflussgrößen einer Kostenfunktion vom Typ B nach Gutenberg¹⁾

- Beschäftigung, determiniert durch
 - Intensität (intensitätsmäßige Anpassung)
 - Betriebszeit (zeitliche Anpassung)
 - Anzahl und Qualität der eingesetzten Maschinen, Aggregate, Arbeitsplätze und anderer Potentialfaktoren (quantitative und selektive Anpassung)
 - Faktorpreise
 - Faktorqualität (Qualität von Werkstoffen und technischer Ausrüstung, Qualifizierung des Personals)
 - Betriebsgröße
 - Fertigungsprogramm
-

Kosteneinflussgrößen („Kostenbestimmungsfaktoren“) nach Kilger²⁾

- Beschäftigung (Ausbringungsmenge, Leistungsmenge; z. B. Absatz-, Produktions- oder Beschaffungsmenge), gemessen mit Bezugsgrößen, determiniert durch
 - Fertigungszeiten
 - Intensitäten
 - Prozessbedingungen (Druck, Temperatur, Verweilzeiten etc.)
 - Bedienungsrelationen (Anzahl der Maschinen, die von einem Mitarbeiter betreut werden)
 - Rohstoffmischungen
 - Eigenerstellung oder Fremdbezug von Vormaterialien, Einzelteilen, Baugruppen und Arbeitsgängen
 - Auflegungshäufigkeit, Losgrößen, Bestellmengen, Auftragsgrößen
 - ausbringungsabhängig disponierte Kapazitäten betrieblicher Teilbereiche
 - Faktorpreise
 - Faktorqualität (Qualität von Werkstoffen und technischer Ausrüstung, Qualifizierung des Personals)
 - Kapazitäten betrieblicher Teilbereiche und ihr Verschleiß (Abschreibungen)
 - Aufbau zeitungebundener Nutzungspotentiale (Forschung und Entwicklung, Werbemaßnahmen, Aus- und Weiterbildung, Informationssysteme)
-

¹⁾ Vgl. Gutenberg [Produktion] 344 ff.

²⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 115 ff.

Kosteneinflussgrößen einer Kostenfunktion vom Typ C nach Heinen

nicht beeinflussbare Einflussgrößen („Koeffizienten“)¹⁾

- Qualitätsmerkmale der Produktionsfaktoren (z-Situation der Aggregate)
- technisch-physikalische Umweltdaten (Temperatur, Luftfeuchtigkeit)

durch betriebliche Entscheidungen festgelegte Einflussgrößen („Parameter“)²⁾

- Kostenwerte (Faktorpreise)
- Fertigungsprogramm (erstellte Leistungen in ihrer art- und mengenmäßigen Zusammensetzung sowie der zeitlichen Verteilung der Fertigstellungstermine)
- Ausstattung (art- und mengenmäßige Zusammensetzung der Potentialfaktoren sowie ihre räumliche Verteilung (führt zu Übergang auf neue Kostenfunktion))
- Produktionsprozess
 - „Verteilungsparameter“: Arbeitsverteilung, Maschinenbelegung, Wahl der Repetierfaktoren, Lohnfabrikation, Fertigungstiefe, Lagerhaltung im Fertigungsbereich (Zwischenlager)
 - Auflagengröße und die daraus resultierenden Rüst- und Anlaufvorgänge
 - Outputniveau (Ausbringungsmengen)
 - Intensität
 - Leistungsbereitschaft („Anfeuern der Kessel“)
- Kapitalverbrauch³⁾

Abb. B-5: Produktionstheoretisch fundierte Kosteneinflussgrößen

In Gutenbergs Kostenfunktionen ist die Beschäftigung in der Regel die einzige unabhängige Variable. Die übrigen Kosteneinflussgrößen bestimmen die Parameter und damit Lage und Verlauf einer Funktion.⁴⁾ Die Beschäftigung hängt wiederum von Intensität, Betriebszeit sowie Art und Anzahl der Produktionsstellen ab. Intensitätsmäßige, zeitliche und quantitative Anpassung sind über den Zusammenhang

¹⁾ Vgl. Heinen [Kostenlehre] 568 f.

²⁾ Vgl. Heinen [Kostenlehre] 571 ff.

³⁾ Vgl. Heinen [Kapital].

⁴⁾ Vgl. Gutenberg [Produktion] 401, 413, 433.

Beschäftigung = Intensität · Betriebszeit · Menge der Potentialfaktoren

multiplikativ verknüpft.¹⁾ Je nach gewählter Anpassung ergibt sich ein anderer Verlauf der stets univariaten Kostenfunktion mit der Beschäftigung als einziger Kosteneinflussgröße.²⁾

Kilgers System der Grenzplankostenrechnung basiert weitgehend auf Gutenberg-Produktionsfunktionen. Er präzisiert allerdings den Begriff der Beschäftigung und führt Bezugsgrößen zu ihrer Messung ein (siehe S. 34). Wenn in einer Kostenstelle unterschiedliche Leistungsarten erbracht – z. B. verschiedene Produktarten hergestellt – oder Leistungen durch unterschiedliche Verfahrens- und Prozessbedingungen (verschiedene Anpassungsformen) realisiert werden können, ist es i. A. notwendig, die Beschäftigung durch mehrere Bezugsgrößen abzubilden. So entstehen multivariate Kostenfunktionen. Kilger bezeichnet diesen Fall als „heterogene Kostenverursachung“.³⁾ Die Determinanten der Beschäftigung ergänzt Kilger um weitere Einflussgrößen.

Heinen führt die Beschäftigung nicht explizit als Einflussgröße auf, da er durch ein differenziertes Einflussgrößensystem direkt darstellen will, welche Variablen bei einer Beschäftigungsänderung angepasst werden.⁴⁾ Er geht also grundsätzlich von multivariaten Kostenfunktionen aus. Dabei berücksichtigt er neben intensitätsmäßigen, zeitlichen und quantitativen Anpassungsformen weitere Handlungsvariable des Produktionsprozesses.

¹⁾ Vgl. Gutenberg [Produktion] 356.

²⁾ Vgl. Gutenberg [Produktion] 389. Ausführliche Analysen der Kostenverläufe bei verschiedenen isolierten und kombinierten Anpassungsformen finden sich bei Corsten/Gösinger [Produktionswirtschaft] 145 ff.; Fandel [Kostentheorie] 357 ff.; Schweitzer/Küpper [Kostentheorie] 308 ff.

³⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 123 ff.

⁴⁾ Vgl. auch im Weiteren Heinen [Kostenlehre] 580.

Gutenbergs Faktorqualität geht in der Wahl der Repetier- und der Potentialfaktoren sowie einigen „Koeffizienten“ auf.

b) Kosteneinflussgrößen in verschiedenen Funktionsbereichen

Riebel und Kilger haben schon früh Bezugsgrößen vorgeschlagen, die über den Produktionsbereich hinaus anwendbar sind.¹⁾ In der Kostenrechnung etabliert haben sie sich allerdings erst mit der Prozesskostenrechnung. Die *cost driver* der Prozesskostenrechnung sind nicht kosten- oder produktions-theoretisch begründet, sondern werden in einem „kreativen“ Auswahlverfahren passend zu den betrachteten Abläufen im Betrieb gewählt.²⁾ Dabei werden die Prozessverantwortlichen und andere Führungskräfte nach relevanten Einflussgrößen gefragt. Bei der Auswahl werden auch Auswirkungen auf das Verhalten der Mitarbeiter berücksichtigt.³⁾ Abb. B-6 gibt eine kursorische Übersicht messbarer Kosteneinflussgrößen in verschiedenen Funktionsbereichen.⁴⁾ Die meisten Beispiele stammen aus praktischen Erfahrungsberichten zur Einführung einer Prozesskostenrechnung.⁵⁾

Viele der in Abb. B-6 aufgeführten Einflussgrößen sind Maßgrößen für die Leistung einer Kostenstelle und damit Bezugsgrößen im Sinne Kilgers (siehe S. 34). In der Regel messen sie allein die Quantität der Leistung – z. B. wie oft ein standardisierter Arbeitsvorgang ausgeführt wurde.

¹⁾ Vgl. Riebel [Einzelkostenrechnung] 16; Kilger/Pampel/Vikas [Plankostenrechnung] 269. Siehe auch Abb. B-6.

²⁾ Vgl. auch im Weiteren Coenenberg/Fischer [Prozeßkostenrechnung] 26, mit weiteren Nachweisen.

³⁾ Vgl. Cooper [Activity-Based Costing 2] 278. Siehe auch Abschnitt II.2, S. 14 ff.

⁴⁾ Eine umfangreiche Übersicht zu konzeptioneller und empirischer Cost-Driver-Literatur geben Banker/Johnston [Cost Driver Research].

⁵⁾ Vgl. dazu die einzelnen Quellennachweise in Abb. B-6.

Kosteneinflussgrößen in verschiedenen Funktionsbereichen

Einkauf	<ul style="list-style-type: none"> - Anzahl abgewickelter Bestellungen^{2),3),5),6)} <ul style="list-style-type: none"> • Anzahl der Abrufe über Rahmenverträge^{6),10)} • Anzahl der Einzelbestellungen/Bestellpositionen^{4),6)} - Anzahl bearbeiteter Angebote²⁾ - Anzahl geprüfter Rechnungen²⁾ - Anzahl fremdbezogener Einzelteile³⁾ - Anzahl der Rahmenverträge^{4),6)} - Anzahl beauftragter Lieferanten⁶⁾ - Investitionsvolumen¹⁰⁾
Wareneingang	<ul style="list-style-type: none"> - Anzahl der Wareneingänge/Anlieferungen^{3),4),10)} <ul style="list-style-type: none"> • Anzahl der Einzelteil-Sendungen³⁾ • Anzahl der Rohmaterial-Sendungen³⁾ - Anzahl der Lieferscheinpositionen^{5),7)} - Anzahl angelieferter Ladungsträger/Behälter^{7),9)} - Anzahl der Eingangsprüfungen⁵⁾
Lagerhaltung	<ul style="list-style-type: none"> - Anzahl der Lagerbewegungen¹⁰⁾ <ul style="list-style-type: none"> • Anzahl der Zugänge^{2),5)} • Anzahl der Abgänge^{2),5)} - beanspruchter Lagerraum in m³, l oder hl^{2),5)} - beanspruchte Lagerfläche in m²^{2),8)} - mengenmäßiger durchschnittlicher Lagerbestand^{2),9)} - wertmäßiger durchschnittlicher Lagerbestand^{2),9)}
Logistik	<ul style="list-style-type: none"> - Anzahl der Fertigungsaufträge^{1),3),4),6),7),9)} - Anzahl verbauter Teilearten <ul style="list-style-type: none"> • Anzahl der Teilenummern^{3),6),8)} <ul style="list-style-type: none"> ▪ Anzahl der Sachnummern für „Kaufteile“⁷⁾ ▪ Anzahl der Sachnummern für eigengefertigte Teile⁷⁾ • Anzahl der Bauplan-/Stücklistenpositionen^{5),6)} • Anzahl der Flachbaugruppen, Axial-, Radial und DIP-Bestückungen³⁾ • Anzahl der A-, B- und C-Teile⁹⁾ - Anzahl Behälter⁹⁾ <ul style="list-style-type: none"> • Anzahl angelieferter Ladungsträger⁷⁾ • Anzahl betriebsinterner Ladungsträger⁷⁾ - Umsatzerlöse^{3),10)} - geleistete Fahrerstunden⁸⁾

Forschung und Entwicklung	<ul style="list-style-type: none"> - Anzahl der Produktänderungen^{4),6)} <ul style="list-style-type: none"> • Zahl der Konstruktionsänderungen⁸⁾ • Zahl der Zeichnungsänderungen⁸⁾ • Zahl der Stücklistenänderungen⁸⁾ - Anzahl der Konstruktionszeichnungen¹⁾ - Anzahl der Analysen²⁾ - Teststunden³⁾ - Anzahl neuer Produkte^{4),6)} - Anzahl der Neuteile⁶⁾
---------------------------	---

Produktion	<ul style="list-style-type: none"> - Anzahl der Fertigungsaufträge^{1),3),4),6),7),9)} - Anzahl verbauter Teilearten <ul style="list-style-type: none"> • Anzahl der Teilenummern^{3),6),8)} <ul style="list-style-type: none"> ▪ Anzahl der Sachnummern für „Kaufteile“⁷⁾ ▪ Anzahl der Sachnummern für eigengefertigte Teile⁷⁾ • Anzahl der Bauplan-/Stücklistenpositionen^{5),6)} • Anzahl der Flachbaugruppen, Axial-, Radial und DIP-Bestückungen³⁾ • Anzahl der A-, B- und C-Teile⁹⁾ - Anzahl der Rüstvorgänge/Auflegungshäufigkeit^{3),5),9)} - Anzahl der Varianten/Produktarten^{4),6),8)} - Anzahl der Montage-/Arbeitsplanpositionen^{5),6)} <ul style="list-style-type: none"> • Anzahl der Lötvorgänge³⁾ • Anzahl der Handbestückungen³⁾ - Anzahl Sorten-, Typen-, Chargen-, Individualaufträge¹⁾ - direkte Arbeitsstunden³⁾ - Maschinenstunden³⁾ - Rüststunden³⁾ - Anzahl der Spezialkomponenten je Los³⁾
------------	--

Qualitätskontrolle	<ul style="list-style-type: none"> - Anzahl der (Stich-)Proben^{1),2),7),8)} - Anzahl der Arbeitsberichte³⁾
--------------------	--

Versand/ Kommissionierung	<ul style="list-style-type: none"> - Anzahl der Versandaufträge/Lieferungen^{2),4),6),9)} <ul style="list-style-type: none"> • Anzahl der Zollsendungen⁵⁾ • Anzahl der Retourenausgänge⁵⁾ • Anzahl der Frachtbriefe⁵⁾ - Anzahl kommissionierter Auftragspositionen/ Versandpositionen^{6),7)} - Anzahl versandter Behälter/Ladungsträger^{7),9)}
------------------------------	--

Vertrieb	<ul style="list-style-type: none"> - Anzahl bearbeiteter Kundenaufträge^{1),2),3),4),5)} - Umsatzerlöse^{3),10)}
----------	--

	- Anzahl der Kundenbesuche ¹⁾
	- Anzahl Vor- und Nachkalkulationen ²⁾
	- Absatzmengen ³⁾
	- Anzahl gestellter Rechnungen ⁵⁾
	- Anzahl aktiver Kunden ⁶⁾
<hr/>	
Buchhaltung	- Anzahl der Buchungen ^{1),2)}
	- Anzahl Lohnabrechnungen ^{2),6)}
	- Anzahl der Mitarbeiter ⁶⁾
	- Anzahl abgerechneter Kostenstellen ⁶⁾
	- Anzahl zu pflegender Konten ⁸⁾

Abb. B-6: Kosteneinflussgrößen in verschiedenen Funktionsbereichen

Einige Einflussgrößen aus Abb. B-6 sind in mehreren Funktionsbereichen wirksam. Sie zählen überwiegend zu den so genannten „Komplexitätstreibern“. Zunehmende Variantenvielfalt und andere Komplexitätsgrößen führen zu Kostensteigerungen in vielen Unternehmensbereichen.¹¹⁾ Es ist ein erklärtes Ziel der Prozesskostenrechnung, diese Kostenwirkungen über geeignete *cost driver* adäquat zu erfassen. Abb. B-7 fasst einige in Frage kommende Komplexitätskennzahlen zusammen.

¹⁾ Vgl. Riebel [Einzelkostenrechnung] 16.

²⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 269.

³⁾ Vgl. Cooper [Activity-Based Costing 3] 348 ff.

⁴⁾ Vgl. Mayer [Prozeßkostenrechnung] 309 f.

⁵⁾ Vgl. Coenberg/Fischer [Prozeßkostenrechnung] 27.

⁶⁾ Vgl. Horváth/Mayer [Konzeption] 21 f., Nachdruck in Horváth/Mayer [Prozesskostenrechnung?] 6.

⁷⁾ Vgl. Hardt [Mercedes] 337.

⁸⁾ Vgl. Rosenberg [Komplexitätsmanagement] 242.

⁹⁾ Vgl. Weber [Logistikkostenrechnung] 268 ff.

¹⁰⁾ Vgl. Mengen/Urmersbach [Thyssen-Krupp] 223 ff.

¹¹⁾ Vgl. auch im Weiteren Fischer [Variantenvielfalt] 28 f.; Lingnau [Variantenvielfalt]; Coenberg/Fischer [Prozeßkostenrechnung] 32; jeweils mit weiteren Nachweisen.

Komplexitätstreiber

Produktkomplexität

- Anzahl der Sachnummern für Einsatzgüter (\approx Anzahl der Stammdatensätze für Rohstoffe, Einzelteile und Baugruppen)
- Anzahl der Ebenen einer Strukturstückliste (\approx Anzahl von Arbeitsgängen/ Teilprozessen zur Erstellung eines Produkts/einer Leistung)
- Anzahl der Teilenummern in einer Stückliste (\approx Anzahl der Teile, aus denen ein Produkt besteht)

Beschaffungskomplexität

- Anzahl der Zulieferer
- Anzahl der Teilenummern, die 80 % der eingesetzten Güter ausmachen
- Anzahl der Bestellungen

Prozesskomplexität

- Anteil der Fremdleistungen (Lohnbearbeitungen, *subcontracting*) unter den Gütereinkäufen
- Anzahl der Materialfluss-Transaktionen

Komplexität der Programmbreite (Variantenvielfalt)

- Anzahl der Produkte auf den Preislisten für externe Kunden
- Anzahl der Produkte, die 80 % der Erträge ausmachen
- Anzahl abgesetzter/lieferbarer Varianten¹⁾
- Anzahl abgesetzter/lieferbarer Zubehör- und Ersatzteile
- Anzahl neu eingeführter Produkte

Vertriebskomplexität

- Anzahl bearbeiteter Kundenbestellungen
- Gesamtzahl der Kunden

Abb. B-7: Kosteneinflussgrößen zur Abbildung betrieblicher Komplexität („Komplexitätstreiber“)²⁾

¹⁾ Zur Berechnung der Variantenzahl bei frei kombinierbaren Produktmerkmalen (Produktkonfiguration) vgl. Rosenberg [Komplexitätsmanagement] 228 f.

²⁾ Vgl. Foster/Gupta [Cost Driver Analysis] 323 ff.; ähnlich Fischer [Variantenvielfalt] 30; Rosenberg [Komplexitätsmanagement] 242. Die aufgeführten Größen korrelieren bei Foster/Gupta signifikant mit den *manufacturing overhead costs*.

c) Spezielle Kosteneinflussgrößen für Dienstleistungsunternehmen

Einige der oben in Abb. B–6 aufgeführten Einflussgrößen sind auch für Betriebe relevant, die keine Sachgüter produzieren. Das gilt insbesondere für die genannten Einflussgrößen in Vertrieb und Buchhaltung. Andere Einflussgrößen sind für Nicht-Industriebetriebe eher ungeeignet. Doch Abb. B–8 zeigt beispielhaft, dass sich auch für Dienstleistungsunternehmen Kosteneinflussgrößen finden lassen, die den Leistungsprozess abbilden.

Kosteneinflussgrößen für Dienstleistungsunternehmen

Kosteneinflussgrößen in Versicherungen:¹⁾

- Anzahl bearbeiteter Anträge
- Anzahl bearbeiteter Schäden
- Anzahl der EDV-Verfügungen
- Anzahl individueller Korrekturen
- Anzahl manueller Mitteilungen
- Anzahl individuell geschriebener A4-Seiten
- Anzahl der Rücksprachen mit der Schadensbearbeitung
- Anzahl laufender Gerichtsprozesse
- Gesamtzahl der Verträge im Bestand
- Höhe der Versicherungssummen
- Anzahl geänderter Vertragselemente/Stornierungen
- Anzahl der Prämienrechnungen
- Anzahl der Postausgänge

Kosteneinflussgrößen in Banken:²⁾

- Anzahl der Überweisungen
 - Anzahl der Schaltertransaktionen
 - Anzahl der Kundenanrufe
-

¹⁾ Vgl. Neugebauer [Versicherungsunternehmen] 220 f.; Rendenbach [Assekuranz] 285 f.

²⁾ Vgl. Smith Bamber/Hughes II [Buckeye] 385.

Kosteneinflussgrößen für IT-Leistungen:¹⁾

- Anzahl laufender Projektberatungen
 - Anzahl bearbeiteter Jobs
 - Datenaufkommen im Netzwerk
 - Anzahl der Serverlogins
 - betreuter Datenspeicher in Gigabyte
 - Anzahl betreuter PCs
 - Anzahl abgehaltener Trainingsstunden
 - Anzahl verwendeter Softwarelizenzen
 - Anzahl der Netzwerkanschlüsse
 - Anzahl der Codeänderungen/aufgewendete Arbeitsstunden für Codeänderungen
 - Anzahl der Wartungsfälle/aufgewendete Arbeitsstunden für Wartung
 - Anzahl durchgeführter Updates/aufgewendete Arbeitsstunden für Updates
 - Anzahl in Betrieb genommener Endgeräte
 - Anzahl bearbeiteter Supportanfragen
-

Kosteneinflussgrößen im Gesundheitsbetrieb:²⁾

- Anzahl der Patienten einer bestimmten Fallgruppe (DRG) oder Pflegekategorie
 - Anzahl der Notfallpatienten
 - Anzahl betreuter Betten
 - geleistete Pflegetage
 - Anzahl angeforderter Laboruntersuchungen
 - Anzahl der Stationen
-

Kosteneinflussgrößen im Event Marketing:³⁾

- Anzahl durchgeführter Events
 - Eventtage insgesamt
 - Komplexität der Events
 - Anzahl aufgetretener Probleme
 - Anzahl der Programmänderungen/Upgrades
 - Anzahl der Dienstreisen
-

¹⁾ Vgl. Gerlinger/Buresch/Krcmar [IV-Leistungsverrechnung] 125; Neumann u. a. [IT Activities] 33; vgl. auch Uebernickel/Scheeg/Brenner [T-Systems] 43 ff.

²⁾ Vgl. Güssow/Greulich/Ott [DRGs] 184; Frodl [Gesundheitsbetrieb] 94 f.

³⁾ Vgl. Gauharou [Client Services] 8 f.

Kosteneinflussgrößen für Fluggesellschaften:¹⁾

- Anzahl der Fluggäste
 - abgesetzte Passagiermeilen (*revenue passenger miles*)²⁾
 - Fracht- und Passagier-Kapazität (*capacity seat miles*)
 - Flugzeugtyp
 - Flugzeuggröße und geflogene Entfernung (bestimmen die Losgröße der abgesetzten Passagiermeilen)
 - Flughäufigkeit (*density*)
 - Anteil der Flüge, die an der „Nabe“ eines Hub-and-Spoke-Netzwerks starten oder landen (*hub concentration*)
-

Abb. B–8: Kosteneinflussgrößen für Dienstleistungsunternehmen

6. Entscheidungsabhängige Auswahl der Kosteneinflussgrößen

Wenn Kostenfunktionen entscheidungsunterstützende Informationen liefern sollen, müssen sie die verschiedenen Handlungsmöglichkeiten des Entscheidungsträgers abbilden.³⁾ Die Möglichkeiten der Kostenbeeinflussung erschöpfen sich in der Regel nicht im Festlegen einer einzelnen Variablenausprägung (z. B. der Produktionsmenge), sondern ergeben sich aus einer Vielzahl denkbarer Einflussgrößenkombinationen. Beispielsweise kann eine größere Produktionsmenge durch längere Arbeitszeiten, eine höhere Fertigungsintensität oder eine Kombination aus beidem erreicht wer-

¹⁾ Vgl. Banker/Johnston [Airline Industry] 579 ff.

²⁾ Eine Passagiermeile entspricht dem Transport eines Fluggastes und seines Gepäcks über eine Entfernung von einer Meile (vgl. Banker/Johnston [Airline Industry] 579).

³⁾ Vgl. auch im Weiteren Heinen [Kostenlehre] 143; Schweitzer/Küpper [Kostentheorie] 324.

den. Diese Einflussgrößen sollten deshalb als unabhängige Variablen betrachtet werden. In einem anderen Entscheidungszusammenhang, wenn z. B. über die Funktionen und Komponenten eines neuen Produkts entschieden wird, interessieren eher die technischen Gestaltungsvariablen des Produkts – die Variablen des Produktionsprozesses sind hier Randbedingungen, die zur Vereinfachung konstant gesetzt werden können.¹⁾ Diese Beispiele zeigen, dass es von der jeweiligen Entscheidung abhängt, welche Einflussgrößen als unabhängige Variablen in die Kostenfunktion aufgenommen werden.

Tendenziell kann eine Kostenfunktion mit vielen unabhängigen Variablen die verschiedenen Handlungsmöglichkeiten detaillierter abbilden und ist deshalb besser als Entscheidungsgrundlage geeignet. Allerdings wird es mit steigender Variablenzahl auch aufwendiger, die Kostenfunktion zu ermitteln und auszuwerten. Deshalb ist man bestrebt, die Anzahl unabhängiger Variablen möglichst gering zu halten. Zu diesem Zweck werden Einflussgrößen, die mit anderen korrelieren, aus dem Modell ausgeschlossen.²⁾ In einem weiteren Schritt können Variablen, die nur einen schwachen Einfluss auf die Kostenhöhe haben, ausgesondert werden.³⁾

Aus entscheidungsorientierter Sicht ist es nicht sinnvoll, die Anzahl der Variablen dadurch zu verringern, dass bestimmte Entscheidungen *a priori* unterstellt werden. So könnte man die multivariate Kostenfunktion

$$K = f(x_\lambda, x_\tau) \quad (\text{B.15})$$

mit x_λ als Fertigungsintensität und x_τ als Fertigungszeit auf eine univariate Kostenfunktion reduzieren, wenn man annimmt, dass bis zur maximalen

¹⁾ Vgl. auch im Weiteren Schweitzer/Küpper [Kostentheorie] 268.

²⁾ Siehe Abschnitt D.V.2, S. 193 ff.

³⁾ Siehe Abschnitt E.I.1, S. 300 ff.

Fertigungszeit x_τ^{\max} immer mit optimaler Intensität x_λ^{opt} gefertigt wird.¹⁾ Damit ist die Ausbringungsmenge $x_A = x_\lambda x_\tau$ stets durch eine eindeutige Kombination aus Intensität und Fertigungszeit gegeben, so dass die Kostenfunktion nur noch von ihr abhängt:

$$K = \tilde{f}(x_A) = \begin{cases} f(x_\lambda^{\text{opt}}, x_\tau) , & x_A < x_\lambda^{\text{opt}} x_\tau^{\max} \\ f(x_\lambda, x_\tau^{\max}), & x_A \geq x_\lambda^{\text{opt}} x_\tau^{\max} \end{cases} \quad (\text{B.16})$$

Diese Vermischung von Erklärung und Gestaltung, wenn Kostenfunktionen zugleich formuliert und optimiert werden, gelingt nur unter starken zusätzlichen Annahmen.²⁾ Dabei werden viele praktische Probleme ausgeklammert – etwa dass neben der Kostenminimierung auch andere Ziele verfolgt werden. Darum sollte die Entscheidung für eine bestimmte Einflussgrößenkombination nicht in der Kostenfunktion vorweggenommen, sondern dem Entscheidungsträger vorbehalten werden. So bleiben die Handlungsmöglichkeiten sichtbar und die Kostenfunktion kann im weiteren Entscheidungsprozess trotzdem noch als Zielfunktion oder Nebenbedingung in konkrete Optimierungsmodelle eingehen.³⁾

¹⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 124 f.; Schweitzer/Küpper [Kostentheorie] 312 ff.; Schweitzer/Küpper [Systeme] 281 ff.

²⁾ Vgl. auch im Weiteren Heinen [Kostenlehre] 143, mit weiteren Nachweisen.

³⁾ Vgl. zur Anwendung von Kostenfunktionen in Entscheidungsmodellen Schweitzer/Küpper [Kostentheorie] 337 ff., mit zahlreichen weiteren Nachweisen.

IV. Anwendungsfälle von Kostenfunktionen für Kostenrechnungszwecke

1. Einordnung der Anwendungsfälle in den Führungsprozess

In Abschnitt II wurden als wesentliche Zwecke der Kostenrechnung die Entscheidungsunterstützung und die Unterstützung der verhaltensorientierten Steuerung herausgestellt. Diese Zwecke werden im Folgenden entlang der Phasen des Führungsprozesses¹⁾ aufgegliedert, um detailliert zu zeigen, in welcher Weise Kostenfunktionen zur Erfüllung der Kostenrechnungszwecke beitragen. Dafür werden in den folgenden Abschnitten zahlreiche Anwendungsfälle von Kostenfunktionen zur Führungsunterstützung beschrieben. Einen Überblick gibt Abb. B–9.

In der Softwareentwicklung dienen Anwendungsfälle (*use cases*) dazu, die Anforderungen an ein Softwaresystem zu erfassen und in Teilfunktionen zu gliedern.²⁾ In diesem Sinne enthalten die im Folgenden erörterten Anwendungsfälle die Anforderungen, denen ein System zur Schätzung von Kostenfunktionen gerecht werden soll. In Abschnitt D.IV.3 (S. 178 ff.) werden sie in ein Anwendungsfalldiagramm nach den Normen der *unified modeling language* überführt (siehe Abb. D–5, S. 181). Daraus wird in den Kapiteln D und E eine Software entwickelt, die diesen Anforderungen genügt.

¹⁾ Siehe Abschnitt II.1, S. 11 ff.

²⁾ Vgl. Seemann/von Gudenberg [Software-Entwurf] 15 ff.

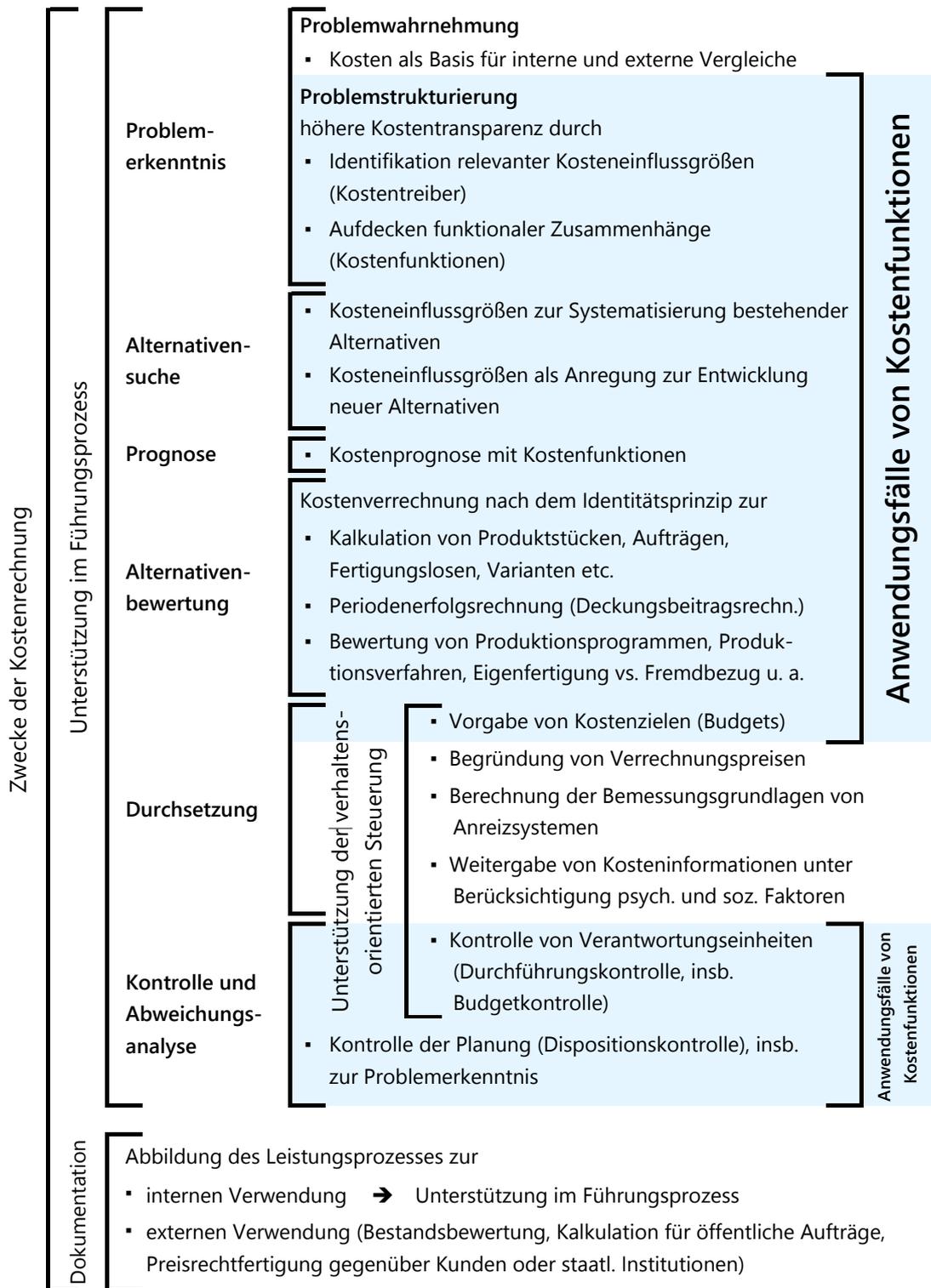


Abb. B-9: Anwendungsfälle von Kostenfunktionen in der Kostenrechnung

2. Problemerkennntnis durch Kostenfunktionen

Probleme sind Abweichungen des gegenwärtigen oder erwarteten Ist-Zustands vom angestrebten Zustand (siehe Abb. B-2, S. 12).¹⁾ Ein Problem wird demnach erst durch einen Soll-Ist-Vergleich **wahrgenommen**. Oft findet dieser Vergleich unbewusst statt. In der Betriebswirtschaftslehre haben sich aber auch einige institutionalisierte Vergleichsinstrumente etabliert – wie z. B. Frühwarn-, Kontroll-, Überwachungs- und Berichtssysteme. Außerdem basieren zahlreiche Techniken der strategischen Planung auf expliziten Soll-Ist-Vergleichen.²⁾ Häufig erleichtern Rückkopplungsinformationen aus der Kontrollphase die Problemerkennntnis.

Die Kostenrechnung liefert quantitative, verdichtete Informationen über betriebliche Vorgänge und bildet sie auf eine einheitliche Bewertungsskala ab. Dadurch werden sie vergleichbar. So können z. B. die Kosten eines Monats mit denen des Vormonats verglichen werden, oder die „vom Markt erlaubten Kosten“ für ein bestimmtes Produkt mit den eigenen Herstellkosten (*target costing*), oder die eigenen Kosten mit denen der Mitbewerber (*benchmarking*). Diese Beispiele zeigen, dass Kostenrechnungsinformationen sowohl in die operative als auch in die taktische sowie die strategische Planung einfließen können.

Mit der Problemwahrnehmung wird ein genereller Handlungsbedarf aufgedeckt. Welche konkreten Handlungsalternativen sich bieten, um eine Wiederholung der Abweichung in Zukunft zu vermeiden, wird aber regelmäßig erst nach der **Problemstrukturierung** deutlich. Darunter versteht man die hierarchische Ordnung von Teilproblemen nach ihren Ursache-

¹⁾ Vgl. Wild [Unternehmensplanung] 66.

²⁾ Vgl. Wild [Unternehmensplanung] 148; Schweitzer [Planung] 85.

Wirkungs-Beziehungen und ihrer Dringlichkeit.¹⁾ Eine Voraussetzung für die Strukturierung von Kostenabweichungen ist die Bestimmung von *Kosteneinflussgrößen* (*Bestimmungsgrößen, Kostentreibern*) und ihren regelmäßigen Zusammenhängen mit der Kostenhöhe. Nur so lassen sich die Abweichungsursachen erkennen. Um eine problematische Kostenabweichung zu strukturieren, reicht es nicht aus, zu wissen wie hoch die Kostenabweichung ist. Es muss außerdem bekannt sein, welche „Kostentreiber“ in welcher Höhe für die Abweichung verantwortlich sind.

Zur Problemstrukturierung eignen sich am besten quantitative Informationen über Kostenzusammenhänge in Form von *Kostenfunktionen*. Die Einflussgrößen der Kostenfunktionen sind Ansatzpunkte, um die Abweichung zwischen Soll- und Ist- bzw. Wird-Kosten zu verringern, d. h. die „Problemlücke“ zu schließen.²⁾ Wenn es möglich ist, die Höhe der Kosteneinflussgrößen zu verändern – wenn die Einflussgrößen also Entscheidungsvariablen sind, können die Kosten durch *Kostenmanagement* zielgerichtet beeinflusst werden.³⁾ „Die Kosteneinflussgrößen als Bestimmungsfaktoren der Kostenhöhe sind die Gestaltungsparameter, über die das Kostenmanagement Einfluss auf die Kosten nimmt.“⁴⁾ Die Kostenrechnung enthält die Informationen, die das Kostenmanagement nutzt, und ist in diesem Sinne das Fundament des Kostenmanagements.⁵⁾

Kostenfunktionen erhöhen demnach – insbesondere indem sie relevante Kostentreiber als Ansatzpunkte des Kostenmanagements offenlegen – die

¹⁾ Vgl. Wild [Unternehmensplanung] 69; Troßmann [Controlling] 76.

²⁾ Vgl. Schweitzer [Planung] 76 f.

³⁾ Die Begriffe *Kostenmanagement, Kostenpolitik, Kostengestaltung* und *Kostensteuerung* sind nicht klar zu trennen. Für eine Literaturübersicht zu diesem Begriffsfeld vgl. Friedl [Kostenmanagement] 1 ff.

⁴⁾ Friedl [Kostenmanagement] 411; vgl. auch Himme [Kostenmanagement] 1065.

⁵⁾ Vgl. Fischer [Basis]; Franz/Kajüter [Proaktives Kostenmanagement] 7.

Transparenz der Kostenstruktur. In Umfragen wird die „Erhöhung der Kostentransparenz“ regelmäßig als herausragend wichtiges Ziel des Kostenmanagements und der Kostenrechnung genannt.¹⁾ Dass kostenrechnerische Informationen für die Problemerkennntnis bedeutend sind, belegt auch eine Studie von Homburg u. a., die zeigt, dass viele Manager die Kostenrechnung hauptsächlich *konzeptionell* nutzen, d. h. um sich einen Überblick über die Zusammenhänge im Betrieb zu verschaffen und geschäftliche Situationen besser zu verstehen.²⁾ „Die Kostenrechnungsinformationen führen zwar in diesem Fall nicht zu konkreten Entscheidungen, beeinflussen aber Denkprozesse und Haltungen der Manager.“³⁾ Die *instrumentelle* Anwendung auf ein konkretes Prognose- oder Bewertungsproblem findet signifikant seltener statt. Auch wenn man von der Nutzungshäufigkeit nicht auf die Wichtigkeit einer Anwendung schließen kann, wird doch deutlich, dass in der Praxis nicht unbedingt Kostenprognosen und Alternativenbewertungen die dominierenden Kostenrechnungszwecke sein müssen.

3. Alternativensuche mit Kostenfunktionen

In der Phase der Alternativensuche werden Handlungsalternativen, die ein erkanntes Problem lösen sollen, formuliert und voneinander abgegrenzt.⁴⁾ Dazu werden die Variablen herangezogen, die ein Entscheidungsträger durch seine Alternativenwahl aktiv beeinflussen kann.⁵⁾ Formal ist eine Alternative ein Tupel von Ausprägungen dieser Entscheidungsvariablen. Im

¹⁾ Vgl. Franz/Kajüter [Befunde] 574; Kajüter [Unternehmenspraxis] 85; Feldbauer-Durstmüller u. a. [Kostenmanagement] 112; Friedl u. a. [Stand] 115.

²⁾ Vgl. Homburg u. a. [Kundenorientierung].

³⁾ Homburg u. a. [Kundenorientierung] 251.

⁴⁾ Vgl. Wild [Unternehmensplanung] 70.

⁵⁾ Vgl. auch im Weiteren Laux/Gillenkirch/Schenk-Mathes [Entscheidungstheorie] 31.

Gegensatz dazu werden Kombinationen von nicht beeinflussbaren Variablenausprägungen als Umweltzustände bezeichnet.

Häufig ist es sinnvoll, nicht unmittelbar die Kosten der Alternativen als Entscheidungsvariable anzusetzen, sondern zunächst die mit den Alternativen verbundenen Ausprägungen der Kosteneinflussgrößen zu ermitteln. Dadurch können bereits bekannte Alternativen differenzierter bewertet und neue Alternativen leichter entdeckt werden. Kostenfunktionen können zusätzliche Handlungsalternativen in den Blick rücken, indem sie relevante Kosteneinflussgrößen und ihre Wirkungen auf die Kostenhöhe aufzeigen. Wenn die Analyse einer Kostenfunktion z. B. ergibt, dass die Vielfalt der verbauten Einzelteile eine bedeutende Kosteneinflussgröße ist, und bisher lediglich Neuprodukte zur Diskussion stehen, die die Teilevielfalt im Unternehmen erhöhen würden, kann dies zum Anlass genommen werden, eine zusätzliche Alternative zu konstruieren, die nur Teile enthält, die in anderen Produkten bereits verwendet werden.¹⁾

4. Prognose mit Kostenfunktionen

Um Alternativen bewerten zu können, müssen ihre zielwirksamen Konsequenzen prognostiziert werden. Eine Prognose ist eine Aussage über ein zukünftiges Ereignis, die aus allgemeinen Hypothesen und konkreten Anzecedensbedingungen abgeleitet wird.²⁾ Abb. B-10 zeigt ein einfaches Beispiel zur Prognose mit Kostenfunktionen.

¹⁾ Vgl. Troßmann [Kennzahlen] 524.

²⁾ Vgl. Hempel/Oppenheim [Studies].

Hypothese	$K = 0,62e^{x_1} - 12\sqrt{x_2} + 3,14x_3$
Antezedens	$x = (8,15; 23,08; 47,11)^T$
Prognose	$\widehat{K} = 2.237,57$

Abb. B-10: Beispiel einer Kostenprognose

Im besten Fall basiert eine Prognose auf theoretisch begründeten, allgemeingültigen – also zeit- und ortsunabhängigen – Gesetzeshypothesen.¹⁾ Universelle Aussagen über soziale Systeme sind allerdings kaum möglich.²⁾ In der Regel beschränkt man sich hier auf „Theorien mittlerer Reichweite“.³⁾ Diese enthalten z. B. raum-zeitlich begrenzte Aussagen der Art „Unter den Bedingungen z hat die unabhängige Variable x einen positiven Einfluss auf Zielgröße y .“ Solche Erkenntnisse können zur Vorauswahl der Prognosevariablen genutzt werden. Um die Auswirkungen einer Handlung im Einzelfall prognostizieren zu können, müssen die Zusammenhänge jedoch situationsbezogen präzisiert und wenn möglich quantifiziert werden, so dass konkrete Prognosen wie in Abb. B-10 möglich werden.⁴⁾ Solche situativen Hypothesen zur Vorhersage konkreter Wird-Kosten sind die charakteristischen Merkmale planungsorientierter Kostenrechnungssysteme.⁵⁾

¹⁾ Vgl. Wild [Unternehmensplanung] 92. Zum grundsätzlichen Problem der Induktion von Gesetzeshypothesen aus der Empirie vgl. Goodman [Fact].

²⁾ Vgl. Siemoneit [Wissenschaftstheorie] 122 ff., mit zahlreichen weiteren Nachweisen. Die Frage, wie breit der Anwendungsbereich einzelner Hypothesen ist und welcher Grad an Allgemeingültigkeit grundsätzlich erreicht werden kann, wird in einigen Teilgebieten der Betriebswirtschaftslehre kontrovers diskutiert (vgl. etwa die Diskussion um die Erfolgsfaktorenforschung, angestoßen von Nicolai/Kieser [Erfolglosigkeit]; einige Diskussionsbeiträge sind unter www.dialog-erfolgsfaktorenforschung.de abrufbar).

³⁾ Vgl. Merton [Theory] 39 ff.

⁴⁾ Siehe Abschnitt III.2, S. 25 ff.

⁵⁾ Vgl. Schweitzer/Küpper [Systeme] 61.

5. Alternativenbewertung mit Kostenfunktionen

In der Kostenrechnung gibt es eine Reihe von Standardrechnungen, mit denen die Kostenwirkungen der Alternativen in häufig wiederkehrenden Entscheidungssituationen bewertet werden. Dazu gehören in erster Linie *produktbezogene* Rechnungen wie

- die *Kalkulation* der Kosten für eine Produkteinheit, ein Fertigungslos, einen Auftrag etc. (Kostenträgerstückrechnung),
- die nach Produktarten, Aufträgen und anderen Bezugsobjekten gegliederte *Periodenerfolgsrechnung* (basierend auf einer Kostenträgerzeitrechnung), die oft in Form einer Deckungsbeitragsrechnung durchgeführt wird.¹⁾

Die direkte Zurechnung auf Kostenträger ist allerdings nur für einen Teil der insgesamt anfallenden Kosten (die Kostenträgereinzelkosten) möglich.²⁾ Die übrigen Kosten (die Kostenträrgemeinkosten) werden zunächst auf Kostenstellen oder Prozesse verteilt. Die *Kostenstellenrechnung* ist üblicherweise das Teilgebiet der Kostenrechnung, in dem *Kostenfunktionen* aufgestellt werden. Dadurch können die Kosten einer Kostenstelle prognostiziert werden. Zugleich werden die Einflussgrößen der Kostenfunktionen zur Weiterverteilung der Kosten auf Kostenträger genutzt. Für Kostenfunktionen, die die Menge eines Kostenträgers als Einflussgröße enthalten, ist dies unmittelbar einleuchtend. So können auf Grundlage einer Kostenfunktion $K = 2.180 + 2x_A$ mit der Ausbringungsmenge x_A einer Produkteinheit Stückkosten von 2 € zugerechnet werden. Bei Kostenfunktionen mit Einflussgrößen, die nicht zugleich Kostenträger sind (was nicht ungewöhnlich ist), werden die Kosten zunächst auf die Einflussgrößen verteilt. Anschließend

¹⁾ Vgl. Schweitzer/Küpper [Systeme] 196 f., 467 ff.

²⁾ Vgl. auch im Weiteren Schweitzer/Küpper [Systeme] 28.

können sie über eine Bezugsgrößenkalkulation¹⁾ weiter auf die Kostenträger verrechnet werden.

Nach dem *Verursachungsprinzip* der Kostenrechnung sind die Kosten denjenigen Leistungen zuzuordnen, zu denen sie in einer zwingenden Beziehung stehen (*Verursachungsprinzip* i. w. S.).²⁾ Eine Kostenfunktion ist die mathematisch exakte Formulierung einer zwingenden Beziehung zwischen der Kostenhöhe und ihren Einflussgrößen (deren Höhe wiederum mit der Leistungserstellung zusammenhängt), und damit die Grundlage der Kostenzurechnung nach dem *Verursachungsprinzip*.³⁾

Kostenfunktionen als Verrechnungsgrundlage sind auch mit dem *Identitätsprinzip* vereinbar, demzufolge zwei Größen einander gerade dann gegenübergestellt („zugerechnet“) werden können, wenn sie denselben dispositiven Ursprung haben, d. h. auf dieselbe, identische Entscheidung zurückgehen.⁴⁾ Das Identitätsprinzip präzisiert das *Verursachungsprinzip* i. w. S. und geht über die enge, kausale Interpretation des *Verursachungsprinzips* i. e. S. hinaus, da keine kausale Beziehung zwischen den betrachteten Größen verlangt wird. Mit einer Kostenfunktion können die zweifachen Konsequenzen einer Entscheidung – zum einen auf die mit der Leistungserstellung verbundenen Einflussgrößen, zum anderen auf die Kostenhöhe – einander eindeutig zugeordnet werden. Zwischen den Einflussgrößen und der Kostenhöhe muss dabei kein Kausalzusammenhang bestehen.

¹⁾ Vgl. z. B. Kilger/Pampel/Vikas [Plankostenrechnung] 520 ff. Siehe auch S. 34 f.

²⁾ Vgl. Schweitzer/Küpper [Systeme] 55 f., mit weiteren Nachweisen.

³⁾ Vgl. Rummel [Kostenrechnung] 17; Schweitzer/Küpper [Systeme] 56.

⁴⁾ Vgl. auch im Weiteren Riebel [Einzelkostenrechnung] 76, 418 ff., 627.

Die so auf die Einflussgrößen verrechneten Kosten sind die Grenzkosten der Entscheidung.¹⁾ Nach dem Grenzprinzip lassen sich auch die Alternativen *spezieller Entscheidungsprobleme*, z. B. über das Produktionsprogramm, das Beschaffungsprogramm, Bestellmengen, Losgrößen, Eigenfertigung oder Fremdbezug, die Auswahl eines Produktionsverfahrens oder die Annahme von (Zusatz-)Aufträgen,²⁾ bewerten.

6. Outputorientierte Budgetierung mit Kostenfunktionen

Kostenfunktionen dienen der Begründung von outputorientierten (leistungsorientierten) Kostenbudgets. Ein Budget ist eine „spezielle Plangröße in Geld, die einer Organisationseinheit für eine Periode zugeordnet wird.“³⁾ Outputorientierte Budgets werden in Abhängigkeit von der Leistung einer Organisationseinheit formuliert. Voraussetzung dafür ist eine Hypothese über den Zusammenhang zwischen der Leistung und dem dafür benötigten Budget.⁴⁾ Wird diese Hypothese in eine Kostenfunktion gefasst, lässt sich für jeden beliebigen Output ein entsprechendes Budget herleiten. Die systematische Begründung outputorientierter Budgets durch Kostenfunktionen wird als *flexible Plankostenrechnung* bezeichnet.⁵⁾ Da outputorientierte Budgets häufig aus dem geplanten Produktions- oder Absatzprogramm abgeleitet werden, spricht man auch von *Programmbudgetierung*.⁶⁾

¹⁾ Siehe Abschnitt III.3, S. 29 f.

²⁾ Vgl. Schweitzer/Küpper [Systeme] 30 f.

³⁾ Troßmann [Controlling] 189; vgl. auch im Weiteren Ewert/Wagenhofer [Unternehmensrechnung] 400 ff., mit weiteren Nachweisen.

⁴⁾ Vgl. auch im Weiteren Troßmann [Controlling] 198 f.

⁵⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 65; Eisele/Knobloch [Technik] 928.

⁶⁾ Vgl. Küpper u. a. [Controlling] 453.

Je nach Budgetierungszweck bieten sich unterschiedliche Kostenfunktionen als Grundlage an. Falls die Budgetierung nur dazu dient, den Geldbedarf einer Organisationseinheit zu prognostizieren, ist eine technisch oder statistisch fundierte *Prognosekostenfunktion* am besten geeignet, die zukünftigen Ist-Kosten möglichst treffsicher vorherzusagen. Ein auf diese Weise ermitteltes Kostenbudget kann beispielsweise Teil eines umfangreicheren *master budgets* angelsächsischer Prägung sein.¹⁾

Wenn hingegen eine verhaltensorientierte Steuerung beabsichtigt ist, wird der Organisationseinheit eine *Sollkostenfunktion* vorgegeben, die angibt, wie hoch die Kosten für unterschiedliche Outputmengen sein sollen.²⁾ Mit der Vorgabe an eine Verantwortungseinheit wird die Budgetierung zu einem Instrument der Plandurchsetzung.³⁾ Die Steuerungswirkung der Budgetvorgabe kann durch anschließende Budgetkontrollen und -abweichungsanalysen⁴⁾ sowie die Integration der Budgetierung in Leistungsbeurteilungs- und Anreizsysteme verstärkt werden.⁵⁾ In jedem Fall sind die Könnens- und Wollensdefizite der budgetierten Einheiten sowie Informationsasymmetrien zwischen den Hierarchieebenen zu beachten.⁶⁾ Daraus resultieren einige typische Anforderungen an das Budget. Vor allem soll es anspruchsvoll, aber realistisch sein sowie von der budgetierten Einheit als

¹⁾ Vgl. Ewert/Wagenhofer [Unternehmensrechnung] 400 ff.; Weber/Schäffer [Controlling] 293 ff.; jeweils mit weiteren Nachweisen.

²⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 65; Eisele/Knobloch [Technik] 931.

³⁾ Vgl. Wild [Unternehmensplanung] 40.

⁴⁾ Siehe dazu Abschnitt 7, S. 68 ff.

⁵⁾ Vgl. Troßmann [Controlling] 190.

⁶⁾ Siehe Abschnitt II.2, S. 14 ff. Vgl. auch Wall/Kießling [Budgetinformationen] 74; Schäffer/Kramer [Budgetverhandlungen].

erreichbar und erstrebenswert angesehen werden.¹⁾ Prognosegrößen genügen diesen Anforderungen nur bedingt. Unter anderem beinhalten sie erwartete Unwirtschaftlichkeiten und setzen somit keinen Anreiz, diese zu beseitigen.²⁾ Deshalb ist es sinnvoll, für die Budgetvorgabe eine von der Prognosekostenfunktion verschiedene Sollkostenfunktion zu verwenden.³⁾

Oft bietet es sich an, zunächst die Prognosekostenfunktion zu bestimmen und daraus unter Berücksichtigung verhaltenswissenschaftlicher und ökonomischer Aspekte eine Sollkostenfunktion abzuleiten.⁴⁾ Abb. B–11 zeigt ein solches Vorgehen. Die nichtlineare Prognosekostenfunktion wird durch eine leichter zu handhabende, lineare Sollkostenfunktion ersetzt. Im relevanten Outputbereich verläuft die Sollkostenfunktion mit der selben Steigung, aber stets unterhalb der Prognosekostenfunktion, um ein höheres Anspruchsniveau zu erzeugen.

Die Sollkostenfunktion aus den Bedingungen eines optimalen Leistungsprozesses, d. h. dem technisch minimal möglichen Faktorverbrauch bei gegebener Leistung, abzuleiten, ist hingegen unter verhaltensorientierten Gesichtspunkten i. d. R. nicht sinnvoll.⁵⁾ Eine Optimalkostenfunktion als Budgetvorgabe kann demotivierend wirken, da das Budget bestenfalls gerade eingehalten, aber nie unterschritten wird. Die Istkosten sind immer mindestens so hoch wie die Optimalkosten. Die Sollkostenfunktion in Abb. B–11 ist demnach keine Optimalkostenfunktion.

¹⁾ Vgl. ausführlich Schweitzer/Küpper [Systeme] 620 ff., mit weiteren Nachweisen; Wall/Kießling [Budgetinformationen] 76 f.; Troßmann [Controlling] 197. Siehe auch S. 14.

²⁾ Vgl. auch im Weiteren Ewert/Wagenhofer [Unternehmensrechnung] 310 f.

³⁾ Siehe S. 16. Vgl. auch Rösli/Bunce [Doppelnatur].

⁴⁾ Vgl. Ewert/Wagenhofer [Unternehmensrechnung] 310 f.; Troßmann [Controlling] 199.

⁵⁾ Vgl. auch im Weiteren Ewert/Wagenhofer [Unternehmensrechnung] 310; Schweitzer/Küpper [Systeme] 685 f.

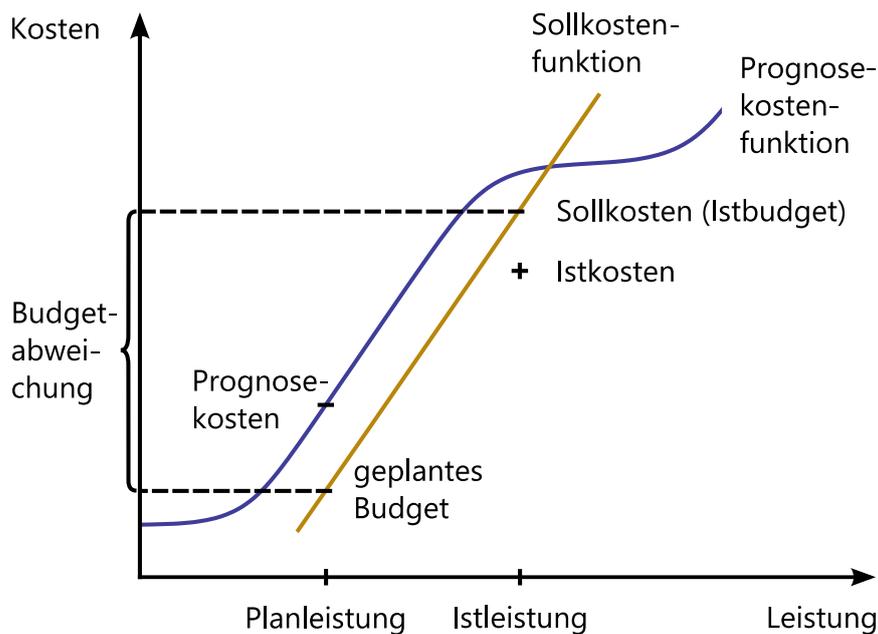


Abb. B-11: Prognose- und Sollkostenfunktionen einer flexiblen Plankostenrechnung

Die flexible Plankostenrechnung ermöglicht eine differenzierte Abweichungsanalyse. Vor allem lässt sich feststellen, welcher Teil der Plan-Ist-Abweichung durch eine Leistungsabweichung verursacht wurde. In Abb. B-11 ist der Fall dargestellt, dass die Istkosten zwar höher als die Plankosten sind, dass aber auch mehr geleistet wurde als geplant und deshalb ein höheres Budget angemessen ist. Die Differenz zwischen dem Budget bei Planleistung und dem bei Istleistung wird als Budgetabweichung bezeichnet.¹⁾ Sie entspricht dem Betrag, um den das (leistungsorientierte) Budget einer Kostenstelle angepasst werden muss, um der geänderten Leistung gerecht zu werden. Unter Berücksichtigung der Budgetabweichung wurde das Budget eingehalten (die Istkosten sind geringer als die Sollkosten). Wenn neben der Leistung noch andere Einflussgrößen in der Kostenfunktion enthalten sind, kann die verbleibende Soll-Ist-Abweichung in weitere Teilabweichungen aufgegliedert werden (siehe Abschnitt 7).

¹⁾ Vgl. Schweitzer/Küpper [Systeme] 703; Troßmann [Controlling] 216.

Die flexible Plankostenrechnung ist nur einsetzbar, wenn die Leistung der budgetierten Kostenstelle quantitativ messbar ist. Das ist durchaus nicht nur für Produktionskostenstellen der Fall. Auch in sogenannten „Gemeinkostenbereichen“ (Sekundärleistungsbereichen) lassen sich geeignete Outputgrößen definieren (siehe Abb. B–6, S. 48). Insbesondere die Prozesskostenrechnung ist darauf ausgelegt, eine flexible Plankostenrechnung für Dienstleistungen zu ermöglichen.¹⁾ In manchen Bereichen ist eine genaue Leistungsquantifizierung aber sehr mühsam oder nicht sinnvoll. Hier kommen andere Budgetierungstechniken zum Einsatz, die nicht auf Kostenfunktionen basieren.

7. Kontrolle und Abweichungsanalyse mit Kostenfunktionen

a) Unterscheidung von Durchführungs- und Dispositionskontrolle

An die Plan-Durchsetzung und -Realisation schließt sich die Kontrolle an. Sie umfasst den Vergleich zwischen einer Plangröße und einer Umsetzungsgröße sowie die Analyse der sich ergebenden Abweichungen. Nach der organisatorischen Ebene, auf die die Kontrolle gerichtet ist, unterscheidet man zwei Arten der Kontrolle (siehe Abb. B–9, S. 56):²⁾

- Auf der Ausführungsebene wird kontrolliert, ob die Planrealisierung den Vorgaben entspricht (*Durchführungskontrolle*). Die Plangrößen, auf die Bezug genommen wird, sind Vorgabegrößen mit normativem Charakter. Zweck der Durchführungskontrolle ist die Sicherstellung einer

¹⁾ Vgl. Troßmann [Gemeinkosten-Budgetierung] 521 ff.

²⁾ Vgl. Riebel [Einzelkostenrechnung] 11 f.; Troßmann [Controlling] 109 ff. Für eine ausführliche Übersicht mit weiteren Nachweisen vgl. Schäffer [Kontrolle] 45 ff.

zielentsprechenden Durchsetzung der Planvorgaben. Sie ist Teil der verhaltensorientierten Steuerung.¹⁾

- Auf der Führungsebene wird kontrolliert, wie stark die realisierten Werte von den Prognosewerten abweichen (*Dispositionskontrolle*). Die Plangrößen, die der Dispositionskontrolle zugrundeliegen, sind rein deskriptive Prognosegrößen. Zweck der Dispositionskontrolle ist es, Probleme zu erkennen, damit aktuelle Pläne angepasst und zukünftige Planungsprozesse verbessert werden können. Dies wird auch als Entscheidungsfunktion oder Lernfunktion der Kontrolle bezeichnet.²⁾

Schäffer identifiziert zusätzlich eine dritte Kontrollart, deren Zweck in der „Sicherstellung eines rationalen Modells“ besteht.³⁾ Eine ähnliche Systematik mit drei Kontrollebenen entwirft Luhmann in seiner sozialwissenschaftlichen Systemtheorie. Hier wird auf der höchsten Ebene kontrolliert, ob die Zwecke des Systems noch geeignet sind, den Fortbestand des Systems zu sichern.⁴⁾ Eine solche dritte Art der Kontrolle lässt sich allerdings nur schwer mit der in der Betriebswirtschaftslehre verbreiteten Begriffsverwendung in Einklang bringen, derzufolge Kontrollen *in den Prozessablauf integrierte* Überwachungstätigkeiten sind.⁵⁾

b) Unterscheidung von Abweichungsursachen

Weder für Durchführungs- noch für Dispositionskontrollen reicht es aus, allein die Höhe der Abweichungen zu kennen. Nur wenn auch die Abwei-

¹⁾ Siehe Abschnitt II.2, S. 14. Vgl. auch Ewert/Wagenhofer [Unternehmensrechnung] 302 ff.

²⁾ Vgl. Ewert/Wagenhofer [Unternehmensrechnung] 302.

³⁾ Vgl. Schäffer [Kontrolle] 45 ff.

⁴⁾ Vgl. Luhmann [Zweckbegriff] 224 ff.

⁵⁾ Vgl. z. B. Freiling/Lück [Überwachung] 997 f.; Baetge [Überwachung] 179; Ewert/Wagenhofer [Unternehmensrechnung] 300.

chungsursachen bekannt sind, kann zukünftigen Abweichungen entgegen-
gewirkt werden. Folgende Abweichungsursachen werden unterschieden:¹⁾

(1) nicht kontrollierbare Ursachen:

- (a) *Zufall*: die Abweichungsursachen lassen sich nicht bestimmen oder nicht messen,
- (b) „*höhere Gewalt*“: die Abweichungsursachen sind bekannt, lassen sich aber nicht beeinflussen;²⁾

(2) Modellierungsfehler:

- (a) fehlerhaftes *Planungsmodell*, weil
 - (α) wichtige Variablen nicht enthalten sind,
 - (β) die modellierten Zusammenhänge nicht den tatsächlichen Zusammenhängen entsprechen (wenn z. B. nichtlineare Zusammenhänge durch lineare Funktionen dargestellt werden),
 - (γ) Parameter falsch geschätzt wurden (z. B. weil die zugrundeliegenden Vergangenheitsdaten fehlerhaft waren oder sich seit der letzten Parametrisierung die Bedingungen geändert haben),³⁾
- (b) fehlerhaftes *Messmodell*: Aufzeichnungsfehler, Fehlbuchungen, Rechenfehler etc.;

(3) Abweichungen der unabhängigen Variablen:

- (a) *Prognosefehler*: Abweichungen zwischen Prognose- und Istwerten der Einflussgrößen,
- (b) *Ausführungsfehler*: Abweichungen zwischen Vorgabe- und Istwerten der Einflussgrößen.

¹⁾ Vgl. ähnlich Demski [Cost Deviation] B486 ff.; Streitferdt [Abweichungsauswertung] 160 ff.; Ewert/Wagenhofer [Unternehmensrechnung] 300 f.

²⁾ Vgl. Ozan/Dyckman [Cost Variances] 90.

³⁾ Siehe Abschnitt C.III.3, S. 134 ff.

Zu (1): Da das Erforschen der genauen Abweichungsursachen Kosten verursacht, lohnt sich eine differenzierte Abweichungsanalyse nur, wenn dadurch zukünftige Abweichungen verhindert werden können. Das ist bei nicht kontrollierbaren Abweichungsursachen nicht möglich. Deshalb werden sie in der Regel auch nicht weiter ausgewertet. In der Literatur werden zahlreiche Entscheidungsmodelle beschrieben, die versuchen, zufällige von nicht zufälligen Abweichungen zu unterscheiden.¹⁾ Sie geben an, bei welchen Abweichungen sich eine genauere Analyse lohnen könnte und bei welchen nicht. Zum Teil werden dabei auch die Kosten der Auswertung explizit berücksichtigt.

Zu (2): Modellierungsfehler sind grundsätzlich vermeidbar. Allerdings steigt mit der Güte des Planungs- und Messmodells häufig der Modellierungsaufwand. Um diesen Aufwand gering zu halten, werden in der Praxis teilweise hohe modellbedingte Abweichungen in Kauf genommen. Kapitel E zeigt aber, dass mit künstlichen neuronalen Netzen reale Sachverhalte besser abgebildet werden können als etwa mit linearen Funktionen, ohne dass der Modellierungsaufwand für den betrieblichen Anwender höher wäre.

Zu (3): Abweichungen der unabhängigen Variablen sind die Ursache für Funktionswertabweichungen $\Delta f = f(\mathbf{x}^{(\text{Ist})}) - f(\mathbf{x}^{(\text{Plan})})$.²⁾ Eine Abweichungsanalyse im engeren Sinn hat das Ziel, die Funktionswertabweichung Δf durch Abweichungen der unabhängigen Variablen zu erklären. Die folgen-

¹⁾ Einen umfassenden Überblick gibt Streitferdt [Abweichungsauswertung] 68 ff.; zur Darstellung grundlegender Modelle vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 156 ff.; Ewert/Wagenhofer [Unternehmensrechnung] 347 ff.

²⁾ Die hochgestellten Indizes kennzeichnen Ist- und Planwerte. Planwerte können sowohl Prognose- als auch Vorgabewerte sein.

den Abschnitte geben einen Überblick über die wichtigsten Abweichungsanalyseverfahren dieser Art in der Kostenrechnung.

c) Differenzierte Abweichungsanalyse als allgemeines Verfahren

Unter einer kostenrechnerischen Abweichungsanalyse versteht man die Aufspaltung der gesamten Abweichung zwischen dem Kostenfunktionswert bei Ist-Einflussgrößen $K^{(\text{Ist})} = f(x^{(\text{Ist})})$ und dem Funktionswert bei Plan-Einflussgrößen $K^{(\text{Plan})} = f(x^{(\text{Plan})})$ in einflussgrößenabhängige Teilabweichungen. Die Aufspaltung ist unproblematisch, wenn die Einflussgrößen additiv verknüpft und voneinander unabhängig sind.¹⁾ Dann kann die Gesamtkostenfunktion $K = f(x_1, \dots, x_I)$ in Teilfunktionen $K_i = f_i(x_i)$ für $i = 1, \dots, I$ aufgegliedert werden, so dass $K = \sum_{i=1}^I K_i$. Die Gesamtabweichung ΔK ergibt sich entsprechend als Summe der Einzelabweichungen ΔK_i die jeweils eindeutig auf eine Einflussgrößenabweichung $\Delta x_i = x_i^{(\text{Ist})} - x_i^{(\text{Plan})}$ zurückzuführen sind:²⁾

$$\Delta K = K^{(\text{Ist})} - K^{(\text{Plan})} = \sum_{i=1}^I \Delta K_i = \sum_{i=1}^I f_i(x_i^{(\text{Ist})}) - f_i(x_i^{(\text{Plan})}). \quad (\text{B.17})$$

Bei nicht-additiv verbundenen Einflussgrößen entstehen allerdings Kostenabweichungen höherer Ordnung, die sich nicht eindeutig einer Einflussgrößenabweichung zuordnen lassen.³⁾ Ein typisches Beispiel dafür sind die

¹⁾ Vgl. auch im Weiteren Ewert/Wagenhofer [Unternehmensrechnung] 316; Schweitzer/Küpper [Systeme] 708 f.

²⁾ Ohne Einschränkung der Allgemeingültigkeit beziehen sich die folgenden Ausführungen ausschließlich auf Ist-Plan-Vergleiche. Plan-Ist-Vergleiche sind für alle dargestellten Verfahren analog möglich.

³⁾ Vgl. auch im Weiteren Kilger/Pampel/Vikas [Plankostenrechnung] 150 f.; Kloock/Bomes [Kostenabweichungsanalyse] 229; Ewert/Wagenhofer [Unternehmensrechnung] 316.

Abweichungen, die durch den multiplikativen Zusammenhang von Faktorpreis q und Faktormenge r (siehe (B.2), S. 22) entstehen. Für eine Kostenfunktion $K = q \cdot r$ gilt

$$\begin{aligned}
 \Delta K &= K^{(\text{Ist})} - K^{(\text{Plan})} = q^{(\text{Ist})}r^{(\text{Ist})} - q^{(\text{Plan})}r^{(\text{Plan})} \\
 &= (q^{(\text{Plan})} + \Delta q)(r^{(\text{Plan})} + \Delta r) - q^{(\text{Plan})}r^{(\text{Plan})} \\
 &= \underbrace{q^{(\text{Plan})}\Delta r + \Delta q r^{(\text{Plan})}}_{\substack{\text{Abweichungen} \\ \text{1. Ordnung}}} + \underbrace{\Delta q \Delta r}_{\substack{\text{Abweichung} \\ \text{2. Ordnung}}}
 \end{aligned} \tag{B.18}$$

mit $\Delta q = q^{(\text{Ist})} - q^{(\text{Plan})}$,
 $\Delta r = r^{(\text{Ist})} - r^{(\text{Plan})}$.

Abweichungen 1. Ordnung hängen jeweils nur von einer Einflussgrößenabweichung (Δq , Δr oder allgemein Δx_i) ab und lassen sich deshalb eindeutig zuordnen. Abweichungen 2. (3., 4., ...) Ordnung ergeben sich hingegen aus zwei (drei, vier, ...) interdependenten Einflussgrößenabweichungen. Eine verursachungsgerechte Zuordnung der Abweichungen höherer Ordnung zu einzelnen Einflussgrößen ist nicht möglich.¹⁾ Die einzige Möglichkeit, alle Teilabweichungen vollständig und ohne Willkür zu berücksichtigen, besteht darin, die Abweichungen höherer Ordnung wie in (B.18) gesondert auszuweisen.²⁾ Diese Art der Abweichungsanalyse wird als *differenzierte Abweichungsanalyse* bezeichnet. Sie gilt unter Berücksichtigung verschiedener Anforderungen als beste Methode zur Abweichungsanalyse.³⁾ Insbesondere wird mit ihr die Gesamtabweichung *vollständig* in Teilabwei-

¹⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 151.; Ewert/Wagenhofer [Unternehmensrechnung] 317 ff.; Fandel u. a. [Kostenrechnung] 331.

²⁾ Vgl. auch im Weiteren Kloock/Bommes [Kostenabweichungsanalyse] 229 ff.; Ewert/Wagenhofer [Unternehmensrechnung] 325 ff.

³⁾ Vgl. auch im Weiteren Kloock/Bommes [Kostenabweichungsanalyse] 233; Kloock [Erfolgskontrolle]; Wilms [Abweichungsanalysemethoden] 81 ff.; Ewert/Wagenhofer [Unternehmensrechnung] 329; Schildbach/Homburg [Leistungsrechnung] 279, mit weiteren Nachweisen; Lengsfeld [Kostenkontrolle] 42 f.

chungen aufgegliedert, die Höhe der Teilabweichungen ist *invariant* gegenüber der Reihenfolge, in der sie abgespalten werden, und die Abweichungen höherer Ordnung werden *nicht willkürlich* einzelnen Einflussgrößen zugerechnet. Zudem enthält sie die gebräuchlicheren alternativen und kumulativen Abweichungsanalysemethoden als Sonderformen (s. u.).¹⁾

Strittig ist allerdings, mit welcher Bezugsbasis die Einflussgrößenabweichungen (z. B. Δq , Δr) gewichtet werden sollen. Häufig wird argumentiert, dass Plangrößen ($q^{(\text{Plan})}$, $r^{(\text{Plan})}$) wie in (B.18) die geeignete Basis zur verhaltensorientierten Steuerung sind, während sich Istgrößen ($q^{(\text{Ist})}$, $r^{(\text{Ist})}$) besser zur Entscheidungsfundierung eignen: Weil die Ausführenden ihre Handlungen an Plangrößen auszurichten haben, soll auch das Ergebnis ihrer Handlungen auf Basis der Plangrößen bewertet werden; demhingegen zeigen Abweichungsanalysen auf Istbezugsbasis auf, welche Kostenänderungen ausgehend von der Istsituation möglich sind, wenn durch geeignete Maßnahmen der Plan- statt des Istwertes einer Einflussgröße realisiert wird, – das so genannte „Kostenänderungspotential“.²⁾

Eine differenzierte Abweichungsanalyse kann an einer beliebigen Kostenfunktion f durchgeführt werden. Die Teilabweichungen können wie in (B.18) durch eine individuelle Analyse und Umformung der vorliegenden Funktion, durch Taylor-Reihen³⁾ oder iterativ berechnet werden. Letzteres

¹⁾ Vgl. Richter/Neubert [Taylor-Reihe] 73.

²⁾ Vgl. Kloock [Kostenkontrollmanagement] 635 f.; Ewert/Wagenhofer [Unternehmensrechnung] 313; Schweitzer/Küpper [Systeme] 713; Lengsfeld/Schiller [Kostencontrolling] 90 ff. Zur Vorteilhaftigkeit von Istgrößen als Bezugsbasis für die Entscheidungsunterstützung vgl. Kloock/Dörner [Kostenkontrolle] 132 ff; Lengsfeld [Kostenkontrolle] 21. Zur Vorteilhaftigkeit von Plangrößen als Bezugsbasis für die verhaltensorientierte Steuerung vgl. Wimmer [Kostenabweichungsanalyse] 995; Ossadnik/Barklage/Wilmsmann [Kostenkontrolle] 172.

³⁾ Vgl. Richter/Neubert [Taylor-Reihe]; Ueckerdt/Lenz [Abweichungsanalyse] 278; Lengsfeld [Kostenkontrolle] 32 ff.

ist in Abb. B–12 dargestellt. Ausgehend von einem Ist-Plan-Vergleich werden zunächst die Teilabweichungen 1. Ordnung bestimmt. Sie ergeben sich als Differenz der Funktionswerte, wenn jeweils nur eine Einflussgröße vom Planwert abweicht. Für die Teilabweichungen 2. Ordnung werden entsprechend die Differenzen der Funktionswerte berechnet, wenn jeweils zwei Einflussgrößen vom Planwert abweichen. Davon werden die Teilabweichungen 1. Ordnung der beteiligten Einflussgrößen abgezogen. So erhält man als differenzierte Abweichung 2. Ordnung $\Delta K_{ij}^{\text{diff}}$ die Teilabweichung, die von den Einflussgrößen x_i und x_j gemeinsam verursacht wurde. Auf den folgenden Ordnungsstufen werden gleichermaßen Funktionswertdifferenzen berechnet und Abweichungen tieferer Ordnung subtrahiert.

Differenzierte Abweichungsanalyse

Gesamtabweichung

$$\Delta K = f(\mathbf{x}^{(\text{Ist})}) - f(\mathbf{x}^{(\text{Plan})})$$

mit $\mathbf{x}^{(\text{Ist})} = (x_1^{(\text{Ist})}, \dots, x_I^{(\text{Ist})})$ Istgrößen
 $\mathbf{x}^{(\text{Plan})} = (x_1^{(\text{Plan})}, \dots, x_I^{(\text{Plan})})$ Plangrößen

Teilabweichungen 1. Ordnung

$$\Delta K_1^{\text{diff}} = f(x_1^{(\text{Ist})}, x_2^{(\text{Plan})}, x_3^{(\text{Plan})}, \dots, x_I^{(\text{Plan})}) - f(\mathbf{x}^{(\text{Plan})})$$

$$\Delta K_2^{\text{diff}} = f(x_1^{(\text{Plan})}, x_2^{(\text{Ist})}, x_3^{(\text{Plan})}, \dots, x_I^{(\text{Plan})}) - f(\mathbf{x}^{(\text{Plan})})$$

⋮

$$\Delta K_I^{\text{diff}} = f(x_1^{(\text{Plan})}, x_2^{(\text{Plan})}, x_3^{(\text{Plan})}, \dots, x_{I-1}^{(\text{Plan})}, x_I^{(\text{Ist})}) - f(\mathbf{x}^{(\text{Plan})})$$

Teilabweichungen 2. Ordnung

$$\Delta K_{1,2}^{\text{diff}} = [f(x_1^{(\text{Ist})}, x_2^{(\text{Ist})}, x_3^{(\text{Plan})}, x_4^{(\text{Plan})}, \dots, x_I^{(\text{Plan})}) - f(\mathbf{x}^{(\text{Plan})})] - [\Delta K_1^{\text{diff}} + \Delta K_2^{\text{diff}}]$$

$$\begin{aligned} \Delta K_{1,3}^{\text{diff}} &= [f(x_1^{(\text{Ist})}, x_2^{(\text{Plan})}, x_3^{(\text{Ist})}, x_4^{(\text{Plan})}, \dots, x_I^{(\text{Plan})}) - f(x^{(\text{Plan})})] - [\Delta K_1^{\text{diff}} + \Delta K_3^{\text{diff}}] \\ &\vdots \\ \Delta K_{I-1,I}^{\text{diff}} &= [f(x_1^{(\text{Plan})}, \dots, x_{I-2}^{(\text{Plan})}, x_{I-1}^{(\text{Ist})}, x_I^{(\text{Ist})}) - f(x^{(\text{Plan})})] - [\Delta K_{I-1}^{\text{diff}} + \Delta K_I^{\text{diff}}] \end{aligned}$$

Teilabweichungen 3. Ordnung

$$\begin{aligned} \Delta K_{1,2,3}^{\text{diff}} &= [f(x_1^{(\text{Ist})}, x_2^{(\text{Ist})}, x_3^{(\text{Ist})}, x_4^{(\text{Plan})}, x_5^{(\text{Plan})}, \dots, x_I^{(\text{Plan})}) - f(x^{(\text{Plan})})] \\ &\quad - [\Delta K_1^{\text{diff}} + \Delta K_2^{\text{diff}} + \Delta K_3^{\text{diff}} + \Delta K_{1,2}^{\text{diff}} + \Delta K_{1,3}^{\text{diff}} + \Delta K_{2,3}^{\text{diff}}] \end{aligned}$$

$$\begin{aligned} \Delta K_{1,2,4}^{\text{diff}} &= [f(x_1^{(\text{Ist})}, x_2^{(\text{Ist})}, x_3^{(\text{Plan})}, x_4^{(\text{Ist})}, x_5^{(\text{Plan})}, \dots, x_I^{(\text{Plan})}) - f(x^{(\text{Plan})})] \\ &\quad - [\Delta K_1^{\text{diff}} + \Delta K_2^{\text{diff}} + \Delta K_4^{\text{diff}} + \Delta K_{1,2}^{\text{diff}} + \Delta K_{1,4}^{\text{diff}} + \Delta K_{2,4}^{\text{diff}}] \end{aligned}$$

$$\vdots$$

$$\begin{aligned} \Delta K_{I-2,I-1,I}^{\text{diff}} &= [f(x_1^{(\text{Plan})}, \dots, x_{I-3}^{(\text{Plan})}, x_{I-2}^{(\text{Ist})}, x_{I-1}^{(\text{Ist})}, x_I^{(\text{Ist})}) - f(x^{(\text{Plan})})] \\ &\quad - [\Delta K_{I-2}^{\text{diff}} + \Delta K_{I-1}^{\text{diff}} + \Delta K_I^{\text{diff}} + \Delta K_{I-2,I-1}^{\text{diff}} + \Delta K_{I-2,I}^{\text{diff}} + \Delta K_{I-1,I}^{\text{diff}}] \end{aligned}$$

$$\vdots$$

Teilabweichung I. Ordnung

$$\begin{aligned} \Delta K_{1,\dots,I}^{\text{diff}} &= [f(x^{(\text{Ist})}) - f(x^{(\text{Plan})})] \\ &\quad - [\Delta K_1^{\text{diff}} + \dots + \Delta K_I^{\text{diff}} + \Delta K_{1,2}^{\text{diff}} + \dots + \Delta K_{I-1,I}^{\text{diff}} + \Delta K_{1,2,3}^{\text{diff}} + \dots] \end{aligned}$$

Abb. B-12: Iterative Berechnung der Teilabweichungen einer differenzierten Abweichungsanalyse¹⁾

Wegen der großen Anzahl zu überblickender Teilabweichungen fällt es häufig schwer, die differenziert ausgewiesenen Teilabweichungen zu interpretieren. Die Anzahl der Abweichungen k . Ordnung bei I Einflussgrößen

¹⁾ Vgl. Pohlen [Kostenkontrolle] 26.

beträgt $\binom{I}{k} = \frac{I!}{k!(I-k)!}$.¹⁾ Schon mit fünf unabhängigen Variablen erhält man zehn Abweichungen 2. Ordnung, zehn Abweichungen 3. Ordnung, fünf Abweichungen 4. Ordnung und eine Abweichung 5. Ordnung.

Hinzu kommt, dass dieselbe Einflussgröße an Abweichungen mit unterschiedlichen Vorzeichen beteiligt sein kann. Beispielsweise könnte $\Delta K_1^{\text{diff}} > 0$ und zugleich $\Delta K_{1,2}^{\text{diff}} < 0$ sein. Die Abweichung der einen Ordnung „kompensiert“ in diesem Fall die Abweichung der anderen Ordnung.²⁾ An Formel (B.18) wird deutlich, dass dieser Effekt u. a. dann auftritt, wenn bei multiplikativ verknüpften Einflussgrößen mindestens eine Einflussgröße ein negatives „Delta“ aufweist. Wilms schlägt vor, Kompensationseffekte dadurch zu vermeiden, Einflussgrößenabweichungen mit den Minima aus Plan- und Istwerten der verbleibenden Einflussgrößen zu gewichten (z. B. $\Delta r \cdot \min\{q^{(\text{Plan})}; q^{(\text{Ist})}\}$ und $\Delta q \cdot \min\{r^{(\text{Plan})}; r^{(\text{Ist})}\}$).³⁾ Dadurch entstehen allerdings neue Interpretationsschwierigkeiten;⁴⁾ insbesondere werden Kostenänderungspotentiale nicht einheitlich ausgewiesen.⁵⁾

¹⁾ Vgl. Wimmer [Kostenabweichungsanalyse] 986 f.

²⁾ Vgl. auch im Weiteren Wilms [Abweichungsanalysemethoden] 86 ff.; Ossadnik/Maus [Kostenabweichungsanalyse] 449 f.; Glaser [Relativität] 27 ff.

³⁾ Vgl. Wilms [Abweichungsanalysemethoden] 106 ff.; Ossadnik/Maus [Kostenabweichungsanalyse] 449; Glaser [Relativität] 31.

⁴⁾ Vgl. Ewert/Wagenhofer [Unternehmensrechnung] 325 ff.; Ossadnik/Barklage/Wilmsmann [Kostenkontrolle].

⁵⁾ Vgl. Kloock [Kostenkontrollmanagement] 632 ff.; Lengsfeld [Kostenkontrolle] 43; Kloock [Kommentar] 34.

d) Alternative und kumulative Abweichungsanalysen als Sonderformen

Aufgrund der oben genannten Interpretationsschwierigkeiten der differenzierten Abweichungsanalyse, werden in der Praxis Verfahren bevorzugt, die zu jeder Einflussgröße genau eine Abweichung ausweisen. Abweichungen höherer Ordnung werden dabei mehr oder weniger gut begründet – jedenfalls nicht verursachungsgerecht – einzelnen Einflussgrößen zugeordnet oder insgesamt vernachlässigt. Am weitesten verbreitet sind alternative und kumulative Abweichungsanalysen.¹⁾

Bei einer *alternativen Abweichungsanalyse* werden je nach gewählter Bezugsbasis einer Einflussgröße x_i entweder nur ihre Abweichung erster Ordnung oder alle Teilabweichungen, an denen sie beteiligt ist, zugeordnet (siehe Abb. B-13).²⁾

Alternative Abweichungsanalyse

auf Planbezugsbasis:

$$\begin{aligned}\Delta K_i^{\text{alt(Plan)}} &= f(x_1^{(\text{Plan})}, \dots, x_{i-1}^{(\text{Plan})}, x_i^{(\text{Ist})}, x_{i+1}^{(\text{Plan})}, \dots, x_I^{(\text{Plan})}) - f(x^{(\text{Plan})}) \\ &= \Delta K_i^{\text{diff}}\end{aligned}$$

auf Istbezugsbasis:

$$\begin{aligned}\Delta K_i^{\text{alt(Ist)}} &= f(x^{(\text{Ist})}) - (x_1^{(\text{Ist})}, \dots, x_{i-1}^{(\text{Ist})}, x_i^{(\text{Plan})}, x_{i+1}^{(\text{Ist})}, \dots, x_I^{(\text{Ist})}) \\ &= \Delta K_i^{\text{diff}} + \Delta K_{1,i}^{\text{diff}} + \dots + \Delta K_{h,i}^{\text{diff}} + \dots + \Delta K_{h,i,j}^{\text{diff}} + \dots\end{aligned}$$

Abb. B-13: Teilabweichungen einer alternativen Abweichungsanalyse

¹⁾ Sie werden in den meisten Lehrbüchern zur Kostenrechnung ausführlich behandelt; vgl. z. B. Kilger/Pampel/Vikas [Plankostenrechnung] 152 ff.; Schweitzer/Küpper [Systeme] 711 f.; Ewert/Wagenhofer [Unternehmensrechnung] 320 f.; Schildbach/Homburg [Leistungsrechnung] 275 ff.; Fandel u. a. [Kostenrechnung] 323 ff.

²⁾ Vgl. Pohlen [Kostenkontrolle] 17 ff.; Fandel u. a. [Kostenrechnung] 323 ff.

Die Abweichungen höherer Ordnung werden auf Planbezugsbasis überhaupt nicht, auf Istbezugsbasis hingegen mehrfach verrechnet. Deshalb stimmt bei einer alternativen Abweichungsanalyse die Summe der Teilabweichungen nur dann mit der Gesamtabweichung überein, wenn es keine Abweichungen höherer Ordnung gibt:

$$\sum_{i=1}^I \Delta K_i^{\text{alt(Plan)}} \leq \Delta K, \quad \sum_{i=1}^I \Delta K_i^{\text{alt(Ist)}} \geq \Delta K. \quad (\text{B.19})$$

Dass die Gesamtabweichung ΔK nicht vollständig in Teilabweichungen aufgespalten wird, d. h. dass nicht notwendigerweise $\sum_{i=1}^I \Delta K_i = \Delta K$, gilt als schwerwiegender Nachteil der alternativen Abweichungsanalyse.¹⁾

Bei der *kumulativen Abweichungsanalyse* werden die Istkosten schrittweise den Plankosten angenähert (siehe Abb. B–14). Die zuerst abgespaltene Abweichung ΔK_1^{kum} enthält alle Teilabweichungen, an denen die Einflussgröße x_1 beteiligt ist. Die zweite Abweichung ΔK_2^{kum} enthält alle Teilabweichungen, an denen die Einflussgröße x_2 beteiligt ist, außer die, die bereits ΔK_1^{kum} zugeordnet wurden, usw. Ein Algorithmus zur Berechnung der Teilabweichungen wird in Kapitel E.IV vorgestellt (siehe Algorithmus E–2, S. 327).

Wenn für jede Teilabweichung ΔK_i^{kum} die enthaltene Abweichung 1. Ordnung getrennt von den Abweichungen höherer Ordnung ausgewiesen wird, bezeichnet man das Verfahren auch als *differenziert-kumulative Abweichungsanalyse*. Dafür werden die differenzierten Teilabweichungen ΔK^{diff} nach Abb. B–12 berechnet und wie in Abb. B–14 auf die kumulativen Teilabweichungen verteilt.

¹⁾ Vgl. Kloock/Bommes [Kostenabweichungsanalyse] 230, mit weiteren Nachweisen; Ueckerdt/Lenz [Abweichungsanalyse] 277.

Kumulative Abweichungsanalyse

$$\begin{aligned}\Delta K_1^{\text{kum}} &= f(\mathbf{x}^{(\text{Ist})}) - (x_1^{(\text{Plan})}, x_2^{(\text{Ist})}, x_3^{(\text{Ist})}, x_4^{(\text{Ist})}) \\ &= \Delta K_1^{\text{diff}} + \Delta K_{1,2}^{\text{diff}} + \Delta K_{1,3}^{\text{diff}} + \Delta K_{1,4}^{\text{diff}} + \Delta K_{1,2,3}^{\text{diff}} + \Delta K_{1,2,4}^{\text{diff}} + \Delta K_{1,3,4}^{\text{diff}} + \Delta K_{1,2,3,4}^{\text{diff}}\end{aligned}$$

$$\begin{aligned}\Delta K_2^{\text{kum}} &= f(x_1^{(\text{Plan})}, x_2^{(\text{Ist})}, x_3^{(\text{Ist})}, x_4^{(\text{Ist})}) - f(x_1^{(\text{Plan})}, x_2^{(\text{Plan})}, x_3^{(\text{Ist})}, x_4^{(\text{Ist})}) \\ &= \Delta K_2^{\text{diff}} + \Delta K_{2,3}^{\text{diff}} + \Delta K_{2,4}^{\text{diff}} + \Delta K_{2,3,4}^{\text{diff}}\end{aligned}$$

$$\begin{aligned}\Delta K_3^{\text{kum}} &= f(x_1^{(\text{Plan})}, x_2^{(\text{Plan})}, x_3^{(\text{Ist})}, x_4^{(\text{Ist})}) - f(x_1^{(\text{Plan})}, x_2^{(\text{Plan})}, x_3^{(\text{Plan})}, x_4^{(\text{Ist})}) \\ &= \Delta K_3^{\text{diff}} + \Delta K_{3,4}^{\text{diff}}\end{aligned}$$

$$\begin{aligned}\Delta K_4^{\text{kum}} &= f(x_1^{(\text{Plan})}, x_2^{(\text{Plan})}, x_3^{(\text{Plan})}, x_4^{(\text{Ist})}) - f(\mathbf{x}^{(\text{Plan})}) \\ &= \Delta K_4^{\text{diff}}\end{aligned}$$

Abb. B-14: Teilabweichungen einer kumulativen Abweichungsanalyse am Beispiel einer Kostenfunktion mit vier unabhängigen Variablen

Bei einer kumulativen Abweichungsanalyse stimmt die Summe der Teilabweichungen immer mit der Gesamtabweichung überein:

$$\sum_{i=1}^I \Delta K_i^{\text{kum}} = \Delta K. \quad (\text{B.20})$$

Ein Nachteil der Methode ist allerdings, dass die Höhe der Teilabweichungen von der Reihenfolge der Abspaltung abhängt.

In der Kostenrechnungsliteratur wird argumentiert, das in Abb. B-14 dargestellte Vorgehen habe gegenüber dem umgekehrten Ansatz – Annäherung der Plan- an die Istkosten – den Vorteil, dass die zuletzt abgespaltene

Teilabweichung ΔK_I^{kum} keine Abweichungen höherer Ordnung enthält.¹⁾ Das gilt allerdings nur, wenn die Abweichungen höherer Ordnung wie in Abb. B–12 auf Planbasis berechnet werden.²⁾ Wenn wir in Abb. B–12 Ist- und Planwerte vertauschen, ergeben sich Teilabweichungen auf Istbasis, von denen ausgehend sich das Bild in Abb. B–14 umkehrt: die zuerst abgespaltene Abweichung $\Delta K_1^{\text{kum}} = f(x^{(\text{Ist})}) - (x_1^{(\text{Plan})}, x_2^{(\text{Ist})}, \dots, x_I^{(\text{Ist})})$ enthält dann nur eine Abweichung 1. Ordnung auf Istbasis $\Delta K_1^{\text{diff(Ist)}}$; die zuletzt abgespaltene Abweichung ΔK_I^{kum} enthält hingegen Abweichungen 1. bis I . Ordnung auf Istbasis. Ob eine Teilabweichung Abweichungen höherer Ordnung enthält, hängt also vom gewählten Bezugssystem ab; Aussagen über Teilabweichungen sind relativ zur gewählten Bezugsbasis.³⁾ Das gilt in gleicher Weise auch für die alternative und andere Formen der Abweichungsanalyse.

Für Durchführungskontrollen hat sich eine kumulative Abweichungsanalyse mit einer festen Abspaltungsreihenfolge etabliert. Ausgehend von einer Kostenfunktion $K = q \cdot a \cdot x$ (siehe (B.5), S. 24) wird zuerst die Preisabweichung $\Delta K_q^{\text{kum}} = \Delta q \cdot a^{(\text{Ist})} \cdot x^{(\text{Ist})}$, dann die Verbrauchsabweichung $\Delta K_a^{\text{kum}} = q^{(\text{Plan})} \cdot \Delta a \cdot x^{(\text{Ist})}$ und schließlich die Budgetabweichung $\Delta K_x^{\text{kum}} = q^{(\text{Plan})} \cdot a^{(\text{Plan})} \cdot \Delta x$ abgespalten (siehe Abb. B–15).⁴⁾

¹⁾ Vgl. z. B. Kilger/Pampel/Vikas [Plankostenrechnung] 155; Glaser [Teilabweichungen] 147; Fandel u. a. [Kostenrechnung] 333.

²⁾ Bisher wurde, um die Übersichtlichkeit zu bewahren, bei den differenzierten Teilabweichungen ΔK^{diff} auf den hochgestellten Index „(Plan)“ verzichtet. Vollständig hätte man sie als $\Delta K^{\text{diff(Plan)}}$ bezeichnen können.

³⁾ Vgl. Glaser [Relativität].

⁴⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 155; Schweitzer/Küpper [Systeme] 317 ff.; eine ausführliche Begründung für diese Reihenfolge geben Schildbach/Homburg [Leistungsrechnung] 279 f.

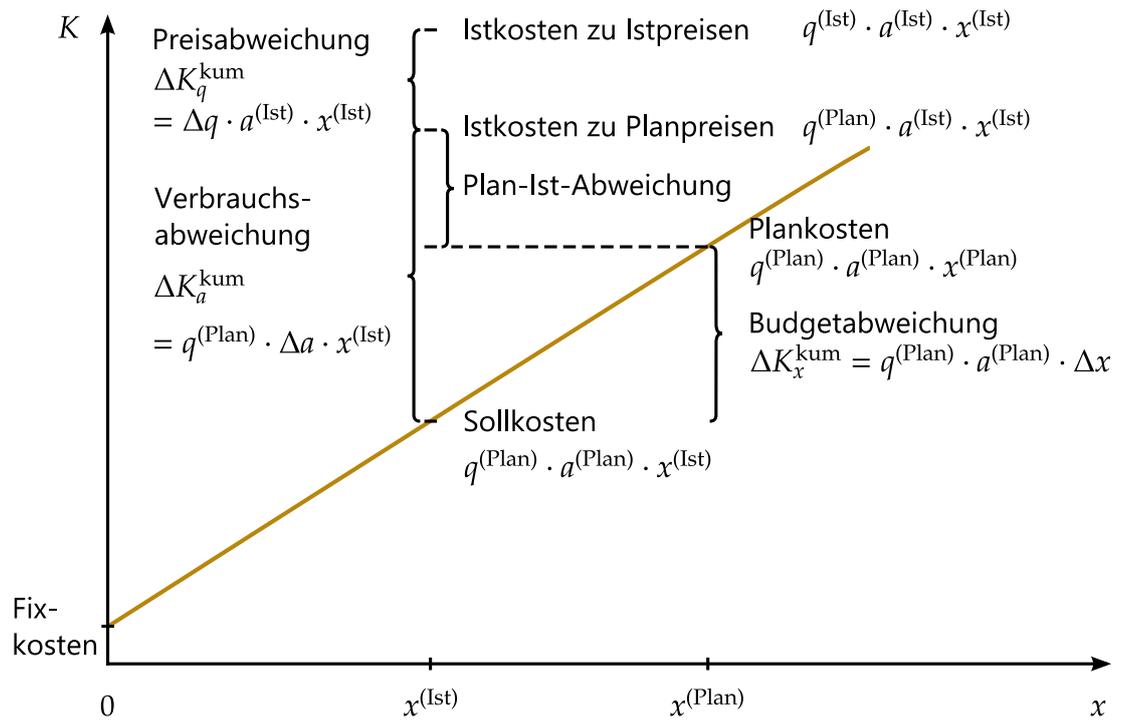


Abb. B-15: Kumulative Abweichungsanalyse in der flexiblen Plankostenrechnung

C. Künstliche neuronale Netze als Instrument zur Schätzung von Kostenfunktionen

1. *Methodische Grundlagen künstlicher neuronaler Netze*

1. Funktionsweise von Feedforward-Netzen

Künstliche neuronale Netze bilden zentrale Eigenschaften natürlicher Nervensysteme abstrakt, mathematisch ab. Sie bestehen aus einfachen Recheneinheiten („Neuronen“), die Signale verarbeiten und über gewichtete Verbindungen („Synapsen“) an andere Einheiten weitergeben. Dargestellt wird ein künstliches neuronales Netz als gerichteter, gewichteter Graph, dessen Knoten „Neuronen“ und dessen Kanten „Verbindungen“ heißen.¹⁾ Jeder Verbindung (i, j) ist ein Gewicht w_{ij} zugeordnet.

Eingehende Signale verarbeitet ein Neuron in zwei Schritten (siehe Abb. C-1). Zunächst wird das Nettoeingangssignal net_j als gewichtete Summe aller eingehenden Signale o_i berechnet:

$$net_j = \sum_i w_{ij} o_i \quad (C.1)$$

mit o_i Ausgangssignal eines vorangehenden Neurons.²⁾

¹⁾ Vgl. Kruse u. a. [Intelligence] 34.

²⁾ Die Notation folgt auch im Weiteren weitgehend Rumelhart/Hinton/McClelland [Framework].

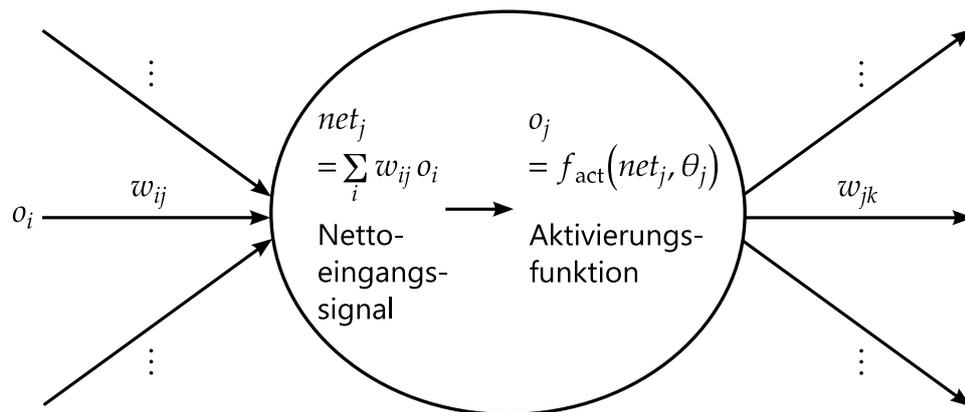


Abb. C-1: Signalverarbeitung im künstlichen Neuron

Wie eine biologische Nervenzelle soll auch ein künstliches Neuron erst aktiv werden, wenn der Eingangsreiz net einen Schwellenwert θ übersteigt.¹⁾ Der Aktivierungswert eines Neurons $f_{act}(net, \theta)$ entscheidet darüber, ob es ein Signal an nachfolgende Neuronen abgibt oder nicht. Er wird üblicherweise direkt als Outputsignal $o = f_{act}(net, \theta)$ an nachfolgende Neuronen weitergeleitet. Um einen Aktivierungszustand in Abhängigkeit vom Nettoeingangssignal net und einem Schwellenwert θ zu modellieren, reicht im Grunde eine binäre Schwellenwertfunktion

$$f_{act}(net, \theta) = \begin{cases} 1 & \text{falls } net \geq \theta \\ 0 & \text{sonst.} \end{cases} \quad (\text{C.2})$$

Allerdings sind für die meisten Anwendungen künstlicher neuronaler Netze differenzierbare Aktivierungsfunktionen notwendig. Deshalb verwendet man sigmoide (s-förmige) Funktionen wie die Tangens-Hyperbolicus-Funktion oder die logistische Funktion

$$f_{act}(net, \theta) = \frac{1}{1 + e^{-(net-\theta)}}. \quad (\text{C.3})$$

¹⁾ Indizes werden auch im Weiteren weggelassen, soweit die Darstellung dadurch übersichtlicher wird.

Abb. C-2 zeigt den Verlauf der logistischen Aktivierungsfunktion (C.3) für verschiedene Schwellenwerte θ .

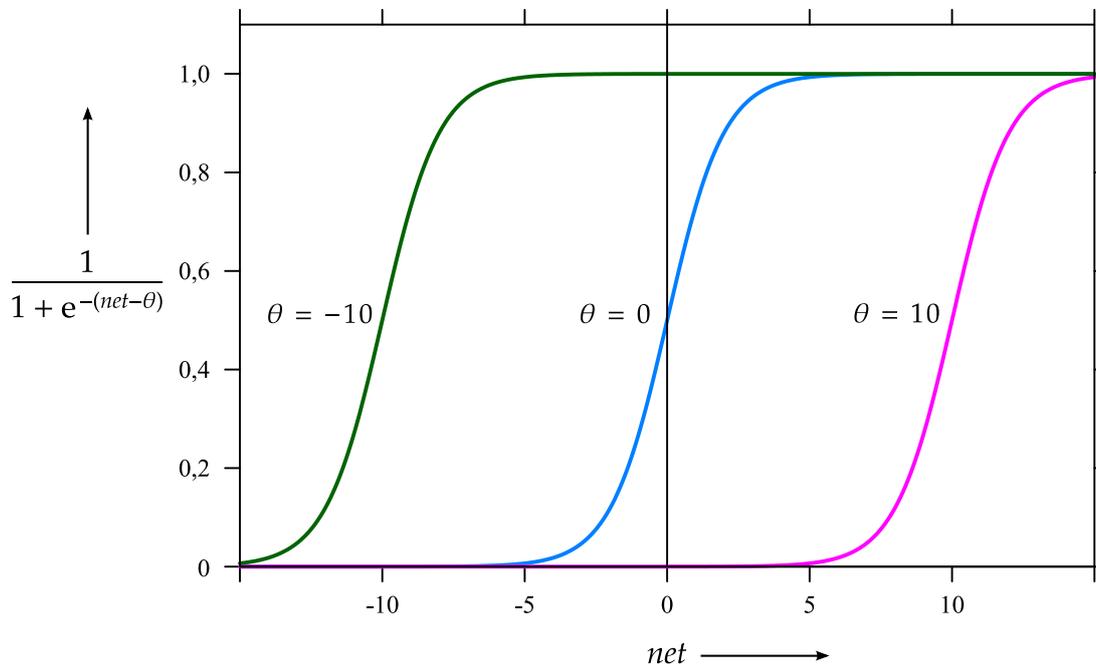


Abb. C-2: Verlauf der logistischen Aktivierungsfunktion für unterschiedliche Schwellenwerte θ

Künstliche Neuronen können auf verschiedene Weise zu einem Netz verbunden werden. Für viele Mustererkennungs-Anwendungen – u. a. zur Klassifizierung und Regression – sind vorwärtsgerichtete (*feedforward*) Netze ausreichend. Sie enthalten keine Rückkopplungen, d. h. der Graph ist zyklensfrei. Abb. C-3 zeigt den am häufigsten verwendeten Aufbau zu einem mehrschichtigen Feedforward-Netz, das als *multilayer perceptron* bezeichnet wird. Seine Neuronen sind in einer Eingabeschicht, einer Ausgabeschicht und mindestens einer „verdeckten“ Schicht (*hidden layer*) angeordnet. Die Neuronen einer Schicht sind vollständig mit den Neuronen der nachfolgenden Schicht verbunden. Innerhalb einer Schicht gibt es keine Verbindungen.

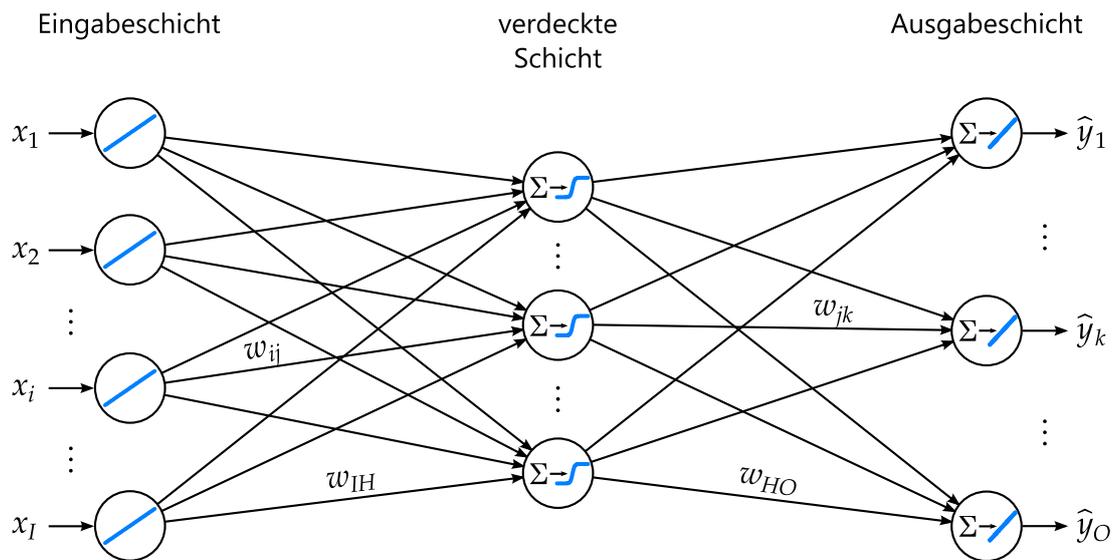


Abb. C-3: Aufbau eines multilayer perceptron

Die Neuronen der Eingabeschicht empfangen die Werte der unabhängigen Variablen x_1, \dots, x_I und geben sie in der Regel unverändert weiter, d. h. ihre Aktivierungsfunktion ist die identische Abbildung $o_i = f_{\text{act}}(x_i) = x_i$. Die Neuronen der verdeckten Schichten verarbeiten eingehende Signale wie in Abb. C-1 sowie den Formeln (C.1) und (C.3) gezeigt zu $o_j = f_{\text{act}}(\text{net}_j, \theta_j) = \frac{1}{1 + e^{-(\sum_i w_{ij} o_i - \theta_j)}}$. Die Neuronen der Ausgabeschicht summieren die gewichteten Ausgangssignale der letzten verdeckten Schicht nach (C.1) auf. Die Wahl der Aktivierungsfunktion der Ausgabeneuronen richtet sich nach der Skalierung der abhängigen Variablen y_1, \dots, y_O . Bei Klassifizierungsproblemen sind sie binär kodiert, so dass eine sigmoidale Aktivierungsfunktion wie in (C.3) am besten geeignet ist. Für Regressionsprobleme ist die Aktivierungsfunktion die identische Abbildung des Nettoeingangssignals $o_k = f_{\text{act}}(\text{net}_k) = \text{net}_k$. Der Output o_k eines Ausgabeneurons ist ein Schätzwert für die abhängige Variable y_k , d. h. $o_k = \hat{y}_k$.

Die Verarbeitungsschritte von der Eingabe- bis zur Ausgabeschicht lassen sich für ein Regressions-Netz mit einer verdeckten Schicht (siehe Abb. C-3) zu einer verketteten Funktion

$$\begin{aligned}\hat{y}_k &= f_{act}(net_k) = net_k = \sum_{j=1}^H w_{jk} o_j = \sum_{j=1}^H w_{jk} f_{act}(net_j, \theta_j) \\ &= \sum_{j=1}^H w_{jk} \frac{1}{1 + e^{-(net_j - \theta_j)}} = \sum_{j=1}^H w_{jk} \frac{1}{1 + e^{-(\sum_{i=1}^I w_{ij} x_i - \theta_j)}}\end{aligned}\quad (C.4)$$

zusammenfassen. Ein künstliches neuronales Netz ist also nichts anderes als eine nichtlineare Funktion, die die Menge der unabhängigen Variablen $\{x_i\}$ auf die Menge der Schätzwerte der abhängigen Variablen $\{\hat{y}_k\}$ abbildet.¹⁾ Als freie Parameter enthält sie Gewichte w und Schwellenwerte θ .

Damit bei der Parametrisierung (siehe Abschnitt 3) nicht verschiedene Arten von Parametern unterschieden werden müssen, können die Schwellenwerte in Gewichte umgewandelt werden. Dazu wird ein Schwellenwertneuron (*bias neuron*) eingeführt, das einen konstanten Output $o = 1$ weitergibt. In der Eingabeschicht wird das durch eine zusätzliche Variable x_0 mit dem konstantem Wert $x_0 = 1$ realisiert. Der Exponent in (C.4) lässt sich dadurch zu

$$\sum_{i=1}^I w_{ij} x_i - \theta_j \stackrel{(w_{0j} = -\theta_j)}{=} \sum_{i=1}^I w_{ij} x_i + w_{0j} \cdot 1 \stackrel{(x_0 = 1)}{=} \sum_{i=0}^I w_{ij} x_i. \quad (C.5)$$

umformen. Der Schwellenwert θ_j wird also durch ein Gewicht $w_{0j} = -\theta_j$ ersetzt.²⁾ Abb. C-4 zeigt das Schwellenwertneuron im Gesamtgraphen. Für alle nachfolgenden Schichten können in ähnlicher Weise Schwellenwertneuronen definiert werden.

¹⁾ Vgl. Bishop [Machine Learning] 228.

²⁾ Vgl. Lämmel/Cleve [Intelligenz] 216 f.; Kruse u. a. [Intelligence] 25 f.

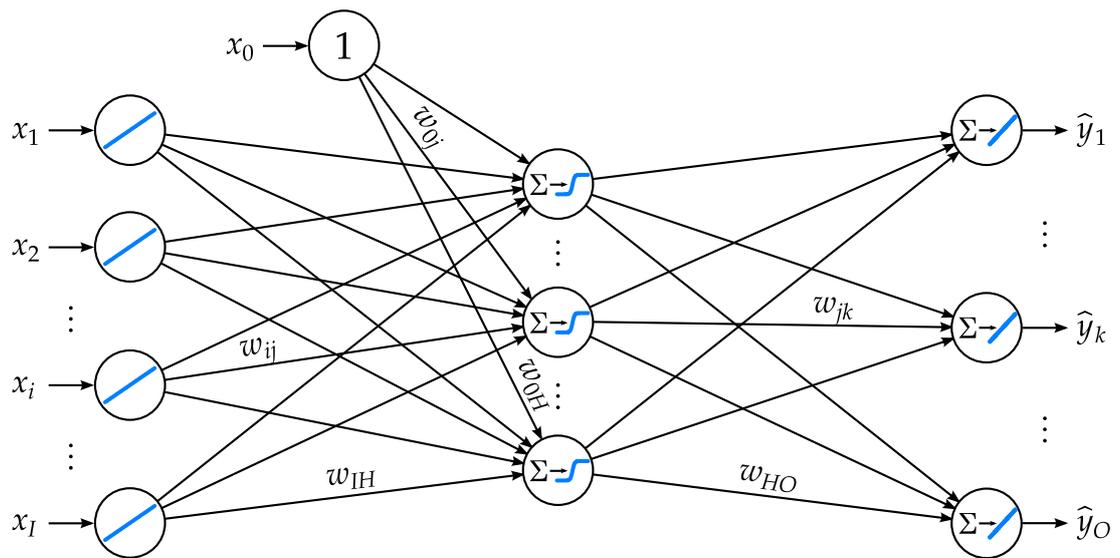


Abb. C-4: Multilayer perceptron mit Schwellenwertneuron

Die ausführliche Gesamtfunktion für das Netz aus Abb. C-4 ist

$$\begin{aligned}
 \hat{y}_k &= f_{\text{act}}(\text{net}_k) = \text{net}_k = \sum_{j=1}^H w_{jk} o_j = \sum_{j=1}^H w_{jk} f_{\text{act}}(\text{net}_j) \\
 &= \sum_{j=1}^H w_{jk} \frac{1}{1 + e^{-(\text{net}_j)}} = \sum_{j=1}^H w_{jk} \frac{1}{1 + e^{-(\sum_{i=0}^I w_{ij} x_i)}}.
 \end{aligned} \tag{C.6}$$

2. Universelle Approximation und Überanpassung

Für ein Feedforward-Netz mit mindestens einer verdeckten Schicht kann gezeigt werden, dass es jede stetige Funktion beliebig genau annähern kann, wenn es hinreichend viele verdeckte Neuronen enthält und die Aktivierungsfunktion der verdeckten Neuronen kein Polynom ist.¹⁾ Netze nach der Art eines *multilayer perceptron* wie in Abb. C-3 und Abb. C-4 sind dem-

¹⁾ Vgl. Hornik [Approximation], mit weiteren Nachweisen.

nach *universelle Approximatoren*.¹⁾ Für Zwecke der Funktionsapproximation – dazu gehören Klassifikation und Regression – ist das Grundmodell (C.6) also umfassend geeignet.

Das Approximations-Theorem zeigt aber nur die grundsätzlichen Möglichkeiten dieser Netzstruktur auf. Wie viele verdeckte Neuronen für eine gute Annäherung tatsächlich notwendig sind, ist im praktischen Einsatz häufig schwer abzuschätzen. Zumal das Netz auch nicht zu viele verdeckte Neuronen enthalten sollte, da mit der Komplexität des Modells die Gefahr der Überanpassung (*overfitting*) an die gegebenen Daten steigt. Dabei approximiert das Netz nicht nur den zugrundeliegenden systematischen Zusammenhang, sondern passt sich auch an zufällige Störgrößen an. Der geschätzte Zusammenhang ist dann nur schlecht generalisierbar – der Schätzfehler für neue, unbekannte Daten ist groß.

Abb. C-5 verdeutlicht die Schwierigkeit, ein angemessen komplexes Modell zu finden, an einem artifiziellen Datensatz. Ausgehend von der Funktion $g(x) = \sin(2\pi x)$ wurden verrauschte Daten $y = g(x) + \epsilon$ mit zufälligen, gleichverteilten Störgrößen $\epsilon \sim \mathcal{U}(-0,35; 0,35)$ erzeugt. Drei künstliche neuronale Netze wurden auf diese Daten trainiert.²⁾ Das Netz mit nur zwei verdeckten Neuronen enthält nicht genügend Freiheitsgrade, um der Schwingung der Sinuskurve zu folgen (Abb. C-5 a). Demhingegen bildet das Netz mit fünf verdeckten Neuronen den Zusammenhang $g(x)$ perfekt ab (Abb. C-5 b). Das Netz mit elf Neuronen ist so flexibel, dass es eine Kurve durch alle Datenpunkte legen kann (Abb. C-5 c). Diese Kurve bildet aber nicht den eigentlich interessierenden Zusammenhang $g(x) = \sin(2\pi x)$ ab.

¹⁾ Vgl. grundlegend Cybenko [Approximation]; Funahashi [Mappings]; Hecht-Nielsen [Theory]; Hornik/Stinchcombe/White [Approximators]; Hornik/Stinchcombe/White [Approximation]. Vgl. auch Kruse u. a. [Intelligence] 50 ff.

²⁾ Mit dem Extreme-learning-machine-Algorithmus, siehe Abschnitt 3.i, S. 118 ff.

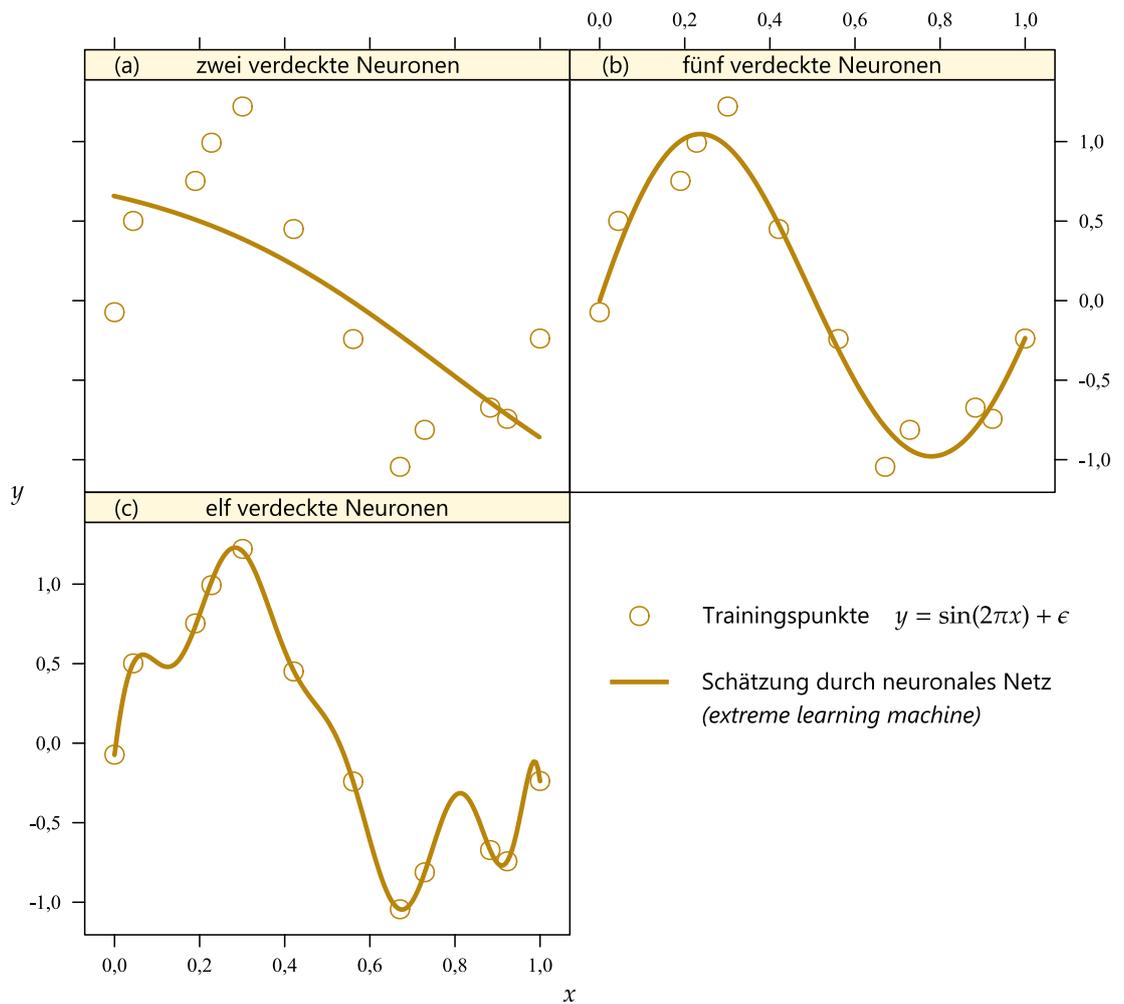


Abb. C-5: Approximation einer verrauschten Sinusfunktion durch neuronale Netze mit zwei, fünf und elf verdeckten Neuronen

Aus diesem Grund ist eine perfekte Annäherung an empirische Daten, die neben dem gesuchten Modellzusammenhang immer auch Rauschen (*noise*) enthalten, i. A. nicht wünschenswert. Das in Abb. C-5 dargestellte Problem, einen Mittelweg zwischen Unter- und Überanpassung zu finden, ist in der Statistik als *bias-variance trade-off* oder *bias-variance dilemma* bekannt.¹⁾ Ein sparsames Modell ist immer schon für bestimmte Strukturen voreingenommen (*biased*). So wie ein lineares Regressionsmodell in jedem Datensatz eine Gerade findet, „entdeckt“ auch ein neuronales Netz nur Strukturen,

¹⁾ Vgl. auch im Weiteren Geman/Bienenstock/Doursat [Dilemma].

die es mit seiner begrenzten Anzahl an Sigmoidalfunktionen abbilden kann. Wenn das Modell nicht zur Struktur passt, die die Daten erzeugt hat, entstehen Schätzfehler. Je mehr verdeckte Neuronen ein künstliches neuronales Netz hat, desto unvoreingenommener (*unbiased*) kann es sich an Strukturen in den Daten anpassen. Dem Theorem der universellen Approximation zufolge, lässt sich die Flexibilität eines künstlichen neuronalen Netzes durch Hinzufügen immer weiterer Neuronen unendlich steigern. Dadurch werden sie zu *nichtparametrischen Schätzern* in dem Sinne, dass sie keine Annahmen über die Struktur des Zusammenhangs zwischen den betrachteten Variablen enthalten. Das „Dilemma“ besteht nun darin, dass mit der Flexibilität des Schätzers zwar der *bias* abnimmt, dafür aber die Abhängigkeit von den gegebenen Datenpunkten zunimmt (man sagt in diesem Fall, dass die *Varianz* des Schätzers hoch ist), was wiederum zu einer verzerrten Schätzung des zugrundeliegenden Zusammenhangs führt (wie in Abb. C-5 c).

In Kapitel D.XI (S. 258 ff.) werden Techniken vorgestellt, die eine angemessene Modellkomplexität sicherstellen sollen, um *under-* und *overfitting* zu vermeiden. Aber zunächst werden im folgenden Abschnitt die Grundlagen der Parametrisierung künstlicher neuronaler Netze betrachtet.

3. Parametrisierung von künstlichen neuronalen Netzen

a) Überwachtes Lernen mit einer Fehlerfunktion

Die Verbindungsgewichte w sind die freien Parameter eines künstlichen neuronalen Netzes. Wie bei anderen statistischen Verfahren auch, werden sie anhand gegebener Daten (z. B. beobachteter Werte) geschätzt. Im Zusammenhang mit künstlichen neuronalen Netzen und anderen Verfahren des maschinellen Lernens spricht man davon, dass das Modell aus Beispieldaten lernt oder auf Trainingsdaten trainiert wird. Je nach Problemtyp werden drei Arten des Lernens unterschieden:¹⁾

- Beim *überwachten Lernen* ist zu jedem Trainingsinput x der korrekte Wert der abhängigen Variablen y bekannt. Somit kann dem neuronalen Netz für jeden Output \hat{y} , den es während des Trainings erzeugt, ein Fehlerwert $E(y, \hat{y})$ zurückgemeldet werden. Überwachtes Lernen ist sowohl bei metrisch als auch bei nominal skalierten abhängigen Variablen möglich. Im ersten Fall handelt es sich um ein Regressionsproblem wie z. B. die Schätzung einer Kostenfunktion, im zweiten Fall um ein Klassifikationsproblem im Sinne einer Diskriminanzanalyse, d. h. die Gruppen, in die ein Inputvektor eingeordnet werden soll, sind vorgegeben.²⁾ Ein typisches Klassifikationsproblem, für das künstliche neuronale Netze eingesetzt werden, ist die Handschriftenerkennung.
- Beim *unüberwachten Lernen* sind keine abhängigen Variablen vorhanden. Ziele unüberwachten Lernens können z. B. das Bilden von Clustern (oh-

¹⁾ Vgl. auch zu den einzelnen Lernarten Bishop [Machine Learning] 3; Sammut/Webb [Machine Learning].

²⁾ Vgl. auch Cheng/Titterton [Statistical Perspective] 17 f.; Backhaus u. a. [Analysemethoden] 189.

ne vordefinierte Gruppen), die Schätzung einer Wahrscheinlichkeitsdichtefunktion oder die Reduktion der Inputdimensionen – beispielsweise durch Hauptkomponentenanalyse (*principal component analysis*) oder Unabhängigkeitsanalyse (*independent component analysis*) – sein.¹⁾

- Beim *bestärkenden Lernen* (*reinforcement learning*) werden künstliche Agenten für richtige Entscheidungen belohnt.²⁾ Typischerweise handelt es sich um sequentielle Entscheidungen wie z. B. Spielzüge in einem Schachspiel, die erst nach einer zeitlichen Verzögerung belohnt werden. Die Belohnungsfunktion steht häufig für ein langfristiges Ziel, beispielsweise den Gewinn der Schachpartie. Der Agent soll eine Politik lernen, die langfristig optimal ist. Der Unterschied zum überwachten Lernen liegt darin, dass dem Agenten keine Beispiele für korrekte Entscheidungen vorgegeben werden. Im Gegensatz zum unüberwachten Lernen erfährt der Agent aber eine Rückmeldung.

Da die Schätzung von Kostenfunktionen ein Regressionsproblem ist, beschränken sich die folgenden Ausführungen auf Methoden des *überwachten Lernens*. Gegeben sei ein Trainingsdatensatz mit N Trainingspunkten (Beobachtungen). Ein Trainingspunkt besteht aus einem Inputvektor $\mathbf{x}^{(n)}$ und einem Vektor der abhängigen Variablen $\mathbf{y}^{(n)}$, der auch als „Zielvektor“ (*target vector*) bezeichnet wird. Wie auch bei klassischen, additiven Regressionsverfahren wird beim überwachten Lernen üblicherweise die Summe der quadrierten Fehler über alle Trainingsmuster ($n = 1, \dots, N$) und abhängigen Variablen ($k = 1, \dots, O$) minimiert:

¹⁾ Vgl. Hinton/Sejnowski [Unsupervised Learning].

²⁾ Vgl. grundlegend Sutton/Barto [Reinforcement Learning]; Szepesvári [Reinforcement Learning]; Wiering/van Otterlo [Reinforcement Learning].

$$\min E(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^O (\hat{y}_k^{(n)} - y_k^{(n)})^2. \quad (\text{C.7})$$

Für das durch (C.6) beschriebene Netz bedeutet das

$$\min E(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^O \left(\sum_{j=1}^H w_{jk} \frac{1}{1 + e^{-(\sum_{i=0}^I w_{ij} x_i^{(n)})}} - y_k^{(n)} \right)^2. \quad (\text{C.8})$$

Da die Trainingspunkte $(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)})$ gegeben sind, ist die Fehlerfunktion E nur vom Gewichtsvektor \boldsymbol{w} abhängig. Der Minimierungsansatz (C.7) bzw. (C.8) ist äquivalent zu einer Maximum-Likelihood-Schätzung unter der Annahme unkorrelierter, normalverteilter Zielvariablen \boldsymbol{y} .¹⁾

b) Optimierung der Fehlerfunktion durch Gradientenabstieg

Es ist unwahrscheinlich, für Gleichung (C.8) in vertretbarer Rechenzeit eine analytische Lösung – durch Nullsetzen des Gradienten $\nabla E(\boldsymbol{w})$ – zu finden.²⁾ Deshalb greift man auf numerische Lösungsverfahren zurück. Grundsätzlich kommen dafür die üblichen Verfahren der nichtlinearen Optimierung in Frage. Ausgehend von einer Startlösung $\boldsymbol{w}^{(0)}$ versuchen sie über iterative Änderungen

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} + \Delta \boldsymbol{w}^{(\tau)} \quad (\text{C.9})$$

mit $\boldsymbol{w}^{(\tau)}$ Gewichtsvektor in Iterationsschritt τ ,

$\Delta \boldsymbol{w}^{(\tau)}$ Änderung des Gewichtsvektors in Iterationsschritt τ

eine gute Lösung zu finden. Die Verfahren unterscheiden sich vor allem in der Wahl von $\Delta \boldsymbol{w}^{(\tau)}$.

¹⁾ Vgl. Bishop [Neural Networks] 195 ff.; Bishop [Machine Learning] 233.

²⁾ Vgl. auch im Weiteren Bishop [Machine Learning] 237; Crone [Neuronale Netze] 195.

Ein einfaches Verfahren, um lokale Minima zu finden, ist der Gradientenabstieg (*gradient descent*). Er nutzt den Gradienten erster Ordnung

$$\nabla E(\mathbf{w}^{(\tau)}) = \left(\frac{\partial E}{\partial w_{01}}(\mathbf{w}^{(\tau)}), \dots, \frac{\partial E}{\partial w_{ij}}(\mathbf{w}^{(\tau)}), \dots, \frac{\partial E}{\partial w_{jk}}(\mathbf{w}^{(\tau)}), \dots, \frac{\partial E}{\partial w_{HO}}(\mathbf{w}^{(\tau)}) \right)^T. \quad (\text{C.10})$$

Geometrisch gesehen zeigt der Gradient $\nabla E(\mathbf{w}^{(\tau)})$ in die Richtung des steilsten Anstiegs der Fehlerfunktion E an der Stelle $\mathbf{w}^{(\tau)}$ (siehe Abb. C-6).

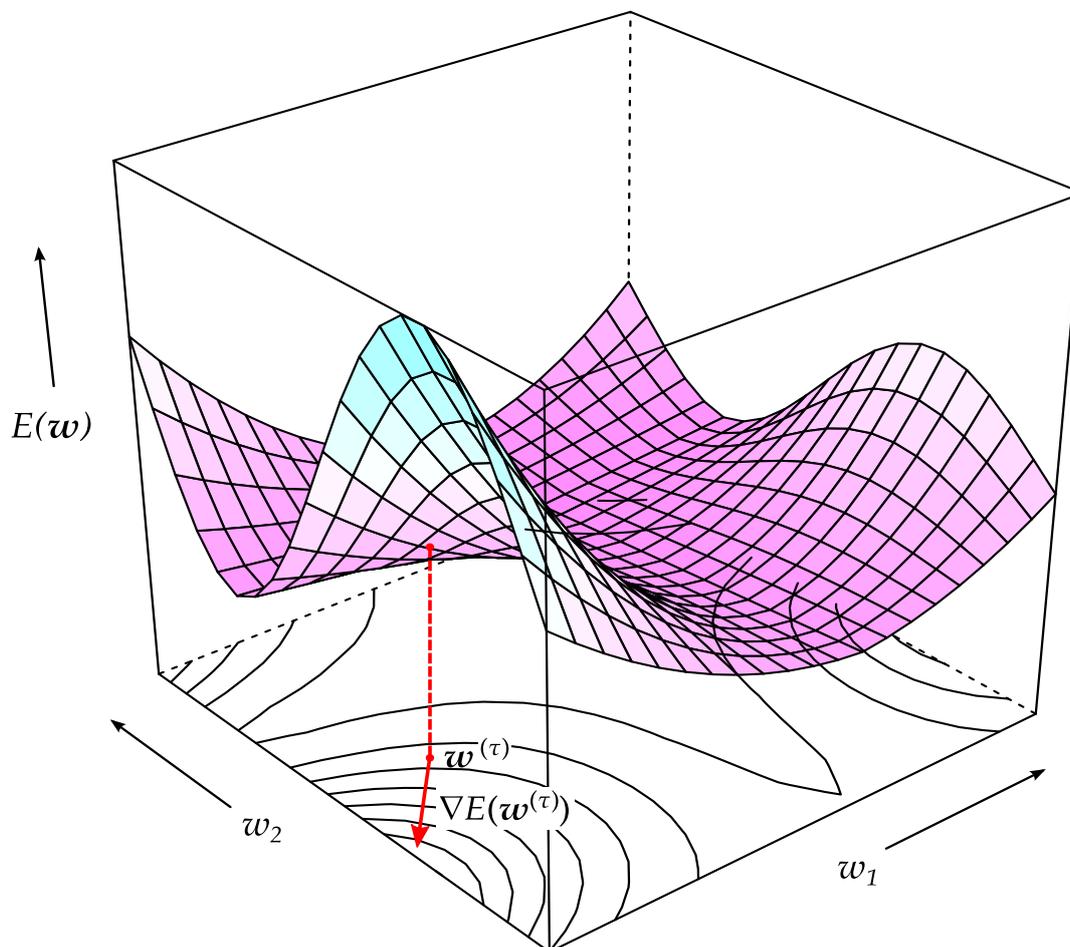


Abb. C-6: Geometrische Interpretation des Gradienten einer Fehlerfunktion

Um den Fehler zu verringern, schreitet der Algorithmus in Richtung des negierten Gradienten voran:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)}) \quad (\text{C.11})$$

mit einer Lernrate $\eta > 0$.

Zusammenfassend zeigt Algorithmus C-1 den Ablauf des Gradientenabstiegsverfahrens als Algorithmus in Pseudocode.

Algorithmus Gradientenabstieg

- 1: Initialisiere Gewichtsvektor $\boldsymbol{w}^{(0)}$.
 - 2: $\tau \leftarrow 0$
 - 3: **wiederhole**
 - 4: **für** $n = 1$ bis N **führe aus**
 - 5: Berechne $\hat{y}^{(n)}$ mit $\boldsymbol{w}^{(\tau)}$ und $\boldsymbol{x}^{(n)}$. (C.6)
 - 6: **ende für**
 - 7: Berechne $\nabla E(\boldsymbol{w}^{(\tau)})$ mit allen $\{(\hat{y}^{(n)}, y^{(n)})\}$. (C.7), (C.10)
 - 8: $\boldsymbol{w}^{(\tau+1)} \leftarrow \boldsymbol{w}^{(\tau)} - \eta \nabla E(\boldsymbol{w}^{(\tau)})$ (C.11)
 - 9: $\tau \leftarrow \tau + 1$
 - 10: **bis** Abbruchkriterium erfüllt,
 z. B. $(\tau \geq \tau^{\max}) \vee (\nabla E(\boldsymbol{w}^{(\tau-1)}) = 0) \vee (E(\boldsymbol{w}^{(\tau-1)}) \leq E^{\text{Ziel}})$.
-

Algorithmus C-1: Gradientenabstieg

- c) Fehler-Rückpropagation zur Berechnung des Gradienten der Fehlerfunktion

Der Gradientenabstieg wird nun auf das neuronale Netz aus Abb. C-4 (S. 88) angewandt. Vorerst genügt es, den Fehler für einen Trainingspunkt n

$$E^{(n)}(\boldsymbol{w}) = \frac{1}{2} \sum_{k=1}^O (\hat{y}_k^{(n)} - y_k^{(n)})^2 \quad (\text{C.12})$$

zu betrachten. Aus (C.6) ist bekannt, dass $\hat{y}_k = f_{\text{act}}(\text{net}_k) = \text{net}_k = \sum_{j=1}^H w_{jk} o_j$.¹⁾ Damit kann (C.12) unter Verwendung der Kettenregel partiell nach den Gewichten w_{jk} abgeleitet werden:

¹⁾ Dabei bezeichnet der Index k ein Neuron in der Ausgabeschicht und der Index j ein Neuron in der verdeckten Schicht.

$$\frac{\partial E^{(n)}}{\partial w_{jk}} = \frac{\partial E^{(n)}}{\partial \hat{y}_k^{(n)}} \frac{\partial f_{\text{act}}(net_k^{(n)})}{\partial net_k^{(n)}} \frac{\partial net_k^{(n)}}{\partial w_{jk}} = (\hat{y}_k^{(n)} - y_k^{(n)}) \cdot f'_{\text{act}}(net_k^{(n)}) \cdot o_j^{(n)}. \quad (\text{C.13})$$

Wenn die Aktivierungsfunktion $f_{\text{act}}(net_k)$ die Identitätsfunktion ist – wie in (C.6) –, beträgt ihre erste Ableitung konstant $f'_{\text{act}}(net_k) = 1$. Da aber u. a. für Klassifikationsmodelle auch andere Aktivierungsfunktionen in der Ausgabeschicht zulässig sind, ist (C.13) allgemein gehalten.

Desweiteren ist aus (C.6) bekannt, dass $o_j = f_{\text{act}}(net_j) = \frac{1}{1+e^{-(net_j)}}$ und $net_j = \sum_{i=0}^I w_{ij} x_i$.¹⁾ Damit kann die Fehlerfunktion (C.12) ausgehend vom Zwischenergebnis (C.13) weiter nach den Gewichten w_{ij} abgeleitet werden:

$$\begin{aligned} \frac{\partial E^{(n)}}{\partial w_{ij}} &= \left[\sum_{k=1}^O \frac{\partial E^{(n)}}{\partial \hat{y}_k^{(n)}} \frac{\partial f_{\text{act}}(net_k^{(n)})}{\partial net_k^{(n)}} \frac{\partial net_k^{(n)}}{\partial o_j^{(n)}} \right] \frac{\partial f_{\text{act}}(net_j^{(n)})}{\partial net_j^{(n)}} \frac{\partial net_j^{(n)}}{\partial w_{ij}} \\ &= \left[\sum_{k=1}^O (\hat{y}_k^{(n)} - y_k^{(n)}) \cdot f'_{\text{act}}(net_k^{(n)}) \cdot w_{jk} \right] \cdot f'_{\text{act}}(net_j^{(n)}) \cdot x_i^{(n)}. \end{aligned} \quad (\text{C.14})$$

Die erste Ableitung der logistischen Aktivierungsfunktion $o_j = f_{\text{act}}(net_j) = \frac{1}{1+e^{-(net_j)}}$ ist $f'_{\text{act}}(net_j) = o_j(1 - o_j)$. Mit den Formeln (C.13) und (C.14) kann der gesamte Gradient $\nabla E^{(n)}$ berechnet werden.

Es ist zu sehen, dass der Faktor $(\hat{y}_k^{(n)} - y_k^{(n)}) \cdot f'_{\text{act}}(net_k^{(n)})$ sowohl in (C.13) als auch in (C.14) vorkommt. Er wird als *Fehlersignal* δ_k bezeichnet:

$$\delta_k = (\hat{y}_k^{(n)} - y_k^{(n)}) \cdot f'_{\text{act}}(net_k^{(n)}). \quad (\text{C.15})$$

Damit verkürzt sich (C.13) zu

$$\frac{\partial E^{(n)}}{\partial w_{jk}} = \delta_k \cdot o_j^{(n)}. \quad (\text{C.16})$$

¹⁾ Dabei bezeichnet der Index i ein Neuron in der Eingabeschicht.

Das Fehlersignal einer verdeckten Schicht ist definiert als

$$\delta_j = \left[\sum_{k=1}^O \delta_k \cdot w_{jk} \right] \cdot f'_{\text{act}}(\text{net}_j^{(n)}). \quad (\text{C.17})$$

Damit verkürzt sich (C.14) zu

$$\frac{\partial E^{(n)}}{\partial w_{ij}} = \delta_j \cdot x_i^{(n)}. \quad (\text{C.18})$$

An Formel (C.17) ist gut zu erkennen, wie das Fehlersignal δ_k entlang der Differenzialkette über die gewichteten Verbindungen w_{jk} rückwärts an die vorgelagerte Netzschicht weitergegeben wird. Diesen Mechanismus bezeichnet man als *Fehler-Rückpropagation* (*error backpropagation*). Er wurde 1986 von Rumelhart, Hinton und Williams bekannt gemacht.¹⁾ Damit konnten sie zum ersten Mal ein künstliches neuronales Netz mit einer verdeckten Schicht trainieren. Die Fehler-Rückpropagation ist aber nicht auf eine verdeckte Schicht beschränkt. Mit der Backpropagation-Formel (C.17) kann ein Fehlersignal zwischen beliebigen benachbarten Schichten rückpropagiert werden.

Die Fehler-Rückpropagation nach (C.15) bis (C.18) ermöglicht eine effiziente Berechnung des Gradienten $\nabla E^{(n)}(\mathbf{w}^{(\tau)})$ für einen gegebenen Gewichtsvektor $\mathbf{w}^{(\tau)}$. Der für das Gradientenverfahren benötigte Gesamtgradient (C.10) über alle Trainingspunkte entspricht der Summe der Teilgradienten $\nabla E(\mathbf{w}^{(\tau)}) = \sum_{n=1}^N \nabla E^{(n)}(\mathbf{w}^{(\tau)})$.

¹⁾ Vgl. Rumelhart/Hinton/Williams [Back-propagating Errors]; Rumelhart/Hinton/Williams [Error Propagation].

d) Der Backpropagation-Algorithmus: Gradientenabstieg mit Fehler-Rückpropagation

In den grundlegenden Veröffentlichungen zur Fehler-Rückpropagation werden die partiellen Ableitungen (C.16) und (C.18) dazu verwendet, die Gewichte nach dem einfachen Gradientenabstieg (C.11) anzupassen:¹⁾

$$\begin{aligned}
 w_{jk}^{(\tau+1)} &= w_{jk}^{(\tau)} - \eta \cdot \sum_{n=1}^N \delta_k \cdot o_j^{(n)}, \\
 w_{ij}^{(\tau+1)} &= w_{ij}^{(\tau)} - \eta \cdot \sum_{n=1}^N \delta_j \cdot x_i^{(n)}
 \end{aligned}
 \tag{C.19}$$

mit δ_k nach (C.15) und δ_j nach (C.17). Dieses Verfahren wird als *Backpropagation-Algorithmus* bezeichnet. Da in (C.19) die Trainingspunkte „stapelweise“ verarbeitet werden – die Gewichtsänderung wird über die Gesamtmenge der Trainingspunkte berechnet –, bezeichnet man dieses Vorgehen als *Batch-Variante* des Backpropagation-Algorithmus. Demhingegen werden in der *Online-Variante* die Gewichte geändert, sobald der Teilgradient für einen Trainingspunkt berechnet wurde:

$$\begin{aligned}
 w_{jk}^{(\tau+1)} &= w_{jk}^{(\tau)} - \eta \cdot \delta_k \cdot o_j^{(n)}, \\
 w_{ij}^{(\tau+1)} &= w_{ij}^{(\tau)} - \eta \cdot \delta_j \cdot x_i^{(n)}.
 \end{aligned}
 \tag{C.20}$$

Ein Vorteil des Online-Algorithmus ist, dass redundante Daten effizienter verarbeitet werden.²⁾ Ein Beispiel soll das verdeutlichen: Angenommen, die Größe des Trainingsdatensatzes wird verdoppelt, indem jeder Trainingspunkt dupliziert wird; dadurch verdoppelt sich die Rechenzeit des Batch-Algorithmus, die Richtung und das Verhältnis der Gewichtsänderungen

¹⁾ Vgl. Rumelhart/Hinton/Williams [Back-propagating Errors]; Rumelhart/Hinton/Williams [Error Propagation].

²⁾ Vgl. auch im Weiteren Bishop [Machine Learning] 241.

bleiben aber gleich, d. h. die Ergebnisse mit einfachem und verdoppeltem Datensatz sind äquivalent. Der Online-Algorithmus hingegen ändert die Gewichte nach jedem einzelnen Punkt und nutzt somit auch die redundanten Punkte für weitere Schritte Richtung Fehlerminimum.

e) Nachteile des Backpropagation-Algorithmus

Der Backpropagation-Algorithmus ist das bekannteste und beliebteste Lernverfahren für künstliche neuronale Netze. Das liegt vor allem daran, dass er recht einfach zu implementieren ist und der erste Algorithmus war, mit dem verdeckte Schichten trainiert werden konnten. Nichtsdestotrotz hat der Backpropagation-Algorithmus in seiner Grundform einige Nachteile, die seine Anwendung erschweren. In erster Linie sind dies die allgemeinen Nachteile einfacher Gradientenverfahren erster Ordnung:¹⁾

(a) *Lokale Suche*: Gradientenverfahren sind lokale Suchverfahren. Ausgehend von einer initialen Lösung finden sie lokale Optima, die in der Nähe der Startlösung liegen (siehe Abb. C-7 a). Deshalb hat die gewählte Startlösung einen großen Einfluss darauf, wie gut die am Ende erreichte Lösung ist und wie schnell sie gefunden wird. Die Wahrscheinlichkeit, ein globales Optimum zu finden, steigt, wenn der Algorithmus mehrmals mit verschiedenen Startlösungen ausgeführt wird. Bei stark zerklüfteten „Fehlergebirgen“, können indessen sehr viele Durchgänge notwendig sein, um auf diese Weise eine gute Lösung zu finden.

(b) *Sattelpunkte der Fehlerfunktion*: Die Fehlerfunktion E kann Sattelpunkte enthalten, an denen der Gradient $\nabla E(\boldsymbol{w}) = 0$ ist, d. h. alle partiellen Ab-

¹⁾ Vgl. für eine ähnliche Übersicht Soria/Martin/Lisboa [Training Methods] 21, mit weiteren Nachweisen; vgl. auch Kruse u. a. [Intelligence] 66 ff.; Backhaus/Erichson/Weiber [Fortgeschrittene Analysemethoden] 197 f.

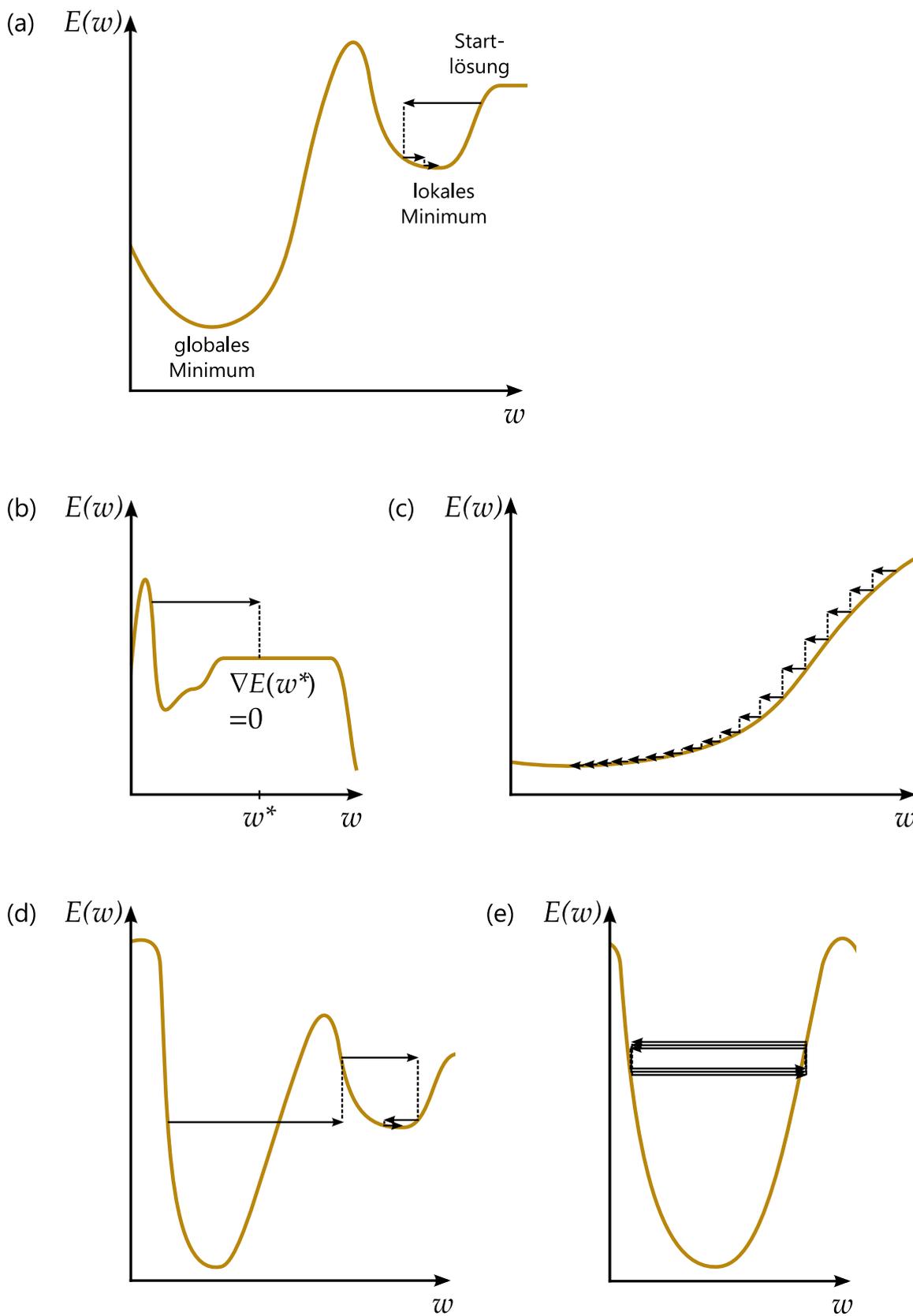


Abb. C-7: Nachteile des Backpropagation-Algorithmus

leitungen null sind (siehe Abb. C-7 b). Wenn der Algorithmus an einem Sattelpunkt anlangt, ändern sich die Gewichte nicht mehr (siehe (C.11)), obwohl noch kein Minimum erreicht ist. Je mehr Dimensionen die Fehlerfunktion hat – je mehr Parameter w also geschätzt werden, desto unwahrscheinlicher ist allerdings das Auftreten von Sattelpunkten.

- (c) *Zu kleine Lernrate*: Besondere Probleme bereitet die richtige Wahl der Lernrate η (siehe (C.11), (C.19), (C.20)). Sie bestimmt – zusammen mit der Höhe des Gradienten – die Schrittweite des Algorithmus. Wird sie zu klein gewählt, dauert es lange, bis E gegen Null konvergiert (siehe Abb. C-7 c).
- (d) *Zu große Lernrate*: Ist die Lernrate zu groß, besteht die Gefahr, dass sich der Algorithmus von einem Optimum wegbewegt oder es sogar überspringt (siehe Abb. C-7 d).
- (e) *Oszillation*: In schmalen „Fehlerschluchten“ kann der Fall eintreten, dass die Schrittweite des Algorithmus mit der Breite der Schlucht übereinstimmt, so dass der Algorithmus zwischen zwei Punkten oszilliert, ohne zu konvergieren (siehe Abb. C-7 e).

Weitere Probleme des Backpropagation-Algorithmus sind:

- (f) *Sättigung der Neuronen*: Über die Fehlersignale δ_k (C.15) und δ_j (C.17) fließen die ersten Ableitungen der Aktivierungsfunktionen $f_{act}(net)$ multiplikativ in die Formeln für die Gewichtsänderungen (C.19), (C.20) ein. In den Extrembereichen einer sigmoidalen Aktivierungsfunktion ist aber die erste Ableitung annähernd null (siehe Abb. C-2, S. 85). Wenn also das Nettoeingangssignal $net_j^{(n)}$ eines Neurons betragsmäßig so groß ist, dass $f'_{act}(net_j^{(n)}) \approx 0$, dann ändern sich die Gewichte der eingehenden Verbindungen w_{ij} nicht mehr, obwohl der Fehler noch größer als Null

ist.¹⁾ Tritt dieser Sättigungseffekt bei einem Großteil der N Trainingspunkte auf, verhindert er die weitere Konvergenz des Backpropagation-Algorithmus gegen $E(w) = 0$. Sowohl die Batch- als auch die Online-Variante des Backpropagation-Algorithmus fahren sich im praktischen Einsatz regelmäßig auf diese Weise fest. Wie schnell die Neuronen gesättigt sind, hängt von der Normierung der Eingabewerte $x_i^{(n)}$, den Initialwerten der Gewichte und der Aktivierungsfunktion der verdeckten Neuronen ab. Eine logistische Aktivierungsfunktion erreicht ihren Sättigungsbereich ungefähr bei einem Nettoeingangssignal $|net| > 5$ (siehe Abb. C-2, S. 85), die Tangens-Hyperbolicus-Funktion ungefähr bei $|net| > 2$. Um eine schnelle Sättigung zu vermeiden, sollten sowohl die Eingabewerte als auch die Initialgewichte regelmäßig nicht über das Intervall $[-2; 2]$ hinausreichen. Häufig werden die Initialgewichte einfach zufällig aus der Standardnormalverteilung gezogen. In der Praxis ist auch die Gewichtsinitialisierung nach Nguyen und Widrow beliebt, bei der für jede Synapse eine individuell optimale Initialisierung berechnet wird.²⁾ Möglichkeiten zur Normierung der Eingabewerte werden in Abschnitt D.VII.1 (S. 230 ff.) diskutiert.

Als pragmatische Maßnahme wird außerdem vorgeschlagen, einen kleinen konstanten Betrag – z. B. 0,01 – zur Ableitung $f'_{act}(net_j^{(n)})$ hinzuzuzuglieren, so dass dieser Term nicht null werden kann.³⁾ Damit wird gleichzeitig ein weiterer Nachteil des Backpropagation-Verfahrens ausgeglichen: Da die erste Ableitung einer sigmoidalen Funktion maximal den Wert 0,25 annimmt – an ihrem Wendepunkt –, wird das Fehlersig-

¹⁾ Vgl. Soria/Martin/Lisboa [Training Methods] 21, mit weiteren Nachweisen.

²⁾ Vgl. Nguyen/Widrow [Initial Values].

³⁾ Vgl. Fahlman [Learning Speed] 8.

nal bei der Rückpropagation durch das Netz stark abgeschwächt. Bei den vorletzten Gewichten kommt maximal noch ein Viertel des Fehler-signals an; bei Netzen mit mehreren verdeckten Schichten erhalten die ersten Verbindungen nur noch einen kleinen Bruchteil des Fehlersignals und ändern sich folglich kaum. Durch die Addition oder Multiplikation einer Konstante kann dieser in der Regel unerwünschte Effekt vermindert werden.

(g) *Quadratische Fehlerfunktion*: Ausgangspunkt des Backpropagation-Algorithmus ist üblicherweise die Summe der quadrierten Fehler $E(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^O (\hat{y}_k^{(n)} - y_k^{(n)})^2$ (C.7). Durch diese Fehlerfunktion werden größere Abweichungen überproportional stärker gewichtet. In den Originalveröffentlichungen zum Backpropagation-Algorithmus¹⁾ wird die Verwendung quadrierter Fehler nicht formal begründet. Sie ist wohl eher durch mathematische Einfachheit – vor allem bei der Berechnung der Ableitungen – und die allgemeine Vertrautheit mit Kleinste-Quadrate-Schätzern motiviert. Grundsätzlich ist die quadratische Fehlerfunktion dadurch begründbar, dass sie äquivalent zu einer Maximum-Likelihood-Schätzung unter der Annahme normalverteilter Residuen ist.²⁾ Problematisch ist sie jedoch bei nicht normalverteilten Störgrößen. Da sie größere Fehler überproportional gewichtet, kann schon ein einzelner Ausreißer, der die Verteilung an einem Ende schwerer und damit nicht-normalverteilt macht, die Schätzung dominieren. Abb. C–8 a illustriert dieses Problem am Beispiel der linearen Regression. Es kann vermieden werden, wenn man statt der quadratischen Fehlerfunk-

¹⁾ Vgl. Rumelhart/Hinton/Williams [Back-propagating Errors]; Rumelhart/Hinton/Williams [Error Propagation].

²⁾ Vgl. Burrascano [Norm Selection]; Bishop [Machine Learning] 233; vgl. auch im Weiteren Bishop [Neural Networks] 195 ff.

tion ein *robustes Fehlermaß* wie z. B. die Manhattan-Metrik einsetzt (siehe Abb. C-8 b).

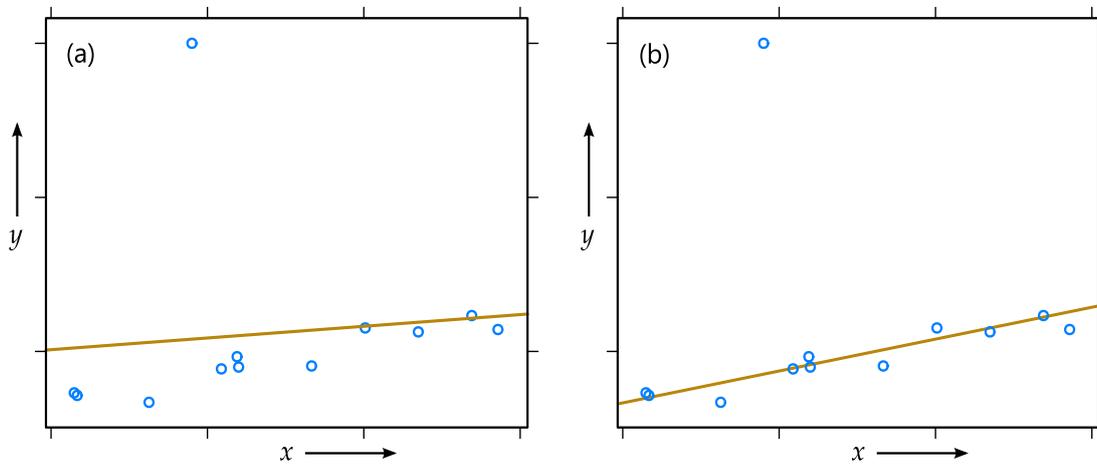


Abb. C-8: Schätzung einer Regressionsgeraden nach der Methode der kleinsten Fehlerquadrate (a) und mit der Manhattan-Metrik (b)

Im folgenden Abschnitt f wird eine generalisierte Fehlerfunktion vorgestellt, die es ermöglicht, robustere Fehlermaße in den Backpropagation-Algorithmus und andere Lernverfahren einzubinden. Darauf aufbauend folgt in Abschnitt g eine Übersicht zu Abwandlungen und Alternativen des Backpropagation-Algorithmus, die einige der oben genannten Probleme lösen. Abschnitt h zeigt, wie die Nachteile einer lokalen Suche durch die Verbindung mit einer global suchenden Metaheuristik umgangen werden können. Abschließend gibt Abschnitt i einen Ausblick auf die in den letzten Jahren sehr erfolgreich eingesetzte *extreme learning machine* als Alternative zum klassischen Training mit Fehlerrückpropagation.

f) Alternative Fehlerfunktionen mit Minkowski-Metrik

In den Backpropagation-Algorithmus können alternative Fehlerfunktionen (Zielfunktionen) integriert werden. Am besten zu handhaben sind Fehlerfunktionen, die an eine Minkowski-Metrik (ℓ_p -Metrik) anknüpfen. Minkowski-Metriken basieren auf der Klasse der p -Normen und definieren den Abstand zwischen zwei Vektoren $\mathbf{a} = (a_1, \dots, a_D)^T$ und $\mathbf{b} = (b_1, \dots, b_D)^T$ als

$$d_p(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_p = \left(\sum_{i=1}^D |a_i - b_i|^p \right)^{\frac{1}{p}}. \quad (\text{C.21})$$

Für $p = 2$ erhält man die euklidische Distanz, für $p = 1$ die Manhattan-Metrik (Cityblock-Metrik, Minkowski-1-Metrik, *taxicab distance*) und für $p = \infty$ die Maximum-Metrik (Tschebyschew-Metrik). Der Name der Manhattan-Metrik stammt daher, dass für ein Taxi in Manhattan der kürzeste Weg zwischen zwei Orten nicht die euklidische Distanz sondern die Strecke entlang der rechtwinklig angeordneten Blöcke ist.

Beim Trainieren künstlicher neuronaler Netze mit Minkowski-Metriken verzichtet man in der Regel auf den Exponenten $1/p$ und erhält so eine generalisierte Fehlerfunktion der Form

$$E_p(\boldsymbol{\omega}) = \frac{1}{p} \sum_{n=1}^N \sum_{k=1}^O |\hat{y}_k^{(n)} - y_k^{(n)}|^p. \quad (\text{C.22})$$

Mit $p = 2$ entspricht sie der traditionellen Backpropagation-Zielfunktion mit quadrierten Fehlern (C.7). Mit $p = 1$ ergibt sich die Summe der absoluten Fehler entsprechend der Manhattan-Metrik.

Abb. C-9 zeigt den Verlauf von (C.22) für einen einzelnen Fehler $(\hat{y}_k^{(n)} - y_k^{(n)})$ mit verschiedenen p -Werten.

¹⁾ Vgl. Hanson/Burr [Minkowski-r Back-Propagation] 350.

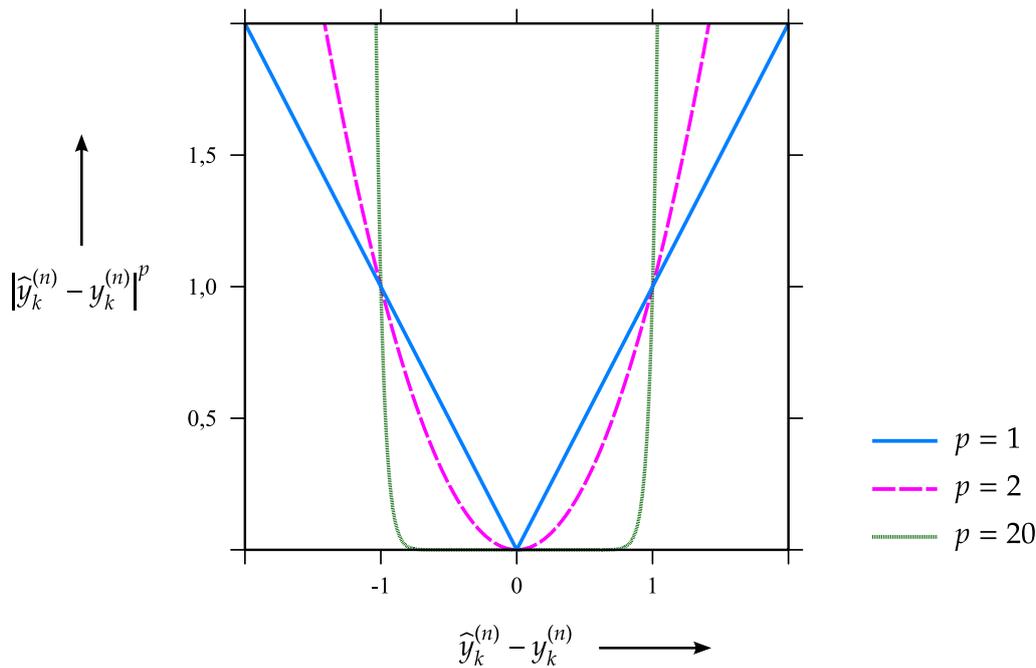


Abb. C-9: Generalisierte Fehlerfunktionen mit $p = 1$, $p = 2$ und $p = 20$

Um die generalisierte Fehlerfunktion E_p (C.22) in die Ableitungsketten (C.13) und (C.14) einzugliedern, die bei der Rückpropagation des Fehlers zum Tragen kommen (siehe S. 97), muss jeweils nur das erste partielle Ableitungsglied $\partial E^{(n)} / \partial \hat{y}_k^{(n)}$ geändert werden. Aus (C.13) wird so

$$\begin{aligned} \frac{\partial E_p^{(n)}}{\partial w_{jk}} &= \frac{\partial E_p^{(n)}}{\partial \hat{y}_k^{(n)}} \frac{\partial f_{\text{act}}(\text{net}_k^{(n)})}{\partial \text{net}_k^{(n)}} \frac{\partial \text{net}_k^{(n)}}{\partial w_{jk}} \\ &= |\hat{y}_k^{(n)} - y_k^{(n)}|^{p-1} \cdot \text{sgn}(\hat{y}_k^{(n)} - y_k^{(n)}) \cdot f'_{\text{act}}(\text{net}_k^{(n)}) \cdot o_j^{(n)} \end{aligned} \tag{C.23}$$

mit der Vorzeichenfunktion sgn .¹⁾ Formel (C.14) ändert sich analog.

An Formel (C.23) ist gut zu erkennen, dass bei der Minimierung der Manhattan-Fehlerfunktion ($p = 1$) die Höhe des Fehlers gänzlich vernachlässigt wird. Das erklärt ihre Robustheit gegenüber Ausreißern. Da nur das Vorzeichen des Fehlers eine Rolle spielt, spricht man auch von einer Median-

¹⁾ Vgl. Hanson/Burr [Minkowski-r Back-Propagation] 350; Bishop [Neural Networks] 209.

Schätzung. Dieser Zusammenhang wird in Abb. C-8 b (S. 105) deutlich: Bei der Regression mit Manhattan-Metrik liegen genauso viele Datenpunkte über wie unter der geschätzten Geraden. Dadurch gibt es ebenso viele Fehler mit positivem wie mit negativem Vorzeichen, so dass die Summe der Vorzeichenfunktion über alle N Datenpunkte $\sum_{n=1}^N \text{sgn}(\hat{y}^{(n)} - y^{(n)}) = 0$ ist.¹⁾ Damit ist auch die erste Ableitung der Fehlerfunktion Null, die Fehlerfunktion selbst demnach an einem Extrempunkt. Mit der Manhattan-Metrik wird also der bedingte Median der Zielvariablen y in Abhängigkeit von den Eingabevariablen x geschätzt.²⁾ Die Kleinste-Quadrate-Schätzung hingegen berechnet den bedingten Mittelwert der Zielvariablen (siehe Abb. C-8 a).

Ebenso wie die quadratische Zielfunktion lassen sich auch generalisierte Fehlerfunktionen E_p (C.22) mit $p \neq 2$ durch das Maximum-Likelihood-Prinzip begründen.³⁾ So ist die Minimierung der Manhattan-Fehlerfunktion äquivalent zu einer Maximum-Likelihood-Schätzung unter der Annahme, dass die Residuen Laplace-verteilt sind. Dagegen korrespondiert eine Tschebyschew-Fehlerfunktion ($p = \infty$) mit gleichverteilten Residuen. Prinzipiell werden mit $p < 2$ unimodale (eingipflige) Verteilungen angenommen, die dickere Enden als eine Normalverteilung haben.⁴⁾ Für verrauschte Daten sind sie in der Regel eine gute Wahl, zumal beispielsweise mit $p = 1$ auch normalverteilte Daten – sollte dieser Fall doch einmal eintreten – nicht wesentlich schlechter geschätzt werden als mit $p = 2$.⁵⁾ Sind die Residuen indessen multimodal und unregelmäßig verteilt, was insbesondere in der

¹⁾ Vgl. auch im Weiteren Bishop [Neural Networks] 209 f.

²⁾ Vgl. auch im Weiteren Bishop [Neural Networks] 201 ff.

³⁾ Vgl. auch im Weiteren Burrascano [Norm Selection].

⁴⁾ Vgl. auch im Weiteren Hanson/Burr [Minkowski-r Back-Propagation] 352.

⁵⁾ Vgl. Burrascano [Norm Selection] 127 f.; Crone [Neuronale Netze] 253 ff.

Anfangsphase des Trainings vorkommt, ist die Tschebyschew-Fehlerfunktion besser geeignet.¹⁾

Neben den bisher aufgeführten Fehlerfunktionen mit Minkowski-Metriken können auch andere Fehlerfunktionen verwendet werden, die z. B. die tatsächlich entstehenden Fehlerkosten zur Fehlerbewertung verwenden. Wenn etwa die betriebswirtschaftlichen Kosten, die entstehen, wenn der Output unterschätzt wird, höher sind als die Kosten einer Überschätzung – das könnte z. B. bei einer Bedarfsprognose der Fall sein –, können diese Kosten direkt in einer asymmetrischen Fehlerfunktion berücksichtigt werden.²⁾

g) Erweiterungen und Alternativen zum Backpropagation-Algorithmus

α) Backpropagation mit Impulsterm

Bei der praktischen Anwendung des Backpropagation-Algorithmus bereitet die Wahl der Lernrate η die größten Probleme (siehe Abschnitt e, Absätze (c) bis (e)). Deshalb werden in zahlreichen Veröffentlichungen Anpassungen der ursprünglichen Iterationsregel $\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)})$ (siehe (C.11), (C.19), (C.20)) vorgeschlagen, die den Algorithmus schneller konvergieren lassen sowie Oszillation und andere unerwünschte Effekte vermeiden sollen.

Schon Rumelhart, Hinton und Williams selbst weisen darauf hin, dass das Backpropagation-Verfahren durch die Einführung eines Impulsterms (*mo-*

¹⁾ Vgl. Burrascano [Norm Selection] 128 f.

²⁾ Crone [Neuronale Netze] 261 ff. diskutiert ausführlich Konzeptionen und Anwendungen betriebswirtschaftlich motivierter Fehlerfunktionen für künstliche neuronale Netze.

momentum term) α verbessert werden kann. Demnach sollen die Gewichte nach der Regel

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \Delta\mathbf{w}^{(\tau)} \quad (\text{C.24})$$

mit $\Delta\mathbf{w}^{(\tau)} = -\eta\nabla E(\mathbf{w}^{(\tau)}) + \alpha\Delta\mathbf{w}^{(\tau-1)}$,
 $0 \leq \alpha \leq 1$

angepasst werden.¹⁾ Dadurch dass zur aktuellen Gewichtsänderung ein Teil der letzten Änderung addiert wird, behält der Algorithmus ähnlich wie ein physikalischer Körper seinen Impuls bei. Selbst wenn der Gradient $\nabla E(\mathbf{w}^{(\tau)}) = 0$ ist, werden die Gewichte weiter verändert, so dass Sattelpunkte der Fehlerfunktion verlassen werden können (siehe Abschnitt e, Absatz (b)). Außerdem konvergiert der Algorithmus auch bei kleinen Lernraten schneller, da die Gewichtsänderung $\Delta\mathbf{w}^{(\tau)}$ immer größer wird, je länger sie in der selben Richtung verläuft (siehe Abschnitt e, Absatz (c) und Abb. C-7 c). Schließlich werden die Probleme einer zu großen Lernrate reduziert, weil der Impulsterm die Gewichtsänderung bei einem Vorzeichenwechsel des Gradienten verringert, so dass der Algorithmus weniger oszilliert (siehe Abschnitt e, Absätze (d) und (e)).

Insgesamt kann ein Impulsterm die Anzahl der notwendigen Iterationsschritte des Backpropagation-Algorithmus um mehr als die Hälfte reduzieren.²⁾ Kritisch ist allerdings, dass zusätzlich zur Lernrate η ein zweiter Parameter α festgesetzt werden muss. Welche Werte von η und α zu guten Ergebnissen führen, hängt von der Struktur der Fehlerfunktion im konkreten Fall ab, so dass diese in einem experimentellen Trial-and-Error-Prozess ermittelt werden müssen.

¹⁾ Vgl. Rumelhart/Hinton/Williams [Back-propagating Errors] 535; vgl. auch im Weiteren Rumelhart/Hinton/Williams [Error Propagation] 330; Bishop [Neural Networks] 267 f.

²⁾ Vgl. Kruse u. a. [Intelligence] 75 f.

Etliche andere Erweiterungen des Backpropagation-Algorithmus versuchen, aus dem Gradienten auf den Verlauf der Fehlerfunktion zu schließen, um die Lernrate dynamisch anzupassen.¹⁾ Hier müssen in der Regel weniger Parameter gesetzt werden als im Backpropagation-Algorithmus mit Impulsterm; gleichwohl handelt es sich bei den meisten Vorschlägen um theoretisch und experimentell kaum fundierte Ad-hoc-Modifikationen.

β) Medianschätzung durch *resilient propagation*

Eine Backpropagation-Abwandlung, deren Überlegenheit gegenüber anderen Verfahren gut belegt ist,²⁾ ist der *Resilient-Propagation-Algorithmus*.³⁾ Er ignoriert die Höhe des Gradienten $\nabla E(\mathbf{w}^{(\tau)})$ und berücksichtigt nur die Vorzeichen der partiellen Ableitungen, um die Gewichte w_{ij} individuell anzupassen:

$$w_{ij}^{(\tau+1)} = \begin{cases} w_{ij}^{(\tau)} - \operatorname{sgn}\left(\frac{\partial E(\mathbf{w}^{(\tau)})}{\partial w_{ij}}\right) \cdot \Delta w_{ij}^{(\tau)} & , \text{ falls } \frac{\partial E(\mathbf{w}^{(\tau-1)})}{\partial w_{ij}} \cdot \frac{\partial E(\mathbf{w}^{(\tau)})}{\partial w_{ij}} \geq 0, \\ w_{ij}^{(\tau-1)} & , \text{ sonst,} \end{cases}$$

$$\text{mit } \Delta w_{ij}^{(\tau)} = \begin{cases} \eta^+ \cdot \Delta w_{ij}^{(\tau-1)} & , \text{ falls } \frac{\partial E(\mathbf{w}^{(\tau-1)})}{\partial w_{ij}} \cdot \frac{\partial E(\mathbf{w}^{(\tau)})}{\partial w_{ij}} > 0, \\ \eta^- \cdot \Delta w_{ij}^{(\tau-1)} & , \text{ falls } \frac{\partial E(\mathbf{w}^{(\tau-1)})}{\partial w_{ij}} \cdot \frac{\partial E(\mathbf{w}^{(\tau)})}{\partial w_{ij}} < 0, \\ \Delta w_{ij}^{(\tau-1)} & , \text{ sonst,} \end{cases} \quad (\text{C.25})$$

$$0 < \eta^- < 1 < \eta^+.$$

¹⁾ Vgl. Crone [Neuronale Netze] 201 f.; Zhang u. a. [Dynamic Learning]; jeweils mit weiteren Nachweisen.

²⁾ Vgl. z. B. Fun/Hagan [Levenberg-Marquardt] 471 ff.; Bajramovic/Gruber/Sick [Comparison] 838 ff.; Torrecilla u. a. [Optimising] 5828 f.

³⁾ Vgl. auch im Weiteren Riedmiller/Braun [RPROP]; Anastasiadis/Magoulas/Vrahatis [Resilient Propagation].

Der erste Teil der Formel bedeutet, dass das Gewicht w_{ij} um einen Betrag $\Delta w_{ij}^{(\tau)}$ entgegen dem Vorzeichen der partiellen Ableitung $\frac{\partial E(\mathbf{w}^{(\tau)})}{\partial w_{ij}}$ angepasst wird, falls sich das Vorzeichen der partiellen Ableitung seit der letzten Iteration nicht geändert hat. Andernfalls wird die vorhergehende Gewichtsänderung zurückgenommen. Die Idee dahinter ist, dass ein Vorzeichenwechsel darauf hindeutet, dass der vorangehende Schritt zu groß war und ein Minimum übersprungen wurde (siehe Abb. C-7, S. 101). Aus dem selben Grund wird der Änderungsbetrag $\Delta w_{ij}^{(\tau)}$ verringert, falls ein Vorzeichenwechsel stattgefunden hat. Bleibt das Vorzeichen des Gradienten hingegen gleich, wird $\Delta w_{ij}^{(\tau)}$ vergrößert, um in flachen Abschnitten der Fehlerfunktion schneller voranzukommen (siehe Abb. C-7 b).

Weil im Fall von $\frac{\partial E(\mathbf{w}^{(\tau-1)})}{\partial w_{ij}} \cdot \frac{\partial E(\mathbf{w}^{(\tau)})}{\partial w_{ij}} < 0$ die vorhergehende Gewichtsänderung zurückgenommen wird, ändert sich in der darauffolgenden Iteration wieder das Vorzeichen des Gradienten. Damit das nicht zu einem endlosen Hin- und Herpendeln führt, wird in diesem Fall in der praktischen Anwendung der Gradient $\frac{\partial E(\mathbf{w}^{(\tau)})}{\partial w_{ij}} \leftarrow 0$ gesetzt. Auf den Fall $\frac{\partial E(\mathbf{w}^{(\tau-1)})}{\partial w_{ij}} \cdot \frac{\partial E(\mathbf{w}^{(\tau)})}{\partial w_{ij}} < 0$ folgt also im nächsten Schritt immer der Fall $\frac{\partial E(\mathbf{w}^{(\tau-1)})}{\partial w_{ij}} \cdot \frac{\partial E(\mathbf{w}^{(\tau)})}{\partial w_{ij}} = 0$.¹⁾

Zwar müssen auch bei diesem Algorithmus zwei Lernraten und die Initialwerte $\Delta w_{ij}^{(0)}$ für die Gewichtsänderungen festgelegt werden.²⁾ Das scheint aber recht unproblematisch zu sein – kommen doch bemerkenswert viele Studien zu dem übereinstimmenden Ergebnis, dass der Resilient-Propaga-

¹⁾ Vgl. Riedmiller/Braun [RPROP] 587.

²⁾ Riedmiller/Braun [RPROP] 588 empfehlen $\eta^- = 0,5$, $\eta^+ = 1,2$ und $\Delta w_{ij}^{(0)} = 0,1$.

tion-Algorithmus stabiler ist und bessere Lösungen in kürzerer Rechenzeit findet als die meisten anderen Lernalgorithmen.¹⁾

Die Überlegenheit des Resilient-Propagation-Algorithmus ist zum Teil darauf zurückzuführen, dass er den Betrag des Gradienten ignoriert. Dadurch ändert sich im Grunde die Fehlerfunktion zu einer Manhattan-Metrik,²⁾ die, wie in Abschnitt f gezeigt wurde, robuster als die quadratische Fehlerfunktion ist. Folglich ist das Resilient-Propagation-Verfahren äquivalent zu einer Medianschätzung des Gewichtsvektors w (siehe S. 107 f.). Das ist auch der Grund dafür, dass *resilient propagation* nur im Batch-Verfahren zuverlässig funktioniert.³⁾

y) Berechnung der Newton-Richtung mit Gradientenverfahren zweiter Ordnung

Der theoretisch am besten gesicherte Weg, die Anzahl der notwendigen Schritte zum Optimum zu reduzieren, führt über die zweite Ableitung der Fehlerfunktion E . Denn mit der Hesse-Matrix $\mathbf{H}_E(w)$, die die partiellen Ableitungen zweiter Ordnung der Fehlerfunktion enthält, lässt sich die Newton-Richtung $-\mathbf{H}_E^{-1}(w) \nabla E(w)$ berechnen.⁴⁾ Wenn die Hesse-Matrix positiv definit ist, zeigt die Newton-Richtung an einem beliebigen Punkt w immer zum nächstgelegenen Minimum; während der negative Gradient erster Ordnung $-\nabla E(w)$ nur in die Richtung des steilsten Abstiegs zeigt, der in der Regel nicht direkt auf das Minimum zuläuft (siehe Abb. C-10). Die vielen Zwischenschritte, die ein Gradientenverfahren erster Ordnung auf

¹⁾ Vgl. z. B. Fun/Hagan [Levenberg-Marquardt] 471 ff.; Bajramovic/Gruber/Sick [Comparison] 838 ff.; Torrecilla u. a. [Optimising] 5828 f.; Kruse u. a. [Intelligence] 72; Lämmel/Cleve [Intelligenz] 219.

²⁾ Vgl. Crone [Neuronale Netze] 254; Kruse u. a. [Intelligence] 72.

³⁾ Vgl. Riedmiller/Braun [RPROP] 587; Kruse u. a. [Intelligence] 72.

⁴⁾ Vgl. auch im Weiteren Bishop [Neural Networks] 285 f.

seinem Zickzack-Weg zum Optimum macht, (in Abb. C-10 durch gestrichelte Pfeile dargestellt) fallen mit der Newton-Richtung weg. Die Iterationsregel des Gradientenabstiegsverfahrens erster Ordnung (C.11) (S. 95) kann als Näherungsverfahren zur Berechnung der Newton-Richtung gesehen werden, in dem die invertierte Hesse-Matrix durch die Lernrate η approximiert wird.

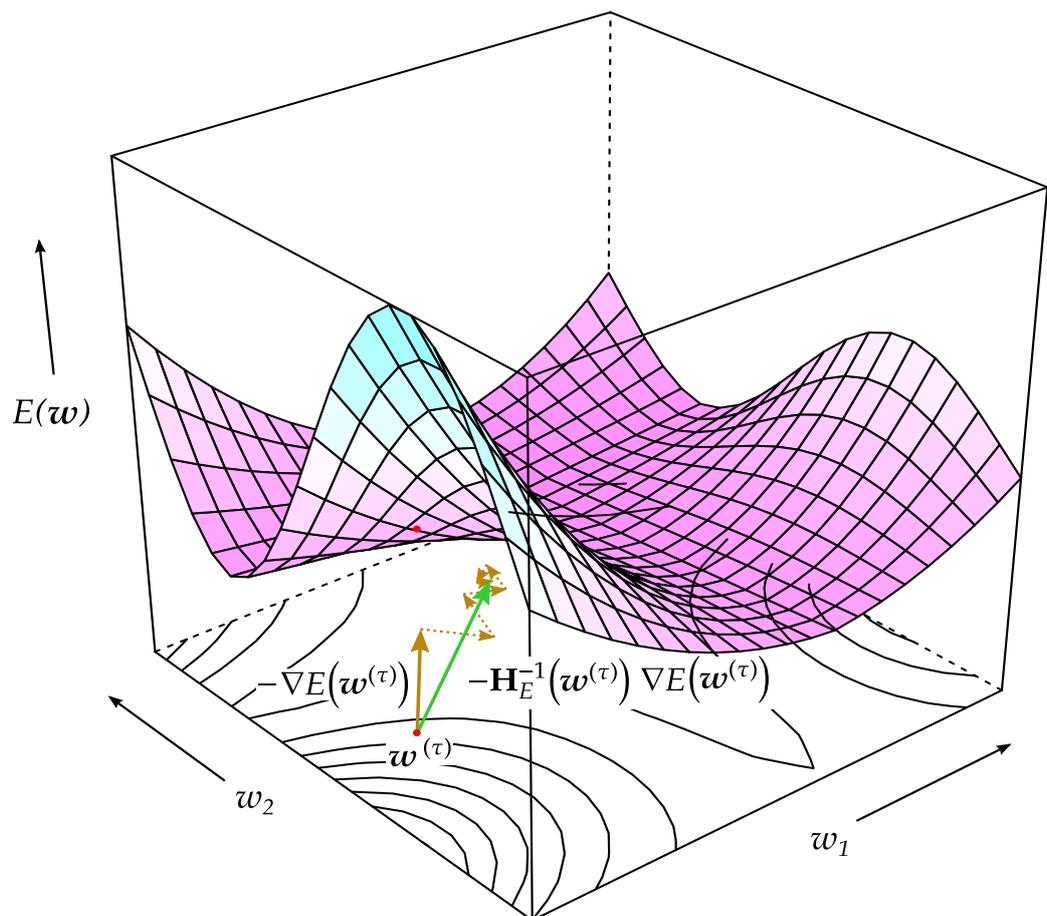


Abb. C-10: Newton-Richtung im Vergleich zum negativen Gradienten

Gleichwohl führt auch die Newton-Richtung nur annähernd zum Minimum, so dass der Gewichtsvektor in mehreren Schritten angepasst werden muss:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \mathbf{H}_E^{-1}(\mathbf{w}^{(\tau)}) \nabla E(\mathbf{w}^{(\tau)}) \quad (C.26)$$

¹⁾ Vgl. auch im Weiteren Bishop [Neural Networks] 285 ff.

Zwar sind bei diesem Vorgehen im Vergleich zu einem Verfahren erster Ordnung bedeutend weniger Iterationsschritte notwendig; dafür muss aber an jeder Stelle $w^{(\tau)}$ die Hesse-Matrix berechnet und invertiert werden. Da beide Operationen sehr rechenintensiv sind, wird die Inverse der Hesse-Matrix im praktischen Einsatz nur näherungsweise bestimmt. Dafür werden häufig der Levenberg-Marquardt-Algorithmus¹⁾ oder das Quasi-Newton-Verfahren mit Broyden-Fletcher-Goldfarb-Shanno-Algorithmus²⁾ eingesetzt. Diese Algorithmen finden bei komplexeren Regressions- und Klassifikationsproblemen deutlich bessere Lösungen und konvergieren schneller als die einfachen Backpropagation-Varianten;³⁾ in vielen Fällen sind sie auch besser als der Resilient-Propagation-Algorithmus.⁴⁾ Konjugierte Gradientenverfahren (*conjugate gradients*), z. B. nach der Methode von Fletcher und Reeves,⁵⁾ konvergieren in einigen Fällen noch etwas schneller und sind ähnlich exakt.⁶⁾ Sie sind demnach eine gute Wahl für sehr große, rechenintensive Probleme.

¹⁾ Vgl. Hagan/Menhaj [Marquardt]; Wilamowski u. a. [Convergence]; Bishop [Neural Networks] 290 ff.; Bishop [Machine Learning] 251.

²⁾ Vgl. Broyden [Double-Rank]; Fletcher [Variable Metric]; Goldfarb [Variable-Metric]; Shanno [Quasi-Newton]; Battiti [Second-Order Methods] 154 ff.; Bishop [Neural Networks] 287 ff.; Nabney [Netlab] 61 ff.

³⁾ Vgl. z. B. Webb/Lowe/Bedworth [Comparison] 18 ff.; Barnard [Optimization] 238 f.; Soria/Martin/Lisboa [Training Methods] 33 f.

⁴⁾ Vgl. z. B. Fun/Hagan [Levenberg-Marquardt] 471 ff.; Bajramovic/Gruber/Sick [Comparison] 838 ff.; Torrecilla u. a. [Optimising] 5828 f.

⁵⁾ Vgl. Fletcher/Reeves [Conjugate Gradients]; Battiti [Second-Order Methods] 146 ff.; Møller [Scaled Conjugate Gradient]; Bishop [Neural Networks] 274 ff.; Nabney [Netlab] 52 ff.

⁶⁾ Vgl. z. B. Webb/Lowe/Bedworth [Comparison] 18 ff.; Soria/Martin/Lisboa [Training Methods] 33 f.; Bajramovic/Gruber/Sick [Comparison] 838 ff.; Torrecilla u. a. [Optimising] 5828 f.

h) Verbindung von lokalen Lernverfahren mit Metaheuristiken zu hybriden Algorithmen

Die bisher betrachteten Lernverfahren sind lokale Suchverfahren, d. h. sie finden ein lokales Minimum der Fehlerfunktion in der Nähe der Startlösung $w^{(0)}$ (siehe Abb. C-7 a, S. 101). Die Suche kann ausgeweitet werden, indem ein lokales Suchverfahren mehrfach mit zufällig gewählten Initiallösungen gestartet wird. Auf global gute Lösungen stößt man auf diese Weise höchstens zufällig und erst nach einigen Durchläufen. Effizienter ist es, ein lokales Suchverfahren mit einer Metaheuristik zu verbinden. Dabei sorgt die Metaheuristik für eine planmäßige, globale Erkundung des Lösungsraums, indem sie beispielsweise systematisch Startlösungen erzeugt und damit das lokale Suchverfahren initialisiert, das von dort aus schnell ein lokales Optimum findet.¹⁾ Diese Verbindungen werden als *hybride Algorithmen* und, wenn sie evolutionäre Strategien als Metaheuristiken enthalten, als *memetische Algorithmen* bezeichnet.²⁾

Grundsätzlich kann jedes lokale Lernverfahren mit einer beliebigen Metaheuristik wie z. B. *tabu search*, *simulated annealing* oder genetischen Algorithmen kombiniert werden. Memetische Algorithmen zeigen in vergleichenden Experimenten besonders gute Resultate. Beim Trainieren von künstlichen neuronalen Netzen finden sie in relativ kurzer Rechenzeit im Durchschnitt bessere Lösungen als reine lokale Suchverfahren oder reine

¹⁾ Vgl. z. B. aktuell Chen/Tseng/Wu [Evolutionary Training]; Fu u. a. [Residence Error]; Matias u. a. [Optimized Extreme Learning Machine]; Xue u. a. [Genetic Ensemble]; Wang/Zeng/Chen [Differential Evolution].

²⁾ Vgl. Moscato [Memetic Algorithms]; Krasnogor [Memetic Algorithms]; Neri/Cotta [Memetic Algorithms]; Kovacs [Genetics-Based Machine Learning] 945, mit zahlreichen weiteren Nachweisen.

Metaheuristiken (ohne integrierte lokale Suche).¹⁾ Am besten schneidet darunter regelmäßig der auf Prinzipien der evolutionären Programmierung basierende EPNet-Algorithmus von Yao und Liu ab.²⁾

Hybride Algorithmen können neben dem Gewichtsvektor w beispielsweise auch die Netzarchitektur – Inputvariablen, verdeckte Neuronen und ihre Verbindungen – sowie die Lernrate η und andere Trainingsparameter optimieren (siehe Kapitel D.XII, S. 267 ff.). Der EPNet-Algorithmus etwa optimiert zugleich die Netzarchitektur und den Gewichtsvektor.

Gerade bei den populationsbasierten memetischen Algorithmen bietet es sich zudem an, am Ende nicht nur das neuronale Netz mit dem minimalen Trainingsfehler zu behalten, sondern eine ganze evolutionär optimierte Population von künstlichen neuronalen Netzen als *Ensemble* zu verwenden. Denn häufig kann die Generalisierungsleistung eines Schätzmodells – die Fähigkeit, neue Daten korrekt einzuordnen oder vorherzusagen – dadurch gesteigert werden, dass mehrere Modelle parametrisiert werden und für Auswertungen z. B. der Median der Modelloutputs berechnet wird. Das lässt sich dadurch erklären, dass jedes Modell zu einem gewissen Teil an zufällige Störgrößen, die nicht vom eigentlich zu schätzenden Modellzusammenhang verursacht werden, überangepasst ist;³⁾ falls aber die Störgrößen wirklich zufällig auftreten, müssten sich die Überanpassungen gegenseitig aufheben, wenn mehrere Modelle, die jeweils verschiedenes Rau-

¹⁾ Vgl. auch im Weiteren Madera/Dorransoro [Estimation of Distribution] 104 f.; Alba/Chicano [Genetic Algorithms] 123 ff.; Krasnogor/Aragón/Pacheco [Memetic Algorithms] 244 f.

²⁾ Vgl. Yao/Liu [Evolutionary System]; Yao [Evolving]; Yao/Liu [Population]; Liu/Yao [Ensemble Learning].

³⁾ Siehe Abschnitt 2, S. 88 ff.

schen enthalten, gemittelt werden.¹⁾ Hintergründe und Varianten dieses Prinzips werden in Kapitel D.XIII (S. 284 ff.) erörtert.

i) Schnelles Lernen mit *extreme learning machine*

Ein Lernverfahren, das erst in den letzten Jahren entwickelt und seither recht erfolgreich eingesetzt wurde,²⁾ ist die *extreme learning machine* (ELM). Sie unterscheidet sich deutlich von den klassischen, gradientenbasierten Verfahren und fußt auf dem Beweis, dass ein künstliches neuronales Netz mit einer verdeckten Schicht auch dann ein universeller Approximator³⁾ ist, wenn die Gewichte zwischen Eingabe- und verdeckter Schicht nicht optimiert, sondern zufällig festgesetzt werden.⁴⁾ Der ELM-Algorithmus macht sich dies zunutze. Wenn die Eingangsgewichte der verdeckten Schicht auf zufällige, konstante Werte festgelegt werden, stehen damit auch die Ausgabewerte der verdeckten Neuronen fest. Da die Ausgangssignale der verdeckten Neuronen linear mit der Ausgabeschicht verbunden sind, reduziert sich das neuronale Netz auf ein lineares Gleichungssystem. Die Gewichte zwischen verdeckter Schicht und Ausgabeschicht können durch Lösen des Gleichungssystems bestimmt werden. Dadurch ist der ELM-Algorithmus bedeutend schneller als andere Lernverfahren und hat zum Teil eine bessere Generalisierungsfähigkeit. Zudem müssen keine Lernraten oder andere Trainingsparameter gesetzt werden.

¹⁾ Siehe Abb. D-54, S. 285. Vgl. Bishop [Machine Learning] 655 f.

²⁾ Vgl. z. B. Sun u. a. [Sales Forecasting]; Bueno-Crespo/García-Laencina/Sancho-Gómez [Architecture Design]; Zhong u. a. [Effectiveness]; Chorowski/Wang/Zurada [Review].

³⁾ Siehe Abschnitt 2, S. 88 ff.

⁴⁾ Vgl. Igelnik/Pao [Stochastic Choice]; Huang/Chen/Siew [Approximation]; vgl. auch Wang/Wan [Comments]; Huang [Reply]; Zhang u. a. [Approximation].

Mit dem künstlichen neuronalen Netz aus Formel (C.6) (S. 88) ergibt sich folgender ELM-Algorithmus.¹⁾ Zunächst werden alle Gewichte w_{ij} ($i = 0, \dots, I; j = 1, \dots, H$) auf einen zufälligen Wert festgelegt. Mit den Gewichten w_{ij} sind bei gegebenen Trainingspunkten $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$ auch die Ausgabesignale $o_j^{(n)}$ der verdeckten Neuronen determiniert. Für die späteren Berechnungen ist es sinnvoll, die Ausgangssignale aller H verdeckten Neuronen für alle N Trainingspunkte in einer $N \times H$ -Matrix festzuhalten:

$$\mathbf{O}_H = \begin{bmatrix} o_1^{(1)} & \cdots & o_H^{(1)} \\ \vdots & \ddots & \vdots \\ o_1^{(N)} & \cdots & o_H^{(N)} \end{bmatrix} \quad (\text{C.27})$$

mit
$$o_j^{(n)} = f_{\text{act}}(\text{net}_j^{(n)}) = \frac{1}{1 + e^{-(\text{net}_j^{(n)})}},$$

$$\text{net}_j^{(n)} = \sum_{i=0}^I w_{ij} x_i^{(n)}.$$

Die verbleibenden Gewichte w_{jk} ($j = 1, \dots, H; k = 1, \dots, O$) sind die freien Parameter der *extreme learning machine*, die es zu optimieren gilt. Sie lassen sich in einer $H \times O$ -Matrix

$$\mathbf{W}_\beta = \begin{bmatrix} w_{11} & \cdots & w_{1O} \\ \vdots & \ddots & \vdots \\ w_{H1} & \cdots & w_{HO} \end{bmatrix} \quad (\text{C.28})$$

zusammenfassen. Da bei Regressionsmodellen in den Outputneuronen nur die eingehenden, gewichteten Signale aufsummiert werden – die Aktivierungsfunktion ist die Identitätsfunktion – gilt in diesem Fall

$$\mathbf{O}_H \mathbf{W}_\beta = \hat{\mathbf{Y}} \quad (\text{C.29})$$

mit $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times O}$ (siehe S. 86).

¹⁾ Die folgenden Ausführungen entsprechen dem ELM-Algorithmus, wie ihn Huang/Zhu/Siew [Extreme Learning Machine] beschreiben.

Die *extreme learning machine* soll den euklidischen Abstand¹⁾ zwischen den geschätzten und den beobachteten Werten minimieren:

$$\min \|\hat{\mathbf{Y}} - \mathbf{Y}\|_2 = \min \|\mathbf{O}_H \mathbf{W}_\beta - \mathbf{Y}\|_2 \quad (\text{C.30})$$

mit
$$\mathbf{Y} = \begin{bmatrix} y_1^{(1)} & \cdots & y_O^{(1)} \\ \vdots & \ddots & \vdots \\ y_1^{(N)} & \cdots & y_O^{(N)} \end{bmatrix}.$$

Eine allgemeine Lösung für (C.30) mit einer beliebigen reellen Matrix \mathbf{O}_H , die nicht vollen Rang haben muss, ist die sogenannte *Pseudonormallösung*

$$\mathbf{W}_\beta^* = \mathbf{O}_H^\dagger \mathbf{Y} \quad (\text{C.31})$$

mit der Pseudoinversen (Moore-Penrose-Inversen) \mathbf{O}_H^\dagger .²⁾ Die Pseudoinverse ist so definiert, dass $\mathbf{W}_\beta^* = \mathbf{O}_H^\dagger \mathbf{Y}$ unter allen Lösung für (C.30) diejenige mit der kleinsten euklidischen Norm ist:

$$\|\mathbf{W}_\beta^*\|_2 = \|\mathbf{O}_H^\dagger \mathbf{Y}\|_2 \leq \|\mathbf{W}_\beta\|_2, \quad (\text{C.32})$$

$$\text{für alle } \mathbf{W}_\beta \in \left\{ \mathbf{W}_\beta \mid \forall \mathbf{Z} \in \mathbb{R}^{H \times O}: \|\mathbf{O}_H \mathbf{W}_\beta - \mathbf{Y}\|_2 \leq \|\mathbf{O}_H \mathbf{Z} - \mathbf{Y}\|_2 \right\}.$$
³⁾

Diese Eigenschaft trägt zur guten Generalisierungsleistung der *extreme learning machine* bei, da kleinere Gewichte tendenziell weniger große Ausschläge der Schätzfunktion nach oben und unten wie z. B. in Abb. C-5 c (S. 90) verursachen.

Die Pseudoinverse kann mit verschiedenen numerischen Verfahren berechnet werden.⁴⁾ Darunter gilt die Singulärwertzerlegung (*singular value decom-*

¹⁾ Siehe Abschnitt f, S. 106 ff., zur Diskussion verschiedener Abstandsnormen.

²⁾ Vgl. Hämmerlin/Hoffmann [Mathematik] 90.

³⁾ Vgl. Deuffhard/Hohmann [Mathematik 1] 85; Huang/Zhu/Siew [Extreme Learning Machine] 492.

⁴⁾ Vgl. Shinozaki/Sibuya/Tanabe [Moore-Penrose direct]; Shinozaki/Sibuya/Tanabe [Moore-Penrose iterative]; Deuffhard/Hohmann [Mathematik 1] 87 f.

position) als besonders stabil, insbesondere auch bei rangdefizienten Matrizen \mathbf{O}_H , die nicht auszuschließen sind.¹⁾ Die numerische Berechnung über die Singulärwertzerlegung hat zudem einen recht geringen relativen Fehler.²⁾

Für jede reelle Matrix $\mathbf{O}_H \in \mathbb{R}^{N \times H}$ mit $N > H$ existiert eine Singulärwertzerlegung

$$\mathbf{O}_H = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (\text{C.33})$$

mit den orthogonalen Matrizen $\mathbf{U} \in \mathbb{R}^{N \times N}$ und $\mathbf{V} \in \mathbb{R}^{H \times H}$ sowie der Matrix

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_H \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \quad (\text{C.34})$$

mit Singulärwerten σ .³⁾ Daraus lässt sich die Pseudoinverse

$$\mathbf{O}_H^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T \quad (\text{C.35})$$

mit $\mathbf{\Sigma}^+ = \begin{bmatrix} 1/\sigma_1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1/\sigma_H & \cdots & 0 \end{bmatrix}$

berechnen.⁴⁾

Zusammenfassend zeigt Algorithmus C-2 die Berechnung einer *extreme learning machine* in Pseudocode. Hier wird deutlich, dass die wachsende

¹⁾ Vgl. Schabach/Werner [Mathematik] 287 f.; Huang/Zhu/Siew [Extreme Learning Machine] 493.

²⁾ Vgl. Shinozaki/Sibuya/Tanabe [Moore-Penrose direct] 202; Hämmerlin/Hoffmann [Mathematik] 97. Bei Shinozaki/Sibuya/Tanabe ist der relative Fehler mit dem Gaußschen Eliminationsverfahren und dem Gram-Schmidtschen Orthogonalisierungsverfahren noch etwas kleiner; Hämmerlin/Hoffmann [Mathematik] 85 kommen zum umgekehrten Ergebnis.

³⁾ Vgl. z. B. Hämmerlin/Hoffmann [Mathematik] 85; Schwarz/Köckler [Mathematik] 293.

⁴⁾ Vgl. Shinozaki/Sibuya/Tanabe [Moore-Penrose direct] 198; Schabach/Werner [Mathematik] 73; Hämmerlin/Hoffmann [Mathematik] 91 f.

Popularität der *extreme learning machine* wohl auch auf ihre einfache Implementierbarkeit zurückzuführen ist.

Algorithmus *Extreme learning machine*

1: Wähle zufällige Werte für w_{ij} ($i = 0, \dots, I; j = 1, \dots, H$).

2: Berechne die Ausgabematrix der verdeckten Schicht \mathbf{O}_H . (C.27)

3: Berechne die Pseudoinverse \mathbf{O}_H^\dagger . (C.35)

4: Berechne die Gewichtsmatrix $\mathbf{W}_\beta^* = \mathbf{O}_H^\dagger \mathbf{Y}$. (C.31)

*Algorithmus C-2: Extreme learning machine*¹⁾

Trotz ihrer bemerkenswerten Vorteile macht die *extreme learning machine* die klassischen Lernverfahren mit Fehlerrückpropagation nicht obsolet. Denn (C.31) ist zwar eine global optimale Lösung für das Optimierungsproblem (C.30), aber nicht für das eigentlich verfolgte Ziel, den Fehler $E(\boldsymbol{w})$ über den gesamten Gewichtsvektor zu minimieren (siehe (C.8), S. 94). Mit anderen Worten: Die Güte einer mit der *extreme learning machine* erzeugten Lösung hängt stark davon ab, wie glücklich der erste Teil des Gewichtsvektors \boldsymbol{w} – die Gewichte zwischen Eingabe- und verdeckter Schicht – gewählt wurde. Auch die *extreme learning machine* sollte deshalb mehrmals mit verschiedenen Initialgewichten gestartet werden, um insgesamt eine gute Lösung zu erhalten. Der Beweis, dass die *extreme learning machine* ein universeller Approximator ist (s. o.), ändert an dieser Tatsache übrigens nichts. Denn das bedeutet nur, dass die *extreme learning machine* bei einer *hinreichend* großen Zahl von verdeckten Neuronen eine stetige Funktion beliebig genau approximieren kann. Praktisch werden aber sehr viel weniger verdeckte Neuronen verwendet.²⁾

¹⁾ Vgl. Huang/Zhu/Siew [Extreme Learning Machine] 492.

²⁾ Siehe Abschnitt 2, S. 88 ff.

II. Anwendungsbereiche künstlicher neuronaler Netze

1. Historische Entwicklung der Anwendungsbereiche

Ursprünglich wurden künstliche neuronale Netze entwickelt, um geistige Vorgänge besser zu verstehen.¹⁾ Insbesondere lassen sich eher intuitive Vorgänge wie Wahrnehmung, Beherrschung der Motorik, Analogiebildung und Kategorisierung leichter auf der Ebene neuronaler Netze erklären und auch künstlich erzeugen als auf der Ebene des lange Zeit vorherrschenden Modells der Turing-Maschine.²⁾ Eine Turing-Maschine folgt einem algorithmischen Schema, nach dem sie eine Symbolfolge sequenziell bearbeitet.³⁾ Dem klassischen Paradigma der Kognitionswissenschaft zufolge kann der Geist als eine symbolmanipulierende Turing-Maschine beschrieben werden.⁴⁾ Künstliche neuronale Netze sind das Gegenstück zur Turing-Maschine im konkurrierenden, *konnektionistischen* Paradigma. Während sich bei der Turing-Maschine die symbolische Struktur des Wissens in physikalisch gespeicherten Symbolfolgen widerspiegelt, liegt das Wissen eines neuronalen Netzes nicht in den Recheneinheiten (den Neuronen) sondern in den Verbindungen zwischen ihnen. Eine neuronale Einheit allein ist kein bedeutungstragendes Symbol (sie *repräsentiert* nichts), aber mehrere Einhei-

¹⁾ Vgl. z. B. die frühen Veröffentlichungen von McCulloch/Pitts [Calculus] und Rosenblatt [Perceptron].

²⁾ Vgl. z. B. die Beiträge in McClelland/Rumelhart/PDP Research Group [Processing 2].

³⁾ Vgl. Turing [Machinery].

⁴⁾ Vgl. Putnam [Minds]; Newell/Simon [Symbols] 119 ff.; Fodor/Pylyshyn [Cognitive Architecture] 4, mit weiteren Nachweisen.

ten können zusammen ein Symbol, d. h. eine geistige Repräsentation, bilden – das als Aktivitätsvektor von Neuronen dargestellt werden kann.¹⁾ Eine geistige Repräsentation ist demnach über viele neuronale Einheiten verteilt und eine neuronale Einheit ist an vielen Repräsentationen beteiligt.

Die zentrale Hypothese des Konnektionismus ist, dass manche geistigen Vorgänge – unter anderem Kategorisierung, Assoziation oder motorische Kontrolle – nur auf der *subsymbolischen* Ebene, d. h. mit einem neuronalen Modell adäquat beschrieben werden können.²⁾ Während das klassische Paradigma besagt, dass eine vollständige Beschreibung kognitiver Prozesse auf der Ebene bedeutungstragender Elemente (der Symbole und Ausdrücke) möglich ist,³⁾ behaupten die Konnektionisten, dass es für eine vollständige Beschreibung kognitiver Prozesse notwendig, aber auch hinreichend ist, die formalen Prinzipien zu betrachten, die zwischen den nicht-bedeutungstragenden Elementen der subsymbolischen Ebene gelten.⁴⁾

Unabhängig von dieser Diskussion haben sich künstliche neuronale Netze nach der Publikation des Backpropagation-Algorithmus im Jahr 1986 rasch in angrenzenden Bereichen der Künstliche-Intelligenz-Forschung, vor allem in der Informatik, als Methode des maschinellen Lernens etabliert. Sie sind heute eines der am häufigsten eingesetzten Verfahren für Spracherken-

¹⁾ Vgl. auch im Weiteren Hinton/McClelland/Rumelhart [Representations]; Smolensky [Connectionism] 5 f.

²⁾ Vgl. Smolensky [Connectionism] 6.

³⁾ Vgl. Newel/Simon [Symbols] 116; Fodor/Pylyshyn [Cognitive Architecture] 12 f.

⁴⁾ Vgl. Smolensky [Connectionism] 3 f.; Chalmers/French/Hofstadter [Perception]; Chalmers [Computation] 46; Chalmers [Connectionism] 318. Einflussreiche Kritik am Konnektionismus stammt u. a. von Broadbent [Levels]; Fodor/Pylyshyn [Cognitive Architecture]; Pinker/Prince [Language]; Fodor/McLaughlin [Systematicity]; Ramsey [Representations]; Marcus [Rethinking]; Fodor/Lepore [Semantic Space]; Marcus [Mind]. Einen umfangreichen Überblick der Diskussion bietet Garson [Connectionism].

nung, Handschriftenerkennung, Gesichtserkennung und andere Bilderkennungsaufgaben.

Seit in den Neunzigerjahren die eher anwendungsgetriebene Erforschung künstlicher neuronaler Netze auf solide statistische Grundlagen gestellt wurde,¹⁾ sind künstliche neuronale Netze ein fester Bestandteil des Statistik-Instrumentariums und werden für vielfältige Zwecke eingesetzt. Bei der Einordnung in das Methodenspektrum der Statistik wird oft die enge Verwandtschaft einzelner Netzarchitekturen zu klassischen Verfahren der Regressions-, Klassifikations- oder Clusteranalyse hervorgehoben.²⁾

Die Anwendungsbereiche künstlicher neuronaler Netze sind heute kaum noch zu überblicken. Zumal regelmäßig neue, schnell wachsende Anwendungsfelder hinzukommen, wie z. B. in jüngerer Zeit die Analyse von Transkriptom- und Proteomdaten in der Bioinformatik³⁾ sowie die Auswertung von sozialen Netzwerken, Blogs, Kurznachrichtendiensten und anderen *social media* für wissenschaftliche und kommerzielle Zwecke.⁴⁾ Beiden Bereichen ist gemeinsam, dass dort sehr große, vieldimensionale Datenmengen anfallen (*big data*), die mit künstlichen neuronalen Netzen vergleichsweise effizient ausgewertet werden können.⁵⁾

¹⁾ Vgl. vor allem die Standardwerke von Bishop [Neural Networks] und Ripley [Pattern Recognition]. Vgl. auch Geman/Bienenstock/Doursat [Dilemma]; Cheng/Titterington [Statistical Perspective]; Ripley [Classification].

²⁾ Vgl. z. B. Cheng/Titterington [Statistical Perspective] 17 ff.; siehe auch Abschnitt I.3.a, S. 92 ff.

³⁾ Vgl. z. B. das Sonderheft zum Thema „Pattern Recognition in Bioinformatics“. In: *Neurocomputing* (145) 2014.

⁴⁾ Vgl. z. B. das Sonderheft zum Thema „Affective Neural Networks and Cognitive Learning Systems for Big Data Analysis“. In: *Neural Networks* (58) 2014.

⁵⁾ Vgl. Chen/Chiang/Storey [Big Data] 1173 ff.

2. Betriebswirtschaftliche Anwendungen

Auch die betriebswirtschaftlichen Einsatzmöglichkeiten für künstliche neuronale Netze sind sehr vielseitig. Abb. C–11 zeigt eine Auswahl typischer, branchenübergreifender Anwendungen, nach Problemtypen gruppiert.¹⁾ Netztypen, die für die jeweilige Problemklasse häufig verwendet werden, sind in der rechten Spalte aufgeführt. Der für diese Arbeit besonders relevante Bereich der Kostenschätzung ist in Abb. C–12 weiter aufgegliedert. Veröffentlichungen auf diesem Gebiet befassen sich überwiegend mit Problemen der konstruktionsbegleitenden Kalkulation oder der Kostenschätzung von Großprojekten. Dabei wird in der Regel das aus Teilkapitel I bekannte *multilayer perceptron* verwendet und mit anderen statistischen Methoden verglichen. Bei allen Vergleichen mit klassischen polynomialen Regressionsverfahren hat das *multilayer perceptron* einen geringeren Schätzfehler.²⁾ Oft ist es auch genauer als aufwendigere Verfahren wie z. B. *support vector machines*,³⁾ Radiale-Basisfunktionen-Netze⁴⁾ oder rückgekoppelte Netze⁵⁾. Die große Menge an positiven Befunden spricht dafür, dass das *multilayer perceptron* für die Kostenschätzung gut geeignet ist und demnach in dieser Arbeit keine alternativen Netztypen betrachtet werden müssen.

¹⁾ Für umfangreichere Literaturübersichten vgl. Alex [Neuronale Netze] 175 ff.; Crone [Neuronale Netze] 212 ff.

²⁾ Vgl. Zhang/Fuh/Chan [Packaging Products] 111; Smith/Mason [Cost Estimation] 156; Shtub/Versano [Steel Pipe Bending] 205 ff.; Bode [Cost Estimation] 1247; Seo u. a. [Life Cycle Cost] 551; Wang/Stockton/Baguley [Cost Modelling] 3818 f.; Kim/An/Kang [Construction Cost] 1231 ff.; Cavalieri/Maccarrone/Pinto [Automotive] 174 f.; Caputo/Pelagagge [Process Vessels] 949 ff.; Verlinden u. a. [Sheet Metal Parts] 491; Liu [Cost Estimation] 216; Weckman u. a. [Manufacturing Cost] 266; Kim u. a. [Building Construction] 5; Du u. a. [Software Cost] 1511 f.; Khan/Qureshi [Software Effort] 1991 f.

³⁾ Vgl. Verlinden u. a. [Sheet Metal Parts] 491; Kim u. a. [Building Construction] 5.

⁴⁾ Vgl. Duran/Maciel/Rodriguez [Piping Elements] 7794. Dagegen zitieren Khan/Qureshi [Software Effort] 1991 f. einige Studien, in denen Radiale-Basisfunktionen-Netze genauer sind als ein *multilayer perceptron*.

⁵⁾ Vgl. Praynlin/Latha [Software Effort] 106.

Betriebswirtschaftliche Anwendungsbereiche künstlicher neuronaler Netze

 Regressionsanalyse

Zeitreihenanalyse²⁾

- Bedarfs- und Absatzprognose, Werbeerfolgsprognose
- Aktienkurs-, Wechselkurs-, Preis- und Zinsprognose

multivariate Regression

- Umsatzprognose³⁾
 - Schätzung von Marktreaktionsfunktionen (*market response models*)⁴⁾
 - Kostenschätzung⁵⁾
-

 Kategorisierung

Klassifikation (vorgegebene Klassen)⁶⁾

- Bonitäts- und Kreditwürdigkeitsprüfung, Rating⁷⁾
 - Insolvenzprognose von Unternehmen und Banken⁸⁾
 - Betrugserkennung bei Kreditkarten, Versicherungen, Jahresabschlüssen etc.
 - Kundensegmentierung, u. a. für Zielkundenmarketing (*customized marketing*) und Abwanderungsprävention⁹⁾
 - Qualitätsprüfung (z. B. Auswertung opt. Sensoren)¹⁰⁾
-

Clusteranalyse (nicht vorgegebene Klassen)

- Clustering von Versicherungsnehmern, Debitoren, Kreditnehmern etc. auf Basis von Risikomerkmale¹¹⁾
 - Kundensegmentierung (s. o.)
-

 Optimierung

Lösung von kombinatorischen Optimierungsproblemen (u. a. Tourenplanung, Reihenfolgeplanung)¹²⁾

Lösung von Differenzialgleichungen¹³⁾

Lösung von Markov-Entscheidungsproblemen (z. B. Auftragsannahme unter Unsicherheit)¹⁴⁾

überwachtes Lernen¹⁾ mit Feedforward-Netzen (vor allem *multilayer perceptrons* und Radiale-Basis-funktionen-Netzen) oder einfachen rückgekoppelten Netzen (*simple recurrent networks*) wie Jordan- und Elman-Netzen

unüberwachtes Lernen mit selbstorganisierenden Karten (Kohonen-Netzen) oder Assoziativspeicher-netzen wie Hopfield-Netzen, Boltzmann-Ma-schinen und Adaptive-Resonanz-Theorie-Netzen (ART-Netzen)

bestärkendes Lernen, u. a. mit Feedforward-Netzen

Abb. C-11: Übersicht betriebswirtschaftlicher Anwendungen künstlicher neuronaler Netze

-
- 1) Zur Unterscheidung zwischen überwachtem, unüberwachtem und bestärkendem Lernen siehe Abschnitt I.3.a, S. 92 ff. Für Erläuterungen zu den verschiedenen Netztypen vgl. Kruse u. a. [Intelligence]; Lämmel/Cleve [Intelligenz].
 - 2) Vgl. Rehkugler/Zimmermann [Ökonomie]; Lackes/Mack [Neuronale Netze] 210 ff.; Paliwal/Kumar [Applications] 5 ff.; Janetzke/Lewandowski [Prognose] 358 ff.; Brabazon u. a. [Finance] 1712 ff.; Zhang [Time-Series] 466; jeweils mit weiteren Nachweisen. Vgl. aktuell z. B. Venkatesh u. a. [Cash Demand]; Marzouk/Amin [Materials Prices].
 - 3) Vgl. Düsing [Anwendungsbereiche] 166 ff.; Janetzke/Lewandowski [Prognose] 358 ff., mit weiteren Nachweisen.
 - 4) Vgl. Paliwal/Kumar [Applications] 8 f., mit weiteren Nachweisen.
 - 5) Siehe unten, Abb. C-12.
 - 6) Vgl. Coakley/Brown [Modeling Issues]; Paliwal/Kumar [Applications] 3 ff.; Brabazon u. a. [Finance] 1724 f.; Janetzke/Lewandowski [Prognose] 358 ff.; jeweils mit weiteren Nachweisen.
 - 7) Vgl. auch Rehkugler/Zimmermann [Ökonomie]; Blanco u. a. [Credit Scoring].
 - 8) Vgl. auch Anders/Szczesny [Insolvenz]; Leker/Schewe [Kreditausfall]. Zu einer fundamentalen Kritik der bilanzanalytischen Insolvenzprognose vgl. Schneider [Warnung].
 - 9) Vgl. auch Säuberlich [Web Mining]; Schneider/Nelke/Poloni [Kundenbindung]; Wiedmann/Jung [Segmentierungsanalyse]; Wiedmann [Zielkundenmanagement].
 - 10) Vgl. Corsten/May [Neuronale Netze] 103 ff.
 - 11) Vgl. Lackes/Mack [Neuronale Netze] 230 ff.
 - 12) Vgl. Simon/Takefuji [Job-Shop]; Corsten/May [Unterstützungspotential], mit ausführlichem Literaturüberblick; Düsing [Anwendungsbereiche] 187 ff.; Jain/Meeran [Job-Shop]; Kruse u. a. [Intelligence] 131 ff.; Lämmel/Cleve [Intelligenz] 260 ff.; Kechadi/Low/Goncalves [Job Shop].
 - 13) Vgl. Kruse u. a. [Intelligence] 144 ff.
 - 14) Vgl. Hing/van Harten/Schuur [Order Acceptance].

Modelle zur Kostenschätzung mit künstlichen neuronalen Netzen

abhängige Variable	Einflussgrößen
Produktstückkosten (konstruktionsbegleitende Kalkulation) ¹⁾	geometrische Gestalt, Gewicht, Werkstoffeigenschaften, Oberflächenbeschaffenheit, Art und Anzahl der Verarbeitungsschritte, Anzahl Schrauben, Schweißnähte, Löcher usw.
Softwareentwicklungsaufwand ²⁾	<i>cost driver</i> des COCOMO-II-Modells ³⁾ zur Aufwandsschätzung von Softwareprojekten
Kosten eines Bauvorhabens ⁴⁾	Gebäudefläche, Grundstücksfläche, Anzahl der Stockwerke, Anzahl der Räume, Keller (ja/nein), Stockwerkshöhe
Prozesskosten im Schiffbau ⁵⁾	geometrische Gestalt, Leistung, Typ

Abb. C–12: Literaturüberblick zur Kostenschätzung mit künstlichen neuronalen Netzen

Den in Abb. C–12 angeführten Modellen ist gemein, dass sie ausschließlich eine möglichst exakte Punktprognose für ein sehr spezifisches Schätzproblem anstreben. Zwar gibt es vereinzelt Ansätze einer Prozesskostenrechnung,⁶⁾ die geschätzten Prozesskosten werden aber nur als Kalkulationshilfe für ein bestimmtes Produkt verwendet. Eine darüber hinausgehende

¹⁾ Vgl. Becker/Prischmann [Kalkulation]; Zhang/Fuh/Chan [Packaging Products]; Smith/Mason [Cost Estimation]; Shtub/Versano [Steel Pipe Bending]; Bode [Cost Estimation]; Seo u. a. [Life Cycle Cost]; Wang/Stockton/Baguley [Cost Modelling]; Cavalieri/Maccarone/Pinto [Automotive]; Ehrlenspiel/Kiewert/Lindemann [Entwickeln] 468 ff.; Caputo/Pelagagge [Process Vessels]; Verlinden u. a. [Sheet Metal Parts]; Liu [Cost Estimation]; Weckman u. a. [Manufacturing Cost]; Duran/Maciel/Rodriguez [Piping Elements]; Wang/Wang/Wang [Injection Molding].

²⁾ Vgl. Praynlin/Latha [Software Effort]; Du u. a. [Software Cost]; Khan/Qureshi [Software Effort]; jeweils mit weiteren Nachweisen.

³⁾ Vgl. Boehm u. a. [COCOMO II].

⁴⁾ Vgl. Kim/An/Kang [Construction Cost]; Kim u. a. [Building Construction].

⁵⁾ Vgl. Urkmez u. a. [Ship Building].

⁶⁾ Vgl. Kim/Han [Activity-Based Costing]; Urkmez u. a. [Ship Building].

Einbettung neuronaler Netze in die Kostenrechnung findet nicht statt. Demhingegen hat diese Dissertation zum Ziel, künstliche neuronale Netze so in die Kostenrechnung zu integrieren, dass sie die traditionellen, linearen Kostenfunktionen bei entscheidungsorientierten Kostenrechnungsaufgaben ersetzen können. Zu diesen Aufgaben gehören neben der Kostenprognose vor allem Alternativenbewertung, Abweichungsanalyse sowie die transparente Darstellung der Kostenstruktur zur Problemerkennung (siehe Kapitel B.IV, S. 55 ff.). Eine solche umfassende Integration künstlicher neuronaler Netze in die Kostenrechnung wurde bislang noch nicht vorgenommen. Ob und unter welchen Umständen sie sinnvoll ist, wird im folgenden Teilkapitel III diskutiert.

III. Zweckmäßigkeit der Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen

1. Analytische Bestimmung von Kostenfunktionen als Mittel der Wahl

In diesem Teilkapitel wird erörtert, wann es zweckmäßig ist, künstliche neuronale Netze zur Schätzung von Kostenfunktionen in der Kostenrechnung einzusetzen. In der Kostenrechnung werden grundsätzlich zwei Arten der Ermittlung von Kostenfunktionen unterschieden: *Statistische* Verfahren leiten eine Kostenfunktion aus den Istkosten der Vergangenheit ab, *analytische* Verfahren aus einer „sorgfältigen technisch-kostenwirtschaft-

lichen Einflussgrößenanalyse.“¹⁾ Künstliche neuronale Netze gehören zu den statistischen Verfahren.

Analytische Verfahren eignen sich besonders zur Berechnung der Material-einzelkosten. Mit Materialstücklisten lassen sie sich detailliert und ohne Rückgriff auf Vergangenheitsdaten kalkulieren.²⁾ Aber auch die Fertigungskosten können analytisch bestimmt werden. Eine rein deduktive Ableitung aus naturwissenschaftlich-technischen Gesetzen scheitert jedoch an der Komplexität der Produktionsvorgänge.³⁾ Deshalb versucht man, der empirischen Kostensituation durch *isolierte Abstraktion* näherzukommen, d. h. indem man den Produktionsprozess in einzelne, isolierte Schritte zerlegt.⁴⁾ Dabei werden betriebliche Teileinheiten bzw. Tätigkeitsbereiche als voneinander unabhängige Kostenstellen behandelt. Aus technisch und produktionstheoretisch fundierten Annahmen – ggf. ergänzt um experimentelle Messungen (Probeläufe, Musterfertigungen) – wird deduziert, welche Wirkung die Änderung einer Kosteneinflussgröße auf die Kostenhöhe in der Kostenstelle hat, wenn die anderen Einflussgrößen konstant bleiben. Für jede Kostenstelle wird also zunächst eine Reihe von Gleichungen

$$\begin{aligned} K &= f(x_1, \bar{x}_2, \dots, \bar{x}_I) \\ K &= f(\bar{x}_1, x_2, \dots, \bar{x}_I) \\ &\dots \\ K &= f(\bar{x}_1, \bar{x}_2, \dots, x_I) \end{aligned} \tag{C.36}$$

¹⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 258; genauso Ewert/Wagenhofer [Unternehmensrechnung] 659 f.; Friedl/Hofmann/Pedell [Kostenrechnung] 214 ff.

²⁾ Vgl. Coenenberg/Fischer/Günther [Kostenrechnung] 529 ff.; Friedl/Hofmann/Pedell [Kostenrechnung] 215.

³⁾ Vgl. Pressmar [Kosten- und Leistungsanalyse] 185 f.

⁴⁾ Vgl. auch im Weiteren Heinen [Kostenlehre] 175 f.; Caputo/Pelagagge [Process Vessels] 936 f., mit weiteren Nachweisen.

bestimmt. Aggregiert ergeben sie die Gesamtkostenfunktion der Kostenstelle $K = f(x_1, \dots, x_I)$. Dieses Vorgehen wird als *planmäßige Kostenauflösung* oder *industrial engineering method* bezeichnet.¹⁾ Bei manuellen Tätigkeiten wird sie um Arbeitszeitstudien und Funktionsanalysen (*work-measurement methods*) ergänzt.²⁾

Mit dieser Art der Kostenschätzung lässt sich detailliert und zukunftsorientiert bestimmen, welche Kosten eine spezifische Entscheidung verursachen wird.³⁾ Außerdem können Optimalkostenfunktionen berechnet und daraus Sollkostenfunktionen zur verhaltensorientierten Steuerung abgeleitet werden.⁴⁾ Die genannten analytischen Verfahren sind demnach das Mittel der Wahl, um die Zwecke der Kostenrechnung⁵⁾ umfassend zu erfüllen. Andere analytische Methoden wie die Schätzung durch Expertengremien (*conference method*) oder die Spaltung der Kostenarten in fixe und variable Bestandteile (buchtechnische Kostenauflösung, *account analysis method*)⁶⁾ sind zwar weniger aufwendig, ihre Zuverlässigkeit und Präzision hängen aber stark vom Wissen und den Fähigkeiten der Experten bzw. des Kostenplaners ab.⁷⁾

¹⁾ Vgl. auch im Weiteren Kilger/Pampel/Vikas [Plankostenrechnung] 289 ff.; Horngren/Datar/Rajan [Cost Accounting] 368.

²⁾ Vgl. auch Friedl/Hofmann/Pedell [Kostenrechnung] 215.

³⁾ Vgl. Troßmann [Plankostenrechnung] 239.

⁴⁾ Siehe Abschnitt B.IV.6, S. 64 ff.

⁵⁾ Siehe Kapitel B.II, S. 11 ff.

⁶⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 291. Siehe auch Abschnitt B.III.4.a, S. 31 ff.

⁷⁾ Vgl. Horngren/Datar/Rajan [Cost Accounting] 368 f.; Schweitzer/Küpper [Systeme] 290; Friedl/Hofmann/Pedell [Kostenrechnung] 216.

2. Statistische Schätzung von Kostenfunktionen als praktische Vereinfachung

Die Zwecke der Kostenrechnung werden am besten mit einer Kostenschätzung durch exakte analytische Verfahren – Detailkalkulation oder planmäßige Kostenauflösung – erfüllt. Diese Methoden sind allerdings recht zeitaufwendig und schwer anwendbar, wenn keine technischen Input-Output-Beziehungen vorliegen.

Statistische Verfahren versuchen, aus Vergangenheitsdaten auf einen allgemeinen Zusammenhang zwischen Einflussgrößen und Kostenhöhe zu schließen, ohne die Einzelheiten der Produktionsabläufe zu berücksichtigen oder die Kostenstruktur im Detail nachzuvollziehen. Deshalb sind sie weniger aufwendig als analytische Verfahren und auf beliebige Zusammenhänge anwendbar, dafür aber auch weniger genau als (gute) analytische Methoden.

Demnach gibt es bei der Wahl des Schätzverfahrens einen Zielkonflikt zwischen der Güte und dem Aufwand der Kostenschätzung.¹⁾ Ob analytische oder statistische Verfahren vorzuziehen sind, muss im Einzelfall abgewogen werden. Welches Verfahren vorteilhafter ist, hängt vor allem von der Datenverfügbarkeit und den betrachteten Kostenstellen und Kostenarten ab. Wenn Materialstücklisten und detaillierte Arbeitspläne vorliegen, bietet es sich an, die Material- und Fertigungseinzelkosten analytisch zu berechnen. Hingegen sind für die Schätzung sekundärer Kosten („Gemeinkosten“) statistische Verfahren oft die bessere Wahl – wenn ausreichende Vergangenheitsdaten vorliegen.

¹⁾ Vgl. auch im Weiteren Coenenberg/Fischer/Günther [Kostenrechnung] 549 f.; Friedl/Hofmann/Pedell [Kostenrechnung] 226. Hingegen lehnen Kilger/Pampel/Vikas [Plankostenrechnung] 287 statistische Verfahren der Kostenplanung grundsätzlich ab.

Auch künstliche neuronale Netze können nicht genauer und entscheidungsorientierter als gute analytische Verfahren sein. Ihr Einsatz ist demnach genauso abzuwägen, wie der anderer statistischer Verfahren auch. Zudem wirft die statistische Kostenschätzung einige grundsätzliche Probleme auf, denen jedes statistische Verfahren und damit auch künstliche neuronale Netze unterliegen. Das schränkt ihren Anwendungsbereich weiter ein. Diese Probleme werden im folgenden Abschnitt 3 betrachtet.

3. Allgemeine Probleme beim Einsatz statistischer Verfahren in der Kostenrechnung

a) Dynamik der Modellumwelt

Grundsätzliche Probleme statistischer Verfahren ergeben sich daraus, dass ihre Güte in hohem Maß von der Datenverfügbarkeit abhängt. Damit ist nicht nur die triviale Tatsache gemeint, dass überhaupt Vergangenheitsdaten vorliegen, sondern vielmehr die Voraussetzungen, dass

- die Umweltbedingungen im Beobachtungszeitraum und darüber hinaus gleich bleiben (dieser Abschnitt),
- die beobachteten Werte der unabhängigen Variablen einen größeren Bereich abdecken (Abschnitt b),
- unkorrelierte Beobachtungswerte der unabhängigen Variablen vorliegen (Abschnitt c).

In einem Modell wird immer unterstellt, dass die Einflussgrößen, die nicht explizit im Modell enthalten sind, – die Umweltbedingungen also – konstant bleiben. Wenn diese Voraussetzung nicht erfüllt ist, erlaubt das Modell keine zuverlässigen Schlüsse. In der Kostenrechnung können verschie-

dene Veränderungen dazu führen, dass eine auf Vergangenheitsdaten beruhende Kostenfunktion nicht in die Zukunft übertragbar ist:

- (1) Änderungen der organisatorischen Rahmenbedingungen, z. B. durch Neustrukturierung der Kostenstellen,¹⁾
- (2) Änderungen der technischen Rahmenbedingungen, z. B. durch Verfahrensänderungen oder neue Produktionspotentiale (Maschinen, Anlagen, Infrastruktur),²⁾
- (3) Änderungen des Produktportfolios bei gleichbleibender Potentialausstattung und Organisation, z. B. durch neue Produktvarianten,³⁾
- (4) Preisänderungen.⁴⁾

Kostenfunktionen sind in diesen Fällen nur haltbar, wenn sie die Veränderungen nicht als Umweltbedingungen behandeln, sondern über geeignete unabhängige Variable abbilden. Preisänderungen (4) lassen sich z. B. über Inflationsindizes einbeziehen.⁵⁾ Dynamische Entwicklungen wie in (3) versucht man z. B. in der Prozesskostenrechnung durch passende *cost driver* zu erfassen.⁶⁾ Auch technische Änderungen (2) können bei ausreichender Datenverfügbarkeit durch differenzierte Kostenfunktionen mit geschickt gewählten Variablen abgebildet werden, ohne dass sich die Struktur der Kostenfunktion ändern muss.⁷⁾ Dass dies in der Praxis zumindest versucht wird, legen empirische Studien nahe, die zu dem Ergebnis kommen, dass

¹⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 283.

²⁾ Vgl. Schweitzer/Küpper [Kostentheorie] 331; Troßmann [Potentialgestaltung].

³⁾ Vgl. Weber [Gestaltung] 33.

⁴⁾ Vgl. Schweitzer/Küpper [Kostentheorie] 331.

⁵⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 283.

⁶⁾ Siehe Abschnitt B.III.4.c, S. 37 ff., sowie Abb. B-6 und vor allem Abb. B-7, S. 48 ff.

⁷⁾ In produktionstheoretisch fundierten Kostenfunktionen sind seit jeher Einflussgrößen vorgesehen, die Produktionspotentiale und technische Prozesseigenschaften abbilden (siehe Abschnitt B.III.5.a, S. 40 ff.).

die „interne Dynamik“ eines Unternehmens – gemessen an den Veränderungen der Produktvielfalt, der Wertschöpfungstiefe, der Fertigungsprozesse und der Organisationsstruktur – positiv mit der Komplexität der Kostenrechnung korreliert.¹⁾

Weber argumentiert hingegen, „[h]ohe Veränderungsgeschwindigkeit und eine filigrane Ausgestaltung der Kostenrechnung vertragen sich nicht“²⁾, und bezieht dies ausdrücklich auf alle vier oben genannten Fälle. Hier muss allerdings differenziert werden, was mit einer „filigranen Ausgestaltung“ gemeint ist. Es können einerseits die Anzahl der Kostenstellen und andererseits die Anzahl der unabhängigen Variablen gemeint sein. Was die Anzahl der Kostenstellen betrifft, so ist Weber zuzustimmen, dass eine „entfeinerte“ Kostenstellenrechnung mit größeren, aggregierten Kostenstellen, organisatorische Änderungen (1) besser abfangen kann.

Für die Differenziertheit der Input-Seite gilt das nicht unbedingt. Zusätzliche unabhängige Variable können helfen, Veränderungen besser abzubilden (s. o.). Gleichwohl müssen ausreichend Vergangenheitsdaten zur Verfügung stehen, um die zusätzlichen Parameter zu schätzen, und es dürfen keine zu großen Korrelationen unter den im Modell enthaltenen Einflussgrößen entstehen (siehe Abschnitt c).³⁾

¹⁾ Vgl. Aust [Kostenrechnung] 178. Die Komplexität der Kostenrechnung wurde an der Anzahl der Kontierungsobjekte, der Anzahl unterschiedlicher Wertansätze und der Anzahl der Verrechnungsbeziehungen gemessen. Der p-Wert des Zusammenhangs wird mit $p = 0,2\%$ angegeben. Zu einer Studie mit ähnlichen Variablen vgl. Frank [Gestaltung] 123 ff.

²⁾ Weber [Gestaltung] 35; vgl. auch Weber [Selektives Rechnungswesen].

³⁾ Siehe auch Abschnitt D.V.2, S. 193 ff.

b) Wertebereich der unabhängigen Variablen

Streng genommen gilt eine statistisch ermittelte Regressionsfunktion nur für den Bereich, in dem sich die Vergangenheitsdaten bewegen, mit denen die Funktion parametrisiert wurde. Denn es gibt statistisch betrachtet keinen Anhaltspunkt dafür, dass der Funktionsverlauf, der aus den Vergangenheitsdaten hervorgeht, auch für Ausprägungen der unabhängigen Variablen gilt, die außerhalb des Bereichs der bisher beobachteten Daten liegen. Abb. C-13 illustriert dieses Problem am Beispiel einer linearen Regression.

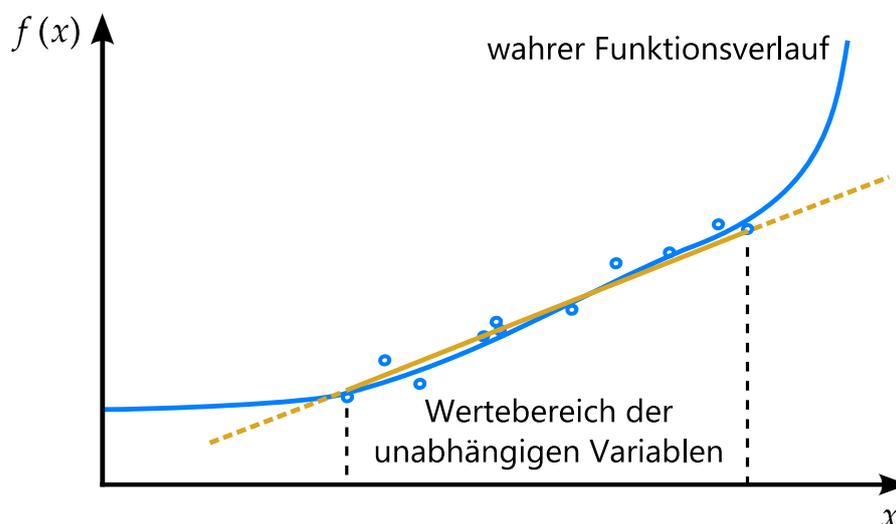


Abb. C-13: Extrapolation einer Regressionsgeraden über den Beobachtungsbereich hinaus

Der Fall, dass eine Kostenfunktion wie in Abb. C-13 außerhalb des beobachteten Bereichs einen ganz anderen Verlauf nimmt, kann z. B. dann eintreten, wenn nur Beobachtungen über die Kostenhöhe bei „Normalbeschäftigung“ vorliegen und die Kosten bei „Überbeschäftigung“ etwa wegen Überstundenzuschlägen stärker ansteigen, während bei „Unterbeschäftigung“ konstante Fixkosten anfallen.¹⁾

¹⁾ Vgl. Schweitzer/Küpper [Kostentheorie] 330 f.

Deshalb sollte bei Prognosen, die weit von den Vergangenheitswerten entfernt liegen, überlegt werden, ob die unterstellte Extrapolation des gefundenen Zusammenhangs über den Beobachtungsbereich hinaus plausibel ist oder eventuell angepasst werden muss.

c) Korrelation von Entscheidungsvariablen

Wie in Abschnitt B.III.6 (S. 52 ff.) dargelegt wurde, soll eine Kostenfunktion, wenn sie als Entscheidungsgrundlage dient, die verfügbaren Entscheidungsalternativen möglichst gut abbilden. Da eine Alternative formal als ein Variablen-tupel beschrieben werden kann (siehe S. 59), läuft dies auf die Forderung hinaus, dass in einer Kostenfunktion alle Variablen enthalten sein sollen, über die unabhängig voneinander entschieden werden kann. Diese Forderung kann von einer statistisch ermittelten Kostenfunktion aber nur erfüllt werden, wenn in der Vergangenheit bereits verschiedene Kombinationen dieser Variablen realisiert und dokumentiert wurden.

Ein einfaches Beispiel soll diese Problematik verdeutlichen: In einem Unternehmen werden zahlreiche Produktarten hergestellt und jede Produktart ist in unterschiedlichen Farben erhältlich. Jede Produktart wird mit einem eigenen Werkzeug gefertigt. Dadurch fallen je Produktart durchschnittlich 3.000 € Rüstkosten im Monat an. Das Einrichten der Maschinen auf die verschiedenen Farben verursacht im Durchschnitt monatliche Kosten in Höhe von 400 € je Farbvariante. Analytisch ergibt sich daraus die Kostenfunktion

$$K = \sum_{i=1}^{x_p} (3.000 + 400x_{F,i}) = 3.000x_p + \sum_{i=1}^{x_p} 400x_{F,i} = 3.000x_p + 400x_F \quad (\text{C.37})$$

mit x_p als Anzahl der Produktarten, $x_{F,i}$ als Anzahl der Farbvarianten der Produktart i sowie $x_F = \sum_{i=1}^{x_p} x_{F,i}$ als Gesamtzahl aller Farbvarianten.

Die Kostenfunktion (C.37) in ihrer letzten Umformung mit den Einflussgrößen x_p und x_F lässt sich auch durch eine lineare Regressionsanalyse statistisch ermitteln. Allerdings nur, wenn sowohl x_p als auch $x_{F,i}$ in einem ausreichend großen Bereich schwanken (siehe auch Abschnitt b). Im Extremfall, wenn beispielsweise alle Produktarten in sieben Farben hergestellt werden und somit $x_{F,i} = 7$ (für alle i), ist aus den Vergangenheitsdaten nicht zu erkennen, welche Kosten den Produktarten und welche den Farbvarianten zuzurechnen sind. Durch Einsetzen von $x_{F,i} = 7$ (für alle i) in (C.37) wird das Problem deutlich:

$$K = 3.000x_p + \sum_{i=1}^{x_p} 400 \cdot 7 = x_p(3.000 + 400 \cdot 7) \quad (\text{C.38})$$

$$\stackrel{(x_F = 7x_p)}{=} x_F \left(\frac{3.000}{7} + 400 \right). \quad (\text{C.39})$$

Die Höhe der Gesamtkosten für das Rüsten der Werkzeuge und das Einrichten der Farben kann vollständig durch eine einzige unabhängige Variable erklärt werden. Wenn eine Regressionsanalyse mit beiden Variablen durchgeführt wird, werden die Gesamtkosten zufällig auf die beiden Variablen verteilt. Das Ergebnis ist dann nicht mehr sinnvoll interpretierbar. Aus der Statistik ist dieses Problem als *Multikollinearität* bekannt. In (C.38)/ (C.39) sind die beiden unabhängigen Variablen perfekt korreliert; Probleme treten aber auch schon bei geringeren Korrelationen auf. Da Multikollinearität die statistische Schätzung verfälscht, werden unabhängige Variable, die stark mit anderen korrelieren, aus dem statistischen Modell ausgeschlossen.¹⁾

Für eine statistische Schätzung kann also nur eine der beiden korrelierten Variablen x_p oder x_F verwendet werden. Je nach Entscheidungssituation

¹⁾ Siehe ausführlich Abschnitt D.V.2, S. 193 ff.

kann zwar entweder Funktion (C.38) oder Funktion (C.39) geschätzt werden, in beiden Fällen werden aber der einen Variablen die Kosten der anderen mit zugerechnet. Wenn z. B. die Anzahl der Produktarten als unabhängige Variable gewählt wird (C.38), fallen jeder Produktart neben den Rüstkosten auch die Kosten für sieben Farbvarianten zu. Bei der Entscheidung über die Einführung einer neuen Produktart, kann mit dieser Kostenfunktion zwar berechnet werden, wie hoch die monatlichen Fertigungskosten wären, wenn die neue Produktart in sieben Farben hergestellt wird (5.800 €), nicht aber, wie hoch die Kosten mit weniger oder mehr als sieben Farbvarianten wären. Wie hoch die Kosten für eine weitere Farbvariante einer bestehenden Produktart wären, kann überhaupt nicht beantwortet werden, weil in (C.39) im Regressionskoeffizienten für x_F die Kosten für ein Siebtel einer zusätzlichen Produktart (statistisch) untrennbar mit den Kosten für eine zusätzliche Farbvariante verwoben sind.

Diese Problematik ist im Grunde nicht verwunderlich. Wenn in der Vergangenheit eine Produktart stets in sieben Farben hergestellt wurde, gibt es keine Erfahrungswerte zu den Kosten bei fünf, sechs, acht oder neun Farbvarianten. Statistische Methoden sind aber auf eine gewisse Menge an Erfahrungswerten angewiesen. Selbst analytische Schätzungen, die häufig auf dem Erfahrungswissen der Experten basieren, sind in diesem Fall schwierig.

Zusammenfassend lässt sich sagen, dass die statistische Schätzung von Kostenfunktionen problematisch ist, wenn über zwei Kosteneinflussgrößen zwar (relativ) unabhängig voneinander entschieden werden kann, diese Einflussgrößen in der Vergangenheit aber hoch korreliert waren. Will man dennoch beide Einflussgrößen in eine Kostenfunktion aufnehmen, muss man zumindest teilweise analytisch vorgehen. So könnte man im Beispiel

oben den Kostenzusammenhang (C.38) statistisch ermitteln und anschließend überlegen, welcher Teil des geschätzten Regressionskoeffizienten für x_p in Höhe von 5.800 € tatsächlich auf produktartbezogene Aktivitäten zurückzuführen ist und welcher Teilbetrag k_F^v auf das Einrichten einer Farbvariante. Am Ende könnte z. B. eine Kostenfunktion der Art

$$K = (5.800 - k_F^v \bar{x}_{F,i}) x_p + k_F^v x_F \quad (\text{C.40})$$

mit $\bar{x}_{F,i}$ als durchschnittlicher Farbvariantenzahl je Produktart stehen.

d) Vergleich von Schlendrian mit Schlendrian? Eignung von Istkostenfunktionen zur Vorgabe und Kontrolle

Statistisch ermittelte Kostenfunktionen sind *Istkostenfunktionen*. Sie enthalten damit auch die *strukturellen Unwirtschaftlichkeiten* der betrieblichen Istsituation.¹⁾ Für Kilger ist das der Hauptgrund dafür, „dass die Kostenplanung einer flexiblen Plankostenrechnung nicht auf statistischen Verfahren basieren darf.“²⁾ Betrachtet man diese sehr pauschale Ablehnung im Kontext des Kilgerschen Kostenrechnungssystems, in dem ausschließlich mit Sollkostenfunktionen gearbeitet wird, kann damit nur gemeint sein, dass sich statistisch ermittelte Kostenfunktionen nicht zur Vorgabe und Kontrolle, d. h. zur verhaltensorientierten Steuerung³⁾ eignen. Hier hat Kilger sicherlich recht, dass eine Istkostenfunktion nicht das beste Instrument etwa zur Vorgabe und Kontrolle von Budgets ist, da die Gefahr besteht, dass nur der alte „Schlendrian“, der in die Istkostenfunktion eingeflossen ist, mit dem aktu-

¹⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 258, 287; Schweitzer/Küpper [Systeme] 289.

²⁾ Kilger/Pampel/Vikas [Plankostenrechnung] 288.

³⁾ Siehe Abschnitt B.II.2, S. 14 ff.

ellen „Schlendrian“ verglichen wird.¹⁾ Allerdings kann aus einer Istkostenfunktion unter Berücksichtigung verhaltenswissenschaftlicher und ökonomischer Aspekte eine Sollkostenfunktion abgeleitet werden.²⁾

Abgesehen davon sind Istkostenfunktionen für andere Kostenrechnungszwecke außer der verhaltensorientierten Steuerung durchaus nützlich. Unter anderem können dadurch, dass die bestehende Kostensituation mit Istkostenfunktionen in einen systematischen, überschaubaren Zusammenhang gebracht wird, eben die von Kilger angesprochenen strukturellen Unwirtschaftlichkeiten aufgedeckt und von zufälligen Fehlern und situativen Abweichungen unterschieden werden. Die ermittelten Istkostenfunktionen können zudem mit denen anderer Betriebe oder mit analytisch berechneten Optimalkostenfunktionen verglichen werden.³⁾ Aber auch ohne externes Benchmarking liefern Istkostenfunktionen relevante Informationen zur Problemerkennung, Alternativensuche, Alternativenbewertung und vor allem zur Kostenprognose.⁴⁾

¹⁾ „Wenn z. B. in einer Abteilung sich schon seit jeher, vielleicht infolge von Überalterung, ein gewisser Schlendrian breitmacht, so vergleicht der Zeitvergleich Schlendrian mit Schlendrian; er wird auf diese Weise dessen nicht gewahr.“ (Schmalenbach [Kostenrechnung] 447). Siehe auch Abschnitt B.IV.6, S. 64 ff., insb. S. 66.

²⁾ Siehe Abschnitt B.IV.6, S. 64 ff.

³⁾ „Wenn aber der Betriebsvergleich Zahlen anderer Betriebe beibringen kann, in denen frischer Wind weht, dann kann der Schlendrian sich nicht länger verbergen.“ (Schmalenbach [Kostenrechnung] 447).

⁴⁾ Siehe Kapitel B.IV, S. 55 ff., und Kapitel E, S. 300 ff.

e) Sind statistische Verfahren „theorielos“?

Statistischen Verfahren wird – nicht nur in der Kostenrechnung – der Vorwurf gemacht, sie seien nicht theoriegestützt und Aussagen, die aus ihnen hervorgehen, deshalb nicht ausreichend begründet. Dieter Schneider formuliert in diesem Zusammenhang gewohnt pointiert:

„Seit dem schottischen Aufklärungsphilosophen David Hume ist bekannt, daß mit Mitteln der Logik (und damit auch der Mathematik und Statistik) allein nicht von der Vergangenheit auf die Zukunft geschlossen werden kann, sondern daß es dazu des Wissens um eine empirische Gesetzmäßigkeit bedarf. Eine solche Gesetzmäßigkeit muß hinter einem Prognosemodell stehen, [...]“¹⁾

Es geht also darum, dass statistisch geschätzte Kostenfunktionen nicht nach einem deduktiv-nomologischen Schema aus gesetzesartigen Aussagen und speziellen Randbedingungen abgeleitet werden, sondern einer induktiven Logik folgend einzelne Beobachtungen verallgemeinern.²⁾ Zu dieser Kritik ist anzumerken:

(1) Die Kritik richtet sich nicht gegen statistische Verfahren selbst, sondern nur gegen eine unbedachte Anwendung derselben. Richtig ist, dass statistische Verfahren ohne Rückgriff auf deduktiv gewonnene und empirisch bestätigte Wirkhypothesen auskommen können. Grundsätzlich ist mit Methoden wie künstlichen neuronalen Netzen ein reines *data mining* als ein freies, exploratives Suchen nach Strukturen in Datenmengen möglich. Mit denselben statistischen Methoden können aber genauso gut spezifische, deduktiv aus einer Theorie abgeleitete Hypothesen überprüft und präzisiert werden. Letzteres wird in Abschnitt B.III.2 (S. 25 ff.) beschrieben.

¹⁾ Schneider [Warnung] 1492.

²⁾ Vgl. Siemoneit [Wissenschaftstheorie] 111 f.

- (2) Es ist umstritten, in welchem Umfang in den empirischen Sozialwissenschaften überhaupt gesetzesartige Aussagen getroffen werden können.¹⁾ Selbst wenn gesetzesartige Aussagen über betriebliche Kostenzusammenhänge möglich sein sollten, gibt es außerhalb des Produktionsbereichs derzeit keine allgemeine, wissenschaftlich fundierte Kostentheorie, aus der sich konkrete Kostenhypothesen ableiten ließen (siehe S. 28). Die Theorielosigkeit praktischer Schätzmodelle ist also eine Folge fehlender Theorien.²⁾
- (3) Explorative statistische Verfahren können zur Hypothesengewinnung beitragen und so die Theoriebildung unterstützen.³⁾
- (4) Statistische Methoden beruhen in der Regel auf ausführlich begründeten Wahrscheinlichkeits-, Schätz- und Testtheorien – sie sind in dieser Hinsicht also gar nicht theorielos.⁴⁾ Es ist nicht einzusehen, warum eine Kostenfunktion, die durch statistische Theorien gestützt ist, schlechter fundiert sein soll als eine Kostenfunktion, die z. B. allein auf einer wenig begründeten, impliziten Kostentheorie eines Kostenrechnungssystems beruht (siehe S. 28 f.). Eine statistisch ermittelte Kostenfunktion ist immer noch besser als eine, die bloß auf Meinungen und vagen Vermutungen basiert.⁵⁾

¹⁾ Siehe Abschnitt B.IV.4, S. 60 f. Vgl. auch die in der Volkswirtschaftslehre einflussreiche „Lucas-Kritik“ (Lucas [Critique]; Lawson [Reality] 69 ff.).

²⁾ Vgl. Rehkugler [Neuronale Netze] 576.

³⁾ Vgl. Bankhofer [Data Mining] 406; Siemoneit [Wissenschaftstheorie] 113 f., mit weiteren Nachweisen. Dagegen Schneider [Warnung] 1493.

⁴⁾ Vgl. Siemoneit [Wissenschaftstheorie] 112.

⁵⁾ Vgl. Siemoneit [Wissenschaftstheorie] 114, mit weiteren Nachweisen. Schneider dagegen: „Aber ein nicht gesetzmäßiges, ja nicht einmal modellmäßig gestütztes ‚Wissen‘ täuscht nur ‚Erfahrungszusammenhänge‘ vor. Für die Entscheidungsfindung ist Selbsttäuschung nicht besser als die Einsicht, nichts zu wissen.“ ([Warnung] 1493).

- (5) Bei der Auswahl und Definition der Variablen und bei der Interpretation der geschätzten Funktionen fließen wenn nicht explizit betriebswirtschaftliche Theorien, so zumindest ökonomische Plausibilitätsüberlegungen in die statistischen Modelle ein. Um ein statistisches Modell zur Schätzung von Kostenfunktionen überhaupt aufstellen zu können, sind zweifellos betriebswirtschaftliche Kenntnisse über mögliche Wirkzusammenhänge und geeignete Maßgrößen notwendig. Außerdem werden die Ergebnisse einer Schätzung – häufig informell – auf ihre ökonomische Plausibilität überprüft. Wenn die Schätzung einen unerwarteten Zusammenhang ergibt, wird besonders gründlich untersucht, ob die geschätzte Funktion möglicherweise nur eine Korrelation, aber keinen Kausalzusammenhang ausweist. Ein treffendes Beispiel hierfür ist Pressmars Beschreibung einer Regressionsanalyse der Ausschussmenge in einer Papierfabrik in Abhängigkeit von der Produktionsgeschwindigkeit:¹⁾ Die Regressionsfunktion zeigt „entgegen der Erwartung“ einen fallenden Verlauf, d. h. mit steigender Produktionsgeschwindigkeit fällt die Ausschussmenge. Technischen Überlegungen zufolge müsste es umgekehrt sein. Der empirisch ermittelte Funktionsverlauf lässt sich aber dadurch erklären, dass der Maschinenführer bei hohem Ausschuss die Geschwindigkeit reduziert, während er bei geringem Ausschuss die Anlage auf die technisch mögliche Höchstgeschwindigkeit einstellt. Hier überlagert sich also die vermutete technische Kausalität, dass der Ausschuss mit der Produktionsgeschwindigkeit steigt, mit der organisatorischen Kausalität, dass der Maschinenführer dem entgegensteuert. Geschwindigkeit und Ausschussmenge wirken wechselseitig aufeinander ein, so dass die Definition der Ausschussmenge als

¹⁾ Vgl. auch im Weiteren Pressmar [Kosten- und Leistungsanalyse] 253. Einen ähnlichen Fall beschreiben Friedl/Hofmann/Pedell ([Kostenrechnung] 224).

abhängige Variable nicht haltbar ist – das Modell muss neu formuliert werden. Das Grundproblem ist hier wie in Abschnitt c, dass zwei Effekte in der Vergangenheit immer gemeinsam auftraten, so dass sie allein mit statistischen Methoden nicht zu trennen sind.

Schließlich sei darauf hingewiesen, dass mit dem in dieser Arbeit entwickelten Ansatz betriebsindividuelle und damit raum-zeitlich begrenzte Istkostenfunktionen ermittelt werden sollen. Es wird ausdrücklich nicht unterstellt, dass diese Kostenfunktionen optimal sind (siehe Abschnitt d). Deshalb droht auch nicht die Gefahr einer theoretisch unbegründeten Optimalitätsannahme, wie sie beispielsweise dem situativen Ansatz der Organisationstheorie (*contingency approach*) vorgeworfen wird.¹⁾ Jener geht u. a. davon aus, dass eine empirisch nachgewiesene hohe Korrelation einzelner Organisationsstrukturmerkmale mit bestimmten Kontextfaktoren darauf schließen lässt, dass die entsprechenden Strukturmerkmale in diesem Kontext optimal sind. Der *optimal fit* von Organisationsstruktur und Kontextbedingungen wird aber nicht ausreichend theoretisch begründet, sondern vor allem deshalb angenommen, weil die Struktur häufig in diesem Kontext vorkommt.

¹⁾ Vgl. auch im Weiteren Türk [Organisationsforschung] 1 ff.; Ortmann [Formen] 274 ff.; Schreyögg [Organisation] 293 ff.

4. Vorteile künstlicher neuronaler Netze

a) Abbildung nichtlinearer Zusammenhänge unter Berücksichtigung multipler Interdependenzen

Als universelle Approximatoren¹⁾ können sich künstliche neuronale Netze an beliebige Zusammenhänge anpassen und somit auch Nichtlinearitäten und Interdependenzen zwischen den unabhängigen Variablen (z. B. multiplikative Beziehungen) abbilden. Dies gereicht ihnen naturgemäß dann zum Vorteil, wenn derartige Strukturen tatsächlich vorliegen. In der Tat sind Nichtlinearitäten und Interaktionseffekte in vielen betriebswirtschaftlichen Zusammenhängen die Regel und separable, lineare Zusammenhänge die Ausnahme.²⁾ Auch für die Kostenrechnung ist nichts anderes zu erwarten.

Interdependenzen zwischen den Einflussgrößen treten z. B. schon in einfachen produktionstheoretisch fundierten Kostenfunktionen auf, wenn nicht nur die Gütermengen, sondern auch die Güterpreise von einer Einflussgröße abhängen. Beispielsweise sei die Gütermenge $r = a \cdot x_1$ von einer Einflussgröße x_1 abhängig und der entsprechende Güterpreis $q = B - b \cdot x_2$ von einer anderen Einflussgröße x_2 . Zusammen bilden sie die Kostenfunktion

$$K = q \cdot r = (B - bx_2) ax_1 = Bax_1 - abx_1x_2 \quad (\text{C.41})$$

(siehe (B.3), S. 22).³⁾ In Abb. C-14 ist sie mit den Parameterwerten $B = 30$, $a = 4$ und $b = 0,1$ dargestellt.

¹⁾ Siehe Abschnitt I.2, S. 88 f.

²⁾ Vgl. Huber/Heitmann/Herrmann [Interaktionseffekte] 697.

³⁾ Vgl. Schweitzer/Küpper [Kostentheorie] 225 f.

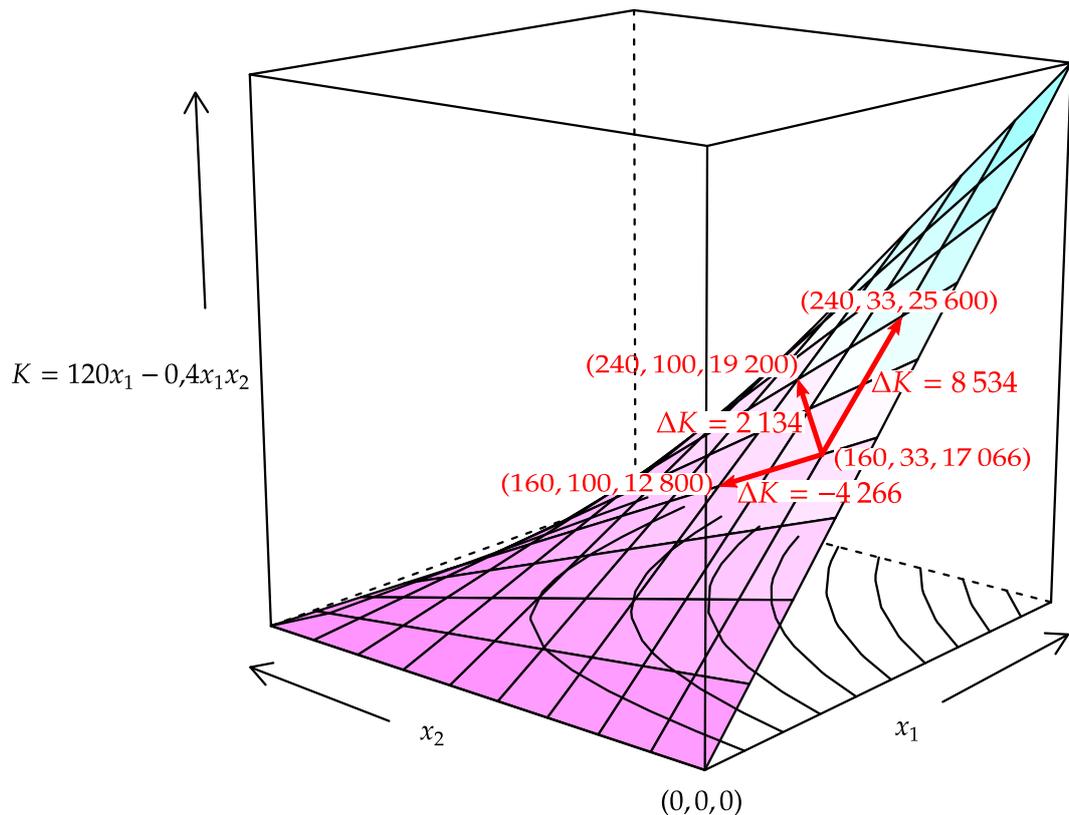


Abb. C-14: Kostenfunktion mit interdependenten Einflussgrößen

Wegen ihrer multiplikativen Verknüpfung sind die beiden Einflussgrößen x_1 und x_2 interdependent. Wenn sich beide Einflussgrößen gleichzeitig ändern, entstehen Kostenänderungen höherer Ordnung, die sich nicht verursachungsgerecht auf die Einflussgrößen verteilen lassen (siehe S. 73). In Abb. C-14 wird das daran deutlich, dass die Kostenänderung bei gleichzeitiger Veränderung beider Einflussgrößen nicht der Summe der Kostenänderungen bei isolierter Variation entspricht. Demnach ist die Kostenfunktion (C.41) *nicht additiv-separabel*, d. h. sie kann nicht in eine Summe unabhängiger Kostenfunktionen $K = f_1(x_1) + f_2(x_2)$ zerlegt werden, die jeweils nur von einer Einflussgröße abhängen (siehe S 24 f.).¹⁾ Das gilt allgemein für jede Kostenfunktion mit nicht-additiven Beziehungen zwischen den Einflussgrößen.

¹⁾ Vgl. Troßmann [Plankostenrechnung] 234 f.

Klassische Kostenrechnungssysteme wie die Grenzplan- und die Prozesskostenrechnung setzen aber Separabilität voraus. Damit das nicht schon bei der simplen Multiplikation von Preisen mit Mengen zum Problem wird, rechnen sie mit konstanten Planpreisen.¹⁾ Interdependenzen können aber nicht nur bei Preis-Mengen-Verknüpfungen, sondern zwischen beliebigen Einflussgrößen auftreten.²⁾ Mit künstlichen neuronalen Netzen sind sie problemlos abbildbar – mit den traditionellen Kostenfunktionen der Kostenrechnung hingegen nicht.

Mit den weit verbreiteten linearen Kostenfunktionen werden aber nicht nur Interaktionseffekte, sondern nichtlineare Zusammenhänge insgesamt vernachlässigt. Dass jene in einem typischen Betrieb durchaus vorkommen, zeigt Kapitel E (S. 300 ff.) an einem empirischen Datensatz. Zwar schlägt Kilger vor, nichtlineare Funktionen durch lineare Teilstücke zu approximieren.³⁾ Dafür muss aber eine Einflussgröße in mehrere Variable mit begrenzten Gültigkeitsbereichen aufgespalten werden.⁴⁾ Dieses Vorgehen ist deutlich aufwendiger als eine Neuronale-Netze-Schätzung.

b) Gute Prognosefähigkeit auch bei unbekanntem Funktionsverläufen

Zahlreiche empirische Studien bescheinigen künstlichen neuronalen Netzen in verschiedenen Anwendungsbereichen, dass sie unbekannte Daten besser vorhersagen bzw. klassifizieren können als polynomiale Regressionsverfahren.

¹⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 170 ff.

²⁾ Ein Beispiel zeigt Troßmann [Plankostenrechnung] 234.

³⁾ Vgl. Kilger/Pampel/Vikas [Plankostenrechnung] 136.

⁴⁾ Vgl. Troßmann [Plankostenrechnung] 234.

ren,¹⁾ einfache Zeitreihenanalysen,²⁾ multivariate Diskriminanzanalysen oder logistische Regressionsverfahren.³⁾ Auch wenn in diesen Ergebnissen sicherlich die ein oder andere Schönfärberei enthalten ist, kann doch wegen der Vielzahl – zum Teil gut abgesicherter – positiver Resultate die gute Prognosefähigkeit künstlicher neuronaler Netze als empirisch bestätigt gelten.

Theoretisch lässt sich die bessere Prognosefähigkeit künstlicher neuronaler Netze gegenüber polynomialen Regressionsmodellen erklären, wenn man jene als Spezialfälle eines allgemeinen, intrinsisch linearen Modells

$$f(x) = w_0 + \sum_{j=1}^J w_j \phi_j(x) \quad (\text{C.42})$$

mit Parametern w und *Basisfunktionen* ϕ betrachtet.⁴⁾ Eine lineare Regressionsfunktion beispielsweise enthält für jede Inputvariable x_i eine identische Abbildung $\phi_i(x_i) = x_i$ als Basisfunktion. Ein Modell der Form (C.42) heißt „intrinsisch linear“ oder „linearisierbar“, weil es zwar je nach Basisfunktion nichtlinear in den Variablen x sein kann, aber linear in den zu schätzenden Parametern w ist.⁵⁾ Verbreitete intrinsisch lineare Modelle sind neben polynomialen Regressionsmodellen u. a. *splines* und *wavelets*. Intrinsisch lineare

¹⁾ Siehe S. 126, Fußnote 2.

²⁾ Künstliche neuronale Netze zeigen durchweg bessere Ergebnisse als exponentielle Glättung und gleitende Durchschnitte. Im Vergleich zu Box-Jenkins-Modellen (vgl. Hill u. a. [Forecasting] 7; Alex [Neuronale Netze] 211 f.; Crone [Neuronale Netze] 225, 359) und anderen State-of-the-art-Methoden der Zeitreihenanalyse (vgl. Crone/Hibon/Nikolopoulos [Forecasting]) sind sie zumindest nicht signifikant schlechter.

³⁾ Vgl. die ausführlichen Metastudien von Hill u. a. [Forecasting]; Adya/Collopy [Forecasting]; Alex [Neuronale Netze] 180 ff.; Paliwal/Kumar [Applications]; Janetzke/Lewandowski [Prognose] 364 ff.

⁴⁾ Vgl. auch im Weiteren Bishop [Machine Learning] 138 f.

⁵⁾ Vgl. auch Backhaus u. a. [Analysemethoden] 510 f.

Modelle haben zwar einige nützliche mathematische Eigenschaften – u. a. lassen sich die Parameter w mit der Methode der kleinsten Fehlerquadrate schätzen –, sie haben allerdings den Nachteil, dass sehr viele Parameter nötig sind, um Interaktionseffekte unter den unabhängigen Variablen und andere nichtlineare Zusammenhänge darzustellen (siehe dazu Abschnitt a). Zum Beispiel enthält bereits ein allgemeines Polynom dritten Grades

$$f(\mathbf{x}) = w_0 + \sum_{i=1}^I w_i x_i + \sum_{i=1}^I \sum_{j=1}^I w_{ij} x_i x_j + \sum_{i=1}^I \sum_{j=1}^I \sum_{k=1}^I w_{ijk} x_i x_j x_k \quad (\text{C.43})$$

mit $\mathbf{x} = (x_1, \dots, x_I)^T$

$1 + \binom{I}{1} + \binom{I}{2} + \binom{I}{3}$ unabhängige Parameter. Bei fünf Inputvariablen ($I = 5$) sind das 26, bei zehn Inputs bereits 181 Parameter. Allgemein wächst die Zahl der zu schätzenden Parameter für ein Polynom k . Grades proportional zu I^k .¹⁾ Die Zahl der für eine eindeutige Schätzung mindestens nötigen Beobachtungen steigt proportional. Dieses Problem wird als „Fluch der Dimensionalität“ (*curse of dimensionality*) bezeichnet.²⁾ Intrinsisch lineare Modelle leiden besonders unter ihm.³⁾ Schon das wenig komplexe Modell (C.43) ist bei mehr als zehn Inputdimensionen praktisch nicht mehr handhabbar.

Aus diesem Grund können intrinsisch lineare Modelle nicht beliebig anpassungsfähig sein. Prinzipiell ist es denkbar, mit Polynomen höherer Ordnung oder mit *splines*, die den Inputraum in viele Teile zerlegen, ein universelles Schätzmodell für beliebige Zusammenhänge zu konstruieren. Das

¹⁾ Vgl. Bishop [Machine Learning] 36.

²⁾ Vgl. Bellman [Control Processes]; Bishop [Machine Learning] 33 ff.; Saint-Mont [Statistik] 417.

³⁾ Vgl. Bishop [Machine Learning] 173.

scheitert aber am Fluch der Dimensionalität. Um ihm zu entgehen, müssen intrinsisch lineare Modelle möglichst schlicht sein. Dadurch sind sie aber schon vor der Anwendung auf bestimmte Strukturen festgelegt. Ein lineares Regressionsmodell etwa findet immer nur lineare Zusammenhänge.

Künstliche neuronale Netze sind dem *curse of dimensionality* weniger ausgesetzt, weil sie die Tatsache ausnutzen, dass vieldimensionale Inputdaten in der Regel durch nichtlineare Transformationen auf einen wenigerdimensionalen Raum reduzierbar sind und dass die Outputvariablen oft nur von wenigen Richtungen in diesem reduzierten Inputraum abhängen.¹⁾ Die mehrschichtige Anordnung der Parameter in einem neuronalen Netz erlaubt es, seine Basisfunktionen auf die relevanten Richtungen im reduzierten Inputraum auszurichten. Ein künstliches neuronales Netz besteht im Grunde wie ein intrinsisch lineares Modell aus einer festen Anzahl an Basisfunktionen, die aber – im Gegensatz zu den Basisfunktionen des linearen Modells – anpassungsfähig sind, weil sie selbst freie Parameter enthalten.²⁾ Dieser Unterschied wird bei einem direkten Vergleich des linearen Modells (C.42) mit dem neuronalen Netz (C.6) (S. 88) deutlich. Während die Basisfunktionen in (C.42) unveränderlich sind, adaptieren sich die Aktivierungsfunktionen in (C.6) über die Parameter w_{ij} an die Trainingsdaten. Deshalb kann sich ein neuronales Netz mit einer gleichbleibenden Architektur und relativ wenigen Parametern an sehr unterschiedliche Zusammenhänge anpassen.

¹⁾ Vgl. auch im Weiteren Bishop [Machine Learning] 173.

²⁾ Vgl. auch im Weiteren Bishop [Machine Learning] 226.

Das ist die theoretische Erklärung für eine Gegebenheit, die in empirischen Studien immer wieder aufscheint:¹⁾ Wenn ein intrinsisch lineares oder ein anderes einfaches Regressionsmodell gut zum betrachteten Zusammenhang passt, sind die Ergebnisse des einfachen Modells nicht schlechter als die eines künstlichen neuronalen Netzes. Nach *Ockham's razor*²⁾ ist in diesem Fall das schlichtere Modell einem neuronalen Netz vorzuziehen. Mit anderen Worten: Wenn ein linearer Zusammenhang vorliegt, ist eine lineare Regressionsfunktion ein besseres Modell als ein künstliches neuronales Netz; wenn ein Sättigungszusammenhang vorliegt, ist eine Sättigungsfunktion³⁾ ein besseres Modell als ein künstliches neuronales Netz. Wenn es aber kein Vorwissen über den betrachteten Zusammenhang gibt, ist ein künstliches neuronales Netz besser als einfache Regressionsmodelle geeignet, den unbekanntem Funktionsverlauf zu schätzen.

c) Einfache Modellbildung und Anwendung

Aus den beiden vorhergehenden Abschnitten wird deutlich, dass künstliche neuronale Netze dem Modellierer einen Teil seiner Arbeit abnehmen. Sie bestimmen selbständig die Form des Zusammenhangs, berücksichtigen nichtlineare Interaktionen unter den unabhängigen Variablen und transformieren die Daten in einen wenigerdimensionalen Raum. Das erleichtert die Modellbildung vor allem dann, wenn keine verlässlichen Informationen über die Art des funktionalen Zusammenhangs und mögliche Interaktionseffekte vorliegen. Die Wahrscheinlichkeit, dass dem so ist, steigt rapide mit den Inputdimensionen. Je mehr Einflussgrößen berücksichtigt werden sol-

¹⁾ Vgl. z. B. Rehkugler [Neuronale Netze] 575; Smith/Mason [Cost Estimation] 156 ff.; Liu [Cost Estimation] 221 f.

²⁾ Vgl. Baker [Simplicity].

³⁾ Vgl. Backhaus u. a. [Analysemethoden] 512 f.

len, desto schwieriger wird z. B. die visuelle Darstellung der Daten, um darin erste Anhaltspunkte für mögliche Funktionsverläufe zu finden.

Auch wenn eine konkrete Regressionsfunktion formuliert werden kann, ist bei komplizierten Zusammenhängen ein künstliches neuronales Netz oftmals einfacher zu handhaben; u. a. weil es ab einer gewissen Problemkomplexität weniger Parameter enthält als etwa eine Polynomfunktion mit mehreren Einflussgrößen (siehe S. 151). Aus demselben Grund benötigt die Parametrisierung künstlicher neuronaler Netze in der Regel weniger Rechenzeit als die der klassischen nichtlinearen Regressionsmodelle. Die Modellierung ist zudem weniger fehleranfällig, weil das künstliche neuronale Netz nicht schon vor der Schätzung auf einen bestimmten Funktionstyp festgelegt und damit voreingenommen (*biased*) ist.¹⁾

In der Kostenrechnung kommt noch hinzu, dass dort regelmäßig sehr viele Kostenfunktionen geschätzt werden müssen. Schon mittelständische Unternehmen haben oft mehr als einhundert Kostenstellen.²⁾ Selbst wenn nur für die Hälfte eine Kostenfunktion ermittelt werden soll, ist eine individuelle Auswahl der passenden Funktionsform sehr zeitintensiv. Mit einer weitgehend selbständigen Funktionsapproximation durch ein neuronales Netz lässt sich dieser Aufwand reduzieren.

Ferner lassen sich künstliche neuronale Netze gut in Metaheuristiken einbinden, um beispielsweise nicht signifikante Einflussgrößen automatisch zu entfernen und das Gesamtmodell zu optimieren (siehe Kapitel D.XII, S. 267 ff.). Insgesamt ist ein geschickt implementiertes Schätzmodell mit künstlichen neuronalen Netzen so benutzerfreundlich, dass es auch ohne

¹⁾ Wenn *over-* und *underfitting* vermieden werden (s. u.).

²⁾ Vgl. Schäffer/Weber/Mahlendorf [Trends] 16.

statistisches Expertenwissen bedient werden kann. Das ist für den praktischen Einsatz in der Kostenrechnung ein entscheidender Vorteil, da Kostenrechner und Controller mit dieser Unterstützung eher bereit sein werden, die stark vereinfachenden Prämissen der traditionellen Kostenrechnung (lineare, additiv-separable Kostenfunktionen)¹⁾ aufzugeben und betriebliche Kostenzusammenhänge adäquat zu modellieren.²⁾ Dadurch sind Kosteninformationen, die in Entscheidungsprozesse einfließen,³⁾ besser fundiert und die Entscheidungsunterstützungsfunktion der Kostenrechnung wird besser erfüllt. Wie eine „geschickte Implementierung“ aussehen kann, zeigen die Kapitel D und E.

5. Besondere Schwierigkeiten beim Einsatz künstlicher neuronaler Netze

a) Vermeidung von *overfitting*

Wie in Abschnitt I.2 (S. 88 ff.) gezeigt wurde, hat die universelle Anpassungsfähigkeit künstlicher neuronaler Netze den Nachteil, dass sie sich „zu gut“ an die Trainingsdaten anpassen können. Darunter leidet die Generalisierungsleistung, also die Fähigkeit, neue, unbekannte Daten korrekt zu prognostizieren. Im Vergleich zu intrinsisch linearen Regressionsmodellen (siehe S. 150) ist die Overfitting-Gefahr neuronaler Netze wegen ihrer größeren Flexibilität tendenziell höher.

Grundsätzlich könnte man *overfitting* vermeiden, indem man dem künstlichen neuronalen Netz genau so viele verdeckte Neuronen gibt, dass es ge-

¹⁾ Siehe S. 24 f. und Abschnitt B.III.4, S. 31 ff.

²⁾ Vgl. Mertens/Rässler [Prognoserechnung] 4.

³⁾ Siehe Kapitel B.IV, S. 55 ff.

rade noch den gesuchten Zusammenhang, nicht aber das zufällige Rauschen der Trainingsdaten abbilden kann. Da aber der gesuchte Zusammenhang unbekannt ist, kann die problemadäquate Modellkomplexität nicht von vorneherein bestimmt werden. Der optimale Netzaufbau muss deshalb in einem heuristischen Prozess gefunden werden.

Einige Heuristiken zur Vermeidung von *overfitting* werden in Kapitel D.XI (S. 258 ff.) vorgestellt. Die meisten davon benötigen neben den eigentlichen Trainingsdaten einen zusätzlichen Validierungsdatensatz, an dem die Generalisierungsfähigkeit des trainierten Netzes überprüft wird. Die Menge der benötigten Validierungsdaten kann zwar durch Techniken wie die Kreuzvalidierung¹⁾ recht klein gehalten werden; im Allgemeinen benötigen künstliche neuronale Netze für eine gute Prognoseleistung aber mehr Vergangenheitsdaten als intrinsisch lineare Regressionsmodelle. In jedem Fall benötigen sie mehr Daten als eine lineare Regression.

b) Interpretation der geschätzten Parameter

Eine lineare Kostenfunktion $K = K^{\text{fix}} + \sum_{i=1}^I k_i^y x_i$ (siehe (B.7), S. 25) hat den Vorteil, dass ihre Parameter K^{fix} und k_i^y direkt als fixe bzw. variable Kosten interpretiert werden können. Mit den Parametern w eines künstlichen neuronalen Netzes ist das nicht ohne weiteres möglich. Dadurch, dass im neuronalen Netz mehrfach summiert und nichtlinear transformiert wird, lässt sich einem einzelnen Parameter keine reale Bedeutung zuordnen. Das wird in der Literatur immer wieder als Nachteil künstlicher neuronaler Netze angeführt.²⁾ Dabei tritt dieses Problem bei jedem Schätzmodell auf, das

¹⁾ Siehe Abschnitt D.XI.1, S. 258 ff.

²⁾ Vgl. z. B. Rehkugler [Neuronale Netze] 575; Smith/Mason [Cost Estimation] 157; Liu [Cost Estimation] 222.

komplizierter als eine lineare Regression ist. Wie sind beispielsweise die Parameter des Polynoms (C.43) (S. 151) zu interpretieren? Auch für die Kostenfunktion (C.41) (S. 147) können mit statistischen Methoden nur die Parameterprodukte Ba und ab als Ganzes geschätzt werden. Eine Aufspaltung in die einzelnen Parameter a, b und B ist rein statistisch nicht möglich, so dass die Interpretation hier ebenfalls schwerfällt.

Vor allem im Zusammenhang mit der Kostenschätzung wird wiederholt erklärt, ein von einem künstlichen neuronalen Netz geschätzter Zusammenhang lasse sich nicht auf seine Plausibilität überprüfen:

„Wenn z. B. bei einer multiplen Regressionsanalyse ein negativer Einfluss des Werkstückgewichts auf die Kosten errechnet wird, kann man davon ausgehen, dass das ein zufälliges statistisches Ergebnis (ähnlich: Zahl der Geburten = Funktion der Anzahl der Störche) ist, das nichts mit der Wirklichkeit zu tun hat. In den neuronalen Netzen ist eine solche sinnlose Verknüpfung nicht zu erkennen.“¹⁾

„Zu beachten bleibt jedoch, dass – im Gegensatz z. B. zur traditionellen Optimierungs- oder Regressionsrechnung – die kostenbestimmenden Zusammenhänge verborgen bleiben, somit auch Scheinkorrelationen unentdeckt bleiben und Beeinflussungsmöglichkeiten im Sinne kostengünstigerer Konstruktion auf analytischem Weg ausgeschlossen sind.“²⁾

In Abschnitt E.III.3 (S. 322 ff.) dieser Arbeit wird gezeigt, wie mit einfachen Differenzrechnungen die absoluten Beiträge einer Einflussgröße zur Kostenhöhe berechnet werden können. Diese Werte lassen sich als variable Kosten der Einflussgröße interpretieren. Außerdem wird in Abschnitt E.I.2 (S. 307 ff.) gezeigt, wie sich aus einem künstlichen neuronalen Netz zweidimensionale Funktionsgraphen ableiten lassen, die die Abhängigkeit der

¹⁾ Ehrlenspiel/Kiewert/Lindemann [Entwickeln] 470. Zum grundsätzlichen Problem, dass eine statistische Korrelation wenig über den zugrundeliegenden Kausalzusammenhang aussagt, siehe Abschnitt 3.e, Absatz (5), S. 145 f.

²⁾ Coenenberg/Fischer/Günther [Kostenrechnung] 542 f. Vgl. auch Rehkugler [Neuronale Netze] 575; Smith/Mason [Cost Estimation] 157.

Kostenhöhe von einer Einflussgröße unter Ceteris-Paribus-Bedingungen darstellen. Damit lassen sich unintuitive Zusammenhänge und Möglichkeiten zur aktiven Kostenbeeinflussung leicht aufdecken.

6. Zusammenfassende Betrachtung der Zweckmäßigkeit künstlicher neuronaler Netze zur Schätzung von Kostenfunktionen

Bei der Wahl eines Verfahrens zur Schätzung von Kostenfunktionen in der Kostenrechnung müssen Aufwand und Nutzen abgewogen werden. Analytische Verfahren erlauben eine präzise, zukunftsorientierte Schätzung, die als Grundlage für Entscheidungszwecke und Zwecke der verhaltensorientierten Steuerung geeignet ist. Sie sind allerdings recht aufwendig und nur schwer durchführbar, wenn keine technischen Input-Output-Beziehungen vorliegen. Statistische Verfahren sind hingegen universell einsetzbar und in der Regel weniger aufwendig, vernachlässigen dafür aber Details und sind auf geeignete Vergangenheitsdaten angewiesen. Außerdem werden auf statistischem Weg grundsätzlich Istkostenfunktionen ermittelt, die sich nur bedingt zur verhaltensorientierten Steuerung eignen. Auch für Prognosezwecke taugen sie nur unter der einschränkenden Bedingung, dass sich die Umweltbedingungen nicht (zu sehr) ändern.

Gegenüber anderen statistischen Verfahren haben künstliche neuronale Netze den Vorteil, dass sie auch unbekannte Funktionstypen zuverlässig schätzen sowie beliebige Funktionsverläufe und Interaktionseffekte unter den unabhängigen Variablen abbilden können. Zudem lassen sich Schätzungen mit künstlichen neuronalen Netzen leicht automatisieren. Eine sorgfältige Auswahl der Modellvariablen nach Kriterien der Entscheidungsrelevanz und die Prüfung der geschätzten Kostenfunktionen auf

ökonomische Plausibilität bleiben auch beim Einsatz von künstlichen neuronalen Netzen unabdingbar, sind aber nicht schwieriger als sonst auch.

Demnach ist die Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen insbesondere dann vorteilhaft, wenn die Form des funktionalen Zusammenhangs nicht bekannt ist oder der Nutzen, den eine gründlichere statistische oder analytische Modellierung bringt, den höheren Aufwand gegenüber der Verwendung künstlicher neuronaler Netze nicht rechtfertigt.

Allerdings werden zur Parametrisierung künstlicher neuronaler Netze in der Regel mehr Vergangenheitsdaten benötigt als bei klassischen Regressionsverfahren. Sinkende Kosten für Speicherplatz und Rechenleistung sowie die zunehmende Verbreitung vernetzter Sensoren, mobiler Endgeräte und internetbasierter Dienstleistungen sorgen jedoch dafür, dass in Betrieben immer mehr Daten automatisiert erhoben und dauerhaft gespeichert werden.¹⁾ Viele davon sind auch für die Kostenrechnung nutzbar. Deshalb ist in Zukunft kein Mangel an Daten zu erwarten, sondern vielmehr ein Überfluss, aus dem es die relevanten Daten herauszufiltern, zu strukturieren und zu analysieren gilt. Diese Trends und Herausforderungen – und zum Teil auch die Methoden zu ihrer Bewältigung – werden unter dem Begriff *big data* zusammengefasst. Künstliche neuronale Netze sind ein fester Bestandteil des Big-Data-Instrumentariums – u. a. weil sie geeignet sind, Daten mit sehr vielen Dimensionen auszuwerten (siehe S. 152).

Als ein Schlüssel zur sinnvollen Verwertung von *big data* gilt die Einbettung von Big-Data-Analysen in übersichtliche und einfach zu bedienende Softwareanwendungen, die von normalen Mitarbeitern, die keine Statistik-

¹⁾ Vgl. auch im Weiteren Manyika u. a. [Big Data]; Chen/Chiang/Storey [Big Data] 1166 ff.; McAfee/Brynjolfsson [Big Data] 63; Buhl u. a. [Big Data] 64; Fan/Bifet [Big Data] 1; Fox/Do [Big Data] 739 f.; Baars u. a. [Gestaltung] 14 f.; jeweils mit weiteren Nachweisen.

experten oder Programmierer sind, verwendet und verstanden werden können.¹⁾ Das Gleiche gilt für fortgeschrittene statistische Methoden in der Kostenrechnung: Nur wenn sie von Kostenrechnern, Controllern und Entscheidungsträgern ohne statistisches Expertenwissen angewendet werden können, werden sie auch eingesetzt. Deshalb zeigt das folgende Kapitel D, welche Grundfunktionen für eine benutzerfreundliche, weitgehend automatisierte Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen notwendig sind und wie sie in einem objektorientierten Programm umgesetzt werden können. Das anschließende Kapitel E veranschaulicht einige zentrale kostenrechnerische Anwendungen dieser Implementierung.

¹⁾ Vgl. Barton/Court [Advanced Analytics] 82; Schermann u. a. [Big Data] 282 f.

D. Objektorientierte Umsetzung der Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen

I. Java-Programm „Cenobi“

Ein herausragender Vorteil künstlicher neuronaler Netze ist ihre Fähigkeit, sich an beliebige funktionale Zusammenhänge anzunähern, ohne dass formal ein gesuchter Funktionstyp vorgegeben werden muss. Damit nehmen sie dem Modellierer eine seiner schwersten Aufgaben ab und machen es möglich, ein Anwendungsprogramm zur Schätzung von Kostenfunktionen zu entwickeln, das von Nutzern ohne statistisches Expertenwissen bedient werden kann. Dass das Expertenwissen, das dem Nutzer möglicherweise fehlt, bei der Programmierung umso mehr benötigt wird, zeigt dieses Kapitel D. In den folgenden Teilkapiteln wird untersucht, welche Funktionen für eine möglichst anwenderfreundliche Software zur Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen wichtig sind und wie sie in einer objektorientierten Programmierung umgesetzt werden können.

Resultat dieser Auseinandersetzung ist das Java-Programm *Cenobi*, das unter www.cenobi.org kostenlos heruntergeladen werden kann. Der Programmcode ist auf sourceforge.net/projects/cenobi/ in Form eines Maven-Projekts¹⁾ frei zugänglich und darf unter den Bedingungen der GNU General Public License, Version 3, genutzt, weitergegeben und an eigene Bedürfnisse angepasst werden.

¹⁾ Maven ist eine Sammlung von Programmierstandards und -werkzeugen für die standardisierte Erstellung und das Projektmanagement von Java-Programmen (vgl. Massol u. a. [Maven]; O'Brien u. a. [Maven]).

Das Akronym *Cenobi* steht für „cost estimation with neural-operated business intelligence“. Eine ausführliche Erörterung des Business-Intelligence-Begriffs soll an dieser Stelle nicht stattfinden.¹⁾ Es sei aber erwähnt, dass Data-Mining-Methoden und damit auch künstliche neuronale Netze in der wissenschaftlichen Literatur üblicherweise zum Business-Intelligence-Instrumentarium gezählt werden.²⁾ Sie fallen allerdings unter einen weit gefassten, analyseorientierten Business-Intelligence-Begriff, der zwar in der Wissenschaft häufig vertreten wird;³⁾ in der betrieblichen Praxis versteht man *business intelligence* jedoch eher als Sammelbegriff für Werkzeuge zur Integration, Aufbereitung, Speicherung und Bereitstellung von Daten (in der Regel in einem *data warehouse*) sowie zu ihrer multidimensionalen, OLAP-basierten Präsentation in Managementberichten, *cockpits*, *dashboards* oder *scorecards*.⁴⁾ Um davon die Analyse der Daten mit fortgeschrittenen Methoden der Statistik und des maschinellen Lernens zu unterscheiden, haben sich mittlerweile die Begriffe *predictive analytics* und *advanced analytics* etabliert.⁵⁾ Eine eher an der Praxis orientierte Bezeichnung für das entstandene Programm wäre demnach „cost estimation with neural-operated advanced analytics“. Die Namenswahl fiel schlussendlich auf das schönere Akronym.

¹⁾ Vgl. dafür Burmester [Business-Intelligence-Systeme] 109 ff.

²⁾ Vgl. Mertens/Meier [Informationsverarbeitung 2] 36; Gluchowski/Gabriel/Dittmar [Management Support Systeme] 198 ff.; Chamoni/Gluchowski [Informationssysteme] 338 ff.; Turban/Sharda/Delen [Decision Support] 241 ff.

³⁾ Vgl. Turban/Sharda/Delen [Decision Support] 19; Kemper/Baars/Mehanna [Business Intelligence] 4 ff.; Burmester [Business-Intelligence-Systeme] 110 f., mit weiteren Nachweisen.

⁴⁾ Vgl. Turban/Sharda/Delen [Decision Support] 20; Bange [Werkzeuge].

⁵⁾ Vgl. Felden [Predictive Analytics]; Barton/Court [Advanced Analytics]; Franks [Taming].

II. Beispieldatensatz eines Herstellers von Kunststoffprodukten im Spritzgussverfahren

Die im Folgenden in den Kapiteln D und E beschriebenen Methoden und Anwendungen werden an einem empirischen Datensatz getestet. Dadurch können einerseits Probleme, die bei der Schätzung von Kostenfunktionen in der Praxis auftreten, exemplarisch aufgezeigt und andererseits die Leistungsfähigkeit der vorgestellten Lösungen überprüft werden. Der Datensatz ist im Anhang vollständig abgedruckt.

Die Daten wurden von einem mittelgroßen, kunststoffverarbeitenden Unternehmen für diesen Zweck zur Verfügung gestellt. Das Unternehmen entwickelt Gebrauchsgegenstände aus Kunststoff in unterschiedlichen Größen, stellt sie im Spritzgussverfahren her, vermarktet und vertreibt sie. Weil es sich zum Teil um wettbewerbsrelevante Daten handelt, wurden sie linear transformiert, damit aus der absoluten Höhe der Zahlen nicht darauf geschlossen werden kann, von welchem Unternehmen sie genau stammen. Insgesamt enthält der Datensatz 32 Kostenstellen mit ihrer monatlichen Kostensumme und 17 Variable, die als Kosteneinflussgrößen in Frage kommen. Die Aufzeichnungen reichen über 63 Monate.

Die Kostenstellendaten stammen aus dem SAP-System des Unternehmens. Berücksichtigt wurden nur Endkostenstellen. Einige Fertigungskostenstellen sowie die ursprünglich nach Regionen und Großkunden gegliederten Vertriebskostenstellen wurden zusammengefasst. Abb. D-1 zeigt einen Ausschnitt mit den ersten Beobachtungswerten einiger Kostenstellen.

	Inmold Labeling ¹⁾	Montage/Verpackung	Siebdruck	Recycling/Mahlen	Auftragsbereitstellung	Methodenzeitmessung ²⁾
	1.655,58	170.820,35	37.953,80	24.262,21	20.857,19	-
	7.041,63	161.476,19	39.476,06	25.690,18	23.683,83	-
	89.529,02	170.754,16	48.022,46	24.522,48	24.531,26	-
	1.905,29	166.760,47	36.080,35	24.407,40	19.218,46	-
	1.905,29	164.059,78	36.983,83	24.953,29	19.263,11	-
	8.337,19	172.737,41	34.158,77	27.452,07	19.487,99	18.621,54
	192.372,50	158.730,25	34.958,51	29.534,24	19.484,27	18.632,93
...	2.047,20	214.083,23	35.828,59	25.741,11	20.886,82	18.326,82
	- 33.879,62	174.832,55	37.305,86	25.046,92	19.219,07	17.726,50
	113.850,49	174.871,95	35.552,79	24.619,48	21.154,15	17.726,50
	12.321,25	162.720,03	35.374,23	25.090,87	19.549,22	17.726,50
	7.240,81	143.122,34	47.568,56	28.149,81	16.883,64	20.094,19
	21.566,10	156.006,67	34.394,14	26.171,06	20.118,35	17.932,00
	20.323,14	144.256,02	34.649,99	25.109,82	28.967,20	18.025,62
	21.529,23	162.318,48	33.579,10	25.784,72	28.967,20	17.844,78
				⋮		

Abb. D-1: Kostenstellen des Beispieldatensatzes (Ausschnitt)

Der Ausschnitt in Abb. D-1 soll einige Probleme empirischer Kostenrechnungsdaten verdeutlichen, die eine Vorverarbeitung der Daten notwendig machen.³⁾ Er ist deshalb nicht repräsentativ für den gesamten Datensatz. Insgesamt kommen fehlende Werte, negative Zahlen oder extreme Schwankungen im Beispieldatensatz seltener vor als in diesem Ausschnitt. Insbesondere an der Kostenstelle Inmold Labeling in Abb. D-1 zeigt sich aber, dass Kosten, die nicht in der Periode erfasst werden, in der sie entstanden sind, die statistische Schätzung von Kostenfunktionen erheblich erschweren. Allem Anschein nach wurden einige Kostenarten für das *in-*

¹⁾ Beim *inmold labeling* werden bedruckte oder strukturierte Folien in die Form eingelegt, bevor der heiße Kunststoff eingespritzt wird. Dadurch wird die Folie mit dem Kunststoff zu einer dauerhaften Verbindung verschweißt.

²⁾ MTM-Analyse (*methods-time measurement analysis*, Arbeitsablauf-Zeitanalyse) – eine Methode zur Analyse und Planung von Arbeitsabläufen.

³⁾ Siehe dazu ausführlich Teilkapitel VI, S. 200 ff.

mold labeling zunächst nur blockweise erfasst (daher die vereinzelt hohen Beträge) und Rückbuchungen wiederum blockweise in anderen Monaten vorgenommen (daher der negative Betrag). Es kann zwar versucht werden, die Ausreißer mit statistischen Methoden zu entdecken und zu beseitigen – genauer gesagt, die Kostenspitzen abzutragen und auf die umliegenden Perioden zu verteilen, ohne die Gesamtsumme der Kosten zu verändern. Besser wäre es allerdings, die verdächtigen Werte im Detail nachzuvollziehen und ggf. manuell zu korrigieren. Damit diese Schwierigkeiten erst gar nicht entstehen, sollten Kosten grundsätzlich in den Perioden erfasst werden, in denen der Güterverzehr stattgefunden hat.¹⁾

Ein weiteres Problem sind Änderungen der Kostenstellenstruktur. Zum Teil kann es dadurch abgefangen werden, dass mehrere Kostenstellen nur aggregiert betrachtet werden (siehe S. 136). Im Beispieldatensatz wird u. a. mit den Vertriebskostenstellen so verfahren, weil in diesem Bereich regelmäßig neue Kostenstellen geschaffen werden (z. B. für neue Großkunden oder zusätzliche Verkaufsregionen). Wenn nur die Kostensumme über alle Vertriebskostenstellen betrachtet wird, sind diese Änderungen der Kostenstellenstruktur innerhalb des Vertriebsbereichs unproblematisch. Anders sieht es bei der Kostenstelle Methodenzeitmessung in Abb. D-1 aus. Sie lässt sich nicht sinnvoll mit anderen Kostenstellen zusammenfassen. Da nur die ersten fünf Beobachtungen fehlen, ist eine zuverlässige Schätzung der Kostenfunktion aber immer noch möglich. Allerdings würde es die Schätzung verfälschen, wenn für die ersten fünf Zeitpunkte ein Betrag von 0 € angesetzt wird. Eine Lösung besteht darin, die Kostenstelle Methodenzeitmessung aus dem Gesamtdatensatz herauszunehmen und ein separates Schätzmodell aufzustellen, das nur die Zeitpunkte enthält, für die Daten zu dieser

¹⁾ Siehe dazu auch die Definition des Kostenbegriffs auf S. 8.

Kostenstelle vorliegen. Wenn die Kostenstelle jedoch im Gesamtmodell verbleiben soll, können die fehlenden Werte geschätzt und in den Datensatz eingefügt werden,¹⁾ so dass die Kostenstelle im weiteren Verlauf der Modellbildung und –anwendung nicht separat gehandhabt werden muss.

Abb. D–2 zeigt die unabhängigen Variablen des Beispieldatensatzes mit ihren ersten Beobachtungswerten. Ziel war es, Vergangenheitsdaten für möglichst viele der in der Literatur häufig genannten Einflussgrößen²⁾ zu finden.

Rohmaterialverbrauch in t	geleistete Std. Spritzgussmaschinen	Zuhaltekraft gesamt in t	Anzahl verkaufter Produktarten	Anzahl verkaufter Varianten	Umsatz mit eigener Produktion in €
-	46.477,44	20.429	1.057	2.563	13.284.602,60
-	50.032,76	20.429	1.021	2.466	12.450.419,07
1.501	47.046,09	22.505	992	2.563	12.206.393,08
⋮					
Umsatz mit Handelsware in €	Umsatz aus Rohmaterialverkauf in €	Anzahl eingesetzter Rohmaterialtypen	Anzahl Kreditoren (inkl. Banken)	Anzahl Einkaufsrechnungen	Anzahl Rechnungen (ohne Einkauf)
2.022.151,13	465.649,87	59	483	-	3.413
1.732.707,38	499.830,41	57	452	-	3.092
1.241.461,91	329.604,99	66	419	448	2.738
⋮					
Anzahl Warenempfänger	Anzahl Lagerbewegungen	Fremdleistung Produktion in €	Fremdleistung Verpackung/Montage in €	Fremdl. Entw. u. Versuchsarb. in €	
1.033	637.024	3.272.255,91	45.034,76	32.385,33	
825	697.582	1.750.977,29	32.163,65	70.561,90	
812	716.911	1.136.597,34	37.297,90	25.398,56	
⋮					

Abb. D–2: Unabhängige Variable des Beispieldatensatzes

¹⁾ Siehe Abschnitt VI.2, S. 211 ff.

²⁾ Siehe Abschnitt B.III.5, S. 40 ff.

Einige Variable aus Abb. D-2 sind erklärungsbedürftig. Die *Zuhaltekraft* ist der Druck, mit dem eine Spritzgussmaschine die beiden Werkzeughälften, in die der Kunststoff eingespritzt wird, zusammenhält. Je mehr Masse eingespritzt wird, desto höher muss die Zuhaltekraft sein. Die Zuhaltekraft ist demnach eine Kennzahl für die Größe der Maschine und der auf ihr hergestellten Teile. Der Beispieldatensatz enthält die Summe der Zuhaltekraft aller Spritzgussmaschinen. Sie ändert sich nur, wenn eine neue Maschine angeschafft oder eine bestehende Maschine erweitert wird.

Die Anzahl der verkauften *Produktarten* wurde ermittelt, indem im SAP-System die Anzahl der Produktnummern abgefragt wurde, für die ein positiver Umsatz im entsprechenden Monat vorhanden ist. Produktionstechnisch betrachtet entspricht die Anzahl der Produktarten der Anzahl der eingesetzten Werkzeuge.

Die Anzahl der *Farbvarianten* wurde auf ähnliche Weise bestimmt. Hier wurde allerdings in der Abfrage zusätzlich nach Farbcodes differenziert. Jede Produktart wird in verschiedenen Farbvarianten hergestellt. Die beiden Variablen stehen also in einem hierarchischen Zusammenhang. Das führt, wie in Abschnitt C.III.3.c (S. 138 ff.) erörtert wurde, dazu, dass beide Variablen hoch korreliert sind. Deshalb sollte nur eine von beiden Variablen in das Schätzmodell aufgenommen werden.¹⁾ Welche das ist, hängt von der Entscheidung ab, die durch das Schätzmodell unterstützt werden soll.²⁾ Wenn über die Einführung einer neuen Produktart zu entscheiden ist, sollte das Modell die Kosten einer zusätzlichen Produktart prognostizieren können. Wenn andererseits über eine zusätzliche Farbvariante für eine bestehende Produktart nachgedacht wird, sollte das Modell diese Kosten pro-

¹⁾ Siehe auch Abschnitt V.2, S. 193 ff.

²⁾ Siehe auch Abschnitt B.III.6, S. 52 f.

agnostizieren können. Je nach Entscheidung müssen demnach Modelle mit unterschiedlichen Einflussgrößen aufgestellt werden. Dabei ist allerdings zu beachten, dass die Kosten zweier hoch korrelierter Einflussgrößen auf statistischem Weg nicht separiert werden können.¹⁾ Die geschätzten Kosten einer zusätzlichen Produktart enthalten immer auch die Kosten für eine durchschnittliche Anzahl an Farbvarianten. Ebenso sind im komplementären Modellansatz in den Kosten für eine zusätzliche Farbvariante immer auch produktartbezogene Kosten anteilig enthalten.

Etwas ungewöhnlich ist die Verwendung von *Umsatzgrößen* als Kosteneinflussgrößen. Sie wurden vor allem deshalb in den Datensatz aufgenommen, weil sie in der bestehenden Kostenplanung des Unternehmens als zentrale Einflussgrößen gebraucht werden. Umsatzgrößen können gute Prädiktoren zur Kostenprognose sein; sie haben allerdings den schwerwiegenden Nachteil, dass sie keine Entscheidungsvariablen sind. Der Umsatz lässt sich zwar durch verschiedene Maßnahmen indirekt beeinflussen, hängt aber auch von vielen externen Faktoren ab. Deshalb lassen sich aus einem Modell mit dem Umsatz als Kosteneinflussgröße keine betriebspolitischen Empfehlungen ableiten. Zum Beispiel kann nicht erwartet werden, dass bei einer (aktiven) Umsatzreduktion auch die Kosten sinken. Aus entscheidungsorientierter Sicht sollten statt des Umsatzes besser die einzelnen beeinflussbaren und nicht beeinflussbaren Variablen, die auf ihn einwirken, betrachtet werden. Die sogenannte „Lucas-Kritik“ nimmt ihren Ausgang bei einem ähnlichen Problem: Lucas kritisiert den wirtschaftspolitischen Fehlschluss, aus der statistisch erwiesenen positiven Korrelation von Inflation und Wirtschafts-

¹⁾ Siehe auch im Weiteren Abschnitt C.III.3.c, S. 138 ff.

wachstum zu folgern, langfristiges Wirtschaftswachstum lasse sich durch eine dauerhaft hoch gehaltene Inflation realisieren.¹⁾

Ebenfalls nicht selbstverständlich sind *Fremdleistungskosten* als Kosteneinflussgrößen. Dieser Zusammenhang ist aber nur auf den ersten Blick paradox. Denn durch die Auslagerung von Teilen des Produktionsprozesses, können sich die Kosten für diese Teilprozesse reduzieren. Andererseits entsteht möglicherweise ein erhöhter Koordinationsbedarf, der an anderer Stelle zu steigenden Kosten führt. Zudem lassen sich die Fremdleistungskosten direkt beeinflussen und vor allem bei langfristigen Verträgen gut planen. Sie sind also Entscheidungsvariablen. Dementsprechend werden Fremdleistungskennzahlen auch in der Kostenrechnungsliteratur des Öfteren als Kosteneinflussgrößen vorgeschlagen.²⁾

III. Schätzung von Kostenfunktionen als Teil eines Knowledge-Discovery-in-Databases-Prozesses

Bei der Einführung des Beispieldatensatzes in vorhergehenden Teilkapitel II ist bereits deutlich geworden, dass vor der eigentlichen Funktionsschätzung einige Arbeiten notwendig sind. Es müssen Variablen ausgewählt, Daten gesammelt, Ausreißer beseitigt und fehlende Werte eingesetzt werden. Außerdem müssen nach der Parametrisierung die Modellgüte überprüft sowie die Ergebnisse ausgewertet und interpretiert werden.

¹⁾ Vgl. Lucas [Critique] 19. Siehe auch Abschnitt C.III.3.e, S. 143 ff.

²⁾ Siehe Abb. B-5, S. 42 f., und Abb. B-7, S. 49. Der Fremdleistungsanteil geht beispielsweise auch in die Berechnung der Fertigungstiefe ein.

Zur systematischen Ordnung und Beschreibung dieses Prozesses, in dem die eigentliche Funktionsschätzung nur ein Teilschritt ist, hat sich in Wissenschaft und Praxis das *Knowledge-Discovery-in-Databases*-Modell etabliert. Es findet sich in zahlreichen Lehrbüchern¹⁾ und Softwareprodukten²⁾. Ausgehend von der ursprünglichen Konzeption nach Fayyad, Piatetsky-Shapiro und Smyth³⁾ wurden zahlreiche erweiterte und abgeänderte Prozessmodelle vorgeschlagen, unter denen der *Cross-Industry Standard Process for Data Mining* (CRISP-DM) heute das beliebteste ist.⁴⁾ CRISP-DM wurde von einem Industriekonsortium um die heutige IBM-Tochter SPSS entwickelt und ist speziell auf den Einsatz in Betrieben ausgerichtet.

Der Begriff *Knowledge Discovery in Databases* (KDD) wurde schon 1989 auf einem interdisziplinären Workshop geprägt.⁵⁾ Das bekannte Prozessschema (siehe Abb. D-3) und eine allgemein akzeptierte Definition wurden aber erst 1996 von Fayyad, Piatetsky-Shapiro und Smyth veröffentlicht. Dort heißt es: „KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.“⁶⁾ Ein *pattern* (Muster) kann eine kompakte Darstellung der Daten oder ein Modell sein,

¹⁾ Vgl. z. B. Chamoni/Gluchowski [Informationssysteme] 281 ff.; Turban/Sharda/Delen [Decision Support] 207 ff.; Kantardzic [Data Mining] 7 ff.

²⁾ Implementierungen des Knowledge-Discovery-in-Databases-Prozesses gibt es beispielsweise in IBM SPSS (das Prozessmodell heißt hier *CRISP-DM*; vgl. Chapman u. a. [CRISP-DM]; Shearer [CRISP-DM]) und SAS Enterprise Miner (unter der Bezeichnung *SEMMA*; vgl. SAS Institute Inc. [Enterprise Miner] 5 f.).

³⁾ Vgl. Fayyad/Piatetsky-Shapiro/Smyth [Knowledge Discovery].

⁴⁾ Vgl. Turban/Sharda/Delen [Decision Support] 215; Kurgan/Musilek [Process Models] 5 ff.; Azevedo/Lourenço [KDD]; Marbán/Mariscal/Segovia [Process Model] 3 ff.; jeweils mit weiteren Nachweisen. Siehe auch Fußnote 2.

⁵⁾ Vgl. Piatetsky-Shapiro [Knowledge Discovery].

⁶⁾ Fayyad/Piatetsky-Shapiro/Smyth [Knowledge Discovery] 40 f. Für eine Diskussion dieser Definition vgl. Düsing [Knowledge Discovery] 283; Kurgan/Musilek [Process Models] 2 f.

das die vorliegenden Daten erklärt und neue zu prognostizieren hilft.¹⁾ Allgemein erlaubt die Mustererkennung eine Beschreibung der Daten auf einer abstrakteren Ebene. Mit *nichttrivial* ist gemeint, dass die Prozessschritte anspruchsvolle Suchverfahren und Berechnungen beinhalten und es z. B. nicht ausreicht, nur einen Mittelwert zu berechnen. Die gefundenen Muster sollen auch für neue Daten *gültig* und zumindest für das KDD-System *neu* sein, d. h. sie sollen ihm nicht schon bei der Entwicklung explizit eingegeben worden sein. Schließlich sollen die gefundenen Muster für den Anwender einen *Nutzen* haben und *verständlich* sein – wenn nicht unmittelbar, so doch nach einer weiteren Aufbereitung. Ebendiese Ziele werden auch mit der in dieser Arbeit erforschten Anwendung künstlicher neuronaler Netze angestrebt. Deshalb orientieren sich die weiteren Ausführungen am Knowledge-Discovery-in-Databases-Prozess.

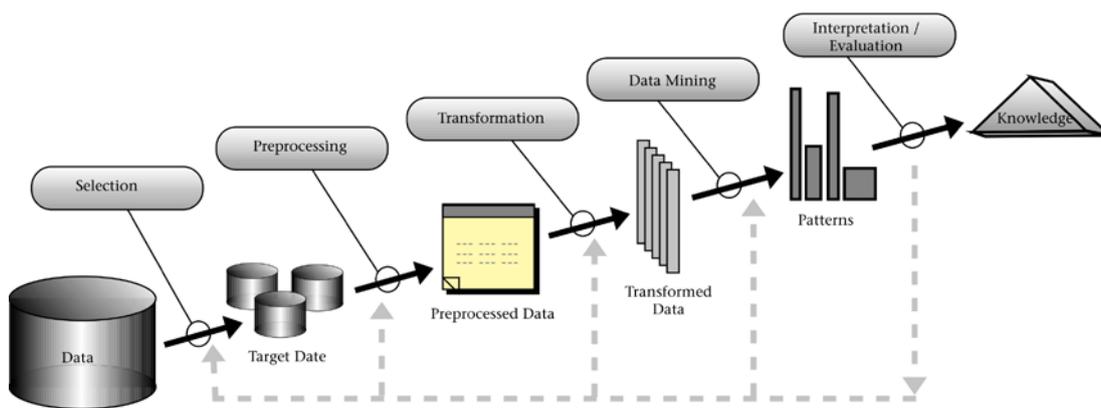


Abb. D-3: Stufen des Knowledge-Discovery-in-Databases-Prozesses²⁾

Fayyad, Piatetsky-Shapiro und Smyth grenzen ihren Ansatz von der „blinden“ Anwendung datengetriebener Analysemethoden ab, die ihrer Meinung nach zu Recht mit dem negativ konnotierten Begriff *data mining* belegt

¹⁾ Vgl. auch im Weiteren Fayyad/Piatetsky-Shapiro/Smyth [Knowledge Discovery] 41.

²⁾ Fayyad/Piatetsky-Shapiro/Smyth [Knowledge Discovery] 41.

ist.¹⁾ Die wissenschaftliche Abneigung gegen das *data mining* rührt daher, dass man in fast jedem Datensatz – selbst wenn er nur Zufallsdaten enthält – Muster entdecken kann, die statistisch signifikant erscheinen, wenn man nur lange genug sucht. Deshalb verlangen Fayyad, Piatetsky-Shapiro und Smyth, dass vor Beginn des *Knowledge Discovery in Databases* die Problemstellung und insbesondere die Bedürfnisse und Ziele des Anwenders klar und deutlich artikuliert werden, damit anschließend Daten und Analysemethoden passend zu diesen Zielen gewählt werden können. *Data mining* ist im Knowledge-Discovery-in-Databases-Prozess nur noch ein Teilschritt, in dem mit einer spezifischen Analysemethode Muster in einem definierten Datensatz gesucht werden.

Für ein *Knowledge Discovery in Databases* zur Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen sind die genannten Vorbedingungen erfüllt. Die Problemstellung wurde in den Kapiteln B (S. 6 ff.) und C.III (S. 130 ff.) gründlich erörtert und die Anwendungszwecke wurden in Kapitel B.IV (S. 55 ff.) ausführlich dargelegt. Mit diesem Vorwissen werden im ersten Schritt des Knowledge-Discovery-in-Databases-Prozesses geeignete Daten gesucht und vorselektiert (siehe Abb. D-3).²⁾ Im zweiten Schritt werden u. a. Ausreißer, fehlende Werte und Strukturbrüche in den Daten behandelt. Im dritten Schritt werden die Daten in eine für die Problemstellung und die gewählte Analysemethode geeignete Form gebracht. Gegebenenfalls kann in diesem Schritt auch die Dimension der Eingabedaten reduziert werden – etwa durch eine Hauptkomponentenanalyse. Im vierten Schritt wird eine adäquate Data-Mining-Methode – hier künstliche neuronale Netze – auf die vorverarbeiteten und transformierten Daten ange-

¹⁾ Vgl. auch im Weiteren Fayyad/Piatetsky-Shapiro/Smyth [Knowledge Discovery] 39 ff.

²⁾ Vgl. auch im Weiteren Fayyad/Piatetsky-Shapiro/Smyth [Knowledge Discovery] 42.

wandt. Die Ergebnisse werden im fünften Schritt ausgewertet, visualisiert und interpretiert. Vor allem dieser Schritt ist mit den vorhergehenden rückgekoppelt, da sich aus den Analyseergebnissen neue Erkenntnisse über geeignete Daten, notwendige Vorverarbeitungsschritte und die Ausgestaltung des Data-Mining-Algorithmus ergeben können. Aber auch zwischen und innerhalb der anderen Schritte gibt es vor- und rückgekoppelte Schleifen.

IV. Entwurf einer Software zur Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen

1. Notwendigkeit einer eigenständigen Software zur Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen

Bis hierher wurden eine eindeutige Problemstellung formuliert (es sollen Kostenfunktionen ermittelt werden, die für die in Kapitel B.IV (S. 55 ff.) beschriebenen Zwecke gut geeignet sind) und ein Lösungsweg grob vorgezeichnet (ein Knowledge-Discovery-in-Databases-Prozess mit künstlichen neuronalen Netzen an zentraler Stelle). Dabei könnte man es belassen, in der weiteren Dissertation geeignete Methoden für die einzelnen Schritte des KDD-Prozesses definieren und sie in einer der üblichen Statistik- oder Data-Mining-Softwareumgebungen, z. B. in R, Matlab, Octave, RapidMiner oder WEKA umsetzen. Diese Programme sind jedoch nicht einfach zu bedienen und setzen ein Expertenwissen voraus, das unter Kostenrechnern und anderen Betriebswirten nicht sehr verbreitet ist. Eine derartige Imple-

mentierung würde also mit dem Ziel der praktischen Anwendbarkeit in der Kostenrechnung (siehe S. 160) konfliktieren.

Um Kostenrechnern und betrieblichen Entscheidungsträgern das volle Potential künstlicher neuronaler Netze zur Schätzung von Kostenfunktionen zugänglich zu machen, bedarf es einer einfach zu bedienenden Software, die alle notwendigen Schritte des KDD-Prozesses in einer übersichtlichen Anwendungsumgebung integriert. Da derzeit keine verfügbare Software diesen Anforderungen genügt, wird im Folgenden eine eigene Lösung entwickelt – unterstützt von den Methoden der objektorientierten Softwareentwicklung.

2. Grundlagen des objektorientierten Softwareentwurfs

a) Prinzipien der Objektorientierung

Objektorientierung ist seit mehr als einem Vierteljahrhundert das vorherrschende Paradigma in der Softwareentwicklung. Die Idee dahinter ist, ein Softwaresystem so zu konzipieren, wie der Mensch seine Umwelt wahrnimmt – als Ansammlung klar abgegrenzter Objekte, die miteinander interagieren (so die vereinfachte Vorstellung).¹⁾ Ein Objekt im Sinne der Objektorientierung ist eine Einheit mit einem spezifischen Zustand und charakteristischen Operationen, die diesen Zustand abfragen und ihn ändern können.²⁾ Und tatsächlich nehmen wir im Alltag Gegenstände, Personen und

¹⁾ Vgl. Coad/Yourdon [Object-Oriented Analysis] 1 ff.; Jacobson u. a. [Software Engineering] 43.

²⁾ Vgl. Jacobson u. a. [Software Engineering] 45; Schlagheck [Referenzmodelle] 10 ff., mit weiteren Nachweisen.

andere Dinge (wie z. B. das Finanzamt) als Einheit aus Zustand und Operationen wahr: das Messer liegt in der Schublade, ist scharf, hat eine zehn Zentimeter lange Klinge etc.; verschiedene Operationen können an ihm oder mit ihm ausgeführt werden: es aus der Schublade nehmen, Gemüse schneiden, die Länge der Klinge messen usw.¹⁾ Bei einigen dieser Operationen ändert sich der Zustand des Objekts und bei mindestens einer Operation auch der eines anderen Objekts: das Messer und das Gemüse interagieren.

Am Anfang der objektorientierten Softwareentwicklung stehen objektorientierte Modelle des Problembereichs, auf den die Software angewendet werden soll. Wegen der alltäglichen Vertrautheit im Umgang mit Objekten ist die semantische Lücke zwischen einem objektorientierten Modell und der natürlichsprachlichen Beschreibung des Problems geringer als bei anderen Modellierungsprinzipien.²⁾ Objektorientierung ermöglicht eine hohe Strukturähnlichkeit (*Homomorphie*) zwischen dem Modell und dem was es abbildet.³⁾ Deshalb sind objektorientierte Modelle verhältnismäßig leicht zu verstehen und zu konzipieren.

Außerdem sorgen die Prinzipien der Datenkapselung, der Polymorphie und der Vererbung für eine geringe Komplexität des objektorientierten Systems und machen es übersichtlich, weniger redundant, einfacher erweiterbar und leichter zu testen als ein nicht objektorientiertes System.⁴⁾ Mit *Datenkapselung* ist gemeint, dass nur das Objekt selbst seinen Zustand ablesen

¹⁾ Vgl. Strohhecker [Simulation] 133.

²⁾ Vgl. auch im Weiteren Jacobson u. a. [Software Engineering] 44.

³⁾ Vgl. auch im Weiteren Strohhecker [Simulation] 134.

⁴⁾ Vgl. auch im Weiteren Coad/Yourdon [Object-Oriented Analysis] 14 ff.; Jacobson u. a. [Software Engineering] 48 ff.; Zimmermann [Geschäftsprozessmanagement] 23 ff.; Lahres/Rayman [Programmierung] 27 ff.

und ändern kann. Es kann aber seinen Zustand anderen Objekten mitteilen, wenn sie danach fragen, und andere Objekte können durch einen Operationsaufruf den inneren Zustand eines Objekts verändern. Das Objekt weiß aber immer, wer gerade mit welcher Operation auf seinen Zustand zugreifen will, und kann deshalb leicht die Konsistenz seines Zustands sicherstellen.

Polymorphie bedeutet, dass unterschiedliche Objekte den gleichen Operationsaufruf unterschiedlich interpretieren können.¹⁾ Dieses Prinzip ist wie die Datenkapselung ein *information hiding*. Denn der Aufrufer einer Operation weiß nicht – und muss es auch nicht wissen – wie ein Objekt seinen Aufruf umsetzt. Er muss nur wissen, dass das Objekt diese Operation unterstützt. Polymorphie macht es möglich, einzelne Komponenten des Systems ohne großen Aufwand auszutauschen. Das wird z. B. in Teilkapitel IX genutzt, um verschiedene Trainingsalgorithmen zu implementieren. Ein Trainingsalgorithmus ist ein Objekt, das u. a. die Operation `startTraining()` unterstützt. Ein Aufrufer muss nicht wissen, ob mit dem Aufruf der Operation `startTraining()` ein Backpropagation-, Resilient-Propagation- oder Extreme-Learning-Algorithmus gestartet wird, solange er am Ende ein trainiertes Netz zurückbekommt. Polymorphie führt also zu einem flexiblen, modular erweiterbaren Programm.

Schließlich ist noch das Prinzip der *Vererbung* zu nennen. Jedes Objekt lässt sich einer oder mehreren Klassen (Kategorien) zuordnen.²⁾ Genauer gesagt ist jedes Objekt eine Instanz einer Klasse. Peter und Paul sind Instanzen der

¹⁾ Vgl. auch im Weiteren Jacobson u. a. [Software Engineering] 56 f.; Zimmermann [Geschäftsprozessmanagement] 23; Lahres/Rayman [Programmierung] 183 ff.

²⁾ Vgl. auch im Weiteren Coad/Yourdon [Object-Oriented Analysis] 12 ff.; Jacobson u. a. [Software Engineering] 50 ff.; Zimmermann [Geschäftsprozessmanagement] 23 ff.; Lahres/Rayman [Programmierung] 84 ff.

Klasse Mann, Marie ist eine Instanz der Klasse Frau. Außerdem sind Peter und Marie Erwerbstätige, Marie ist zudem Akademikerin und Peter und Paul sind Europäer. Alle drei sind Menschen, Säugetiere und Lebewesen. Jede Unterklasse *erbt* die Eigenschaften und Operationen der übergeordneten Klasse. Das hat den Vorteil, dass gemeinsame Eigenschaften und Operationen nur einmal beschrieben werden müssen. Wenn beispielsweise bereits eine Klasse Mensch existiert, müssen in den Klassen Mann und Frau nur noch die Unterschiede formuliert werden. Ein objektorientiertes System ist deswegen kaum redundant. Außerdem hängt das Prinzip der Vererbung eng mit der Polymorphie zusammen. Beispielsweise erben die Klassen Backpropagation, Resilient-Propagation und Extreme Learning Machine die (abstrakte) Operation `startTraining()` von der Klasse Trainingsalgorithmus.

b) Objektorientierte Modellierung mit der *unified modeling language*

Die *unified modeling language* (UML) ist eine international standardisierte graphische Sprache für die Visualisierung, Spezifizierung, Konstruktion und Analyse objektorientierter Modelle.¹⁾ Sie wird nicht nur für die Modellierung von Softwaresystemen, sondern z. B. auch zur Modellierung von Geschäftsprozessen verwendet.²⁾ In der Version 2.4 umfasst sie vierzehn Diagrammtypen, von denen die eine Hälfte die Struktur und die andere Hälfte das dynamische Verhalten des Systems beschreibt.³⁾ In der Softwareentwicklung begleiten die Diagramme der UML den gesamten Prozess von der Anforderungsanalyse, über den Systementwurf bis zur detaillierten

¹⁾ Siehe ISO [UML Infrastructure]; ISO [UML Superstructure].

²⁾ Vgl. ausführlich Zimmermann [Geschäftsprozessmanagement].

³⁾ Vgl. ISO [UML Superstructure] 700 f.

Beschreibung der Implementierung.¹⁾ Spezielle Anwendungsprogramme erlauben sogar die automatische Übersetzung einiger UML-Diagramme in Programmcode.

3. Erfassung der Systemanforderungen in einem Anwendungsfalldiagramm

Am Anfang der objektorientierten Softwareentwicklung mit der *unified modeling language* steht die Erfassung der funktionalen Anforderungen an das Softwaresystem in einem Anwendungsfalldiagramm (*use case diagram*). Anwendungsfalldiagramme wurden erstmals 1992 von Ivar Jacobson als Grundlage der objektorientierten Softwareentwicklung beschrieben²⁾ und von ihm wenige Jahre später in die UML eingebracht.³⁾

Ein Anwendungsfalldiagramm enthält Akteure, Anwendungsfälle und ein System (siehe Abb. D-4).⁴⁾ Die *Akteure* stehen außerhalb des Systems und interagieren mit ihm. Jeder Akteur repräsentiert eine Rolle, die ein Nutzer oder ein anderes Computersystem bei der Interaktion mit dem System einnimmt. Mehrere Rollen können von ein und demselben Nutzer wahrgenommen werden, wie z. B. beim Monopoly-Spiel ein Spieler zugleich auch die Bank ist.⁵⁾

¹⁾ Vgl. ISO [UML Infrastructure] x; Seemann/von Gudenberg [Software-Entwurf] 147 ff.

²⁾ Vgl. Jacobson u. a. [Software Engineering].

³⁾ Vgl. Object Management Group [UML]; vgl. auch Seemann/von Gudenberg [Software-Entwurf] 2 f.

⁴⁾ Vgl. auch im Weiteren Jacobson u. a. [Software Engineering] 152 ff.; ISO [UML Superstructure] 603 ff.

⁵⁾ Vgl. Seemann/von Gudenberg [Software-Entwurf] 18 f.

Ein *Anwendungsfall* beschreibt eine in sich abgeschlossene Folge von Ereignissen, die im System stattfindet.¹⁾ Er wird in der Regel von einem Akteur angestoßen. In jedem Fall ist das Ergebnis des Anwendungsfalls vom Akteur beobachtbar und hat für ihn einen Wert. Ein Anwendungsfall beschreibt also eine nützliche Funktion, die das System für den Akteur erfüllt.

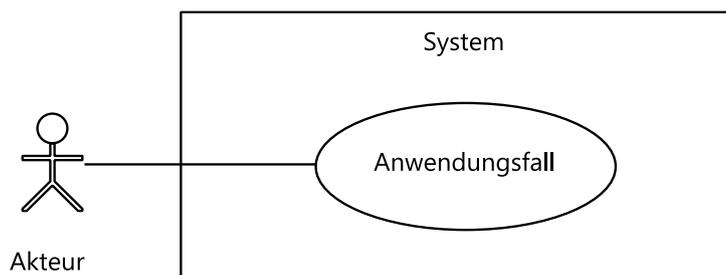


Abb. D-4: Elemente eines Anwendungsfalldiagramms

Das *System* besteht aus der Menge seiner Anwendungsfälle. Sie beschreiben seinen Funktionsumfang.²⁾ Damit ist das Anwendungsfalldiagramm, wenn es am Beginn des Systementwurfs steht, ein Instrument, um die Anforderungen an das System möglichst präzise und umfänglich zu erfassen. Außerdem können Anwendungsfalldiagramme am Ende der Systementwicklung dazu genutzt werden, die Funktionen des fertigen Systems zu dokumentieren. Während der Entwicklung erleichtern Anwendungsfalldiagramme die Aufteilung des Systems in Module, die Abgrenzung geeigneter Objekte³⁾ und die Definition von Testfällen.

¹⁾ Vgl. auch im Weiteren Jacobson u. a. [Software Engineering] 154 ff.; ISO [UML Superstructure] 612; Seemann/von Gudenberg [Software-Entwurf] 16 f.

²⁾ Vgl. auch im Weiteren Jacobson u. a. [Software Engineering] 130 ff., 151 f.; ISO [UML Superstructure] 612; Seemann/von Gudenberg [Software-Entwurf] 16 f.

³⁾ Nach Jacobson *ist* ein Anwendungsfall bereits ein Objekt (vgl. Jacobson u. a. [Software Engineering] 130).

Das Anwendungsfalldiagramm in Abb. D-5 fasst die bisher in den Kapiteln B, C und D erörterten Anforderungen an ein System zur Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen zusammen. Auf Details wurde zugunsten einer zusammenhängenden Darstellung des Gesamtsystems verzichtet. Alle UML-Diagramme in dieser Arbeit sind wie der Programmcode, den sie dokumentieren, durchgängig in Englisch gehalten, u. a. um nicht die englischen Schlüsselworte mit deutschen Inhalten zu vermengen. Bei der Beschreibung der Diagramme lässt sich diese Vermischung allerdings nicht ganz vermeiden.

In Abb. D-5 werden die Nutzerrollen Modellierer und Kostenrechner unterschieden. Der Modellierer ist mit den ersten vier Schritten des Knowledge-Discovery-in-Databases-Prozesses (siehe Abb. D-3, S. 171) befasst. Er wählt Daten aus, unterzieht sie einer Vorverarbeitung, um Ausreißer zu beseitigen und fehlende Werte zu ergänzen, transformiert sie in eine analyserechte Form und analysiert sie mit einer Data-Mining-Methode. Die ersten drei Schritte werden in den Teilkapiteln D.V, D.VI und D.VII näher beleuchtet. Der vierte Schritt ist am kompliziertesten und deshalb schon an dieser Stelle detaillierter aufgeführt. Als passende Data-Mining-Methode wurde in Kapitel C für künstliche neuronale Netze argumentiert. Ein darauf basierendes Schätzmodell besteht im Wesentlichen aus einem künstlichen neuronalen Netz (D.VIII) und einem Trainingsalgorithmus (D.IX).¹⁾ Dabei sollte eine Überanpassung an die Trainingsdaten vermieden werden (D.XI).²⁾ Eine Metaheuristik kann helfen, dies systematisch sicherzustellen (D.XII).

¹⁾ Siehe auch Abschnitt C.I.1, S. 83 ff., und Abschnitt C.I.3, S. 92 ff.

²⁾ Siehe auch Abschnitt C.I.2, S. 88 ff.

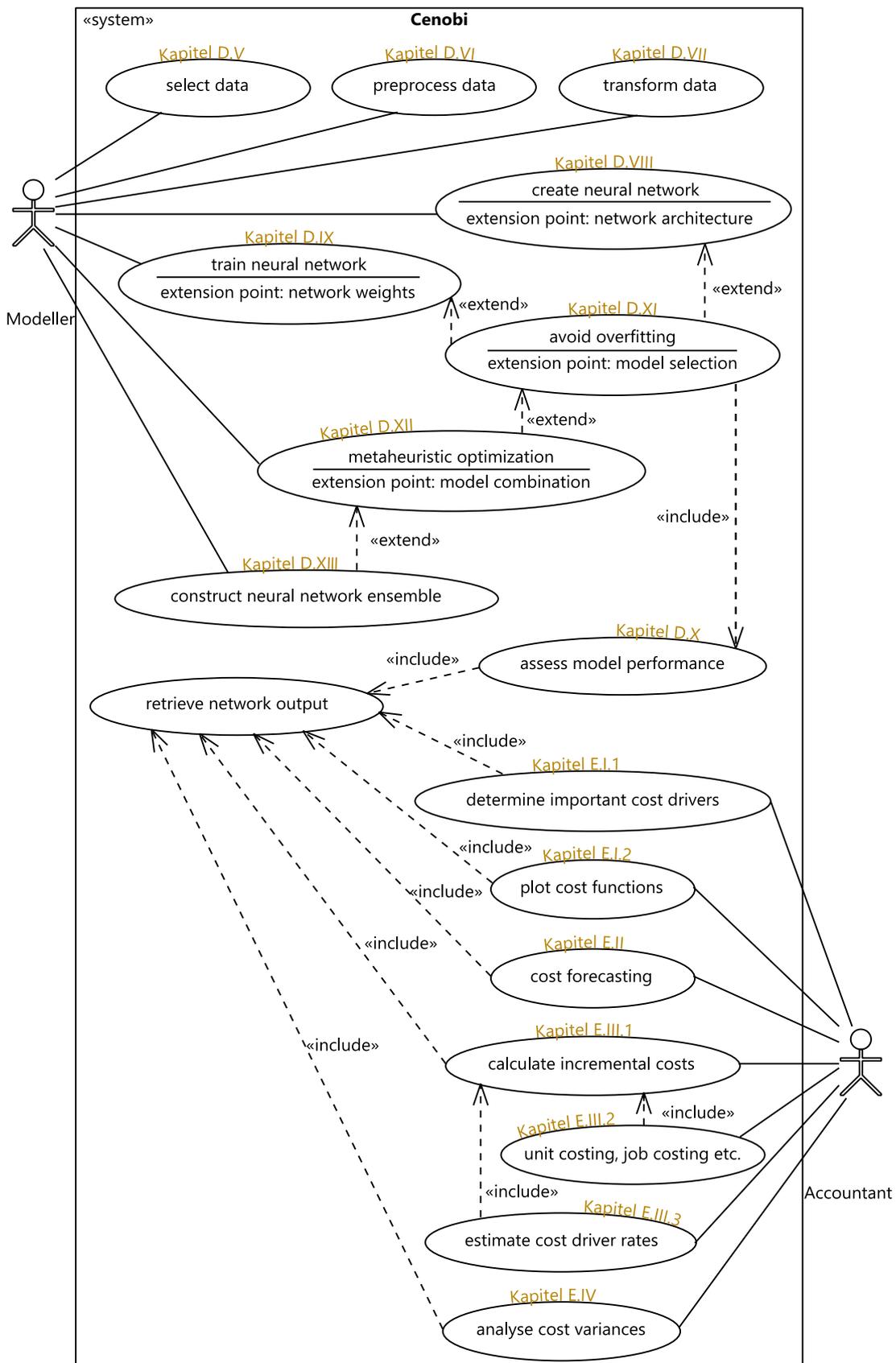


Abb. D-5: Anwendungsfalldiagramm für ein System zur Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen

In der UML-Notation werden Erweiterungen von Anwendungsfällen durch das Schlüsselwort «extend» und einen gestrichelten Pfeil vom erweiternden zum erweiterten Anwendungsfall gekennzeichnet.¹⁾ Außerdem zeigt ein extension point an, an welcher Stelle der erweiternde Anwendungsfall eingreift. Der erweiterte Anwendungsfall soll auch ohne die Erweiterung vollständig und in sich abgeschlossen sein, wohingegen der erweiternde Anwendungsfall in der Regel nur ergänzenden Charakter hat und nicht unabhängig sein muss. Im Gegensatz zur «extend»-Beziehung, die eine *mögliche* Erweiterung darstellt, drückt die «include»-Beziehung aus, dass ein Anwendungsfall einen anderen *immer* beinhaltet, weil er z. B. auf dessen Ergebnis – seinen Rückgabewert – angewiesen ist.²⁾

In Abb. D-5 beinhaltet z. B. die Vermeidung von *overfitting* die Bewertung der Generalisierungsfähigkeit eines Modells. Welche Gütemaße für diese Bewertung geeignet sind, wird in Teilkapitel D.X untersucht. Zuletzt gibt es aus Sicht des Modellierers noch den Anwendungsfall, dass nicht nur ein künstliches neuronales Netz parametrisiert wird, sondern ein ganzes Ensemble künstlicher neuronaler Netze (D.XIII). Dabei ist es sinnvoll, das Ensemble nicht zufällig zusammenzustellen, sondern auch hier Optimierungsalgorithmen zu nutzen. In der einfachsten Ausführung wird eine bestehende Heuristik dahingehend erweitert, dass am Ende nicht ein bestes Modell, sondern mehrere gute Modelle behalten werden.

Der Kostenrechner möchte mit dem System kostenrechnerische Aufgaben erfüllen. Grundsätzlich können dabei alle Anwendungsfälle von Kostenfunktionen für Kostenrechnungszwecke eintreten, die in Kapitel B.IV (S. 55 ff.) beschrieben wurden. Für diese Dissertation werden einige typische

¹⁾ Vgl. auch im Weiteren ISO [UML Superstructure] 607 ff.

²⁾ Vgl. ISO [UML Superstructure] 611.

Anwendungsfälle aus verschiedenen Phasen des Führungsprozesses¹⁾ herausgegriffen, um an ihnen die Fähigkeiten künstlicher neuronaler Netze in der Kostenrechnung zu demonstrieren. Dazu gehören die Identifikation relevanter Kosteneinflussgrößen („Kostentreiber“) und die Analyse des Funktionsgraphen. Beide Anwendungsfälle dienen der Problemstrukturierung²⁾ und werden in Kapitel E.I näher erläutert. In der Kostenrechnungspraxis besonders häufige Anwendungsfälle sind die Kostenprognose³⁾ (E.II) und die Berechnung der Grenzkosten einer Entscheidung⁴⁾ (E.III.1). Die Berechnung der Grenzkosten dient in erster Linie der Alternativenbewertung.⁵⁾ Eine besonders oft wiederkehrende spezielle Alternativenbewertung ist die Kalkulation von Produktstücken, Aufträgen etc. Bei der Kalkulation können entweder direkt die zusätzlichen Kosten berechnet werden, die entstehen, wenn zusätzlich zum bestehenden Produktionsniveau ein weiteres Produktstück bzw. ein weiterer Auftrag produziert und abgesetzt wird (E.III.2), oder es kann indirekt eine Bezugsgrößenkalkulation⁶⁾ mit festen Kostensätzen je Bezugsgrößeneinheit erfolgen (E.III.3). Beide Varianten beinhalten eine Grenzkostenberechnung (siehe Abb. D–5). Für die Bezugsgrößenkalkulation wird jeder Bezugsgröße (Einflussgröße) ein konstanter Kostensatz (*cost driver rate*)⁷⁾ zugeordnet, der die Kosten enthält, die entstehen, wenn eine zusätzliche Einheit der Bezugsgröße anfällt. Die Kostensätze je Einflussgröße können außerdem in der Phase der Alternativensuche dazu

¹⁾ Siehe Abschnitt B.II.1, S. 11 ff.

²⁾ Siehe Abschnitt B.IV.2, S. 57 ff.

³⁾ Siehe Abschnitt B.IV.4, S. 60 f.

⁴⁾ Siehe Abschnitt B.III.3, S. 29 ff.

⁵⁾ Siehe Abschnitt B.IV.5, S. 62 ff.

⁶⁾ Siehe Abschnitt B.III.4.b, S. 33 ff.

⁷⁾ Siehe Abschnitt B.III.4.c, S. 37 ff.

genutzt werden, kostengünstige Alternativen zu finden.¹⁾ Schließlich ist im Anwendungsfalldiagramm in Abb. D-5 mit der Abweichungsanalyse (E.IV) auch die Kontrollphase des Führungsprozesses vertreten. Zahlreiche weitere kostenrechnerische Anwendungsfälle sind denkbar und können den Funktionsumfang des Systems erweitern.

Alle kostenrechnerischen Anwendungsfälle greifen direkt oder indirekt auf ein trainiertes neuronales Netz zu, übergeben ihm einen oder mehrere Inputvektoren und rufen den damit berechneten Netzoutput ab (siehe Abb. D-5). Ein passendes Netz muss zuvor vom Modellierer erzeugt und mit Vergangenheitsdaten parametrisiert worden sein. Der Modellierungsprozess soll in Cenobi aber so weit automatisiert werden, dass der Modellierer keine ausgeprägten statistischen Fachkenntnisse besitzen muss. Dann kann der Kostenrechner die Modellierungsaufgaben übernehmen, so dass die Nutzerrollen Modellierer und Kostenrechner von derselben Person wahrgenommen werden.

4. Ableitung der Programmstruktur aus dem Anwendungsfalldiagramm

Die *unified modeling language* und einige Programmiersprachen – darunter Java – erlauben es, Klassen²⁾ und andere Systemelemente in Paketen anzuordnen. Auch Pakete selbst können wiederum in einem Paket enthalten sein. Ein Element kann immer nur einem einzigen Paket direkt „gehören“. So ergibt sich eine strikte Baumstruktur, wie man sie z. B. von den Dateisystemen moderner Betriebssysteme kennt.

¹⁾ Siehe Abschnitt B.IV.3, S. 59 f.

²⁾ Siehe Abschnitt D.IV.2.b, S. 174 ff.

Aus programmiertechnischer Sicht ist das Paket der Namensraum seiner Elemente.¹⁾ D. h. ein Element ist über seine Paketzugehörigkeit eindeutig identifizierbar, auch wenn es denselben einfachen Namen wie ein anderes Element trägt. Das ist zum einen nützlich, um z. B. die Klasse Parser im Paket Excel von der Klasse Parser im Paket Textfile zu unterscheiden. Vor allem aber wird das System durch Pakete übersichtlicher. Das liegt nicht zuletzt daran, dass ein Element, das außerhalb des eigenen Pakets liegt, explizit importiert werden muss, um darauf zugreifen zu können. Dadurch treten die Abhängigkeiten zwischen den Paketen klar hervor. Es gibt immer eine gerichtete Beziehung vom importierenden zum importierten Paket. Die Darstellung des gerichteten Graphen, der sich daraus ergibt, in einem Paketdiagramm verschafft einen guten Überblick über die Abhängigkeitsstruktur des Systems. Das erleichtert u. a. die Definition von Schnittstellen zur Modularisierung.

In einer guten Paketstruktur haben die Pakete nachvollziehbar abgegrenzte Funktionsbereiche und zugleich möglichst wenige Abhängigkeitsbeziehungen zu anderen Paketen. Eine Paketstruktur für Cenobi, die diese Bedingungen gut erfüllt, lässt sich aus dem Anwendungsfalldiagramm (Abb. D-5) ableiten. Dort wird deutlich, dass die zentralen Modellierungs-Anwendungsfälle (D.VIII bis D.XIII) vielfältig untereinander verbunden sind, während sie mit den kostenrechnerischen Anwendungsfällen nur den gemeinsamen Anwendungsfall retrieve network output haben. Demnach ist es sinnvoll, zwei unabhängige Pakete für Modellierung und Kostenrechnung zu definieren, wobei das Kostenrechnungspaket über eine Schnittstelle den Output eines trainierten Netzes abrufen kann. Die Anwendungsfälle Datenselektion, -vorverarbeitung und -transformation stehen mit den anderen

¹⁾ Vgl. ISO [UML Infrastructure] 163.

in keiner direkten Beziehung und werden in einem dritten Paket zusammengefasst. Das Paketdiagramm in Abb. D–6 zeigt diese Struktur mit weiteren Unterpaketen und einigen Klassen, die beispielhaft den Funktionsumfang der Pakete illustrieren.

In einem Paketdiagramm werden die Pakete als abstrahierte Hängeordner mit einem linksbündigen Reiter dargestellt (siehe Abb. D–6). Die Art der im Paket enthaltenen Elemente kann durch Symbole gekennzeichnet werden. So sind in Abb. D–6 Klassen mit dem Symbol  gekennzeichnet. Das Schlüsselwort «import» neben einem Pfeil zeigt an, dass das Paket am Pfeilanfang Elemente aus dem Paket an der Pfeilspitze importiert, ohne dass genauer spezifiziert wird, welche Elemente das im Einzelnen sind. Um das Diagramm übersichtlich zu halten, wurde in Abb. D–6 darauf verzichtet, direkte Importe explizit darzustellen, wenn sich eine Paketabhängigkeit bereits transitiv ergibt. Beispielsweise importiert das Kostenrechnungspaket direkt die Klasse VariableSet aus dem Datenverarbeitungspaket. Eine Abhängigkeit ergibt sich aber schon indirekt über das Modellierungspaket. Ein direkter «import»-Pfeil vom Kostenrechnungs- zum Datenverarbeitungspaket würde deshalb kaum zusätzliche Erkenntnisse über die Abhängigkeitsstruktur des Gesamtsystems bringen.

Die Unterpakete des Modellierungspakets und die Klassen des Kostenrechnungspakets spiegeln fast unverändert die Anwendungsfälle aus Abb. D–5 und ihre Beziehungen wider. Für das Datenverarbeitungspaket gilt das nicht in gleichem Maße, da einige Teilfunktionen der Anwendungsfälle in andere Pakete eingehen. Die Datenselektion etwa findet größtenteils über die graphische Benutzeroberfläche statt, die in Abb. D–6 nicht repräsentiert ist.

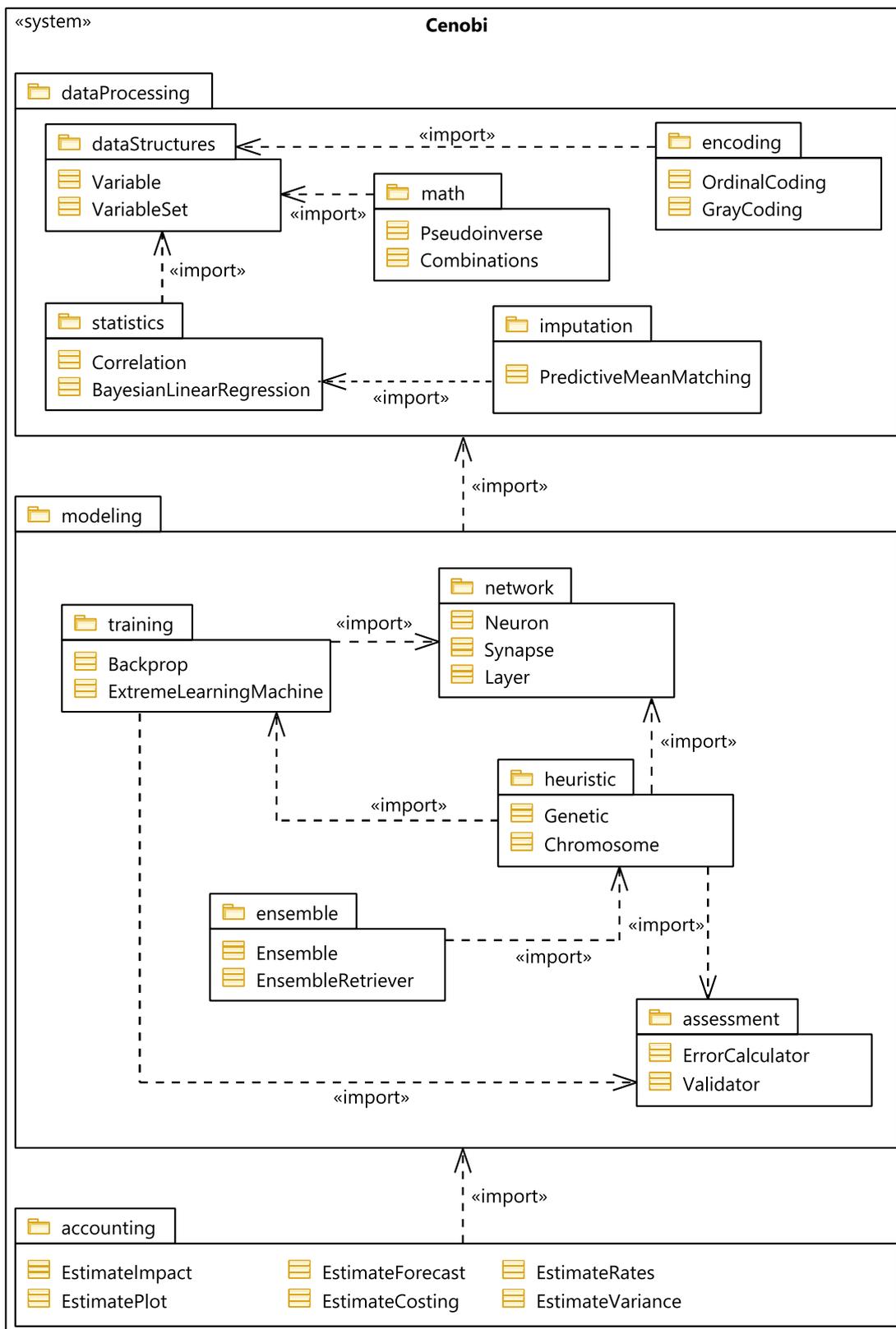


Abb. D-6: Paketdiagramm für ein objektorientiertes Programm zur Schätzung von Kostenfunktionen mit künstlichen neuronalen Netzen

Außerdem wird die Transformation der Daten zum Teil von den Eingabeneuronen des künstlichen neuronalen Netzes übernommen.¹⁾ Dafür kommt das Unterpaket `dataStructures` hinzu, das sich nicht unmittelbar aus einem Anwendungsfall ergibt. Es stellt grundlegende Datenstrukturen zur Verfügung, mit denen Daten, die zu einer Variablen (einer Einflussgröße oder einer Kostengröße) gehören, in ein einheitliches Format gebracht werden. Fast alle anderen Pakete nutzen diese Datenstrukturen. Deshalb ist dieses Paket die Senke des Abhängigkeitsgraphen. Abschließend ist noch anzumerken, dass der Graph zyklensfrei ist. Zyklische Abhängigkeiten können bei der Compilierung zu Problemen führen.

5. Gestaltung der Benutzeroberfläche

Die in den Anwendungsfall- und Paketdiagrammen entwickelte Struktur spiegelt sich an der Benutzeroberfläche wider (siehe Abb. D-7). Die Kostenrechnungsfunktionen sind prominent auf der linken Seite des Hauptfensters platziert und deutlich von den Datenvorverarbeitungsfunktionen auf der rechten Seite getrennt. Da die Modellierungsaufgaben für den typischen Nutzer nicht im Vordergrund stehen, sind sie nicht direkt über das Hauptfenster sondern nur über ein Einstellungsmenü erreichbar. Mit den Grundeinstellungen können alle kostenrechnerischen Analysen zufriedenstellend durchgeführt werden, so dass der Nutzer die Modellierungseinstellungen nicht unbedingt ändern muss.

Der Nutzer wird mit farbigen Akzenten und Ausblendungen durch den Knowledge-Discovery-in-Databases-Prozess²⁾ geführt. In einer neuen Ceno-

¹⁾ Siehe Abschnitt VIII.1, S. 237 ff.

²⁾ Siehe Teilkapitel III, S. 169 ff.

bi-Datei ist die Schaltfläche für den Datenimport hervorgehoben, während die anderen Schaltflächen ausgeblendet sind (siehe Abb. D–8).

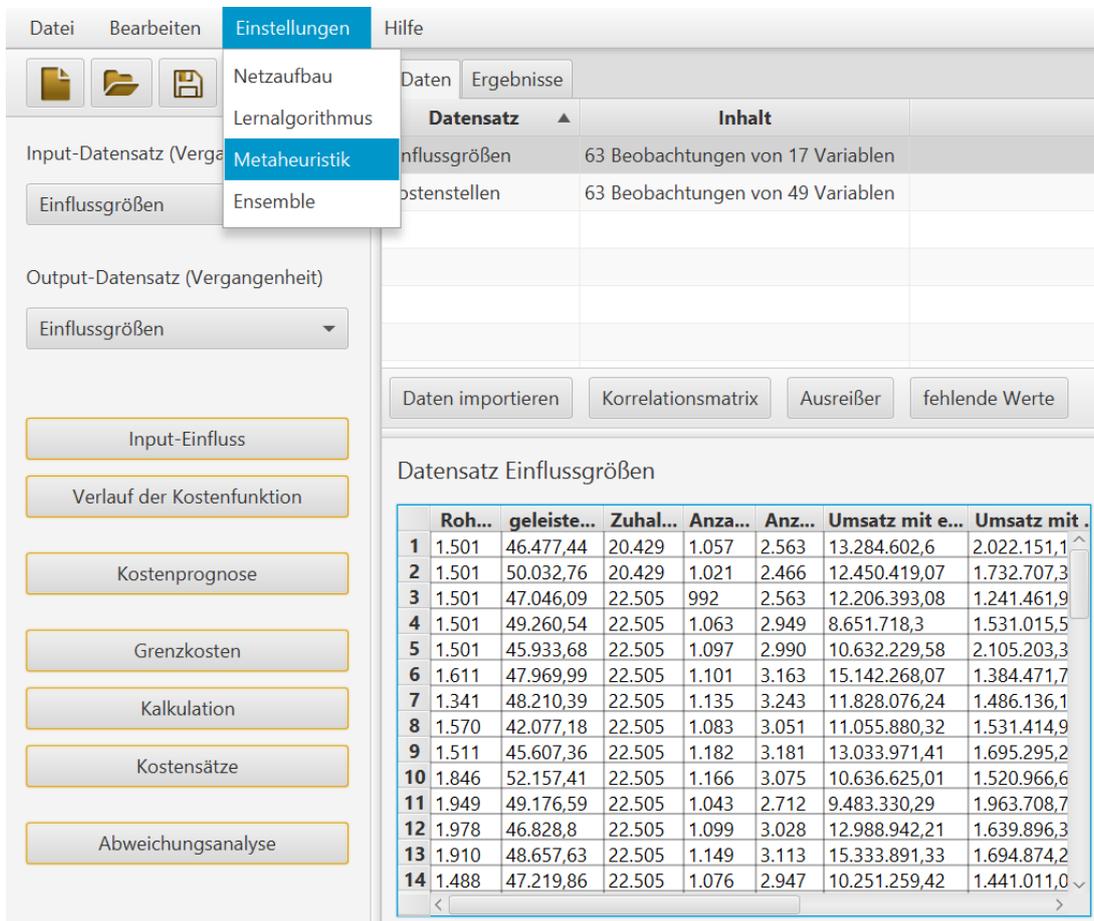


Abb. D–7: Cenobi-Benutzeroberfläche

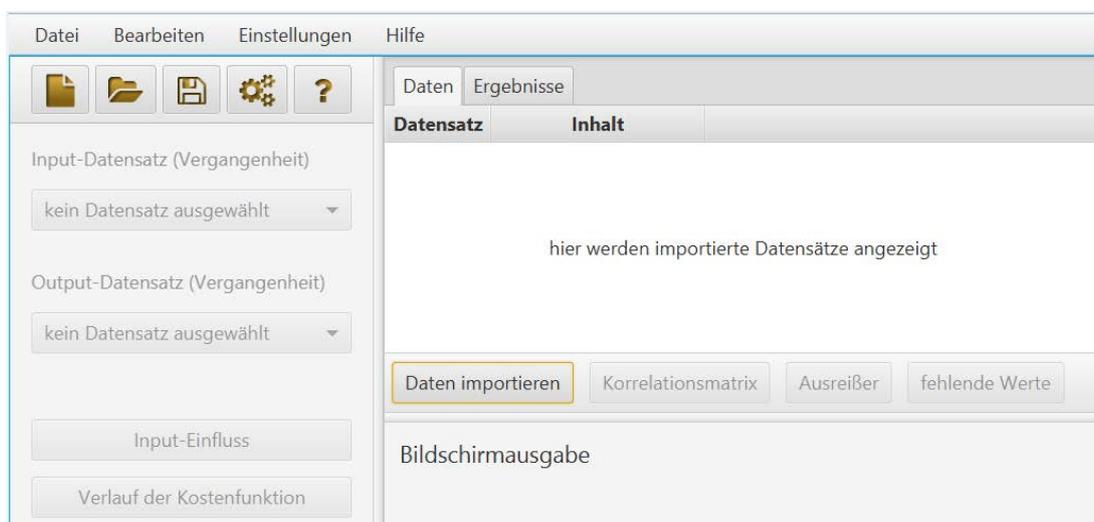


Abb. D–8: Benutzerführung in Cenobi – Datenimport

Nach dem Datenimport leuchten unterhalb der Datensatz-Tabelle die Schaltflächen für Datenselektion und -vorverarbeitung auf, mit denen der Nutzer den jeweils angewählten Datensatz bearbeiten kann (siehe Abb. D–9). Eine Vorverarbeitung der Datensätze wird vom Nutzer nicht zwingend verlangt; er kann einen Datensatz auch direkt über die Auswahlkästen auf der linken Seite den Analysemethoden übergeben.

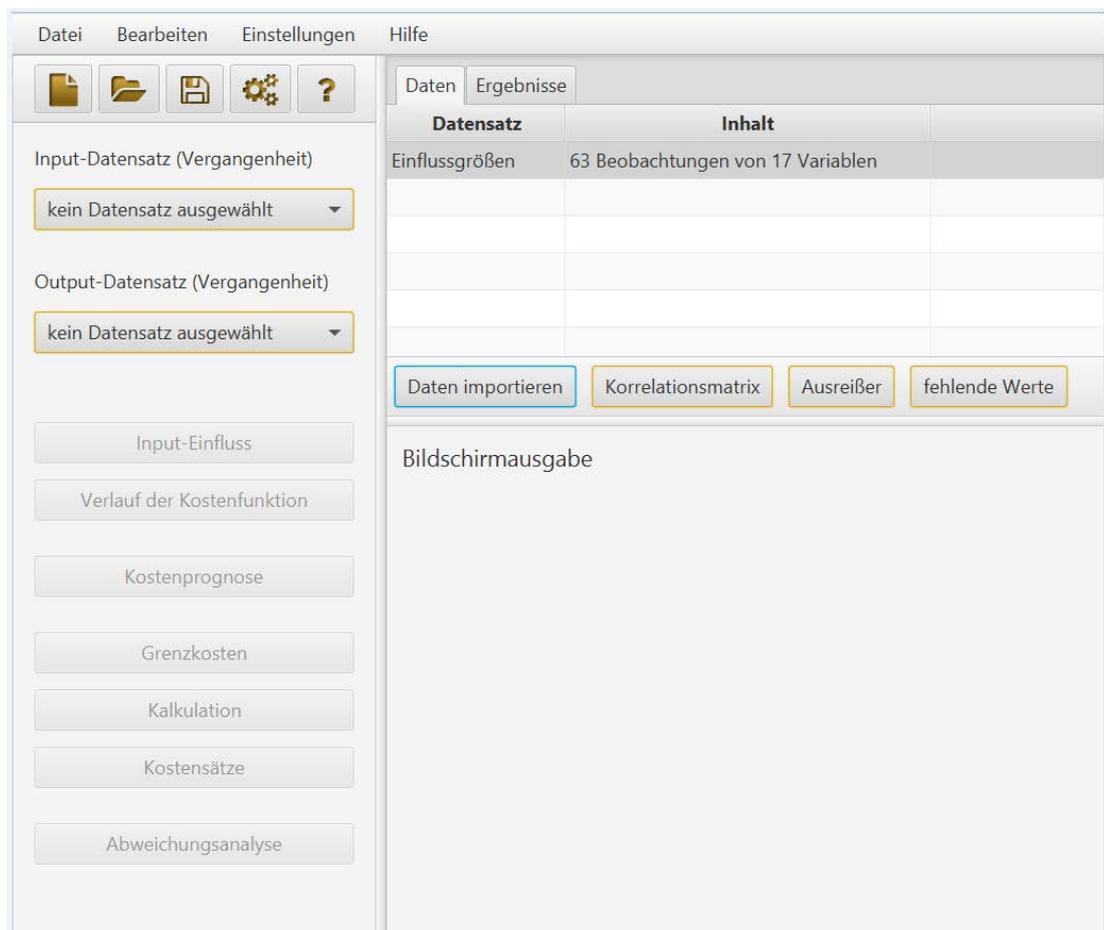


Abb. D–9: Benutzerführung in Cenobi – Datenvorverarbeitung

Wenn in den entsprechenden Auswahlkästen Vergangenheitsdaten der Einflussgrößen und Kostenhöhen ausgewählt sind, stehen die kostenrechnerischen Analysefunktionen zur Verfügung (siehe Abb. D–10). Ein neuronales Netz wird beim Betätigen einer Analyseschaltfläche automatisch im Hintergrund erstellt, trainiert, optimiert, ausgewertet und gespeichert. Details

über diese Analyseschritte erfährt der Nutzer nur, wenn er die Einstellungsmenüs für Netzaufbau, Lernalgorithmus, Metaheuristik und Ensemblebildung aufruft. Ein einmal trainiertes neuronales Netz kann für alle Kostenrechnungsaufgaben ausgewertet werden. Deshalb wird es gespeichert und bei jeder Analyse wieder aufgerufen, wenn dieselben Vergangenheitsdaten und Einstellungen ausgewählt sind.

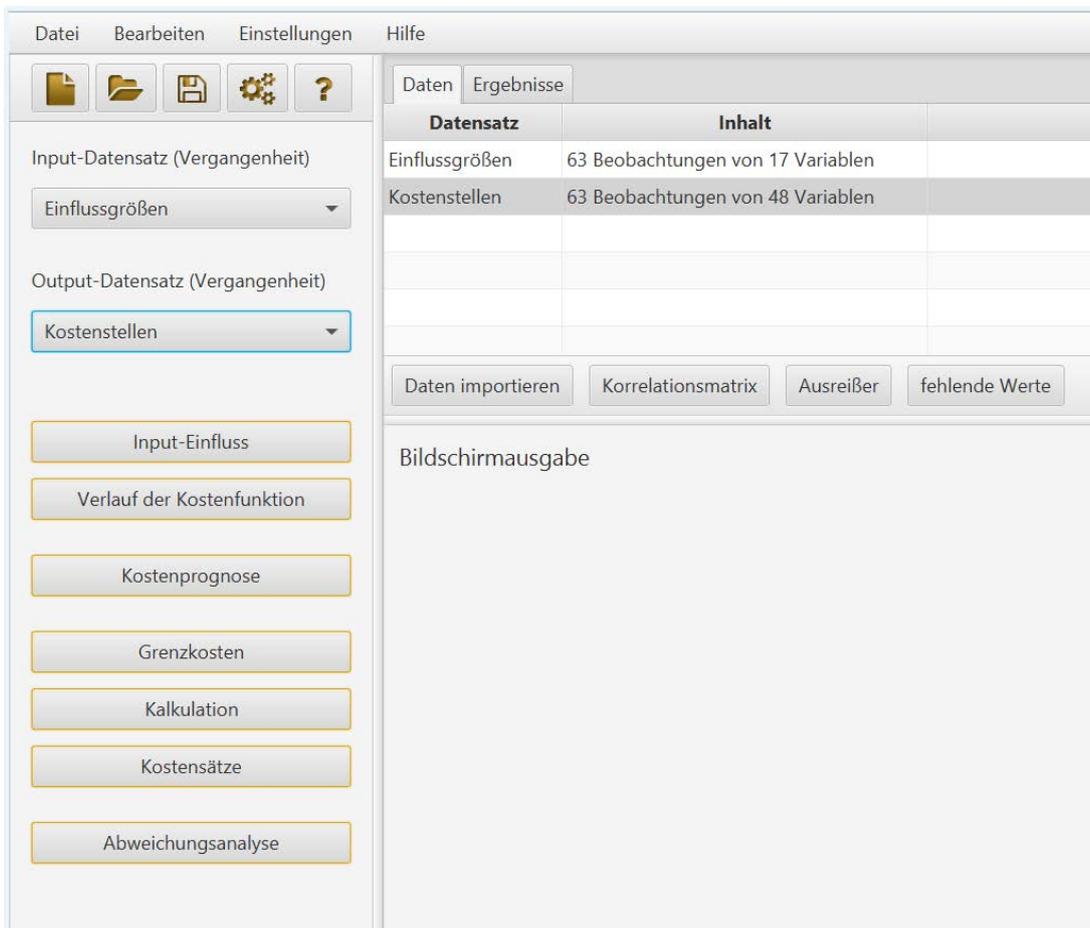


Abb. D-10: Benutzerführung in Cenobi – Kostenrechnung

V. Auswahl der unabhängigen Variablen

1. Kriterien zur Auswahl von Kosteneinflussgrößen

Eine Kostenfunktion soll die Einflussmöglichkeiten auf die Kostenhöhe abbilden (siehe S. 13 f.). Da sich die Kostenhöhe in der Regel nicht nur auf eine Weise beeinflussen lässt, sind dafür üblicherweise mehrere Einflussgrößen notwendig.¹⁾ Neben den direkt beeinflussbaren Entscheidungsvariablen müssen zum Teil auch nicht beeinflussbare Umweltvariable in die Kostenfunktion aufgenommen werden, um die Kosteneinflussmöglichkeiten adäquat abzubilden.²⁾

Eine systematische Übersicht in der wissenschaftlichen Literatur vorgeschlagener und in der Kostenrechnungspraxis verwendeter Einflussgrößen gibt Abschnitt B.III.5 (S. 40 ff.). Insbesondere die in Abb. B-6 (S. 48), Abb. B-7 (S. 49) und Abb. B-8 (S. 52) aufgeführten *cost driver* sind i. A. nicht schwer zu erfassen.³⁾ Ihre Daten sind in vielen Betrieben wenn nicht in einem zentralen Informationssystem, so doch in verschiedenen Fachsystemen bereits gespeichert.

Grundsätzlich kann man annehmen, dass die Abbildungsgenauigkeit der Kostenfunktion mit der Anzahl der berücksichtigten, Einflussgrößen steigt. Mit der Anzahl der unabhängigen Variablen nimmt aber auch der Aufwand der Datenbeschaffung zu und die Kostenfunktion wird für den Entscheidungsträger weniger übersichtlich. In der Literatur werden einige Ent-

¹⁾ Siehe Abschnitt B.III.6, S. 52 ff.

²⁾ Siehe Abschnitt C.III.3.a, S. 134 ff.

³⁾ Vgl. Cooper [Activity-Based Costing 2] 277.

scheidungsmodelle vorgeschlagen, die unter Berücksichtigung dieses Zielkonflikts die Einflussgrößenauswahl erleichtern sollen.¹⁾ In dieser Arbeit wird in Abschnitt E.I.1 (S. 300 ff.) eine Methode vorgestellt, die misst, wie stark eine unabhängige Variable die Kostenhöhe beeinflusst. Die Ergebnisse können genutzt werden, um Variablen mit einem geringen Einfluss auszusortieren. Dieses Vorgehen kann in einen explorativen Selektionsprozess eingebunden werden, in dem aus einer anfänglich größeren Variablenmenge – möglicherweise inspiriert von den Übersichten in Abschnitt B.III.5 – sukzessive die relevanten Variablen ausgewählt werden. Dabei sollte allerdings auch in der Startauswahl bereits sichergestellt sein, dass die unabhängigen Variablen nicht zu stark miteinander korrelieren.

2. Prüfung auf Korrelation zwischen den unabhängigen Variablen

Zwischen den unabhängigen Variablen, die in ein statistisches Modell eingehen, sollte kein systematischer Zusammenhang bestehen. Den positiven Aspekt dieses Prinzips hat Kurt Rummel in Form seines Leitsatzes zur „Austauschbarkeit der Maßgrößen“ in der Kostenrechnung verankert: „Man kann den Maßstab für irgendeine Größe durch einen anderen Maßstab ersetzen, wenn die Maßstäbe untereinander proportional sind.“²⁾

Bei statistischen Analysen gilt indes nicht nur, dass von zwei korrelierten Variablen eine vernachlässigt werden *kann*; für ein gutes Modell ist es sogar *erforderlich*, dass die unabhängigen Variablen unkorreliert sind. Denn korrelierte Eingabevariablen erschweren die Interpretation der mit ihnen be-

¹⁾ Vgl. Babad/Balachandran [Optimization]; Homburg/Zimmer [Auswahl]; Homburg [Selection].

²⁾ Rummel [Kostenrechnung] 5.

rechneten Funktionen und verringern die Zuverlässigkeit der geschätzten Parameter.¹⁾ Wenn beispielsweise in einem linearen Regressionsmodell $y = b_0 + b_1x_1 + b_2x_2$ der Regressor x_1 linear von x_2 abhängig ist, d. h. $x_1 = a + bx_2$, – man spricht in diesem Fall von *perfekter Multikollinearität* – lässt sich der Einfluss von x_1 und x_2 auf y nicht eindeutig einer Variablen zuordnen. Das Schätzproblem der Regressionskoeffizienten b_1 und b_2 hat in diesem Fall keine eindeutige Lösung, wodurch die Varianz der Regressionskoeffizienten zunimmt. Sie können nicht signifikant erscheinen, obwohl die gemeinsame Ursache, die x_1 und x_2 zugrundeliegt, einen signifikanten Einfluss auf y hat. Deshalb sollte nur eine der beiden Variablen in das Modell aufgenommen werden. Die gleichen Probleme ergeben sich auch in nicht-linearen Modellen wie z. B. künstlichen neuronalen Netzen.

Um einen ersten Überblick über bestehende Korrelationen zu erhalten, bietet es sich an, eine *Korrelationsmatrix* zu erstellen, in der paarweise Zusammenhänge zwischen den Variablen dargestellt sind. Für jedes Variablenpaar (x_i, x_j) wird ein Korrelationskoeffizient r_{ij} berechnet:

$$r_{ij} = \frac{s_{x_i x_j}}{s_{x_i} s_{x_j}} = \frac{\sum_{n=1}^N (x_i^{(n)} - \bar{x}_i)(x_j^{(n)} - \bar{x}_j)}{\sqrt{\sum_{n=1}^N (x_i^{(n)} - \bar{x}_i)^2 \sum_{n=1}^N (x_j^{(n)} - \bar{x}_j)^2}} \quad (\text{D.1})$$

mit \bar{x}_i

Mittelwert von x_i

$$s_{x_i} = \sqrt{\frac{\sum_{n=1}^N (x_i^{(n)} - \bar{x}_i)^2}{N - 1}}$$

empirische Standardabweichung

$$s_{x_i x_j} = \frac{\sum_{n=1}^N (x_i^{(n)} - \bar{x}_i)(x_j^{(n)} - \bar{x}_j)}{N - 1}$$

empirische Kovarianz.²⁾

¹⁾ Vgl. auch im Weiteren Hedderich/Sachs [Statistik] 702; Backhaus u. a. [Analysemethoden] 93 f.

²⁾ Vgl. auch im Weiteren Hedderich/Sachs [Statistik] 109 f., 673 f.

Der Korrelationskoeffizient r_{ij} ist eine Maßzahl für die Stärke des Zusammenhangs zwischen zwei Variablen. Er ist unabhängig von der Variablen-dimension und kann Werte zwischen 1 und -1 annehmen. Bei $r > 0$ ist der Zusammenhang positiv, bei $r < 0$ ist er negativ.

Setzt man für x_i und x_j kardinal gemessene Beobachtungswerte ein, erhält man den *Produkt-Moment-Korrelationskoeffizienten* nach *Bravais* und *Pearson*. Er misst die Stärke der linearen Abhängigkeit zwischen normalverteilten, kardinalskalierten Merkmalen. Ein Wert von $r = \pm 1$ bedeutet in diesem Fall, dass alle Punkte $(x_i^{(n)}, x_j^{(n)})$ auf einer Geraden liegen. Problematisch ist, dass der Bravais-Pearson-Koeffizient nur lineare Zusammenhänge abbildet und empfindlich auf die Häufigkeitsverteilung der Daten, insbesondere auf Ausreißer reagiert.¹⁾

Ein robuster Korrelationskoeffizient, der auch ordinalskalierte Daten zulässt und nicht-lineare Zusammenhänge erfasst, ist der *Rangkorrelationskoeffizient* r^s nach *Spearman*.²⁾ Kardinale Daten x werden zunächst durch die Zuordnung von Rangzahlen \check{x} auf eine Ordinalskala abgebildet (siehe Abb. D-11). Der Rangkorrelationskoeffizient r^s ist der Korrelationskoeffizient r (D.1) aus den Rangzahlen \check{x} . Ein Wert von $r_{ij}^s = 1$ ergibt sich, wenn die Werte von x_i und x_j in derselben Rangfolge angeordnet sind. Das bedeutet, dass x_i ansteigt, wenn auch x_j zunimmt. Fällt dagegen x_i immer, wenn x_j steigt, ist $r_{ij}^s = -1$. Dabei wird die Höhe des Anstiegs nicht berücksichtigt. Ein Wert von $r^s = \pm 1$ kann sich demnach auch dann ergeben, wenn die Rangabstände nicht proportional sind – der Zusammenhang also nicht linear ist. Spearman's r^s erfasst allerdings nur monotone Abhängigkeiten, wie z. B. in einer s-förmigen Funktion, bei denen sich das Vorzeichen des Zusammenhangs

¹⁾ Vgl. Hedderich/Sachs [Statistik] 110.

²⁾ Vgl. auch im Weiteren Hedderich/Sachs [Statistik] 110 ff.

nicht ändert. U-förmige Korrelationen werden beispielsweise nicht erkannt.¹⁾

geleistete Std. Spritzgussmaschinen		Anzahl eingesetzter Rohmaterialtypen	
x_i	Rang \check{x}_i	x_j	Rang \check{x}_j
46.477,44	9	59	11
50.032,76	2	57	12
47.046,09	7	66	9
49.260,54	3	69	5,5
45.933,68	10	69	5,5
47.969,99	6	73	4
48.210,39	5	74	2,5
42.077,18	12	67	7,5
45.607,36	11	67	7,5
52.157,41	1	78	1
49.176,59	4	74	2,5
46.828,80	8	62	10

Abb. D–11: Zuordnung von Rangzahlen zu kardinalen Daten für die Berechnung des Rangkorrelationskoeffizienten

Wenn Korrelationen von mehr als zwei Variablen untersucht werden, bietet sich für den paarweisen Vergleich eine *Korrelationsmatrix* \mathbf{R} an. Ihre Berechnung lässt sich durch folgenden Zusammenhang vereinfachen:

$$r_{ij} = \frac{s_{x_i x_j}}{s_{x_i} s_{x_j}} = \frac{1}{N-1} \sum_{n=1}^N \frac{(x_i^{(n)} - \bar{x}_i)(x_j^{(n)} - \bar{x}_j)}{s_{x_i} s_{x_j}} = \frac{1}{N-1} \sum_{n=1}^N z_i^{(n)} z_j^{(n)} \quad (\text{D.2})$$

mit $z_i^{(n)} = \frac{x_i^{(n)} - \bar{x}_i}{s_{x_i}}$ standardisierter Wert (z-Transformation, z score) von $x_i^{(n)}$.²⁾

¹⁾ Vgl. Hedderich/Sachs [Statistik] 674.

²⁾ Vgl. auch im Weiteren Hedderich/Sachs [Statistik] 353; Backhaus u. a. [Analysemethoden] 338.

In Matrizenscheibweise ergibt sich

$$R = \frac{1}{N - 1} Z^T Z \tag{D.3}$$

mit $R \in \mathbb{R}^{I \times I}$ Korrelationsmatrix

$Z \in \mathbb{R}^{N \times I}$ Matrix mit den Elementen $z_i^{(n)}$.

Damit können sowohl Bravais-Pearson-, als auch Spearman-Korrelationskoeffizienten berechnet werden. Abb. D-12 zeigt die Rangkorrelationsmatrix für die unabhängigen Variablen des Beispieldatensatzes (siehe Anhang A1).

	Ro...	gel...	Zuh...	Anz...	Anz...	Um...	Um...	Um...	Anz...	Anz...	Anz...	Anz...	Anz...	Anz...	Fre...	Fre...	...
Rohmaterialverbrauch in t	1																
geleistete Std. Spritzgussma...	0,65	1															
Zuhaltekraft ...	0,44	0,57	1														
Anzahl verkaufter Produktarten	0,27	0,42	0,66	1													
Anzahl verkaufter Varianten	0,28	0,52	0,84	0,85	1												
Umsatz mit eigener Produktion in...	-0,18	-0,24	-0,28	0,07	-0,02	1											
Umsatz mit Handelsware ...	-0,02	-0,1	-0,43	-0,28	-0,38	0,17	1										
Umsatz aus Rohmaterialv...	0,06	0,22	-0,19	-0,3	-0,3	0,16	0,3	1									
Anzahl eingesetzter Rohmaterialt...	-0,15	-0,27	-0,41	-0,35	-0,45	-0,04	0,27	0,18	1								
Anzahl Kreditoren ...	0,26	0,37	0,53	0,26	0,35	-0,17	-0,28	0,03	-0,14	1							
Anzahl Einkaufsrech...	0,31	0,55	0,49	0,37	0,47	-0,02	-0,17	0,26	0,05	0,43	1						
Anzahl Rechnungen ...	0,32	0,2	0,43	0,58	0,53	0,1	-0,12	-0,19	-0,17	0,3	0,18	1					
Anzahl Warenempfä...	0,22	0,26	0,26	0,26	0,31	-0,08	0,01	-0,13	-0,22	0,29	0,06	0,56	1				
Anzahl Lagerbewegu...	0,42	0,59	0,85	0,7	0,81	-0,24	-0,41	-0,29	-0,48	0,48	0,39	0,38	0,36	1			
Fremdleistung Produktion in...	-0,22	-0,08	-0,51	-0,41	-0,49	0,28	0,42	0,73	0,3	-0,23	0,06	-0,28	-0,24	-0,61	1		
Fremdleistung Verpackung/...	0,15	0,42	0,54	0,42	0,58	-0,1	-0,32	-0,11	-0,38	0,35	0,38	0,17	0,07	0,53	-0,24	1	
Fremdl. Entw...	-0,25	-0,13	-0,07	-0,18	-0,08	0,14	0,1	0,05	-0,04	0,14	-0,18	-0,01	0,25	-0,06	0,03	-0,15	1

Abb. D-12: Matrix mit Rangkorrelationskoeffizienten für die unabhängigen Variablen des Beispieldatensatzes

Da R symmetrisch ist, wird nur die untere Dreiecksmatrix gezeigt. In der Literatur wird als Faustregel angegeben, dass die Korrelationskoeffizienten zwischen unabhängigen Variablen nicht größer als 0,7 sein sollten.¹⁾ Deshalb sind in Abb. D–12 alle Korrelationseffizienten mit einem Absolutwert über 0,7 rot eingefärbt. Die höchste Korrelation besteht zwischen der Anzahl der Produktarten und der Anzahl der (Farb-)Varianten. Gründe dafür wurden bereits in Abschnitt C.I.3.c (S. 138 ff.) erörtert. Wenn nicht gerade eine Entscheidung über die Einführung einer zusätzlichen Variante ansteht, ist es besser, die hierarchisch höhere Einflussgröße – also die Anzahl der Produktarten – im Modell zu behalten, um eine Schlüsselung der produktartbezogenen Kosten auf die Varianten zu vermeiden (siehe S. 140). Außerdem ist die Variantenzahl hoch mit der Zuhaltekraft und der Anzahl der Lagerbewegungen korreliert, so dass gleich drei problematische Multikollinearitäten verschwinden, wenn die Variantenzahl aus dem Modell herausgenommen wird. Zu beachten ist allerdings, dass nun in den Kosten für eine Produktart auch die Kosten für eine durchschnittliche Anzahl an Varianten dieser Produktart enthalten sind (siehe S. 140).

Für die hohe Korrelation zwischen Zuhaltekraft und Lagerbewegungen ist vermutlich eine dritte, nicht im Datensatz enthaltene Variable verantwortlich. Wenn eine neue Maschine angeschafft wird, steigt zum einen die Gesamtzuhaltekraft und zum anderen können mehr Produkte gefertigt werden, was zu mehr Lagerbewegungen führt. Die Multikollinearität kann leicht vermieden werden, wenn die Anzahl der Lagerbewegungen nur in Kostenfunktionen für Lagerkostenstellen und die Zuhaltekraft nur in Kostenfunktionen für Fertigungskostenstellen eingeht. Dieses Beispiel zeigt, dass es nicht sinnvoll ist, in jeder Kostenstelle alle Einflussgrößen zu ver-

¹⁾ Vgl. z. B. Horngren/Datar/Rajan [Cost Accounting] 396.

wenden. Am besten wird für jede Kostenstelle ein eigenes Schätzmodell mit einer sinnvollen Vorauswahl möglicher Einflussgrößen aufgestellt, das dann individuell verfeinert werden kann.

Multiple Korrelationen zwischen einer unabhängigen Variablen und einer Kombination aus den anderen unabhängigen Variablen lassen sich mit den bisher genannten Korrelationsmaßen nicht erkennen. Sie können aufgedeckt werden, indem jede unabhängige Variable x_i durch ein Regressionsmodell $x_i = f(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_I)$ der übrigen unabhängigen Variablen abgebildet wird. Wenn die Regression gut ist, also z. B. ein hohes Bestimmtheitsmaß R_i^2 aufweist, korreliert die betreffende Variable stark mit den anderen und kann demnach bei weiteren Analysen vernachlässigt werden. Häufig wird dabei der Varianzinflationsfaktor (*variance inflation factor*) einer Variablen x_i

$$VIF_i = \frac{1}{1 - R_i^2} \quad (\text{D.4})$$

berechnet, der seinen Namen daher hat, dass die Varianz des Regressionskoeffizienten von x_i mit steigendem R_i^2 um eben diesen Faktor zunimmt.¹⁾ Für den Varianzinflationsfaktor werden in der Literatur unterschiedliche Grenzwerte empfohlen.²⁾

¹⁾ Vgl. Backhaus u. a. [Analysemethoden] 95, mit weiteren Nachweisen.

²⁾ Vgl. Hedderich/Sachs [Statistik] 703; kritisch O'Brien [Variance Inflation Factors].

VI. Vorverarbeitung der Daten

1. Umgang mit Ausreißern

a) Problematik der Ausreißerererkennung

Ein Ausreißer ist eine Beobachtung, die so stark von den anderen Beobachtungen abweicht, dass der Verdacht aufkommt, sie könnte von einem anderen Zusammenhang verursacht worden sein als die übrigen Beobachtungen.¹⁾ In einigen Anwendungen sucht man nach Ausreißern, weil man sich gerade für diese Beobachtungen interessiert, z. B. um Betrugsfälle zu erkennen. Bei der Datenvorverarbeitung für Regressionsmodelle sucht man hingegen nach Ausreißern, die einer anderen Grundgesamtheit entstammen als die restliche Stichprobe und dadurch die Schätzung der Modellparameter verzerren, um sie aus dem Modellierungsdatensatz zu entfernen.²⁾ Solche Ausreißer werden vor allem durch Fehler bei der Datenerfassung und -übertragung verursacht. In der Kostenrechnung können z. B. Fehlbuchungen die Schätzung von Kostenfunktionen verzerren.³⁾ Je umfangreicher ein Datensatz ist, desto schwieriger wird es, Ausreißer durch „scharfes Hinsehen“ aufzuspüren. Statistische Methoden können die Suche erleichtern.

Eine sehr einfache statistische Ausreißeranalyse besteht darin, Werte die außerhalb des Vier-Sigma-Bereichs ($\mu \pm 4\sigma$) einer Variablen liegen als Aus-

¹⁾ Vgl. Hawkins [Outliers].

²⁾ Vgl. Hedderich/Sachs [Statistik] 427.

³⁾ Siehe Teilkapitel II, S. 163 ff.

reißer zu kennzeichnen.¹⁾ Diese Methode ist auch als *z-Wert-Test* bekannt, weil der standardisierte Wert

$$z_i^{(n)} = \frac{x_i^{(n)} - \bar{x}_i}{s_{x_i}} \quad (\text{D.5})$$

die Anzahl der empirischen Standardabweichungen s_{x_i} (siehe (D.1), S. 194) einer Beobachtung $x_i^{(n)}$ vom Mittelwert \bar{x}_i misst (siehe auch (D.2), S. 196).²⁾ Der z-Wert ist also ein standardisiertes Maß dafür, wie stark eine Beobachtung vom Mittelwert abweicht. Nach der Vier-Sigma-Regel gelten Beobachtungen mit $|z_i^{(n)}| \geq 4$ als Ausreißer.

Ein in der Literatur häufig genannter Nachteil dieses und ähnlicher Tests ist die Tatsache, dass sie die Dimensionen eines Datensatzes separat betrachten. Dadurch finden sie keine Ausreißer wie z. B. den in Abb. D–13 dargestellten, der weder aus dem typischen Wertebereich der Variablen x noch aus dem der Variablen y ausbricht, aber in seiner Kombination beider Dimensionen außergewöhnlich ist.³⁾

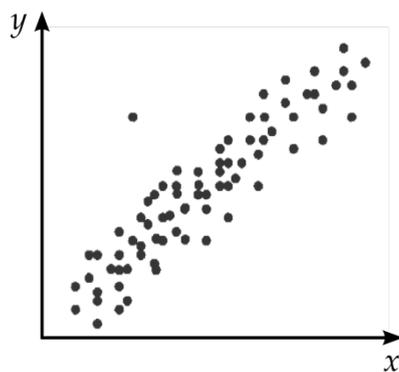


Abb. D–13: Zweidimensionaler Datensatz mit Ausreißer⁴⁾

¹⁾ Vgl. Hedderich/Sachs [Statistik] 428. Auch zwei (Kantardzic [Data Mining] 44) oder drei Sigma (Aggarwal [Outlier Analysis] 8) werden als Grenzwert genannt.

²⁾ Vgl. Aggarwal [Outlier Analysis] 7 f.

³⁾ Vgl. Kantardzic [Data Mining] 44 f.; Aggarwal [Outlier Analysis] 8 f.

⁴⁾ Kantardzic [Data Mining] 43.

Ebenfalls problematisch ist, dass der statistische Ausreißertest (D.5) normalverteilte Beobachtungen voraussetzt.¹⁾ Ein Großteil der Variablen des Beispieldatensatzes²⁾ erfüllt diese Voraussetzung jedoch nicht. Eine andere charakteristische Verteilung ist genauso wenig auszumachen. Allgemein gibt es auch keinen Grund, in der Kostenrechnung eine spezifische Verteilung zu erwarten. Deshalb sollte zur Ausreißerererkennung eine Methode gewählt werden, die keine bestimmte Verteilung voraussetzt.

Eine einfach zu implementierende Ausreißeranalyse, die keine spezifische Verteilung voraussetzt und alle D Dimensionen eines Datensatzes berücksichtigt, besteht darin, die D -dimensionalen paarweisen Abstände³⁾ zwischen allen Datenpunkten zu berechnen und als Ausreißermaß für einen Punkt $x^{(n)}$ beispielsweise zu berechnen, wie viele andere Punkte $x^{(m)}$ ($m \neq n$) weiter als eine Schwellenwertdistanz \tilde{d} von ihm entfernt sind.⁴⁾ Allerdings ist diese Methode sehr rechenintensiv, weil sie unter dem *curse of dimensionality*⁵⁾ leidet. Einen Ausweg bieten – neben zahlreichen anderen Methoden⁶⁾ – auch hier wieder künstliche neuronale Netze,⁷⁾ die, wie der folgende Abschnitt zeigt, mit einer speziellen Netzarchitektur zur Ausreißerererkennung genutzt werden können.

¹⁾ Vgl. Hedderich/Sachs [Statistik] 428; Aggarwal [Outlier Analysis] 8. Für weitere verteilungsbasierte Tests zur Ausreißerererkennung vgl. Hedderich/Sachs [Statistik] 428 ff.; Aggarwal [Outlier Analysis] 41 ff.

²⁾ Siehe Teilkapitel II, S. 163 ff., sowie Anhang A1 und A2.

³⁾ Siehe Formel (C.21), S. 106.

⁴⁾ Vgl. auch im Weiteren Hodge/Austin [Outlier Detection] 94 f.; Kantardzic [Data Mining] 45 ff.; Aggarwal [Outlier Analysis] 108 ff.

⁵⁾ Siehe S. 151 f. Vgl. auch Bishop [Machine Learning] 34 ff.; Aggarwal [Outlier Analysis] 112, 135.

⁶⁾ Vgl. die umfassenden Übersichten bei Hodge/Austin [Outlier Detection] 95 ff. und Aggarwal [Outlier Analysis] 112 ff.

⁷⁾ Siehe auch S. 152.

b) Erkennen von Ausreißern mit autoassoziativen künstlichen neuronalen Netzen

Ein autoassoziatives künstliches neuronales Netz (*autoassociative neural network*) ist ein Feedforward-Netz mit drei verdeckten Schichten. Von einem typischen *multilayer perceptron* (siehe Abb. C-4, S. 88) unterscheidet es sich nur dadurch, dass Eingabe- und Ausgabeschicht dieselben Variablen enthalten (siehe Abb. D-14). Das Netz wird auf eine identische Abbildung trainiert, d. h. es soll die Eingabewerte in der Ausgabeschicht identisch reproduzieren.¹⁾ Deshalb wird es auch als *Replikationsnetz* (*replicator neural network*) bezeichnet.²⁾ Die mittlere verdeckte Schicht enthält weniger Neuronen als die Eingabe- und Ausgabeschichten. Dieser „Flaschenhals“ zwingt das neuronale Netz dazu, den Eingabevektor auf einen wenigerdimensionalen Raum abzubilden, aus dem die darauffolgenden Schichten wieder die Eingabewerte rekonstruieren. Durch das Training mit identischen Werten in Eingabe- und Ausgabeschicht soll erreicht werden, dass bei dieser Dimensionsreduktion möglichst wenig Informationen verloren gehen.

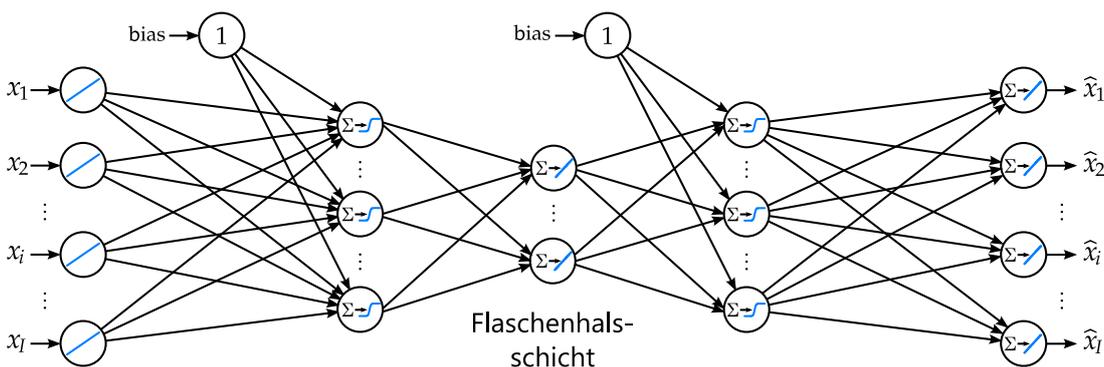


Abb. D-14: Aufbau eines autoassoziativen künstlichen neuronalen Netzes

¹⁾ Vgl. auch im Weiteren Kramer [Autoassociative Neural Networks] 233.

²⁾ Vgl. Hawkins u. a. [Replicator Neural Networks].

Autoassoziative neuronale Netze werden hauptsächlich für nichtlineare Hauptkomponentenanalysen (*nonlinear principal component analysis, NLPCA*) genutzt.¹⁾ Die Idee dahinter ist, dass ein neuronales Netz, das die Eingabewerte trotz des Flaschenhalses gut reproduzieren kann, im Flaschenhals eine gute dimensionsreduzierte Repräsentation der Inputvariablen enthalten muss. Sigmoidale Aktivierungsfunktionen in der ersten und dritten verdeckten Schicht sorgen dafür, dass eine nichtlineare Abbildung in den wenigerdimensionalen Raum möglich ist (siehe Abb. D–14).²⁾ Die Flaschenhalschicht muss hingegen keine sigmoidalen Aktivierungsfunktionen enthalten.

Der gleiche Aufbau wird auch zur Ausreißerererkennung verwendet.³⁾ Dabei macht man sich zunutze, dass das autoassoziative Netz während des Trainings implizit ein (dimensionsreduziertes) Modell des Datensatzes erstellt, das möglichst viele Beobachtungen gut beschreibt. Mit diesem Modell kann das trainierte Netz die meisten Beobachtungen gut komprimieren und anschließend wieder rekonstruieren. Oft gibt es aber einige wenige Beobachtungen, die es nicht gut rekonstruieren kann. Diese Beobachtungen lassen sich folglich nur schlecht durch denselben Modellzusammenhang wie die übrigen Daten beschreiben. Sie sind demnach Ausreißer. Als Maß für die Stärke des Ausreißens dient der absolute Rekonstruktionsfehler

$$E_i^{(n)} = |\hat{x}_i^{(n)} - x_i^{(n)}| \quad (\text{D.6})$$

mit dem Netzausgabewert $\hat{x}_i^{(n)}$.

¹⁾ Vgl. Kramer [Autoassociative Neural Networks]; Hsieh [Nonlinear Principal Component Analysis].

²⁾ Vgl. auch im Weiteren Kramer [Autoassociative Neural Networks] 235 f.

³⁾ Vgl. Markou/Singh [Novelty Detection] 2510 ff., mit weiteren Nachweisen; vgl. auch im Weiteren Hawkins u. a. [Replicator Neural Networks].

Ein autoassoziatives neuronales Netz kann mit einem normalen Backpropagation-Algorithmus¹⁾ trainiert werden und unterscheidet sich bei der Implementierung²⁾ nicht von einem normalen *multilayer perceptron*. In Cenobi ist es die Standardmethode zur Ausreißerererkennung.

Die Besonderheiten der Ausreißerererkennung mit autoassoziativen neuronalen Netzen lassen sich am besten an einem artifiziellen Testdatensatz demonstrieren. Dafür wurde ein autoassoziatives neuronales Netz mit fünf Neuronen in der ersten und dritten verdeckten Schicht sowie einem Neuron als Flaschenhals mit einem Backpropagation-Algorithmus auf die identische Abbildung der in Abb. D–15 dargestellten Werte trainiert.

	Variable 1	Variable 2	Variable 3	Variable 4
1	3	1	1	1
2	1	4	1	1
3	1	1	0	1
4	10	10	10	10
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
9	1	1	1	1
10	1	1	1	1
11	1	1	1	1
12	1	1	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1
16	1	1	1	1
17	1	1	1	1
18	1	1	1	1
19	1	1	1	1
20	1	1	1	1
21	1	1	1	1
22	1	1	1	1
23	1	1	1	1
24	1	1	1	1
25	1	1	1	1
26	1	1	1	1
27	1	1	1	1
28	1	1	1	1
29	1	1	1	1
30	1	1	1	1
31	1	1	1	1
32	1	1	1	1
33	1	1	1	1
34	1	1	1	1
35	1	1	1	1
36	1	1	1	1
37	1	1	1	1
38	1	1	1	1
39	1	1	1	1
40	1	1	1	1

Abb. D–15: Artifizieller Testdatensatz zur Ausreißerererkennung

Die Variablen wurden vor dem Training auf den Wertebereich $[0; 1]$ normiert.³⁾ Abbruchkriterium für den Trainingsalgorithmus war das Erreichen eines durchschnittlichen quadrierten Fehlers von 0,01 über die normierten Werte aller Variablen. Dieser Zielwert wurde bereits nach wenigen Millise-

¹⁾ Siehe Abschnitt C.I.3, S 94 ff.

²⁾ Siehe Teilkapitel VIII, S. 237 ff.

³⁾ Siehe Abschnitt VII.1, S. 230 ff.

kunden erreicht. Dem trainierten Netz wurden nochmals die 40 Datenpunkte in normierter Form nacheinander eingegeben und jeweils der Netzoutput berechnet. Damit wurde für jeden Wert der absolute Rekonstruktionsfehler nach Formel (D.6) berechnet. Die Normierung macht die Rekonstruktionsfehler über alle Variablen vergleichbar. Das Ergebnis ist in Abb. D-16 dargestellt.

	Variable 1	Variable 2	Variable 3	Variable 4
1	,139985279...	,123284046...	,086126158...	,009694867...
2	,085399031...	,206133780...	,089028481...	,009981595...
3	,016582375...	,034078968...	,078081677...	,002870834...
4	,367871641...	,424439010...	,428533762...	,969302815...
5	,000000907...	,000000622...	,000001769...	,000002576...
6	,000000907...	,000000622...	,000001769...	,000002576...
7	,000000907...	,000000622...	,000001769...	,000002576...
8	Legende			
9				
10	99.999,99	gehört zu den 1 % stärksten Ausreißern		
11	99.999,99	gehört zu den 2 % stärksten Ausreißern		
12	99.999,99	gehört zu den 5 % stärksten Ausreißern		
13				
14				
15				
16	,000000907...	,000000622...	,000001769...	,000002576...
17	,000000907...	,000000622...	,000001769...	,000002576...
18	,000000907...	,000000622...	,000001769...	,000002576...
19	,000000907...	,000000622...	,000001769...	,000002576...
20	,000000907...	,000000622...	,000001769...	,000002576...

Abb. D-16: Absolute Rekonstruktionsfehler eines trainierten autoassoziativen Netzes für die normalisierten Werte aus Abb. D-15

Es ist zu erkennen, dass relativ größere Abweichungen der Variablenwerte auch durch einen eindeutig höheren Rekonstruktionsfehler gekennzeichnet sind. Diese Mindestanforderung an einen brauchbaren Ausreißertest ist also erfüllt. Zudem wird deutlich, dass jeder Datenpunkt (jede Tabellenzeile) als Ganzes behandelt wird. Eine Abweichung bei einer Variablen macht es dem Netz auch schwerer, die Werte der anderen Variablen korrekt zu rekonstruieren. So kommen die in den ersten vier Zeilen durchweg höheren Rekonstruktionsfehler auch bei Werten, die eigentlich keine Ausreißer sind, zustande. Das ist einerseits ein Nachteil, weil Werte fälschlicherweise als Ausreißer gekennzeichnet werden können, wie das z. B. für den ersten Wert

der Variablen 2 in Abb. D-16 der Fall ist. Andererseits können auf diese Weise Ausreißer identifiziert werden, die nur bei mehrdimensionaler Betrachtung als solche zu erkennen sind (siehe Abb. D-13, S. 201). Das Beispiel zeigt jedenfalls, dass vergleichsweise schwach gekennzeichnete Ausreißer ein methodisches Artefakt sein können. In Cenobi werden deshalb nur Ausreißer mit einem Rekonstruktionsfehler größer als 0,1 farblich hervorgehoben. Zudem sind die Hervorhebungen nach der Höhe des Rekonstruktionsfehlers abgestuft (siehe Abb. D-16).

Abb. D-17 zeigt das Ergebnis der Ausreißererkennung mit einem autoassoziativen neuronalen Netz für die Einflussgrößen des Spritzguss-Datensatzes¹⁾. Netz- und Trainingsparameter waren bis auf den Flaschenhals die selben wie oben. Für die Flaschenhalsschicht wurde in Cenobi grundsätzlich festgelegt, dass sie ein Drittel der Neuronen der Eingabeschicht enthalten soll. Da die absolute Höhe der Rekonstruktionsfehler keinen besonderen Informationswert hat, wird in Cenobi darauf verzichtet, sie explizit auszuweisen. Stattdessen werden die Werte mit den höchsten Rekonstruktionsfehlern im Originaldatensatz farblich hervorgehoben. Das erleichtert dem Nutzer die Interpretation und die Korrektur der Ausreißerwerte, die direkt in der Ergebnistabelle vorgenommen werden kann.

In Abb. D-17 fällt auf, dass die Nullwerte bei den geleisteten Spritzgusstunden (zweite Spalte) als klare Ausreißer markiert sind, während die 0 € Fremdleistungen für Entwicklungs- und Versuchsarbeiten (letzte Spalte) nicht gekennzeichnet sind. Diese Einordnung ist nachvollziehbar: bei den Spritzgusstunden handelt es sich augenscheinlich um einen Erfassungsfehler; bei den Fremdleistungen, die ohnehin sehr stark schwanken, könnte es sein, dass sie in diesen Monaten tatsächlich nicht genutzt wurden.

¹⁾ Siehe Teilkapitel II, S. 163 ff., sowie Anhang A1 und A2.

	Roh...	geleiste...	Zuha...	Anz...	Umsatz mit ...	Umsatz mi...	Umsatz	A...	A...	Anz...	Anz...	Anzahl ...	Fremdleist...	Fremdl...	Fremdl. ...
1	1.501	46.477,44	20.429	1.057	13.284.602,6	2.022.151,13	465.649,87	59	483	0	3.413	1.033	637.024	3.272.255,91	45.034,76	32.385,33
2	1.501	50.032,76	20.429	1.021	12.450.419,07	1.732.707,38	499.830,41	57	452	0	3.092	825	697.582	1.750.977,29	32.163,65	70.561,9
3	1.501	47.046,09	22.505	992	12.206.393,08	1.241.461,91	329.604,99	66	419	448	2.738	812	716.911	1.136.597,34	37.297,9	25.398,56
4	1.501	49.260,54	22.505	1.063	8.651.718,3	1.531.015,57	328.448,72	69	440	476	2.852	931	824.828	1.696.812,87	40.562,52	63.514,56
5	1.501	45.933,68	22.505	1.097	10.632.229,58	2.105.203,31	167.050,05	69	415	474	3.688	883	680.417	1.064.863,78	61.311,01	0
6	1.611	47.969,99	22.505	1.101	15.142.268,07	1.384.471,72	369.811,78	73	478	491	3.819	990	688.318	1.107.613,42	47.212,27	52.045,93
7	1.341	48.210,39	22.505	1.135	11.828.076,24	1.486.136,12	194.599,27	74	459	398	4.219	990	708.293	1.025.925,45	10.072,19	29.450,2
8	1.570	42.077,18	22.505	1.083	11.055.880,32	1.531.414,94	333.011,59	67	408	343	4.724	860	691.575	847.307,62	28.207,78	0
9	1.511	45.607,36	22.505	1.182	13.033.971,41	1.695.295,21	552.662,35	67	381	398	5.302	1.121	821.711	956.082,57	15.931,53	44.771,38
10	1.846	52.157,41	22.505	1.166	10.636.625,01	1.520.966,67	326.144,68	78	417	395	4.662	959	792.775	581.390,41	20.627,45	31.343,12
11	1.949	49.176,59	22.505	1.043	9.483.330,29	1.963.708,72	388.207,22	74	419	346	3.842	907	733.575	433.959,87	46.368,64	18.874,18
12	1.978	46.828,8	22.505	1.099	12.988.942,21	1.639.896,32	53.630,27	62	462	351	7.931	1.511	760.650	282.353,53	34.398,86	27.028,61
13	1.916	48.657,63	22.505	1.149	15.333.891,33	1.694.874,29	196.026,46	61	401	433	4.887	936	808.068	513.698,53	26.675,15	20.127,5
14	1.488	47.219,86	22.505	1.076	10.251.259,42	1.441.011,02	262.206,07	73	424	417	3.857	827	774.915	682.040,96	27.234,02	18.607,07
15	1.585	50.536,29	22.505	1.168	12.329.177,04	1.324.431,96	286.815,43	62	412							21.952,07
16	1.523	46.947,78	22.505	1.146	11.628.211,29	1.564.452,87	169.256,63	74	440							1.436,84
17	1.641	51.103,77	22.505	1.218	11.116.669,44	1.819.107,42	640.454,71	71	452							0
18	1.625	51.470,44	22.505	1.215	14.854.366,5	1.305.406,5	342.864,99	66	467							0
19	1.637	52.487,2	22.523	1.165	14.004.607,56	3.227.576,49	517.309,94	71	424							35.779,98
20	1.786	53.246,01	22.523	1.210	10.852.483,81	1.738.672,73	186.429,14	66	441							0
21	1.843	52.337,79	22.661	1.218	13.060.558,51	1.631.669,24	315.449,26	66	450							0
22	1.860	54.230,69	22.661	1.191	9.889.620,27	1.776.325,44	237.432,29	64	467							50.944,46
23	1.845	51.153,04	22.661	1.165	9.720.395,95	1.488.496,24	425.526,51	66	466							11.856,24
24	1.834	43.380	22.661	1.159	13.693.590,69	1.468.819,49	355.650,31	64	517							46.702,29
25	1.833	48.773,76	22.661	1.149	12.053.647,58	2.064.330,58	151.218,39	69	455							21.317,14
26	1.566	48.366,64	22.661	1.149	9.340.956,19	1.673.521,73	645.942,24	71	479							14.194,96
27	1.772	52.819,08	22.661	1.113	9.467.942,37	1.390.025,56	460.757,44	66	509							141.245,28
28	1.545	49.336,73	22.834	1.140	13.081.814,5	1.136.716,08	364.740,85	67	474							15.304,48
29	1.661	53.373,69	22.834	1.137	11.672.396,33	1.596.798,71	352.612,64	62	478							63.911,62
30	1.682	49.718,52	23.440	1.175	11.421.306,2	1.682.710,21	496.096,37	54	433							28.623,8
31	1.677	55.999,83	23.440	1.102	8.728.667,52	2.630.834,1	666.860,72	59	500							38.455,79
32	1.936	57.924,43	23.440	1.156	10.473.271,49	1.860.679,71	469.946,54	71	452							1.532,54
33	1.883	54.964,86	23.440	1.210	15.106.513,35	2.036.194,94	342.164,58	76	467							45.666,14
34	2.028	57.460,71	23.440	1.177	10.816.734,53	1.604.723,5	290.719,26	71	519							13.839,77
35	1.895	0	25.343	1.095	7.161.758,32	1.161.099,14	169.045,59	74	478	511	3.994	1.023	1.090.913	351.425,16	3.084,29	1.263,27
36	1.523	0	25.343	1.158	10.922.128,99	1.313.339,22	143.674,57	66	583	445	4.970	1.063	1.025.517	217.956,89	61.968,35	29.405,56
37	1.684	50.050,43	25.378	1.125	12.539.385,1	2.062.437,24	291.884,24	62	460	421	4.361	1.113	1.075.508	364.893,65	23.500,53	46.420,3
38	1.649	51.622,12	25.378	1.151	8.807.963,46	1.408.753,91	78.275,17	69	467	436	3.947	954	1.045.235	145.050,14	26.389,19	63.953,07
39	1.755	58.125,28	25.689	1.159	8.276.192,33	1.278.972,19	365.099,13	66	536	600	4.248	1.092	1.143.555	328.253,17	66.446,39	14.051,24
40	1.654	54.841,85	25.689	1.198	14.765.554,12	1.471.169,94	404.642,32	55	459	502	3.634	903	1.229.538	485.563,74	162.414,11	37.135,93
41	1.930	60.044,1	25.689	1.203	14.617.752,77	1.117.746,38	1.160.708,...	59	495	606	4.821	1.140	1.349.647	707.544,77	131.060,17	25.437,41
42	1.886	57.189,57	26.070	1.210	9.998.887,79	1.292.656,37	1.225.203,...	64	478	734	4.610	952	1.148.238	1.823.006,3	118.031,67	0

Abb. D-17: Ergebnis der Ausreißererkennung für die Einflussgrößen des Beispieldatensatzes (Ausschnitt)

Insbesondere einige der schwächeren Ausreißer scheinen – wie oben angesprochen – methodische Artefakte zu sein. Insgesamt lässt sich – außer bei den wenigen offenbar fehlenden Werten – zu den meisten markierten Werten nicht sagen, ob es sich tatsächlich um Fehler handelt, die korrigiert werden müssen, oder nicht. Dem Kostenrechner bleibt es an dieser Stelle nicht erspart, zumindest die dunkelrot markierten Werte im Einzelnen fachkundig auf ihre Richtigkeit zu prüfen und gegebenenfalls zu korrigieren.

Das Ergebnis der Ausreißererkennung für die Kostenstellen des Beispieldatensatzes ist in Abb. D-18 dargestellt. Netz- und Trainingsparameter waren dieselben wie oben.

	QM Pro...	Angebo...	Fertigu...	Spritzgus...	Etiketti...	Inmold L...	Montage...	Siebdruck	Recycli...	Auftrag...	Method...
1	47.301,29	48.851,48	24.695,88	365.442,71...	19.497,68	1.655,58	170.820,35	37.953,8	24.262,21	20.857,19	0
2	49.952,97	49.503,55	48.764,97	384.424,06...	17.855,43	7.041,63	161.476,19	39.476,06	25.690,18	23.683,83	0
3	51.386,38	35.640,12	66.212,1	383.568,42...	17.556,05	89.529,02	170.754,16	48.022,46	24.522,48	24.531,26	0
4	54.859,26	40.823,82	48.306,02	389.206,89...	27.348,08	1.905,29	166.760,47	36.080,35	24.407,4	19.218,46	0
5	50.904,71	15.485,84	44.449,11	364.211,66...	20.375,85	1.905,29	164.059,78	36.983,83	24.953,29	19.263,11	0
6	35.422,06	15.489,83	43.001	372.770,43	18.053,69	8.337,19	172.737,41	34.158,77	27.452,07	19.487,99	18.621,54
7	56.566,67	16.242,61	43.346,94	394.923,75...	18.959,49	192.372,5	158.730,25	34.958,51	29.534,24	19.484,27	18.632,93
8	55.876,42	15.921,61	47.528,78	352.011,48	18.959,49	2.047,2	214.083,23	35.828,59	25.741,11	20.886,82	18.326,82
9	47.435,37	15.472,04	42.964,4					05,86	25.046,92	19.219,07	17.726,5
10	48.283,66	15.466,59	42.925,4					52,79	24.619,48	21.154,15	17.726,5
11	46.499,31	15.464,5	43.030,4					74,23	25.090,87	19.549,22	17.726,5
12	47.522,13	15.737,57	39.692,4					68,56	28.149,81	16.883,64	20.094,19
13	47.229,28	15.497,03	39.238,4					94,14	26.171,06	20.118,35	17.932
14	47.244,61	15.442,47	34.113,4					49,99	25.109,82	28.967,2	18.025,62
15	43.262,08	15.442,47	42.174,4					579,1	25.784,72	28.967,2	17.844,78
16	47.491,81	15.442,47	36.982,4					35,53	25.109,82	17.307,94	18.316,43
17	37.381,07	33.853,41	36.714,4					47,51	25.536,91	20.210,29	18.320,15
18	40.486,55	30.389,28	36.647,4					12,54	26.438,87	28.739,95	18.149,27
19	37.524,95	29.239,36	34.989,4					26,94	29.461,9	19.357,93	17.932
20	38.109,48	15.834,46	36.868,4					70,24	27.491,96	22.687,59	18.017,83
21	37.379,5	15.877,35	38.594,4					852,3	25.972,57	21.107,14	18.888,71
22	37.497,26	15.843,55	36.589,4					32,04	26.247,03	21.897,07	22.465,42
23	44.365,13	15.970,67	36.629,4					35,81	25.556,29	20.214,36	18.441,38
24	37.531,56	16.519,75	47.332,4					70,95	46.071,9	15.274,71	20.781,89
25	37.354,85	15.953,09	36.735,4					87,83	25.755,99	21.837,91	17.943,42
26	40.053,93	15.914,6	38.520,4					79,72	26.145,01	30.097,67	17.870,83
27	41.229,23	15.997,44	32.456,4					89,87	26.182,74	20.593,17	19.123,41
28	41.127,98	16.202,28	36.523,4					56,89	27.063,74	20.415,91	18.082,04
29	38.732,24	15.900,07	37.650,4					09,39	38.780,36	25.166,7	17.983,03

Legende

99.999,99 gehört zu den 1 % stärksten Ausreißern

99.999,99 gehört zu den 2 % stärksten Ausreißern

99.999,99 gehört zu den 5 % stärksten Ausreißern

Ausreißer korrigieren

Doppelklick auf Zelle -> Wert ändern -> Enter

Ausreißer als fehlenden Wert behandeln

Wert zu "-1" korrigieren

-> Schaltfläche "fehlende Werte" anwählen

-> Haken bei "-1 markiert fehlende Werte" setzen

Abb. D-18: Ergebnis der Ausreißerererkennung für die Kostenstellen des Beispieldatensatzes (Ausschnitt)

Die fehlenden Kostendaten für die Methodenzeitmessung (letzte Spalte) werden zuverlässig als Ausreißer ausgezeichnet. Ebenso sind die bereits in Teilkapitel II (S. 164 f.) beschriebenen Unregelmäßigkeiten in der Kostenstelle Inmold Labeling hervorgehoben. Zur Einordnung der übrigen potentiellen Ausreißer bedarf es auch hier eines fachkundigen Kostenrechners.

c) Korrektur der Ausreißer

Die Korrektur der Ausreißer sollte nicht automatisch erfolgen. Zwar können mit statistischen Methoden potentielle Ausreißer aufgezeigt werden. Die Entscheidung, ob es sich tatsächlich um einen so ungewöhnlichen Wert handelt, dass er von der weiteren Modellierung ausgenommen oder durch einen anderen Wert ersetzt werden muss, sollte allerdings einem fachkundigen Kenner der datengenerierenden Prozesse überlassen bleiben. Wenn

dieser feststellt, dass es sich tatsächlich um einen Ausreißer handelt, können zwei Fälle unterschieden werden:

- (1) der korrekte Wert kann ermittelt werden,
- (2) der korrekte Wert kann nicht ermittelt werden.

Der erste Fall ist unproblematisch – der Ausreißer wird einfach durch den korrekten Wert ersetzt. Im zweiten Fall ist wiederum zu unterscheiden, ob der Ausreißer

- (i) wie ein fehlender Wert behandelt werden kann oder ob er
- (ii) mit anderen Werten zusammenhängt.

Beispiele für Fall (i) sind die beiden Nullwerte in der zweiten Spalte von Abb. D-17 (geleistete Spritzgussstunden). Methoden zum korrekten Umgang mit ihnen werden im folgenden Abschnitt 2 erläutert. Ein Beispiel für Fall (ii) sind die Kosten in Höhe von 192.372,50 € in der Kostenstelle Inmold Labeling in Abb. D-18 (sechste Spalte). Diese Kosten sind tatsächlich angefallen. Möglicherweise wurden sie nicht in voller Höhe in der Periode verursacht, in der sie verbucht wurden. Wenn sie aber aus dem Datensatz gestrichen werden, reduzieren sich damit die durchschnittlichen Inmold-Labeling-Kosten. Schätzungen der zukünftigen Kostenstellenkosten würden dadurch verzerrt. Deshalb sollten die 192.372,50 € – wenn es sich tatsächlich um einen Ausreißer handelt und nicht festgestellt werden kann, wie sie verursachungsgerecht zugeordnet werden können – mittelwertneutral auf die umliegenden Perioden verteilt werden.

2. Umgang mit fehlenden Werten

a) Gründe für fehlende Werte

In fast jedem empirischen Datensatz gibt es fehlende Werte. Das Ziel im Umgang mit fehlenden Werten ist nicht, ihre „wahren“ Ausprägungen zu finden, sondern trotz der fehlenden Werte eine unverzerrte Schätzung zu erhalten.¹⁾ Der richtige Umgang mit fehlenden Werten hängt davon ab, wie ihr Fehlen zustande gekommen ist. Drei Ursachen werden unterschieden:²⁾

(1) *Missing completely at random* (MCAR): die Wahrscheinlichkeit, dass ein Wert $x_i^{(n)}$ fehlt, hängt von Umwelteinflüssen ab, die nicht im Modell enthalten sind.

Beispiel: die Antwort auf eine schriftliche Befragung geht auf dem Postweg verloren.

(2) *Missing at random* (MAR): die Wahrscheinlichkeit, dass ein Wert $x_i^{(n)}$ fehlt, hängt von Werten der anderen Modellvariablen $x_j^{(n)}$ ($j \neq i$) ab.

Beispiel: publizitätspflichtige Unternehmen antworten eher auf eine Frage nach dem Ausmaß der betrieblichen Umweltschutzmaßnahmen. Wenn sowohl die Publizitätspflichtigkeit als auch das Ausmaß der Umweltschutzmaßnahmen Modellvariablen sind, liegt ein MAR-Zusammenhang vor.

(3) *Missing not at random* (MNAR): die Wahrscheinlichkeit, dass ein Wert $x_i^{(n)}$ fehlt, hängt von seiner Ausprägung ab.

Beispiel: Unternehmen, die viel Geld für Umweltschutzmaßnahmen ausgeben, antworten eher auf die Frage „Wie viel Geld gibt ihr Unternehmen für Umweltschutz aus?“

¹⁾ Vgl. Schafer/Graham [Missing Data] 149; Backhaus/Blechsmidt [Fehlende Werte] 266.

²⁾ Vgl. Rubin [Missing Data] 582; Schafer/Graham [Missing Data] 151; van Buuren [Imputation] 6 f. Die Beispiele stammen von Göthlich [Fehlende Daten] 121.

In den ersten beiden Fällen sind die Gründe für das Fehlen der Werte für die weitere Analyse der Daten irrelevant – in den Sozialwissenschaften spricht man von einem *ignorable nonresponse*.¹⁾

In der Kostenrechnung fehlen Werte in der Regel *completely at random* (MCAR). Auch MAR-Fälle sind denkbar. MNAR-Szenarien sind hingegen kaum vorstellbar. Grundsätzlich ist also anzunehmen, dass die Gründe für das Fehlen der Werte ignoriert werden können.

b) Nachteile einfacher Verfahren im Umgang mit fehlenden Werten

Unter MCAR-Bedingungen erhält man eine unverzerrte Schätzung der Modellparameter durch das naheliegendste und wohl am häufigsten praktizierte Verfahren: man verwendet **nur die vollständigen Fälle** – wenn ein Wert fehlt, wird die gesamte Zeile im Datensatz gestrichen (*complete-case analysis, listwise deletion*).²⁾ Zwar erhält man damit für MCAR-Daten unverzerrte Modellparameter, die Varianz der Schätzung wird jedoch wegen der verringerten Fallzahl grundsätzlich zu hoch ausgewiesen. Außerdem können in multivariaten Modellen mit vielen Variablen schon wenige fehlende Werte dazu führen, dass zahlreiche Beobachtungen gelöscht werden. Bei komplexeren Modellen wie z. B. künstlichen neuronalen Netzen kann diese Verschwendung von Informationen die Modellgüte stark beeinträchtigen. Zudem führt der zeilenweise Ausschluss unvollständiger Beobachtungen unter MAR- und MNAR-Bedingungen zu verzerrten Schätzungen, weil der reduzierte Datensatz in diesen Fällen keine repräsentative Stichprobe des kompletten Datensatzes ist.

¹⁾ Vgl. Rubin [Missing Data] 581 ff.

²⁾ Vgl. auch im Weiteren Schafer/Graham [Missing Data] 155 ff.; Backhaus/Blechsmidt [Fehlende Werte] 271; van Buuren [Imputation] 8 f.; jeweils mit weiteren Nachweisen.

Ein weiteres einfaches Verfahren besteht darin, für einen fehlenden Wert den **Mittelwert** der beobachteten Variablenwerte einzusetzen (*mean imputation*). Dadurch werden zwar keine Informationen verschwendet und der Mittelwert der Variablen bleibt erhalten.¹⁾ Die übrigen Verteilungsmaße wie Varianz, Median und andere Quantile werden aber verzerrt, wenn die Variable nicht normalverteilt ist. In der Folge wird die Varianz der Schätzung unterschätzt, so dass u. a. die Konfidenzintervalle der Schätzung zu klein ausgewiesen werden.

Bei der **Regressionsimputation** (*conditional mean imputation*) wird für jede Variable mit fehlenden Werten ein eigenes Regressionsmodell $y = f(x)$ mit der unvollständigen Variablen als abhängiger Variable y aufgestellt. Als Prädiktorvariable x werden häufig die übrigen Variablen des Datensatzes verwendet. Wenn es sich bei dem unvollständigen Datensatz allerdings um die Einflussgrößen eines Regressionsmodells handelt, sollten die Variablen voneinander unabhängig sein.²⁾ In diesem Fall bietet es sich an, für die Imputation das eigentliche Regressionsmodell umzudrehen und für eine unvollständige Einflussgröße x_i die abhängigen Variablen y (im Falle von Kostenfunktionen also die Kostenhöhe) als Prädiktoren zu verwenden: $x_i = f(y)$. Im Weiteren wird die Regressionsimputation zwar durchgehend anhand der Regressionsgleichung $y = f(x)$ beschrieben, diese Notation beschreibt aber nur das Verhältnis der verwendeten Variablen in der Regressionsimputation. Die Variable y einer Regressionsimputation kann wie eben beschrieben durchaus eine Variable x_i des eigentlich interessierenden Schätzmodells sein. Die Regressionsfunktion wird mit den beobachteten Werten parametrisiert. Anschließend werden die fehlenden Werte durch

¹⁾ Vgl. auch im Weiteren Schafer/Graham [Missing Data] 159; Backhaus/Blechs Schmidt [Fehlende Werte] 271; van Buuren [Imputation] 10 f.

²⁾ Vgl. Abschnitt V.2, S. 193 ff.

das Regressionsmodell geschätzt. Dieses Verfahren erlaubt eine unverzerrte Schätzung unter MCAR- und MAR-Bedingungen.¹⁾ Wenn es keinen Zusammenhang zwischen y und x gibt (im MCAR-Fall also), ist die Regressionsimputation äquivalent zur Mittelwertimputation. Demnach wird auch bei der Regressionsimputation die Varianz unterschätzt. Bei MAR-Daten kommt noch hinzu, dass der Zusammenhang zwischen y und x künstlich verstärkt wird. Dadurch wird die Korrelation zwischen y und x überschätzt. Ein weiterer Nachteil ist, dass die eingesetzten Werte außerhalb des Definitionsbereichs der Variablen liegen können. Beispielsweise ist es möglich, dass negative Werte bei einer Variablen eingesetzt werden, die eigentlich nicht negativ werden kann.

c) Multiple Imputation zur Berücksichtigung der Unsicherheit über die fehlenden Werte

α) Grundidee der multiplen Imputation

Das grundsätzliche Problem der in Abschnitt b beschriebenen Imputationsverfahren besteht darin, dass sie, indem sie einen einzelnen Wert anstelle des fehlenden Wertes einsetzen, eine falsche Sicherheit über die Höhe des fehlenden Wertes in den Datensatz einbringen.²⁾ Die fehlenden Werte werden behandelt, als ob ihre wahre Ausprägung bekannt wäre. Dadurch werden die Unsicherheit der Daten und damit auch die Varianz einer auf ihnen beruhenden Schätzung unterschätzt. Für dieses Problem haben sich zwei Lösungen etabliert:

¹⁾ Vgl. auch im Weiteren Schafer/Graham [Missing Data] 159 ff.; Göthlich [Fehlende Daten] 125; van Buuren [Imputation] 11 f.

²⁾ Vgl. auch im Weiteren Rubin [Imputation] 12 f.; Schafer/Graham [Missing Data] 161.

- (1) eine Modifizierung der Maximum-Likelihood-Schätzer unter Verwendung des Expectation-Maximization-Algorithmus (EM-Algorithmus),¹⁾
- (2) multiple Imputation: statt eines einzelnen Wertes werden verschiedene Werte eingesetzt, die zufällig aus einer Wahrscheinlichkeitsverteilung gezogen werden.²⁾ Dadurch entstehen mehrere vollständige Datensätze, die separat analysiert werden. Die Analyseergebnisse werden anschließend unter Berücksichtigung ihrer Varianz zusammengelegt (siehe Abb. D–19).

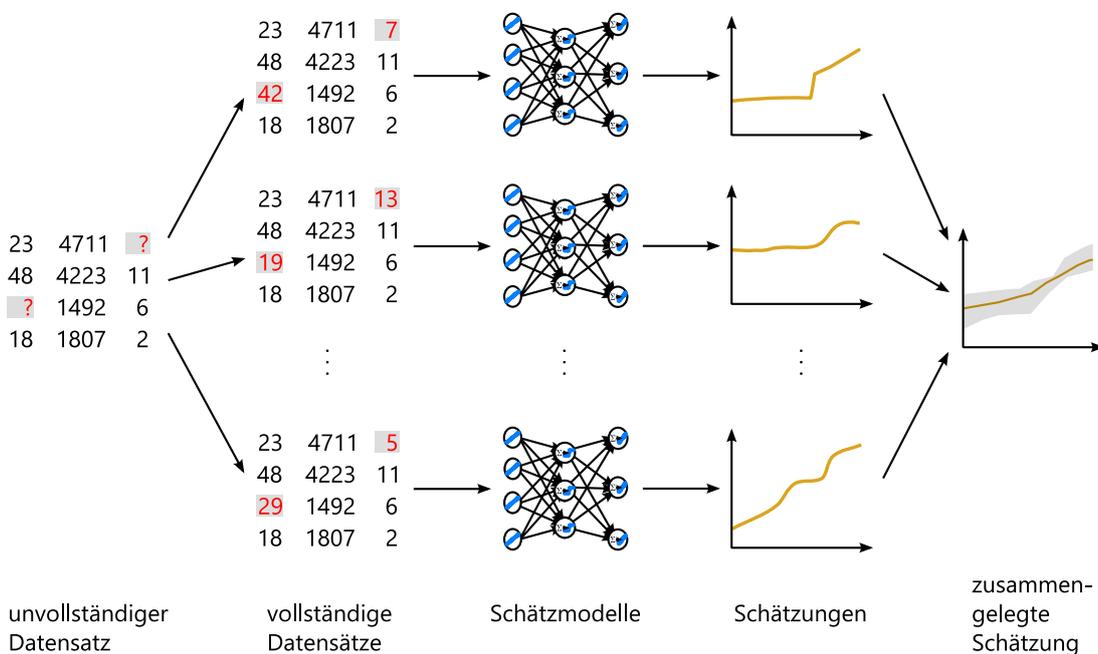


Abb. D–19: Ablauf einer multiplen Imputation

In vergleichenden Experimenten führen modifizierte Maximum-Likelihood-Schätzung und multiple Imputation zu ähnlich guten Schätzungen mit fast unverzerrten Modellparametern und Varianzen.³⁾ Die Verfahren

¹⁾ Vgl. Schafer/Graham [Missing Data] 162 ff.; Backhaus/Blehschmidt [Fehlende Werte] 272 f.; jeweils mit weiteren Nachweisen.

²⁾ Vgl. Rubin [Imputation]; van Buuren [Imputation] 16 ff.

³⁾ Vgl. Schafer/Graham [Missing Data] 169 f.; Göthlich [Fehlende Daten] 129 f.; Backhaus/Blehschmidt [Fehlende Werte] 270; jeweils mit weiteren Nachweisen.

unterscheiden sich vor allem darin, dass die modifizierte Maximum-Likelihood-Schätzung das Problem fehlender Daten mit einer speziell angepassten Modellparametrisierung löst, während die multiple Imputation vollständige Datensätze erzeugt, die anschließend mit einem beliebigen Schätzmodell analysiert werden können.¹⁾ Diese Trennung des Fehlende-Werte-Problems vom (eigentlichen) Analysevorgang erleichtert die Anwendung der multiplen Imputation, da für die Schätzung der Modellparameter vertraute Standardmethoden verwendet werden können, die auf vollständige Datensätze angewiesen sind.²⁾ Zudem ist die multiple Imputation ein robustes Verfahren, das auf MCAR- und MAR-Daten ohne problemspezifische Anpassungen angewendet werden kann. Deshalb ist es in Cenobi das Standardverfahren zum Umgang mit fehlenden Werten.

β) Bayessche Regressionsimputation zur Erzeugung repräsentativer Stichproben

Um die vollständigen Datensätze für eine multiple Imputation zu erzeugen, benötigt man für jeden fehlenden Wert eine Wahrscheinlichkeitsverteilung, die die Unsicherheit über die Ausprägung des Wertes repräsentiert. Als besonders zweckmäßig hat sich hierfür eine bayessche Erweiterung der Regressionsimputation (s. o.) erwiesen.³⁾

Algorithmus D–1 zeigt den Ablauf des Verfahrens in Pseudocode.⁴⁾ Zur Vereinfachung wird nur eine unvollständige Variable y betrachtet, von der N_{obs} Werte vorliegen und N_{mis} Werte fehlen. Es wird davon ausgegangen, dass ein vollständiger Datensatz X zur Verfügung steht, der $N_{\text{obs}} + N_{\text{mis}}$

¹⁾ Vgl. auch im Weiteren Schafer/Graham [Missing Data] 170.

²⁾ Vgl. auch Rubin [Imputation] 15 f.

³⁾ Vgl. auch im Weiteren Rubin [Imputation] 166 f.; Schafer/Graham [Missing Data] 166 f.; van Buuren [Imputation] 55 ff.

⁴⁾ Vgl. zum Algorithmus und auch im Weiteren van Buuren [Imputation] 58.

Beobachtungen von P Prädiktorvariablen x_i ($i = 1, \dots, P$) enthält. Zunächst werden die Regressionskoeffizienten $\hat{\beta}$ einer linearen Regressionsfunktion

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \dots + \hat{\beta}_p x_p \quad (\text{D.7})$$

durch Minimierung der Fehlerquadrate berechnet (Zeile 4 bis 6).¹⁾

Algorithmus Multiple Imputation mit bayesscher Regressionsimputation

- 1: $\mathbf{y}_{\text{obs}} \leftarrow (N_{\text{obs}} \times 1)$ -Spaltenvektor mit beobachteten Werten der unvollständigen Variablen y
 - 2: $\mathbf{X}_{\text{obs}} \leftarrow (N_{\text{obs}} \times P)$ -Matrix mit Prädiktorwerten für die beobachteten Werte von y
 - 3: $\mathbf{X}_{\text{mis}} \leftarrow (N_{\text{mis}} \times P)$ -Matrix mit Prädiktorwerten für die fehlenden Werte von y
 - 4: $\mathbf{S} \leftarrow \mathbf{X}_{\text{obs}}^T \cdot \mathbf{X}_{\text{obs}}$
 - 5: $\mathbf{V} \leftarrow (\mathbf{S} + \text{diag}(\mathbf{S})\kappa)^{-1}$
 - 6: $\hat{\beta} \leftarrow \mathbf{V} \mathbf{X}_{\text{obs}}^T \mathbf{y}_{\text{obs}}$
 - 7: Berechne $\mathbf{V}^{1/2}$ durch Cholesky-Zerlegung.
 - 8: **wiederhole**
 - 9: Ziehe einen zufälligen Wert $\dot{g} \sim \chi^2_\nu$ mit $\nu = N_{\text{obs}} - P$.
 - 10: $\dot{\sigma}^2 \leftarrow \frac{(\mathbf{y}_{\text{obs}} - \mathbf{X}_{\text{obs}} \hat{\beta})^T (\mathbf{y}_{\text{obs}} - \mathbf{X}_{\text{obs}} \hat{\beta})}{\dot{g}}$
 - 11: $\dot{z}_1 \leftarrow (P \times 1)$ -Vektor mit unabhängigen Zufallszahlen aus $\mathcal{N}(0, 1)$
 - 12: $\dot{\beta} \leftarrow \hat{\beta} + \mathbf{V}^{1/2} \dot{z}_1 \dot{\sigma}$
 - 13: $\dot{z}_2 \leftarrow (N_{\text{mis}} \times 1)$ -Vektor mit unabhängigen Zufallszahlen aus $\mathcal{N}(0, 1)$
 - 14: $\dot{y} \leftarrow \mathbf{X}_{\text{mis}} \dot{\beta} + \dot{z}_2 \dot{\sigma}$
 - 15: Vervollständige y durch \dot{y} .
 - 16: **bis** m vollständige Vektoren y erzeugt
-

Algorithmus D-1: Multiple Imputation mit bayesscher Regressionsimputation

¹⁾ Vgl. z. B. Hedderich/Sachs [Statistik] 699.

Der Y-Achsenabschnitt $\hat{\beta}_0$ ist in Algorithmus D-1 nicht explizit aufgeführt, wird aber über eine Dummyvariable in Matrix X berücksichtigt. Der sogenannte Ridge-Parameter κ (Zeile 5) ist eine sehr kleine Zahl (z. B. $\kappa = 0,00001$), die sicherstellt, dass die Matrix V invertierbar ist.

In den Zeilen 9 bis 12 werden die bayesschen Regressionskoeffizienten berechnet. Sie sind nicht einwertig, sondern werden – etwas vereinfacht – von einer Normalverteilung repräsentiert, die die Kleinste-Quadrate-Regressionskoeffizienten $\hat{\beta}$ als Mittelwert enthält.¹⁾ Für die multiple Imputation werden in jedem Imputationsdurchgang neue Regressionskoeffizienten $\hat{\beta}$ aus der bayesschen A-posteriori-Verteilung gezogen (Zeile 12). Damit wird die Unsicherheit über die wahre Ausprägung der Regressionskoeffizienten berücksichtigt. Jeder Wert, der zufällig aus einer Verteilung gezogen wird, ist mit einem Punkt über dem Symbol gekennzeichnet.

Zur Unsicherheit über die Höhe der Parameter kommt noch die Unsicherheit über die Streuung der fehlenden Werte um die Regressionsgerade hinzu. Sie fließt über den Störterm $\dot{\epsilon} = z_2\dot{\sigma}$ in die Berechnung der Imputationswerte \dot{y} ein (Zeile 14). Insgesamt berücksichtigt die Imputationsfunktion

$$\dot{y} = \dot{\beta}_0 + \dot{\beta}_1x_1 + \dots + \dot{\beta}_px_p + \dot{\epsilon}, \text{ mit } \dot{\epsilon} \sim \mathcal{N}(0, \dot{\sigma}^2) \quad (\text{D.8})$$

also sowohl die Parameterunsicherheit als auch die Streuung um die Regressionsfunktion.²⁾

Die Anzahl der Imputationen m muss nicht besonders groß sein. Experimentell wurde gezeigt, dass bei wenigen fehlenden Werten bereits drei Imputationen eine fast unverzerrte Schätzung ermöglichen.³⁾ Die in der wis-

¹⁾ Vgl. grundlegend Box/Tiao [Bayesian Inference] 113 ff.

²⁾ Vgl. van Buuren [Imputation] 57.

³⁾ Vgl. auch im Weiteren van Buuren [Imputation] 49 f., 60 f., mit weiteren Nachweisen.

senschaftlichen Literatur ausgesprochenen Empfehlungen lassen sich in der Faustregel zusammenfassen, dass die Anzahl der Imputationen dem Anteil der unvollständigen Fälle in Prozent entsprechen sollte. Wenn also z. B. 5 % der Fälle fehlende Werte enthalten, reichen fünf imputierte Datensätze aus, um die Unsicherheit über die fehlenden Werte angemessen zu berücksichtigen.

γ) Robuste Imputation durch *predictive mean matching*

Die bayessche Regressionsimputation hat gegenüber der einfachen Regressionsimputation aus Abschnitt b den Vorteil, dass sie – integriert in ein multiples Imputationsverfahren – eine fast unverzerrte Schätzung von Modellparametern und Varianz ermöglicht (unter MCAR- und MAR-Bedingungen).¹⁾ Es besteht aber weiterhin das Problem, dass unrealistische oder unsinnige Werte eingesetzt werden können (z. B. negative Körpergrößen). Zudem ist die Güte der bayesschen Regressionsimputation davon abhängig, wie gut die getroffenen Modellannahmen – linearer Zusammenhang, normalverteilte Parameter und Störgrößen – zutreffen. Beide Probleme können durch *predictive mean matching* reduziert werden.²⁾

Predictive mean matching ist eine Hot-Deck-Methode, d. h. fehlende Werte werden durch tatsächlich beobachtete Werte aus vollständigen Fällen ersetzt, die dem unvollständigen Fall ähnlich sind. Als Ähnlichkeitsmaß d dient beim *predictive mean matching* der Abstand zwischen der bayesschen

¹⁾ Vgl. z. B. die experimentellen Ergebnisse bei van Buuren [Imputation] 61 f.

²⁾ Vgl. grundlegend Rubin [Statistical Matching]; Little [Missing-Data Adjustments] 291. Vgl. auch Rubin [Imputation] 168; Koller-Meinfelder [Predictive Mean Matching] 32; van Buuren [Imputation] 69.

Schätzung für einen fehlenden Wert $y^{(n_{\text{mis}})}$ und der Kleinste-Quadrate-Schätzung für einen beobachteten Wert $\hat{y}^{(n_{\text{obs}})}$:¹⁾

$$d(y^{(n_{\text{mis}})}, \hat{y}^{(n_{\text{obs}})}) = |y^{(n_{\text{mis}})} - \hat{y}^{(n_{\text{obs}})}| \quad (\text{D.9})$$

mit $y^{(n_{\text{mis}})} = \mathbf{x}^{(n_{\text{mis}})} \hat{\boldsymbol{\beta}} + z_2^{(n_{\text{mis}})} \hat{\sigma}$ siehe (D.8) und Algorithmus D-1, Zeile 14,

$\hat{y}^{(n_{\text{obs}})} = \mathbf{x}^{(n_{\text{obs}})} \hat{\boldsymbol{\beta}}$ siehe (D.7) und Algorithmus D-1, Zeile 6,

$n_{\text{mis}} = 1, \dots, N_{\text{mis}}$ Index der fehlenden Werte von y ,

$n_{\text{obs}} = 1, \dots, N_{\text{obs}}$ Index der beobachteten Werte von y ,

$\mathbf{x}^{(n_{\text{mis}})}$ $(1 \times P)$ -Zeilenvektor, eine Zeile aus \mathbf{X}_{mis}
(siehe Algorithmus D-1, Zeile 3),

$\mathbf{x}^{(n_{\text{obs}})}$ $(1 \times P)$ -Zeilenvektor, eine Zeile aus \mathbf{X}_{obs}
(siehe Algorithmus D-1, Zeile 2).

Algorithmus D-2 zeigt, wie *predictive mean matching* in eine multiple Imputation eingebunden werden kann.²⁾ Die Zeilen 1 bis 7 dienen der Berechnung des Abstands nach Formel (D.9). Anhand dieses Abstands werden die drei nächsten Nachbarn eines fehlenden Wertes bestimmt und ihre Indizes in der *SPENDER*-Liste festgehalten (Zeile 8 bis 13). Von diesen Nachbarn wird zufällig einer ausgewählt (Zeile 15), der dem fehlenden Wert $y^{(n_{\text{mis}})}$ seinen Beobachtungswert $y^{(s^*)}$ (nicht den Schätzwert $\hat{y}^{(s^*)}$) „spendet“ (Zeile 16). Die zufällige Spenderwahl verhindert, dass in jedem der m Imputationsdurchläufe derselbe Wert eingesetzt wird. Dies würde den Datensatz verzerren und die Vorteile der multiplen Imputation zunichte machen. Eine Spenderliste mit drei Elementen hat sich in vergleichenden Experimenten als vorteilhaft erwiesen. Wenn die Liste größer wird, wächst die Gefahr,

¹⁾ Vgl. Little [Missing-Data Adjustments] 292; Koller-Meinfelder [Predictive Mean Matching] 43; van Buuren [Imputation] 71 f.

²⁾ Vgl. zum Algorithmus und auch im Weiteren van Buuren [Imputation] 71 ff., mit weiteren Nachweisen.

dass unpassende Werte eingesetzt werden. Algorithmus D-2 mit $|SPENDER| = 3$ ist das Standardverfahren in der verbreiteten Imputationsmethodenbibliothek MICE für die Statistiksoftware R.¹⁾

Algorithmus Multiple Imputation mit bayesschem *predictive mean matching*

- 1: Berechne $\hat{\beta}$ wie in Algorithmus D-1.
 - 2: **wiederhole**
 - 3: Berechne $\hat{\beta}$, \hat{z}_2 und $\hat{\sigma}$ wie in Algorithmus D-1.
 - 4: **für** $n_{\text{mis}} = 1, \dots, N_{\text{mis}}$ **führe aus**
 - 5: $SPENDER \leftarrow \emptyset$
 - 6: **für** $n_{\text{obs}} = 1, \dots, N_{\text{obs}}$ **führe aus**
 - 7: Berechne $d(\hat{y}^{(n_{\text{mis}})}, \hat{y}^{(n_{\text{obs}})})$ nach Formel (D.9).
 - 8: **wenn** $d(\hat{y}^{(n_{\text{mis}})}, \hat{y}^{(n_{\text{obs}})}) < d(\hat{y}^{(n_{\text{mis}})}, \hat{y}^{(s)})$ mit $s \in SPENDER$ **dann**
 - 9: $SPENDER \leftarrow SPENDER \cup \{n_{\text{obs}}\}$
 - 10: **ende wenn**
 - 11: **wenn** $|SPENDER| > 3$
 - 12: $SPENDER \leftarrow SPENDER \setminus s$
 mit $s = \operatorname{argmax}\{d(\hat{y}^{(n_{\text{mis}})}, \hat{y}^{(s)}) \mid s \in SPENDER\}$
 - 13: **ende wenn**
 - 14: **ende für**
 - 15: Ziehe zufällig einen Spender $s^* \in SPENDER$.
 - 16: $y^{(n_{\text{mis}})} \leftarrow y^{(s^*)}$
 - 17: **ende für**
 - 18: **bis** m vollständige Vektoren y erzeugt
-

Algorithmus D-2: Multiple Imputation mit bayesschem predictive mean matching

Predictive mean matching hat den Vorteil, dass immer plausible Werte eingesetzt werden. Außerdem ist es robust gegenüber Fehlspezifizierungen des

¹⁾ Vgl. van Buuren/Groothuis-Oudshoorn [MICE] 11.

Imputationsmodells, wie Abb. D–20 zeigt. Der Zusammenhang ist hier eigentlich eher parabelförmig, die Regressionsfunktionen sind aber linear. Der imputierte Punkt $(x^{(n_{\text{mis}})}, y^{(s^*)})$ liegt dennoch nahe beim vermuteten Kurvenverlauf.¹⁾

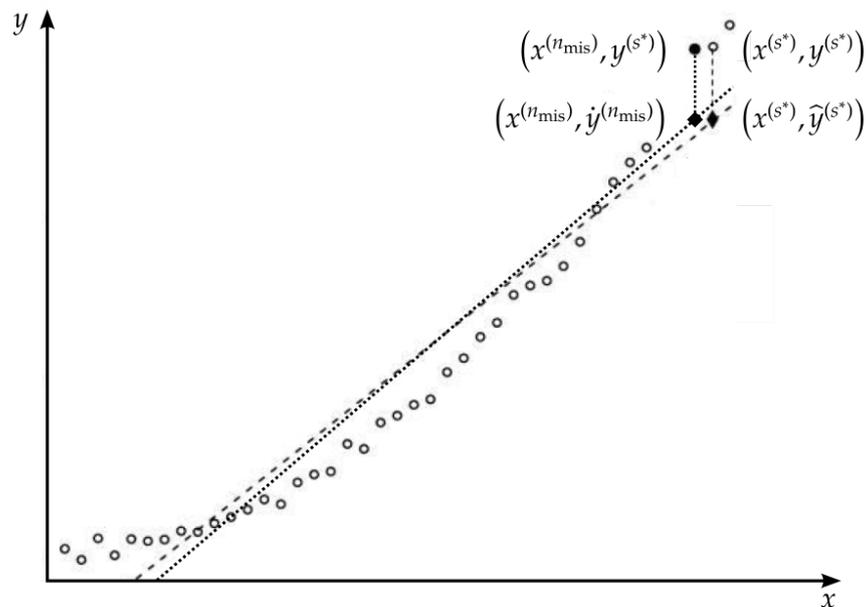


Abb. D–20: Predictive mean matching mit einem Prädiktor x und einer unvollständigen Variablen y ²⁾

Eine explizite Modellierung des Zusammenhangs zwischen der unvollständigen Variablen y und den Prädiktorvariablen x sowie der Wahrscheinlichkeitsverteilung der fehlenden Werte ist demnach bei einer multiplen Imputation mit bayesschem *predictive mean matching* nicht zwingend erforderlich.³⁾ Zugleich wurde experimentell gezeigt, dass Schätzungen, die auf Daten beruhen, die nach diesem Verfahren vervollständigt wurden, die wahren Parameter im Mittel um weniger als 1 % verfehlen, wenn weniger als die Hälfte der Fälle vervollständigt werden muss und MCAR- oder

¹⁾ Vgl. Koller-Meinfelder [Predictive Mean Matching] 32.

²⁾ Vgl. Koller-Meinfelder [Predictive Mean Matching] 32, abgeändert.

³⁾ Vgl. van Buuren [Imputation] 69.

MAR-Bedingungen vorliegen.¹⁾ Wegen dieser Eigenschaften eignet sich die multiple Imputation mit bayesschem *predictive mean matching* sehr gut als Standardmethode im Umgang mit fehlenden Werten (nicht nur) in der Kostenrechnung.

δ) Objektorientierte Implementierung der multiplen Imputation

In Cenobi wird eine multiple Imputation mit bayesschem *predictive mean matching* automatisch durchgeführt, wenn der Nutzer eine kostenrechnerische Analyse startet. Er muss zuvor nur angeben, welche Zeichen in einem Datensatz einen fehlenden Wert kennzeichnen (siehe Abb. D–21).

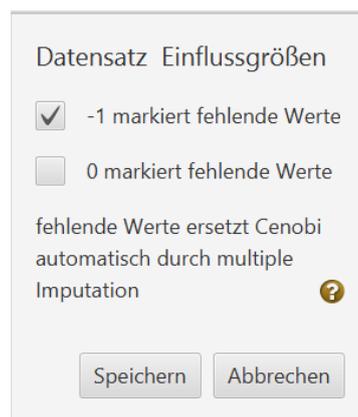


Abb. D–21: Dialogfenster zur Kennzeichnung fehlender Werte

Die multiple Imputation wird von einem Ensemble-Objekt koordiniert. Ein neues Ensemble wird immer dann erzeugt, wenn für eine Analyse (z. B. eine Prognose oder eine Kalkulation) ein neues Schätzmodell notwendig ist. Das ist der Fall, wenn neue Daten ausgewählt oder die Modellierungseinstellungen geändert wurden. Eine Ensembleschätzung hat eigentlich den Zweck, das *overfitting* einzelner Schätzungen auszugleichen und damit das

¹⁾ Vgl. Schafer/Graham [Missing Data] 169; Koller-Meinfelder [Predictive Mean Matching] 55, 86 f.; van Buuren [Imputation] 74, mit weiteren Nachweisen.

*bias-variance dilemma*¹⁾ zu überwinden.²⁾ Da bei einer Ensembleschätzung ohnehin mehrere Schätzmodelle parametrisiert und anschließend zusammengeführt („gepoolt“) werden, lässt sich hier eine multiple Imputation ohne großen Aufwand einfügen.

Das Klassendiagramm in Abb. D–22 zeigt die Struktur der Ensemble-Klasse und ihre Beziehung zu den Klassen, die sie für die multiple Imputation nutzt.³⁾ Im ersten Fach eines Klassenrechtecks steht der Klassenname, im zweiten stehen die Attribute und im dritten die Operationen, die ein Objekt dieser Klasse besitzt.⁴⁾ In Abb. D–22 sind nur die wichtigsten Attribute und Operationen ohne Implementierungsdetails eingezeichnet.

Ein Ensemble besitzt einen input- und einen target-Datensatz. Der Datensatz mit Vergangenheitswerten der abhängigen Variablen heißt target, weil er das Ziel ist, auf das die neuronalen Netze trainiert werden. Beide Datensätze können fehlende Werte enthalten. Deshalb gibt es einen inputImputator und einen targetImputator. Beide sind PredictiveMeanMatching-Objekte, die auf die Anfrage `getImputedVariableSet()` einen vollständigen Datensatz zurückgeben. Dafür erzeugt ein PredictiveMeanMatching-Objekt für jede Variable des unvollständigen `toBeImputed`-Datensatzes eine BayesianLinearRegression mit dem predictor-Datensatz als input und den beobachteten Werten der zu imputierenden Variablen als target. Ein BayesianLinearRegression-Objekt gibt auf die Anfrage `getEstimate()` eine bayessche Schätzung nach Formel (D.8) und auf die Anfrage `getOLSEstimate()` eine Kleinste-Quadrate-

¹⁾ Siehe Abschnitt C.I.2, S. 88 ff.

²⁾ Siehe Kapitel D.XIII, S. 284 ff.

³⁾ Die Instanzen einer Klasse heißen *Objekte*. Die Objekte einer Klasse besitzen die für die Klasse definierten Eigenschaften und Operationen (siehe S. 176).

⁴⁾ Zur Notation von Klassendiagrammen vgl. ausführlich ISO [UML Superstructure] 39 ff.; Seemann/von Gudenberg [Software-Entwurf] 43 ff.

Schätzung nach Formel (D.7) zurück. Beide werden nach Algorithmus D-1 berechnet. Diese Schätzungen verarbeitet das PredictiveMeanMatching-Objekt wie in Algorithmus D-2 beschrieben. Die in Algorithmus D-2 mit einem Punkt gekennzeichneten Zufallswerte werden bei jedem Aufruf der Operation getImputedVariableSet() neu gezogen. Das Ensemble-Objekt lässt sich auf diese Weise für jedes neuronale Netz, das es trainiert, neue vervollständigte Datensätze von inputImputator und targetImputator geben. Somit wird jedes neuronale Netz, das das Ensemble in seiner trainedNetworks-Liste speichert, auf einen anderen vollständigen Datensatz trainiert (siehe Abb. D-19, S. 215).

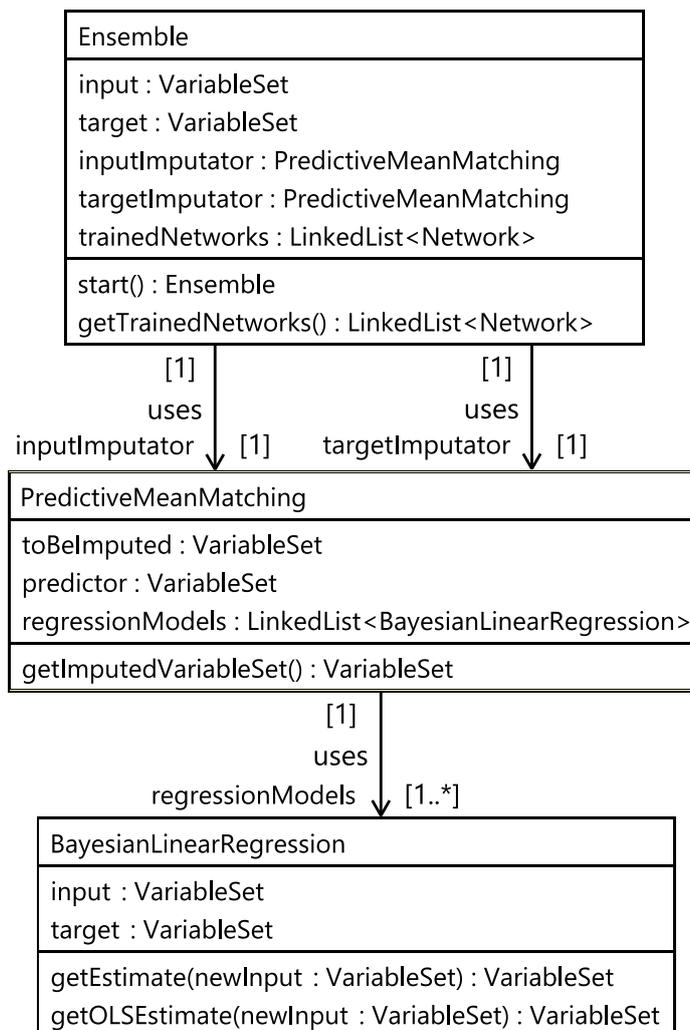


Abb. D-22: Klassendiagramm zur multiplen Imputation

Eine dynamische Ansicht dieser Interaktionen zeigen die Sequenzdiagramme in Abb. D-23 und Abb. D-24. Ein Sequenzdiagramm illustriert, wie Objekte im Zeitablauf erzeugt werden und welche Nachrichten sie austauschen. Ein Objekt wird als Rechteck dargestellt. Im Rechteck wird die Klassenzugehörigkeit hinter einem Doppelpunkt angegeben. Ein Objektname kann vor dem Doppelpunkt angeführt werden. Jedes Objekt hat eine Lebenslinie, auf der es Operationen ausführt sowie Nachrichten sendet und empfängt.¹⁾

In Abb. D-23 wird zunächst ein Ensemble erzeugt. Gleichzeitig werden ihm von der Benutzeroberfläche die dort ausgewählten Input- und Outputdatensätze übergeben (im Diagramm nicht dargestellt). Daraufhin erzeugt das Ensemble einen `inputImputator` und einen `targetImputator`. Dabei werden die Datensätze wechselseitig als Prädiktoren verwendet (siehe dazu S. 213). Für jede Variable des jeweiligen `toBelImputed`-Datensatzes erzeugen die beiden Imputatoren eine `BayesianLinearRegression` mit den beobachteten Werten der zu imputierenden Variablen als Zielvektor und den passenden Zeilen des Prädiktordatensatzes als Input. Anschließend rufen sie eine Kleinst-Quadrat-Schätzung für die beobachteten Werte ab (siehe Algorithmus D-2, Zeile 1).

Wenn die Ensemblebildung (in der Regel von der Benutzeroberfläche) über die `start()`-Operation gestartet wird, erzeugt und trainiert das Ensemble in einer Schleife mehrere neuronale Netze (siehe Abb. D-24). Wie viele Netze erzeugt und trainiert werden, hängt vom `ensembleSize`-Parameter ab, der über ein Einstellungsmenü auf der Benutzeroberfläche geändert werden kann.

¹⁾ Zur Notation von Sequenzdiagrammen vgl. ausführlich ISO [UML Superstructure] 479 ff.; Seemann/von Gudenberg [Software-Entwurf] 79 ff.

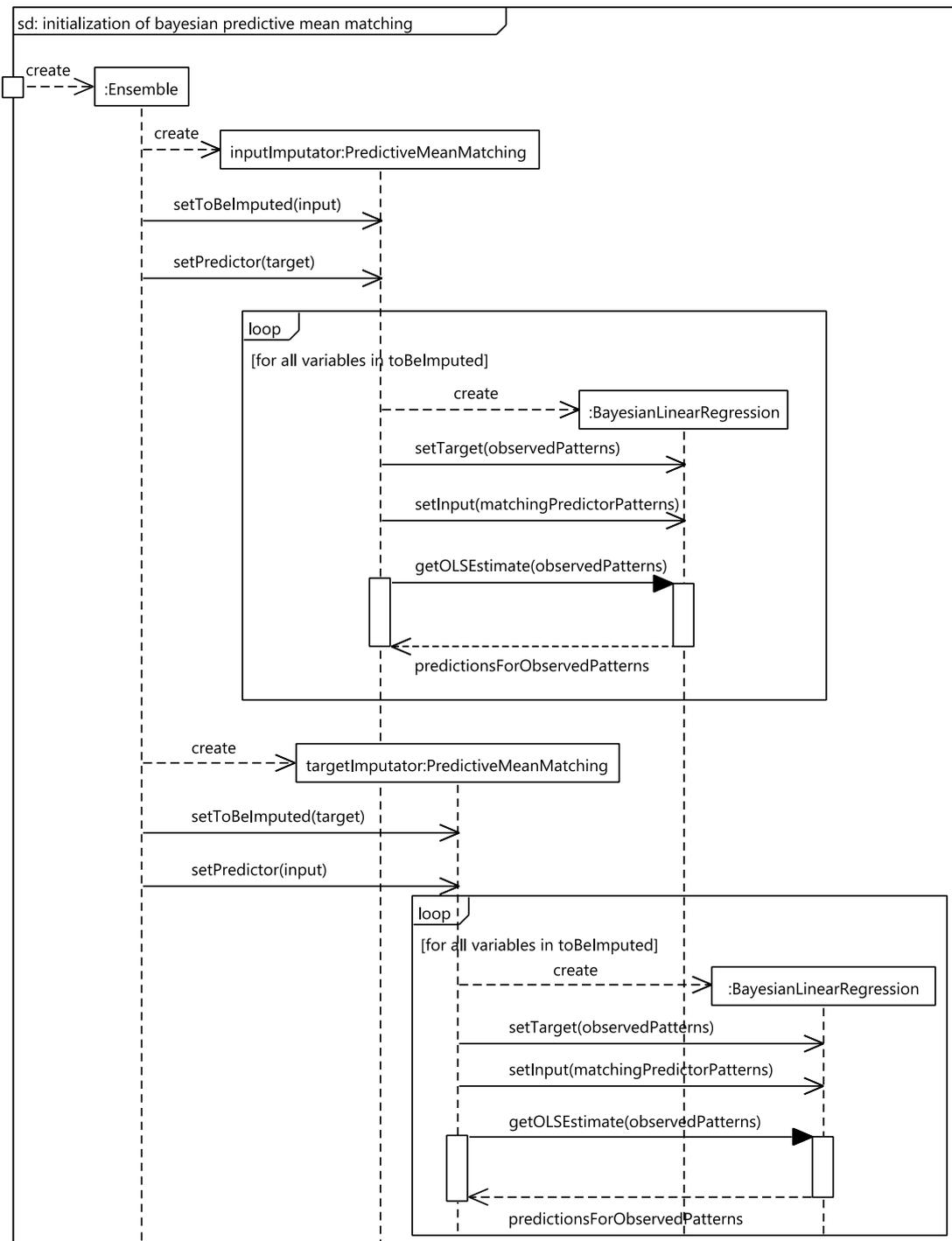


Abb. D–23: Sequenzdiagramm zur Initialisierung des predictive mean matching bei der Erzeugung eines neuen Ensemble-Objekts

In jedem Durchlauf ruft das Ensemble neue vollständige Datensätze von den Imputator-Objekten ab, die dafür wiederum eine neue bayessche Schätzung für die fehlenden Werte von den BayesianLinearRegression-Objekten anfordern (siehe Algorithmus D–2, Zeile 3). Damit vervollständigen die Imputa-

toren ihren toBelImputed-Datensatz wie in den Zeilen 4 bis 17 in Algorithmus D-2 beschrieben. Die vollständigen Datensätze übergibt das Ensemble einem Trainer, der damit ein neuronales Netz trainiert und optimiert. Das vom Trainer zurückgegebene trainierte Netz wird schließlich der trainedNetworks-Liste hinzugefügt. Die Sequenz in Abb. D-24 zeigt nur einen Teil der Ensemblebildung in Cenobi. Der vollständige Ablauf ist in Algorithmus D-4 (S. 295) dargestellt.

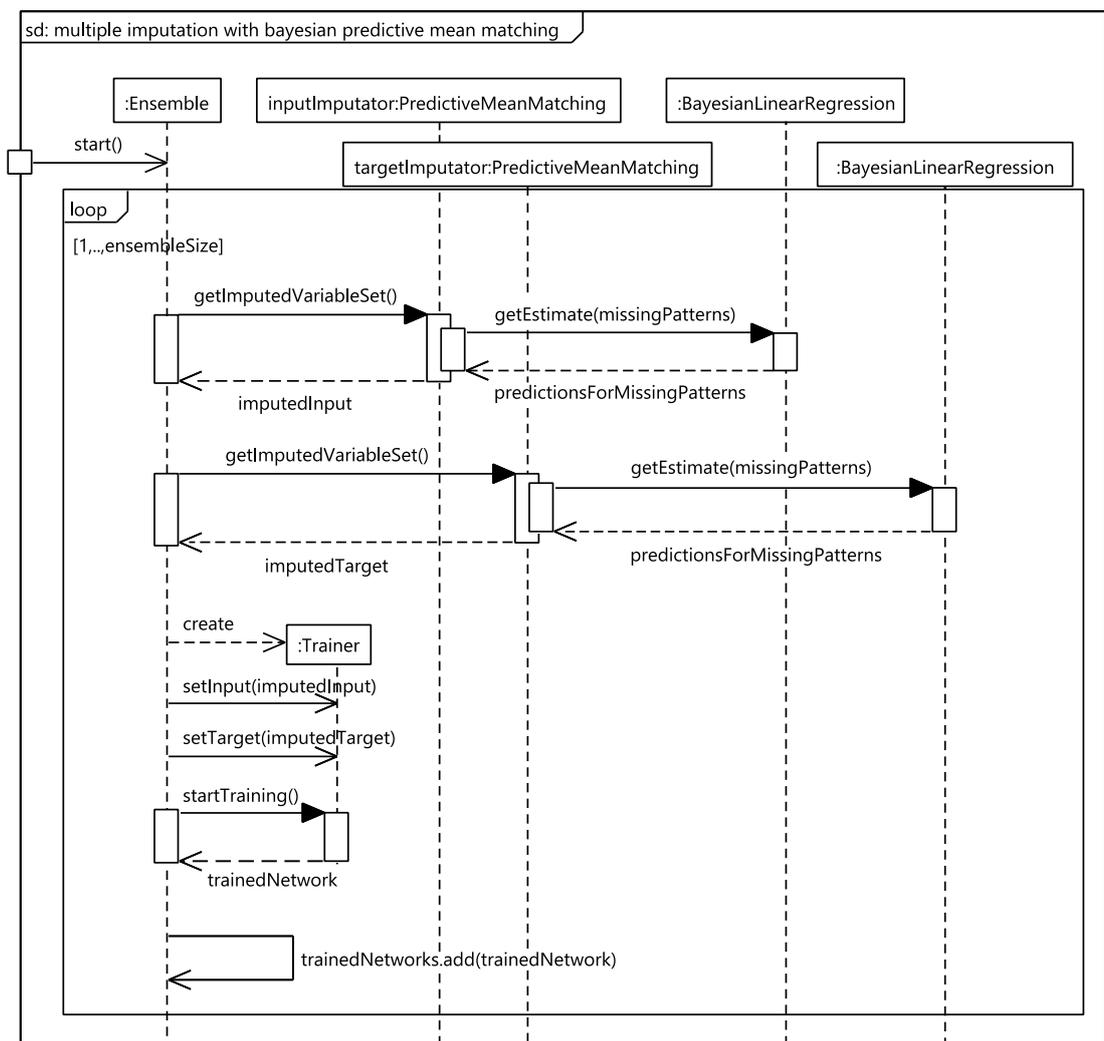


Abb. D-24: Sequenzdiagramm zur multiplen Imputation bei der Ensemblebildung

Bisher wurde davon abstrahiert, dass sowohl die zu vervollständigende Variable als auch ihre Prädiktoren unvollständig sein können. Wenn so-

wohl im zu imputierenden Datensatz als auch im Prädiktordatensatz sehr viele Werte fehlen, sind für eine gute Imputation aufwendige iterative Verfahren notwendig.¹⁾ Wenn aber im Prädiktordatensatz nur wenige Werte fehlen, reicht eine simple iterative Heuristik aus. Bei Regressionsproblemen wie der Schätzung von Kostenfunktionen kommen fehlende Werte unter den abhängigen Variablen nur selten vor. Deshalb ist die im folgenden Java-Codefragment gezeigte iterative Initialisierung, mit der das Ensemble-Objekt seine Imputatoren erzeugt, in der Regel ausreichend:

```
inputImputator = new PredictiveMeanMatching(input, target);
targetImputator = new PredictiveMeanMatching(target,
                                             inputImputator.getImputedVariableSet());
inputImputator = new PredictiveMeanMatching(input,
                                             targetImputator.getImputedVariableSet());
targetImputator = new PredictiveMeanMatching(target,
                                             inputImputator.getImputedVariableSet());
```

In der ersten Zeile werden einem neu erzeugten `PredictiveMeanMatching`-Objekt der `input`-Datensatz als zu imputierender Datensatz und der (möglicherweise unvollständige) `target`-Datensatz als Prädiktordatensatz übergeben. Der von diesem Objekt erzeugte vollständige Inputdatensatz wird wiederum in der zweiten Zeile einem `PredictiveMeanMatching`-Objekt als Prädiktor für den `target`-Datensatz übergeben usw. Wenn im ursprünglichen `target`-Datensatz nur wenige Werte fehlen, führt das wechselseitige `PredictiveMeanMatching` mit vervollständigten Datensätzen schon nach wenigen Iterationen zu guten Ergebnissen. Sollten in einem mehrvariablen `target`-Datensatz allerdings viele Beobachtungen fehlen, ist es sinnvoll, das Problem aufzuspalten und für jede abhängige Variable ein eigenes Modell zu erstellen, das nur die für diese Variable vorliegenden Beobachtungen als Zieldaten nutzt.

¹⁾ Vgl. van Buuren [Imputation] 95 ff.

The image shows a comparison between 'Datensatz Input' and 'Datensatz Target'. The input data has several missing values (indicated by -1) in columns like 'Anzahl Einkaufsrech...', 'Anzahl Kreditoren...', and 'Methodenzeitm...'. These missing values are replaced in the target data by imputed values, shown in red boxes with labels like 'imputiert: {511; 488; 512; 637}'. Red arrows indicate the mapping from the imputed values in the target back to the original input data.

Abb. D–25: Multiple Imputation im Beispieldatensatz

Abb. D–25 zeigt beispielhaft, welche Werte das in diesem Abschnitt vorgestellte Ensemble-Objekt in mehreren Durchläufen in die dargestellten Datensätze einfügt.

VII. Transformation der Daten

1. Normierung metrischer Daten

In Abschnitt C.I.3.e, Absatz (f) (S. 102 f.), wurde bereits das Problem der Neuronensättigung angesprochen: Die Ableitung der Aktivierungsfunktion in einem verdeckten Neuron ist annähernd Null ($f'_{\text{act}}(\text{net}_j^{(n)}) \approx 0$), wenn das Nettoeingangssignal $|\text{net}_j^{(n)}| > 2$ (bei der Tangens-Hyperbolicus-Funktion) bzw. $|\text{net}_j^{(n)}| > 5$ (bei der logistischen Funktion, siehe Abb. C–2, S. 85) ist.

Wenn viele Neuronen gesättigt sind, konvergiert der Trainingsalgorithmus nur noch langsam oder überhaupt nicht.

Der Trainingsprozess kann beschleunigt und eine Neuronensättigung vermieden werden, wenn die Eingabewerte $x_i^{(n)}$ auf den Teil des Definitionsbereichs der Aktivierungsfunktion normiert werden, in dem die erste Ableitung am größten ist. Das ist bei den üblichen Aktivierungsfunktionen – der Tangens-Hyperbolicus- und der logistischen Funktion – ein symmetrischer Bereich nahe bei Null.

Als besonders vorteilhaft hat sich die statistische Standardisierung (z -Transformation) einer Inputvariablen x_i zu einer standardisierten Variablen z_i mit Erwartungswert $E(z_i) = 0$ und Varianz $\text{Var}(z_i) = 1$ nach Formel (D.5)¹⁾ erwiesen.²⁾ Diese Normierung hat gegenüber einer linearen Transformation z. B. auf das Intervall $[-1; 1]$ den Vorteil, dass Ausreißer, die mehrere Standardabweichungen vom Mittelwert entfernt liegen, immer noch in die Extrembereiche der Aktivierungsfunktion gelangen, wo eine Änderung der absoluten Höhe des Eingangssignals kaum noch Auswirkungen auf das Ausgangssignal des Neurons hat. Das neuronale Netz wird dadurch robuster gegenüber Ausreißern.

Die Outputwerte müssen grundsätzlich auf den Wertebereich der Aktivierungsfunktion der Ausgabeneuronen normiert werden; bei Klassifikationsproblemen also auf das Intervall $]-1; 1[$ für die Tangens-Hyperbolicus- oder $]0; 1[$ für die logistische Funktion. Für Regressionsprobleme wird in der Ausgabeschicht üblicherweise die identische Abbildung (Wertebereich

¹⁾ Siehe S. 201. Siehe auch (D.2), S. 196; (D.1), S. 194.

²⁾ Vgl. auch im Weiteren Shanker/Hu/Hung [Standardization]; Coakley/Brown [Modeling Issues] 136; Kruse u. a. [Intelligence] 42.

] $-\infty; \infty$]) verwendet (siehe S. 86). Hier müssen demnach die Outputwerte nicht normiert werden.

Trotzdem ist bei Regressionsproblemen mit mehreren Outputvariablen eine Normierung sinnvoll, um die Schätzfehler vergleichbar zu machen. In der klassischen Zielfunktion für den Backpropagation-Algorithmus $E(w) = \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^O (\hat{y}_k^{(n)} - y_k^{(n)})^2$ (C.7) wird ein Schätzfehler – z. B. in Höhe von $(\hat{y}_k^{(n)} - y_k^{(n)}) = 100$ – immer gleich gewichtet, unabhängig davon, ob die Variable y_k Werte zwischen 0 und 100 oder zwischen 1.000.000 und 2.000.000 annimmt. In vielen Anwendungsfällen wird man den Fehler jedoch in Relation zur Breite des Wertebereichs der Variablen gewichten wollen. Das ist durch eine Normierung der Outputvariablen auf ein einheitliches Intervall z. B. durch lineare Skalierung möglich:

$$y_i^{\text{norm}(n)} = \frac{y_i^{(n)} - y_i^{\min}}{y_i^{\max} - y_i^{\min}}. \quad (\text{D.10})$$

Die normierten Werte $y_i^{\text{norm}(n)}$ nach (D.10) bewegen sich im Intervall $[0; 1]$. Im Allgemeinen ist eine Normierung auf dieses Intervall gemeint, wenn von *normierten Fehlern* die Rede ist (siehe (D.12), S. 253).

In Cenobi muss der Nutzer die Normierung nicht selbst vornehmen. Sie wird von den Neuronen in der Eingabe- bzw. Ausgabeschicht übernommen.¹⁾ Die Eingabeneuronen standardisieren die Eingabewerte nach Formel (D.5) (S. 201) und die Ausgabeneuronen skalieren die Ausgabewerte nach Formel (D.10). Die notwendigen Parameter – Mittelwert, Standardabweichung, Maximum und Minimum der Variablen – werden ihnen bei der Initialisierung des neuronalen Netzes übergeben.

¹⁾ Siehe Abschnitt VIII.1, S. 237 ff.

2. Kodierung nominaler und ordinaler Daten

Nicht-metrische Daten (kategoriale Daten i. e. S.) werden für das neuronale Netz in einen Binärkode überführt. Grundsätzlich könnte man für jede mögliche Ausprägung einer kategorialen Variablen ein eigenes Neuron einführen, das den Wert 1 annimmt, wenn die Ausprägung vorliegt und den Wert 0, wenn nicht (n -Kodierung, siehe Abb. D-26). Diese Kodierung benötigt aber unnötig viele Neuronen, die mit zusätzlichen Parametern einhergehen, die geschätzt werden müssen. Am sparsamsten wäre in dieser Hinsicht eine Kodierung in Dualzahlen. Dadurch würden die Ausprägungen allerdings in eine Rangfolge gebracht, was bei nominalskalierten Variablen nicht erwünscht ist.

	n -Kodierung	$\binom{n}{k}$ -Kodierung
Januar	(1,0,0,0,0,0,0,0,0,0,0)	(1,1,0,0,0,0)
Februar	(0,1,0,0,0,0,0,0,0,0,0)	(1,0,1,0,0,0)
⋮	⋮	⋮
November	(0,0,0,0,0,0,0,0,0,1,0)	(0,0,1,0,1,0)
Dezember	(0,0,0,0,0,0,0,0,0,0,1)	(0,0,1,0,0,1)

Abb. D-26: Binärkodierung nominaler Ausprägungen

Um die Bildung einer Rangfolge zu verhindern, sollte jede Ausprägung einer nominalskalierten Variablen durch dieselbe Anzahl an Einsen und Nullen repräsentiert werden. Bei der Frage, wie viele Binärstellen für diese Kodierung mindestens notwendig sind, hilft der Binomialkoeffizient $\binom{n}{k}$.¹⁾ Er gibt an, auf wie viele Arten man k Elemente aus einer Menge von n Elementen auswählen kann. Für die Binärkodierung lässt sich mit ihm berechnen, auf wie viele Arten man k Einsen auf n Binärstellen verteilen kann. Möchte man beispielsweise die zwölf Kalendermonate mit möglichst wenigen Bi-

¹⁾ Vgl. auch im Weiteren Lämmel/Cleve [Intelligenz] 233.

närstellen kodieren, sucht man einen Binomialkoeffizienten mit minimalem n , der mindestens zwölf Kombinationsmöglichkeiten ergibt. Das ist z. B. für $\binom{6}{2} = 15$ der Fall (siehe Abb. D-26).

Bei ordinalskalierten Variablen soll hingegen die Rangfolge der Ausprägungen erhalten bleiben. Bei der sogenannten Thermometer-Kodierung wird die Rangfolge durch die Anzahl der Einsen wiedergegeben (siehe Abb. D-27).¹⁾ Da hier die erste Stelle bei allen Ausprägungen gleich ist, kann man sie ohne Informationsverlust streichen ($n-1$ -Kodierung).

	Thermometer-Kodierung	$n-1$ -Kodierung
schlecht	(1,0,0)	(0,0)
mittel	(1,1,0)	(1,0)
gut	(1,1,1)	(1,1)

Abb. D-27: Binärkodierung ordinaler Ausprägungen

Der Algorithmus zur $n-1$ -Kodierung ordinaler Variablen ist unkompliziert und muss hier nicht näher erläutert werden. Bei der $\binom{n}{k}$ -Kodierung nominaler Variablen gibt es hingegen zwei Schwierigkeiten. Erstens muss eine $n-k$ -Kombination mit minimalem n gefunden werden, die eine Kodierung aller Ausprägungen erlaubt.

Dabei kann man sich den Zusammenhang des pascalschen Dreiecks

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \quad (\text{D.11})$$

zunutze machen und ausgehend von $\binom{n}{0} = \binom{n}{n} = 1$ iterativ alle Binomialkoeffizienten nach (D.11) berechnen, bis $\binom{n}{k}$ zum ersten Mal mindestens so groß wie die Anzahl der zu kodierenden Ausprägungen ist.

¹⁾ Vgl. auch im Weiteren Lämmel/Cleve [Intelligenz] 233 f.; Crone/Lessmann/Stahlbock [Preprocessing] 788 ff.

Zweitens wird für jede Ausprägung eine einzigartige Kombinationen aus k Einsen und $(n - k)$ Nullen benötigt. Einen schlanken Algorithmus, der alle $\binom{n}{k}$ Kombinationen von k Elementen aus einer n -elementigen Menge in lexikographischer Ordnung (wie in Abb. D–26 und Abb. D–30 dargestellt) erzeugt, hat z. B. McCaffrey formuliert.¹⁾ Dieser wird in Cenobi genutzt.

In Cenobi werden Variable automatisch kodiert, wenn der Nutzer sie im Importdialogfenster als nominal oder ordinal skaliert kennzeichnet (siehe Abb. D–28). Dabei wird für jede nicht-metrische Variable eine Kodiertabelle angelegt (siehe Abb. D–29) und bei nachfolgenden Importen derselben Variablen wieder aufgerufen. Im Datensatz selbst erscheinen nicht-metrische Variable nur in kodierter Form (siehe Abb. D–30).

The image shows a software dialog box for importing data from an Excel file. The file path is 'D:\Kategorial.xlsx' and the sheet is '1'. The range is 'A1' to 'D13'. Two checkboxes are checked: 'Variable in Spalten (Werte sind von oben nach unten abgetragen)' and 'erste Reihe bzw. Spalte enthält die Variablennamen'. A section titled 'Datensatz enthält nicht-metrische Variable' contains two input fields: '1, 2' for nominal scaling and '3, 4' for ordinal scaling. Below this is a warning section 'Hinweis zu nicht-metrischen Variablen' with text explaining that a complete set of values should be imported for automatic coding to work. The dataset name is 'Kategorial'. Buttons for 'Importieren' and 'Abbrechen' are at the bottom.

Abb. D–28: Kennzeichnung nicht-metrischer Variablen im Cenobi-Importdialog

¹⁾ Vgl. McCaffrey [Combinations], mit weiteren Nachweisen.

Datensatz Kodiertabelle Farbe (nominal skaliert)

	Farbe	Farbe Dummy1	Farbe Dummy2	Farbe Dummy3	Farbe Dummy4
1	1	1	0	0	0
2	2	0	1	0	0
3	3	0	0	1	0
4	4	0	0	0	1
5	5	1	1	1	0
6	6	0	0	1	1

Datensatz Kodiertabelle Schulnote (ordinal skaliert)

	Schulnote	Schulnote Dummy1	Schulnote Dummy2	Schulnote Dummy3	Schulnote Dummy4	Schulnote Dummy5
1	1	0	0	0	0	0
2	2	1	0	0	0	0
3	3	0	1	0	0	0
4	4	0	0	1	0	0
5	5	1	1	1	0	0
6	6	0	1	1	1	1

Abb. D-29: Kodiertabellen (mappings) für nicht-metrische Variable

Datensatz Spritzguss

	Ro...	ge...	Zu...	An...	An...	U...	U...	U...	...	An...	An...	An...	An...	Fr...	Fr...	Fr...	Monat Dummy1	Mo...	M...	M...	
1	-1	46...	20...	1.057	2.563	13...	2.0...	465...	59	483	-1	3.413	1.033	637...	3.2...	45...	32...	1	1	0	0	0	0
2	-1	50...	20...	1.021	2.466	12...	1.7...	499...	57	452	-1	3.092	825	697...	1.7...	32...	70...	1	0	1	0	0	0
3	-1	47...	22...	992	2.563	12...	1.2...	329...	66	419	448	2.738	812	716...	1.1...	37...	25...	1	0	0	1	0	0
4	-1	49...	22...	1.063	2.949	8.6...	1.5...	328...	69	440	476	2.852	931	824...	1.6...	40...	63...	1	0	0	0	1	0
5	1.501	45...	22...	1.097	2.990	10...	2.1...	167...	69	415	474	3.688	883	680...	1.0...	61...	0	1	0	0	0	0	1
6	1.611	47...	22...	1.101	3.163	15...	1.3...	369...	73	478	491	3.819	990	688...	1.1...	47...	52...	0	1	1	0	0	0
7	1.341	48...	22...	1.135	3.243	11...	1.4...	194...	74	459	398	4.219	990	708...	1.0...	10...	29...	0	1	0	1	0	0
8	1.570	42...	22...	1.083	3.051	11...	1.5...	333...	67	408	343	4.724	860	691...	847...	28...	0	0	1	0	0	1	0
9	1.511	45...	22...	1.182	3.181	13...	1.6...	552...	67	381	398	5.302	1.121	821...	956...	15...	44...	0	1	0	0	0	1
10	1.846	52...	22...	1.166	3.075	10...	1.5...	326...	78	417	395	4.662	959	792...	581...	20...	31...	0	0	1	1	0	0
11	1.949	49...	22...	1.043	2.712	9.4...	1.9...	388...	74	419	346	3.842	907	733...	433...	46...	18...	0	0	1	0	1	0
12	1.978	46...	22...	1.099	3.028	12...	1.6...	53...	62	462	351	7.931	1.511	760...	282...	34...	27...	0	0	1	0	0	1
13	1.910	48...	22...	1.149	3.113	15...	1.6...	196...	61	401	433	4.887	936	808...	513...	26...	20...	1	1	0	0	0	0
14	1.488	47...	22...	1.076	2.947	10...	1.4...	262...	73	424	417	3.857	827	774...	682...	27...	18...	1	0	1	0	0	0
15	1.585	50...	22...	1.168	3.285	12...	1.3...	286...	62	412	602	4.241	1.111	808...	602...	79...	21...	1	0	0	1	0	0
16	1.523	46...	22...	1.146	3.220	11...	1.5...	169...	74	440	427	3.755	786	718...	620...	80...	14...	1	0	0	0	1	0
17	1.641	51...	22...	1.218	3.264	11...	1.8...	640...	71	452	528	4.390	907	927...	1.1...	37...	0	1	0	0	0	0	1
18	1.625	51...	22...	1.215	3.264	14...	1.3...	342...	66	467	531	4.397	884	893...	1.0...	42...	0	0	1	1	0	0	0
19	1.637	52...	22...	1.165	3.290	14...	3.2...	517...	71	424	576	4.598	1.083	869...	1.2...	13...	35...	0	1	0	1	0	0
20	1.786	52...	22...	1.210	3.285	10...	1.7...	186...	66	441	478	4.525	1.021	895...	1.0...	37...	0	0	1	0	0	1	0

Abb. D-30: Einflussgrößen des Spritzguss-Datensatzes mit Kalendermonat in Binärkode

VIII. Objektorientierte Implementierung eines künstlichen neuronalen Netzes

1. Klassenstruktur der Netzelemente

Mit den Formeln (C.4) und (C.6) (S. 87 f.) wurde gezeigt, dass sich ein künstliches neuronales Netz in einer einzigen Gleichung beschreiben lässt. Damit ist es möglich, ein künstliches neuronales Netz zu implementieren, ohne die mehrschichtige Struktur des Netzes und die Signalverarbeitung in den Neuronen und Synapsen explizit im Programm nachzubilden. Die Eingangs- und Ausgangssignale der Neuronen sowie die aktuellen Gewichte werden dabei in Vektoren oder Matrizen gespeichert.¹⁾ Dieser Ansatz hat den Vorteil, dass sich ein einfacher Backpropagation-Algorithmus schon mit relativ wenigen Programmzeilen umsetzen lässt.

Demhingegen benötigt eine objektorientierte Umsetzung der Netzstruktur einige Programmzeilen mehr, dafür ist die Implementierung anschaulicher – und dadurch leichter nachvollziehbar und überprüfbar – sowie vor allem flexibler und die einzelnen Programmteile sind besser wiederverwendbar. Beispielsweise ist es mit einer objektorientierten Implementierung ohne weiteres möglich, während des Trainings einzelne Synapsen nach bestimmten Kriterien zu löschen, um die Komplexität des Modells dynamisch anzupassen.²⁾ Und die Netzelemente – Neuronen, Synapsen und Schichten – lassen sich ohne spezifische Anpassungen in verschiedenen Netzarchitektu-

¹⁾ Vgl. z. B. die Matlab-Implementierung Netlab (Nabney [Netlab] 149 ff.) sowie die Java-Implementierungen WEKA (Bouckaert u. a. [WEKA] 2538) und Encog (Heaton [Encog]).

²⁾ Diese Technik nennt sich *weight pruning* (vgl. Le Cun/Denker/Solla [Brain Damage]; Reed [Pruning]).

ren und Trainingsalgorithmen wiederverwenden. Für Cenobi wurde daher eine durchgängig objektorientierte Implementierung gewählt.

Abb. D-31 zeigt die Elemente des implementierten künstlichen neuronalen Netzes in einem Klassendiagramm. Im ersten Fach eines Klassenrechtecks steht der Klassenname, im zweiten stehen die Attribute und im dritten die Operationen, die ein Objekt dieser Klasse besitzt. Zwischen den Klassen gibt es Assoziationen. Eine Assoziation mit einer ausgefüllten Raute ist eine Komposition, d. h. das durch eine Raute markierte Objekt besteht aus den Objekten an der Pfeilspitze.¹⁾ Die Teilobjekte können nur innerhalb dieser Komposition existieren und verschwinden, wenn die Komposition vernichtet wird. Pfeile ohne Rauten drücken weniger strenge Assoziationen aus. Die Pfeilspitzen kennzeichnen die Zugriffsrechte: Das Objekt am anderen Pfeilende kann Nachrichten an das Objekt an der Pfeilspitze senden, das daraufhin eine Operation ausführt und ggf. ein Ergebnis zurückmeldet. Auch bidirektionale Beziehungen mit Pfeilspitzen an beiden Seiten sind möglich, erfordern aber besondere Synchronisationsmaßnahmen. Die ausgefüllten Dreiecke neben dem Namen einer Assoziation geben die Leserichtung an. Um die Diagramme übersichtlich zu halten, sind in Abb. D-31 wie auch in den folgenden Klassendiagrammen nur die wichtigsten Attribute und Operationen aufgeführt. Auch die konventionellen Operationen für den Zugriff auf Objektattribute – die sogenannten *getter* und *setter*, z. B. `getLayers()` und `setLayers()` für den Zugriff auf die Schichten eines `SimpleNetwork` – sind ausgeblendet.

¹⁾ Vgl. auch im Weiteren ISO [UML Superstructure] 42 ff.; Seemann/von Gudenberg [Software-Entwurf] 56 ff.

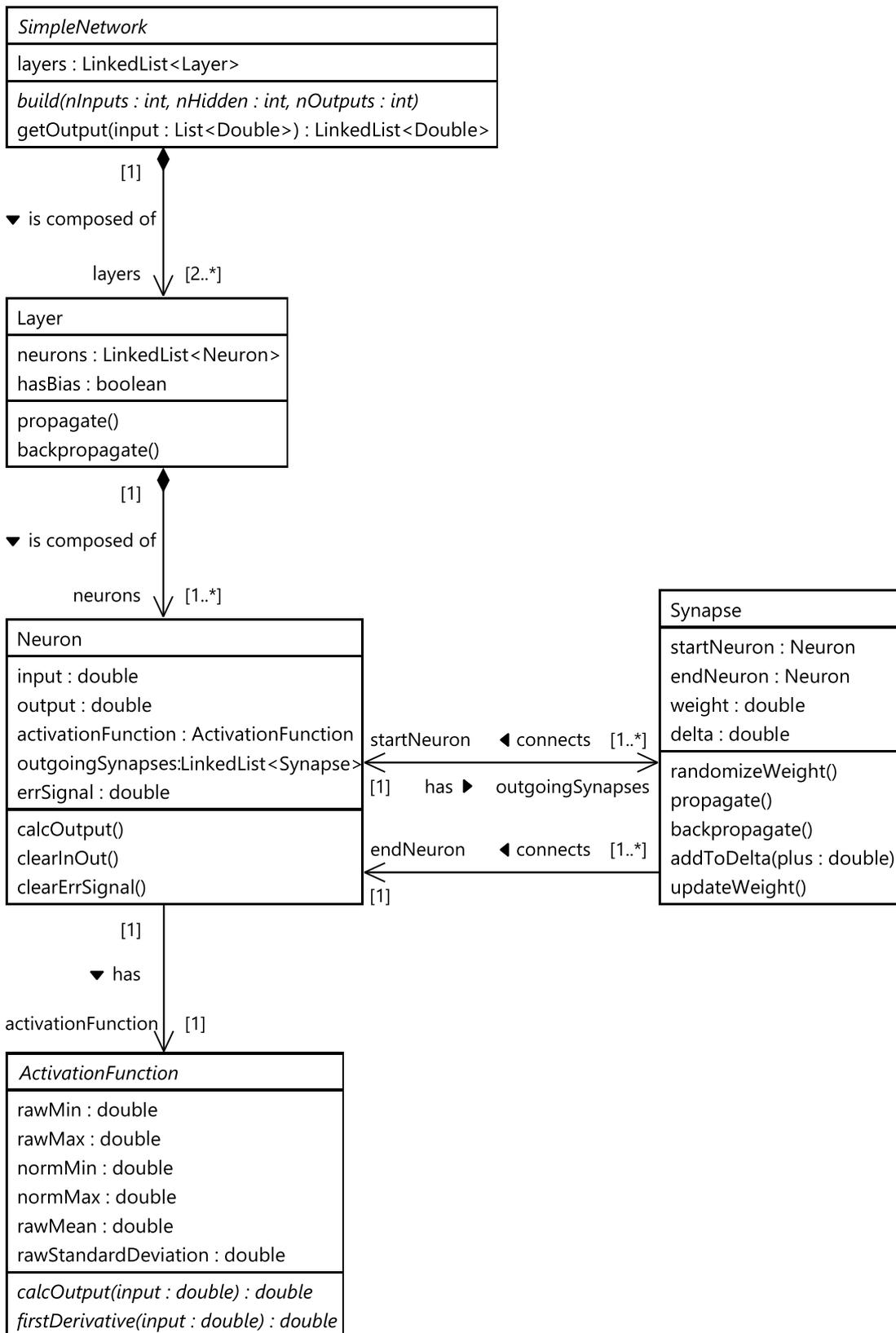


Abb. D–31: Klassendiagramm der Netzelemente

In Abb. D-31 ist zu sehen, dass ein SimpleNetwork aus mindestens zwei Schichten (Layer) besteht, die in einer verketteten Liste (LinkedList) mit der Bezeichnung layers abgelegt sind. Eine Schicht besteht wiederum aus mindestens einem Neuron. Jedes Neuron besitzt ein Eingangssignal (input), ein Ausgangssignal (output), eine Aktivierungsfunktion (activationFunction), eine Liste der ausgehenden Synapsen (outgoingSynapses) und ein Fehlersignal (errSignal).

Eine Synapse verbindet jeweils ein startNeuron mit einem endNeuron und kann auf beide Neuronen zugreifen. Ein Neuron kann beliebig viele eingehende und ausgehende Synapsen haben. In der Ausgabeschicht ist die Liste der outgoingSynapses leer. Grundsätzlich reicht es aus, dass ein Neuron nur seine outgoingSynapses kennt. Will man aber beispielsweise in einer Metaheuristik verdeckte Neuronen dynamisch hinzufügen und löschen, kann es sinnvoll sein, das Neuron auch auf seine eingehenden Synapsen zugreifen zu lassen, um sowohl eingehende als auch ausgehende Synapsen zusammen mit dem Neuron zu löschen.

Die Klasse ActivationFunction ist abstrakt, d. h. sie dient nur dazu, gemeinsame Attribute und Operationen ihrer konkreten Unterklassen zu definieren. Abb. D-32 zeigt, welche konkreten Aktivierungsfunktionen in Cenobi implementiert sind. Pfeile mit einem unausgefüllten Dreieck als Spitze symbolisieren eine Vererbungsbeziehung. Alle konkreten Aktivierungsfunktionen erben die abstrakten Operationen calcOutput() und firstDerivative(), die sie jeweils unterschiedlich umsetzen, sowie sechs Attribute. In den Attributen rawMin, rawMax, rawMean und rawStandardDeviation sind die Lage- und Verteilungsparameter derjenigen Variablen gespeichert, die das Neuron verarbeitet, das die activationFunction besitzt. Damit kann ein Normalization-Objekt Rohwerte auf das Intervall [normMin; normMax] normieren bzw. ein

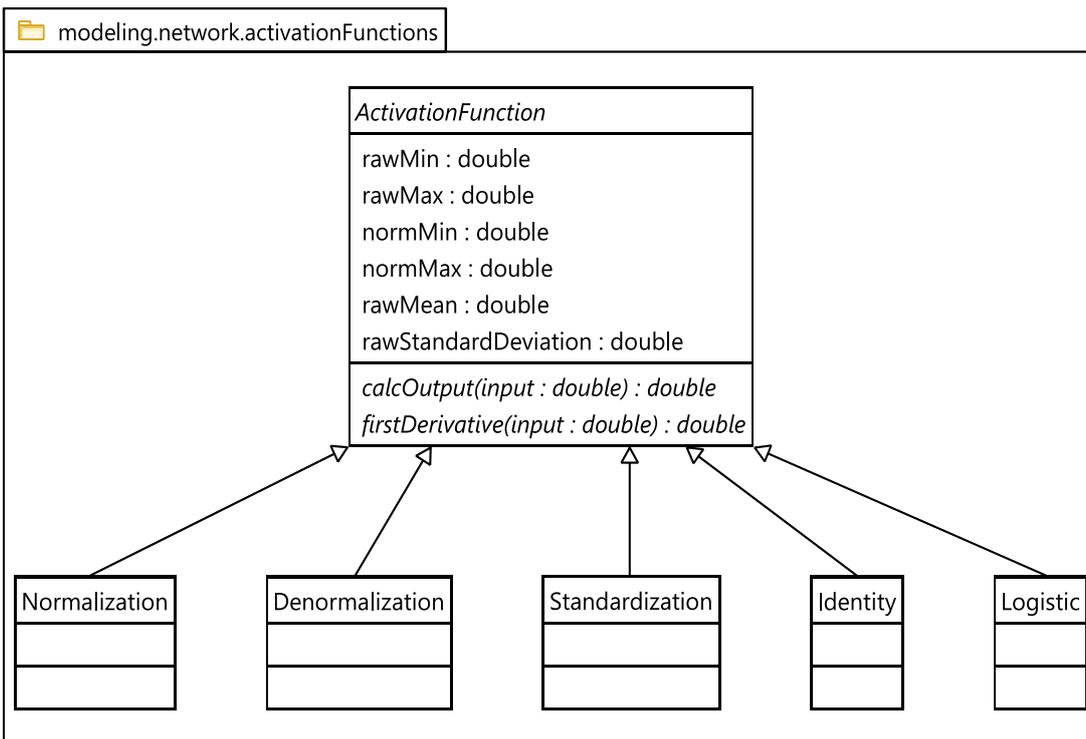


Abb. D-32: Klassendiagramm der Aktivierungsfunktionen

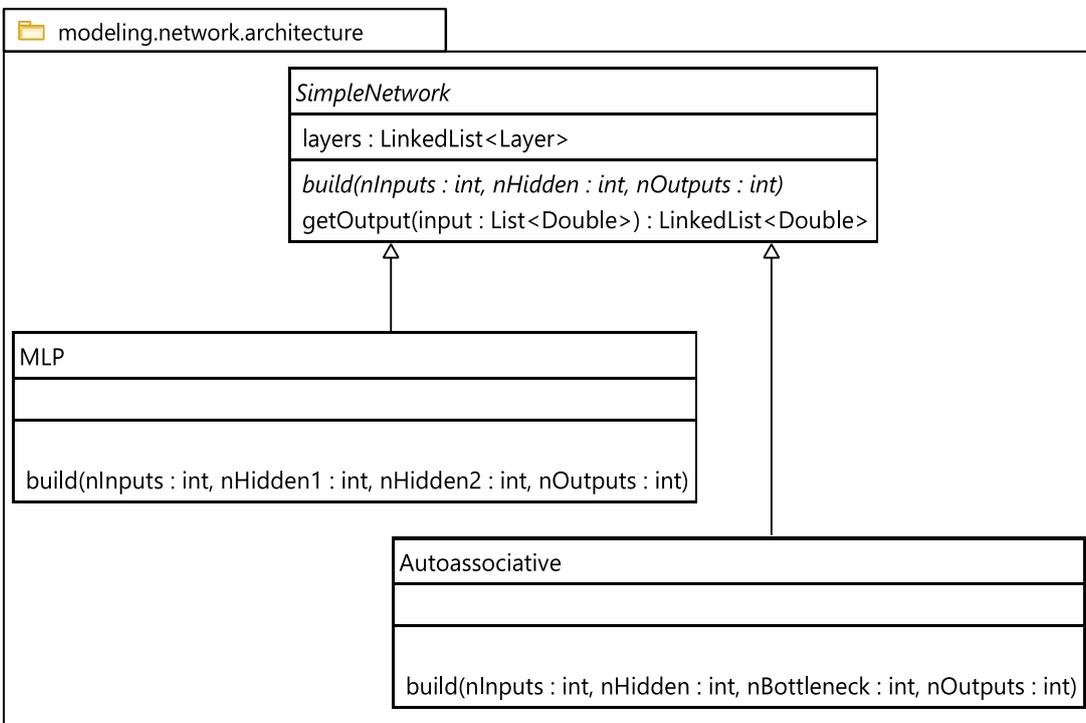


Abb. D-33: Klassendiagramm der Netzarchitekturen

Denormalization-Objekt normalisierte Werte auf das Intervall [rawMin; rawMax] zurücktransformieren. Ein Standardization-Objekt standardisiert input-Werte nach Formel (D.5) (S. 201). Normalization- und Standardization-Funktionen werden in der Eingabeschicht eines neuronalen Netzes verwendet, Denormalization-Funktionen in der Ausgabeschicht.

Auch die Klasse SimpleNetwork ist abstrakt. Zwei konkrete Unterklassen sind in Abb. D-33 dargestellt. Sie unterscheiden sich nur darin, wie sie die abstrakte Funktion build() umsetzen, in der die Neuronenschichten erzeugt und durch Synapsen verknüpft werden. Die MLP-Klasse bietet außerdem eine zweite build()-Funktion an, um neuronale Netze mit zwei verdeckten Schichten zu erzeugen. Auch die Autoassociative-Klasse besitzt eine zusätzliche build()-Funktion, mit der Flaschenhalsschichten unterschiedlicher Größe erzeugt werden können.¹⁾

2. Vorwärtsgerichtete Signalverarbeitung im künstlichen neuronalen Netz

Das Sequenzdiagramm in Abb. D-34 zeigt die Interaktionen der in Abb. D-31 vorgestellten Objekte bei der vorwärtsgerichteten Propagation eines Eingabevektors durch das neuronale Netz:

- (1) Die Propagation wird durch einen Aufruf der Operation getOutput() eines SimpleNetwork in Gang gesetzt. Der Aufruf kann beispielsweise von einem Trainingsalgorithmus stammen. Der Aufrufer übergibt einen Inputvektor input in Form einer Liste mit Gleitkommazahlen.²⁾

¹⁾ Siehe Abschnitt VI.1.b, S. 203 ff.

²⁾ Das Schlüsselwort Double bzw. double steht für Gleitkommazahlen mit doppelter Genauigkeit.

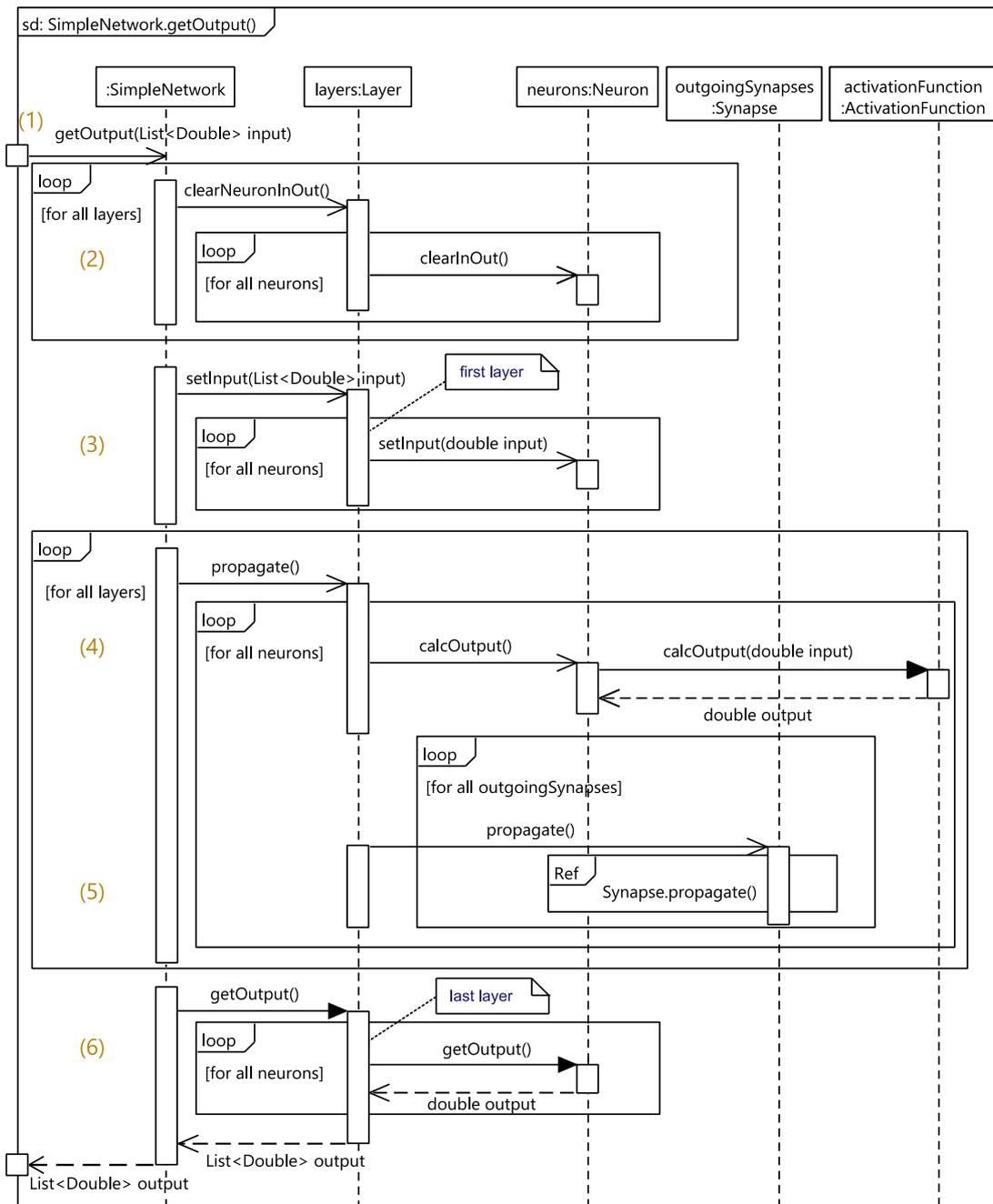


Abb. D-34: Sequenzdiagramm der Vorwärtspropagation eines Eingavektors im neuronalen Netz

- (2) Das SimpleNetwork initialisiert zunächst in allen Schichten die Eingangs- und Ausgangssignale aller Neuronen.
- (3) Anschließend gibt es den input-Vektor an die Neuronen der ersten Netzschicht weiter.

- (4) Dieses Signal wird dann nacheinander durch alle Netzschichten propagiert. Dabei berechnen zuerst alle Neuronen mit ihrer Aktivierungsfunktion aus dem gegebenen Eingangssignal ein Ausgangssignal.
- (5) Das Ausgangssignal wird über die ausgehenden Synapsen an die Neuronen der folgenden Schicht weitergeleitet (siehe dazu Abb. D-35).

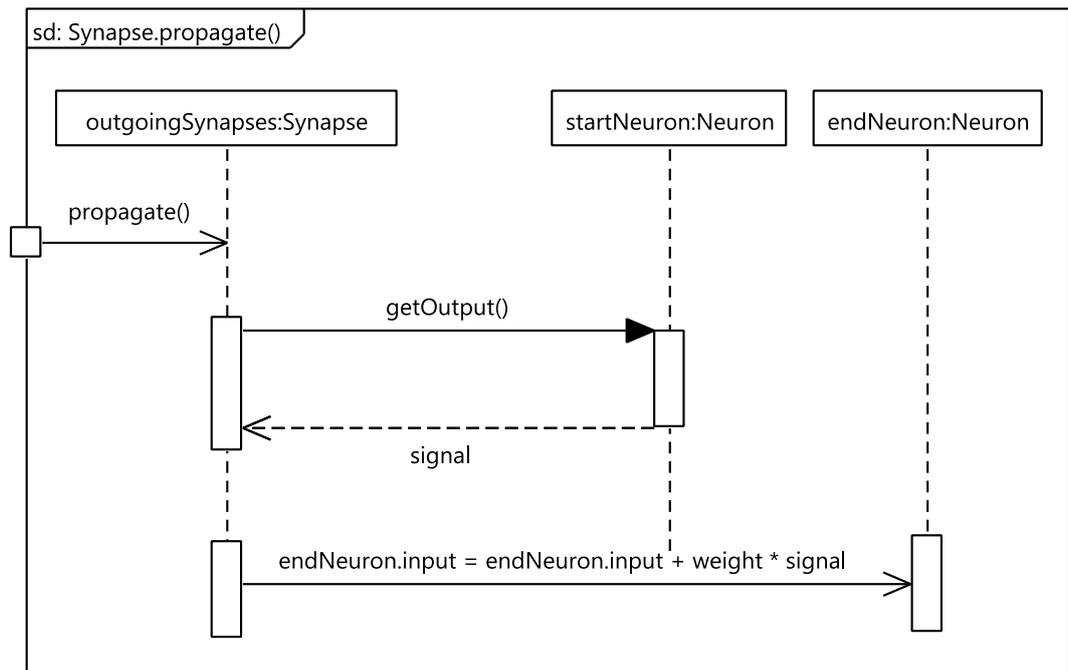


Abb. D-35: Sequenzdiagramm der Vorwärtspropagation einer Synapse

- (6) Nachdem das Signal durch alle Schichten propagiert wurde, ruft das `SimpleNetwork` die Ausgangssignale der Neuronen in der letzten Schicht ab, fasst sie zu einer Liste zusammen und gibt diese Liste als Ergebnis der Operation `getOutput()` zurück.

Schwellenwert-Neuronen (*bias neurons*) haben ein konstantes Ausgangssignal (siehe S. 87 f.) und werden deshalb von den Schritten (2), (3) und (4) ausgenommen. Sie nehmen demnach nur am Propagationsschritt (5) teil.

IX. Objektorientierte Umsetzung der Trainingsalgorithmen

1. Struktur der Trainingsklassen

In Cenobi sind der klassische Backpropagation-Algorithmus als Grundlage und Referenz sowie die in Vergleichsexperimenten besonders erfolgreichen Algorithmen *resilient propagation* und *extreme learning machine* implementiert.¹⁾ Das Klassendiagramm in Abb. D-36 zeigt die Struktur der Trainingsklassen. Die Klasse `TrainingSettings` speichert die auf der Benutzeroberfläche getroffenen Trainingseinstellungen (siehe Abb. D-37). Das Attribut `trainingType` legt die Art und `errType` die Zielfunktion des Trainingsalgorithmus fest. Die folgenden drei Attribute sind Abbruchkriterien. Der Wahrheitswert `onlineLearning` bestimmt, ob die Synapsengewichte nach jedem Trainingspunkt angepasst werden (*online learning*) oder erst nachdem der gesamte Trainingsdatensatz betrachtet wurde (*batch learning*).²⁾ Wenn `earlyStopping` wahr ist, wird der Trainingsalgorithmus bei steigendem Validierungsfehler abgebrochen, um *overfitting* zu vermeiden.³⁾ Die übrigen Attribute sind spezifische Parameter des klassischen Backpropagation-Algorithmus.⁴⁾

¹⁾ Siehe Abschnitte C.I.3.d, S. 99 ff., C.I.3.g.β, S. 111 ff., sowie C.I.3.i, S. 118 ff.

²⁾ Siehe Abb. D-38. Siehe auch S. 99.

³⁾ Siehe ausführlich Teilkapitel XI.2, S. 261 ff.

⁴⁾ Siehe Formel (C.24), S. 110.

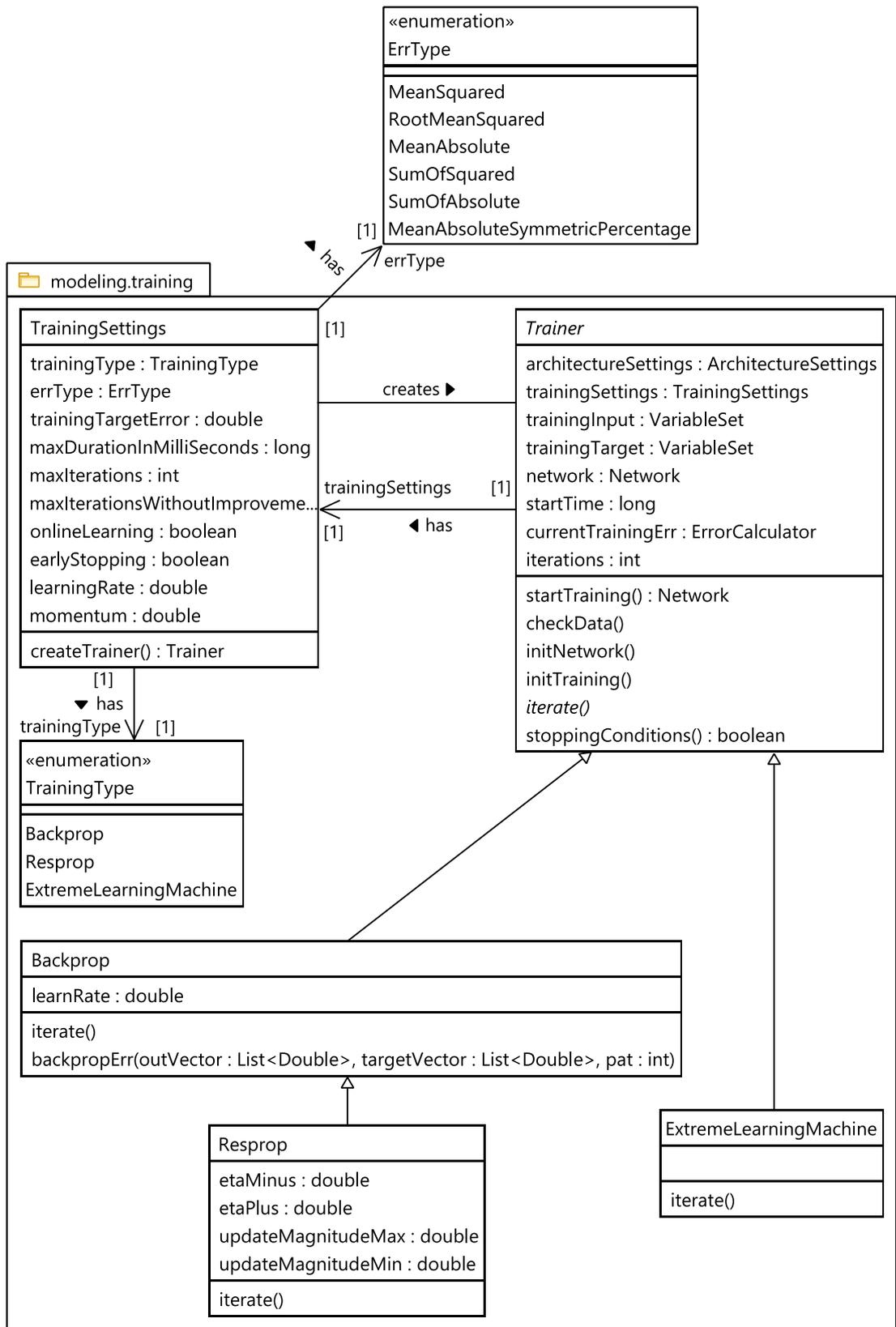


Abb. D-36: Klassendiagramm der Trainingsalgorithmen

Lernalgorithmus	Backprop
Zielfunktion	MeanAbsolute
Early Stopping	<input type="checkbox"/>
Lernrate	0.1
Impulsterm	0.8
Online Learning	<input checked="" type="checkbox"/>
Abbruchkriterien	
Trainingsfehler kleiner als	0.01
maximale Dauer in Sek.	40
maximale Anzahl Iterationen gesamt	3000
max. aufeinanderfolgende Iterationen ohne Verbesserung des Trainingsfehlers	20
Speichern Abbrechen Standardeinstellungen ?	

Abb. D-37: Einstellungsmenü der Trainingsalgorithmen

Die Operation `createTrainer()` eines `TrainingSettings`-Objekts liefert ein konkretes `Trainer`-Objekt der Art `trainingType` zurück, das das `TrainingSettings`-Objekt selbst als Attribut enthält und so auf die Einstellungen zugreifen kann. Die abstrakte Klasse `Trainer` definiert die gemeinsamen Attribute und Operationen aller Trainingsalgorithmen. Gemeinsame Attribute sind neben den `trainingSettings` vor allem die Trainingsdatensätze und das zu trainierende neuronale Netz. Die gemeinsamen Operationen sind in erster Linie Initialisierungen sowie die Überprüfung der Abbruchbedingungen. Die abstrakte Methode `iterate()` wird hingegen in jeder Unterklasse individuell umgesetzt.

2. Objektorientierte Umsetzung des Backpropagation-Algorithmus

Das Sequenzdiagramm in Abb. D-38 zeigt einen Iterationsschritt, wie er in der Klasse `Backprop` implementiert ist. Die gesamte Sequenz entspricht den Schritten 4 bis 9 in Algorithmus C-1 (S. 96):

- (1) Das `Backprop`-Objekt fragt zunächst einen Trainingsdatensatz, wie viele Trainingspunkte $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$ vorliegen und erhält den Wert `nPat` zurück (`nPat` ist die Anzahl der Beobachtungen N).
- (2) In einer Schleife ruft das `Backprop`-Objekt alle Trainingspunkte ab und lässt sich jeweils für den Inputvektor $\mathbf{x}^{(n)}$ einen Outputvektor $\hat{\mathbf{y}}^{(n)}$ vom zu trainierenden `network` berechnen (siehe dazu Abb. D-34).
- (3) Vom Datensatz `trainingTarget` erhält es den Zielvektor $\mathbf{y}^{(n)}$.
- (4) Mit diesen Vektoren berechnet der Trainingsalgorithmus die Fehlersignale der Neuronen und die Gewichtsänderungen Δw durch Fehlerrückpropagation.¹⁾ Der Ablauf der Fehlerrückpropagation ist in Abb. D-39 ausführlich dargestellt.
- (5) Je nach gewählter Einstellung werden die Gewichte nach jedem Trainingspunkt angepasst (`onlineLearning == true`, siehe (C.20), S. 99) oder erst nachdem alle Trainingspunkte betrachtet wurden (`onlineLearning == false`, siehe (C.19), S. 99). Letzteres entspricht der Darstellung in Algorithmus C-1.

Die `iterate()`-Sequenz wird vom Trainingsalgorithmus so lange wiederholt, bis ein Abbruchkriterium erfüllt ist (siehe Algorithmus C-1, Zeile 10).

¹⁾ Siehe C.I.3.c, S. 96 ff.

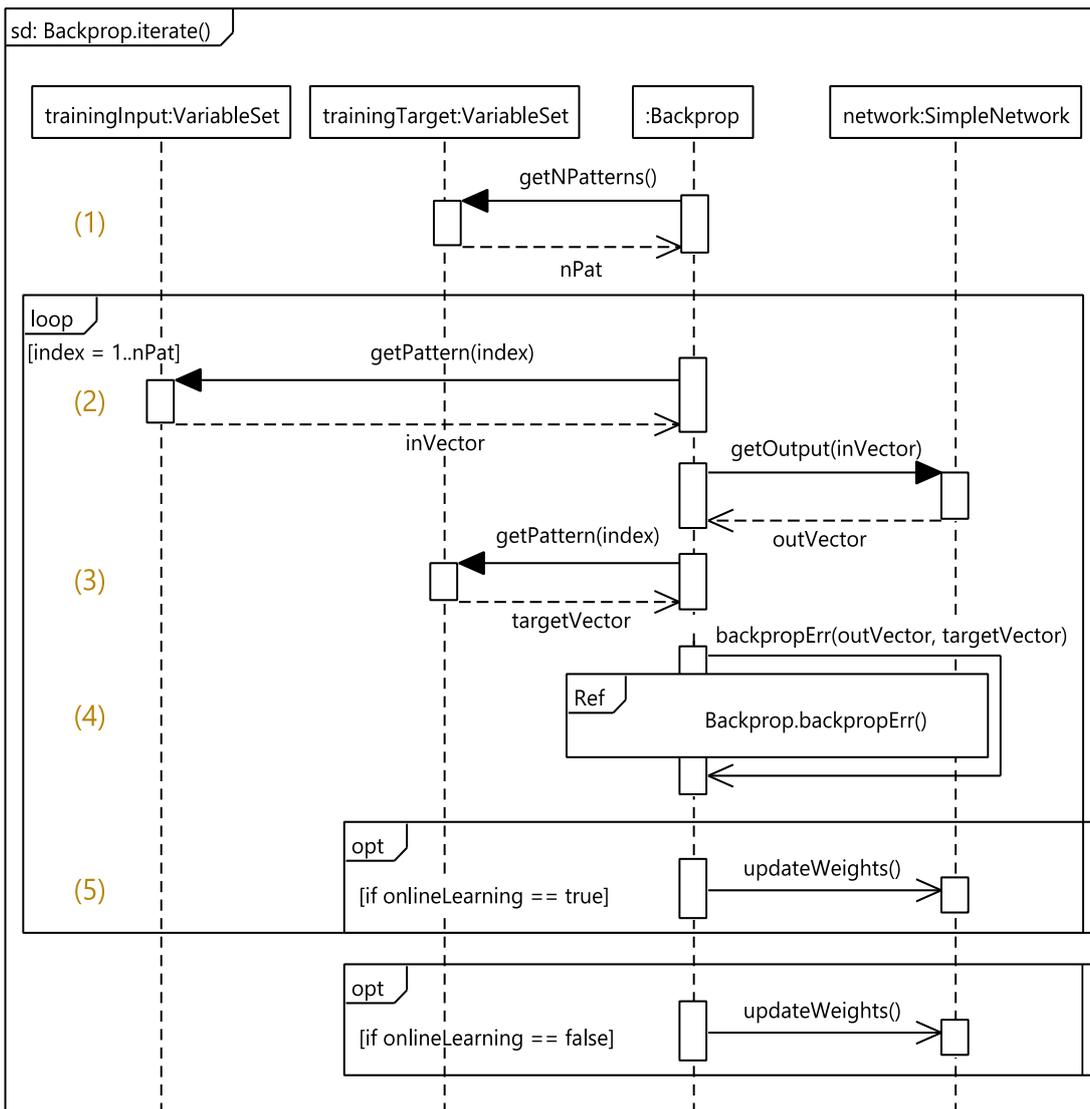


Abb. D-38: Sequenzdiagramm einer Iteration des Backpropagation-Algorithmus

Das Sequenzdiagramm in Abb. D-39 zeigt die in den Abschnitten C.I.3.c bis C.I.3.f (S. 96 ff.) formal hergeleitete Fehlerrückpropagation als Interaktion zwischen Trainingsalgorithmus, Neuronen und Synapsen:

- (1) Das Backprop-Objekt berechnet die Fehlersignale der Neuronen in der Ausgabeschicht entsprechend der partiellen Ableitung (C.23) einer generalisierten Fehlerfunktion (C.22).¹⁾

¹⁾ Siehe S. 106 ff.; siehe auch (C.15), S. 97 ff.

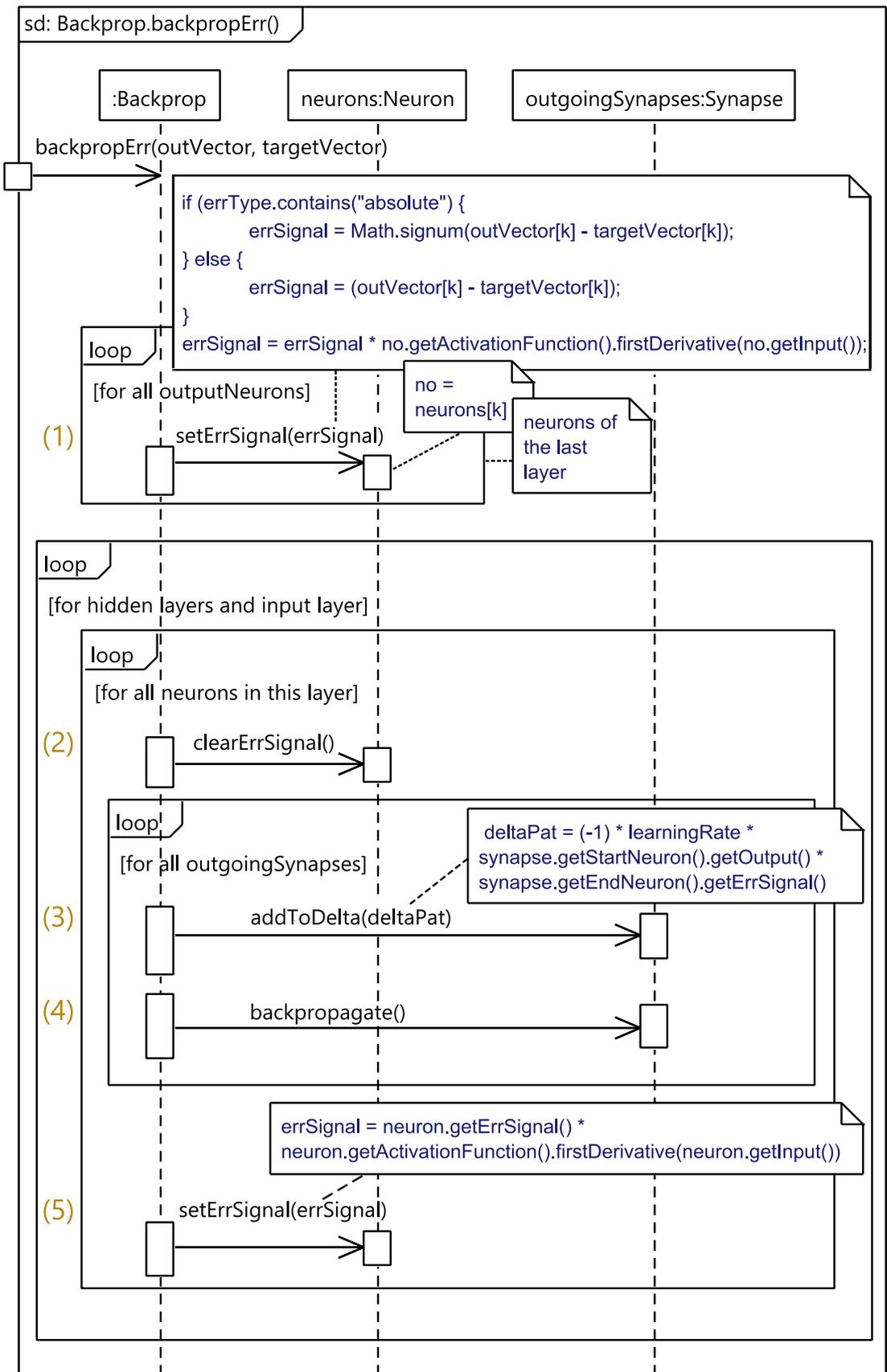


Abb. D-39: Sequenzdiagramm der Fehlerrückpropagation im künstlichen neuronalen Netz

- (2) Die Fehlersignale der Neuronen in den übrigen Schichten werden initialisiert.
- (3) Für alle ausgehenden Synapsen wird der Fehlergradient nach Formel (C.16) bzw. (C.18) berechnet, mit einer konstanten Lernrate multipliziert und zur Gewichtsänderung Δw hinzuaddiert (siehe (C.20)). Die Gewichtsänderungen Δw addiert sich über alle Trainingspunkte auf (siehe (C.19)), wenn nicht nach jedem Trainingspunkt die Operation `updateWeight()` aufgerufen wird (siehe Abb. D-38). Die Operation `updateWeight()` (siehe auch Abb. D-31, S. 239) ändert das Gewicht einer Synapse zu $\text{weight} = \text{weight} + \Delta w$ (siehe (C.9), S. 94) und setzt Δw zurück auf Null. An dieser Stelle wird auch der Impulsterm (momentum) berücksichtigt (in den Abbildungen nicht dargestellt; siehe (C.24), S. 110).
- (4) Die Synapsen geben das gewichtete Fehlersignal ihres Endneurons nach (C.17) an das Startneuron weiter. Die Operation `backpropagate()` enthält nur eine Zeile:

```
startNeuron.errSignal = startNeuron.errSignal + weight * endNeuron.errSignal.
```
- (5) Das `Backprop`-Objekt berechnet die Fehlersignale der Neuronen in der aktuellen Schicht nach (C.17).

3. *Resilient propagation* als Unterklasse der *backpropagation*

Der Resilient-Propagation-Algorithmus ist als Unterklasse des Backpropagation-Algorithmus implementiert. Dadurch erbt er die Operation `backpropErr()` und kann sie weitgehend unverändert zur Berechnung des Fehlergradienten verwenden. Einzig die Berechnung der Gewichts Anpassung (Schritt (3) in Abb. D-39) ersetzt das `Resprop`-Objekt durch (C.25) (S. 111).

Dafür muss die Klasse Synapse um zwei Attribute erweitert werden, die Gradient und Gewichtsänderung der vorhergehenden Iteration speichern.

Die Klasse Resprop ignoriert die Einstellung `onlineLearning`, da der Resilient-Propagation-Algorithmus nur als *batch learning* zuverlässig funktioniert.¹⁾

4. Implementierung der *extreme learning machine*

Die *extreme learning machine* beinhaltet keine Fehlerrückpropagation und unterscheidet sich dadurch wesentlich von den anderen Trainingsalgorithmen. Bei der Implementierung nach Algorithmus C-2 (S. 122) wird zur Berechnung der Ausgabematrix der verdeckten Schicht O_H jede Beobachtung des Eingavektors wie in Abb. D-34 vorwärtspropagiert. Für die Singulärwertzerlegung stehen zahlreiche Methodensammlungen zur Verfügung. Cenobi verwendet die freie Methodenbibliothek Commons-Math 3.3 von Apache.

Da die *extreme learning machine* die optimalen Gewichte zwischen verdeckter Schicht und Ausgabeschicht in einem Schritt berechnet, haben die Abbruchbedingungen sowie die `onlineLearning`- und `earlyStopping`-Attribute der `TrainingSettings` für sie keine Bedeutung. Zudem kann eine *extreme learning machine* nur *mutlilayer perceptrons* mit einer einzigen verdeckten Schicht trainieren.²⁾

¹⁾ Vgl. Riedmiller/Braun [RPROP] 587; Kruse u. a. [Intelligence] 72.

²⁾ Siehe Abschnitt C.I.3.i, S. 118 ff.

X. Bewertung der Modellgüte

1. Fehlermaße zur Beurteilung der Prognosegenauigkeit

Die Güte eines entscheidungsorientierten Schätzmodells bemisst sich vor allem an seiner Prognosegenauigkeit.¹⁾ In der wissenschaftlichen Literatur werden die Vor- und Nachteile verschiedener Prognosegenauigkeitsmaße kontrovers diskutiert. Man ist sich jedoch weitgehend einig, dass der normierte mittlere absolute Fehler

$$\text{NMAE} = \frac{1}{N} \sum_{n=1}^N \left| \frac{\hat{y}^{(n)} - y^{(n)}}{y^{\max} - y^{\min}} \right| \quad (\text{D.12})$$

und der symmetrische mittlere absolute prozentuale Fehler

$$\text{MAPE}_S = \frac{1}{N} \sum_{n=1}^N \left| \frac{\hat{y}^{(n)} - y^{(n)}}{1/2(\hat{y}^{(n)} + y^{(n)})} \right| \quad (\text{D.13})$$

eine gute Einschätzung der Prognosegenauigkeit erlauben. Gegenüber dem einfachen $\text{MAPE} = \frac{1}{N} \sum_{n=1}^N |(\hat{y}^{(n)} - y^{(n)})/y^{(n)}|$ hat der MAPE_S die Vorteile, dass er auch für $y^{(n)} = 0$ definiert ist und positive wie negative Abweichungen gleich gewichtet. Quadratische Fehlermaße sowie das Bestimmtheitsmaß R^2 gelten hingegen als ungeeignet, weil sie größere Fehler überproportional gewichten und dadurch zu sensibel auf einzelne große Fehler – beispielsweise durch Ausreißer – reagieren.²⁾

¹⁾ Vgl. auch im Weiteren Barrot [Prognosegütemaße]; Küsters [Evaluation]; Crone [Neuronale Netze] 128 ff., 356 ff.; jeweils mit weiteren Nachweisen.

²⁾ Vgl. Barrot [Prognosegütemaße] 557; Küsters [Evaluation] 433; Crone [Neuronale Netze] 142; jeweils mit weiteren Nachweisen.

2. Empirische Prognosegenauigkeit der implementierten Trainingsalgorithmen

Die Prognosegenauigkeit der Trainingsalgorithmen wurde an einer Auswahl der Variablen aus dem Beispieldatensatz getestet (siehe Anhang A3). Das Netz wurde auf die ersten 57 Beobachtungen trainiert und sollte die verbleibenden 6 Beobachtungen prognostizieren.

Als Zielfunktion der Trainingsalgorithmen wurde der mittlere absolute Fehler verwendet. Der Backpropagation-Algorithmus wurde im Online-Verfahren mit einer Lernrate $\eta = 0,1$ ausgeführt.¹⁾ Der Resilient-Propagation-Algorithmus wurde im Batch-Verfahren mit einem Initialdelta $\Delta w_{ij}^{(0)} = 0,1$ und den Lernraten $\eta^+ = 1,1$ und $\eta^+ = 0,6$ ausgeführt.²⁾ Bei der *extreme learning machine* müssen keine Parameter gesetzt werden. Alle Verfahren wurden mit zufällig aus der Standardnormalverteilung gezogenen Gewichten initialisiert. Für *backpropagation* und *resilient propagation* galten die in Abb. D–37 (S. 247) gezeigten Abbruchbedingungen. Die *extreme learning machine* ignoriert diese Einstellungen, da sie grundsätzlich nur eine Iteration benötigt.

Die Algorithmen wurden zehnmal neu initialisiert und ausgewertet. Abb. D–40 zeigt die daraus berechneten mittleren Prognosefehler nach (D.12) und (D.13) für die 57 Beobachtungen der Trainingsmenge und die 6 Beobachtungen der Testmenge in Abhängigkeit von der Anzahl der Neuronen in der verdeckten Schicht.

¹⁾ Vgl. die Empfehlungen von Rumelhart/Hinton/Williams [Back-propagating Errors]; Rumelhart/Hinton/Williams [Error Propagation] 330.

²⁾ Vgl. die Empfehlungen von Riedmiller/Braun [RPROP] 588; Kruse u. a. [Intelligence] 72.

Anzahl verdeckter Neuronen		4	6	8	10	12	14	16
NMAE Training	<i>backpropagation</i>	0,079	0,063	0,061	0,046	0,050	0,052	0,035
	<i>resilient propagation</i>	0,057	0,053	0,048	0,043	0,039	0,037	0,035
	<i>extreme learning m.</i>	0,136	0,098	0,084	0,068	0,071	0,064	0,061
MAPE _S Training	<i>backpropagation</i>	0,114	0,094	0,090	0,071	0,074	0,078	0,053
	<i>resilient propagation</i>	0,086	0,080	0,072	0,065	0,059	0,057	0,054
	<i>extreme learning m.</i>	0,215	0,146	0,125	0,101	0,104	0,095	0,090
NMAE Test	<i>backpropagation</i>	0,168	0,137	0,171	0,162	0,192	0,298	0,192
	<i>resilient propagation</i>	0,138	0,136	0,142	0,140	0,149	0,156	0,159
	<i>extreme learning m.</i>	0,172	0,155	0,139	0,144	0,149	0,141	0,141
MAPE _S Test	<i>backpropagation</i>	0,209	0,177	0,214	0,216	0,252	0,397	0,254
	<i>resilient propagation</i>	0,182	0,181	0,188	0,187	0,197	0,208	0,209
	<i>extreme learning m.</i>	0,227	0,210	0,185	0,192	0,200	0,184	0,188

- ⊠ *backpropagation*
- △ *resilient propagation*
- × *extreme learning machine*

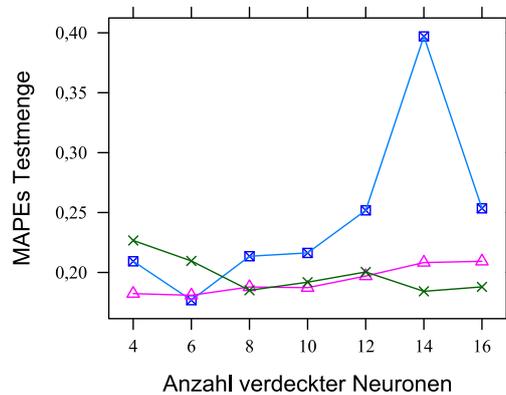
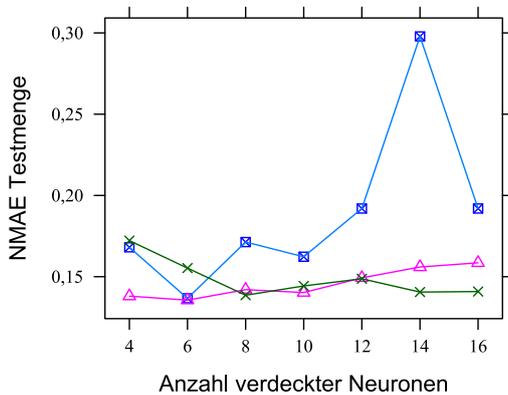
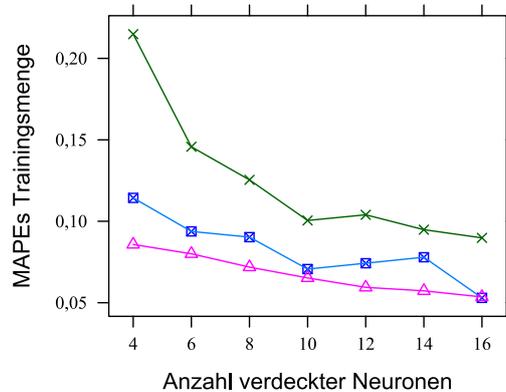
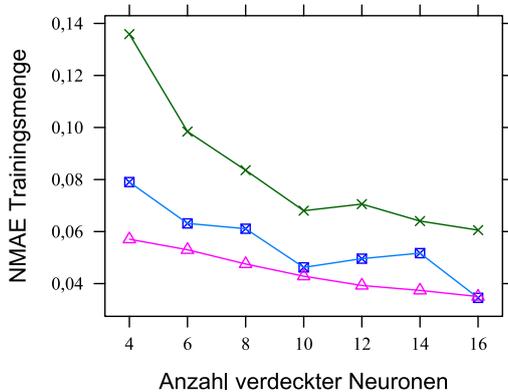


Abb. D-40: Prognosefehler der implementierten Trainingsalgorithmen

Es ist zu erkennen, dass sich die Trainingsalgorithmen mit steigender Neuronenzahl immer besser an die Trainingspunkte anpassen. Die Out-of-Sample-Fehler für die sechs unbekanntenen Testpunkte sinken zunächst ebenfalls, steigen ab einer gewissen Anzahl verdeckter Neuronen aber wieder an. Backpropagation- und Resilient-Propagation-Algorithmus erreichen ihre beste Generalisierungsleistung bei 6 verdeckten Neuronen, die *extreme learning machine* je nach Fehlermaß bei 8 oder 14. Danach verringert die Überanpassung an die Trainingsdaten die Generalisierungsfähigkeit.

Dass die optimale Neuronenzahl bei der *extreme learning machine* größer ist, liegt daran, dass bei diesem Verfahren nur die Gewichte zwischen verdeckter Schicht und Ausgabeschicht freie Parameter sind. Die davor liegenden Gewichte sind auf einen konstanten, zufällig gewählten Wert festgelegt. Dadurch ergeben sich bei der *extreme learning machine* auch stärkere Schwankungen zwischen den einzelnen Testdurchläufen, je nachdem wie glücklich die vorderen Gewichte gesetzt wurden. Im besten der zehn Durchläufe erreichte die *extreme learning machine* einen $MAPE_S$ über die Testmenge in Höhe von 0,166, im schlechtesten nur einen in Höhe von 0,200. Im Durchschnitt waren es die in Abb. D-40 gezeigten 0,184. Damit erreicht die *extreme learning machine* den niedrigsten $MAPE_S$ in einem einzelnen Durchlauf, schneidet im Durchschnitt aber etwas schlechter ab, als die beiden anderen Algorithmen, mit einem minimalen Test- $MAPE_S$ von 0,177 (*backpropagation* mit 6 verdeckten Neuronen) bzw. 0,181 (*resilient propagation* mit 6 verdeckten Neuronen). Dafür ist die *extreme learning machine* tatsächlich „extrem“ schnell. Selbst bei 16 verdeckten Neuronen benötigt sie nur sieben Millisekunden Rechenzeit (siehe Abb. D-41).

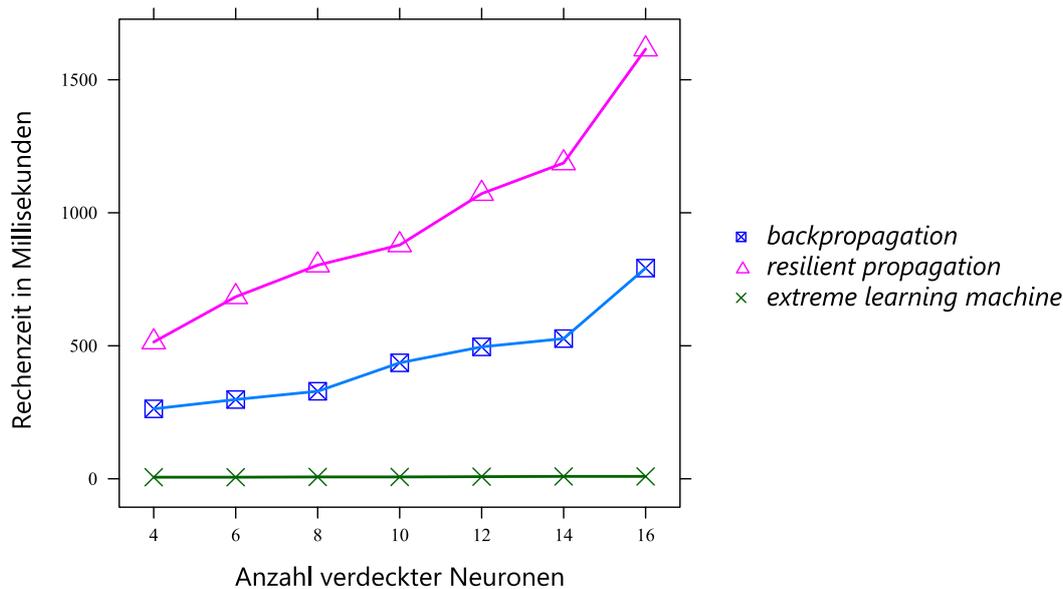


Abb. D–41: Rechendauer der implementierten Trainingsalgorithmen¹⁾

Die Verläufe der Out-of-Sample-Fehler in Abb. D–40 zeigen, dass es eine optimale Anzahl verdeckter Neuronen gibt, bei der die Prognosegenauigkeit für neue Daten, d. h. die Generalisierungsleistung, am höchsten ist, weil das neuronale Netz gerade so viele freie Parameter enthält, dass es sich gut an den Modellzusammenhang, aber nicht an zufälliges Rauschen anpassen kann. Die optimale Größe der verdeckten Schicht hängt von der Komplexität des Modellzusammenhangs und – wie in Abb. D–40 deutlich zu sehen – vom gewählten Trainingsalgorithmus ab. Eine allgemeine Empfehlung zur Anzahl der verdeckten Neuronen kann deshalb nicht ausgesprochen werden. Doch es gibt einige Techniken, die es erlauben, die Suche nach einem möglichst *overfitting*-freien neuronalen Netz zu automatisieren. Sie werden im folgenden Teilkapitel vorgestellt.

¹⁾ Auf einem Intel i5-Prozessor mit 3,4 GHz Rechenleistung.

XI. Techniken zur Vermeidung von *overfitting*

1. Modellselektion durch Kreuzvalidierung

Bei der Modellselektion (*model selection*) geht es im Kern darum, unter vielen möglichen Modellen (z. B. künstlichen neuronalen Netzen mit unterschiedlich vielen verdeckten Neuronen) dasjenige mit der besten Generalisierungsleistung zu finden. Eine naheliegende Lösung besteht darin, ähnlich wie oben in Abschnitt X.2 vorzugehen. Da der Fehler über die Trainingsdaten (C.7) (S. 94) wegen des möglichen *overfittings*¹⁾ nicht aussagekräftig ist, verwendet man einen Teil der verfügbaren Daten nicht zum Training.²⁾ Man trainiert verschiedene Modelle auf die Trainingsdaten, vergleicht die Prognosefehler über die unbekanntes Daten (die Validierungsmenge) und wählt das Modell mit dem besten Ergebnis. Weil in diesem Auswahlprozess eine Überanpassung an die Validierungsmenge auftreten kann – sozusagen ein *overfitting* zweiter Ordnung –, hält man häufig noch eine dritte Datenmenge (die Testmenge) zurück, an der die Generalisierungsleistung des schließlich ausgewählten Modells getestet wird.

In vielen Anwendungsbereichen – insbesondere auch in der Kostenrechnung – steht jedoch insgesamt nur eine sehr begrenzte Datenmenge zur Verfügung. Grundsätzlich ist man bestrebt, den Modellen einen möglichst großen Anteil dieser Daten für das Training zur Verfügung zu stellen. Wenn dadurch aber die Validierungsmenge so klein wird, dass sie nicht

¹⁾ Zur Problematik des *overfittings* siehe ausführlich Abschnitt C.I.2, S. 88 ff.

²⁾ Vgl. auch im Weiteren Bishop [Machine Learning] 32. Die Begriffe *Validierungsmenge* und *Testmenge* werden in der wissenschaftlichen Literatur nicht einheitlich verwendet. Es beginnen sich aber die in Bishops Standardwerk gewählten und hier wiedergegebenen Bezeichnungen durchzusetzen.

mehr repräsentativ für die Grundgesamtheit ist, lässt sie keine zuverlässigen Schlüsse auf die Generalisierungsfähigkeit des Modells zu. Dieser Zielkonflikt zwischen Maximierung der Trainingsmenge und Maximierung der Validierungsmenge kann durch Kreuzvalidierung entschärft werden.

Abb. D-42 verdeutlicht die Grundidee der Kreuzvalidierung. Bei einer S -fachen Kreuzvalidierung wird die Datenmenge in S gleich große Teile geteilt, von denen in jedem Validierungsdurchlauf jeweils $S - 1$ Teile zum Training und ein Teil für die Validierung verwendet werden, bis jeder Teil einmal Validierungsmenge war. Anschließend werden die Validierungsergebnisse aller Durchläufe gemittelt, z. B.

$$\text{NMAE}_{\text{KV}} = \frac{1}{S} \sum_{i=1}^S \text{NMAE}_i. \quad (\text{D.14})$$

Wenn besonders wenige Daten zur Verfügung stehen, kann eine *leave-one-out* Kreuzvalidierung mit $S = N$ durchgeführt werden. Dadurch wird ein Modell in jedem Durchlauf mit der größtmöglichen Trainingsmenge trainiert und trotzdem über den gesamten Datensatz validiert.¹⁾



Abb. D-42: Schema einer vierfachen Kreuzvalidierung

¹⁾ Vgl. Bishop [Machine Learning] 33; Kuhlmann [Kreuzvalidierung].

Die Kreuzvalidierung ist einfacher anzuwenden und weniger voraussetzungsreich¹⁾ als **andere Modellselektionskriterien** wie z. B.

- das Akaike Informationskriterium AIC,²⁾
- das bayessche Informationskriterium BIC,³⁾
- das adjustierte Bestimmtheitsmaß R_{adj}^2 ,⁴⁾
- F-Statistiken,⁵⁾
- Hypothesentests zur Bestimmung der Parametersignifikanz,⁶⁾
- der auf der bayesschen *model evidence* (auch *marginal likelihood*) beruhende Bayes-Faktor.⁷⁾

Die Kreuzvalidierung ist ein universell einsetzbares Verfahren, das ab $S \geq 10$ zuverlässig eine fast unverzerrte Schätzung der wahren Generalisierungsleistung eines Modells erlaubt.⁸⁾ Ein Nachteil der Kreuzvalidierung ist allerdings ihr hoher Rechenaufwand. Bei einer S -fachen Kreuzvalidierung muss jeder Modellkandidat S mal trainiert werden – bei den anderen ge-

¹⁾ Vgl. Anders/Korn [Model Selection] 314; Bishop [Machine Learning] 23.

²⁾ Vgl. grundlegend Akaike [Model Identification]; Burnham/Anderson [Model Selection]; Burnham/Anderson [Multimodel Inference]. Vgl. speziell zur Auswahl eines optimalen neuronalen Netzes Fogel [Neural Network Selection]; Hagiwara/Toda/Usui [Applying AIC]; Murata/Yoshizawa/Amari [Network Information Criterion]; Anders/Korn [Model Selection].

³⁾ Vgl. Schwarz [Dimensions]; Burnham/Anderson [Model Selection]; Burnham/Anderson [Multimodel Inference]; Lv/Liu [Model Selection].

⁴⁾ Vgl. Bendel/Afifi [Stopping Rules] 48; Hedderich/Sachs [Statistik] 708; Backhaus u. a. [Analysemethoden] 76.

⁵⁾ Vgl. Pope/Webster [F-Statistic]; Bendel/Afifi [Stopping Rules] 48; Hedderich/Sachs [Statistik] 708 f.; Backhaus u. a. [Analysemethoden] 76 ff.

⁶⁾ Vgl. Anders/Korn ([Model Selection] 311 ff., mit weiteren Nachweisen) für eine Übersicht zu speziellen Wald- und Lagrange-Multiplikator-Tests für die Gewichte eines neuronalen Netzes.

⁷⁾ Vgl. speziell für künstliche neuronale Netze Bishop [Machine Learning] 161 ff. Vgl. auch Link/Barker [Multimodel Inference].

⁸⁾ Vgl. Anders/Korn [Model Selection] 320 ff.; Kohavi [Cross-Validation]; Liu [Unbiased Estimate]; Edwards/Murray [Model Selection]; Olden/Jackson [Torturing Data] 506 f.; Cawley/Talbot [Cross-Validation] 1467 f.

nannten Selektionskriterien hingegen nur einmal. Demnach bietet es sich an, die Kreuzvalidierung mit schnellen Lernalgorithmen wie z. B. der *extreme learning machine* zu kombinieren.

Ein weiteres Problem, das aber alle Modellselektionskriterien betrifft, tritt auf, wenn mehr als ein Modellmerkmal variiert wird. Soll beispielsweise neben der Anzahl der verdeckten Neuronen auch die Auswahl der Inputvariablen optimiert werden, kann das bereits sehr viele Kombinationsmöglichkeiten ergeben. Wenn dann noch Trainingsparameter, z. B. Lernraten, mitoptimiert werden sollen, ist es kaum noch möglich, alle Modellkandidaten zu testen. Hier bieten Metaheuristiken wie z. B. genetische Algorithmen einen Ausweg.¹⁾ Sie können flexibel eingesetzt werden und ermöglichen eine effiziente Suche nach dem optimalen Modell.

2. *Early stopping* der Parametrisierung

Auch nachdem (z. B. durch Kreuzvalidierung) eine optimale Anzahl verdeckter Neuronen und anderer Modellmerkmale ermittelt wurde, enthält das neuronale Netz in der Regel noch zu viele Freiheitsgrade.²⁾ Um bei gegebener Neuronenzahl *overfitting* zu vermeiden, gibt es grundsätzlich zwei Ansätze:³⁾

(1) Man reduziert die Anzahl der Parameter, d. h. der Synapsen – entweder vor dem Training, z. B. mit den typischen Modellselektionsverfahren (s. o.), oder während des Trainings durch sogenanntes *pruning*⁴⁾.

¹⁾ Siehe Teilkapitel D.XII, S. 267 ff.

²⁾ Vgl. Finnoff/Hergert/Zimmermann [Model Selection] 780.

³⁾ Vgl. Prechelt [Early Stopping] 761, mit zahlreichen weiteren Nachweisen.

⁴⁾ Vgl. Mozer/Smolensky [Skeletonization]; Mozer/Smolensky [Relevance]; Le Cun/Denker/Solla [Brain Damage]; Reed [Pruning].

(2) Man beschränkt die absolute Höhe der Parameterwerte, indem man z. B. hohe Parameterwerte bestraft (*regularization, weight decay*)¹⁾ oder das Training frühzeitig stoppt (*early stopping*)²⁾. Dadurch verhindert man steile Ausschläge der Schätzfunktion nach oben und unten, wie sie z. B. in Abb. C–5 c (S. 90) zu sehen sind.

Das frühzeitige Stoppen (*early stopping*) der Parametrisierung ist einfach anzuwenden und bringt im Allgemeinen gute Generalisierungsergebnisse.³⁾ Deshalb wird es sehr häufig genutzt. Beim *early stopping* wird ähnlich wie bei der Kreuzvalidierung die verfügbare Datenmenge in eine Trainings- und eine Validierungsmenge gespalten. In iterativen Lernverfahren wie etwa den klassischen Gradientenabstiegsverfahren, sinkt der Fehler über die Trainingsmenge im Verlauf der Iterationen immer weiter, während der Fehler über die nicht im Training verwendete Validierungsmenge zunächst ebenfalls sinkt, dann aber wieder steigt, wenn das *overfitting* an die Trainingsdaten einsetzt (siehe Abb. D–43).

In Lehrbüchern wird *early stopping* regelmäßig durch idealisierte Fehlerverläufe wie in Abb. D–43 motiviert. In der Praxis sehen die Fehlerverläufe allerdings eher wie in Abb. D–44 aus. Dort sind die normalisierten absoluten Fehler des Resilient-Propagation-Algorithmus für sechs verdeckte Neuronen aus Abb. D–40 zu sehen.⁴⁾

¹⁾ Die Bezeichnung *regularization* ist allgemeiner; der Ausdruck *weight decay* („Gewichtsverfall“) wird vor allem bei künstlichen neuronalen Netzen verwendet. Vgl. Bishop [Machine Learning] 144 ff., 257 ff.

²⁾ Vgl. Morgan/Bourlard [Generalization] 635.

³⁾ Vgl. z. B. Finnoff/Hergert/Zimmermann [Model Selection] 780.

⁴⁾ NMAE Training aus Abb. D–40 heißt in Abb. D–44 E_{train} und NMAE Test heißt E_{val} . Die Testmenge aus Abschnitt D.X.2 (S. 254 ff.) ist identisch mit der Validierungsmenge in Abb. D–44. Die Trainingsmengen sind ebenfalls identisch.

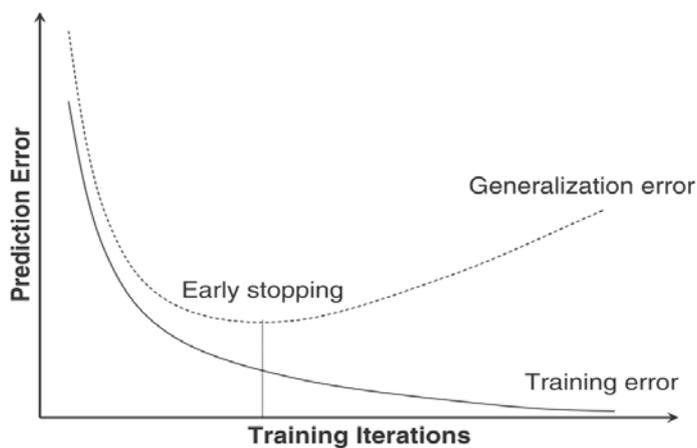


Abb. D-43: Illustration der Early-Stopping-Idee¹⁾

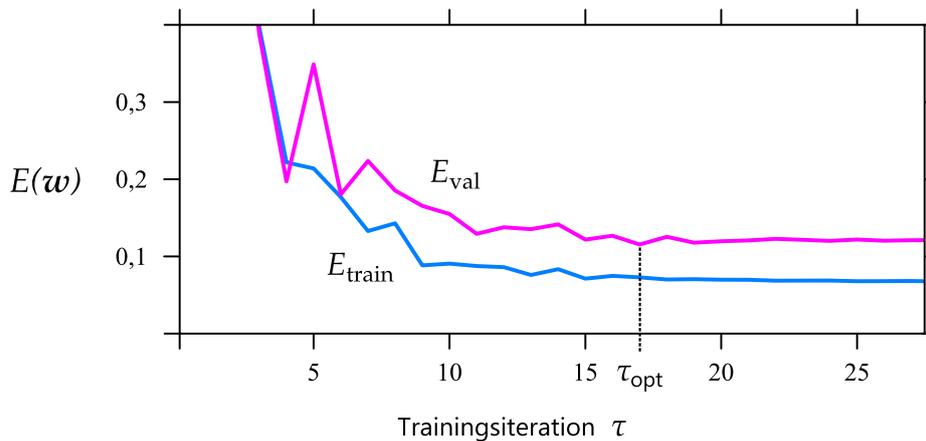


Abb. D-44: Tatsächlicher Verlauf der Trainings- und Validierungsfehler eines Resilient-Propagation-Algorithmus

Der Fehler über die Validierungsmenge E_{val} erreicht sein globales Minimum in der Iteration τ_{opt} . Das Training kann also nach dieser Iteration abgebrochen werden. Das Problem dabei ist, dass der Fehler über die Trainingsmenge E_{train} bei einfachen Gradientenverfahren im Verlauf der Iterationen nicht zwingend monoton fällt, sondern zwischendurch ansteigen kann – z. B. wenn er ein lokales Maximum überwindet (siehe auch Abb. C-

¹⁾ Perlich [Learning Curves] 578.

7 d, S. 101).¹⁾ E_{val} vollzieht diesen Verlauf in der Regel mit und hat demnach mehrere lokale Minima (siehe Abb. D–44). Deshalb sollte man das Training nicht sofort abbrechen, wenn E_{val} steigt, sondern zunächst abwarten, ob E_{val} im weiteren Verlauf noch einmal fällt. Um zu entscheiden, wann man das Training beenden kann, ist es nützlich, sich den Generalisierungsverlust (*generalization loss*)

$$GL^{(\tau)} = 100 \cdot \left(\frac{E_{\text{val}}^{(\tau)}}{E_{\text{val}}^{\text{opt}(\tau)}} - 1 \right) \quad (\text{D.15})$$

mit $E_{\text{val}}^{(\tau)}$ Fehler über die Validierungsmenge, gemessen nach Iteration τ ,
 $E_{\text{val}}^{\text{opt}(\tau)} = \min\{E_{\text{val}}^{(\tau')} \mid \tau' \leq \tau\}$ kleinster Fehler über die Validierungsmenge bis zur aktuellen Iteration τ

einer Iteration τ anzusehen.²⁾ Diese Kennzahl drückt aus, um wieviel Prozent der aktuelle Validierungsfehler $E_{\text{val}}^{(\tau)}$ über dem bislang niedrigsten Fehler $E_{\text{val}}^{\text{opt}(\tau)}$ liegt. Je größer $GL^{(\tau)}$ wird, desto weiter entfernt sich das neuronale Netz von seinem Generalisierungsoptimum. Deshalb ist es sinnvoll, das Training abubrechen, wenn der Generalisierungsverlust einen bestimmten Schwellenwert überschreitet. Empirisch hat sich das Abbruchkriterium $GL^{(\tau)} > 5$ als guter Kompromiss aus Rechendauer und Generalisierungsleistung erwiesen.³⁾ Das Training wird in diesem Fall nach der Iteration τ abgebrochen, in der zum ersten Mal $GL^{(\tau)} > 5$ ist.

Bei einem Fehlerverlauf wie in Abb. D–44 wäre es allerdings fatal, das Training allein nach diesem Kriterium abubrechen. Denn dort sind $GL^{(5)} = 77$,

¹⁾ Nur wenige Lernverfahren – darunter konjugierte Gradientenverfahren – haben eine im Iterationsverlauf monoton fallende Trainingsfehlerfunktion (vgl. Bishop [Machine Learning] 259). Sie sind folglich für das *early stopping* besonders gut geeignet.

²⁾ Vgl. auch im Weiteren Prechelt [Early Stopping] 763.

³⁾ Vgl. Prechelt [Early Stopping] 765 f.

$GL^{(7)} = 24$ und $GL^{(12)} = 7$, aber erst in Iteration $\tau_{opt} = 17$ erreicht der Validierungsfehler sein globales Minimum $E_{val}^{(17)} = 0,1155$. Deshalb führt der in Cenobi implementierte Early-Stopping-Algorithmus immer mindestens 100 Iterationen aus, bevor er das Training abbricht. Insgesamt ergibt sich damit die Abbruchbedingung

wenn $(GL^{(\tau)} > 5) \wedge (\tau \geq 100)$ **dann**
beende Training.

Als Trainingsergebnis wird anschließend nicht der aktuelle Gewichtsvektor w behalten, sondern der Gewichtsvektor, der zum niedrigsten Validierungsfehler $E_{val}^{opt(\tau)}$ geführt hat. Im Early-Stopping-Objekt muss also neben $E_{val}^{opt(\tau)}$ auch der dafür verantwortliche Gewichtsvektor, d. h. das beste trainierte neuronale Netz gespeichert werden (siehe Abb. D-45).

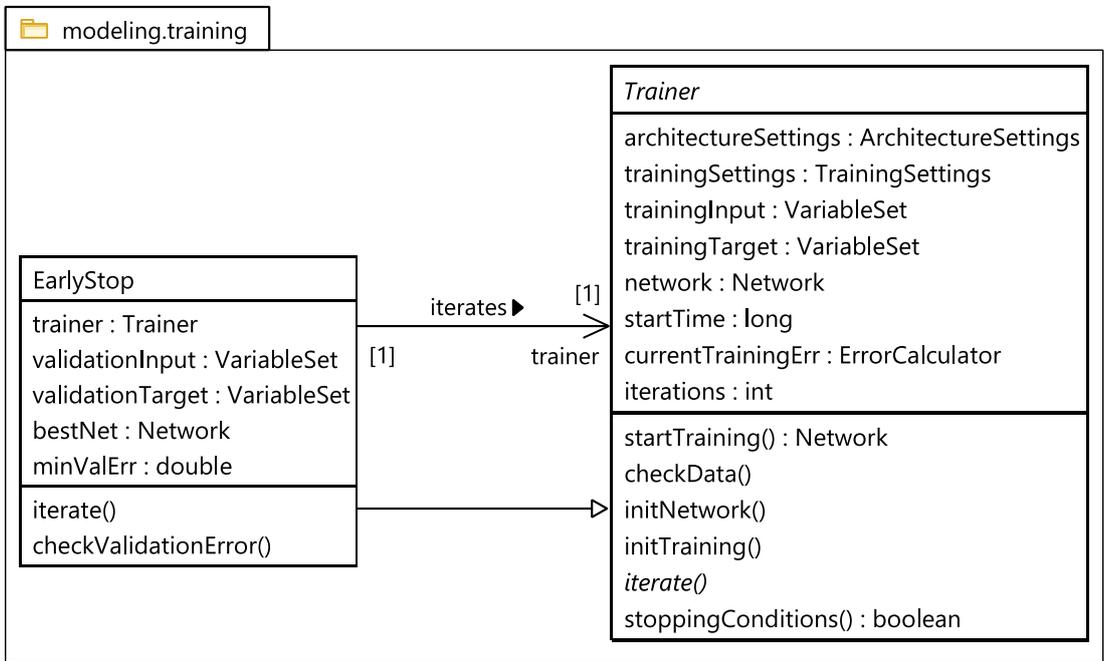


Abb. D-45: Klassendiagramm der EarlyStop-Klasse in Cenobi

In Cenobi ist die Klasse EarlyStop eine konkrete Unterklasse der abstrakten Klasse Trainer und übernimmt damit deren Felder und Operationen. Au-

ßerdem besitzt sie einen Trainer (ein Backprop- oder ein Resprop-Objekt) als Attribut, dessen `iterate()`-Operation sie schrittweise ausführt. Nach jeder Iteration berechnet sie mit der Operation `checkValidationError()` den Generalisierungsverlust (D.15) und beendet ggf. das Training.

Auch beim *early stopping* besteht der Zielkonflikt zwischen einer möglichst großen Trainingsmenge, die man dem Netz zur Verfügung stellen möchte, und einer ausreichend großen, repräsentativen Validierungsmenge (siehe S. 259). Wenn die Validierungsmenge zu klein ist, kann das *early stopping* zu einer Überanpassung an die Validierungsmenge führen. Eine pragmatische Lösung bietet auch hier die Kreuzvalidierung. In Verbindung mit der Early-Stoppping-Methode führt sie dazu, dass man am Ende des Trainingsprozesses nicht ein parametrisiertes Netz, sondern ein Ensemble¹⁾ aus S parametrisierten Netzen erhält (bei S -facher Kreuzvalidierung). Für das erste Netz wird die Teilmenge 1 zur Validierung und die Teilmengen 2 bis S zum Training verwendet, für das zweite Netz ist die Teilmenge 2 die Validierungsmenge usw. Dieses Vorgehen wird als *Rotationstraining* (*rotational training*) bezeichnet.²⁾

Bei der *extreme learning machine* ist kein *early stopping* möglich. Die anderen oben genannten Techniken stehen aber zur Verfügung. Zudem ergibt sich durch die Minimumeigenschaft (C.32) (S. 120) der Moore-Penrose-Inversen bereits ein gewisser Gewichtsverfall (*weight decay*).

¹⁾ Siehe Teilkapitel XIII, S. 284 ff.

²⁾ Vgl. Edwards/Murray [Model Selection] 11.

XII. Modelloptimierung mit Metaheuristiken

1. Möglichkeiten der metaheuristischen Optimierung künstlicher neuronaler Netze

In Abschnitt XI.1 wurde bereits darauf hingewiesen, dass Metaheuristiken die Modellselektion – die im Fall künstlicher neuronaler Netze hauptsächlich in der Suche nach der Netzarchitektur mit der besten Generalisierungsleistung besteht – unterstützen können. Metaheuristiken definieren eine problemunabhängige Folge von Schritten zur näherungsweise Lösung beliebiger Optimierungsprobleme. In den letzten Jahrzehnten wurden zahlreiche Metaheuristiken im Allgemeinen und auch speziell zur Optimierung künstlicher neuronaler Netze entwickelt.¹⁾

Als besonders geeignet zur Optimierung (nicht nur) von künstlichen neuronalen Netzen erweisen sich im empirischen Vergleich regelmäßig evolutionäre Algorithmen,²⁾ von denen vor allem die einfache Unterform der genetischen Algorithmen weit verbreitet ist.³⁾ Evolutionäre Algorithmen werden häufig dazu genutzt, die Netzarchitektur (vor allem die Größe der Eingabeschicht und der verdeckten Schicht), die Parameter des Lernverfahrens (z. B. Lernrate und Impulsterm des Backpropagation-Algorithmus) und die

¹⁾ Für einen ersten Überblick zu verschiedenen Metaheuristiken für neuronale Netze vgl. Alba/Martí [Metaheuristic Procedures].

²⁾ Vgl. Sexton/Dorsey/Johnson [Genetic Algorithm] 599; Madera/Dorransoro [Estimation of Distribution] 104 f.; Alba/Chicano [Genetic Algorithms] 123 ff.; Soares/Antunes/Araújo [Comparison].

³⁾ Genetische Algorithmen wurden zuerst von Holland [Adaptation] systematisch beschrieben; für einen aktuellen Überblick vgl. Whitley/Sutton [Genetic Algorithms]. Zu evolutionären Algorithmen allgemein vgl. Kruse u. a. [Intelligence] 155 ff.; De Jong [Evolutionary Algorithms].

Initialwerte der Gewichte zu optimieren;¹⁾ seltener wird die gesamte Parametrisierung (das Netztraining) evolutionären Strategien überlassen. Werden in einem Algorithmus sowohl Netzarchitektur als auch Lernparameter und (Initial-)Gewichte optimiert, ergibt sich eine verschachtelte Struktur (siehe Abb. D-46).

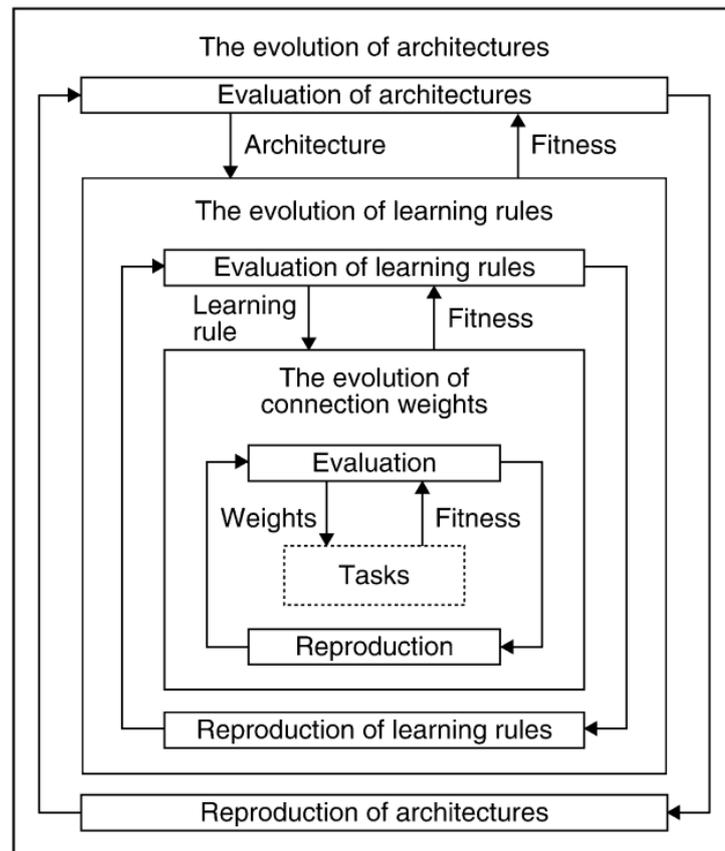


Abb. D-46: Struktur der evolutionären Optimierung von Netzarchitektur, Lernverfahren und Gewichten²⁾

Neben der metaheuristischen Optimierung der Netzarchitektur, die im folgenden Abschnitt 2 beschrieben wird, sind Metaheuristiken vor allem sinnvoll, um systematisch Startlösungen, d. h. Initialgewichte, für ein lokales

¹⁾ Vgl. ausführlich Schaffer/Whitley/Eshelman [Combinations]; Yao [Evolving]; Kovacs [Genetics-Based Machine Learning] 954 ff.; Matias u. a. [Optimized Extreme Learning Machine] 428 f.; jeweils mit zahlreichen weiteren Nachweisen.

²⁾ Yao [Evolving] 1439, Wiederabdruck in Kovacs [Genetics-Based Machine Learning] 958.

Lernverfahren (z. B. einen Backpropagation-Algorithmus) zu erzeugen. Dabei sorgt die Metaheuristik für eine planmäßige, globale Erkundung des Lösungsraums, während das lokale Suchverfahren von dort aus schnell ein (lokales) Optimum findet.¹⁾

Besonders nützlich ist die metaheuristische Optimierung der Initialgewichte bei der *extreme learning machine*, da ihre Schätzgüte stark von den konstanten Gewichten der ersten Synapsenschicht abhängt.²⁾ Dementsprechend wurden bereits zahlreiche metaheuristische Algorithmen speziell für *extreme learning machines* vorgeschlagen.³⁾

2. Implementierung eines genetischen Algorithmus zur Optimierung der Netzarchitektur

a) Ziele der Optimierung

Da die Anzahl der verdeckten Neuronen einen großen Einfluss auf die Generalisierungsleistung des künstlichen neuronalen Netzes hat,⁴⁾ soll bei der Schätzung betrieblicher Kostenfunktionen mit Cenobi die Entscheidung über die Anzahl der verdeckten Neuronen nicht dem Nutzer überlassen bleiben.⁵⁾ Zudem sollen Inputvariable, die keinen oder nur einen sehr geringen Einfluss auf die Outputvariablen haben, automatisch entfernt wer-

¹⁾ Siehe Abschnitt C.I.3.h, S. 116 ff. Vgl. z. B. Moscato [Memetic Algorithms]; Krasnogor [Memetic Algorithms]; Neri/Cotta [Memetic Algorithms]; Kovacs [Genetics-Based Machine Learning] 945; Chen/Tseng/Wu [Evolutionary Training]; Wang/Zeng/Chen [Differential Evolution].

²⁾ Siehe Abschnitt C.I.3.i, S. 118 ff.

³⁾ Vgl. z. B. Fu u. a. [Residence Error]; Matias u. a. [Optimized Extreme Learning Machine]; Xue u. a. [Genetic Ensemble].

⁴⁾ Siehe Abb. D-40, S. 255.

⁵⁾ Der erfahrene Nutzer darf natürlich die Anzahl der verdeckten Neuronen in einem Einstellungsmenü festlegen.

den. Dadurch wird einerseits eine Überanpassung an eigentlich irrelevante Inputvariable verhindert; andererseits wird damit die Lösung des ersten kostenrechnerischen Anwendungsfalls aus Abb. D-5 (S. 181), die Identifikation relevanter Einflussgrößen, vorbereitet.¹⁾ Standardmäßig optimiert Cenobi die Auswahl der Inputvariablen und die Anzahl der verdeckten Neuronen mit einem genetischen Algorithmus und erzeugt automatisch die optimale Architektur.²⁾

Auf eine metaheuristische Optimierung der Lernverfahren wird verzichtet, weil mit *resilient propagation* und *extreme learning machine* zwei Verfahren zur Verfügung stehen, bei denen die Wahl der Lernparameter kaum einen Einfluss auf den Lernerfolg hat³⁾ bzw. keine Parameter gewählt werden müssen⁴⁾.

Eine metaheuristische Optimierung der Initialgewichte findet ebenfalls nicht statt, da bei der gleichzeitigen Optimierung von Gewichten und Architektur so viele Kombinationsmöglichkeiten entstehen, dass die Suche auch mit einem genetischen Algorithmus länger dauert, als es im betrieblichen Alltag zumutbar wäre. Dementsprechend wird häufig das Problem der Architekturoptimierung vom Problem der Gewichtsoptimierung ge-

¹⁾ Siehe Abschnitt E.I.1, S. 300 ff.

²⁾ Ähnliche Ansätze verfolgen u. a. Guo/Uhrig [Select Inputs]; Derigs/Schirp [Genetische Modellierung]; Yao [Evolving] 1437; Donate u. a. [Evolving]. Andere Heuristiken zur Architekturoptimierung beschreiben z. B. Kourentzes/Crone [Automatic Modelling]; Bueno-Crespo/García-Laencina/Sancho-Gómez [Architecture Design].

³⁾ Siehe Abschnitt C.I.3.g.β, S. 111 ff.

⁴⁾ Siehe Abschnitt C.I.3.i, S. 118 ff.

trennt.¹⁾ Ein bekanntes Verfahren, das hingegen beide Probleme simultan löst, ist der EPNet-Algorithmus von Yao und Liu.²⁾

b) Grundlagen evolutionärer Algorithmen

Evolutionäre Algorithmen sind eine Klasse von Metaheuristiken, die in Anlehnung an Darwins Evolutionstheorie eine Menge von Lösungen durch *Selektion*, *Vererbung* und *Mutation* zu verbessern suchen.³⁾ Dabei ist jede Lösung ein *Individuum*, dessen Eigenschaften in kodierter Form durch *Gene* beschrieben werden, die auf einem *Chromosom* angeordnet sind. Eine *Population* ist eine Menge von Individuen. Die Population zu einem bestimmten Zeitpunkt (bzw. Iterationsschritt) wird als *Generation* bezeichnet. Die *fittesten* Individuen einer Generation haben eine höhere Wahrscheinlichkeit, sich fortzupflanzen. Bei der Fortpflanzung entstehen durch die *Kreuzung* von mindestens zwei Chromosomen die Individuen der nachfolgenden Generation. Zufällige *Mutationen* einzelner Gene diversifizieren den Genpool einer Generation. Genetische Algorithmen sind eine bestimmte Form evolutionärer Algorithmen, die sich vor allem dadurch auszeichnet, dass ausschließlich binärkodierte Chromosomen – sogenannte *Bitstrings* – verwendet werden.⁴⁾

¹⁾ Vgl. z. B. aktuell Chen/Tseng/Wu [Evolutionary Training] 351; Fu u. a. [Residence Error] 80.

²⁾ Vgl. Yao/Liu [Evolutionary System] 695. Siehe auch Abb. D-46.

³⁾ Vgl. auch im Weiteren Kruse u. a. [Intelligence] 158 ff.

⁴⁾ Vgl. Kruse u. a. [Intelligence] 194; De Jong [Evolutionary Algorithms] 627; grundlegend Holland [Adaptation].

c) Kodierung der Netzarchitektur

Für den genetischen Algorithmus zur Optimierung der Netzarchitektur in Cenobi wird ein Individuum (eine Architektur) durch ein Chromosom dargestellt, das für jede Inputvariable ein Bit und die Anzahl der verdeckten Neuronen als natürliche Zahl in Gray-Kode enthält (siehe Abb. D-47). Das Chromosom ist der *Genotyp* eines Individuums und die dekodierte Architektur sein *Phänotyp*.

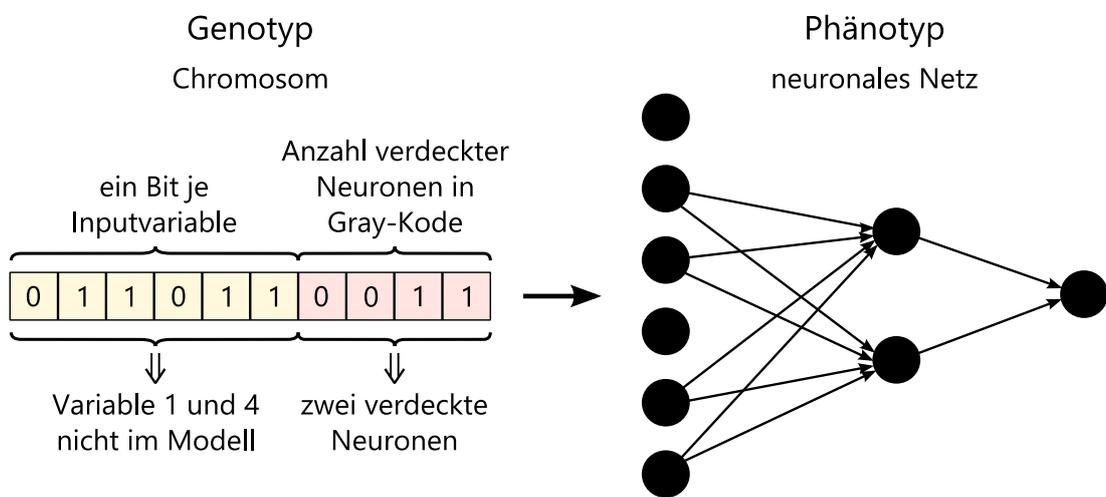


Abb. D-47: Binärcodierung der Netzarchitektur auf einem Chromosom

Im dekodierten neuronalen Netz erhalten alle Inputvariablen ein Neuron in der Eingabeschicht. Wenn das entsprechende Gen auf 0 steht, haben sie allerdings keine ausgehenden Synapsen und damit keinen Einfluss auf die Schätzung. Ein anderes Objekt, das das neuronale Netz aufruft, kann aber mit ihm interagieren, als ob es alle Inputvariablen verarbeitet. Das erleichtert die gleichzeitige Handhabung unterschiedlicher Modelle – insbesondere in einem Ensemble.

In Gray-Kode unterscheiden sich benachbarte Zahlen nur in einem Bit (siehe Abb. D–48).¹⁾ Dadurch erzeugen ähnliche Genotypen, d. h. Chromosomen, die sich nur in wenigen Bits unterscheiden, ähnliche Phänotypen, also in diesem Fall ähnliche Architekturen, mit ähnlicher Fitness. Nur so kann der genetische Algorithmus einen strukturellen Zusammenhang zwischen genetischer Repräsentation und Fitness entdecken und versuchen zu optimieren. Eine Gray-Kodierung enthält weniger Sprünge – sogenannte *Hamming-Klippen* – als eine Kodierung in Dualzahlen. Man vergleiche beispielsweise die Kodierungen der natürlichen Zahlen 7 und 8 in Abb. D–48. Dadurch erzeugt die Gray-Kodierung weniger kodierungsbedingte lokale Optima („Löcher“) im Lösungsraum, die die Suche erschweren.

natürliche Zahl	0	1	2	3	4	5	6	7	8
Gray-Kode	0000	0001	0011	0010	0110	0111	0101	0100	1100
Dualzahl	0000	0001	0010	0011	0100	0101	0110	0111	1000

Abb. D–48: Gray-Kode für natürliche Zahlen

Der Gray-Kode für eine natürliche Zahl z lässt sich in zwei Programmzeilen erzeugen:²⁾

```
int grayInt = z ^ ( z >>> 1 );
String grayCode = Integer.toBinaryString(grayInt);
```

Dabei verschiebt die Binärshift-Operation $z \ggg 1$ die Dualzahldarstellung von z um eine Stelle nach rechts und fügt am linken Ende eine 0 an. Das binäre Exklusiv-Oder \wedge vergleicht paarweise jedes Bit in den Dualzahldar-

¹⁾ Vgl. auch im Weiteren Kruse u. a. [Intelligence] 169 ff.; Whitley/Sutton [Genetic Algorithms] 647, 663, mit weiteren Nachweisen.

²⁾ Vgl. Whitley/Sutton [Genetic Algorithms] 646.

stellungen von z und $(z \ggg 1)$. Das folgende Beispiel verdeutlicht die Operationen an der natürlichen Zahl $z = 4$:

```
int z = 4;
Integer.toBinaryString(z);           // 100
Integer.toBinaryString(z >>> 1);    // 010
Integer.toBinaryString(z ^ ( z >>> 1 )); // 110
```

Um natürliche Zahlen im Intervall $[a; b]$ in Gray-Kode darzustellen, benötigt man $\lceil \log_2(b - a) \rceil$ Bits.¹⁾ Reelle Zahlen lassen sich nach einer Transformation in den Bereich der natürlichen Zahlen ebenso kodieren.

Um den Suchraum des genetischen Algorithmus einzuschränken, kann der Cenobi-Nutzer eine minimale und eine maximale Anzahl verdeckter Neuronen definieren. Abb. D-49 zeigt die Standardeinstellung in Cenobi, mit minimal einem und maximal fünfzehn Neuronen. Dieses Intervall ist für die Schätzung typischer Kostenfunktionen ausreichend (erfahrungsgemäß werden zwischen 2 und 10 Neuronen benötigt) und muss i. A. nicht geändert werden. Zugelassen sind aber alle natürlichen Zahlen im Intervall $[1; N - 1]$. N ist hier die Anzahl der Trainingspunkte.

Abb. D-49: Einstellungsmenü der Netzarchitektur

¹⁾ Vgl. auch im Weiteren Kruse u. a. [Intelligence] 170 f.

Da bei einer *extreme learning machine* mit N verdeckten Neuronen die Matrix O_H (C.27) (S. 119) quadratisch und damit invertierbar ist, kann sie sich perfekt an alle N Trainingspunkte anpassen.¹⁾ Diese Approximation ist mit Sicherheit nicht generalisierbar. Deshalb soll die Anzahl der verdeckten Neuronen kleiner als die Anzahl der Trainingspunkte sein. Andere Lernverfahren benötigen noch deutlich weniger verdeckte Neuronen als die *extreme learning machine*,²⁾ so dass ein Maximum von $N - 1$ verdeckten Neuronen für alle Lernverfahren und beliebige Problemstellungen ausreicht. Falls der Nutzer im Einstellungsmenü ein höheres Maximum definiert hat, wird es automatisch auf $N - 1$ reduziert, sobald feststeht, wie viele Trainingspunkte zur Verfügung stehen.

d) Ablauf des genetischen Algorithmus

Algorithmus D-3 zeigt den Ablauf des genetischen Algorithmus zur Optimierung der Netzarchitektur in Pseudocode. In den Zeilen 1 bis 7 werden verschiedene Parameter übergeben, die der Nutzer in einem Einstellungsmenü auf der graphischen Benutzeroberfläche ändern kann (siehe Abb. D-50). In Zeile 9 wird eine Anfangspopulation der Größe $|POP^{(\tau)}| = P$ mit zufällig gewählten Binärstrings erzeugt.

Während der Evolutionsphase wird zunächst jedes Individuum durch Kreuzvalidierung bewertet (Zeile 11 bis 15).³⁾ Dafür wird jedes Chromosom dekodiert und das entsprechende neuronale Netz S mal mit verschiedenen Datenmengen trainiert und validiert. Da der Fitnesswert F umso höher sein soll, je besser die Lösung ist, definiert Zeile 14 die Fitness eines Individuums als Kehrwert seines Kreuzvalidierungsfehlers.

¹⁾ Vgl. Huang/Zhu/Siew [Extreme Learning Machine] 493.

²⁾ Siehe Abb. D-40, S. 255.

³⁾ Siehe Abschnitt XI.1, S. 258 ff.

Algorithmus Genetische Optimierung der Netzarchitektur

Übergabe der Benutzereinstellungen

- 1: $P \leftarrow$ Populationsgröße
 - 2: $S \leftarrow$ Anzahl der Kreuzvalidierungsdurchläufe
 - 3: $\epsilon \leftarrow$ Elitismusrate
 - 4: $T \leftarrow$ Anzahl der Fitnesssturnierteilnehmer
 - 5: $p_\chi \leftarrow$ Kreuzungswahrscheinlichkeit
 - 6: $K \leftarrow$ Anzahl der Schnittpunkte bei der Chromosomenkreuzung
 - 7: $p_\mu \leftarrow$ Mutationswahrscheinlichkeit
-

Initialisierung

- 8: $\tau \leftarrow 0$
 - 9: $POP^{(\tau)} \leftarrow$ Anfangspopulation mit P zufällig erzeugten Individuen
-

Evolution

- 10: **wiederhole**
- 11: **für** alle Individuen in $POP^{(\tau)}$ **führe aus**
- 12: Dekodiere das Chromosom und erzeuge seinen Phänotyp.
 Trainiere das neuronale Netz (ohne *early stopping*) und berechne
- 13: den Validierungsfehler, z. B. $NMAE_{KV}$ (D.14), durch S -fache
 Kreuzvalidierung.
- 14: Berechne die individuelle Fitness $F = \frac{1}{NMAE_{KV}}$.
- 15: **ende für**
- 16: $ELITE \leftarrow$ selektiere die $\epsilon \cdot P$ fittesten Individuen aus $POP^{(\tau)}$
- 17: $POP^{(\tau+1)} \leftarrow ELITE$
- 18: **wiederhole**
- 19: $AUSWAHL \leftarrow$ zwei Individuen, die jeweils ein Fitnesssturnier
 mit T zufällig ausgewählten Teilnehmern gewonnen haben
- 20: **wenn** neue Zufallszahl $\in [0; 1] < p_\chi$ **dann**
- 21: $AUSWAHL \leftarrow$ K -Punkt-Kreuzung der beiden Individuen
- 22: **ende wenn**
- 23: **wenn** neue Zufallszahl $\in [0; 1] < p_\mu$ **dann**
- 24: $AUSWAHL \leftarrow$ negiere bei jedem Individuum ein zufällig
 ausgewähltes Gen

- 25: **ende wenn**
- 26: $POP^{(\tau+1)} \leftarrow POP^{(\tau+1)} \cup AUSWAHL$
- 27: **bis** $|POP^{(\tau+1)}| = P$
- 28: $\tau \leftarrow \tau + 1$
- 29: **bis** Abbruchkriterium erfüllt
- 30: **Rückgabe** Liste mit Phänotypen der letzten Generation $POP^{(\tau)}$
-

Algorithmus D-3: Genetische Optimierung der Netzarchitektur

Heuristik Genetic

Fitnessberechnung

Anzahl Kreuzvalidierungen

Art des Validierungsfehlers MeanAbsolute

Evolution

Populationsgröße

Elitismusrate

Anz. Fitnesssturnierteilnehmer

Kreuzungswahrscheinlichkeit

Anz. Schnittpunkte bei Kreuzung

Mutationswahrscheinlichkeit

Abbruchkriterien

Kreuzvalidierungsfehler kleiner als

maximale Dauer in Sekunden

max. aufeinanderfolgende Iterationen ohne Verbesserung des Trainingsfehlers

Speichern Abbrechen Standardeinstellungen ?

Abb. D-50: Einstellungsmenü des genetischen Algorithmus

Nachdem die Fitness aller Individuen der aktuellen Generation $POP^{(\tau)}$ bekannt ist, wird ein Anteil $\epsilon \in [0; 1[$ der besten Individuen unverändert, d. h. ohne Kreuzung oder Mutation in die folgende Generation $POP^{(\tau+1)}$ übernommen (Zeile 16 und 17). Insgesamt gelangen auf diese Weise $\epsilon \cdot P$ Individuen in die nachfolgende Population. Dieser sogenannte *Elitismus* stellt sicher, dass die Fitness des besten Individuums von einer Generation zur nächsten nicht abnimmt und der Algorithmus konvergiert.¹⁾ Dafür ist eine Elitismusrate von $\epsilon = 1/P$ ausreichend. Höhere Elitismusraten schränken in der Regel die Populationsentwicklung zu stark ein und führen zu einer verfrühten Konvergenz auf ein lokales Minimum.

Die übrigen Plätze der folgenden Generation $POP^{(\tau+1)}$ werden in den Zeilen 18 bis 27 gefüllt. Zuerst wird ein Elternpaar durch *Turnierauswahl* bestimmt. Für jedes Turnier werden T Individuen ohne Zurücklegen zufällig aus $POP^{(\tau)}$ gezogen. Der fitteste Turnierteilnehmer gewinnt und darf sich fortpflanzen. Nach dem Turnier werden alle Teilnehmer – auch der Gewinner – in die aktuelle Population zurückgelegt, können also an einem weiteren Turnier teilnehmen. Die Turnierauswahl gilt als eine der besten Selektionsstrategien, weil sie durch die zufällige Auswahl der Turnierteilnehmer – unabhängig von ihrer Fitness – eine zu schnelle Konvergenz auf eine Lösung verhindert.²⁾ Die Anzahl der Turnierteilnehmer sollte dafür nicht zu groß sein, da mit der Turniergröße der Selektionsdruck, der wiederum Diversität reduziert, wächst.

Nach zwei Turnieren enthält die *AUSWAHL*-Menge ein Elternpaar (Zeile 19), das seine Chromosomen mit Wahrscheinlichkeit p_χ kreuzt (Zeile 20).

¹⁾ Vgl. Kruse u. a. [Intelligence] 182; Rudolph [Convergence] 859.

²⁾ Vgl. auch im Weiteren Kruse u. a. [Intelligence] 181; Whitley/Sutton [Genetic Algorithms] 642 ff..

Wenn durch eine Elitismusrate $\epsilon \geq 1/p$ sichergestellt ist, dass gute Lösungen erhalten bleiben, kann eine Kreuzungswahrscheinlichkeit p_χ nahe 1 gewählt werden.

Bei der hier implementierten K -Punkt-Kreuzung (Zeile 21) werden K Chromosom-Schnittpunkte zufällig gesetzt, zwischen denen die beiden Individuen Teilstrings tauschen (siehe Abb. D-51).

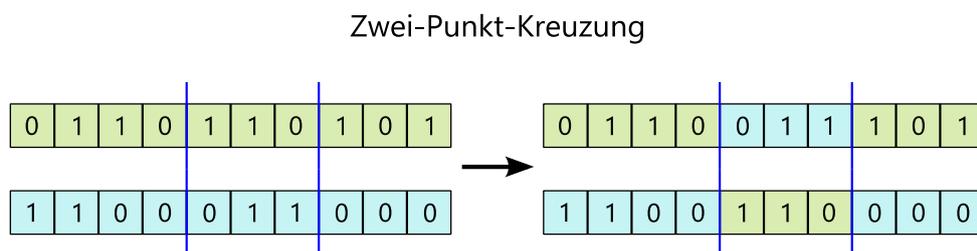


Abb. D-51: Beispiel einer K -Punkt-Kreuzung mit $K = 2$

Je größer die Anzahl der Schnittpunkte ist, desto *disruptiver* ist die Kreuzung, d. h. es steigt die Wahrscheinlichkeit, dass zwei nebeneinanderliegende Gene auseinandergerissen werden.¹⁾ Das kann einerseits schlecht sein, wenn dadurch eine gute Teillösung – beispielsweise eine vorteilhafte Anzahl verdeckter Neuronen in Gray-Kode – zerstört wird. Andererseits fördern disruptive Kreuzungen die Diversität – es können weit entfernt liegende Lösungen entstehen, die ohne Disruption deutlich später oder überhaupt nicht erreicht worden wären; der Lösungsraum wird also umfangreicher erkundet.

Die beiden in der *AUSWAHL*-Menge enthaltenen Individuen (gekreuzt oder nicht) mutieren mit Wahrscheinlichkeit p_μ (Zeile 23). Wenn sie mutieren, wird jeweils ein Gen zufällig ausgesucht und negiert (Zeile 24). Die Wahrscheinlichkeit eines einzelnen Gens, dass es mutiert, beträgt demnach

¹⁾ Vgl. auch im Weiteren Kruse u. a. [Intelligence] 190; Whitley/Sutton [Genetic Algorithms] 648 f.

$p_{\mu} \cdot 1/Chromosomlänge$. Für Bitstrings wird in diesem Zusammenhang eine Mutationswahrscheinlichkeit p_{μ} nahe 1 empfohlen.¹⁾

e) Struktur der Heuristikklassen

Das Klassendiagramm in Abb. D-52 zeigt die Struktur der Klassen im Cenobi-Heuristikpaket.²⁾ Sie ähnelt der Struktur der Trainingsklassen (siehe Abb. D-36, S. 246). Die Klasse `HeuristicSettings` speichert die auf der Benutzeroberfläche getroffenen Einstellungen (siehe Abb. D-50). Die Operation `createHeuristic()` eines `HeuristicSettings`-Objekts liefert ein konkretes `Heuristic`-Objekt der Art `heuristicType μ` zurück, das das `HeuristicSettings`-Objekt selbst als Attribut enthält und so auf diese Einstellungen zugreifen kann.

Die abstrakte Klasse `Heuristic` definiert die gemeinsamen Attribute und Operationen aller Heuristiken. Gemeinsame Attribute sind neben den `heuristicSettings` vor allem die Datensätze sowie ein `ArchitectureSettings`- und ein `TrainingSettings`-Objekt. Damit kann ein `Heuristic`-Objekt über die jeweils angebotenen `create...()`-Operationen ein neuronales Netz und einen Trainingsalgorithmus entsprechend der Benutzereinstellungen erzeugen. Zudem ist eine abstrakte Operation `startOptimization()` definiert, die in jeder Unterklasse individuell umgesetzt wird und eine Liste mit (untrainierten) neuronalen Netzen zurückgibt.

Zur abstrakten `Heuristic`-Klasse gibt es zwei konkrete Unterklassen. Die Klasse `NoHeuristic` dient im Grunde nur dazu, die Cenobi-Gesamtstruktur zu vereinfachen. Bei einem Aufruf von `startOptimization()` erzeugt sie ein neuronales Netz, wie es in den `architectureSettings` und durch die Anzahl der Variablen in den Input- und Output-Datensätzen vorgegeben ist und

¹⁾ Vgl. Kruse u. a. [Intelligence] 185.

²⁾ Für eine Gesamtübersicht der Cenobi-Paketstruktur siehe Abb. D-6, S. 187.

gibt eine Liste mit eben diesem Netz als einzigem Element zurück. Falls minNoHiddenNeurons und maxNoHiddenNeurons nicht übereinstimmen (siehe Abb. D–49), zieht es eine zufällige Anzahl verdeckter Neuronen in diesem Bereich.

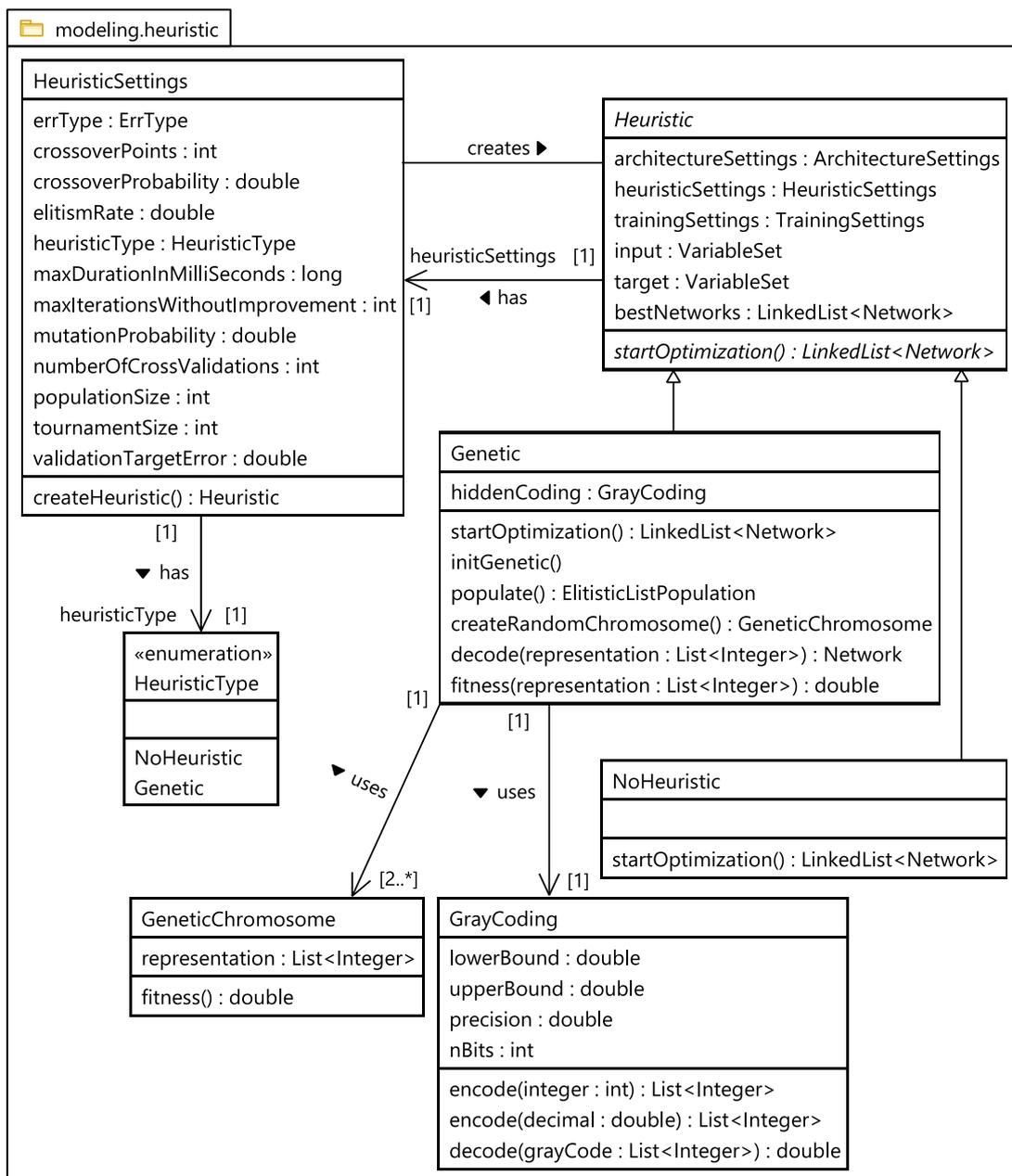


Abb. D–52: Klassendiagramm des Heuristikpakets

Die Klasse Genetic hingegen führt beim Aufruf von `startOptimization()` den oben beschriebenen Algorithmus D–3 aus und gibt eine Liste mit den Phä-

notypen der letzten Generation zurück (siehe Zeile 30 in Algorithmus D–3). Die übrigen Operationen der Genetic-Klasse sind Hilfsfunktionen, die innerhalb von `startOptimization()` aufgerufen werden, um u. a. ein `GrayCoding`-Objekt zu erzeugen und die Fitness der `GeneticChromosome`-Objekte zu bewerten. Zudem benutzt `Genetic` mehrere Klassen der freien Methodenbibliothek `Commons-Math 3.3` von Apache (in Abb. D–52 nicht dargestellt), die grundlegende Bausteine für einen genetischen Algorithmus wie z. B. Auswahl-, Kreuzungs- und Mutationsoperatoren bereitstellt.

3. Empirische Ergebnisse der metaheuristischen Optimierung

Um zu überprüfen, ob die Optimierung der Netzarchitektur durch einen genetischen Algorithmus tatsächlich die Prognosegenauigkeit verbessert, wurde der in Abschnitt 2 vorgestellte Algorithmus mit denselben Daten (siehe Anhang A3) und Trainingseinstellungen getestet wie die Trainingsalgorithmen in Abschnitt X.2 (S. 254 ff.). Der genetische Algorithmus war wie in Abb. D–50 (S. 277) und die verwendeten Lernalgorithmen wie in Abschnitt X.2 eingestellt. Es wurde kein *early stopping* angewendet.

Dem genetischen Algorithmus wurden die ersten 57 Beobachtungen (1/2009 bis 9/2013) des Datensatzes übergeben. Von diesen 57 Beobachtungen nutzte der Algorithmus in 10 Kreuzvalidierungsdurchläufen jeweils $\frac{9}{10}$ zum Netztraining und $\frac{1}{10}$ zur Validierung ($S = 10$). Nach dem Ende des genetischen Algorithmus wurde aus der letzten Generation nur das fitteste Individuum ausgewählt und auf alle 57 Beobachtungen trainiert. Dieses abschließende Training der besten Netzarchitektur wurde zehnmal neu initialisiert, ausgewertet und die Testergebnisse gemittelt. Getestet wurde der Prognosefehler über die verbleibenden 6 Beobachtungen (10/2013 bis 3/

2014), die weder im genetischen Algorithmus, noch beim anschließenden Training verwendet wurden. Abb. D–53 zeigt die Prognosefehler nach (D.12) und (D.13) für die 6 Beobachtungen der Testmenge sowie die Merkmale der optimierten Netzarchitektur und einige Leistungskennzahlen des genetischen Algorithmus in Verbindung mit dem jeweils beteiligten Trainingsalgorithmus.

verwendeter Trainingsalgorithmus	<i>back-propagation</i>	<i>resilient propagation</i>	<i>extreme learning machine</i>
NMAE über die Testmenge ¹⁾	0,1213	0,1221	0,1237
MAPE _S über die Testmenge ¹⁾	0,1603	0,1606	0,1631
verbliebende Einflussgrößen ¹⁾	Zuhaltekraft	Spritzgussstd., Zuhaltekraft	Spritzgussstd., Zuhaltekraft
Anzahl verdeckter Neuronen ¹⁾	3	5	4
Fitness des besten Individuums	13,25	13,69	13,63
erfülltes Abbruchkriterium	Zeitbeschränkung (180 Sek.)	Zeitbeschränkung (180 Sek.)	10 Iter. ohne Verbesserung
Anzahl durchgeführter Iterationen	16	11	23
Rechenzeit ²⁾	180 Sekunden	180 Sekunden	7,5 Sekunden

Abb. D–53: Leistung des genetischen Algorithmus mit verschiedenen Trainingsalgorithmen

Im Vergleich mit Abb. D–40 (S. 255) fällt auf, dass der genetische Algorithmus die Prognosegenauigkeit für unbekannte Testdaten tatsächlich signifikant erhöht. Etwas erstaunlich ist, dass diese bessere Generalisierungsleistung mit nur einer bzw. zwei Einflussgrößen (von sechs im Datensatz enthaltenen) erreicht wird. Offenbar stören die übrigen Einflussgrößen die

¹⁾ Alle Testergebnisse und Angaben zur Architektur beziehen sich auf das fitteste Individuum der letzten Generation.

²⁾ Auf einem Intel i5-Prozessor mit 3,4 GHz Rechenleistung. Die Rechenzeit kann durch parallele Berechnungen auf mehreren Prozessorkernen verkürzt werden (vgl. van Heeswijk u. a. [Parallelized ELM]).

Schätzung mehr als sie ihr nutzen. Die gezeigten Ergebnisse sind robust und auch mit veränderten Algorithmeinstellungen reproduzierbar.

Der bereits in Abschnitt X.2 beobachtete Geschwindigkeitsunterschied zwischen den Trainingsalgorithmen wird durch die vielfache Kreuzvalidierung im genetischen Algorithmus multipliziert. Der Algorithmus mit *extreme learning machine* läuft mit großem Abstand am schnellsten und hat dabei nur eine geringfügig schlechtere Generalisierungsleistung. Für Anwendungen, bei denen mit ungeduldigen Nutzern gerechnet werden muss, ist die *extreme learning machine* demnach die bessere Wahl.

XIII. *Bessere Generalisierungsfähigkeit durch Neuronale-Netze-Ensembles*

1. Vorteilhaftigkeit einer Ensembleschätzung

Bisher wurde stets ein einzelnes neuronales Netz als Schätzmodell verwendet. Im Allgemeinen kann man die Prognosegüte jedoch steigern, wenn man, statt ein einzelnes Modell isoliert zu betrachten, mehrere Modelle kombiniert.¹⁾ Eine solche Modellkombination wird *Ensemble* oder *Komitee* genannt. Im einfachsten Fall trainiert man L verschiedene Modelle und nutzt als Prognosewert den Durchschnitt der Prognosen aller L Modelle. Die bessere Generalisierungsfähigkeit eines Neuronale-Netze-Ensembles gegenüber einzelnen neuronalen Netzen wurde bereits 1990 in zwei grundlegenden Aufsätzen beschrieben.²⁾

¹⁾ Vgl. z. B. die empirischen Ergebnisse bei Andrawis/Atiya/El-Shishiny [Combinations]; Adeodato u. a. [MLP Ensembles]; Graeber/Kleine [Combination] 424 ff.

²⁾ Vgl. Hansen/Salamon [Network Ensembles]; Lincoln/Skrzypek [Clustering].

Die Vorteilhaftigkeit einer Ensembleschätzung lässt sich am *bias-variance trade-off* verdeutlichen.¹⁾ Ein komplexes Modell mit vielen trainierbaren Parametern kann sich flexibel an verschiedene Zusammenhänge anpassen, die es in den Daten findet. Komplexe Modelle suchen demnach *unvoreingenommen* (*unbiased*) einen passenden Zusammenhang. Da sie sich aber nicht nur an den gesuchten Modellzusammenhang anpassen, sondern auch an das in den Daten enthaltene zufällige Rauschen (*noise*), entsteht eine hohe *Varianz* zwischen Modellen, die sich auf unterschiedliche Weise an das Rauschen anpassen oder die auf unterschiedliche (Teil-)Datensätze (mit unterschiedlichem Rauschen) trainiert werden. Abb. D-54 illustriert Letzteres. Hier wurden $L = 100$ gleichartige Modelle auf 100 verschiedene Datensätze trainiert, die jeweils $N = 25$ zufällig gezogene Datenpunkte einer verrauschten Sinuskurve enthalten.²⁾

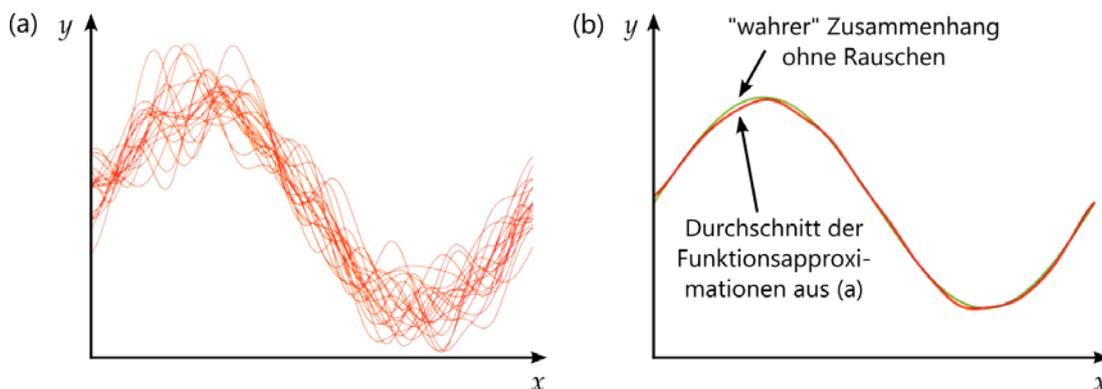


Abb. D-54: Funktionsapproximationen mit hoher Varianz (a), die im Durchschnitt den wahren Zusammenhang fast unverzerrt wiedergeben (b)³⁾

In Abb. D-54 a sind die Schätzungen der einzelnen Modelle dargestellt (um die Übersichtlichkeit zu bewahren, werden nur 20 Modelle gezeigt). In Abb.

¹⁾ Siehe dazu Abschnitt C.I.2, S. 88 ff. Vgl. auch im Weiteren Brown u. a. [Diversity Creation] 6 ff.; Bishop [Machine Learning] 149 ff., 655 f.

²⁾ Siehe auch Abb. C-5, S. 90.

³⁾ Bishop [Machine Learning] 150.

D-54 b sind das arithmetische Mittel der Schätzungen der 100 Modelle und der „wahre“ Zusammenhang – die unverrauschte Sinuskurve – eingezeichnet. Es ist deutlich zu erkennen, dass die kombinierte Schätzung besser als jede einzelne Schätzung ist.

Die grundlegende Idee der Ensemblebildung besteht also darin, hinreichend komplexe Modelle so zu kombinieren, dass sich das jeweils enthaltene Rauschen überlagert. Daraus ergeben sich drei grundlegende Bedingungen für die Ensemblebildung:¹⁾

- (1) Die Ensemblemitglieder sollen komplex genug sein, um eine unvoreingenommene Schätzung durchzuführen (siehe Abschnitt 2).
- (2) Die Ensemblemitglieder sollen vielfältig (*diverse*) sein (siehe Abschnitt 3). Im besten Fall bringt jedes Ensemblemitglied einzigartige, nützliche Informationen in die Ensembleschätzung ein.²⁾
- (3) Die einzelnen Schätzungen sollen sinnvoll kombiniert werden (siehe Abschnitt 5).

2. Auswahl guter Ensemblemitglieder

Wie in Abschnitt 1 erläutert wurde, müssen die einzelnen Modelle eines Ensembles komplex genug sein, um den gesuchten Zusammenhang unverzerrt zu schätzen. Ein Ensemble kann nicht die Verzerrungen von unter-spezifizierten Modellen, sondern nur die Überanpassungen hinreichend flexibler Modelle ausgleichen.³⁾ Das Ensemble soll zwar verschiedene, aber

¹⁾ Vgl. u. a. den grundlegenden Aufsatz von Bates/Granger [Combination].

²⁾ Vgl. Graeber/Kleine [Combination] 420, mit weiteren Nachweisen.

³⁾ Vgl. Brown u. a. [Diversity Creation] 7.

keine „schlechten“ Modelle enthalten.¹⁾ Das Problem der Modellselektion bleibt also bestehen.

Nachdem in Abschnitt XI.1 (S. 258 ff.) gezeigt wurde, dass der Kreuzvalidierungsfehler ein geeignetes Modellselektionskriterium ist, und in Teilkapitel XII (S. 267 ff.) gezeigt wurde, dass evolutionäre Algorithmen eine effiziente Modelloptimierung ermöglichen, ist es nicht verwunderlich, dass evolutionäre Algorithmen in Verbindung mit Kreuzvalidierungstechniken häufig zur Auswahl von Ensemblemitgliedern, insbesondere auch für Neuronale-Netze-Ensembles, genutzt werden.²⁾

Es gibt zwei naheliegende Möglichkeiten, einen evolutionären Algorithmus zur Auswahl von Ensemblemitgliedern zu verwenden:

(1) Man führt während der Evolution eine Ensembleliste mit, in der man die besten Individuen jeder Generation sammelt (siehe Abb. D–55).³⁾

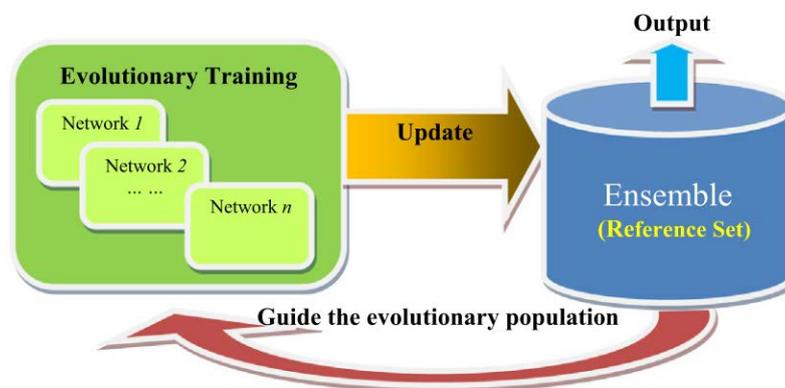


Abb. D–55: Evolutionäre Ensemblebildung mit einer mitlaufenden Ensembleliste⁴⁾

¹⁾ Vgl. Opitz/Shavlik [Network Ensemble] 339; Zhou/Wu/Tang [Ensembling]; Andrawis/Atiya/El-Shishiny [Combinations] 674; jeweils mit weiteren Nachweisen.

²⁾ Speziell zu evolutionär optimierten Neuronale-Netze-Ensembles vgl. u. a. Opitz/Shavlik [Network Ensemble]; Yao/Liu [Population]; Zhou/Wu/Tang [Ensembling]; Islam/Yao [Evolving Ensembles]; Soares/Antunes/Araújo [Comparison]; Xue u. a. [Genetic Ensemble]; Chen/Tseng/Wu [Evolutionary Training].

³⁾ Vgl. Chen/Tseng/Wu [Evolutionary Training] 353 f.

⁴⁾ Chen/Tseng/Wu [Evolutionary Training] 353.

(2) Die besten Individuen der letzten Generation bilden das Ensemble.¹⁾

Wenn durch Elitismus²⁾ sichergestellt ist, dass die letzte Generation auch das insgesamt beste Individuum enthält, ist Variante (2) die elegantere Lösung. Sie wird auch in Cenobi genutzt. Allerdings hat Variante (1) den Vorteil, dass für die Aufnahme in das Ensemble andere Kriterien definiert werden können als nur die Fitness der Individuen. Dadurch kann hier flexibler auf die Diversität der Ensemblemitglieder eingegangen werden (s. u.).

Beispielhaft zeigt Abb. D–56 die letzte Generation des genetischen Algorithmus mit *extreme learning machine* aus dem Experiment in Abschnitt XII.3 (S. 282 ff.). Sie enthält neben einer „Spitzengruppe“ mit ähnlich guten Fitnesswerten auch deutlich schlechtere Individuen, die nicht in das Ensemble aufgenommen werden sollten.

Fitness 13.62595981503784. Chromosom [0, 1, 1, 0, 0, 0, 0, 0, 1, 0], Anz. verd. Neuronen 4.
Fitness 12.311157975788781. Chromosom [0, 1, 1, 0, 0, 0, 0, 1, 1, 0], Anz. verd. Neuronen 5.
Fitness 12.262303566200487. Chromosom [0, 1, 1, 0, 0, 0, 0, 1, 1, 0], Anz. verd. Neuronen 5.
Fitness 11.827801334588846. Chromosom [0, 1, 1, 0, 0, 0, 0, 1, 1, 1], Anz. verd. Neuronen 6.
Fitness 11.42530431642636. Chromosom [1, 1, 1, 0, 0, 0, 0, 1, 1, 0], Anz. verd. Neuronen 5.
Fitness 11.332341869791744. Chromosom [0, 1, 0, 0, 0, 0, 0, 1, 1, 0], Anz. verd. Neuronen 5.
Fitness 11.120048030948254. Chromosom [0, 1, 0, 0, 0, 0, 0, 1, 1, 0], Anz. verd. Neuronen 5.
⋮
Fitness 9.297290912472572. Chromosom [1, 1, 1, 0, 1, 0, 0, 1, 1, 1], Anz. verd. Neuronen 6.
Fitness 8.308181367947034. Chromosom [0, 0, 1, 0, 0, 0, 0, 1, 1, 0], Anz. verd. Neuronen 5.
Fitness 7.794848988042314. Chromosom [0, 1, 1, 0, 1, 1, 0, 0, 1, 0], Anz. verd. Neuronen 4.
Fitness 7.4814271165762865. Chromosom [0, 1, 1, 0, 0, 1, 0, 0, 1, 0], Anz. verd. Neuronen 4.
Fitness 6.771948991968771. Chromosom [0, 1, 1, 0, 1, 0, 1, 0, 1, 0], Anz. verd. Neuronen 13.
Fitness 6.273250778167713. Chromosom [0, 1, 1, 1, 0, 1, 0, 0, 1, 0], Anz. verd. Neuronen 4.
Fitness 2.648896535382431. Chromosom [0, 1, 1, 0, 0, 0, 1, 0, 1, 0], Anz. verd. Neuronen 13.

Abb. D–56: Finale Generation des genetischen Algorithmus mit extreme learning machine aus Abb. D–53

¹⁾ Vgl. Opitz/Shavlik [Network Ensemble] 338; Yao/Liu [Population]; Xue u. a. [Genetic Ensemble].

²⁾ Siehe S. 278.

Wie viele „gute“ Individuen in der letzten Generation sind, hängt von den Datensätzen und den gewählten Einstellungen ab. Deshalb wäre es problematisch, immer eine feste Anzahl L aus der letzten Generation in das Ensemble zu übernehmen. Stattdessen werden in Cenobi das fitteste Individuum sowie alle Individuen, deren Fitness maximal zwanzig Prozent schlechter ist, in das Ensemble aufgenommen. Diese schlichte Auswahlregel hat den Vorteil, dass keine weiteren Berechnungen oder zusätzliche Validierungsmengen notwendig sind. Wenn allerdings genug Daten für weitere Validierungen zur Verfügung stehen, kann durch anspruchsvollere Auswahlverfahren die Generalisierungsleistung des Ensembles noch weiter verbessert werden.¹⁾ Da Kostenrechnungsdaten aber grundsätzlich knapp sind, bleibt es in Cenobi bei der einfachen Auswahlregel mit dem Fitnesswert als einzigem Kriterium.

3. Sicherstellung der Vielfalt im Ensemble

Ein Ensemble ist nur dann vorteilhaft, wenn die Ensemblemitglieder *vielfältig* sind. Damit ist gemeint, dass sie verschiedene Fehler machen, die sich gegenseitig ausgleichen. Formal kann gezeigt werden, dass der Prognosefehler des Ensembles umso kleiner ist, je weniger die Prognosefehler der einzelnen Modelle korrelieren.²⁾ Der Ensemblefehler ist also minimal, wenn die individuellen Fehler der Ensemblemitglieder perfekt negativ korreliert sind. Anschaulich bedeutet eine perfekt negative Korrelation der Prognosefehler von zwei Modellen, dass das eine Modell den wahren Wert immer zu hoch schätzt, wenn das andere Modell ihn zu niedrig schätzt.

¹⁾ Vgl. z. B. Yao/Liu [Population] 423 f.

²⁾ Vgl. Sharkey [Combining] 305; Brown u. a. [Diversity Creation] 8.

In der wissenschaftlichen Literatur werden zahlreiche Techniken vorgeschlagen, um die gewünschte Diversität in das Ensemble einzubringen. Oft werden verschiedene Modelltypen – z. B. künstliche neuronale Netze, intrinsisch lineare Modelle¹⁾ und physikalisch-technische Modelle – kombiniert.²⁾ Diese Dissertation beschränkt sich jedoch auf Ensembles die ausschließlich aus künstlichen neuronalen Netzen bestehen. Dabei verbleiben trotzdem noch zahlreiche Möglichkeiten, für Diversität zu sorgen. Grundsätzlich können

- (1) Gewichtsinitialisierungen,
 - (2) Trainingsdaten,
 - (3) Netzarchitekturen und
 - (4) Lernverfahren
- variiert werden.³⁾

Zu (1): Unterschiedliche Gewichtsinitialisierungen sorgen bei gradientenbasierten Lernverfahren wie *backpropagation* und *resilient propagation* nur für wenig Diversität. In der Regel konvergieren die Netze auch mit unterschiedlichen Initialgewichten auf ähnliche lokale Minima.⁴⁾ Anders ist es bei der *extreme learning machine*. Da hier die in die verdeckte Schicht eingehenden Gewichte nicht verändert werden, hat die Initialisierung einen größeren Einfluss auf die gefundene Lösung (siehe S. 122). Dementsprechend zeigen ELM-Ensembles, selbst wenn ihre Diversität allein durch Variation

¹⁾ Siehe (C.42), S. 150.

²⁾ Vgl. z. B. Andrawis/Atiya/El-Shishiny [Combinations] 674 f.; Graeber/Kleine [Combination] 419 f.

³⁾ Vgl. Sharkey [Combining] 301; Brown u. a. [Diversity Creation] 12 f.; Soares/Antunes/Araújo [Comparison] 499.

⁴⁾ Vgl. Sharkey [Combining] 308; Brown u. a. [Diversity Creation] 13; Soares/Antunes/Araújo [Comparison] 499; jeweils mit weiteren Nachweisen.

der Initialgewichte erzeugt wird, in vergleichenden Experimenten gute Resultate.¹⁾

Zu (2): Die Variation der Trainingsdaten wird am häufigsten eingesetzt und ist, wie auch das Beispiel in Abschnitt 1 zeigt, im Allgemeinen recht effektiv.²⁾ Die beiden bekanntesten Ensemble-Techniken – *bagging* (*bootstrap aggregating*)³⁾ und *boosting*⁴⁾ – gehören in diese Kategorie.

Zu (3): Seltener wird bei der Ensemblebildung die Netzarchitektur variiert. Das liegt vermutlich daran, dass eine Veränderung der Netzarchitektur schnell dazu führt, dass die Schätzung des veränderten Netzes so schlecht wird, dass es für das Ensemble nicht mehr brauchbar ist (siehe Abschnitt 2). Die richtige Balance zwischen Modellgüte und Diversität zu finden, ist hier besonders schwierig. Opitz und Shavlik nutzen deshalb einen genetischen Algorithmus zur Ensemblebildung, der ähnlich wie Algorithmus D-3 die Netzarchitektur optimiert.⁵⁾ In die individuelle Fitnessberechnung fließt allerdings nicht nur die Prognosegenauigkeit des Individuums, sondern auch die Abweichung der individuellen Prognose vom Mittelwert aller Prognosen der aktuellen Generation ein. Als multikriterielle Bewertungsformel für den Fitnesswert des i . Individuums schlagen sie vor:

$$F_i = (1 - E_{\text{val},i}) + \lambda \left\| \hat{Y}_{\text{val},i} - \frac{1}{P} \sum_{j=1}^P \hat{Y}_{\text{val},j} \right\| \quad (\text{D.16})$$

mit $E_{\text{val},i}$ Fehler des i . Individuums über die Validierungsmenge,

¹⁾ Vgl. Sun u. a. [Sales Forecasting]; Xue u. a. [Genetic Ensemble] 177 ff.

²⁾ Vgl. auch im Weiteren Sharkey [Combining] 306 ff.; Brown u. a. [Diversity Creation] 13 f., mit zahlreichen weiteren Nachweisen; Bishop [Machine Learning] 656 ff..

³⁾ Vgl. Breiman [Bagging].

⁴⁾ Vgl. Schapire/Singer [Boosting]; Friedman [Boosting]; kritisch Long/Servedio [Boosters].

⁵⁾ Vgl. auch im Weiteren Opitz/Shavlik [Network Ensemble] 339 f.

$\hat{Y}_{\text{val},i}$	Matrix mit Schätzwerten für die abhängigen Variablen über alle Validierungspunkte,
P	Populationsgröße,
λ	Gewichtungsfaktor.

Dadurch werden in der Evolutionsphase sowohl die Prognosegüte als auch die Diversität maximiert.

Zu (4): Die Zielfunktion des Lernverfahrens kann dahingehend erweitert werden, dass nicht nur der individuelle Trainingsfehler, sondern auch die Korrelation zu den Fehlern der übrigen Modelle minimiert wird. Ein besonders erfolgreicher Ansatz ist das *negative correlation learning*.¹⁾ Dabei wird die Zielfunktion (C.7) (S. 94) um einen Korrelationsterm ergänzt und die Ensemblemitglieder werden simultan trainiert. Die Zielfunktion für das i . Ensemblemitglied lautet dann

$$\begin{aligned} \min \quad & \frac{1}{2NO} \sum_{n=1}^N \sum_{k=1}^O \|\hat{y}_{k,i}^{(n)} - y_k^{(n)}\| \\ & + \frac{\lambda}{NO} \sum_{n=1}^N \sum_{k=1}^O \left((\hat{y}_{k,i}^{(n)} - y_k^{(n)}) \sum_{j \neq i} (\hat{y}_{k,j}^{(n)} - y_k^{(n)}) \right) \end{aligned} \quad (\text{D.17})$$

mit $\hat{y}_{k,i}^{(n)}$	Schätzwert des i . Ensemblemitglieds für Beobachtung n der abhängigen Variablen k ,
λ	Gewichtungsfaktor.

Der Rechenaufwand für das Training mit (D.17) ist recht hoch. Er kann aber z. B. dadurch reduziert werden, dass die Netze individuell, d. h. mit der einfachen Zielfunktion (C.7) (S. 94), trainiert werden und die multikriterielle Zielfunktion (D.17) – ähnlich wie (D.16) – nur zur Fitnessberechnung

¹⁾ Vgl. auch im Weiteren Liu/Yao [Negative Correlation]; Liu/Yao [Simultaneous Training]; Liu/Yao/Higuchi [Negative Correlation Learning]; Brown [Diversity]; Brown u. a. [Diversity Creation] 16 f.; Tang u. a. [Negative Correlation Learning].

über die Validierungsmenge in einem evolutionären Algorithmus herangezogen wird.¹⁾

4. Implementierung der Ensemblebildung in Cenobi

In Cenobi werden bei der Ensemblebildung die Initialgewichte, die Trainingsdaten und die Netzarchitektur variiert. Algorithmus D-4 zeigt die Ensemblebildung in Pseudocode. Variationen der Architektur ergeben sich daraus, dass aus der letzten Generation des genetischen Algorithmus (Algorithmus D-3, S. 277) nicht nur das fitteste Individuum, sondern auch die Individuen, die weniger als zwanzig Prozent schlechter sind, übernommen werden (Zeile 5; siehe auch Abschnitt 2, S. 286 ff.). Zudem wird der genetische Algorithmus mehrmals neu initialisiert und durchlaufen (Zeile 3 bis 6), um der Gefahr entgegenzuwirken, dass nur ein einziges, besonders fittes Individuum ausgewählt wird. Der in Cenobi implementierte Algorithmus wiederholt die Zeilen 4 und 5 standardmäßig viermal, so dass die *ARCHITEKTUREN*-Menge immer mindestens vier Elemente enthält.

Die Sortierung der Architekturen nach absteigender Fitness (Zeile 7) sorgt dafür, dass Architekturen mit höherer Fitness häufiger im Ensemble enthalten sind, falls L kein Vielfaches der Anzahl der Architekturen $|ARCHITEKTUREN|$ ist.

Eine Variation der Trainingsdaten kommt bei unvollständigen Datensätzen schon dadurch zustande, dass für jedes Ensemblemitglied ein neuer imputierter Datensatz durch bayessches *predictive mean matching* gewonnen wird (Zeile 18; siehe ausführlich Abb. D-24, S. 228).

¹⁾ Vgl. Chen/Tseng/Wu [Evolutionary Training] 353.

Algorithmus Ensemblebildung in Cenobi

```

1: TRAINIERT ← ∅
2: ARCHITEKTUREN ← ∅
3: wiederhole
4:   Genetische Optimierung der Netzarchitektur (Algorithmus D–3).
   ARCHITEKTUREN ←
5:    $ARCHITEKTUREN \cup \{MODELL_i \in POP^{(final)} \mid F_i > 0,8 \cdot F^{max}\}$ 
6: bis  $|ARCHITEKTUREN|$  ausreichend
7: Sortiere ARCHITEKTUREN in absteigender Fitness-Reihenfolge.
8:  $N \leftarrow$  Anzahl der Datenpunkte
9:  $L \leftarrow$  Ensemblegröße
10: wenn  $L > N$  dann
11:    $S \leftarrow N$ 
12: sonst
13:    $S \leftarrow L$ 
14: ende wenn
15:  $j \leftarrow 1$ 
16:  $k \leftarrow 1$ 
17: für  $i = 1$  bis  $L$  führe aus
18:   DATEN ← erzeuge einen vollständigen, imputierten Datensatz
   (Algorithmus D–2).
19:   wenn earlyStopping = wahr dann
20:     wenn  $k > S$  dann
21:        $k \leftarrow 1$ 
22:     ende wenn
23:     Teile DATEN in  $S$  Teilmengen.
24:     VALIDIERUNG ←  $k$ . Teilmenge
25:     TRAINING ← übrige Teilmengen
26:      $k \leftarrow k + 1$ 
27:   sonst
28:     TRAINING ← DATEN \ zufällig gewählter Punkt
29:   ende wenn

```

```

30:  wenn  $j > |\text{ARCHITEKTUREN}|$  dann
31:       $j \leftarrow 1$ 
32:  ende wenn
33:   $\text{NETZ} \leftarrow$  initialisiere die  $j$ . Architektur mit zufälligen Gewichten.
34:  Trainiere  $\text{NETZ}$  auf die Menge  $\text{TRAINING}$ ,
35:      stoppe ggf., wenn  $(GL^{(\tau)} > 5) \wedge (\tau \geq 100)$ .          (D.15) (S. 264)
36:   $\text{TRAINIERT} \leftarrow \text{TRAINIERT} \cup \text{NETZ}$ 
37:   $j \leftarrow j + 1$ 
38: ende für
39: Rückgabe  $\text{TRAINIERT}$ 

```

Algorithmus D-4: Ensemblebildung in Cenobi

Ferner wird, wenn die Option *early stopping* gewählt ist, ein Rotationstraining angewandt, bei dem jedes Ensemblemitglied eine andere Validierungsmenge erhält (Zeile 20 bis 26; siehe auch S. 266). Falls kein *early stopping* stattfindet, wird zufällig ein Punkt aus der Trainingsmenge gelöscht (Zeile 28).

Der Iterationsindex j wird immer wieder zurückgesetzt, wenn er die *ARCHITEKTUREN*-Menge durchlaufen hat (Zeile 30 bis 32). Die gleiche Architektur kann demnach mehrmals im Ensemble vorkommen. Sie wird aber stets mit neuen zufälligen Gewichten initialisiert (Zeile 33) und auf eine andere Trainingsmenge trainiert (Zeile 34). Das trainierte Netz wird der Menge *TRAINIERT* hinzugefügt (Zeile 36).

In Algorithmus D-4 wird die Ensemblediversität eher zufällig erzeugt. Eine explizite Steuerung der Diversität über multikriterielle Zielfunktionen wie in (D.16) oder (D.17) sowie die Anwendung eines Boosting-Algorithmus könnten die Generalisierungsfähigkeit des Ensembles noch weiter verbes-

sern. In Cenobi wurde jedoch wegen des hohen Implementierungs- und Rechenaufwandes darauf verzichtet.

Das Klassendiagramm in Abb. D-57 zeigt die Klassen des Ensemblepakets in Cenobi (siehe auch Abb. D-22, S. 225). Algorithmus D-4 ist in der start()-Operation der Klasse Ensemble implementiert. Die Ensemblegröße L ist auf der Benutzeroberfläche einstellbar und wird in einem EnsembleSettings-Objekt gespeichert. Da das Ensemble-Objekt während der Ensemblebildung einen genetischen Algorithmus zur Optimierung der Netzarchitektur sowie die Trainer für das Training der neuronalen Netze erzeugt, erhält es von der Benutzeroberfläche neben den EnsembleSettings auch die dort gewählten ArchitectureSettings, HeuristicSettings und TrainingSettings und speichert diese als Attribute.

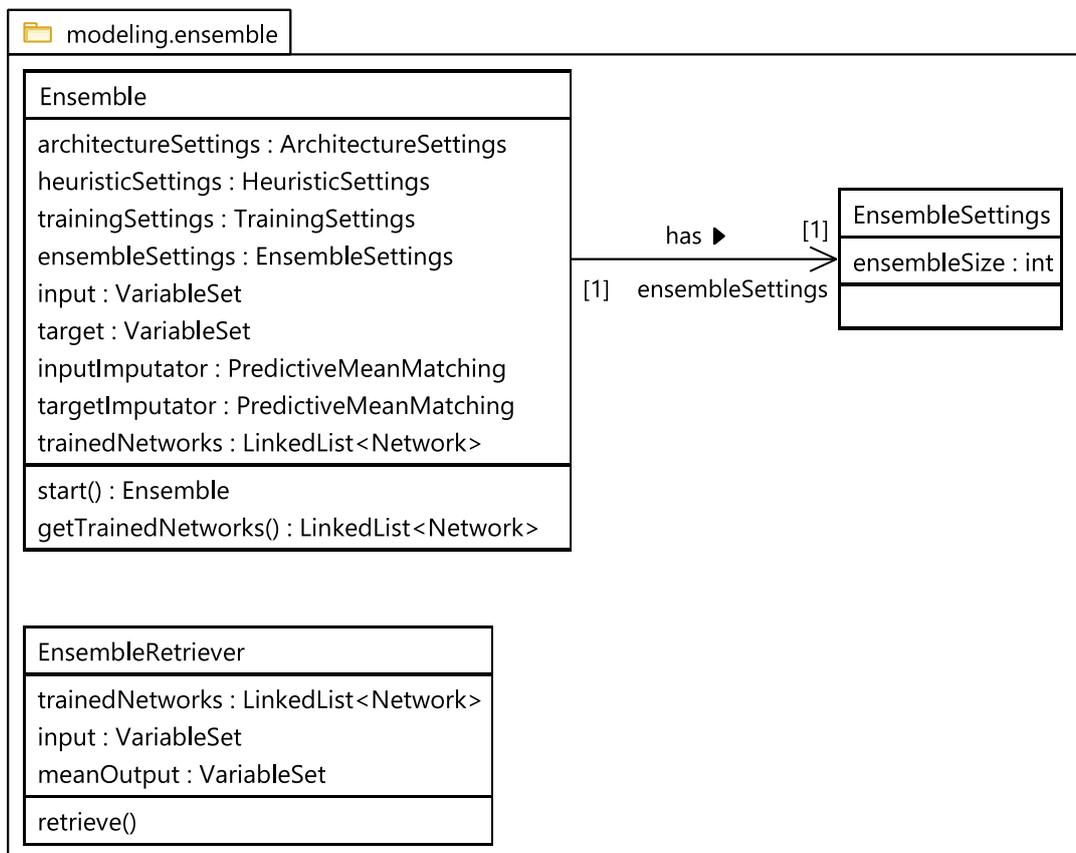


Abb. D-57: Klassendiagramm des Ensemblepakets

Der EnsembleRetriever ruft für einen gegebenen input die Outputs aller neuronalen Netze in der Liste trainedNetworks ab und berechnet daraus das mit den Fitnesswerten der Netze gewichtete arithmetische Mittel (meanOutput).

5. Kombination der einzelnen Schätzungen

Zahlreiche empirische Arbeiten haben gezeigt, dass einfache Aggregationsmethoden eine bessere Generalisierungsleistung ermöglichen, als Methoden, die zusätzliche Parameter in die Ensembleschätzung einbringen.¹⁾ Deshalb ist es i. A. vorteilhaft, einfach das arithmetische Mittel oder besser noch den Median der Einzelmodellschätzungen als Ensembleschätzung anzusetzen.²⁾ Wenn man allerdings weiß, dass die einzelnen Modelle unterschiedlich gut sind, ist es durchaus sinnvoll, den besseren Modellen ein größeres Gewicht zu geben. Als Indikator für die Modellgüte dient in der Regel der Prognosefehler über eine Validierungsmenge (bzw. mehrere Validierungsmengen im Falle einer Kreuzvalidierung).³⁾

Da der Fitnesswert der Individuen in Algorithmus D-3 (Zeile 14, S. 276) gerade der Kehrwert des Kreuzvalidierungsfehlers ist, bietet es sich an, die Outputs der einzelnen trainierten Netze mit eben diesem Wert zu gewichten. In Cenobi geschieht das in der retrieve()-Operation des EnsembleRetriervers (siehe Abb. D-57).

¹⁾ Vgl. auch im Weiteren Opitz/Shavlik [Network Ensemble] 339; Andrawis/Atiya/El-Shishiny [Combinations] 674; Graeber/Kleine [Combination] 421; jeweils mit weiteren Nachweisen.

²⁾ Vgl. Barrow/Crone/Kourentzes [Network Ensembles] 5 f.; Soares/Antunes/Araújo [Comparison] 505.

³⁾ Vgl. z. B. Bates/Granger [Combination] 452 f.; Opitz/Shavlik [Network Ensemble] 341; Andrawis/Atiya/El-Shishiny [Combinations] 674; Graeber/Kleine [Combination] 423 f.; Ando/Li [Model-Averaging] 256.

6. Empirische Ergebnisse der Ensembleschätzung

Die in Cenobi implementierte Ensembleschätzung wurde mit denselben Daten (siehe Anhang A3) und Einstellungen getestet wie der genetische Algorithmus in Abschnitt XII.3 (S. 282 ff.) und die Lernverfahren in Abschnitt X.2 (S. 254 ff.). Dem Ensemble-Objekt wurden die ersten 57 Beobachtungen (1/2009 bis 9/2013) zur Ensemblebildung übergeben. Anschließend wurden dem EnsembleRetriever die trainierten Netze sowie die verbleibenden 6 Inputwerte (10/2013 bis 3/2014) übergeben. Für die zurückgegebene Ensembleschätzung wurden der normierte mittlere absolute Fehler (D.12) sowie der symmetrische mittlere absolute prozentuale Fehler (D.13) berechnet. Dieses Vorgehen wurde mit verschiedenen Ensemblegrößen L und unterschiedlichen Trainingsalgorithmen jeweils viermal wiederholt. Die durchschnittlichen Fehler aus diesen vier Durchgängen sind in Abb. D-58 dargestellt.

Die besten Ergebnisse wurden mit einer Ensemblegröße von $L = 10$ und der *extreme learning machine* als Trainingsalgorithmus erzielt. Warum *early stopping* die Testergebnisse zum Teil verschlechterte, konnte nicht abschließend geklärt werden. Möglicherweise kam es trotz des angewendeten Rotationstrainings zu einer Überanpassung an die Validierungsmenge (*overfitting* zweiter Ordnung, siehe S. 266). Insgesamt jedoch hatte das Neuronale-Netze-Ensemble durchweg einen geringeren Testfehler als die einzelnen Modelle in Abb. D-40 (S. 255) und Abb. D-53 (S. 283). Weitere Experimente mit den Datensätzen aus Anhang A4 bis A7 haben bestätigt, dass die *extreme learning machine* der stabilste, genaueste und mit Abstand schnellste der drei implementierten Trainingsalgorithmen ist und dass die Generalisierungsleistung des Ensembles ab einer Ensemblegröße von ungefähr $L = 20$ nicht mehr zunimmt. Aus diesem Grund ist die Ensembleschätzung mit *extreme learning machine* und $L = 20$ die Standardeinstellung in Cenobi.

verwendeter Trainingsalgorithmus	<i>backpropagation</i> ohne <i>early stopping</i>	<i>backpropagation</i> mit <i>early stopping</i>	<i>extreme learning machine</i>
Ensemblegröße $L = 5$			
NMAE über die Testmenge			0,1268
MAPE _S über die Testmenge			0,1681
Ensemblegröße $L = 10$			
NMAE über die Testmenge	0,1208	0,1214	0,1186
MAPE _S über die Testmenge	0,1597	0,1604	0,1567
Ensemblegröße $L = 30$			
NMAE über die Testmenge	0,1204	0,1203	0,1206
MAPE _S über die Testmenge	0,1592	0,1588	0,1592
Ensemblegröße $L = 50$			
NMAE über die Testmenge	0,1212	0,1230	0,1213
MAPE _S über die Testmenge	0,1601	0,1626	0,1606

Abb. D-58: Testergebnisse der Ensembleschätzung

Die empirischen Ergebnisse, die mit allgemeinen Grundeinstellungen der Algorithmen, d. h. ohne problemspezifisches *Tuning*¹⁾ erzielt wurden, bestätigen, dass mit dem genetisch optimierten Neuronale-Netze-Ensemble ein Schätzmodell zur Verfügung steht, das für die Schätzung betrieblicher Kostenfunktionen gut geeignet ist. Wie das folgende Kapitel E zeigt, ist es nicht schwer, dieses Modell – wenn es denn einmal aufgestellt ist – für kostenrechnerische Zwecke auszuwerten.

¹⁾ Vgl. Teytaud/Vazquez [Optimal Search Algorithm]; Bartz-Beielstein/Preuss [Tuning].

E. Anwendungen künstlicher neuronaler Netze in der Kostenrechnung

1. Problemstrukturierung durch transparente Darstellung der Kostenstruktur

1. Identifikation wichtiger Kosteneinflussgrößen

Die Identifikation relevanter Kosteneinflussgrößen erleichtert die Wahrnehmung und Strukturierung betrieblicher Kostenzusammenhänge und fördert dadurch die Problemerkennntnis.¹⁾ Relevant ist eine Einflussgröße dann, wenn sie die Kostenhöhe so stark beeinflusst, dass sie in den folgenden Phasen des Entscheidungsprozesses²⁾ berücksichtigt werden sollte.

Das Herausfiltern von irrelevanten Inputvariablen, die keinen Beitrag zur Erklärung der abhängigen Variablen leisten, ist eines der grundlegenden Ziele der Modellselektion.³⁾ In den Kapiteln D.XI und D.XII wurde dargelegt, dass sich dieses Ziel mit Kreuzvalidierung und genetischen Algorithmen gut erreichen lässt. Das genetisch optimierte Neuronale-Netze-Ensemble (siehe Kapitel D.XIII) enthält bereits nur noch die Inputvariablen, die einen merklichen Einfluss auf die Kostenhöhe haben.

Davon ausgehend wird in diesem Abschnitt eine Kennzahl gesucht, die es erlaubt, die verbleibenden Einflussgrößen nach ihrer Wichtigkeit zu ordnen. Mit *Wichtigkeit* ist hier nicht die absolute Höhe der variablen Kosten

¹⁾ Siehe Abb. B-9, S. 56, sowie ausführlich Abschnitt B.IV.2, S. 57 ff.

²⁾ Siehe Abb. B-2, S. 12.

³⁾ Siehe Abschnitt D.XI.1, S. 258.

einer Einflussgröße gemeint. Denn eine Einflussgröße x_1 mit niedrigen variablen Kosten kann einen größeren Einfluss auf die Kostenhöhe haben als eine Einflussgröße x_2 mit hohen variablen Kosten, wenn der Bereich, in dem die Werte von Variable x_1 schwanken, entsprechend größer ist als der Bereich von Variable x_2 . Das könnte etwa der Fall sein, wenn x_1 die produzierte Stückzahl und x_2 die Anzahl der genutzten Produktionsanlagen ist.

Die Frage ist vielmehr, welche Einflussgrößen für eine gute Prognose der abhängigen Variablen am wichtigsten sind. In der wissenschaftlichen Literatur werden zahlreiche Methoden vorgeschlagen, um diese Frage zu beantworten.¹⁾ Viele davon sind nur bedingt aussagekräftig.²⁾ Eine der wenigen Methoden, die sowohl theoretisch fundiert ist, als auch in vergleichenden Experimenten mit artifiziellen Datensätzen gute Ergebnisse zeigt, ist die Sensitivitätsanalyse der Prognosegenauigkeit durch systematische Randomisierung der Inputvariablen.³⁾ Algorithmus E-1 zeigt die Berechnung eines Wichtigkeitswertes W_i für jede unabhängige Variable x_i nach dieser Methode.

Für jedes trainierte neuronale Netz im Ensemble wird zunächst der Prognosefehler E^{orig} über alle verfügbaren Datenpunkte berechnet (Zeile 4). Anschließend werden der Reihe nach alle Einflussgrößen einzeln randomisiert

¹⁾ Vgl. Mozer/Smolensky [Relevance]; Garson [Connection Weights]; Lek u. a. [Environmental Variables]; Sung [Importance]; Olden/Jackson [Illuminating]; Gevrey/Dimopoulos/Lek [Review and Comparison]; Kingston/Maier/Lambert [Input Pruning]; Bowden/Dandy/Maier [Input Determination]; Gevrey/Dimopoulos/Lek [Sensitivity Analysis]; Watts/Worner [Relative Contribution]; Fernando/Maier/Dandy [Partial Mutual Information]; Hippert/Taylor [Bayesian Techniques]; Paliwal/Kumar [Contribution of Variables]; Fock [Global Sensitivity].

²⁾ Vgl. auch im Weiteren Sarle [Importance]; Olden/Joy/Death [Comparison]. Vgl. auch Mozer/Smolensky [Relevance] 4 f.

³⁾ Vgl. Scardi/Harding [Phytoplankton] 220; Kemp/Zaradic/Hansen [Importance].

zufällige Werte ersetzt werden. Entscheidend ist, dass für x_i nur Werte aus dem Intervall $[x_i^{\min}; x_i^{\max}]$ eingesetzt werden, da das Schätzmodell nur in diesem Bereich gilt.¹⁾

Die Randomisierung des Datensatzes und die Berechnung von $W_i^{j,h}$ sollte so lange wiederholt werden, bis sich mögliche Verzerrungen durch die zufällige Wahl der randomisierten Werte ausgleichen (siehe Zeile 6 bis 13).²⁾ In Cenobi wird diese Schleife so oft ausgeführt, bis die Menge $\{W_i^{j,h}\}$ für jede Inputvariable x_i mindestens zweihundert Elemente enthält. Als Wichtigkeitswert W_i wird der Median dieser Menge zurückgegeben (Zeile 15 bis 18).

Eine ähnliche Methode, die Wichtigkeit einer Variablen zu messen, besteht darin, nicht nur die Variablenwerte zu randomisieren, sondern die Variable insgesamt aus dem Modell herauszunehmen, um dann die Prognosefehler der Modelle mit und ohne die betreffende Variable zu vergleichen.³⁾ Ein Nachteil daran ist, dass für jede Variable ein neues Modell aufgestellt, trainiert und validiert werden muss. Das kann bei großen Modellen sehr lange dauern. Der Wichtigkeitswert nach Algorithmus E-1 lässt sich deutlich schneller berechnen, da dasselbe Modell für alle unabhängigen Variablen verwendet wird.

Abb. E-1 zeigt die nach Algorithmus E-1 berechneten Wichtigkeitswerte für die unabhängigen Variablen des bereits in den Abschnitten D.X.2 (S. 254 ff.), D.XII.3 (S. 282 ff.) und D.XIII.6 (S. 298 ff.) verwendeten Testdatensatzes (siehe Anhang A3). Wie nach den Experimenten in Kapitel D zu erwarten war, erweisen sich nur Zuhaltkraft und geleistete Spritzgussstun-

¹⁾ Siehe Abschnitt C.III.3.b, S. 137 f.

²⁾ Vgl. Kemp/Zaradic/Hansen [Importance] 328.

³⁾ Vgl. Mozer/Smolensky [Relevance] 4 f.

den als relevante Einflussgrößen, da die übrigen Einflussgrößen vom genetischen Algorithmus aussortiert wurden (siehe Abb. D–53, S. 283, und Abb. D–56, S. 288). Die absolute Höhe des Wichtigkeitswertes hängt u. a. vom verwendeten Fehlermaß ab und ist deshalb schwierig zu interpretieren. Die Abstände zwischen den Wichtigkeitswerten können aber durchaus so interpretiert werden, dass die Zuhaltekraft einen deutlich größeren Einfluss auf die Prognosegenauigkeit hat als die Spritzgussstunden.

Variable	Wichtigkeit
Zuhaltekraft gesamt in t	2,37
geleistete Std. Spritzgussmaschinen	0,65
Rohmaterialverbrauch in t	0,00
Anzahl verkaufter Produktarten	0,00
Fremdleistung Produktion in €	0,00
Anzahl eingesetzter Rohmaterialtypen	0,00

Abb. E–1: Wichtige Einflussgrößen im Fertigungsbereich¹⁾

Das in Abb. E–1 dargestellte Ergebnis ist einerseits interessant, weil es zeigt, dass die Kostenhöhe in den betrachteten Fertigungskostenstellen hauptsächlich von der Maschinengröße und nicht etwa von der Anzahl der Produktarten oder dem Fremdleistungsvolumen abhängt. Andererseits liegt die Vermutung nahe, dass der große Einfluss der Zuhaltekraft durch Abschreibungen zustande kommt, die nicht die tatsächliche Inanspruchnahme der Maschinenkapazität widerspiegeln. Schließlich ändert sich die Gesamtzuhaltekraft immer dann, wenn eine neue Maschine (mit einer bestimmten Zuhaltekraft) angeschafft wird. Wenn diese Maschine dann in den Fertigungskostenstellen beispielsweise linear abgeschrieben wird, steigen die monatlichen Kosten dieser Kostenstellen dauerhaft um einen gleichbleibenden Betrag. Die graphische Darstellung der Kostenfunktionen

¹⁾ Berechnet mit den Daten aus Anhang A3; Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D–50, S. 277.

in Abb. E-6 (S. 308) bestätigt diesen Eindruck. Der vom Neuronale-Netze-Ensemble ermittelte Zusammenhang ist also logisch nachvollziehbar; aufgrund der vermutlich nicht entscheidungsorientierten Abschreibungen ist er jedoch für Entscheidungszwecke wenig brauchbar.

Für einige andere Kostenstellen aus dem vorliegenden Beispieldatensatz¹⁾ gibt es in dieser Hinsicht erfreulichere Resultate. Auf Grundlage der Literaturübersicht in Abb. B-6 (S. 48) wurde für Kostenstellen aus verschiedenen Unternehmensbereichen eine Vorauswahl in Frage kommender Einflussgrößen getroffen und nach Algorithmus E-1 ausgewertet. Die Ergebnisse sind in den Abbildungen E-2 bis E-5 zu sehen.

Variable	Wichtigkeit
Anzahl Einkaufsrechnungen	4,06
Umsatz mit Handelsware in €	2,61
Anzahl verkaufter Produktarten	2,02
Umsatz mit eigener Produktion in €	0,00
Anzahl Kreditoren (inkl. Banken)	0,00

Abb. E-2: Wichtige Einflussgrößen in der Kostenstelle Einkauf²⁾

Variable	Wichtigkeit
Anzahl Lagerbewegungen	2,33
Anzahl verkaufter Produktarten	0,00

Abb. E-3: Wichtige Einflussgrößen in der Kostenstelle Hochregallager³⁾

¹⁾ Siehe Kapitel D.II, S. 163, sowie Anhang A1 und A2.

²⁾ Berechnet mit den Daten aus Anhang A4; Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

³⁾ Berechnet mit den Daten aus Anhang A5; Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

Variable	Wichtigkeit
Fremdleistung Produktion in €	7,39
Anzahl verkaufter Produktarten	4,66
geleistete Std. Spritzgussmaschinen	4,55
Rohmaterialverbrauch in t	0,00
Anzahl eingesetzter Rohmaterialy...	0,00
Umsatz mit eigener Produktion in €	0,00

Abb. E-4: Wichtige Einflussgrößen in der Kostenstelle Verfahrenstechnik¹⁾

Variable	Wichtigkeit
Anzahl Lagerbewegungen	7,76
Anzahl verkaufter Produktarten	3,49
Umsatz mit Handelsware in €	0,00
Umsatz mit eigener Produktion in €	0,00
Anzahl Rechnungen (ohne Einkauf)	0,00
Anzahl Warenempfänger	0,00

Abb. E-5: Wichtige Einflussgrößen in der Kostenstelle Auftragsbereitstellung²⁾

Die dargestellten Rangfolgen erscheinen durchweg plausibel und sind vielversprechende Anknüpfungspunkte für weitere Analysen. Weiterführende Experimente mit nicht zur Berechnung der Wichtigkeitswerte verwendeten Testdaten haben gezeigt, dass die ermittelten Werte generalisierbar sind und dass die Prognosegenauigkeit für unbekannte Daten deutlich abnimmt, wenn Variablen mit einem Wichtigkeitswert größer 2 nicht im Modell enthalten sind.

Da die Wichtigkeitswerte noch keine Informationen über die Form des Zusammenhangs enthalten, ist es sinnvoll, sich im nächsten Schritt die Funktionsgraphen der wichtigsten Einflussgrößen anzusehen, um einen Ein-

¹⁾ Berechnet mit den Daten aus Anhang A6; Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

²⁾ Berechnet mit den Daten aus Anhang A7; Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

druck von der Stärke und Richtung des jeweiligen Zusammenhangs zu bekommen.

2. Graphische Darstellung mehrdimensionaler nichtlinearer Kostenfunktionen

Die graphische Darstellung einer mit einem neuronalen Netz geschätzten Kostenfunktion wirft zwei Probleme auf: erstens muss die mehrdimensionale Kostenfunktion auf zwei oder drei Dimensionen reduziert werden und zweitens muss das „Wissen“ um den funktionalen Zusammenhang aus dem neuronalen Netz, dessen kompliziertes Innenleben eine Analyse erschwert, extrahiert werden (*knowledge extraction*).

Beide Probleme lassen sich durch eine diskrete Sensitivitätsanalyse lösen. Dabei wird, um den Verlauf der Kostenhöhe K in Abhängigkeit von einer Einflussgröße x_i zu ermitteln, die Einflussgröße x_i im Intervall $[x_i^{\min}; x_i^{\max}]$ in kleinen Schritten erhöht, während die anderen Einflussgrößen auf einen konstanten Wert \bar{x}_j ($j \neq i$) festgelegt sind. In jedem Schritt wird die resultierende Kostenhöhe $K = f(\bar{x}_1, \dots, x_i, \dots, \bar{x}_I)$ erfasst.

Der Grundgedanke ist hier, wie in Abschnitt 1, nicht die Parameter (die Verbindungsgewichte) des neuronalen Netzes zu analysieren, sondern das neuronale Netz als *black box* zu behandeln, diese aber dadurch zu erhellen, dass man die Reaktionen der Netzausgabe auf systematische Inputvariationen beobachtet.¹⁾

In Cenobi werden die gerade nicht betrachteten Einflussgrößen \bar{x}_j auf ihren Medianwert festgesetzt und die jeweilige Kostenhöhe durch die in Kapitel D.XIII beschriebene Ensembleschätzung bestimmt. Abb. E-6 zeigt die auf

¹⁾ Vgl. Livingstone u. a. [Extraction of Information] 233, mit weiteren Nachweisen.

diese Weise ermittelten Kostenverläufe für die einzelnen Kostenstellen des in Kapitel D bereits ausgiebig betrachteten Datensatzes in Abhängigkeit von der Zuhaltekraft. Es fällt auf, dass die Beobachtungen zwar stark streuen, eine gewisse steigende Tendenz aber erkennbar ist. Der treppenförmige Verlauf der Funktionen legt nahe, dass der Zusammenhang hauptsächlich durch gleichbleibende Abschreibungen zustande kommt (siehe S. 304).

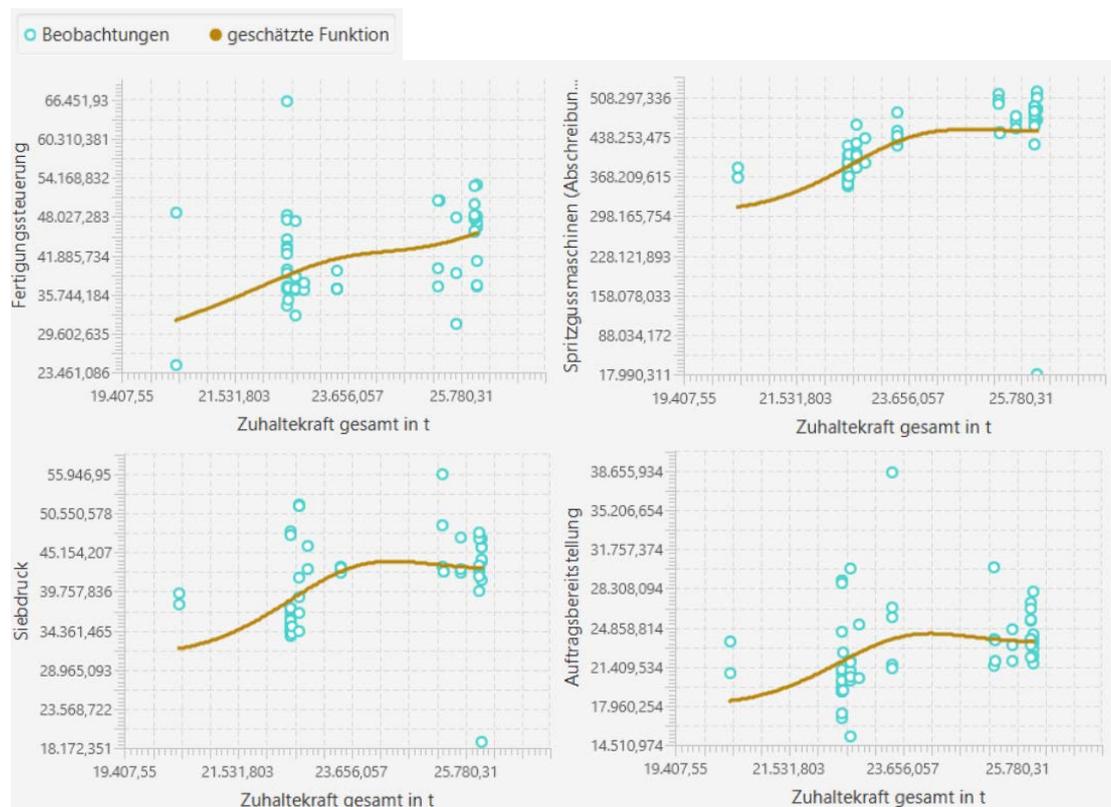


Abb. E-6: Kostenverläufe im Fertigungsbereich in Abhängigkeit von der Zuhaltekraft¹⁾

Bei den Kostenstellen Spritzgussmaschinen und Siebdruck fallen zwei Ausreißer mit sehr niedriger Kostenhöhe ins Auge. Ohne sie wird der Zusammenhang noch deutlicher (siehe Abb. E-7). Allerdings ist wegen der geringen Variation der Zuhaltekraft fraglich, ob der gefundene Zusammenhang

¹⁾ Berechnet mit den Daten aus Anhang A3; Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

verallgemeinerbar ist. Eine verlässliche Prognose der Kostenhöhe für eine Erhöhung der Zuhaltekraft z. B. auf 30.000 t scheint kaum möglich – ganz abgesehen von den oben auf S. 304 angesprochenen Interpretationsproblemen.

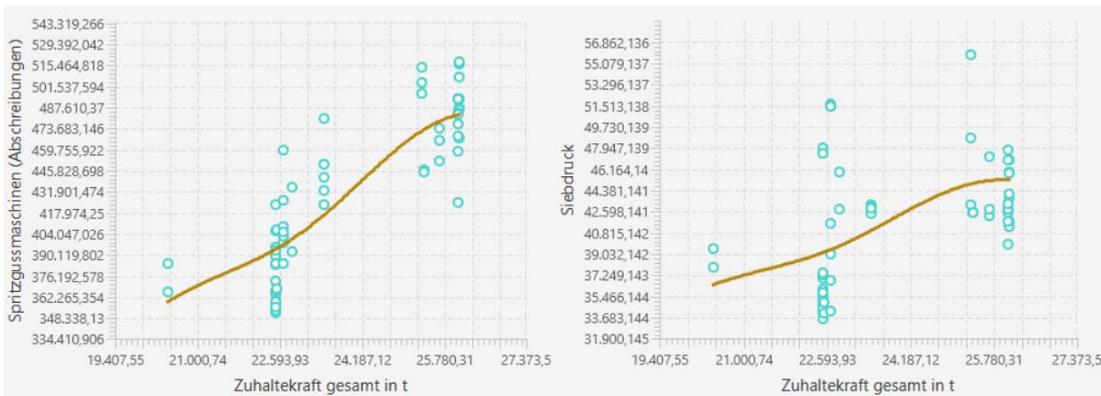


Abb. E-7: Kostenverläufe im Fertigungsbereich nach Ausreißerbeseitigung

Wendet man sich nun der zweitwichtigsten Einflussgröße (siehe Abb. E-1) – den geleisteten Spritzgussstunden – zu, kann man den deutlich niedrigeren Wichtigkeitswert dieser Einflussgröße an den Streudiagrammen in Abb. E-8 und Abb. E-9 gut nachvollziehen. Ein aussagekräftiger Zusammenhang ist nicht erkennbar.

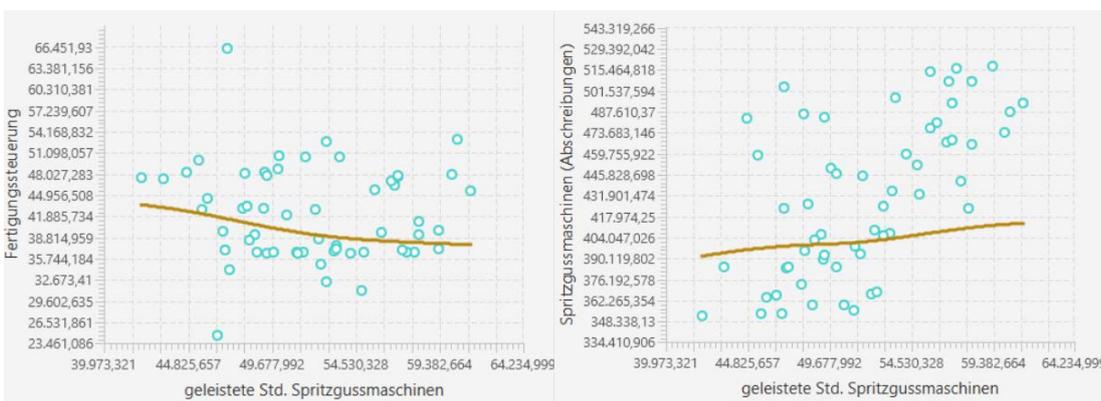


Abb. E-8: Kostenverläufe im Fertigungsbereich in Abhängigkeit von den geleisteten Spritzgussstunden¹⁾

¹⁾ Berechnet mit dem Datensatz aus Anhang A3; Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

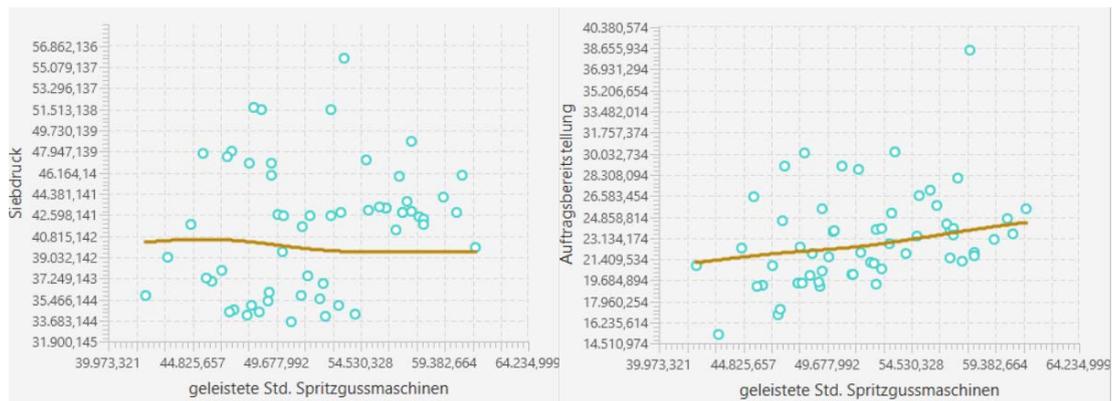


Abb. E-9: Kostenverläufe im Fertigungsbereich in Abhängigkeit von den geleisteten Spritzgussstunden (Fortsetzung)

Ein anderes Bild ergibt sich für die Einkaufskostenstelle. Insbesondere ist der positive Zusammenhang zwischen der Anzahl bearbeiteter Rechnungen und der Kostenhöhe klar zu erkennen (siehe Abb. E-10).

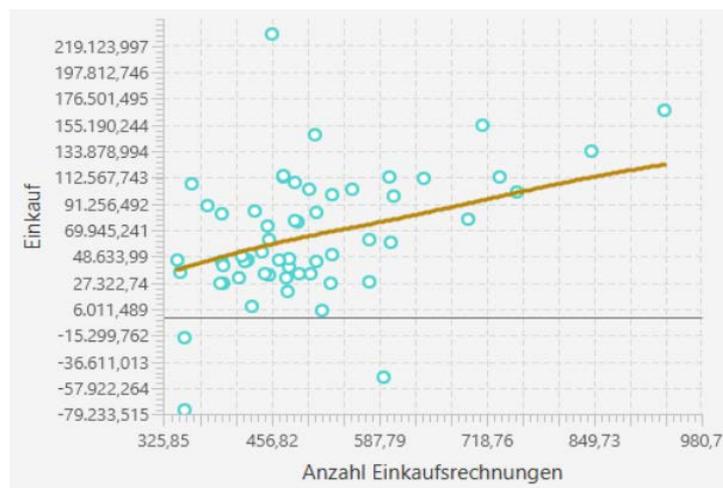


Abb. E-10: Kostenverlauf im Einkauf in Abhängigkeit von der Anzahl bearbeiteter Einkaufsrechnungen¹⁾

Der negative Einfluss des Handelswarenumsatzes auf die Kostenhöhe der Einkaufskostenstelle (siehe Abb. E-11) ist hingegen weniger selbstverständlich. Die Erklärung, dass ein höherer Handelswarenumsatz mit einem geringeren eigenen Produktionsvolumen einhergeht, trifft nicht zu, da beide

¹⁾ Berechnet mit den Daten aus Anhang A4; Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

Umsätze schwach positiv korrelieren. Der Zusammenhang scheint aber nicht zufällig zu sein – schließlich verbessert sich die Prognosegüte deutlich, wenn diese Einflussgröße im Modell enthalten ist (siehe Abb. E–2).

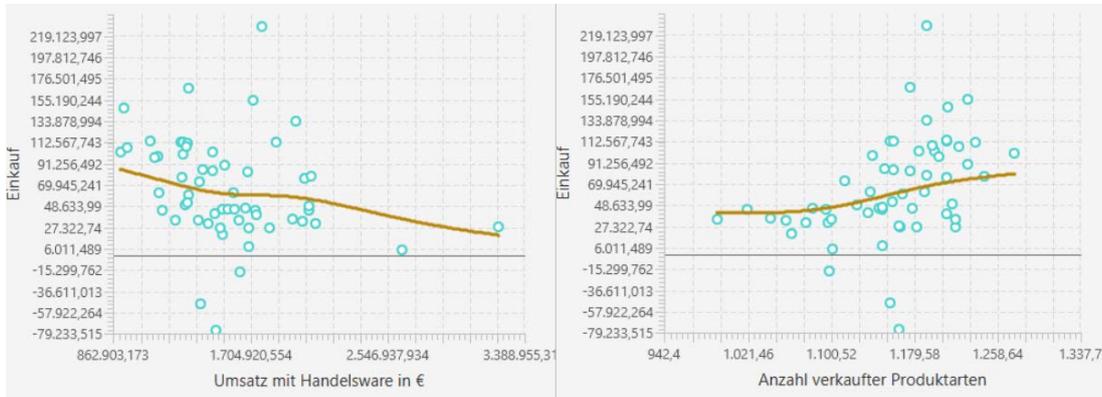


Abb. E–11: Kostenverläufe im Einkauf in Abhängigkeit von Handelswarenumsatz und Anzahl verkaufter Produktarten¹⁾

Der Kostenverlauf im Hochregallager (siehe Abb. E–12) deutet auf abnehmende Grenzkosten der Lagerbewegungen hin. Das universell anpassungsfähige neuronale Netz kann diesen Verlauf im Gegensatz zu einem linearen Modell ohne weiteres nachzeichnen und somit auch bei anschließenden Entscheidungsrechnungen adäquat berücksichtigen.

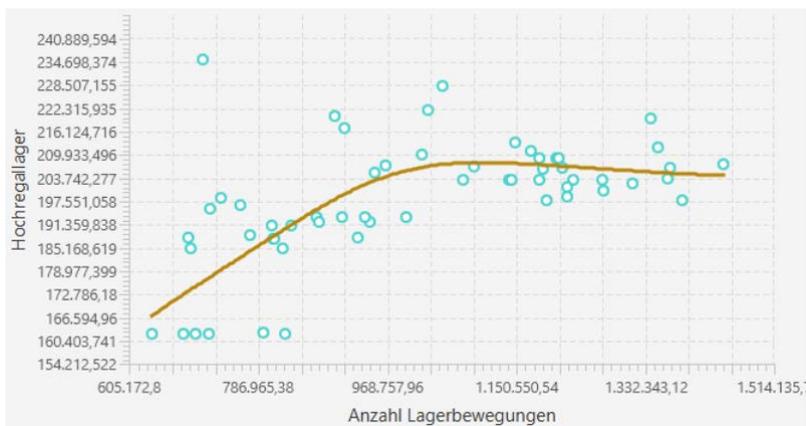


Abb. E–12: Kostenverlauf im Hochregallager²⁾

¹⁾ Berechnet mit den Daten aus Anhang A4; Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D–50, S. 277.

²⁾ Berechnet mit den Daten aus Anhang A5; Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D–50, S. 277.

Die Interpretation der Kostenverläufe in der Verfahrenstechnik (siehe Abb. E-13) wird dadurch erschwert, dass es einen Strukturbruch in den Daten gibt. Es besteht zwar bei allen drei relevanten Einflussgrößen ein sichtbarer Zusammenhang; gerade bei der Produktionsfremdleistung könnte es aber sein, dass für die Kostenreduktion nicht die Erhöhung der Fremdleistung an sich, sondern z. B. eine Umstrukturierungsmaßnahme verantwortlich ist, bei der die eigene Verfahrenstechnik reduziert und als Fremdleistung ausgelagert wurde. Wenn dem so ist, lässt sich an der Kostenfunktion aber immerhin ablesen, dass die Auslagerung tatsächlich zu einer Kostenreduktion in der Verfahrenstechnik geführt hat – wenn auch nicht in der Höhe, in der die Fremdleistungskosten gestiegen sind.

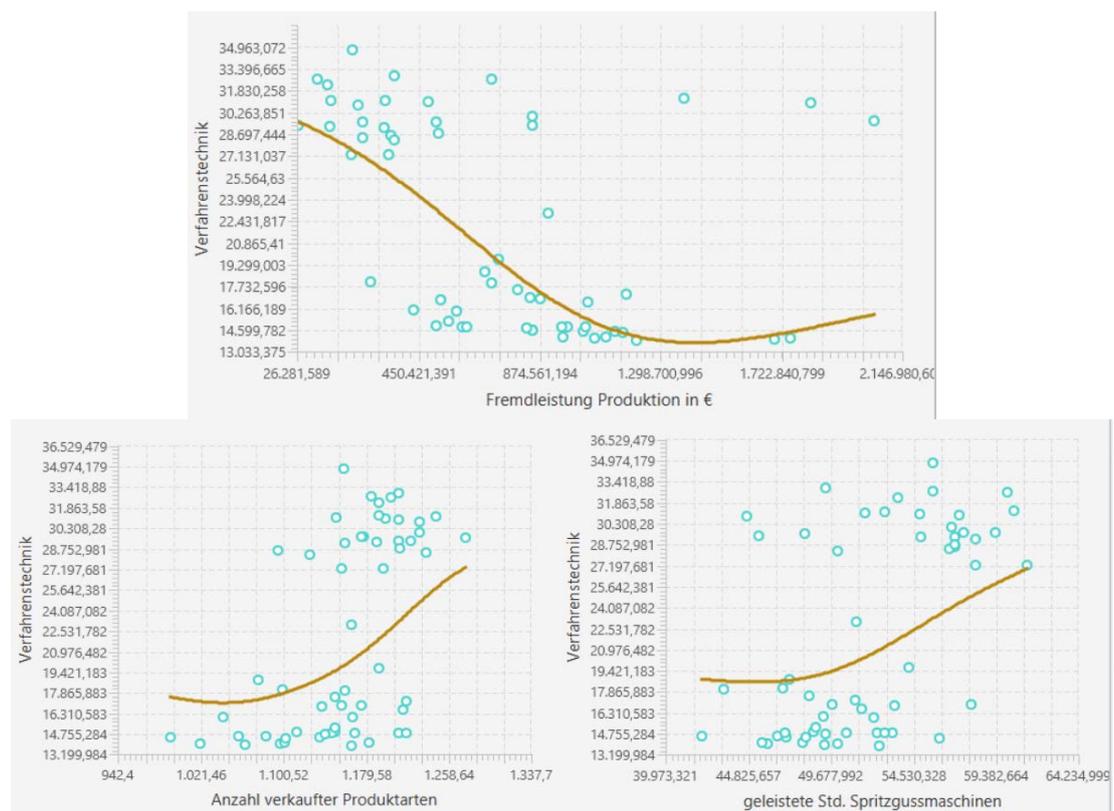


Abb. E-13: Kostenverläufe in der Kostenstelle Verfahrenstechnik¹⁾

¹⁾ Berechnet mit den Daten aus Anhang A6; Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

Die Auftragsbereitstellung hat einen ähnlichen Kostenverlauf in Abhängigkeit von der Anzahl der Lagerbewegungen (siehe Abb. E-14) wie das Hochregallager. Der Sättigungseffekt schlägt hier sogar in einen negativen Zusammenhang um – was nicht unbedingt plausibel erscheint. Ein Zusammenhang der Auftragsbereitstellungskosten mit der Anzahl der Produktarten ist trotz des relativ hohen Wichtigkeitswertes dieser Einflussgröße (siehe Abb. E-5) nicht wirklich erkennbar.

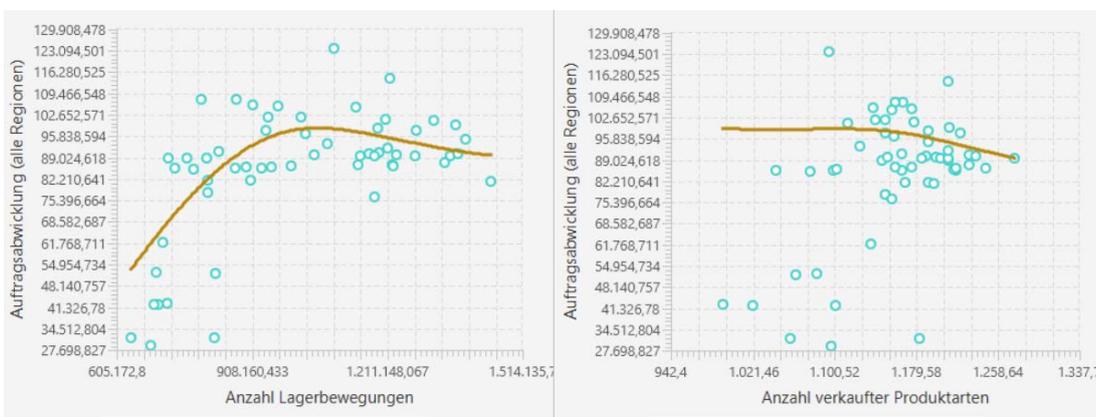


Abb. E-14: Kostenverläufe in der Auftragsbereitstellung¹⁾

Die betrachteten Funktions- und Streudiagramme zeigen, dass die visuelle Analyse der Kostenfunktionen ein wichtiger Schritt bei der Auswahl relevanter Einflussgrößen ist. Der genetische Algorithmus sortiert zwar einige irrelevante Inputvariable aus; kostenrechnerisch unplausible und eher zufällige Zusammenhänge lassen sich aber teilweise erst durch die visuelle Prüfung eines mit dem Leistungsprozess vertrauten Fachmanns erkennen.

Zudem fördert die graphische Analyse der Kostenverläufe das Verständnis der betrieblichen Kostenstruktur. Ein Nachteil der zweidimensionalen Aufbereitung ist allerdings, dass Interdependenzen unter den unabhängigen

¹⁾ Berechnet mit den Daten aus Anhang A7; Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

Variablen nicht dargestellt werden können.¹⁾ Die Erweiterung der Diagramme um eine Tiefendimension oder die Darstellung der Kostenverläufe für unterschiedliche Niveaus \bar{x}_j könnten hier etwas Abhilfe schaffen, würden aber die Darstellung schwerer lesbar machen.

II. Kostenprognosen mit künstlichen neuronalen Netzen

Die Prognose zukünftiger Kosten ist die Grundlage für alle weiteren Anwendungsfälle betrieblicher Kostenfunktionen im naturgemäß zukunftsgerichteten Führungsprozess.²⁾ Dementsprechend ist eine Kostenfunktion umso besser für diese Anwendungsfälle geeignet, je genauer ihre Prognosen sind.³⁾ Künstliche neuronale Netze ermöglichen oft bessere Prognosen als einfachere Schätzverfahren wie z. B. lineare Regressionsmodelle, weil sie sich an beliebige Zusammenhänge anpassen können.⁴⁾ Das lässt sich auch an dem vorliegenden Beispieldatensatz mit realen Kostenrechnungsdaten belegen (s. u.).

Für eine Prognose werden konkrete Ausprägungen der unabhängigen Variablen als Antezedens benötigt.⁵⁾ In der Kostenrechnung sind das üblicherweise die geplanten Einflussgrößenmengen für den Zeitraum, in dem die Kostenhöhe prognostiziert werden soll (siehe Abb. E-15).

¹⁾ Siehe Abschnitt C.III.4.a, S. 147.

²⁾ Siehe Abschnitt B.II.1, S. 11 ff., sowie Kapitel B.IV, S. 55 ff.

³⁾ Siehe Abschnitt D.X.1, S. 253 ff.

⁴⁾ Siehe Abschnitt C.III.4, S. 147 ff.

⁵⁾ Siehe Abb. B-10, S. 61.

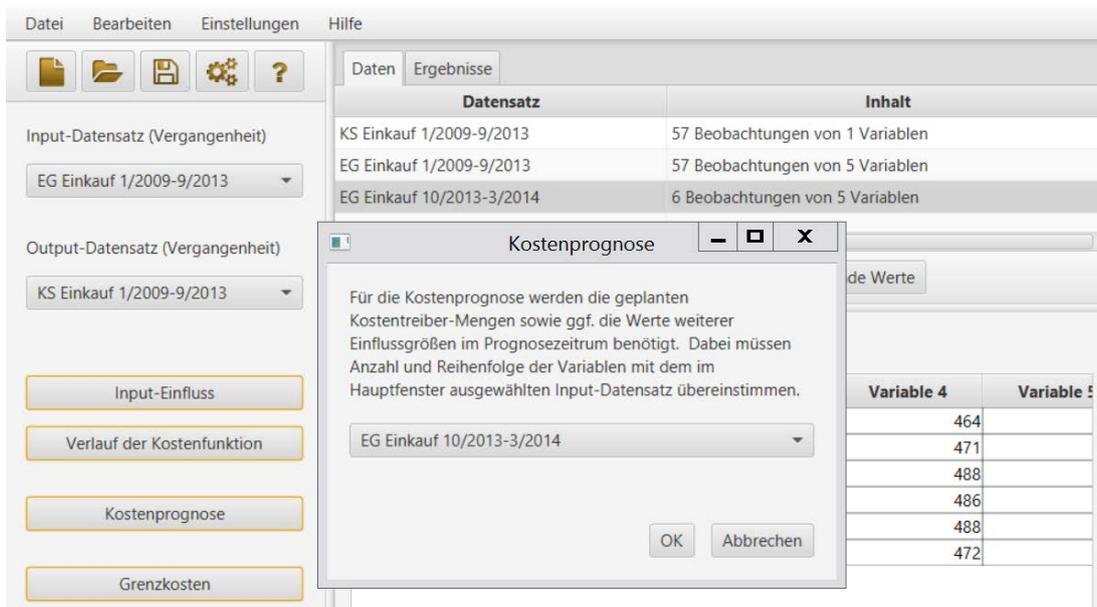


Abb. E–15: Kostenprognose in Cenobi

Die Prognosegenauigkeit des implementierten Neuronale-Netze-Ensembles wurde mit den Datensätzen aus Anhang A4 bis A7 getestet. Das Ensemble wurde jeweils mit den Daten der ersten 57 Monate parametrisiert und prognostizierte die Kostenhöhe in den verbleibenden Monaten. Abb. E–16 zeigt die daraus berechneten normierten mittleren absoluten Fehler sowie die symmetrischen mittleren absoluten prozentualen Fehler über die Testmenge.¹⁾ Zum Vergleich wurde ein lineares Regressionsmodell $\widehat{K} = b_0 + b_1x_1 + \dots + b_Ix_I$ mit denselben Daten parametrisiert und getestet. Das Neuronale-Netze-Ensemble hatte bei allen ausgewählten Kostenstellen einen geringeren Prognosefehler.

¹⁾ Zur Berechnung der Fehlermaße siehe (D.12) und (D.13), S. 253.

Kostenstelle (Datensatz)	Neuronale-Netze-Ensemble ¹⁾		lineare Regression ²⁾	
	NMAE	MAPE _S	NMAE	MAPE _S
Einkauf (A4)	0,041	0,157	0,051	0,190
Hochregallager (A5)	0,152	0,241	0,163	0,251
Verfahrenstechnik (A6)	0,064	0,044	0,121	0,081
Auftragsabwicklung (A7)	0,042	0,063	0,055	0,082

Abb. E-16: Prognosegenauigkeit des Neuronale-Netze-Ensembles im Vergleich zur linearen Regression für Out-of-Sample-Testdaten ausgewählter Kostenstellen des Beispieldatensatzes

Zwar gibt es im Beispieldatensatz vereinzelt auch Kostenstellen, deren Kostenhöhe das lineare Modell etwas besser prognostizieren konnte. In diesen Fällen ist aber der Prognosefehler insgesamt schon so hoch, dass davon auszugehen ist, dass eigentlich kein Zusammenhang zwischen den zur Verfügung stehenden Einflussgrößen und der gemessenen Kostenhöhe besteht – die bessere Prognose des linearen Modells also eher ein Glückstreffer ist.

¹⁾ Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

²⁾ Für die geschätzten linearen Regressionsmodelle siehe Anhang A8.

III. Alternativenbewertung mit künstlichen neuronalen Netzen

1. Berechnung der Grenzkosten einer Entscheidung

Eine entscheidungsorientierte Bewertungsrechnung soll zeigen, wie sich die (prognostizierten) Kosten ändern, wenn eine bestimmte Handlungsalternative statt einer anderen realisiert wird. Dies sind die Grenzkosten der Entscheidung.¹⁾ In Abschnitt B.III.3 (S. 29 ff.) wurde gezeigt, dass für einen Entscheidungsträger in erster Linie die absolute Kostenänderung, die infolge einer bestimmten Entscheidung eintritt, relevant ist.²⁾ Diese *Grenzkosten im weiteren Sinn* sind die *Differenzkosten* (Schichtkosten, *incremental costs*)

$$DK = f(\mathbf{x}^{(0)} + \Delta\mathbf{x}) - f(\mathbf{x}^{(0)})$$

einer Änderung der unabhängigen Variablen um $\Delta\mathbf{x}$, ausgehend vom Punkt $\mathbf{x}^{(0)} = (x_1^{(0)}, \dots, x_I^{(0)})^T$ (siehe (B.8), S. 29).³⁾ Sie lassen sich problemlos mit einem trainierten neuronalen Netz berechnen.

Die Differenzkosten DK werden in Cenobi berechnet, indem die Ensembleschätzung für die Nullalternative $\mathbf{x}^{(0)}$ von der Ensembleschätzung für das geänderte Einflussgrößenniveau $(\mathbf{x}^{(0)} + \Delta\mathbf{x})$ subtrahiert wird. Abb. E-17 zeigt den Ablauf dieser grundlegenden Grenzkostenrechnung beispielhaft für die Einkaufskostenstelle des Beispieldatensatzes. Es werden die Grenzkosten einer Managementmaßnahme berechnet, die innerhalb von drei

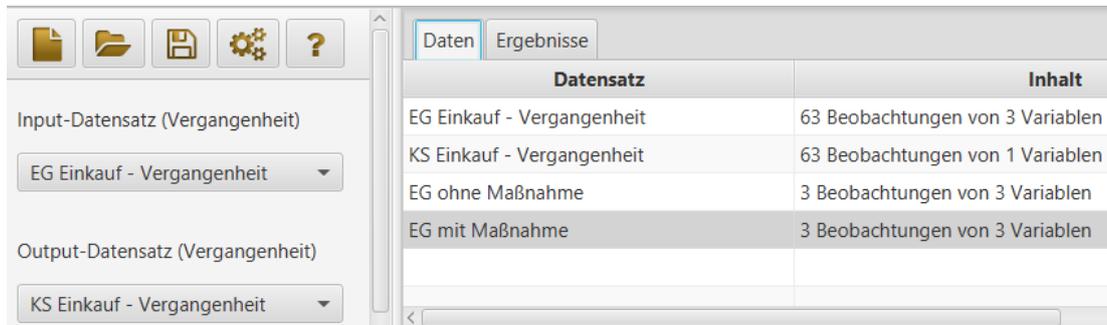
¹⁾ Vgl. Loitlsberger [Grenzbetrachtungen]; Lücke [Grenzdenken].

²⁾ Vgl. auch Adam [Grenzkostenrechnung] 824 f.; Hummel [Relevante Kosten]; Männel [Grenz- und Residualkosten] 819; Horngren/Datar/Rajan [Cost Accounting] 415 f.

³⁾ Vgl. Mellerowicz [Theorie der Kosten] 356 f.; Demski [Managerial Uses] 16 f.

Monaten die Anzahl der verkauften Produktarten und die Anzahl der bearbeiteten Einkaufsrechnungen schrittweise reduzieren soll (a).

(a) Datensätze



Datensatz	Inhalt
EG Einkauf - Vergangenheit	63 Beobachtungen von 3 Variablen
KS Einkauf - Vergangenheit	63 Beobachtungen von 1 Variablen
EG ohne Maßnahme	3 Beobachtungen von 3 Variablen
EG mit Maßnahme	3 Beobachtungen von 3 Variablen



Datensatz EG ohne Maßnahme

	Anzahl Einkaufsrechnungen	Umsatz mit Handelsware in €	Anzahl verkaufter Produktarten
1	580	1.400.000	1.250
2	580	1.400.000	1.250
3	580	1.400.000	1.250



Datensatz EG mit Maßnahme

	Anzahl Einkaufsrechnungen	Umsatz mit Handelsware in €	Anzahl verkaufter Produktart...
1	580	1.400.000	1.250
2	530	1.400.000	1.100
3	480	1.400.000	1.000

(b) Auswahldialog

Nullalternative

geänderte Einflussgrößen

Grenzkosten = Kosten bei geänderten Einflussgrößen
- Kosten der Nullalternative

(c) Ergebnis

Grenzkosten für Änderung von EG ohne Maßnahme zu EG mit Maßnahme, ...

	Einkauf
1	0
2	-40.690,4685093131
3	-50.406,75896471025

Abb. E-17: Grenzkostenrechnung in Cenobi¹⁾

Die Nullalternative ist in diesem Fall die erwartete Einflussgrößenmenge, wenn die Maßnahme nicht umgesetzt wird; die geänderten Einflussgrößen ergeben sich entsprechend aus der erwarteten Einflussgrößenreduktion in Folge der Maßnahme (b). Im Ergebnis kann die Maßnahme, wenn sie erfolgreich umgesetzt wird, die Kosten der Einkaufskostenstelle um rund 40.000 € im zweiten Monat und 50.000 € im dritten Monat reduzieren (c) – immer vorausgesetzt, dass der aus den Vergangenheitsdaten geschätzte Zusammenhang weiterhin gilt.

2. Kalkulation mit exakten Grenzkosten

Eine Kalkulation verrechnet anfallende Kosten auf einzelne Kostenträger. Ein Kostenträger kann eine Leistungseinheit, z. B. ein Produktstück, oder eine Menge von Leistungen, z. B. ein Fertigungslos oder ein Auftrag, sein. Neben den direkt zurechenbaren Einzelkosten werden einem Kostenträger indirekt über die mit ihm zusammenhängenden Einflussgrößen auch Teile der Kostenstellenkosten zugerechnet. Eine entscheidungsorientierte Kalkulation folgt dabei dem Identitätsprinzip; demnach können Kosten und Leis-

¹⁾ Als Vergangenheitswerte dienten die Daten aus Anhang A4. Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

tungen gerade dann einander zugerechnet werden, wenn sie auf eine identische Entscheidung zurückgehen.¹⁾ Die Kalkulation wird damit zu einer speziellen Grenzkostenrechnung.²⁾

Wie die Grenzkostenrechnung im Allgemeinen (s. o.) ist auch eine Kalkulation mit künstlichen neuronalen Netzen ohne weiteres möglich. Die einzige Besonderheit besteht darin, dass die mit künstlichen neuronalen Netzen geschätzten nichtlinearen Kostenfunktionen nicht an jeder Stelle dieselbe Steigung haben. Deshalb muss ein Ausgangsniveau $x^{(0)}$ der Einflussgrößen benannt werden, an dem die Kalkulation ansetzt (siehe Abb. E-18).

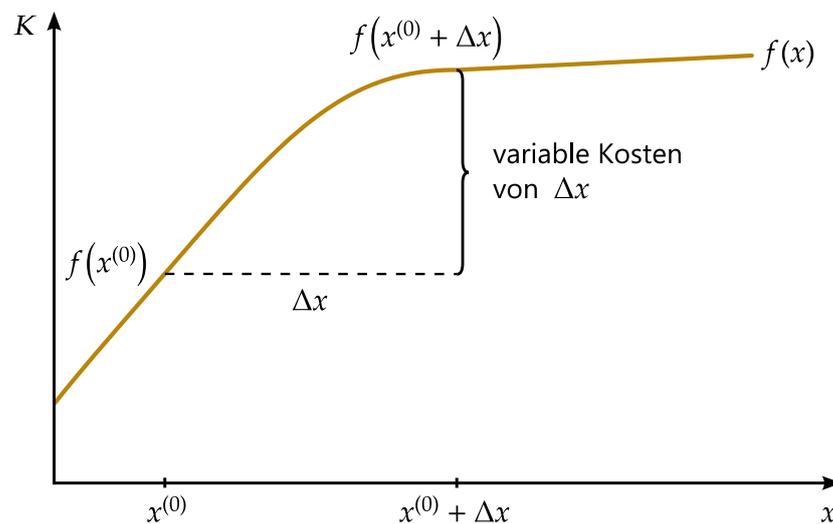


Abb. E-18: Kalkulation mit Differenzkosten

Abb. E-19 zeigt beispielhaft die Kalkulation der variablen internen Logistikkosten eines Auftrags, der 420 Lagerbewegungen zusätzlich zu den bereits disponierten 1.000.000 Lagerbewegungen erfordert (a). Die Anzahl der verkauften Produktarten ändert sich durch den Auftrag nicht. Die variablen Logistikkosten des Auftrags werden, wie in Abb. E-18 dargestellt, als Differenzkosten berechnet (b).

¹⁾ Vgl. Riebel [Einzelkostenrechnung] 76, 418 ff., 627.

²⁾ Siehe Abschnitt B.IV.5, S. 62 ff.

(a) Datensätze

Datensatz	Inhalt
EG Logistik - Vergangenheit	63 Beobachtungen von 2 Variablen
KS Hochregallager/Auftragsabwick...	63 Beobachtungen von 2 Variablen
EG Ausgangsniveau	1 Beobachtungen von 2 Variablen
EG von Auftrag beansprucht	1 Beobachtungen von 2 Variablen



Datensatz EG Ausgangsniveau

	Anzahl Lagerbewegungen	Anzahl verkaufter Produktarten
1	1.000.000	1.250



Datensatz EG von Auftrag beansprucht

	Anzahl Lagerbewegungen	Anzahl verkaufter Produktarten
1	420	0

(b) Auswahldialog

$x^{(0)}$ Ausgangsniveau (für diese Periode bereits disponierte Einflussgrößenmengen) EG Ausgangsniveau

Δx vom Kostenträger zusätzlich beanspruchte Einflussgrößenmengen EG von Auftrag beansprucht

Es wird mit Differenzkosten kalkuliert:

$$\text{variable Kosten} = K^{\text{inkremental}} - K^{\text{Ausgangsniveau}} = f(x^{(0)} + \Delta x) - f(x^{(0)})$$

(c) Ergebnis

Variable Kosten für EG von Auftrag beansprucht,...

	Hochregallager	Auftragsabwicklung ...	Summe
1	24,905698420741...	8,934184121186263	33,8398...

Abb. E-19: Kalkulation in Cenobi

Im Ergebnis können dem Auftrag variable Kosten in Höhe von 24,91 € im Hochregallager und 8,93 € in der Auftragsabwicklung, in Summe also Logistikkosten in Höhe von 33,84 € zugerechnet werden (c).¹⁾

3. Berechnung von Kostensätzen für Bezugsgrößenkalkulation und Periodenerfolgsrechnung

Mit den bisher erörterten Auswertungsverfahren lassen sich im Grunde alle in Kapitel B.IV (S. 55 ff.) betrachteten Anwendungsfälle betrieblicher Kostenfunktionen erfüllen. Da aber der Umgang mit nichtlinearen Kostenfunktionen für den ein oder anderen Kostenrechner etwas ungewohnt sein wird, besteht möglicherweise der Bedarf, Kostensätze je Einflussgrößeneinheit auszurechnen, um sie beispielsweise in eine klassische Bezugsgrößenkalkulation oder eine Periodenerfolgsrechnung (z. B. eine mehrstufige Deckungsbeitragsrechnung) einzubringen.²⁾

In einer entscheidungsorientierten Rechnung entspricht der Kostensatz einer Einflussgrößeneinheit den Grenzkosten, die entstehen, wenn die Einflussgröße um eine Mengeneinheit zunimmt. In Kapitel B wurden solche Kostensätze als variable Kosten k^v einer Einflussgrößeneinheit bezeichnet.³⁾

Bei linearen Kostenfunktionen entspricht k^v der Geradensteigung und ist somit konstant. Bei nichtlinearen Kostenfunktionen ist beides nicht der Fall. Zur Berechnung des variablen Kostensatzes k_i^v einer Einflussgröße x_i muss hier ein Ausgangsniveau $x^{(0)}$ bekannt sein, von dem aus die Differenzkosten

¹⁾ Als Vergangenheitswerte dienten die Daten aus Anhang A5 und A7. Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

²⁾ Siehe Abschnitt B.III.4, S. 31 ff.

³⁾ Siehe Abschnitt B.III.1, S. 20 ff.

$$k_i^y = f \left(\mathbf{x}^{(0)} + \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \right) - f(\mathbf{x}^{(0)}) \quad (\text{E.1})$$

einer Erhöhung von x_i um eine Mengeneinheit berechnet werden. Abb. E-20 illustriert dieses Vorgehen für eine einzelne Einflussgröße x .

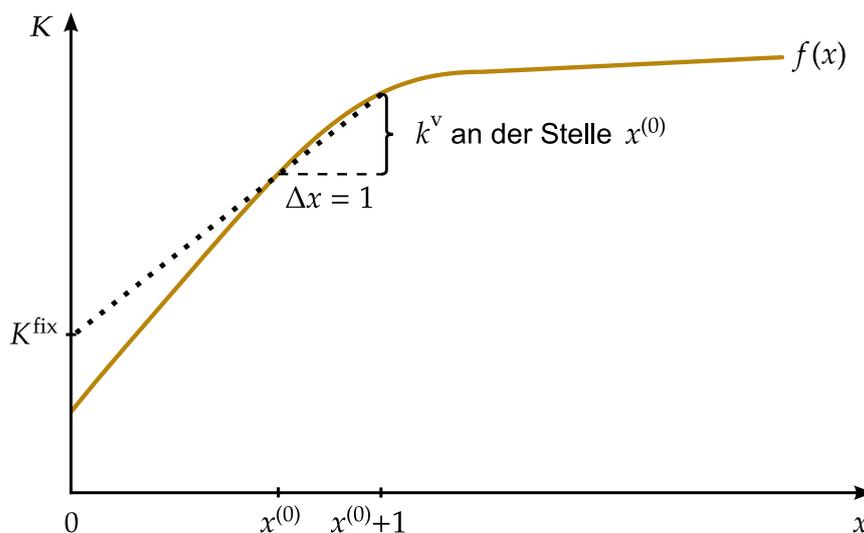


Abb. E-20: Berechnung eines Grenzkostensatzes

Abb. E-21 zeigt beispielhaft den Ablauf einer Berechnung von Kostensätzen nach (E.1) in Cenobi.¹⁾ Es werden die bereits aus Abb. E-19 bekannten Einflussgrößen und Kostenstellen verwendet (a) und auch das Ausgangsniveau der Einflussgrößen wird von dort übernommen (b). Die berechneten Kostensätze (c) können – wie in Abb. E-22 angedeutet – in eine typische Bezugsgrößenkalkulation übernommen werden. Die in Abb. E-22 errechneten Auftragskosten stimmen nicht exakt mit denen aus Abb. E-19 überein, weil bei der Bezugsgrößenkalkulation der nichtlineare Verlauf der Kostenfunktion vernachlässigt wird. Die Berechnung in Abb. E-19 ist also genauer.

¹⁾ Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

(a) Datensätze

siehe Abb. E-19 a

(b) Auswahldialog

Der Kostensatz einer Einflussgröße entspricht den Grenzkosten, die entstehen, wenn die Einflussgröße um eine Mengeneinheit zunimmt. Da die Grenzkosten i. d. R. nicht konstant sind, sollte für alle Einflussgrößen ein Ausgangsniveau angegeben werden, von dem aus der Kostensatz berechnet wird.

Ausgangsniveau (für diese Periode bereits disponierte Einflussgrößenmengen)

EG Ausgangsniveau

Wenn kein Ausgangsniveau angegeben ist, werden durchschnittliche Kostensätze berechnet.

OK

Abbrechen

(c) Ergebnis

Grenzkostensätze für EG Ausgangsniveau, berechnet mit (EG Logistik - Vergangenheit, KS ...

		Hochregall...	Auftragsabw...	Summe
1	variable Kosten je Mengeneinheit Anzahl Lagerbewegungen	0,0594	0,0214	0,0808
2	variable Kosten je Mengeneinheit Anzahl verkaufter Produ...	7,3306	8,9269	16,2575
3	fixe Kosten	136.493,3553	63.638,5616	200.131,91..

Abb. E-21: Berechnung von Kostensätzen in Cenobi

Kostenposition	Kostensatz	Menge	Gesamt
Materialeinzelkosten
⋮	⋮	⋮	⋮
beanspruchte Lagerbewegungen	0,0808 €	420	33,94 €
⋮	⋮	⋮	⋮

Abb. E-22: Bezugsgrößenkalkulation für einen Auftrag

In Cenobi werden zusammen mit den variablen Kostensätzen auch die Fixkosten der Kostenstellen angegeben (siehe Abb. E-21 c). Wie in Abb. E-20 gezeigt, sind die fixen Kosten K^{fix} das Absolutglied der linearen Funktion $K = K^{\text{fix}} + k^v x$ (siehe (B.7), S. 25) und erfüllen die Gleichung

$$K^{\text{fix}} + k^v x^{(0)} \stackrel{!}{=} f(x^{(0)}) \quad (\text{E.2})$$

wobei $f(x^{(0)})$ die Neuronale-Netze-Schätzung der Kostenhöhe für das Einflussgrößenniveau $x^{(0)}$ ist. Die in Abb. E-21 c angegebenen variablen und fixen Kostensätze sind folglich eine lineare Annäherung an die nichtlineare Neuronale-Netze-Schätzung. Diese Vereinfachung kann z. B. für die Ableitung von Sollkostenfunktionen zur Budgetierung¹⁾ oder für eine mehrstufige Deckungsbeitragsrechnung²⁾ hilfreich sein. Es sollte allerdings beachtet werden, dass die lineare Funktion $K = K^{\text{fix}} + k^v x$ den tatsächlichen Kostenverlauf nur in der Nähe von $x^{(0)}$ gut repräsentiert. Bei veränderten Einflussgrößenniveaus wird eine Neuberechnung der Kostensätze notwendig.

Falls das nicht möglich oder zu unbequem ist, bietet Cenobi die Berechnung durchschnittlicher Kostensätze \bar{k}_i^y an. Sie entsprechen der Steigung einer Geraden zwischen der geschätzten Kostenhöhe bei minimaler und bei maximaler Einflussgrößenmenge, während die übrigen Einflussgrößen x_j ($j \neq i$) ihren Medianwert annehmen:

$$\bar{k}_i^y = \frac{f(\bar{x}_1, \dots, x_i^{\text{max}}, \dots, \bar{x}_I) - f(\bar{x}_1, \dots, x_i^{\text{min}}, \dots, \bar{x}_I)}{x_i^{\text{max}} - x_i^{\text{min}}} \quad (\text{E.3})$$

mit \bar{x}_j als Median von x_j .

¹⁾ Siehe Abb. B-11, S. 67.

²⁾ Vgl. Schweitzer/Küpper [Systeme] 467 ff.

Die fixen Kosten werden wie in (E.2) berechnet. Abb. E-23 illustriert dieses Vorgehen für eine einzelne Einflussgröße x .

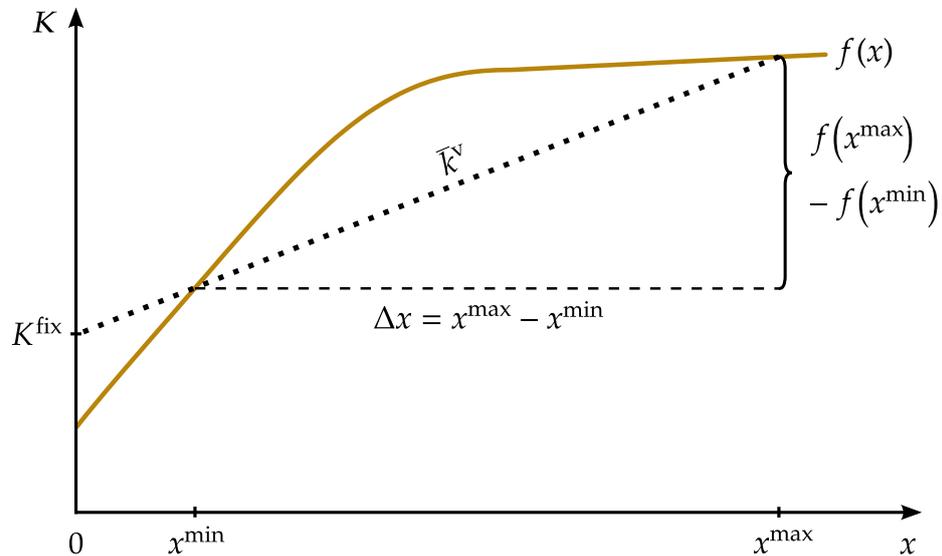


Abb. E-23: Berechnung eines mittleren Kostensatzes

Dabei wird deutlich, dass $K = K^{\text{fix}} + \bar{K}^v x$ stark von der eigentlich geschätzten Kostenfunktion f abweichen kann, und jene nur in den Extremwerten x^{min} und x^{max} trifft. Die Verwendung von Grenzkostensätzen nach (E.1) mit einem realistisch gewählten Ausgangsniveau ist deshalb i. A. die bessere Lösung.

IV. Kumulative Abweichungsanalysen mit künstlichen neuronalen Netzen

Abweichungsanalysen in der Kostenrechnung basieren in der Regel auf diskreten Auswertungen der Kostenfunktion und können deshalb unverändert auch mit künstlichen neuronalen Netzen durchgeführt werden.¹⁾ In Cenobi ist eine kumulative Abweichungsanalyse implementiert, die, wie in Abb. B-14 (S. 80) dargestellt, die Gesamtabweichung $\Delta K = f(\mathbf{x}^{(\text{Ist})}) - f(\mathbf{x}^{(\text{Plan})})$ schrittweise in Teilabweichungen zerlegt und jeder Einflussgröße eine Teilabweichung zuordnet. Algorithmus E-2 verdeutlicht die Berechnung der Teilabweichungen in Pseudocode.

Algorithmus Kumulative Abweichungsanalyse

- 1: $\mathbf{x}^{(\text{Ist})} = (x_1^{(\text{Ist})}, \dots, x_I^{(\text{Ist})}) \leftarrow$ Istgrößen
 - 2: $\mathbf{x}^{(\text{Plan})} = (x_1^{(\text{Plan})}, \dots, x_I^{(\text{Plan})}) \leftarrow$ Plangrößen
 - 3: $\mathbf{x}^{(\text{Minuend})} \leftarrow \mathbf{x}^{(\text{Ist})}$
 - 4: **für** $i = 1$ bis I **führe aus**
 - 5: $\mathbf{x}^{(\text{Subtrahend})} \leftarrow \mathbf{x}^{(\text{Minuend})}$
 - 6: $x_i^{(\text{Subtrahend})} \leftarrow x_i^{(\text{Plan})}$
 - 7: $\Delta K_i^{\text{kum}} \leftarrow f(\mathbf{x}^{(\text{Minuend})}) - f(\mathbf{x}^{(\text{Subtrahend})})$
 - 8: $\mathbf{x}^{(\text{Minuend})} \leftarrow \mathbf{x}^{(\text{Subtrahend})}$
 - 9: **ende für**
 - 10: **Rückgabe** $\{\Delta K_i^{\text{kum}}\}$
-

Algorithmus E-2: Kumulative Abweichungsanalyse

¹⁾ Siehe dazu die Abschnitte B.IV.7.c und d, S. 72 ff.

Die Besonderheiten sowie Vor- und Nachteile dieser Art der Abweichungsanalyse wurden in Abschnitt B.IV.7.d (S. 78 ff.) ausführlich erörtert. Abb. E-24 zeigt den Ablauf einer kumulativen Abweichungsanalyse aus Sicht des Cenobi-Nutzers. Die Datensätze (a) sind teilweise bereits aus Abb. E-17 (S. 319) bekannt. Im Beispiel waren für die betrachtete Periode 480 Einkaufsrechnung und 1.000 verkaufte Produktarten geplant; tatsächlich sind aber 580 bzw. 1.250 Einheiten angefallen (b). Die Abweichungsanalyse zeigt, dass durch die 100 zusätzlichen Einkaufsrechnungen Kosten in Höhe von 4.344,00 € und durch die 250 zusätzlichen Produktarten Kosten in Höhe von 46.062,76 € angefallen sind (c).

(a) Datensätze

Datensatz	Inhalt
EG Einkauf - Vergangenheit	63 Beobachtungen von 3 Variablen
KS Einkauf - Vergangenheit	63 Beobachtungen von 1 Variablen
Plan-EG Einkauf	1 Beobachtungen von 3 Variablen
Ist-EG Einkauf	1 Beobachtungen von 3 Variablen



Datensatz Plan-EG Einkauf

	Anzahl Einkaufsrechnungen	Umsatz mit Handelsware in €	Anzahl verkaufter Produktarten
1	480	1.400.000	1.000



Datensatz Ist-EG Einkauf

	Anzahl Einkaufsrechnungen	Umsatz mit Handelsware in €	Anzahl verkaufter Produktarten
1	580	1.400.000	1.250

(b) Auswahldialog

$x^{(\text{Plan})}$ geplante Einflussgrößen Plan-EG Einkauf ▼
 $x^{(\text{Ist})}$ realisierte Einflussgrößen Ist-EG Einkauf ▼

Es wird die Gesamtabweichung

$$\Delta K = f(x^{(\text{Ist})}) - f(x^{(\text{Plan})})$$

berechnet und mit einer kumulativen Abweichungsanalyse in Teilabweichungen aufgespalten.

(c) Ergebnis

Kumulative Analyse der Ist-Plan-Kostenabweichung für Ist = Ist-EG Einkauf

		Mengenabw. ...	Kostenabw. Einkauf	Summe...
1	Anzahl Einkaufsrechnungen	100	4.343,9958	4.343,99...
2	Umsatz mit Handelsware in €	0	0	0
3	Anzahl verkaufter Produktarten	250	46.062,7632	46.062,7...
4	gesamte Ist-Plan-Abweichung	◆	50.406,759	50.406,7...

Abb. E-24: Abweichungsanalyse in Cenobi¹⁾

¹⁾ Als Vergangenheitswerte dienten die Daten aus Anhang A4. Ensemblegröße $L = 20$; trainiert mit *extreme learning machine*; genetischer Algorithmus eingestellt wie in Abb. D-50, S. 277.

F. Grenzen der statistischen Schätzung betrieblicher Kostenfunktionen

Die vorliegende Dissertation hat argumentativ und an einem ausgewählten realen Datensatz empirisch gezeigt, dass es zweckmäßig und in vielen Fällen vorteilhaft ist, betriebliche Kostenfunktionen mit künstlichen neuronalen Netzen zu schätzen. Die detailliert erörterte und in einer Softwareanwendung umgesetzte Konzeption ermöglicht es einem breiten Anwenderkreis, zentrale Aufgaben der Kostenrechnung mit künstlichen neuronalen Netzen zu lösen. Dabei wurde – bei aller methodischen Stringenz – durchgehend darauf geachtet, dass die entstandene Softwarelösung einfach zu bedienen bleibt. Die Resultate dieser Arbeit haben demnach einen unmittelbaren Nutzen für die Kostenrechnungspraxis.

Insbesondere die Auseinandersetzung mit dem empirischen Datensatz hat aber auch die Grenzen der statistischen Schätzung betrieblicher Kostenfunktionen offengelegt. Für einige der enthaltenen Kostenstellen konnte keine generalisierbare Kostenfunktion gefunden werden. Das kann in jedem Einzelfall verschiedene Gründe haben. Unwahrscheinlich ist jedoch, dass es überhaupt keinen regelmäßigen Zusammenhang gibt – die Kostenhöhe würde dann zufällig zustande kommen. Schon eher mag es sein, dass relevante Einflussgrößen, die zur Erklärung der Kostenhöhe notwendig wären, nicht im Datensatz enthalten sind, aber von einem mit den spezifischen Einzelheiten des Leistungsprozesses vertrauten Kostenrechner aufgespürt und erfasst werden könnten. Freilich wird es bei einigen (wenigen) Kostenstellen auch der Fall sein, dass keine messbaren Einflussgrößen feststellbar sind oder eines der anderen in Abschnitt C.III.3 (S. 134 ff.) beschriebenen Probleme vorliegt.

Die zum Teil sehr hohe Varianz der Kostenhöhe mit vereinzelt sogar negativen Kosten legt jedoch noch einen anderen Schluss nahe: dass nämlich ein merklicher Anteil der im Betrieb angefallenen Kosten nicht an den Stellen und nicht in den Perioden erfasst wurde, in denen der Güterverzehr tatsächlich stattgefunden hat, sondern nur mit dem Ziel, die Rechnung insgesamt aufgehen zu lassen. Das ist schade. Denn ohne eine geeignete Datenbasis, die den Betrachtungsgegenstand einigermaßen zutreffend abbildet, ist auch mit der besten Methodik – seien es nun künstliche neuronale Netze oder eine der zahlreichen anderen Business-Intelligence-Methoden – keine verlässliche Schätzung möglich; und ohne eine zuverlässige Kostenschätzung können die in dieser Dissertation erörterten Entscheidungsrechnungen ihr Unterstützungspotential nicht voll entfalten. Dabei müssten nur die bekannten Prinzipien der Kostenrechnung konsequent angewendet werden, um eine gute Datenqualität sicherzustellen.

Die Daten*quantität* stellt heute in aller Regel kein Problem mehr dar und mit der in dieser Arbeit entwickelten Anwendungssoftware werden auch die methodischen Hürden auf dem Weg zu besseren Entscheidungsrechnungen beseitigt. Die entwickelte Konzeption vereinfacht das Arbeiten mit statistischen Schätzmethoden im Allgemeinen und mit nichtlinearen Kostenfunktionen im Speziellen. Insgesamt bleibt daher zu hoffen, dass sie den einen oder anderen Kostenrechner dazu bewegt, diese Methoden einzusetzen und eine geeignete Datenbasis dafür zu schaffen.

Anhang

A1. Einflussgrößen des Beispieldatensatzes

Einflussgrößen									
Jahr	Monat	Rohmaterialverbr. in t	gel. Std. Spritzgussmaschinen	Zuhaltekraft gesamt in t	Anz. verk. Produktarten	Anz. verk. Varianten	Umsatz mit eigener Produktion in €	Umsatz mit Handelsware in €	Umsatz aus Rohmaterialverkauf in €
2009	1	-	46.477,44	20.429	1.057	2.563	13.284.602,60	2.022.151,13	465.649,87
2009	2	-	50.032,76	20.429	1.021	2.466	12.450.419,07	1.732.707,38	499.830,41
2009	3	-	47.046,09	22.505	992	2.563	12.206.393,08	1.241.461,91	329.604,99
2009	4	-	49.260,54	22.505	1.063	2.949	8.651.718,30	1.531.015,57	328.448,72
2009	5	1.501	45.933,68	22.505	1.097	2.990	10.632.229,58	2.105.203,31	167.050,05
2009	6	1.611	47.969,99	22.505	1.101	3.163	15.142.268,07	1.384.471,72	369.811,78
2009	7	1.341	48.210,39	22.505	1.135	3.243	11.828.076,24	1.486.136,12	194.599,27
2009	8	1.570	42.077,18	22.505	1.083	3.051	11.055.880,32	1.531.414,94	333.011,59
2009	9	1.511	45.607,36	22.505	1.182	3.181	13.033.971,41	1.695.295,21	552.662,35
2009	10	1.846	52.157,41	22.505	1.166	3.075	10.636.625,01	1.520.966,67	326.144,68
2009	11	1.949	49.176,59	22.505	1.043	2.712	9.483.330,29	1.963.708,72	388.207,22
2009	12	1.978	46.828,80	22.505	1.099	3.028	12.988.942,21	1.639.896,32	53.630,27
2010	1	1.910	48.657,63	22.505	1.149	3.113	15.333.891,33	1.694.874,29	196.026,46
2010	2	1.488	47.219,86	22.505	1.076	2.947	10.251.259,42	1.441.011,02	262.206,07
2010	3	1.585	50.536,29	22.505	1.168	3.285	12.329.177,04	1.324.431,96	286.815,43
2010	4	1.523	46.947,78	22.505	1.146	3.220	11.628.211,29	1.564.452,87	169.256,63
2010	5	1.641	51.103,77	22.505	1.218	3.264	11.116.669,44	1.819.107,42	640.454,71
2010	6	1.625	51.470,44	22.505	1.215	3.264	14.854.366,50	1.305.406,50	342.864,99
2010	7	1.637	52.487,20	22.523	1.165	3.290	14.004.607,56	3.227.576,49	517.309,94
2010	8	1.786	53.246,01	22.523	1.210	3.285	10.852.483,81	1.738.672,73	186.429,14
2010	9	1.843	52.337,79	22.661	1.218	3.388	13.060.558,51	1.631.669,24	315.449,26
2010	10	1.860	54.230,69	22.661	1.191	3.340	9.889.620,27	1.776.325,44	237.432,29
2010	11	1.845	51.153,04	22.661	1.165	3.025	9.720.395,95	1.488.496,24	425.526,51
2010	12	1.834	43.380,00	22.661	1.159	3.054	13.693.590,69	1.468.819,49	355.650,31
2011	1	1.833	48.773,76	22.661	1.149	2.926	12.053.647,58	2.064.330,58	151.218,39
2011	2	1.566	48.366,64	22.661	1.149	2.990	9.340.956,19	1.673.521,73	645.942,24
2011	3	1.772	52.819,08	22.661	1.113	3.065	9.467.942,37	1.390.025,56	460.757,44
2011	4	1.545	49.336,73	22.834	1.140	3.260	13.081.814,50	1.136.716,08	364.740,85
2011	5	1.661	53.373,69	22.834	1.137	3.231	11.672.396,33	1.596.798,71	352.612,64
2011	6	1.682	49.718,52	23.440	1.175	3.349	11.421.306,20	1.682.710,21	496.096,37
2011	7	1.677	55.999,83	23.440	1.102	3.127	8.728.667,52	2.630.834,10	666.860,72
2011	8	1.936	57.924,43	23.440	1.156	3.150	10.473.271,49	1.860.679,71	469.946,54
2011	9	1.883	54.964,86	23.440	1.210	3.582	15.106.513,35	2.036.194,94	342.164,58
2011	10	2.028	57.460,71	23.440	1.177	3.217	10.816.734,53	1.604.723,50	290.719,26
2011	11	1.895	-	25.343	1.095	3.155	7.161.758,32	1.161.099,14	169.045,59
2011	12	1.523	-	25.343	1.158	3.203	10.922.128,99	1.313.339,22	143.674,57
2012	1	1.684	50.050,43	25.378	1.125	3.340	12.539.385,10	2.062.437,24	291.884,24
2012	2	1.649	51.622,12	25.378	1.151	3.272	8.807.963,46	1.408.753,91	78.275,17
2012	3	1.755	58.125,28	25.689	1.159	3.641	8.276.192,33	1.278.972,19	365.099,13
2012	4	1.654	54.841,85	25.689	1.198	3.842	14.765.554,12	1.471.169,94	404.642,32
2012	5	1.930	60.044,10	25.689	1.203	3.814	14.617.752,77	1.117.746,38	1.160.708,93
2012	6	1.886	57.189,57	26.070	1.210	3.745	9.998.887,79	1.292.656,37	1.225.203,96
2012	7	1.999	59.319,47	26.070	1.175	3.954	11.616.424,90	1.323.740,14	888.976,12
2012	8	1.990	60.400,97	26.070	1.191	3.895	9.995.432,17	1.984.651,05	619.661,55
2012	9	1.878	56.760,09	26.070	1.230	4.169	16.176.099,86	1.722.810,63	396.556,97
2012	10	1.888	56.590,05	26.070	1.237	3.890	9.738.910,69	1.314.686,72	291.697,45
2012	11	2.000	58.136,14	26.070	1.156	3.636	7.049.496,60	1.399.925,92	440.340,07
2012	12	1.639	48.093,15	26.070	1.274	4.065	11.950.244,56	1.287.234,81	272.643,42

2013	1	1.708	53.593,08	25.343	1.191	3.849	9.376.897,71	2.073.050,20	312.398,20
2013	2	1.917	56.956,18	26.036	1.189	3.764	8.916.328,26	1.140.425,63	27.224,12
2013	3	2.018	61.176,19	26.036	1.196	3.873	9.535.348,88	1.310.915,02	205.872,52
2013	4	1.803	52.827,59	26.036	1.246	3.864	10.890.606,96	1.285.244,62	45.393,41
2013	5	1.831	56.918,62	26.036	1.211	3.902	14.068.302,23	924.609,91	382.623,75
2013	6	1.493	49.358,26	26.036	1.210	3.703	6.789.991,59	1.087.031,28	62.835,78
2013	7	1.886	55.623,30	26.036	1.184	3.719	10.554.238,45	908.319,13	68.245,36
2013	8	1.554	44.719,72	26.036	1.230	3.997	11.800.078,01	1.542.139,20	12.673,30
2013	9	1.544	45.417,64	26.036	1.222	3.901	11.802.935,76	945.119,63	26.834,85
2013	10	1.852	54.158,06	26.037	1.282	3.940	10.380.928,15	1.408.680,24	18.892,78
2013	11	2.020	55.919,72	26.037	1.218	3.892	8.533.715,83	1.065.614,49	6.432,02
2013	12	1.601	49.960,43	26.037	1.249	4.177	11.637.342,87	1.258.689,80	5.309,90
2014	1	1.616	53.530,96	26.037	1.268	4.117	12.704.546,11	1.728.324,61	13.578,15
2014	2	1.561	53.664,69	26.037	1.201	3.978	8.249.400,52	1.014.816,53	61.050,38
2014	3	1.945	56.468,59	26.037	1.256	4.203	10.381.940,95	1.396.979,42	21.629,12

Einflussgrößen

Jahr	Mo- nat	Anzahl ingesetz- ter Rohma- terialtypen	Anzahl Kreditoren (inkl. Ban- ken)	Anzahl Einkaufs- rechnun- gen	Anzahl Rechn. (ohne Einkauf)	Anzahl Waren- emp- fänger	Anzahl Lagerbe- wegun- gen	Fremdleis- tung Pro- duktion in €	Fremd- leistung Verpackung/ Montage in €	Fremdl. Entw. u. Versuchs- arbeiten in €
2009	1	59	483	-	3.413	1.033	637.024	3.272.255,91	45.034,76	32.385,33
2009	2	57	452	-	3.092	825	697.582	1.750.977,29	32.163,65	70.561,90
2009	3	66	419	448	2.738	812	716.911	1.136.597,34	37.297,90	25.398,56
2009	4	69	440	476	2.852	931	824.828	1.696.812,87	40.562,52	63.514,56
2009	5	69	415	474	3.688	883	680.417	1.064.863,78	61.311,01	,00
2009	6	73	478	491	3.819	990	688.318	1.107.613,42	47.212,27	52.045,93
2009	7	74	459	398	4.219	990	708.293	1.025.925,45	10.072,19	29.450,20
2009	8	67	408	343	4.724	860	691.575	847.307,62	28.207,78	,00
2009	9	67	381	398	5.302	1.121	821.711	956.082,57	15.931,53	44.771,38
2009	10	78	417	395	4.662	959	792.775	581.390,41	20.627,45	31.343,12
2009	11	74	419	346	3.842	907	733.575	433.959,87	46.368,64	18.874,18
2009	12	62	462	351	7.931	1.511	760.650	282.353,53	34.398,86	27.028,61
2010	1	61	401	433	4.887	936	808.068	513.698,53	26.675,15	20.127,50
2010	2	73	424	417	3.857	827	774.915	682.040,96	27.234,02	18.607,07
2010	3	62	412	602	4.241	1.111	808.260	602.703,40	79.518,51	21.952,07
2010	4	74	440	427	3.755	786	718.285	620.406,87	80.864,76	1.436,84
2010	5	71	452	528	4.390	907	927.150	1.179.181,49	37.486,53	,00
2010	6	66	467	531	4.397	884	893.630	1.042.630,74	42.999,34	,00
2010	7	71	424	576	4.598	1.083	869.136	1.215.142,81	13.669,23	35.779,98
2010	8	66	441	478	4.525	1.031	805.856	1.033.642,81	27.283,60	,00
2010	9	66	450	504	4.662	1.080	994.200	971.846,49	38.104,32	,00
2010	10	64	467	457	4.774	1.137	903.826	728.458,04	28.567,73	50.944,46
2010	11	66	466	351	4.300	910	833.564	905.281,34	18.839,74	11.856,24
2010	12	64	517	511	4.724	865	872.234	704.940,40	16.381,89	46.702,29
2011	1	69	455	424	3.696	1.040	943.676	554.218,16	21.684,96	21.317,14
2011	2	71	479	479	4.273	1.071	936.972	798.726,95	27.861,42	14.194,96
2011	3	66	509	452	4.120	1.080	1.314.051	952.987,65	57.807,68	141.245,28
2011	4	67	474	530	4.525	999	1.016.588	827.851,87	16.746,17	15.304,48
2011	5	62	478	453	4.212	1.037	908.000	527.724,61	26.489,73	63.911,62
2011	6	54	433	396	3.978	1.123	965.885	875.718,87	22.540,79	28.623,80
2011	7	59	500	519	4.787	1.215	950.994	1.167.807,69	50.255,81	38.455,79
2011	8	71	452	472	4.196	1.047	1.139.528	842.555,79	59.843,62	1.532,54
2011	9	76	467	488	4.492	1.012	1.215.117	848.556,10	34.416,43	45.666,14
2011	10	71	519	466	5.358	1.159	1.206.504	511.561,28	34.507,63	13.839,77
2011	11	74	478	511	3.994	1.023	1.090.913	351.425,16	3.084,29	1.263,27
2011	12	66	583	445	4.970	1.063	1.025.517	217.956,89	61.968,35	29.405,56
2012	1	62	460	421	4.361	1.113	1.075.508	364.893,65	23.500,53	46.420,30

2012	2	69	467	436	3.947	954.1045.235	145.050,14	26.389,19	63.953,07
2012	3	66	536	600	4.248	1.092.1.143.555	328.253,17	66.446,39	14.051,24
2012	4	55	459	502	3.634	903.1.229.538	485.563,74	162.414,11	37.135,93
2012	5	59	495	606	4.821	1.140.1.349.647	707.544,77	131.060,17	25.437,41
2012	6	64	478	734	4.610	952.1.148.238	1.823.006,30	118.031,67	,00
2012	7	62	472	934	5.325	1.121.1.221.158	2.044.743,43	129.391,61	30.064,88
2012	8	64	509	846	5.278	948.1.188.216	1.381.685,20	138.688,01	13.341,12
2012	9	66	486	713	4.977	896.1.191.854	850.588,22	89.391,51	23.675,14
2012	10	66	485	642	5.865	1.121.1.169.741	255.531,85	41.287,86	37.327,15
2012	11	64	464	594	4.032	907.1.182.217	347.105,46	58.756,86	2.115,98
2012	12	76	559	755	5.593	974.1.181.661	255.211,88	69.598,52	32.232,69
2013	1	64	474	696	3.894	934.1.384.330	132.188,29	58.136,73	11.381,58
2013	2	64	445	576	4.181	1.011.1.367.221	137.319,87	69.402,71	33.537,50
2013	3	54	493	485	4.157	929.1.442.034	212.855,26	73.644,72	4.499,25
2013	4	55	481	486	3.989	898.1.222.109	333.722,54	70.983,02	10.622,84
2013	5	55	460	509	4.368	990.1.363.891	520.858,84	56.609,30	5.202,96
2013	6	61	467	471	4.226	1.023.1.210.054	365.335,54	44.938,20	46.054,21
2013	7	57	511	555	4.465	1.038.1.271.283	97.198,55	147.373,45	2.403,59
2013	8	55	441	379	5.657	1.092.1.339.361	239.921,33	88.532,83	40.008,94
2013	9	50	431	360	4.716	926.1.273.602	27.664,83	58.761,10	4.174,74
2013	10	55	464	369	6.034	1.187.1.381.838	26.537,75	56.598,05	10.374,16
2013	11	55	471	443	4.778	1.033.1.456.849	6.983,22	27.654,71	8.419,46
2013	12	59	488	360	5.058	1.113.1.292.734	97.793,24	597.343,76	74.122,34
2014	1	64	486	637	5.524	1.217.1.415.114	46.266,72	55.894,59	26.636,13
2014	2	64	488	459	5.730	1.353.1.371.865	63.153,57	328.572,19	24.243,18
2014	3	59	472	512	6.393	1.346.1.493.963	301.949,99	204.941,28	9.759,66

A2. Kostenstellen des Beispieldatensatzes

Kostenstellen									
Jahr	Mo- nat	Produkt- entwick- lung	Konstruk- tion	Vetrieb (alle Regionen)	Auftragsab- wicklung (alle Regionen)	Zollab- wicklung	Waren- ausgang/ Versand	Kommis- sionierung	Hochregal- lager
2009	1	95.166,42	30.191,57	257.913,75	31.713,85	44.253,18	1.186.161,89	81.006,46	,00
2009	2	74.059,49	40.132,76	400.268,71	42.210,95	22.910,13	465.874,51	113.612,23	,00
2009	3	86.661,63	49.151,95	392.982,91	42.601,40	19.922,57	705.256,44	114.016,55	,00
2009	4	119.565,09	87.791,24	337.804,85	52.135,42	20.636,29	756.105,16	158.186,68	,00
2009	5	52.995,78	39.103,69	357.914,28	29.156,66	20.079,19	553.386,36	130.985,66	162.328,97
2009	6	111.253,18	39.830,94	394.189,61	42.156,77	19.964,59	1.377.698,19	129.653,11	187.940,55
2009	7	119.481,44	46.442,03	379.826,86	61.974,61	20.863,77	1.092.721,79	134.478,71	235.315,06
2009	8	60.017,70	41.100,67	336.210,66	52.380,86	21.526,95	428.352,25	122.896,00	185.254,80
2009	9	150.031,74	40.329,05	295.070,53	31.697,64	22.663,22	858.586,27	92.546,06	185.254,80
2009	10	81.734,46	68.247,94	449.167,93	107.439,44	50.879,15	491.004,70	88.856,32	162.758,14
2009	11	73.853,71	45.795,03	175.858,50	85.505,68	41.926,83	907.079,20	82.763,67	198.579,74
2009	12	53.852,49	47.188,06	143.134,83	-20.150,15	51.253,33	514.484,58	90.821,90	196.524,07
2010	1	95.881,70	46.293,73	304.465,32	77.945,32	50.524,17	1.430.981,30	85.802,59	187.791,73
2010	2	78.486,03	43.261,75	243.144,71	85.469,51	26.833,76	494.799,59	75.440,16	188.551,70
2010	3	83.626,16	42.407,14	248.370,69	81.828,78	26.881,07	857.440,51	74.156,81	187.781,09
2010	4	78.183,97	60.411,72	401.943,46	89.021,14	26.833,76	812.023,43	89.276,12	195.661,74
2010	5	89.514,97	41.328,30	243.800,43	85.650,20	26.834,92	457.610,64	81.984,73	187.968,24
2010	6	49.064,24	41.014,94	256.050,99	85.936,87	27.365,87	758.631,50	88.455,62	220.218,07
2010	7	73.024,36	46.015,71	326.210,49	85.823,61	28.735,45	1.206.534,50	89.851,19	193.400,32
2010	8	73.772,93	41.392,40	256.657,17	88.818,31	31.418,77	221.159,02	77.482,28	191.323,70
2010	9	45.220,52	56.178,66	239.737,68	86.283,36	29.259,64	645.309,72	83.992,42	193.400,32
2010	10	147.866,44	46.661,13	370.565,59	81.955,52	27.066,49	717.542,79	80.242,85	193.296,14
2010	11	105.259,39	47.110,41	274.674,59	90.933,89	26.829,59	430.236,36	77.641,60	191.323,70

2010	12	125.318,07	51.285,18	485.939,37	107.550,66	38.694,17	1.292.392,17	81.153,48	192.009,89
2011	1	75.516,53	63.343,12	252.323,99	101.906,75	52.229,94	626.037,64	76.131,98	192.232,22
2011	2	75.929,70	48.990,66	326.724,83	97.699,48	28.065,08	509.599,02	77.340,24	193.556,86
2011	3	148.239,61	57.784,42	281.654,81	100.743,17	30.047,07	840.059,39	73.778,11	202.398,98
2011	4	87.079,41	59.399,19	290.433,95	101.741,01	28.620,98	247.946,72	84.750,76	210.070,34
2011	5	151.422,12	58.248,14	247.438,76	105.893,09	28.830,54	454.425,34	84.810,36	217.233,56
2011	6	103.063,50	53.795,67	241.549,56	105.289,35	27.894,13	1.107.890,14	78.693,61	207.209,56
2011	7	82.630,42	56.868,87	288.015,23	85.886,71	24.891,60	735.957,24	83.059,97	205.167,80
2011	8	37.851,13	39.971,77	252.125,75	105.181,74	27.900,17	397.248,68	88.346,11	203.437,29
2011	9	102.800,47	53.260,96	320.261,66	114.343,72	27.957,48	897.166,08	84.843,68	206.553,08
2011	10	33.880,54	50.885,63	299.983,81	101.084,38	28.718,16	825.357,52	78.854,93	209.147,63
2011	11	56.659,69	43.912,86	245.303,22	123.722,36	27.968,75	421.381,86	85.337,79	206.916,83
2011	12	142.295,75	39.591,80	397.359,56	96.615,47	29.035,43	931.399,70	95.857,52	222.059,61
2012	1	65.003,27	69.913,53	241.298,56	93.471,26	47.302,29	956.988,49	74.358,28	203.435,56
2012	2	131.480,89	42.032,15	223.554,41	89.855,38	29.420,84	545.864,65	85.333,13	228.307,17
2012	3	51.561,27	58.188,27	245.853,77	86.838,26	28.497,54	705.114,40	74.839,67	203.435,56
2012	4	71.246,61	47.464,32	437.299,07	90.004,13	28.106,41	709.383,21	83.231,79	203.435,56
2012	5	80.168,13	40.249,90	223.782,13	89.671,08	27.816,79	1.024.573,47	69.990,85	212.093,92
2012	6	50.603,90	44.099,42	245.389,89	89.568,67	27.738,31	457.380,79	83.231,24	213.316,44
2012	7	72.234,19	61.300,58	413.884,48	86.729,35	28.188,56	975.765,95	100.204,69	198.873,06
2012	8	82.684,20	43.286,06	259.286,50	98.403,22	29.707,89	260.188,25	121.662,97	206.363,62
2012	9	63.305,22	54.471,16	290.645,26	90.750,29	27.723,91	409.206,13	84.947,14	197.897,92
2012	10	79.933,92	48.799,08	273.627,27	90.099,63	27.439,76	1.154.125,30	81.859,84	210.983,47
2012	11	57.393,29	46.443,38	192.023,16	76.606,57	26.850,87	567.507,60	74.854,67	31.860,46
2012	12	79.566,12	31.371,80	172.436,54	89.395,54	28.270,46	568.295,99	93.719,89	209.103,78
2013	1	67.083,71	62.622,65	260.556,41	94.796,10	46.686,94	414.699,21	98.763,03	197.897,92
2013	2	79.349,70	64.177,76	329.124,72	90.188,64	27.424,66	500.924,29	91.874,66	206.419,14
2013	3	74.222,61	49.464,84	265.736,50	81.520,18	27.062,84	386.472,39	70.521,06	207.568,89
2013	4	58.713,88	49.661,60	473.996,46	86.314,84	27.991,66	497.222,03	76.498,38	201.479,49
2013	5	47.958,81	58.614,61	262.012,30	99.326,82	27.540,84	667.411,28	71.974,25	203.736,15
2013	6	100.331,03	43.201,39	292.395,48	92.079,83	27.473,18	872.914,18	85.021,45	209.252,33
2013	7	64.388,64	49.844,43	290.892,05	89.469,99	29.263,14	530.587,83	94.621,38	203.307,29
2013	8	100.658,71	44.305,98	350.174,61	87.300,30	27.455,18	521.399,57	98.330,04	219.600,34
2013	9	67.213,89	48.488,09	269.227,11	97.792,03	27.332,49	779.423,27	93.079,15	200.599,77
2013	10	65.211,15	58.212,23	406.360,59	88.513,20	27.391,50	61.470,72	66.250,60	210.124,51
2013	11	52.312,87	58.769,08	320.232,51	98.076,96	27.514,71	446.442,78	74.172,26	198.709,36
2013	12	94.551,90	-4.608,27	510.978,50	93.416,09	27.999,95	710.796,94	91.820,63	39.242,55
2014	1	90.644,75	48.945,10	297.485,65	107.629,07	29.840,88	252.960,55	83.902,10	206.526,96
2014	2	131.248,87	92.641,99	339.006,01	100.412,39	53.731,64	669.956,97	189.041,89	203.487,34
2014	3	56.007,23	49.309,10	267.420,48	94.935,16	28.871,02	487.086,59	115.346,46	199.640,06

Kostenstellen

Jahr	Mo- nat	Marketing	Werkzeuge (alle Produkt- arten)	Werkzeug- bau	Techn. Leitung	QM Pro- duktions- prüfung	Angebots- kalkulation	Fertigungs- steuerung	Spritzguss- maschinen
2009	1	237.162,07	1.271.182,04	69.719,67	31.620,84	47.301,29	48.851,48	24.695,88	365.442,72
2009	2	360.909,15	2.562.990,01	78.180,34	32.496,84	49.952,97	49.503,55	48.764,97	384.424,07
2009	3	188.171,32	2.621.781,62	82.644,09	31.863,99	51.386,38	35.640,12	66.212,10	383.568,43
2009	4	191.231,30	2.668.858,09	71.190,45	65.744,67	54.859,26	40.823,82	48.306,02	389.206,90
2009	5	164.762,50	2.661.460,99	92.905,18	90.357,80	50.904,71	15.485,84	44.449,11	364.211,66
2009	6	191.502,19	2.674.710,82	88.723,76	57.865,60	35.422,06	15.489,83	43.001,00	372.770,43
2009	7	201.085,92	2.752.091,36	93.120,04	52.706,14	56.566,67	16.242,61	43.346,94	394.923,76
2009	8	206.928,84	2.790.442,41	92.128,15	57.438,33	55.876,42	15.921,61	47.528,78	352.011,48
2009	9	210.052,76	2.767.121,04	90.531,18	57.575,73	47.435,37	15.472,04	42.964,69	353.560,46
2009	10	159.651,23	2.799.464,61	85.844,21	49.386,57	48.283,66	15.466,59	42.925,79	366.791,26
2009	11	168.709,13	2.893.357,80	85.167,84	54.220,43	46.499,31	15.464,50	43.030,16	405.893,12
2009	12	223.480,67	2.823.407,51	76.726,81	15.656,43	47.522,13	15.737,57	39.692,48	353.299,13
2010	1	39.164,61	2.562.579,65	83.035,43	54.043,08	47.229,28	15.497,03	39.238,78	359.136,36

2010	2	71.716,79	2.428.105,40	76.244,02	51.723,38	47.244,61	15.442,47	34.113,10	384.503,85
2010	3	61.070,71	2.421.729,17	83.810,82	44.342,99	43.262,08	15.442,47	42.174,95	359.484,86
2010	4	185.799,51	2.444.093,83	81.589,66	43.596,90	47.491,81	15.442,47	36.982,79	423.039,52
2010	5	53.859,46	2.467.974,78	81.194,27	45.307,55	37.381,07	33.853,41	36.714,81	355.347,14
2010	6	90.459,67	2.509.414,12	86.808,43	45.457,75	40.486,55	30.389,28	36.647,35	393.129,57
2010	7	76.959,12	2.522.206,07	73.840,91	53.176,84	37.524,95	29.239,36	34.989,52	368.058,38
2010	8	85.798,19	2.501.398,86	87.145,45	49.388,39	38.109,48	15.834,46	36.868,29	406.682,96
2010	9	80.374,68	2.493.852,37	89.716,78	44.559,13	37.379,50	15.877,35	38.594,63	408.800,58
2010	10	120.314,76	2.636.687,80	86.916,80	48.734,86	37.497,26	15.843,55	36.589,64	459.429,20
2010	11	98.682,70	2.657.643,67	91.793,31	45.854,73	44.365,13	15.970,67	36.629,87	398.409,56
2010	12	294.592,59	2.654.567,59	90.136,60	81.936,64	37.531,56	16.519,75	47.332,82	384.808,25
2011	1	53.007,32	2.680.146,76	87.706,62	48.306,66	37.354,85	15.953,09	36.735,75	402.234,17
2011	2	143.526,52	2.573.175,97	92.170,08	49.514,39	40.053,93	15.914,60	38.520,93	426.529,81
2011	3	313.215,31	2.619.968,06	101.593,89	46.819,82	41.229,23	15.997,44	32.456,36	405.536,29
2011	4	89.263,28	2.659.887,88	106.658,55	45.630,04	41.127,98	16.202,28	36.523,08	392.185,90
2011	5	177.984,03	2.658.034,50	92.207,42	45.804,84	38.732,24	15.900,07	37.650,31	434.859,02
2011	6	180.783,17	2.624.615,09	95.651,68	47.512,13	39.779,61	15.936,08	36.645,33	449.791,58
2011	7	76.628,10	2.645.648,68	91.353,72	54.786,93	37.377,35	16.118,61	39.578,65	480.466,28
2011	8	185.033,08	2.662.815,04	92.318,80	45.536,04	37.843,26	15.946,13	36.712,38	423.587,60
2011	9	99.490,66	2.622.183,83	101.065,39	46.742,19	38.249,48	15.906,25	36.718,18	433.027,51
2011	10	116.475,03	2.681.637,26	110.349,14	45.794,15	37.837,20	15.971,80	36.656,65	441.180,09
2011	11	220.746,98	2.716.359,35	89.823,36	45.291,87	37.707,26	15.914,90	39.949,83	514.453,94
2011	12	146.146,90	2.762.845,13	100.658,50	44.590,47	39.724,69	16.605,67	37.127,36	503.943,26
2012	1	71.217,93	2.849.288,19	125.341,30	71.888,16	38.001,48	15.847,91	50.623,94	446.764,12
2012	2	232.819,45	2.787.196,29	120.206,21	80.241,47	37.831,18	15.799,23	50.502,79	445.096,09
2012	3	293.387,28	2.784.515,39	87.875,40	72.960,51	37.719,44	3.737,21	39.223,65	465.999,62
2012	4	145.763,89	2.745.210,94	102.578,54	71.734,84	37.591,36	5.156,32	31.151,56	452.441,93
2012	5	533.001,31	2.802.865,63	106.794,43	71.190,51	34.234,07	9.895,50	47.984,77	473.969,41
2012	6	127.473,01	2.884.752,89	105.571,34	73.099,17	32.961,26	15.704,57	37.112,12	516.599,18
2012	7	239.101,02	2.820.674,93	106.754,92	81.504,70	35.005,96	-2.603,04	37.254,04	517.446,92
2012	8	125.096,57	2.840.719,94	111.342,34	71.234,44	37.685,24	9.220,77	53.069,81	487.633,60
2012	9	247.208,80	2.888.098,55	108.428,88	88.549,39	38.046,45	14.459,73	46.377,90	507.370,37
2012	10	154.648,80	2.930.982,04	102.788,34	64.524,50	38.838,11	-373,10	47.101,95	467.360,09
2012	11	91.520,34	1.492.996,18	84.490,03	60.922,16	35.218,80	845,96	41.137,18	18.937,17
2012	12	143.310,07	1.552.517,26	116.063,95	47.022,47	36.896,68	3.319,73	48.122,64	485.874,52
2013	1	266.328,41	2.964.371,65	120.099,26	77.563,76	43.913,50	2.985,65	50.601,51	496.913,20
2013	2	385.020,93	2.873.566,90	105.523,07	70.536,03	36.794,67	15.956,97	47.730,24	493.284,61
2013	3	136.124,62	2.941.692,63	109.491,83	80.947,53	35.101,45	19.810,56	45.635,93	493.389,18
2013	4	151.037,07	2.998.633,84	107.318,12	70.012,82	36.696,79	1.932,98	52.824,16	424.482,75
2013	5	88.842,79	3.077.451,86	107.687,80	71.728,18	36.871,94	1.932,98	47.909,06	468.982,70
2013	6	121.262,86	3.101.083,52	119.735,74	72.775,19	41.135,12	1.932,98	47.894,89	484.022,40
2013	7	114.837,62	3.030.852,77	109.145,90	77.360,47	39.828,78	1.932,98	45.771,44	476.993,73
2013	8	78.528,09	3.016.289,75	111.766,76	69.035,84	37.310,80	-6.896,61	48.363,40	483.001,83
2013	9	96.170,61	3.008.208,15	105.137,32	70.176,62	48.745,95	2.709,81	50.053,77	458.485,67
2013	10	108.932,89	3.018.944,67	103.021,37	48.403,10	42.081,56	1.932,98	47.618,42	463.316,96
2013	11	90.580,70	3.053.988,81	108.996,18	67.238,62	38.117,15	1.932,98	50.073,84	477.788,68
2013	12	162.019,80	1.619.633,18	94.986,10	64.016,79	35.883,82	2.064,16	46.842,98	111.620,15
2014	1	409.044,75	1.618.973,30	109.473,95	98.892,77	40.249,79	1.941,98	52.133,55	468.310,61
2014	2	206.382,19	3.091.926,82	202.105,54	76.094,00	42.888,13	1.941,98	55.194,22	510.653,94
2014	3	290.138,10	3.129.221,09	104.817,28	68.770,73	36.862,40	1.941,98	48.890,74	470.143,85

Kostenstellen

Jahr	Mo- nat	Etikettier- anlagen Produktion	Inmold Labeling	Montage/ Verpa- ckung	Siebdruck	Recycling/ Mahlen	Auftrags- bereit- stellung	Methoden- zeit- messung	Einrichter
2009	1	19.497,68	1.655,58	170.820,35	37.953,80	24.262,21	20.857,19	,00	66.075,79
2009	2	17.855,43	7.041,63	161.476,19	39.476,06	25.690,18	23.683,83	,00	68.633,14
2009	3	17.556,05	89.529,02	170.754,16	48.022,46	24.522,48	24.531,26	,00	72.158,36

2009	4	27.348,08	1.905,29	166.760,47	36.080,35	24.407,40	19.218,46	,00	68.540,42
2009	5	20.375,85	1.905,29	164.059,78	36.983,83	24.953,29	19.263,11	,00	70.145,57
2009	6	18.053,69	8.337,19	172.737,41	34.158,77	27.452,07	19.487,99	18.621,54	71.312,49
2009	7	18.959,49	192.372,50	158.730,25	34.958,51	29.534,24	19.484,27	18.632,93	71.814,20
2009	8	18.959,49	2.047,20	214.083,23	35.828,59	25.741,11	20.886,82	18.326,82	86.918,18
2009	9	18.959,49	-33.879,62	174.832,55	37.305,86	25.046,92	19.219,07	17.726,50	82.715,42
2009	10	25.183,80	113.850,49	174.871,95	35.552,79	24.619,48	21.154,15	17.726,50	87.588,32
2009	11	25.013,08	12.321,25	162.720,03	35.374,23	25.090,87	19.549,22	17.726,50	83.730,28
2009	12	21.953,28	7.240,81	143.122,34	47.568,56	28.149,81	16.883,64	20.094,19	96.286,45
2010	1	23.708,47	21.566,10	156.006,67	34.394,14	26.171,06	20.118,35	17.932,00	81.682,84
2010	2	18.469,88	20.323,14	144.256,02	34.649,99	25.109,82	28.967,20	18.025,62	85.095,03
2010	3	19.364,00	21.529,23	162.318,48	33.579,10	25.784,72	28.967,20	17.844,78	93.165,38
2010	4	24.880,80	24.978,83	165.010,47	34.435,53	25.109,82	17.307,94	18.316,43	84.994,56
2010	5	20.162,20	27.088,05	152.067,40	35.847,51	25.536,91	20.210,29	18.320,15	94.006,44
2010	6	19.583,96	23.640,63	155.558,90	37.512,54	26.438,87	28.739,95	18.149,27	81.678,98
2010	7	21.220,41	21.502,79	136.196,71	34.026,94	29.461,90	19.357,93	17.932,00	93.153,49
2010	8	21.070,72	25.623,04	163.597,30	34.970,24	27.491,96	22.687,59	18.017,83	84.975,54
2010	9	19.339,95	24.361,73	159.686,84	36.852,30	25.972,57	21.107,14	18.888,71	93.407,44
2010	10	24.176,05	24.938,41	165.258,12	34.232,04	26.247,03	21.897,07	22.465,42	113.532,06
2010	11	20.165,66	31.744,45	156.555,76	41.635,81	25.556,29	20.214,36	18.441,38	112.604,10
2010	12	19.056,57	34.998,78	172.971,38	39.070,95	46.071,90	15.274,71	20.781,89	116.071,95
2011	1	21.129,04	50.802,23	163.085,53	51.487,83	25.755,99	21.837,91	17.943,42	107.321,46
2011	2	20.499,65	38.267,10	154.455,76	51.679,72	26.145,01	30.097,67	17.870,83	106.846,67
2011	3	23.614,57	69.831,97	175.886,65	51.489,87	26.182,74	20.593,17	19.123,41	123.138,18
2011	4	20.499,65	37.362,74	174.037,79	45.956,89	27.063,74	20.415,91	18.082,04	110.159,15
2011	5	20.515,15	43.021,70	167.586,75	42.809,39	38.780,36	25.166,70	17.983,03	103.250,72
2011	6	20.932,28	40.116,15	164.967,86	42.677,77	28.190,13	21.580,04	17.879,57	95.010,42
2011	7	22.431,35	41.643,06	154.657,77	43.185,26	26.722,40	25.819,52	17.966,70	76.060,98
2011	8	20.596,31	42.639,76	159.649,34	42.414,84	23.706,18	38.457,69	18.478,24	87.372,44
2011	9	20.919,89	42.823,51	165.399,64	43.043,65	18.338,24	26.600,52	18.297,48	89.498,43
2011	10	20.877,38	40.784,71	167.975,10	42.957,23	24.895,31	21.282,12	18.537,99	98.639,23
2011	11	21.455,50	57.781,82	163.806,71	43.206,11	27.410,00	21.503,14	23.730,50	109.743,81
2011	12	20.397,55	58.335,76	214.023,51	48.817,94	19.520,35	23.851,43	19.282,41	100.852,33
2012	1	20.515,22	44.395,40	157.218,09	42.550,15	26.409,78	23.714,11	18.056,51	99.810,51
2012	2	21.488,91	62.122,13	157.807,99	42.587,84	26.350,08	21.915,36	20.212,00	110.910,49
2012	3	21.218,85	50.535,02	175.137,77	42.280,19	28.109,33	21.957,74	17.413,85	99.873,97
2012	4	23.202,71	61.490,67	162.654,07	47.214,52	26.311,05	23.301,04	18.066,82	101.514,98
2012	5	23.261,82	48.011,61	150.598,45	42.801,68	27.896,12	24.708,72	19.032,87	115.587,77
2012	6	21.678,82	46.820,15	172.586,50	43.783,29	26.418,26	27.987,97	19.941,45	132.304,19
2012	7	25.861,48	46.398,58	159.948,41	44.129,62	30.577,03	23.044,97	17.119,51	132.765,83
2012	8	23.840,06	48.328,23	149.546,96	45.933,63	31.332,34	23.519,34	16.251,90	130.263,51
2012	9	22.205,93	47.422,44	171.612,91	45.871,21	27.531,76	23.825,28	15.649,16	130.567,50
2012	10	22.098,64	49.067,67	149.933,94	41.387,99	30.471,92	24.290,96	17.301,44	124.859,57
2012	11	,00	2.881,07	118.150,50	19.128,79	20.512,63	21.721,11	15.096,85	179.741,99
2012	12	21.427,20	55.961,85	189.477,25	46.939,85	28.066,66	22.369,07	18.582,88	130.035,66
2013	1	21.631,40	46.644,68	174.671,07	55.852,51	27.650,73	30.178,68	30.913,40	132.641,67
2013	2	22.945,83	45.896,77	134.324,46	42.857,64	30.646,68	23.894,72	18.208,05	131.048,44
2013	3	23.411,51	56.654,64	129.802,56	39.878,62	26.776,74	25.503,75	6.232,35	104.577,73
2013	4	21.973,70	51.312,02	126.059,98	42.523,60	28.310,92	23.879,49	17.733,06	116.748,44
2013	5	22.555,08	54.493,75	149.498,09	42.802,06	26.104,69	23.370,74	17.775,22	112.855,77
2013	6	21.631,40	73.014,04	170.432,53	47.023,94	28.992,40	25.569,10	18.436,10	126.113,33
2013	7	21.631,40	68.557,30	147.797,77	43.272,17	29.983,63	27.032,61	18.026,31	125.453,11
2013	8	22.091,72	63.096,98	162.516,42	41.811,51	8.659,14	22.304,66	17.908,81	124.947,83
2013	9	22.808,84	59.464,86	148.034,78	47.826,60	33.713,79	26.500,74	17.996,98	121.184,31
2013	10	21.631,40	59.684,16	142.137,48	41.999,20	18.153,25	22.654,40	17.909,33	123.733,79
2013	11	21.631,40	63.360,21	152.833,95	42.352,12	22.853,77	23.229,10	18.250,07	119.474,69
2013	12	,00	7.586,93	174.448,49	23.724,53	10.716,46	21.047,79	18.598,65	-28.709,41
2014	1	22.943,48	62.764,06	162.373,26	44.881,01	16.342,92	24.040,73	31.929,85	125.290,23
2014	2	21.623,51	60.938,38	282.536,21	61.208,93	26.066,41	50.674,71	20.099,81	252.597,09
2014	3	22.291,14	72.784,87	150.661,79	41.050,92	15.405,42	23.682,70	18.061,10	123.029,09

Kostenstellen

Jahr	Mo- nat	Ver- fahrens- technik	Aushilfen, Jobber, Schueler	Einkauf	Lager Fertig- und Halb- fertigware	Lager Roh-, Hilfs-, Be- triebstoffe	Warenum- zeichnung	Waren- eingang	QM Waren- eingangs- prüfung
2009	1	14.589,69	137.190,94	33.781,66	287.041,76	12.812,64	9.997,95	-26.779,38	11.322,64
2009	2	14.055,65	104.301,87	44.442,84	280.647,62	11.171,88	10.825,22	23.950,60	11.183,90
2009	3	14.513,89	128.074,87	34.495,82	328.935,41	26.163,44	9.537,80	24.892,61	11.357,01
2009	4	13.976,83	165.838,71	20.495,98	360.539,91	11.114,08	10.347,86	22.971,77	11.378,57
2009	5	14.055,65	161.118,91	31.360,29	432.081,88	11.167,38	10.399,08	22.774,08	11.363,58
2009	6	14.137,85	209.891,84	34.522,73	320.640,35	11.253,87	9.454,91	23.103,79	11.899,92
2009	7	14.511,99	198.735,40	41.143,67	384.100,71	13.275,69	10.662,12	20.863,42	10.436,70
2009	8	14.587,27	84.953,18	46.179,33	374.209,26	15.816,16	11.294,53	23.712,40	11.713,56
2009	9	14.122,10	243.813,43	27.219,30	282.999,43	11.239,04	10.044,50	24.019,44	11.726,90
2009	10	16.012,17	229.094,38	27.292,79	338.859,77	11.234,94	10.278,12	24.526,57	13.369,38
2009	11	16.081,39	221.824,16	35.658,81	399.979,74	11.456,89	10.490,11	24.957,33	11.355,98
2009	12	18.118,90	216.414,92	-17.096,30	361.717,14	-10.417,80	12.128,12	73.700,51	6.391,01
2010	1	14.933,02	231.189,17	8.186,31	331.758,62	11.367,42	15.717,46	25.473,71	10.923,96
2010	2	18.824,77	280.923,83	31.212,51	297.074,69	11.173,05	21.111,99	24.596,74	11.748,69
2010	3	14.860,25	302.687,71	60.122,63	298.678,04	11.190,36	20.821,87	24.521,03	10.924,17
2010	4	14.860,25	520.034,42	46.284,96	299.431,19	11.190,36	15.990,02	15.434,49	19.311,96
2010	5	17.238,93	546.403,97	27.187,07	328.603,53	11.189,47	22.347,49	22.553,84	19.221,23
2010	6	16.618,54	536.374,26	49.965,76	325.610,40	16.816,86	14.845,46	24.403,82	20.346,84
2010	7	13.894,72	657.596,73	27.859,10	336.540,42	29.826,05	12.049,38	16.169,94	20.159,26
2010	8	14.860,25	440.076,53	40.251,47	262.726,18	13.673,57	17.997,76	18.640,77	284,83
2010	9	14.860,25	441.260,03	34.312,61	374.261,78	27.285,13	16.351,09	24.305,13	1.449,20
2010	10	19.706,47	303.084,25	228.985,95	311.577,47	40.324,57	1.752,14	22.757,26	18.359,42
2010	11	23.032,53	213.892,44	-75.460,49	335.572,86	63.985,90	18.609,15	22.698,84	-911,25
2010	12	18.043,19	32.403,15	85.002,26	381.700,06	36.483,70	3.620,32	11.970,66	17.726,24
2011	1	15.254,38	267.339,40	44.970,65	268.737,55	54.047,45	15.428,12	22.413,47	9.987,48
2011	2	17.565,51	296.233,67	46.572,19	257.590,63	72.070,97	17.009,20	28.881,52	12.211,74
2011	3	14.886,23	373.671,07	73.907,08	317.199,78	61.141,06	14.317,31	24.404,72	10.836,80
2011	4	14.779,35	281.745,65	99.066,55	291.568,32	50.780,91	14.920,13	22.777,38	10.268,91
2011	5	16.821,27	437.585,03	61.985,60	342.432,57	27.221,94	14.055,05	22.291,54	9.252,86
2011	6	16.908,40	477.726,42	83.822,19	301.785,16	63.046,63	14.792,42	22.476,20	10.144,37
2011	7	14.416,86	667.041,26	5.494,72	283.437,68	54.067,33	14.527,56	22.437,51	6.886,57
2011	8	16.935,49	572.579,00	113.544,60	240.465,18	29.317,78	14.112,76	22.754,19	10.086,08
2011	9	29.322,75	633.645,84	76.215,90	388.803,57	24.899,56	15.232,49	22.936,71	10.233,06
2011	10	29.640,30	386.414,22	46.217,38	289.631,61	24.967,57	14.595,31	22.319,92	10.286,81
2011	11	28.613,93	294.412,14	44.292,12	302.674,17	16.947,25	13.967,66	23.362,76	9.928,04
2011	12	34.789,98	112.162,70	52.611,80	371.920,43	28.162,18	20.638,69	26.442,69	10.395,50
2012	1	28.299,33	276.491,82	48.714,99	299.284,67	14.471,13	14.230,09	23.926,15	9.856,06
2012	2	31.111,74	467.226,91	85.470,97	273.409,46	17.568,63	16.297,79	21.825,01	9.781,48
2012	3	29.205,66	476.133,23	113.114,54	326.589,51	19.153,07	13.052,73	22.629,60	9.768,85
2012	4	31.037,26	959.203,91	102.967,35	249.230,80	18.898,13	19.296,86	22.135,43	9.807,23
2012	5	32.645,39	672.642,93	97.914,75	324.114,49	24.269,74	12.923,64	21.749,34	10.779,90
2012	6	30.977,69	866.299,89	113.427,69	370.772,11	25.552,22	15.021,11	23.544,73	9.764,97
2012	7	29.667,82	1.039.732,43	166.955,44	412.896,42	18.764,93	14.181,81	25.571,00	10.738,99
2012	8	31.273,94	787.413,97	134.017,05	389.257,60	33.902,08	13.467,88	22.860,57	9.719,73
2012	9	30.021,50	528.360,00	155.477,61	350.623,43	1.516,38	14.656,23	13.830,48	9.977,65
2012	10	28.441,40	215.563,42	111.775,14	337.494,81	23.980,52	12.797,31	19.993,73	10.232,16
2012	11	27.266,30	205.091,92	-49.228,61	330.209,22	20.626,53	12.912,30	18.415,49	7.839,20
2012	12	29.596,78	288.292,96	101.210,58	424.256,99	13.706,63	18.172,11	19.772,12	9.400,14
2013	1	32.261,80	386.553,78	78.612,53	358.162,27	47.368,38	6.476,79	24.749,80	10.237,55
2013	2	29.293,85	434.267,05	62.952,97	335.136,25	58.694,06	8.749,82	23.925,58	9.637,60
2013	3	27.263,70	533.387,14	108.952,24	412.162,75	16.611,17	-8.946,83	27.158,97	8.724,76
2013	4	31.180,42	389.604,41	77.600,34	352.344,01	43.423,05	14.210,45	29.363,52	9.450,09
2013	5	28.794,86	362.889,46	147.656,48	294.427,05	84.135,06	13.970,69	21.272,41	9.530,02
2013	6	32.938,74	426.785,16	113.673,42	429.702,62	112.485,37	4.554,19	32.913,74	9.628,40
2013	7	32.700,45	507.842,79	103.569,62	377.699,62	17.112,17	11.478,40	25.099,70	9.455,43
2013	8	30.832,54	385.162,36	90.325,89	378.518,96	13.636,44	11.478,40	25.762,45	9.102,58
2013	9	29.379,43	437.607,18	107.546,68	289.126,46	29.626,54	9.221,90	24.409,57	9.522,40
2013	10	28.975,87	378.143,84	96.574,11	264.350,37	21.753,06	11.478,40	16.272,87	9.504,06

2013	11	29.247,65	525.116,42	89.819,14	289.925,45	18.230,17	11.478,40	13.549,56	10.025,57
2013	12	31.299,00	-53.862,56	69.773,09	407.981,97	13.679,96	10.318,35	12.696,16	9.263,43
2014	1	31.288,94	409.622,43	58.635,47	276.078,83	42.492,23	8.330,34	18.335,82	11.032,08
2014	2	58.576,87	578.706,04	88.923,31	455.643,70	54.331,66	16.585,23	40.295,58	19.960,40
2014	3	29.536,37	509.794,44	90.124,47	287.785,40	33.672,04	8.330,34	17.300,02	9.878,27

A3. Datensatz Fertigung

Einflussgrößen	Rohmaterialverbrauch in t, gel. Std. Spritzgussmaschinen, Zuhaltkraft gesamt in t, Anz. verk. Produktarten, Fremdleistung Produktion in €, Anz. eingesetzter Rohmaterialtypen. Für die Experimente in den Abschnitten D.X.2, D.XII.3 und D.XIII.6 wurden die fehlenden Werte der Einflussgrößen Rohmaterialverbrauch und gel. Std. Spritzgussmaschinen durch den Mittelwert der angrenzenden Beobachtungen ersetzt. Bei diesen Experimenten wurde also keine multiple Imputation vorgenommen.
Kostenstellen	Fertigungssteuerung, Spritzgussmaschinen, Montage/Verpackung, Siebdruck, Auftrags-bereitstellung.
Trainingsdaten	1/2009 bis 9/2013 (Trainingsdaten für die Trainingsalgorithmen in Abschnitt D.X.2 bzw. Trainings- und Validierungsdaten für die Algorithmen in D.XII.3 und D.XIII.6).
Testdaten	10/2013 bis 3/2014 (zur Berechnung des Out-of-Sample-Fehlers).

A4. Datensatz Einkauf

Einflussgrößen	Umsatz mit eigener Produktion in €, Umsatz mit Handelsware in €, Anzahl Einkaufsrechnungen, Anzahl Kreditoren (inkl. Banken), Anzahl verkaufter Produktarten
Kostenstellen	Einkauf
Trainingsdaten	1/2009 bis 9/2013
Testdaten	10/2013 bis 3/2014

A5. Datensatz Hochregallager

Einflussgrößen	Anzahl Lagerbewegungen, Anzahl verkaufter Produktarten.
Kostenstellen	Hochregallager (der Ausreißerwert 11/2012 wurde beseitigt und als fehlender Wert behandelt)
Trainingsdaten	5/2009 bis 9/2013
Testdaten	10/2013 bis 3/2014

A6. Datensatz Verfahrenstechnik

Einflussgrößen	Rohmaterialverbrauch in t, geleistete Std. Spritzgussmaschinen, Anzahl verkaufter Produktarten, Fremdleistung Produktion in € (der Ausreißerwert 1/2009 wurde beseitigt und als fehlender Wert behandelt), Anzahl eingesetzter Rohmaterialtypen, Umsatz mit eigener Produktion in €.
Kostenstellen	Verfahrenstechnik
Trainingsdaten	1/2009 bis 9/2013
Testdaten	10/2013 bis 1/2014 und 3/2014 (der Ausreißer in 2/2014 wurde nicht berücksichtigt).

A7. Datensatz Auftragsabwicklung

Einflussgrößen	Umsatz mit Handelsware in €, Umsatz mit eigener Produktion in €, Anzahl Rechnungen (ohne Einkauf) (der Ausreißerwert 12/2009 wurde beseitigt und als fehlender Wert behandelt), Anzahl Warenempfänger (der Ausreißerwert 12/2009 wurde beseitigt und als fehlender Wert behandelt), Anzahl Lagerbewegungen, Anzahl verkaufter Produktarten.
Kostenstellen	Auftragsabwicklung (alle Regionen)
Trainingsdaten	1/2009 bis 9/2013
Testdaten	10/2013 bis 3/2014

A8. Lineare Regressionsmodelle für A4 bis A7

Einkauf

	<i>Koeffizienten</i>
Y-Achsenabschnitt	-287264,978811081
Umsatz mit Handelsware in €	-0,0216793075
Umsatz mit eigener Produktion in €	0,0023417019
Anzahl Einkaufsrechnungen	81,3894823681
Anzahl Kreditoren (inkl. Banken)	184,0861155258
Anzahl verkaufter Produktarten	202,736960957

Hochregallager

Koeffizienten

Y-Achsenabschnitt	42589,7911584208
Anzahl Lagerbewegungen	0,0087029286
Anzahl verkaufter Produktarten	124,1353378928

Verfahrenstechnik

Koeffizienten

Y-Achsenabschnitt	-18799,0260674405
Rohmaterialverbrauch in t	0,0256325922
geleistete Std. Spritzgussmaschinen	-0,0970998476
Anzahl verkaufter Produktarten	64,5822113088
Fremdleistung Produktion in €	-0,0021963941
Anzahl eingesetzter Rohmaterialtypen	-284,7413803438
Umsatz mit eigener Produktion in €	-0,0007761093

Auftragsabwicklung

Koeffizienten

Y-Achsenabschnitt	-101360,564107836
Umsatz mit Handelsware in €	0,007112259
Umsatz mit eigener Produktion in €	-0,0015128643
Anzahl Rechnungen (ohne Einkauf)	-8,7353931335
Anzahl Warenempfänger	-19,6413015237
Anzahl Lagerbewegungen	0,0380172923
Anzahl verkaufter Produktarten	181,5294854619

Literaturverzeichnis

- Adam, Dietrich*: [Kostenbewertung] Entscheidungsorientierte Kostenbewertung. Wiesbaden 1970.
- Adam, Dietrich*: [Grenzkostenrechnung] Grenzkostenrechnung. In: Handwörterbuch des Rechnungswesens. Hrsg. von K. Chmielewicz und M. Schweitzer. 3. Aufl., Stuttgart 1993, S. 824–832.
- Adeodato, Paulo J. L., Adrian L. Arnaud, Germano C. Vasconcelos, Rodrigo C. L. V. Cunha und Domingos S. M. P. Monteiro*: [MLP Ensembles] MLP ensembles improve long term prediction accuracy over single networks. In: International Journal of Forecasting (27) 2011, S. 661–671.
- Adya, Monica und Fred Collopy*: [Forecasting] How effective are neural networks at forecasting and prediction? A review and evaluation. In: Journal of Forecasting (17) 1998, S. 481–495.
- Aggarwal, Charu C.*: [Outlier Analysis] Outlier Analysis. New York u. a. 2013.
- Akaike, Hirotugu*: [Model Identification] A New Look at the Statistical Model Identification. In: IEEE Transactions on Automatic Control (19) 1974, S. 716–723.
- Alba, Enrique und Francisco Chicano*: [Genetic Algorithms] Genetic Algorithms. In: Metaheuristic Procedures for Training Neural Networks. Hrsg. von E. Alba und R. Martí. New York 2006, S. 109–137.
- Alba, Enrique und Rafael Martí (Hrsg.)*: [Metaheuristic Procedures] Metaheuristic Procedures for Training Neural Networks. New York 2006.
- Alex, Björn*: [Neuronale Netze] Künstliche Neuronale Netze in Management-Informationssystemen. Wiesbaden 1998.
- Altenburger, Otto A.*: [Dienstleistungen] Ansätze zu einer Produktions- und Kostentheorie der Dienstleistungen. Berlin 1980.
- Anastasiadis, Aristoklis D., George D. Magoulas und Michael N. Vrahatis*: [Resilient Propagation] New globally convergent training scheme based on the resilient propagation algorithm. In: Neurocomputing (64) 2005, S. 253–270.
- Anders, Ulrich und Olaf Korn*: [Model Selection] Model Selection in Neural Networks. In: Neural Networks (12) 1999, S. 309–323.
- Anders, Ulrich und Andrea Szczesny*: [Insolvenz] Prognose von Insolvenzwahrscheinlichkeiten mit Hilfe logistischer neuronaler Netzwerke. In: Zeitschrift für betriebswirtschaftliche Forschung (50) 1998, S. 892–915.

- Anderson*, Shannon W. und Karen L. *Sedatole*: [Cost Hierarchy] Evidence on the Cost Hierarchy: The Association between Resource Consumption and Production Activities. In: *Journal of Management Accounting Research* (25) 2013, S. 119–141.
- Ando*, Tomohiro und Ker-Chau *Li*: [Model-Averaging] A Model-Averaging Approach for High-Dimensional Regression. In: *Journal of the American Statistical Association* (109) 505/2014, S. 254–265.
- Andrawis*, Robert R., Amir F. *Atiya* und Hisham *El-Shishiny*: [Combinations] Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition. In: *International Journal of Forecasting* (27) 2011, S. 672–688.
- Aust*, René: [Kostenrechnung] Kostenrechnung als unternehmensinterne Dienstleistung. Wiesbaden 1999.
- Azevedo*, Ana und Isabel Rojão *Lourenço*: [KDD] KDD , SEMMA and CRISP-DM: A Parallel Overview. Instituto Politécnico do Porto. Porto 2008. Abrufbar unter: <http://hdl.handle.net/10400.22/136>. Abgerufen am: 23.9.2014.
- Baars*, Henning, Carsten *Felden*, Peter *Gluchowski*, Andreas *Hilbert*, Hans-Georg *Kemper* und Sebastian *Olbrich*: [Gestaltung] Gestaltung der nächsten Inkarnation von Business Intelligence. In: *Wirtschaftsinformatik* (56) 2014, S. 13–19.
- Babad*, Yair M. und Bala V. *Balachandran*: [Optimization] Cost Driver Optimization in Activity-Based Costing. In: *Accounting Review* (68) 1993, S. 563–575.
- Backhaus*, Klaus und Boris *Blechsmidt*: [Fehlende Werte] Fehlende Werte und Datenqualität. In: *Die Betriebswirtschaft* (69) 2009, S. 265–287.
- Backhaus*, Klaus, Bernd *Erichson*, Wulff *Plinke* und Rolf *Weiber*: [Analysemethoden] Multivariate Analysemethoden. 13. Aufl., Berlin, Heidelberg 2011.
- Backhaus*, Klaus, Bernd *Erichson* und Rolf *Weiber*: [Fortgeschrittene Analysemethoden] Fortgeschrittene Multivariate Analysemethoden. Berlin, Heidelberg 2011.
- Baetge*, Jörg: [Überwachung] Überwachung. In: *Vahlens Kompendium der Betriebswirtschaftslehre*. Band 2. Hrsg. von M. Bitz u. a. 3. Aufl., München 1993, S. 175–218.
- Bajramovic*, Ferid, Christian *Gruber* und Bernhard *Sick*: [Comparison] A comparison of first- and second-order training algorithms for dynamic neural networks. In: *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks* (2) 2004, S. 837–842.

- Baker, Alan*: [Simplicity] Simplicity. In: The Stanford Encyclopedia of Philosophy. Winter 2013 Edition. Hrsg. von E. N. Zalta. Stanford (CA) 2013.
- Bange, Carsten*: [Werkzeuge] Werkzeuge für analytische Informationssysteme. In: Analytische Informationssysteme. Hrsg. von P. Chamoni und P. Gluchowski. 4. Aufl., Heidelberg u. a. 2010, S. 131–156.
- Banker, Rajiv D. und Holly Hanson Johnston*: [Airline Industry] An empirical study of cost drivers in the US airline industry. In: Accounting Review (3) 1993, S. 576–601.
- Banker, Rajiv D. und Holly Hanson Johnston*: [Cost Driver Research] Cost and Profit Driver Research. In: Handbook of Management Accounting Research. Band 2. Hrsg. von C. S. Chapman, A. G. Hopwood und M. D. Shields. 2007, S. 531–556.
- Bankhofer, Udo*: [Data Mining] Data Mining und seine betriebswirtschaftliche Relevanz. In: Betriebswirtschaftliche Forschung und Praxis (56) 2004, S. 395–412.
- Barnard, E.*: [Optimization] Optimization for training neural nets. In: IEEE Transactions on Neural Networks (3) 1992, S. 232–240.
- Barrot, Christian*: [Prognosegütemaße] Prognosegütemaße. In: Methodik der empirischen Forschung. Hrsg. von S. Albers u. a. 3. Aufl., Wiesbaden 2009, S. 548–560.
- Barrow, Devon K., Sven F. Crone und Nikolaos Kourentzes*: [Network Ensembles] An evaluation of neural network ensembles and model selection for time series prediction. In: Proceedings of the 2010 International Joint Conference on Neural Networks, IJCNN. Barcelona 2010.
- Barton, Dominic und David Court*: [Advanced Analytics] Making Advanced Analytics Work For You. In: Harvard Business Review (90) 10/2012, S. 79–83.
- Bartz-Beielstein, Thomas und Mike Preuss*: [Tuning] Experimental Analysis of Optimization Algorithms: Tuning and Beyond. In: Theory and Principled Methods for the Design of Metaheuristics. Hrsg. von Y. Borenstein und A. Moraglio. Berlin, Heidelberg 2014, S. 205–245.
- Bates, J. M. und Clive W. J. Granger*: [Combination] The Combination of Forecasts. In: Operational Research Quarterly (20) 1969, S. 451–468.
- Battiti, Roberto*: [Second-Order Methods] First- and Second-Order Methods for Learning: Between Steepest Descent and Newton's Method. In: Neural Computation (4) 1992, S. 141–166.

- Bea, Franz Xaver*: [Führung] Einleitung: Führung. In: Allgemeine Betriebswirtschaftslehre. Band 2: Führung. Hrsg. von F. X. Bea und M. Schweitzer. 10. Aufl., Stuttgart 2011, S. 23–37.
- Bea, Franz Xaver*: [Entscheidungen] Entscheidungen des Unternehmens. In: Allgemeine Betriebswirtschaftslehre. Band 1: Grundfragen. Hrsg. von F. X. Bea und M. Schweitzer. 10. Aufl., Stuttgart 2009, S. 332–437.
- Becker, Jörg* und *Martin Prischmann*: [Kalkulation] Konstruktionsbegleitende Kalkulation mit Neuronalen Netzen. In: Kostenrechnungspraxis (38) 1994, S. 167–171.
- Bellman, Richard*: [Control Processes] Adaptive Control Processes: A Guided Tour. Princeton (NJ) 1961.
- Bendel, Rober B.* und *A. A. Afifi*: [Stopping Rules] Comparison of Stopping Rules in Forward "Stepwise" Regression. In: Journal of the American Statistical Association (72) 357/1977, S. 46–53.
- Birnberg, Jacob G.*: [Framework] A Proposed Framework for Behavioral Accounting Research. In: Behavioral Research in Accounting (23) 2011, S. 1–43.
- Bishop, Christopher M.*: [Neural Networks] Neural Networks for Pattern Recognition. Oxford u. a. 1995.
- Bishop, Christopher M.*: [Machine Learning] Pattern Recognition and Machine Learning. New York 2006.
- Blanco, Antonio, Rafael Pino-Mejías, Juan Lara* und *Salvador Rayo*: [Credit Scoring] Credit scoring models for the microfinance industry using neural networks: Evidence from Peru. In: Expert Systems with Applications (40) 2013, S. 356–364.
- Bode, Jürgen*: [Cost Estimation] Neural networks for cost estimation: simulations and pilot application. In: International Journal of Production Research (38) 2000, S. 1231–1254.
- Boehm, Barry W., Chris Abts, A. Winsor Brown, Sunita Chulani, Bradford K. Clark, Ellis Horowitz, Ray Madachy, Donald Reifer* und *Bert Steece*: [COCOMO II] Software Cost Estimation with COCOMO II. Upper Saddle River (NJ) 2000.
- Bouckaert, Remco R., Eibe Frank, Mark A. Hall, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann* und *Ian H. Witten*: [WEKA] WEKA. Experiences with a Java Open-Source Project. In: Journal of Machine Learning Research (11) 2010, S. 2533–2541.

- Bowden, Gavin J., Graeme C. Dandy und Holger R. Maier:* [Input Determination] Input determination for neural network models in water resources applications. In: *Journal of Hydrology* (301) 2005, S. 75–107.
- Box, George E. P. und George C. Tiao:* [Bayesian Inference] Bayesian Inference in Statistical Analysis. Reading (MA) u. a. 1973.
- Brabazon, Anthony, Jing Dang, Ian Dempsey, Michael O’Neill und David Edelman:* [Finance] Natural Computing in Finance – A Review. In: *Handbook of Natural Computing*. Hrsg. von G. Rozenberg, T. Bäck und J. N. Kok. Berlin, Heidelberg 2012.
- Breiman, Leo:* [Bagging] Bagging Predictors. In: *Machine Learning* (24) 1996, S. 123–140.
- Broadbent, Donald:* [Levels] A Question of Levels: Comment on McClelland und Rumelhart. In: *Journal of Experimental Psychology: General* (114) 1985, S. 189–192.
- Brown, Gavin, Jeremy Wyatt, Rachel Harris und Xin Yao:* [Diversity Creation] Diversity Creation Methods: A Survey and Categorisation. In: *Information Fusion* (6) 2005, S. 5–20.
- Brown, Gavin:* [Diversity] Diversity in Neural Network Ensembles. Diss. University of Birmingham, School of Computer Science. Birmingham 2004. Abrufbar unter: <http://eprints.pascal-network.org/archive/00004765/>. Abgerufen am: 17.11.2014.
- Broyden, C. G.:* [Double-Rank] The Convergence of a Class of Double-Rank Minimization Algorithms. In: *Journal of the Institute of Mathematics and its Applications* (6) 1970, S. 76–90.
- Budäus, Dietrich:* [Aufträge] Kalkulation öffentlicher Aufträge. In: *Handwörterbuch Unternehmensrechnung und Controlling*. Hrsg. von H.-U. Küpper und A. Wagenhofer. 4. Aufl., Stuttgart 2002, S. 907–921.
- Bueno-Crespo, Andrés, Pedro J. García-Laencina und José-Luis Sancho-Gómez:* [Architecture Design] Neural architecture design based on extreme learning machine. In: *Neural Networks* (48) 2013, S. 19–24.
- Buhl, Hans Ulrich, Maximilian Röglinger, Florian Moser und Julia Heidemann:* [Big Data] Big Data. In: *Wirtschaftsinformatik* (55) 2013, S. 63–68.
- Bungenstock, Christian:* [Kostenrechnungssysteme] Entscheidungsorientierte Kostenrechnungssysteme. Eine entwicklungsgeschichtliche Analyse. Wiesbaden 1995.

- Burmester, Lars*: [Business-Intelligence-Systeme] Adaptive Business-Intelligence-Systeme. Wiesbaden 2011.
- Burnham, Kenneth P. und David R. Anderson*: [Model Selection] Model Selection and Multimodel Inference. 2. Aufl., New York 2002.
- Burnham, Kenneth P. und David R. Anderson*: [Multimodel Inference] Multimodel Inference. Understanding AIC and BIC in Model Selection. In: *Sociological Methods and Research* (33) 2004, S. 261–304.
- Burrascano, Pietro*: [Norm Selection] A norm selection criterion for the generalized delta rule. In: *IEEE Transactions on Neural Networks* (2) 1991, S. 125–130.
- van Buuren, Stef und Karin Groothuis-Oudshoorn*: [MICE] MICE: Multivariate imputation by chained equations in R. In: *Journal of Statistical Software* (45) 3/2011, S. 1–67.
- van Buuren, Stef*: [Imputation] Flexible Imputation of Missing Data. Boca Raton, London, New York 2012.
- Caputo, Antonio C. und Pacifico M. Pelagagge*: [Process Vessels] Parametric and neural methods for cost estimation of process vessels. In: *International Journal of Production Economics* (112) 2008, S. 934–954.
- Cavalieri, Sergio, Paolo Maccarrone und Roberto Pinto*: [Automotive] Parametric vs. neural network models for the estimation of production costs: A case study in the automotive industry. In: *International Journal of Production Economics* (91) 2004, S. 165–177.
- Cawley, Gavin C. und Nicola L. C. Talbot*: [Cross-Validation] Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. In: *Neural Networks* (17) 2004, S. 1467–1475.
- Chalmers, David J., Robert M. French und Douglas R. Hofstadter*: [Perception] High-level perception, representation, and analogy. A critique of artificial intelligence methodology. In: *Journal of Experimental and Theoretical Artificial Intelligence* (4) 1992, S. 185–211.
- Chalmers, David J.*: [Connectionism] Connectionism and compositionality. Why Fodor and Pylyshyn were wrong. In: *Philosophical Psychology* (6) 1993, S. 305–319.
- Chalmers, David J.*: [Computation] Subsymbolic Computation and the Chinese Room. In: *The Symbolic and Connectionist Paradigms. Closing the Gap*. Hrsg. von J. Dinsmore. Hillsdale (NJ) 1992, S. 25–48.

- Chamoni, Peter* und *Peter Gluchowski (Hrsg.): [Informationssysteme] Analytische Informationssysteme*. 4. Aufl., Heidelberg u. a. 2010.
- Chapman, Pete, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer* und *Rüdiger Wirth: [CRISP-DM] CRISP-DM 1.0. Step-by-step data mining guide*. 2000. Abrufbar unter: <ftp://ftp.software.ibm.com/software/analytics/spss/support/Modeler/Documentation/14/UserManual/CRISP-DM.pdf>. Abgerufen am: 23.9.2014.
- Chen, Hsinchun, Roger H. L. Chiang* und *Veda C. Storey: [Big Data] Business Intelligence and Analytics: From Big Data to Big Impact*. In: *MIS Quarterly* (36) 2012, S. 1165–1188.
- Chen, Wen-Ching, Lin-Yu Tseng* und *Chih-Sheng Wu: [Evolutionary Training] A unified evolutionary training scheme for single and ensemble of feedforward neural network*. In: *Neurocomputing* (143) 2014, S. 347–361.
- Cheng, Bing* und *DM Titterington: [Statistical Perspective] Neural networks: A review from a statistical perspective*. In: *Statistical Science* (9) 1994, S. 2–30.
- Chorowski, Jan, Jian Wang* und *Jacek M. Zurada: [Review] Review and performance comparison of SVM- and ELM-based classifiers*. In: *Neurocomputing* (128) 2014, S. 507–516.
- Coad, Peter* und *Edward Yourdon: [Object-Oriented Analysis] Object-Oriented Analysis*. Englewood Cliffs (NJ) 1990.
- Coakley, James R.* und *Carol E. Brown: [Modeling Issues] Artificial Neural Networks in Accounting and Finance: Modeling Issues*. In: *International Journal of Intelligent Systems in Accounting, Finance and Management* (9) 2000, S. 119–144.
- Coenenberg, Adolf G.* und *Thomas M. Fischer: [Prozeßkostenrechnung] Prozeßkostenrechnung - Strategische Neuorientierung in der Kostenrechnung*. In: *Die Betriebswirtschaft* (51) 1991, S. 21–36.
- Coenenberg, Adolf Gerhard, Thomas M. Fischer* und *Thomas Günther: [Kostenrechnung] Kostenrechnung und Kostenanalyse*. 8. Aufl., Stuttgart 2012.
- Cooper, Robin* und *Robert S. Kaplan: [Activity-Based Systems] Activity-Based Systems: Measuring the Costs of Resource Usage*. In: *Accounting Horizons* (6) September/1992, S. 1–13.
- Cooper, Robin* und *Robert S. Kaplan: [Measure] Measure Costs Right: Make the Right Decisions*. In: *Harvard Business Review* (66) September-October/1988, S. 96–103.

- Cooper, Robin* und *Robert S. Kaplan*: [Priorities] Profit Priorities from Activity-Based Costing. In: *Harvard Business Review* (69) May-June/1991, S. 130–135.
- Cooper, Robin*: [Activity-Based Costing 1] Activity-Based Costing. Was ist ein Activity-Based Cost-System? In: *Kostenrechnungspraxis* (34) 1990, S. 210–220.
- Cooper, Robin*: [Activity-Based Costing 2] Activity-Based Costing. Wann brauche ich ein Activity-Based Cost-System und welche Kostentreiber sind notwendig? (Teil 2). In: *Kostenrechnungspraxis* (34) 1990, S. 271–279.
- Cooper, Robin*: [Activity-Based Costing 3] Activity-Based Costing. Einführung von Systemen des Activity-Based Costing (Teil 3). In: *Kostenrechnungspraxis* (34) 1990, S. 345–351.
- Corsten, Hans* und *Ralf Gössinger*: [Produktionswirtschaft] *Produktionswirtschaft*. 13. Aufl., München 2012.
- Corsten, Hans* und *Ralf Gössinger*: [Überlegungen] Überlegungen zur Produktionstheorie. Zugleich Stellungnahme und Ergänzung zu dem Beitrag "Neukonzeption der Produktionstheorie" von Harald Dyckhoff in der Zeitschrift für Betriebswirtschaft, 73. Jg. (2003), S. 705-732. In: *Zeitschrift für Betriebswirtschaft* (74) 2004, S. 511–522.
- Corsten, Hans* und *Constantin May (Hrsg.)*: [Neuronale Netze] *Neuronale Netze in der Betriebswirtschaft*. Wiesbaden 1996.
- Corsten, Hans* und *Constantin May*: [Unterstützungspotential] Unterstützungspotential Künstlicher Neuronaler Netze für die Produktionsplanung und -steuerung. In: *Information Management* (10) 1995, S. 44–55. Nachdruck in: *Neuronale Netze in der Betriebswirtschaft*. Hrsg. von H. Corsten und C. May. Wiesbaden 1996, S. 235–257.
- Crone, Sven F.*, *Michèle Hibon* und *Konstantinos Nikolopoulos*: [Forecasting] Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. In: *International Journal of Forecasting* (27) 2011, S. 635–660.
- Crone, Sven F.*, *Stefan Lessmann* und *Robert Stahlbock*: [Preprocessing] The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing. In: *European Journal of Operational Research* (173) 2006, S. 781–800.
- Crone, Sven F.*: [Neuronale Netze] *Neuronale Netze zur Prognose und Disposition im Handel*. Wiesbaden 2010.

- Le Cun, Yann, John S. Denker* und *Sara A. Solla*: [Brain Damage] Optimal Brain Damage. In: *Advances in Neural Information Processing Systems 2*. Hrsg. von *D. S. Touretzky*. San Mateo (CA) 1990, S. 598–605.
- Cybenko, G.*: [Approximation] Approximation by superpositions of a sigmoidal function. In: *Mathematics of Control, Signals and Systems (2)* 1989, S. 303–314.
- Demski, Joel S.* und *Gerald A. Feltham*: [Cost Determination] Cost Determination. A Conceptual Approach. Ames (IA) 1976.
- Demski, Joel S.*: [Managerial Uses] Managerial Uses of Accounting Information. 2. Aufl., New York 2010.
- Demski, Joel S.*: [Cost Deviation] Optimizing the Search for Cost Deviation Sources. In: *Management Science (16)* 1970, S. B486–B494.
- Derigs, Ulrich* und *Gunnar Schirp*: [Genetische Modellierung] Genetische Modellierung von Künstlichen Neuronalen Netzen. In: *OR Spektrum (19)* 1997, S. 285–293.
- Deuffhard, Peter* und *Andreas Hohmann*: [Mathematik 1] Numerische Mathematik. Band 1. Berlin, New York 2008.
- Donate, Juan Peralta, Xiaodong Li, Germán Gutiérrez Sánchez* und *Araceli Sanchis Miguel*: [Evolving] Time series forecasting by evolving artificial neural networks with genetic algorithms, differential evolution and estimation of distribution algorithm. In: *Neural Computing and Applications (22)* 2011, S. 11–20.
- Du, Wei Lin, Luiz Fernando Capretz, Ali Bou Nassif* und *Danny Ho*: [Software Cost] A Hybrid Intelligent Model for Software Cost Estimation. In: *Journal of Computer Science (9)* 2013, S. 1506–1513.
- Duran, Orlando, Juan Maciel* und *Nibaldo Rodriguez*: [Piping Elements] Comparisons between two types of neural networks for manufacturing cost estimation of piping elements. In: *Expert Systems with Applications (39)* 2012, S. 7788–7795.
- Düsing, Roland*: [Anwendungsbereiche] Betriebswirtschaftliche Anwendungsbereiche konnektionistischer Systeme. Hamburg 1997.
- Düsing, Roland*: [Knowledge Discovery] Knowledge Discovery in Databases. In: *Analytische Informationssysteme*. Hrsg. von *P. Chamoni* und *P. Gluchowski*. 4. Aufl., Heidelberg u. a. 2010.
- Dyckhoff, Harald* und *Thomas S. Spengler*: [Produktionswirtschaft] Produktionswirtschaft. 3. Aufl., Berlin, Heidelberg 2010.

- Dyckhoff, Harald*: [Neukonzeption] Neukonzeption der Produktionstheorie. In: Zeitschrift für Betriebswirtschaft (73) 2003, S. 705–732.
- Edwards, Peter J. und Alan F. Murray*: [Model Selection] A study of early stopping and model selection applied to the papermaking industry. In: International Journal of Neural Systems (10) 2000, S. 9–18.
- Ehrlenspiel, Klaus, Alfons Kiewert und Udo Lindemann*: [Entwickeln] Kostengünstig Entwickeln und Konstruieren. Kostenmanagement bei der integrierten Produktentwicklung. 6. Aufl., Berlin, Heidelberg, New York 2007.
- Eisele, Wolfgang und Alois Paul Knobloch*: [Technik] Technik des betrieblichen Rechnungswesens. 8. Aufl., München 2011.
- Esser, Jan*: [Produktionstheorie] Entscheidungsorientierte Erweiterung der Produktionstheorie. Frankfurt/Main 2001.
- Ewert, Ralf und Alfred Wagenhofer*: [Unternehmensrechnung] Interne Unternehmensrechnung. 8. Aufl., Berlin, Heidelberg 2014.
- Ewert, Ralf*: [Entscheidungssteuerung] "Kostenrechnung als Instrument der Entscheidungssteuerung" – Ein Kommentar zum Beitrag von Dieter Pfaff. In: Zeitschrift für Controlling und Management (50) Sonderheft 1/2006, S. 100–111.
- Fahlman, Scott E.*: [Learning Speed] An empirical study of learning speed in back-propagation networks. Technical Report CMU-CS-88-162. Carnegie Mellon University. Pittsburgh (PA) 1988.
- Fan, Wei und Albert Bifet*: [Big Data] Mining big data: current status, and forecast to the future. In: ACM SIGKDD Explorations (14) 2/2013, S. 1–5.
- Fandel, Günter, Andrea Fey, Birgit Heuft und Thomas Pitz*: [Kostenrechnung] Kostenrechnung. 3. Aufl., Berlin, Heidelberg 2009.
- Fandel, Günter*: [Kostentheorie] Produktions- und Kostentheorie. 8. Aufl., Berlin, Heidelberg 2010.
- Fayyad, Usama, Gregory Piatetsky-Shapiro und Padhraic Smyth*: [Knowledge Discovery] From data mining to knowledge discovery in databases. In: AI Magazine (17) 3/1996, S. 37–54.
- Fehrenbacher, Dennis*: [Behavioral Accounting] Behavioral Accounting. In: Controlling (22) 2010, S. 505–506.

- Feldbauer-Durstmüller, Birgit, Ines Wolfsgruber und Christine Duller:* [Kostenmanagement] Kostenmanagement in österreichischen Industriekonzernen. In: Zeitschrift für Controlling und Management (55) 2011, S. 111–116.
- Felden, Carsten:* [Predictive Analytics] Predictive Analytics. In: Analytische Informationssysteme. Hrsg. von P. Chamoni und P. Gluchowski. 4. Aufl., Heidelberg u. a. 2010, S. 307–328.
- Fernando, T. M. K. G., H. R. Maier und G. C. Dandy:* [Partial Mutual Information] Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach. In: Journal of Hydrology (367) 2009, S. 165–176.
- Finnoff, William, Ferdinand Hergert und Hans-Georg Zimmermann:* [Model Selection] Improving Model Selection by Nonconvergent Methods. In: Neural Networks (6) 1993, S. 771–783.
- Fischer, Thomas M.:* [Basis] Die Kostenrechnung als Basis für das Kostenmanagement. In: Kostenmanagement. Hrsg. von K.-P. Franz und P. Kajüter. 2. Aufl., Stuttgart 2002, S. 47–58.
- Fischer, Thomas M.:* [Kostenmanagement] Kostenmanagement strategischer Erfolgsfaktoren. München 1993.
- Fischer, Thomas M.:* [Variantenvielfalt] Variantenvielfalt und Komplexität als betriebliche Kostenbestimmungsfaktoren? In: Kostenrechnungspraxis (37) 1993, S. 27–31.
- Fletcher, R. und C. M. Reeves:* [Conjugate Gradients] Function Minimization by Conjugate Gradients. In: The Computer Journal (7) 1964, S. 149–154.
- Fletcher, R.:* [Variable Metric] A new approach to variable metric algorithms. In: The Computer Journal (13) 1970, S. 317–322.
- Fock, Eric:* [Global Sensitivity] Global Sensitivity Analysis Approach for Input Selection and System Identification Purposes - A New Framework for Feed-forward Neural Networks. In: IEEE Transactions on Neural Networks and Learning Systems (25) 2014, S. 1484–1495.
- Fodor, Jerry A. und Zenon W. Pylyshyn:* [Cognitive Architecture] Connectionism and Cognitive Architecture: A Critical Analysis. In: Cognition (28) 1988, S. 3–71.
- Fodor, Jerry A. und Ernie Lepore:* [Semantic Space] All at Sea in Semantic Space: Churchland on Meaning Similarity. In: The Journal of Philosophy (96) 1999, S. 381–403.

- Fodor, Jerry* und *Brian P. McLaughlin*: [Systematicity] Connectionism and the Problem of Systematicity: Why Smolensky's Solution Doesn't Work. In: *Cognition* (35) 1990, S. 183–204.
- Fogel, David B.*: [Neural Network Selection] An Information Criterion for Optimal Neural Network Selection. In: *IEEE Transactions on Neural Networks* (2) 1991, S. 490–497.
- Foster, George* und *Mahendra Gupta*: [Cost Driver Analysis] Manufacturing Overhead Cost Driver Analysis. In: *Journal of Accounting and Economics* (12) 1990, S. 309–337.
- Fox, S.* und *T. Do*: [Big Data] Getting real about Big Data: applying critical realism to analyse Big Data hype. In: *International Journal of Managing Projects in Business* (6) 2013, S. 739–760.
- Frank, Stefan*: [Gestaltung] Erfolgreiche Gestaltung der Kostenrechnung. Wiesbaden 2000.
- Franks, Bill*: [Taming] Taming the Big Data Tidal Wave. Finding Opportunities in Huge Data Streams with Advanced Analytics. Hoboken (NJ) 2012.
- Franz, Klaus-Peter* und *Peter Kajüter*: [Befunde] Kostenmanagement in Deutschland. Empirische Befunde zur Praxis des Kostenmanagements in deutschen Unternehmen. In: *Kostenmanagement*. Hrsg. von K.-P. Franz und P. Kajüter. 2. Aufl., Stuttgart 2002, S. 569–585.
- Franz, Klaus-Peter* und *Peter Kajüter*: [Proaktives Kostenmanagement] Proaktives Kostenmanagement. In: *Kostenmanagement*. Hrsg. von K.-P. Franz und P. Kajüter. 2. Aufl., Stuttgart 2002, S. 3–32.
- Franz, Klaus-Peter*: [Vollkostenidee] Prozeßkostenrechnung – Renaissance der Vollkostenidee? In: *Die Betriebswirtschaft* (51) 1991, S. 536–539.
- Freiling, Klaus* und *Wolfgang Lück*: [Überwachung] Interne Überwachung und Jahresabschlußprüfung. In: *Zeitschrift für betriebswirtschaftliche Forschung* (38) 1986, S. 996–1006.
- Frese, Erich, Matthias Graumann* und *Ludwig Theuvsen*: [Organisation] Grundlagen der Organisation. 10. Aufl., Berlin, Heidelberg 2012.
- Friedl, Birgit*: [Kostenmanagement] Kostenmanagement. Stuttgart 2009.
- Friedl, Gunther, Kerstin Frömberg, Carola Hammer, Hans-Ulrich Küpper* und *Burkhard Pedell*: [Stand] Stand und Perspektiven der Kostenrechnung in deutschen Großunternehmen. In: *Zeitschrift für Controlling und Management* (53) 2009, S. 111–116.

- Friedl, Gunther, Christian Hofmann und Burkhard Pedell:* [Kostenrechnung] Kostenrechnung. Eine entscheidungsorientierte Einführung. 2. Aufl., München 2013.
- Friedl, Gunther, Hans-Ulrich Küpper und Burkhard Pedell:* [Relevance] Relevance Added: Combining ABC with German Cost Accounting. In: Strategic Finance (86) June/2005, S. 56–61.
- Friedman, Jerome H.:* [Boosting] Greedy Function Approximation: A Gradient Boosting Machine. In: Annals of Statistics (29) 2001, S. 1189–1232.
- Frodl, Andreas:* [Gesundheitsbetrieb] Kostenmanagement und Rechnungswesen im Gesundheitsbetrieb. Wiesbaden 2011.
- Fu, Ai-Min, Xi-Zhao Wang, Yu-Lin He und Lai-Sheng Wang:* [Residence Error] A study on residence error of training an extreme learning machine and its application to evolutionary algorithms. In: Neurocomputing (146) 2014, S. 75–82.
- Fun, Meng-Hock und Martin T. Hagan:* [Levenberg-Marquardt] Levenberg-Marquardt training for modular networks. In: Proceedings of the 1996 International Conference on Neural Networks (1) 1996, S. 468–473.
- Funahashi, Ken-Ichi:* [Mappings] On the approximate realization of continuous mappings by neural networks. In: Neural Networks (2) 1989, S. 183–192.
- Garson, G. David:* [Connection Weights] Interpreting Neural-Network Connection Weights. In: AI Expert (6) 1991, S. 47–51.
- Garson, James:* [Connectionism] Connectionism. In: The Stanford Encyclopedia of Philosophy. Winter 2012 Edition. Hrsg. von E. N. Zalta. Stanford (CA) 2012.
- Gauharou, Barbara:* [Client Services] Activity-based costing at DSL Client Services. In: Management Accounting Quarterly (1) 4/2000, S. 4–12.
- Geman, Stuart, Elie Bienenstock und René Doursat:* [Dilemma] Neural Networks and the Bias/Variance Dilemma. In: Neural Computation (4) 1992, S. 1–58.
- Gerlinger, Annette, Alexander Buresch und Helmut Krcmar:* [IV-Leistungsverrechnung] Prozeßorientierte IV-Leistungsverrechnung – Der Weg zur totalen Transparenz? In: IV-Controlling auf dem Prüfstand. Hrsg. von H. Krcmar, A. Buresch und M. Reb. Wiesbaden 2000, S. 105–134.
- Gevey, Muriel, Ioannis Dimopoulos und Sovan Lek:* [Review and Comparison] Review and comparison of methods to study the contribution of variables in artificial neural network models. In: Ecological Modelling (160) 2003, S. 249–264.

- Gevrey, Muriel, Ioannis Dimopoulos und Sovan Lek*: [Sensitivity Analysis] Two-way interaction of input variables in the sensitivity analysis of neural network models. In: *Ecological Modelling* (195) 2006, S. 43–50.
- Glaser, Horst*: [Entscheidungsrelevanz] Zur Entscheidungsrelevanz prozeßorientierter Stückkosten. In: *Kostenrechnungspraxis* (37) Sonderheft 2/1993, S. 43–47.
- Glaser, Horst*: [Teilabweichungen] Zur Erfassung von Teilabweichungen und Abweichungsüberschneidungen bei der Kostenkontrolle. In: *Kostenrechnungspraxis* (30) 1986, S. 141–148.
- Glaser, Horst*: [Relativität] Zur Relativität von Kostenabweichungen. In: *Betriebswirtschaftliche Forschung und Praxis* (51) 1999, S. 21–32.
- Glaser, Katja*: [Deckungsbeitragsrechnung] Prozeßorientierte Deckungsbeitragsrechnung. München 1998.
- Gluchowski, Peter, Roland Gabriel und Carsten Dittmar*: [Management Support Systeme] Management Support Systeme und Business Intelligence. 2. Aufl., Berlin, Heidelberg 2008.
- Goldfarb, Donald*: [Variable-Metric] A Family of Variable-Metric Methods Derived by Variational Means. In: *Mathematics of Computation* (24) 1970, S. 23–26.
- Goodman, Nelson*: [Fact] Fact, Fiction, and Forecast. 4. Aufl., Cambridge, London 1983.
- Göthlich, Stephan E.*: [Fehlende Daten] Zum Umgang mit fehlenden Daten in großzahligen empirischen Erhebungen. In: *Methodik der empirischen Forschung*. Hrsg. von S. Albers u. a. 3. Aufl., Wiesbaden 2009.
- Graeber, Dietmar und Andreas Kleine*: [Combination] The combination of forecasts in the trading of electricity from renewable energy sources. In: *Journal of Business Economics* (83) 2013, S. 409–435.
- Guo, Zhichao und Robert E. Uhrig*: [Select Inputs] Using Genetic Algorithms to Select Inputs for Neural Networks. In: *International Workshop on Combinations of Genetic Algorithms and Neural Networks 1992 - COGANN-92*. Baltimore (MD) 1992, S. 223–234.
- Güssow, Jan, Andreas Greulich und Robert Ott*: [DRGs] Beurteilung und Einsatz der Prozesskostenrechnung als Antwort der Krankenhäuser auf die Einführung der DRGs. In: *Kostenrechnungspraxis* (46) 2002, S. 179–189.
- Gutenberg, Erich*: [Produktion] Grundlagen der Betriebswirtschaftslehre. Band I: Die Produktion. 24. Aufl., Berlin, Heidelberg, New York 1983.

- Hagan*, Martin T. und Mohammad B. *Menhaj*: [Marquardt] Training feedforward networks with the Marquardt algorithm. In: IEEE Transactions on Neural Networks (5) 1994, S. 989–993.
- Hagiwara*, Katsuyuki, Naohiro *Toda* und Shiro *Usui*: [Applying AIC] On the Problem of Applying AIC to Determine the Structure of a Layered Feed-forward Neural Network. In: Proceedings of the International Joint Conference on Neural Networks, IJCNN (3) 1993, S. 2263–2266.
- Hämmerlin*, Günther und Karl-Heinz *Hoffmann*: [Mathematik] Numerische Mathematik. 4. Aufl., Berlin u. a. 1994.
- Hansen*, Lars Kai und Peter *Salamon*: [Network Ensembles] Neural Network Ensembles. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (12) 1990, S. 993–1001.
- Hanson*, Stephen José und David J. *Burr*: [Minkowski-r Back-Propagation] Minkowski-r Back-Propagation: Learning in Connectionist Models with Non-Euclidian Error Signals. In: Neural Information Processing Systems. Hrsg. von D. Z. Anderson. New York 1988, S. 348–357.
- Hardt*, Rosemarie: [Mercedes-Benz] Von der flexiblen Plankostenrechnung zur Prozeßkostenrechnung: Theoretisches Konzept und empirische Umsetzung am Beispiel des Werkes Hamburg der Mercedes-Benz AG. In: Zukunftsgerichtetes Controlling. Hrsg. von C. Steinle, B. Eggers und D. Lawa. 3. Aufl., Wiesbaden 1998, S. 323–343.
- Hawkins*, Douglas M.: [Outliers] Identification of Outliers. London 1980.
- Hawkins*, Simon, Hongxing *He*, Graham *Williams* und Rohan *Baxter*: [Replicator Neural Networks] Outlier Detection Using Replicator Neural Networks. In: Data Warehousing and Knowledge Discovery. Proceedings of the 4th International Conference, DaWaK 2002. Hrsg. von Y. Kambayashi, W. Winiwarter und M. Arikawa. Berlin, Heidelberg, New York 2002, S. 170–180.
- Heaton*, Jeff: [Encog] Encog 3.3: Development Guide. 2014. Abrufbar unter: www.heatonresearch.com/encog. Abgerufen am: 27.10.2014.
- Hecht-Nielsen*, Robert: [Theory] Theory of the backpropagation neural network. In: Proceedings of the International Joint Conference on Neural Networks, IJCNN (1) 1989, S. 593–605.
- Hedderich*, Jürgen und Lothar *Sachs*: [Statistik] Angewandte Statistik. 14. Aufl., Berlin, Heidelberg 2012.

- van Heeswijk*, Mark, *Yoan Miche*, *Erkki Oja* und *Amaury Lendasse*: [Parallelized ELM] GPU-accelerated and parallelized ELM ensembles for large-scale regression. In: *Neurocomputing* (74) 2011, S. 2430–2437.
- Heinen*, Edmund: [Kostenlehre] Betriebswirtschaftliche Kostenlehre. 6. Aufl., Wiesbaden 1983.
- Heinen*, Edmund: [Kapital] Das Kapital in der betriebswirtschaftlichen Kostentheorie. Wiesbaden 1966.
- Heinen*, Edmund: [Einführung] Einführung in die Betriebswirtschaftslehre. 9. Aufl., Wiesbaden 1985.
- Hempel*, Carl G. und *Paul Oppenheim*: [Studies] Studies in the Logic of Explanation. In: *Philosophy of Science* (15) 1948, S. 135–175.
- Hill*, Tim, *Leorey Marquez*, *Marcus O'Connor* und *William Remus*: [Forecasting] Artificial neural network models for forecasting and decision making. In: *International Journal of Forecasting* (10) 1994, S. 5–15.
- Himme*, Alexander: [Kostenmanagement] Kostenmanagement: Bestandsaufnahme und kritische Beurteilung der empirischen Forschung. In: *Zeitschrift für Betriebswirtschaft* (79) 2009, S. 1051–1098.
- Hing*, M. *Mainegra*, A. *van Harten* und P. C. *Schuur*: [Order Acceptance] Reinforcement learning versus heuristics for order acceptance on a single resource. In: *Journal of Heuristics* (13) 2006, S. 167–187.
- Hinton*, Geoffrey E., *James L. McClelland* und *David E. Rumelhart*: [Representations] Distributed Representations. In: *Parallel Distributed Processing*. Band 1: Foundations. Hrsg. von D. E. Rumelhart, J. L. McClelland und PDP Research Group. Cambridge (MA), London 1986, S. 77–109.
- Hinton*, Geoffrey E. und *Terrence J. Sejnowski* (Hrsg.): [Unsupervised Learning] Unsupervised Learning. Foundations of Neural Computation. Cambridge (MA) 1999.
- Hippert*, Henrique S. und *James W. Taylor*: [Bayesian Techniques] An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting. In: *Neural Networks* (23) 2010, S. 386–395.
- Hodge*, Victoria J. und *Jim Austin*: [Outlier Detection] A Survey of Outlier Detection Methodologies. In: *Artificial Intelligence Review* (22) 2004, S. 85–126.

- Hoffjan, Andreas, Sebastian Lührs und Anja Kolburg:* [Transparency] Cost Transparency in Supply Chains: Demystification of the Cooperation Tenet. In: *Schmalenbach Business Review* (63) 2011, S. 230–251.
- Hoffjan, Andreas und Jeanette Reinemann:* [Absatzsegmentrechnung] Absatzsegmentrechnung. In: *Wirtschaftswissenschaftliches Studium* (29) 2000, S. 129–135.
- Hofmann, Christian und Hans-Ulrich Küpper:* [Neurobiologie] Neurobiologie und Unternehmensrechnung. In: *Zeitschrift für betriebswirtschaftliche Forschung* (63) Sonderheft 63/2011, S. 168–196.
- Holland, John H.:* [Adaptation] *Adaptation in Natural and Artificial Systems*. Oxford 1975.
- Homburg, Carsten und Kirstin Zimmer:* [Auswahl] Optimale Auswahl von Kostentreibern in der Prozeßkostenrechnung. In: *Zeitschrift für betriebswirtschaftliche Forschung* (51) 1999, S. 1042–1055.
- Homburg, Carsten:* [Selection] A note on optimal cost driver selection in ABC. In: *Management Accounting Research* (12) 2001, S. 197–205.
- Homburg, Christian, Jürgen Weber, Jan Thido Karlshaus und René Aust:* [Kundenorientierung] Interne Kundenorientierung der Kostenrechnung? Ergebnisse einer empirischen Untersuchung in deutschen Industrieunternehmen. In: *Die Betriebswirtschaft* (60) 2000, S. 241–256.
- Horngren, Charles T., Srikant M. Datar und Madhav V. Rajan:* [Cost Accounting] *Cost Accounting*. 14. Aufl., Boston u. a. 2012.
- Hornik, Kurt, Maxwell Stinchcombe und Halbert White:* [Approximators] Multilayer Feedforward Networks are Universal Approximators. In: *Neural Networks* (2) 1989, S. 359–366.
- Hornik, Kurt, Maxwell Stinchcombe und Halbert White:* [Approximation] Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks. In: *Neural Networks* (3) 1990, S. 551–560.
- Hornik, Kurt:* [Approximation] Some new results on neural network approximation. In: *Neural Networks* (6) 1993, S. 1069–1072.
- Horváth, Péter und Reinhold Mayer:* [Anmerkungen] Anmerkungen zum Beitrag von A. G. Coenenberg/T.M. Fischer: "Prozeßkostenrechnung – Strategische Neuorientierung in der Kostenrechnung." In: *Die Betriebswirtschaft* (51) 1991, S. 540–542.

- Horváth, Péter* und *Reinhold Mayer*: [Konzeption] Prozeßkostenrechnung – Konzeption und Entwicklung. In: *Kostenrechnungspraxis* (37) Sonderheft 2/1993, S. 15–28.
- Horváth, Péter* und *Reinhold Mayer*: [Prozeßkostenrechnung] Prozeßkostenrechnung. Der neue Weg zu mehr Kostentransparenz und wirkungsvolleren Unternehmensstrategien. In: *Controlling* (1) 1989, S. 214–219.
- Horváth, Péter* und *Reinhold Mayer*: [Prozesskostenrechnung?] Was ist aus der Prozesskostenrechnung geworden? In: *Zeitschrift für Controlling und Management* (55) Sonderheft 2/2011, S. 5–10.
- Hsieh, William W.*: [Nonlinear Principal Component Analysis] Nonlinear Principal Component Analysis of Noisy Data. In: *Neural Networks* (20) 2007, S. 434–443.
- Huang, Guang-Bin, Lei Chen* und *Chee-Kheong Siew*: [Approximation] Universal approximation using incremental constructive feedforward networks with random hidden nodes. In: *IEEE Transactions on Neural Networks* (17) 2006, S. 879–892.
- Huang, Guang-Bin, Qin-Yu Zhu* und *Chee-Kheong Siew*: [Extreme Learning Machine] Extreme Learning Machine: Theory and Applications. In: *Neurocomputing* (70) 2006, S. 489–501.
- Huang, Guang-Bin*: [Reply] Reply to "Comments on 'The Extreme Learning Machine.'" In: *IEEE Transactions on Neural Networks* (19) 2008, S. 1495–1496.
- Huber, Frank, Mark Heitmann* und *Andreas Herrmann*: [Interaktionseffekte] Ansätze zur Kausalmodellierung mit Interaktionseffekten. In: *Die Betriebswirtschaft* (66) 2006, S. 696–710.
- Hummel, Siegfried*: [Relevante Kosten] Relevante Kosten. In: *Handwörterbuch des Rechnungswesens*. Hrsg. von K. Chmielewicz und M. Schweitzer. 3. Aufl., Stuttgart 1993, S. 1713–1717.
- Igel'nik, Boris* und *Yoh Han Pao*: [Stochastic Choice] Stochastic choice of basis functions in adaptive function approximation and the functional-link net. In: *IEEE Transactions on Neural Networks* (6) 1995, S. 1320–1329.
- Islam, Md Monirul* und *Xin Yao*: [Evolving Ensembles] Evolving Artificial Neural Network Ensembles. In: *Computational Intelligence: A Compendium*. Hrsg. von J. Fulcher und L. C. Jain. Berlin, Heidelberg 2008, S. 851–880.
- ISO (Hrsg.)*: [UML Infrastructure] Information technology – Object Management Group Unified Modeling Language (OMG UML), Infrastructure. ISO/IEC 19505-1. 2012.

- ISO (Hrsg.): [UML Superstructure] Information Technology – Object Management Group Unified Modeling Language (OMG UML), Superstructure. ISO/IEC 19505-2. 2012.
- Jacobson, Ivar, Magnus Christerson, Patrik Jonsson und Gunnar Övergaard: [Software Engineering] Object Oriented Software Engineering. A Use Case Driven Approach. Wokingham u. a. 1992.
- Jain, A. S. und S. Meeran: [Job-Shop] Job-shop scheduling using neural networks. In: International Journal of Production Research (36) 1998, S. 1249–1272.
- Janetzke, Philipp und Achim Lewandowski: [Prognose] Der Beitrag der Künstlichen Intelligenz zur betrieblichen Prognose. In: Prognoserechnung. Hrsg. von P. Mertens und S. Rässler. Berlin, Heidelberg 2012, S. 341–382.
- de Jong, Kenneth: [Evolutionary Algorithms] Generalized Evolutionary Algorithms. In: Handbook of Natural Computing. Hrsg. von G. Rozenberg, T. Bäck und J. N. Kok. Berlin, Heidelberg 2012, S. 625–635.
- Kagermann, Henning: [Standardsoftware] Abbildung prozeßorientierter Kostenrechnungssysteme mit Hilfe von Standardsoftware. In: Die Betriebswirtschaft (51) 1991, S. 391–392.
- Kagermann, Henning: [Einbindung] Einbindung der Prozeßkostenrechnung in ein integriertes Standard-Softwaresystem. In: Kostenrechnungspraxis (37) Sonderheft 2/1993, S. 87–92.
- Kajüter, Peter: [Unternehmenspraxis] Kostenmanagement in der deutschen Unternehmenspraxis. In: Zeitschrift für betriebswirtschaftliche Forschung (57) 2005, S. 79–100.
- Kantardzic, Mehmed: [Data Mining] Data Mining. Concepts, Models, Methods, and Algorithms. 2. Aufl., Hoboken (NJ) 2011.
- Kechadi, M. Tahar, Kok Seng Low und G. Goncalves: [Job Shop] Recurrent neural network approach for cyclic job shop scheduling problem. In: Journal of Manufacturing Systems (32) 2013, S. 689–699.
- Kemp, Stanley J., Patricia Zaradic und Frank Hansen: [Importance] An approach for determining relative input parameter importance and significance in artificial neural networks. In: Ecological Modelling (204) 2007, S. 326–334.
- Kemper, Hans-Georg, Henning Baars und Walid Mehanna: [Business Intelligence] Business Intelligence. Grundlagen und praktische Anwendungen. 3. Aufl., Wiesbaden 2010.

- Khan, Muhammad Waseem und Imran Qureshi*: [Software Effort] Neural Network based Software Effort Estimation: A Survey. In: International Journal of Advanced Networking and Applications (5) 2014, S. 1990–1995.
- Kilger, Wolfgang, Jochen Pampel und Kurt Vikas*: [Plankostenrechnung] Flexible Plankostenrechnung und Deckungsbeitragsrechnung. 13. Aufl., Wiesbaden 2012.
- Kim, Gwang-Hee, Sung-Hoon An und Kyung-In Kang*: [Construction Cost] Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning. In: Building and Environment (39) 2004, S. 1235–1242.
- Kim, Gwang-hee, Jae-min Shin, Sangyong Kim und Yoonseok Shin*: [Building Construction] Comparison of School Building Construction Costs Estimation Methods Using Regression Analysis, Neural Network, and Support Vector Machine. In: Journal of Building Construction and Planning Research (1) 2013, S. 1–7.
- Kim, Kyoung-jae und Ingoo Han*: [Activity-Based Costing] Application of a hybrid genetic algorithm and neural network approach in activity-based costing. In: Expert Systems with Applications (24) 2003, S. 73–77.
- Kingston, G. B., H. R. Maier und M. F. Lambert*: [Input Pruning] A statistical input pruning method for artificial neural networks used in environmental modelling. In: Complexity and Integrated Resources Management – Transactions of the 2nd Biennial Meeting of the International Environmental Modelling and Software Society. Hrsg. von C. Pahl-Wostl u. a. Manno 2004, S. 87–92.
- Kloock, Josef und E. Dörner*: [Kostenkontrolle] Kostenkontrolle bei mehrstufigen Produktionsprozessen. In: OR Spektrum (10) 10/1988, S. 129–143.
- Kloock, Josef und Wolfgang Bommers*: [Kostenabweichungsanalyse] Methoden der Kostenabweichungsanalyse. In: Kostenrechnungspraxis (26) 1982, S. 225–237.
- Kloock, Josef und Ulf Schiller*: [Zwecke] Zwecke der Kostenrechnung. In: Die Betriebswirtschaft (59) 1999, S. 135–137.
- Kloock, Josef*: [Erfolgskontrolle] Erfolgskontrolle mit der differenziert-kumulativen Abweichungsanalyse. In: Zeitschrift für Betriebswirtschaft (58) 1988, S. 423–434.
- Kloock, Josef*: [Flexible Prozesskostenrechnung] Flexible Prozesskostenrechnung und Deckungsbeitragsrechnung. In: Kostenrechnungspraxis (37) Sonderheft 2/1993, S. 55–62.

- Kloock, Josef*: [Kommentar] Kommentar zum Beitrag von Horst Glaser. In: Betriebswirtschaftliche Forschung und Praxis (51) 1999, S. 32–34.
- Kloock, Josef*: [Kostenkontrollmanagement] Neuere Entwicklungen des Kostenkontrollmanagements. In: Neuere Entwicklungen im Kostenmanagement. Hrsg. von K. Dellmann und K. P. Franz. Bern, Stuttgart, Wien 1994, S. 607–644.
- Kloock, Josef*: [Rückschritt] Prozesskostenrechnung als Rückschritt und Fortschritt der Kostenrechnung (Teil 1) u. (Teil 2). In: Kostenrechnungspraxis (36) 1992, S. 183–193 u. 237–245.
- Knöbel, Ulf*: [Prozessmanagement] Was kostet ein Kunde? Kundenorientiertes Prozessmanagement. In: Kostenrechnungspraxis (39) 1995, S. 7–13.
- Koch, Helmut*: [Kostenbegriff] Zur Diskussion über den Kostenbegriff. In: Zeitschrift für handelswissenschaftliche Forschung (10) 1958, S. 355–399.
- Kohavi, Ron*: [Cross-Validation] A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI (14) 1995, S. 1137–1145.
- Koller-Meinfelder, Florian*: [Predictive Mean Matching] Analysis of Incomplete Survey Data – Multiple Imputation via Bayesian Bootstrap Predictive Mean Matching. Diss. Fakultät Sozial- und Wirtschaftswissenschaften, Otto-Friedrich-Universität Bamberg. Bamberg 2009.
- Kosiol, Erich*: [Aktionszentrum] Die Unternehmung als wirtschaftliches Aktionszentrum. Reinbek bei Hamburg 1972.
- Kosiol, Erich*: [Erkenntnisgegenstand] Erkenntnisgegenstand und methodologischer Standort der Betriebswirtschaftslehre. In: Zeitschrift für Betriebswirtschaft (31) 1961, S. 129–136.
- Kosiol, Erich*: [Wesensmerkmale] Kritische Analyse der Wesensmerkmale des Kostenbegriffs. In: Betriebsökonomisierung durch Kostenanalyse, Absatzrationalisierung und Nachwuchserziehung. Hrsg. von E. Kosiol und F. Schlieper. Köln, Opladen 1958, S. 7–37.
- Kourentzes, Nikolaos und S. Crone*: [Automatic Modelling] Automatic modelling of neural networks for time series prediction – in search of a uniform methodology across varying time frequencies. In: Proceedings of the 2nd European Symposium on Time Series Prediction. Helsinki 2008.
- Kovacs, Tim*: [Genetics-Based Machine Learning] Genetics-Based Machine Learning. In: Handbook of Natural Computing. Hrsg. von G. Rozenberg, T. Bäck und J. N. Kok. Berlin, Heidelberg 2012, S. 937–986.

- Kramer, Mark A.*: [Autoassociative Neural Networks] Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. In: *AIChE Journal* (37) 1991, S. 233–243.
- Krasnogor, Natalio, Alberto Aragón und Joaquín Pacheco*: [Memetic Algorithms] Memetic Algorithms. In: *Metaheuristic Procedures for Training Neural Networks*. Hrsg. von E. Alba und R. Martí. New York 2006, S. 225–248.
- Krasnogor, Natalio*: [Memetic Algorithms] Memetic Algorithms. In: *Handbook of Natural Computing*. Hrsg. von G. Rozenberg, T. Bäck und J. N. Kok. Berlin, Heidelberg 2012, S. 905–935.
- Kruse, Rudolf, Christian Borgelt, Frank Klawonn, Christian Moewes, Georg Ruß und Matthias Steinbrecher*: [Intelligence] Computational Intelligence. Wiesbaden 2011.
- Kuhlmann, Jan*: [Kreuzvalidierung] Ausgewählte Verfahren der Holdout- und Kreuzvalidierung. In: *Methodik der empirischen Forschung*. Hrsg. von S. Albers u. a. 3. Aufl., Wiesbaden 2009, S. 537–546.
- Küpper, Hans-Ulrich, Gunther Friedl, Christian Hofmann, Yvette Hofmann und Burkhard Pedell*: [Controlling] Controlling. 6. Aufl., Stuttgart 2013.
- Küpper, Hans-Ulrich*: [Bedeutung] Bedeutung und Berechnung von Zinsen in der Kostenrechnung. In: *Risikomanagement und kapitalmarktorientierte Finanzierung*. Hrsg. von K. Schäfer u. a. Frankfurt/Main 2009, S. 337–356.
- Küpper, Hans-Ulrich*: [Berechnung] Bestands- und zahlungsstromorientierte Berechnung von Zinsen in der Kosten- und Leistungsrechnung. In: *Zeitschrift für betriebswirtschaftliche Forschung* (43) 1991, S. 3–20.
- Kurgan, Lukasz A. und Petr Musilek*: [Process Models] A Survey of Knowledge Discovery and Data Mining Process Models. In: *The Knowledge Engineering Review* (21) 2006, S. 1–24.
- Küsters, Ulrich*: [Evaluation] Evaluation, Kombination und Auswahl betriebswirtschaftlicher Prognoseverfahren. In: *Prognoserechnung*. Hrsg. von P. Mertens und S. Rässler. Berlin, Heidelberg 2012, S. 423–467.
- Lackes, Richard und Dagmar Mack*: [Neuronale Netze] Neuronale Netze in der Unternehmensplanung. München 2000.
- Lahres, Bernhard und Gregor Rayman*: [Programmierung] Objektorientierte Programmierung. 2. Aufl., Bonn 2009.
- Lämmel, Uwe und Jürgen Cleve*: [Intelligenz] Künstliche Intelligenz. 4. Aufl., München 2012.

- Laßmann, Gert*: [Kostenrechnung] Die Kosten- und Erlösrechnung als Instrument der Planung und Kontrolle in Industriebetrieben. Düsseldorf 1968.
- Laux, Helmut, Robert M. Gillenkirch und Heike Y. Schenk-Mathes*: [Entscheidungstheorie] Entscheidungstheorie. 8. Aufl., Berlin, Heidelberg 2012.
- Lawson, Tony*: [Reality] Economics and Reality. London 1997.
- Lek, Sovan, Alain Belaud, Philippe Baran, Ioannis Dimopoulos und Marc Delacoste*: [Environmental Variables] Role of some environmental variables in trout abundance models using neural networks. In: Aquatic Living Resources (9) 1996, S. 23–29.
- Leker, Jens und Gerhard Schewe*: [Kreditausfall] Beurteilung des Kreditausfallrisikos im Firmenkundengeschäft der Banken. In: Zeitschrift für betriebswirtschaftliche Forschung (50) 1998, S. 877–891.
- Lengsfeld, Stephan und Ulf Schiller*: [Kostencontrolling] Kostencontrolling. Kostenkontrolle in Teams. In: Zeitschrift für Betriebswirtschaft (71) Ergänzungsheft 2/2001, S. 81–96.
- Lengsfeld, Stephan*: [Kostenkontrolle] Kostenkontrolle und Kostenänderungspotentiale. Wiesbaden 1999.
- Lincoln, William P. und Josef Skrzypek*: [Clustering] Synergy of Clustering Multiple Back Propagation Networks. In: Advances in Neural Information Processing Systems 2. Hrsg. von D. S. Touretzky. San Mateo (CA) 1990, S. 650–657.
- Lingnau, Volker*: [Variantenvielfalt] Kostenwirkungen der Variantenvielfalt. In: Kostenrechnungspraxis (38) 1994, S. 307–315.
- Link, William A. und Richard J. Barker*: [Multimodel Inference] Model Weights and the Foundations of Multimodel Inference. In: Ecology (87) 2006, S. 2626–2635.
- Little, Roderick J. A.*: [Missing-Data Adjustments] Missing-Data Adjustments in Large Surveys. In: Journal of Business and Economic Statistics (6) 1988, S. 287–296.
- Liu, Heping*: [Cost Estimation] Cost Estimation and Sensitivity Analysis on Cost Factors: A Case Study on Taylor Kriging, Regression and Artificial Neural Networks. In: The Engineering Economist (55) 2010, S. 201–224.
- Liu, Yong, Xin Yao und Tetsuya Higuchi*: [Negative Correlation Learning] Evolutionary Ensembles with Negative Correlation Learning. In: IEEE Transactions on Evolutionary Computation (4) 2000, S. 380–387.

- Liu, Yong* und *Xin Yao*: [Negative Correlation] Ensemble Learning via Negative Correlation. In: *Neural Networks* (12) 1999, S. 1399–1404.
- Liu, Yong* und *Xin Yao*: [Simultaneous Training] Simultaneous training of negatively correlated neural networks in an ensemble. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* (29) 1999, S. 716–725.
- Liu, Yong*: [Unbiased Estimate] Unbiased Estimate of Generalization Error and Model Selection in Neural Network. In: *Neural Networks* (8) 1995, S. 215–219.
- Livingstone, David J., Antony Browne, Raymond Crichton, Brian D. Hudson, David C. Whitley* und *Martyn G. Ford*: [Extraction of Information] The Extraction of Information and Knowledge from Trained Neural Networks. In: *Artificial Neural Networks. Methods and Protocols*. Hrsg. von D. J. Livingstone. Totowa (NJ) 2008, S. 231–248.
- Loitlsberger, Erich*: [Grenzbetrachtungen] Grenzbetrachtungen in der Betriebswirtschaftslehre. In: *Handwörterbuch der Betriebswirtschaft*. Band 1. Hrsg. von W. Wittmann u. a. 5. Aufl., Stuttgart 1993, S. 1504–1518.
- Long, Philip M.* und *Rocco A. Servedio*: [Boosters] Random classification noise defeats all convex potential boosters. In: *Machine Learning* (78) 2010, S. 287–304.
- Lucas, Robert E.*: [Critique] Econometric Policy Evaluation: A Critique. In: *Carnegie-Rochester Conference Series on Public Policy* (1) 1976, S. 19–46.
- Lücke, Wolfgang*: [Grenzdenken] Grenzdenken und Marginalanalysen. In: *Handwörterbuch der Produktionswirtschaft*. Hrsg. von W. Kern, H.-H. Schröder und J. Weber. 2. Aufl., Stuttgart 1996, S. 588–601.
- Luhmann, Niklas*: [Zweckbegriff] Zweckbegriff und Systemrationalität. Tübingen 1968.
- Lu, Jinchu* und *Jun S. Liu*: [Model Selection] Model Selection Principles in Misspecified Models. In: *Journal of the Royal Statistical Society: Series B* (76) 2014, S. 141–167.
- Madera, Julio* und *Bernabé Dorronsoro*: [Estimation of Distribution] Estimation of Distribution Algorithms. In: *Metaheuristic Procedures for Training Neural Networks*. Hrsg. von E. Alba und R. Martí. New York 2006, S. 87–108.
- Männel, Wolfgang*: [Grenz- und Residualkosten] Grenz- und Residualkosten. In: *Handwörterbuch des Rechnungswesens*. Hrsg. von K. Chmielewicz und M. Schweitzer. 3. Aufl., Stuttgart 1993, S. 819–824.

- Manyika, James, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh und Angela Hung Byers:* [Big Data] Big data: The next frontier for innovation, competition, and productivity. 2011. Abrufbar unter: http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation. Abgerufen am: 15.9.2014.
- Marbán, Óscar, Gonzalo Mariscal und Javier Segovia:* [Process Model] A Data Mining & Knowledge Discovery Process Model. In: Data Mining and Knowledge Discovery in Real Life Applications. Hrsg. von J. Ponce und A. Karahoca. Wien 2009, S. 1–17.
- Marcus, Gary F.:* [Rethinking] Rethinking Eliminative Connectionism. In: Cognitive Psychology (37) 1998, S. 243–282.
- Marcus, Gary F.:* [Mind] The algebraic mind. Integrating connectionism and cognitive science. Cambridge (MA) 2001.
- Markou, Markos und Sameer Singh:* [Novelty Detection] Novelty detection: a review – part 2: neural network based approaches. In: Signal Processing (83) 2003, S. 2499–2521.
- Marzouk, Mohamed und Ahmed Amin:* [Materials Prices] Predicting Construction Materials Prices Using Fuzzy Logic and Neural Networks. In: Journal of Construction Engineering and Management (139) 2013, S. 1190–1198.
- Massol, Vincent, Jason van Zyl, Brett Porter, John Casey und Carlos Sanchez:* [Maven] Better Builds with Maven. 2008. Abrufbar unter: maestrodev.com/better-build-maven. Abgerufen am: 16.9.2014.
- Matias, Tiago, Francisco Souza, Rui Araújo und Carlos Henggeler Antunes:* [Optimized Extreme Learning Machine] Learning of a single-hidden layer feedforward neural network using an optimized extreme learning machine. In: Neurocomputing (129) 2014, S. 428–436.
- Mayer, Reinhold:* [Prozeßkostenrechnung] Prozeßkostenrechnung. In: Kostenrechnungspraxis (34) 1990, S. 307–312.
- McAfee, Andrew und Erik Brynjolfsson:* [Big Data] Big Data: The Management Revolution. In: Harvard Business Review (90) 10/2012, S. 61–68.
- McCaffrey, James:* [Combinations] Using Combinations to Improve Your Software Test Case Generation. In: MSDN Magazine (19) July/2004. Abrufbar unter: msdn.microsoft.com/en-us/magazine/cc163957. Abgerufen am: 24.9.2014.
- McClelland, James L., David E. Rumelhart und PDP Research Group (Hrsg.):* [Processing 2] Parallel Distributed Processing. Band 2: Psychological and Biological Models. Cambridge (MA), London 1986.

- McCulloch*, Warren S. und *Walter Pitts*: [Calculus] A logical calculus of the ideas immanent in nervous activity. In: *Bulletin of Mathematical Biophysics* (5) 1943, S. 115–133. Nachdruck in: *Bulletin of Mathematical Biology* (52) 1990, S. 99–115.
- Mellerowicz*, Konrad: [Theorie der Kosten] *Kosten und Kostenrechnung*. Band 1: *Theorie der Kosten*. 5. Aufl., Berlin, New York 1973.
- Mengen*, Andreas und *Kerstin Urmersbach*: [Thyssen-Krupp] *Prozesskostenrechnung im Industrieunternehmen*. Ein Konzept zur Weiterentwicklung der Prozesskostenrechnung in einem Unternehmen des Thyssen-Krupp Konzerns. In: *Zeitschrift für Controlling und Management* (50) 2006, S. 218–226.
- Mertens*, Peter und *Marco C. Meier*: [Informationsverarbeitung 2] *Integrierte Informationsverarbeitung 2. Planungs- und Kontrollsysteme in der Industrie*. 10. Aufl., Wiesbaden 2009.
- Mertens*, Peter und *Susanne Rässler*: [Prognoserechnung] *Prognoserechnung – Einführung und Überblick*. In: *Prognoserechnung*. Hrsg. von P. Mertens und S. Rässler. 7. Aufl., Berlin, Heidelberg 2012, S. 3–10.
- Merton*, Robert King: [Theory] *Social Theory and Social Structure*. 3. Aufl., New York 1968.
- Møller*, Martin Fodslette: [Scaled Conjugate Gradient] *A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning*. In: *Neural Networks* (6) 1993, S. 525–533.
- Morgan*, N. und *H. Bourlard*: [Generalization] *Generalization and Parameter Estimation in Feedforward Nets: Some Experiments*. In: *Advances in Neural Information Processing Systems 2*. Hrsg. von D. S. Touretzky. San Mateo (CA) 1990, S. 630–637.
- Moscato*, Pablo: [Memetic Algorithms] *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts. Towards Memetic Algorithms*. C3P Report 826. Caltech Concurrent Computation Program 158-79, California Institute of Technology. Pasadena (CA) 1989.
- Mozer*, Michael C. und *Paul Smolensky*: [Skeletonization] *Skeletonization: A Technique for Trimming the Fat from a Network via Relevance Assessment*. In: *Advances in Neural Information Processing Systems 1*. Hrsg. von D. S. Touretzky. San Mateo (CA) 1989, S. 107–115.
- Mozer*, Michael C. und *Paul Smolensky*: [Relevance] *Using relevance to reduce network size automatically*. In: *Connection Science* (1) 1989, S. 3–16.

- Murata, Noboru, Shuji Yoshizawa und Shun-ichi Amari*: [Network Information Criterion] Network Information Criterion – Determining the Number of Hidden Units for an Artificial Neural Network Model. In: IEEE Transactions on Neural Networks (5) 1994, S. 865–872.
- Nabney, Ian T.*: [Netlab] Netlab. Algorithms for Pattern Recognition. London, Berlin, Heidelberg 2002.
- Neri, Ferrante und Carlos Cotta*: [Memetic Algorithms] Memetic algorithms and memetic computing optimization: A literature review. In: Swarm and Evolutionary Computation (2) 2012, S. 1–14.
- Neugebauer, Harald*: [Versicherungsunternehmen] Kostentheorie und Kostenrechnung für Versicherungsunternehmen. Ein institutionenökonomischer Ansatz. Karlsruhe 1995.
- Neumann, Bruce R., James H. Gerlach, Erwin Moldauer, Michael Finch und Christine Olson*: [IT Activities] Cost Management Using ABC for IT Activities and Services. In: Management Accounting Quarterly (6) 1/2004, S. 29–40.
- Newel, Allen und Herbert A. Simon*: [Symbols] Computer Science as Empirical Inquiry: Symbols and Search. In: Communications of the ACM (19) 1976, S. 113–126.
- Nguyen, Dominique und Bernard Widrow*: [Initial Values] Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In: International Joint Conference on Neural Networks (3) 1990, S. 21–26.
- Nicolai, Alexander und Alfred Kieser*: [Erfolglosigkeit] Trotz eklatanter Erfolglosigkeit: Die Erfolgsfaktorenforschung weiter auf Erfolgskurs. In: Die Betriebswirtschaft (62) 2002, S. 579–596.
- O'Brien, Robert M.*: [Variance Inflation Factors] A Caution Regarding Rules of Thumb for Variance Inflation Factors. In: Quality and Quantity (41) 2007, S. 673–690.
- O'Brien, Tim, Manfred Moser, John Casey, Brian Fox, Jason van Zyl, Eric Redmond und Larry Shatzer*: [Maven] Maven: The Complete Reference. Sebastopol (CA) 2008. Abrufbar unter: sonatype.com/resources/books/maven-the-complete-reference. Abgerufen am: 16.9.2014.
- Obermaier, Robert*: [Entscheidungsrelevanz] Zur Entscheidungsrelevanz der Grenzplankostenrechnung: Relevance lost – or ever had? In: Zeitschrift für Planung und Unternehmenssteuerung (17) 2006, S. 177–199.

- Object Management Group*: [UML] OMG Unified Modeling Language Specification. Version 1.3. 2000. Abrufbar unter: www.omg.org/spec/UML/1.3/. Abgerufen am: 16.9.2014.
- Olden, Julian D. und Donald A. Jackson*: [Illuminating] Illuminating the "black box". A randomization approach for understanding variable contributions in artificial neural networks. In: *Ecological Modelling* (154) 2002, S. 135–150.
- Olden, Julian D. und Donald A. Jackson*: [Torturing Data] Torturing Data for the Sake of Generality: How Valid Are Our Regression Models? In: *Écoscience* (7) 2000, S. 501–510.
- Olden, Julian D., Michael K. Joy und Russell G. Death*: [Comparison] An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. In: *Ecological Modelling* (178) 2004, S. 389–397.
- Opitz, David W. und Jude W. Shavlik*: [Network Ensemble] Actively Searching for an Effective Neural Network Ensemble. In: *Connection Science* (8) 1996, S. 337–354.
- Ortmann, Günther*: [Formen] Formen der Produktion. Organisation und Rekursivität. Opladen 1995.
- Ossadnik, Wolfgang, David Barklage und Dirk Wilmsmann*: [Kostenkontrolle] Kostenkontrolle und Verhaltenssteuerung: Eine Evaluierung differenziert-kumulativer Methoden der Kostenabweichungsanalyse. In: *Jahrbuch für Controlling und Rechnungswesen* 2007. Hrsg. von G. Seicht. Wien 2007, S. 155–174.
- Ossadnik, Wolfgang und Stefan Maus*: [Kostenabweichungsanalyse] Kostenabweichungsanalyse als Instrument des operativen Controlling. In: *Wirtschaftswissenschaftliches Studium* (23) 1994, S. 446–451.
- Ozan, T. und T. Dyckman*: [Cost Variances] A Normative Model for Investigation Decisions Involving Multiorigin Cost Variances. In: *Journal of Accounting Research* (9) 1971, S. 88–115.
- Paliwal, Mukta und Usha A. Kumar*: [Contribution of Variables] Assessing the contribution of variables in feed forward neural network. In: *Applied Soft Computing* (11) 2011, S. 3690–3696.
- Paliwal, Mukta und Usha A. Kumar*: [Applications] Neural networks and Statistical Techniques: A Review of Applications. In: *Expert Systems with Applications* (36) 2009, S. 2–17.

- Perlich, Claudia*: [Learning Curves] Learning Curves in Machine Learning. In: Encyclopedia of Machine Learning. Hrsg. von C. Sammut und G. I. Webb. New York 2011, S. 577–580.
- Pfaff, Dieter, Jürgen Weber* und *Barbara E. Weißenberger*: [Relevance] Relevance Lost and Found. Kostenrechnung als Steuerungsinstrument und Sprache. In: Die Betriebswirtschaft (59) 1999, S. 138–143.
- Pfaff, Dieter* und *Jürgen Weber*: [Zweck] Zweck der Kostenrechnung? In: Die Betriebswirtschaft (58) 1998, S. 151–165.
- Pfaff, Dieter*: [Entscheidungssteuerung] Kostenrechnung als Instrument der Entscheidungssteuerung – Chancen und Probleme. In: Kostenrechnungspraxis (40) 1996, S. 151–156. Nachdruck in: Zeitschrift für Controlling und Management (50) Sonderheft 1/2006, S. 94–99.
- Piatetsky-Shapiro, Gregory*: [Knowledge Discovery] Knowledge Discovery in Real Databases: A Report on the IJCAI-89 Workshop. In: AI Magazine (11) 5/1990, S. 68–70.
- Pinker, Steven* und *Alan Prince*: [Language] On Language and Connectionism: Analysis of a Parallel Distributed Processing Model of Language Acquisition. In: Cognition (28) 1988, S. 73–193.
- Pohlen, Knut*: [Kostenkontrolle] Kostenkontrolle unter Risiko und besonderer Berücksichtigung kostentheoretischer Ansätze. Diss. TU Berlin. Berlin 1998.
- Pope, P. T.* und *J. T. Webster*: [F-Statistic] The Use of an F-Statistic in Stepwise Regression Procedures. In: Technometrics (14) 1972, S. 327–340.
- Praynlin, E.* und *P. Latha*: [Software Effort] Performance Analysis of Software Effort Estimation Models Using Neural Networks. In: International Journal of Information Technology and Computer Science (5) 9/2013, S. 101–107.
- Prechelt, Lutz*: [Early Stopping] Automatic Early Stopping Using Cross Validation: Quantifying the Criteria. In: Neural Networks (11) 1998, S. 761–767.
- Pressmar, Dieter B.*: [Kosten- und Leistungsanalyse] Kosten- und Leistungsanalyse im Industriebetrieb. Wiesbaden 1971.
- Putnam, Hilary*: [Minds] Minds and Machines. In: Dimensions of Mind. Hrsg. von S. Hook. New York 1960, S. 148–179.
- Ramsey, William*: [Representations] Do Connectionist Representations Earn Their Explanatory Keep? In: Mind and Language (12) 1997, S. 34–66.

- Reed, Russell*: [Pruning] Pruning Algorithms. A Survey. In: IEEE Transactions on Neural Networks (4) 1993, S. 740–747.
- Rehkugler, Heinz* und *Hans-Georg Zimmermann* (Hrsg.): [Ökonomie] Neuronale Netze in der Ökonomie. München 1994.
- Rehkugler, Heinz*: [Neuronale Netze] Neuronale Netze in der Ökonomie. In: Wirtschaftswissenschaftliches Studium (25) 1996, S. 572–576.
- Rendenbach, Hans-Georg*: [Assekuranz] Prozesskostenmanagement in der Assekuranz. In: Kostenmanagement. Hrsg. von K.-P. Franz und P. Kajüter. 2. Aufl., Stuttgart 2002, S. 279–296.
- Richter, Dietmar* und *Joachim Neubert*: [Taylor-Reihe] Die Taylor-Reihe als allgemeine Methode zur Behandlung der Abweichungsaufspaltung und ihre Nutzung in der betrieblichen Analysetätigkeit. In: Methoden zur Verbesserung von Leitung und Planung in der Bergbauindustrie. Freiburger Forschungshefte D 129. Hrsg. von der Bergakademie Freiberg. Leipzig 1980, S. 73–88.
- Riebel, Paul* und *Werner Sinzig*: [Datenbank] Zur Realisierung der Einzelkosten- und Deckungsbeitragsrechnung mit einer relationalen Datenbank. In: Zeitschrift für betriebswirtschaftliche Forschung (33) 1981, S. 457–489.
- Riebel, Paul*: [Rechnen] Das Rechnen mit Einzelkosten und Deckungsbeiträgen. In: Zeitschrift für handelswissenschaftliche Forschung (11) 1959, S. 213–238.
- Riebel, Paul*: [Einzelkostenrechnung] Einzelkosten- und Deckungsbeitragsrechnung. 7. Aufl., Wiesbaden 1994.
- Riedmiller, Martin* und *Heinrich Braun*: [RPROP] A Direct Adaptive Method for Faster Backpropagation Learning. The RPROP Algorithm. In: Proceedings of the IEEE International Conference on Neural Networks (1) 1993, S. 586–591.
- Rieper, Bernd*: [Fundierung] Produktions- und kapitaltheoretische Fundierung betrieblicher Kostenfunktionen – eine Erörterung am Beispiel einer linear-limitationalen Produktionssituation. In: Zeitschrift für Betriebswirtschaft (73) 2003, S. 473–497.
- Ripley, Brian D.*: [Classification] Neural Networks and Related Methods for Classification. In: Journal of the Royal Statistical Society, B (56) 1994, S. 409–456.
- Ripley, Brian D.*: [Pattern Recognition] Pattern Recognition and Neural Networks. Cambridge 1996.
- Röögli, Franz* und *Peter Bunce*: [Doppelnatur] Gefährliche Doppelnatur von Budgets und ihre Überwindung. In: Zeitschrift für Controlling und Management (56) Sonderheft 2/2012, S. 23–27.

- Rosenberg, Otto*: [Komplexitätsmanagement] Kostensenkung durch Komplexitätsmanagement. In: Kostenmanagement. Hrsg. von K.-P. Franz und P. Kajüter. 2. Aufl., Stuttgart 2002, S. 225–246.
- Rosenblatt, F.*: [Perceptron] The perceptron: a probabilistic model for information storage and organization in the brain. In: Psychological Review (65) 1958, S. 386–408.
- Rubin, Donald B.*: [Missing Data] Inference and Missing Data. In: Biometrika (63) 1976, S. 581–592.
- Rubin, Donald B.*: [Imputation] Multiple Imputation for Nonresponse in Surveys. New York u. a. 1987.
- Rubin, Donald B.*: [Statistical Matching] Statistical Matching Using File Concatenation With Adjusted Weights and Multiple Imputations. In: Journal of Business and Economic Statistics (4) 1986, S. 87–94.
- Rudolph, Günter*: [Convergence] Stochastic Convergence. In: Handbook of Natural Computing. Hrsg. von G. Rozenberg, T. Bäck und J. N. Kok. Berlin, Heidelberg 2012, S. 847–869.
- Rumelhart, David E., Geoffrey E. Hinton und James L. McClelland*: [Framework] A General Framework for Parallel Distributed Processing. In: Parallel Distributed Processing. Band 1: Foundations. Hrsg. von D. E. Rumelhart, J. L. McClelland und PDP Research Group. Cambridge (MA), London 1986, S. 45–76.
- Rumelhart, David E., Geoffrey E. Hinton und Ronald J. Williams*: [Error Propagation] Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing. Band 1: Foundations. Hrsg. von D. E. Rumelhart, J. L. McClelland und PDP Research Group. Cambridge (MA), London 1986, S. 318–362.
- Rumelhart, David E., Geoffrey E. Hinton und Ronald J. Williams*: [Back-propagating Errors] Learning representations by back-propagating errors. In: Nature (323) 1986, S. 533–536.
- Rummel, Kurt*: [Kostenrechnung] Einheitliche Kostenrechnung. 3. Aufl., Düsseldorf 1949.
- Sahl, Niels*: [Integration] Integration der Prozeßkostenrechnung in die Planungs- und Abrechnungssystematik der Grenzplankostenrechnung. In: Kostenrechnungspraxis (38) Sonderheft 1/1994, S. 41–49.
- Saint-Mont, Uwe*: [Statistik] Statistik im Forschungsprozess. Berlin, Heidelberg 2011.

- Sammur, Claude und Geoffrey I. Webb (Hrsg.):* [Machine Learning] Encyclopedia of Machine Learning. New York 2011.
- Sarle, Warren S.:* [Importance] How to measure importance of inputs? SAS Institute. Cary (NC) 2000. Abrufbar unter: <ftp://ftp.sas.com/pub/neural/importance.html>. Abgerufen am: 20.10.2013.
- SAS Institute Inc.:* [Enterprise Miner] Data Mining Using SAS Enterprise Miner. A Case Study Approach. 3. Aufl., Cary (NC) 2013.
- Säuberlich, Frank:* [Web Mining] Web Mining: Effektives Marketing im Internet. In: Neuronale Netze im Marketing-Management. Hrsg. von K.-P. Wiedmann und F. Buckler. Wiesbaden 2001, S. 103–121.
- Scardi, Michele und Lawrence W. Harding:* [Phytoplankton] Developing an empirical model of phytoplankton primary production: a neural network case study. In: Ecological Modelling (120) 1999, S. 213–223.
- Schabach, Robert und Helmut Werner:* [Mathematik] Numerische Mathematik. 4. Aufl., Berlin u. a. 1993.
- Schafer, Joseph L. und John W. Graham:* [Missing Data] Missing Data: Our View of the State of the Art. In: Psychological Methods (7) 2002, S. 147–177.
- Schaffer, J. David, Darrell Whitley und Larry J. Eshelman:* [Combinations] Combinations of Genetic Algorithms and Neural Networks: A Survey of the State of the Art. In: International Workshop on Combinations of Genetic Algorithms and Neural Networks 1992, COGANN-92. Baltimore (MD) 1992, S. 1–37.
- Schäffer, Utz und Stephan Kramer:* [Budgetverhandlungen] Experimentelle Erkenntnisse zu menschlichem Verhalten in Budgetverhandlungen. In: Zeitschrift für Controlling und Management (53) 2009, S. 254–256.
- Schäffer, Utz, Jürgen Weber und Matthias Mahlendorf:* [Trends] Trends in Kostenrechnung und Incentivierung. Eine Studie des WHU-Controllerpanels (2012). Vallendar 2012.
- Schäffer, Utz:* [Kontrolle] Kontrolle als Lernprozess. Wiesbaden 2001.
- Schapire, Robert E. und Yoram Singer:* [Boosting] Improved Boosting Algorithms Using Confidence-rated Predictions. In: Machine Learning (37) 1999, S. 297–336.
- Schermann, Michael, Helmut Krcmar, Holmer Hensen, Volker Markl, Christoph Buchmüller, Till Bitter und Thomas Hoeren:* [Big Data] Big Data. Eine interdisziplinäre Chance für die Wirtschaftsinformatik. In: Wirtschaftsinformatik (56) 2014, S. 281–287.

- Schierenbeck, Henner* und *Claudia B. Wöhle*: [Grundzüge] Grundzüge der Betriebswirtschaftslehre. 18. Aufl., München 2012.
- Schildbach, Thomas* und *Carsten Homburg*: [Leistungsrechnung] Kosten- und Leistungsrechnung. 10. Aufl., Stuttgart 2009.
- Schlagheck, Bernhard*: [Referenzmodelle] Objektorientierte Referenzmodelle für das Prozess- und Projektcontrolling. Wiesbaden 2000.
- Schmalenbach, Eugen*: [Kostenrechnung] Kostenrechnung und Preispolitik. 8. Aufl., Köln, Opladen 1963.
- Schneeweiß, Christoph*: [Dienstleistungsproduktion] Zur Erweiterung der Produktionstheorie auf die Dienstleistungsproduktion. In: Aktuelle Aspekte des Controllings. Hrsg. von V. Lingnau und H. Schmitz. Heidelberg 2002, S. 199–224.
- Schneider, Dieter*: [Warnung] Eine Warnung vor Frühwarnsystemen. In: Der Betrieb (38) 1985, S. 1489–1494.
- Schneider, Tanja*, *Martin Nelke* und *Marco Poloni*: [Kundenbindung] Kundenbindung und Abwanderungsprävention. In: Neuronale Netze im Marketing-Management. Hrsg. von K.-P. Wiedmann und F. Buckler. Wiesbaden 2001, S. 133–163.
- Schreyögg, Georg*: [Organisation] Organisation. 5. Aufl., Wiesbaden 2008.
- Schwarz, Gideon*: [Dimensions] Estimating the Dimensions of a Model. In: Annals of Statistics (6) 1978, S. 461–464.
- Schwarz, Hans Rudolf* und *Norbert Köckler*: [Mathematik] Numerische Mathematik. 8. Aufl., Wiesbaden 2011.
- Schweitzer, Marcell* und *Hans-Ulrich Küpper*: [Kostentheorie] Produktions- und Kostentheorie. 2. Aufl., Wiesbaden 1997.
- Schweitzer, Marcell* und *Hans-Ulrich Küpper*: [Systeme] Systeme der Kosten- und Erlösrechnung. 10. Aufl., München 2011.
- Schweitzer, Marcell*: [Gegenstand] Gegenstand und Methoden der Betriebswirtschaftslehre. In: Allgemeine Betriebswirtschaftslehre. Band 1: Grundfragen. Hrsg. von F. X. Bea und M. Schweitzer. 10. Aufl., Stuttgart 2009, S. 23–80.
- Schweitzer, Marcell*: [Planung] Planung und Steuerung. In: Allgemeine Betriebswirtschaftslehre. Band 2: Führung. Hrsg. von F. X. Bea und M. Schweitzer. 10. Aufl., Konstanz, München 2011, S. 38–177.

- Seemann, Jochen* und *Jürgen Wolff von Gudenberg*: [Software-Entwurf] Software-Entwurf mit UML 2. 2. Aufl., Berlin, Heidelberg 2006.
- Selke, Stephan*: [Gestaltung] Die Gestaltung der Kosten- und Leistungsrechnung unter besonderer Berücksichtigung formaler Organisationsstrukturen. Frankfurt/Main 1997.
- Seo, Kwang-Kyu, Ji-Hyung Park, Dong-Sik Jang* und *David Wallace*: [Life Cycle Cost] Prediction of the life cycle cost using statistical and artificial neural network methods in conceptual product design. In: International Journal of Computer Integrated Manufacturing (15) 2002, S. 541–554.
- Sexton, Randall S., Robert E. Dorsey* und *John D. Johnson*: [Genetic Algorithm] Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing. In: European Journal of Operational Research (114) 1999, S. 589–601.
- Shanker, M., M. Y. Hu* und *M. S. Hung*: [Standardization] Effect of Data Standardization on Neural Network Training. In: Omega (24) 1996, S. 385–397.
- Shanno, D. F.*: [Quasi-Newton] Conditioning of Quasi-Newton Methods for Function Minimization. In: Mathematics of Computation (24) 1970, S. 647–656.
- Sharkey, Amanda J. C.*: [Combining] On Combining Artificial Neural Nets. In: Connection Science (8) 1996, S. 299–314.
- Shearer, Colin*: [CRISP-DM] The CRISP-DM model. The new blueprint for data mining. In: Journal of Data Warehousing (5) 4/2000, S. 13–22.
- Shinozaki, Nobuo, Masaaki Sibuya* und *Kunio Tanabe*: [Moore-Penrose direct] Numerical algorithms for the Moore-Penrose inverse of a matrix: direct methods. In: Annals of the Institute of Statistical Mathematics (24) 1972, S. 193–203.
- Shinozaki, Nobuo, Masaaki Sibuya* und *Kunio Tanabe*: [Moore-Penrose iterative] Numerical algorithms for the Moore-Penrose inverse of a matrix: iterative methods. In: Annals of the Institute of Statistical Mathematics (24) 1972, S. 621–629.
- Shtub, Avraham* und *Ronen Versano*: [Steel Pipe Bending] Estimating the cost of steel pipe bending, a comparison between neural networks and regression analysis. In: International Journal of Production Economics (62) 1999, S. 201–207.
- Siemoneit, Oliver*: [Wissenschaftstheorie] Eine Wissenschaftstheorie der Betriebswirtschaftslehre. Diss. Philosophisch-Historische Fakultät, Universität Stuttgart. Stuttgart 2010.

- Simon*, Foo Yoon-Pin und Y. *Takefuji*: [Jop-Shop] Stochastic neural networks for solving job-shop scheduling. In: IEEE International Conference on Neural Networks 1988. Band 2. San Diego (CA) 1988, S. 275–290.
- Smith*, Alice E. und Anthony K. *Mason*: [Cost Estimation] Cost Estimation Predictive Modeling: Regression Versus Neural Network. In: The Engineering Economist (42) 1997, S. 137–161.
- Smith Bamber*, Linda und K. E. *Hughes II*: [Buckeye] Activity-based costing in the service sector: The Buckeye National Bank. In: Issues in Accounting Education (16) 2001, S. 381–408.
- Smolensky*, Paul: [Connectionism] On the Proper Treatment of Connectionism. Report CU-CS-359-87. Department of Computer Science, University of Colorado at Boulder. Boulder 1987.
- Soares*, Symone, Carlos Henggeler *Antunes* und Rui *Araújo*: [Comparison] Comparison of a genetic algorithm and simulated annealing for automatic neural network ensemble development. In: Neurocomputing (121) 2013, S. 498–511.
- Soria*, Emilio, José David *Martin* und Paulo J. G. *Lisboa*: [Training Methods] Classical Training Methods. In: Metaheuristic Procedures for Training Neural Networks. Hrsg. von E. Alba und R. Martí. New York 2006, S. 7–36.
- Streitferdt*, Lothar: [Abweichungsauswertung] Entscheidungsregeln zur Abweichungsauswertung. Würzburg, Wien 1983.
- Strohhecker*, Jürgen: [Simulation] System- und objektorientierte Simulation betriebswirtschaftlicher Entscheidungen. Berlin 1998.
- Sun*, Zhan-Li, Tsan-Ming *Choi*, Kin-Fan *Au* und Yong *Yu*: [Sales Forecasting] Sales forecasting using extreme learning machine with applications in fashion retailing. In: Decision Support Systems (46) 2008, S. 411–419.
- Sung*, A. H.: [Importance] Ranking importance of input parameters of neural networks. In: Expert Systems with Applications (15) 1998, S. 405–411.
- Sutton*, Richard S. und Andrew G. *Barto*: [Reinforcement Learning] Reinforcement Learning. An Introduction. Cambridge (MA) 1998.
- Szepesvári*, Csaba: [Reinforcement Learning] Algorithms for Reinforcement Learning. San Rafael (CA) 2010.
- Tang*, Ke, Minlong *Lin*, Fernanda L. *Minku* und Xin *Yao*: [Negative Correlation Learning] Selective negative correlation learning approach to incremental learning. In: Neurocomputing (72) 2009, S. 2796–2805.

- Teytaud, Olivier und Emmanuel Vazquez*: [Optimal Search Algorithm] Designing an Optimal Search Algorithm with Respect to Prior Information. In: Theory and Principled Methods for the Design of Metaheuristics. Natural Computing Series. Hrsg. von Y. Borenstein und A. Moraglio. Berlin, Heidelberg 2014, S. 111–128.
- Thagard, Paul*: [Cognitive Science] Cognitive Science. In: The Stanford Encyclopedia of Philosophy. Hrsg. von E. N. Zalta. Fall 2012 Edition, Stanford (CA) 2012.
- Torrecilla, José S., Francisco Rodríguez, José L. Bravo, Gadi Rothenberg, Kenneth R. Seddon und Ignacio López-Martin*: [Optimising] Optimising an artificial neural network for predicting the melting point of ionic liquids. In: Physical Chemistry Chemical Physics (10) 2008, S. 5826–5831.
- Troßmann, Ernst, Alexander Baumeister und Clemens Werkmeister*: [Fallstudien] Management-Fallstudien im Controlling. 3. Aufl., München 2013.
- Troßmann, Ernst und Alexander Baumeister*: [Rechnungswesen] Internes Rechnungswesen. Kostenrechnung als Standardinstrument im Controlling. München 2015.
- Troßmann, Ernst*: [Controlling] Controlling als Führungsfunktion. Eine Einführung in die Mechanismen betrieblicher Koordination. München 2013.
- Troßmann, Ernst*: [Plankostenrechnung] Flexible Plankostenrechnung nach Kilger. In: Handbuch Kostenrechnung. Hrsg. von W. Männel. Wiesbaden 1992, S. 226–246.
- Troßmann, Ernst*: [Gemeinkosten-Budgetierung] Gemeinkosten-Budgetierung als Controlling-Instrument in Bank und Versicherung. In: Controlling. Grundlagen, Informationssysteme, Anwendungen. Hrsg. von K. Spremann und E. Zur. Wiesbaden 1992, S. 511–540.
- Troßmann, Ernst*: [Rechnungswesen] Internes Rechnungswesen. In: Betriebswirtschaftslehre. Band 1. Hrsg. von H. Corsten und M. Reiß. 4. Aufl., München 2008, S. 99–220.
- Troßmann, Ernst*: [Kennzahlen] Kennzahlen als Instrument des Produktionscontrolling. In: Handbuch Produktionsmanagement. Hrsg. von H. Corsten. Wiesbaden 1994, S. 517–536.
- Troßmann, Ernst*: [Kostenfunktionen] Kostenfunktionen. In: Handwörterbuch des Rechnungswesens. Hrsg. von K. Chmielewicz und M. Schweitzer. 3. Aufl., Stuttgart 1993, S. 1194–1208.

- Troßmann, Ernst*: [Kostentheorie] Kostentheorie und Kostenrechnung. In: Handwörterbuch der Betriebswirtschaft. Band 2. Hrsg. von W. Wittmann u. a. 5. Aufl., Stuttgart 1993, S. 2385–2401.
- Troßmann, Ernst*: [Potentialgestaltung] Planungs- und Steuerungssysteme für die Potentialgestaltung. In: Einführung in das Produktionscontrolling. Hrsg. von H. Corsten und B. Friedl. München 1999, S. 107–139.
- Turban, Efraim, Ramesh Sharda und Dursun Delen*: [Decision Support] Decision Support and Business Intelligence Systems. 9. Aufl., Upper Saddle River (NJ) 2011.
- Turing, Alan*: [Machinery] Computing Machinery and Intelligence. In: Mind (59) 1950, S. 433–460.
- Türk, Klaus*: [Organisationsforschung] Neuere Entwicklungen in der Organisationsforschung. Stuttgart 1989.
- Uebernicker, Falk, Jochen Scheeg und Walter Brenner*: [T-Systems] Produkt- und serviceorientiertes IT-Controlling am Fallbeispiel T-Systems. In: HMD Praxis der Wirtschaftsinformatik (44) 2/2007, S. 39–49.
- Ueckerdt, Birgit und Hans-J. Lenz*: [Abweichungsanalyse] Zur Abweichungsanalyse im operativen, modellgestützten Controlling. In: OR Spektrum (19) 1997, S. 273–283.
- Urkmez, Sualp Ömer, Erdem Bilgili, Reza Ziarati und David Stockton*: [Ship Building] Application of Novel Artificial Intelligent Techniques in Ship Building Using Activity Based Costing and Neural Networks. In: Proceedings of the International Maritime Lecturers' Association 16th Conference. Izmir 2008, S. 377–385.
- Venkatesh, Kamini, Vadlamani Ravi, Anita Prinzie und Dirk van den Poel*: [Cash Demand] Cash demand forecasting in ATMs by clustering and neural networks. In: European Journal of Operational Research (232) 2014, S. 383–392.
- Verlinden, Bart, J. R. Duflou, P. Collin und D. Cattrysse*: [Sheet Metal Parts] Cost estimation for sheet metal parts using multiple regression and artificial neural networks: A case study. In: International Journal of Production Economics (111) 2008, S. 484–492.
- Wagenhofer, Alfred*: [Agency] Kostenrechnung und Agency Theorie. In: Zur Neuausrichtung der Kostenrechnung. Hrsg. von J. Weber. Stuttgart 1993, S. 161–185.

- Wagenhofer, Alfred*: [Zwecke] Zwecke der Kostenrechnung. In: Die Betriebswirtschaft (59) 1999, S. 133–135.
- Wall, Friederike und Dirk Kießling*: [Budgetinformationen] Verhaltensorientiertes Controlling und Budgetinformationen – Praktische Erfahrungen und ausgewählte Forschungsergebnisse. In: Zeitschrift für Controlling und Management (52) Sonderheft 1/2008, S. 74–80.
- Wang, H. S., Y. N. Wang und Y. C. Wang*: [Injection Molding] Cost estimation of plastic injection molding parts through integration of PSO and BP neural network. In: Expert Systems with Applications (40) 2013, S. 418–428.
- Wang, Lin, Yi Zeng und Tao Chen*: [Differential Evolution] Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. In: Expert Systems with Applications (42) 2015, S. 855–863.
- Wang, Lipo P. und Chunru R. Wan*: [Comments] Comments on "The extreme learning machine". In: IEEE Transactions on Neural Networks (19) 2008, S. 1494.
- Wang, Q., D. J. Stockton und P. Baguley*: [Cost Modelling] Process Cost Modelling Using Neural Networks. In: International Journal of Production Research (38) 2000, S. 3811–3821.
- Watts, Michael J. und S. P. Worner*: [Relative Contribution] Using artificial neural networks to determine the relative contribution of abiotic factors influencing the establishment of insect pest species. In: Ecological Informatics (3) 2008, S. 64–74.
- Webb, A. R., David Lowe und M. D. Bedworth*: [Comparison] A Comparison of Non-linear Optimisation Strategies for Feed-Forward Adaptive Layered Networks. Memorandum 4157. Royal Signals and Radar Establishment. Malvern (Worcestershire) 1988. Abrufbar unter: <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA200044>. Abgerufen am: 25.8.2014.
- Weber, Jürgen und Utz Schäffer*: [Controlling] Einführung in das Controlling. 14. Aufl., Stuttgart 2014.
- Weber, Jürgen*: [Gestaltung] Gestaltung der Kostenrechnung. Wiesbaden 2005.
- Weber, Jürgen*: [Verhaltensorientierung] Kostenrechnung zwischen Verhaltens- und Entscheidungsorientierung. In: Kostenrechnungspraxis (38) 1994, S. 99–104.
- Weber, Jürgen*: [Logistikkostenrechnung] Logistikkostenrechnung. 3. Aufl., Berlin, Heidelberg 2012.

- Weber, Jürgen*: [Selektives Rechnungswesen] Selektives Rechnungswesen. In: Zeitschrift für Betriebswirtschaft (66) 1996, S. 925–946.
- Weber, Max*: [Wirtschaft] Wirtschaft und Gesellschaft. In: Grundriss der Sozialökonomik. III. Abteilung. 3. Aufl., Tübingen 1947.
- Weckman, Gary R., Helmut W. Paschold, John D. Dowler, Harry S. Whiting und William A. Young*: [Manufacturing Cost] Using Neural Networks with Limited Data to Estimate Manufacturing Cost. In: Journal of Industrial and Systems Engineering (3) 2010, S. 257–274.
- White, Halbert*: [Nonparametric Regression] Connectionist Nonparametric Regression: Multilayer Feedforward Networks Can Learn Arbitrary Mappings. In: Neural Networks (3) 1990, S. 535–549.
- Whitley, Darrell und Andrew M. Sutton*: [Genetic Algorithms] Genetic Algorithms – A Survey of Models and Methods. In: Handbook of Natural Computing. Hrsg. von G. Rozenberg, T. Bäck und J. N. Kok. Berlin, Heidelberg 2012, S. 637–671.
- Wiedmann, Klaus-Peter und Hans Hermann Jung*: [Segmentierungsanalyse] Neuronale Netze zur Segmentierungsanalyse in der Automobilindustrie. In: Neuronale Netze im Marketing-Management. Hrsg. von K.-P. Wiedmann und F. Buckler. Wiesbaden 2001, S. 209–227.
- Wiedmann, Klaus-Peter*: [Zielkundenmanagement] Neuronale Netze als Basis eines effizienten Zielkundenmanagements in der Finanzdienstleistungsbranche. In: Neuronale Netze im Marketing-Management. Hrsg. von K.-P. Wiedmann und F. Buckler. Wiesbaden 2001, S. 287–320.
- Wiering, Marco und Martijn van Otterlo (Hrsg.)*: [Reinforcement Learning] Reinforcement Learning. State-of-the-Art. Berlin u. a. 2012.
- Wilamowski, Bogdan M., Serdar Iplikci, Okyay Kaynak und M. Önder Efe*: [Convergence] An Algorithm for Fast Convergence in Training Neural Networks. In: Proceedings of the International Joint Conference on Neural Networks, IJCNN (3) 2001, S. 1778–1782.
- Wild, Jürgen*: [Unternehmensplanung] Grundlagen der Unternehmensplanung. 4. Aufl., Opladen 1982.
- Wilms, Stefan*: [Abweichungsanalysemethoden] Abweichungsanalysemethoden der Kostenkontrolle. Bergisch Gladbach, Köln 1988.
- Wimmer, Konrad*: [Kostenabweichungsanalyse] Kostenabweichungsanalyse und Kostensenkung. In: Zeitschrift für Betriebswirtschaft (64) 1994, S. 981–998.

- Windolph, Melanie* und *Klaus Moeller*: [Open-book] Open-book Accounting: Reason for failure of inter-firm cooperation? In: *Management Accounting Research* (23) 2012, S. 47–60.
- Witte, Eberhard*: [Phasen-Theorem] Phasen-Theorem und Organisation komplexer Entscheidungsverläufe. In: *Zeitschrift für betriebswirtschaftliche Forschung* (20) 1968, S. 625–647.
- Xue, Xiaowei, Min Yao, Zhaohui Wu* und *Jianhua Yang*: [Genetic Ensemble] Genetic ensemble of extreme learning machine. In: *Neurocomputing* (129) 2014, S. 175–184.
- Yao, Xin* und *Yong Liu*: [Evolutionary System] A new evolutionary system for evolving artificial neural networks. In: *IEEE Transactions on Neural Networks* (8) 1997, S. 694–713.
- Yao, Xin* und *Yong Liu*: [Population] Making use of population information in evolutionary artificial neural networks. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* (28) 1998, S. 417–425.
- Yao, Xin*: [Evolving] Evolving Artificial Neural Networks. In: *Proceedings of the IEEE* (87) 1999, S. 1423–1447.
- Zelewski, Stephan*: [Forschungsprogramme] Forschungsprogramme der Produktionstheorie. Stellungnahme zum *Zeitschrift für Betriebswirtschaft-Beitrag* von Herrn Univ.-Prof. Dr. Harald Dyckhoff: "Neukonzeption der Produktionstheorie". In: *Zeitschrift für Betriebswirtschaft* (74) 2004, S. 487–497.
- Zhang, G. Peter*: [Time-Series] Neural Networks for Time-Series Forecasting. In: *Handbook of Natural Computing*. Hrsg. von G. Rozenberg, T. Bäck und J. N. Kok. Berlin, Heidelberg 2012, S. 461–477.
- Zhang, Rui, Yuan Lan, Guang-Bin Huang* und *Zong-Ben Xu*: [Approximation] Universal approximation of extreme learning machine with adaptive growth of hidden nodes. In: *IEEE Transactions on Neural Networks and Learning Systems* (23) 2012, S. 365–371.
- Zhang, Rui, Zongben Xu, Guang-Bin Huang* und *Dianhui Wang*: [Dynamic Learning] Global Convergence of Online BP Training with Dynamic Learning Rate. In: *IEEE Transactions on Neural Networks and Learning Systems* (23) 2012, S. 330–341.
- Zhang, Y. F., Jerry Y. H. Fuh* und *W. T. Chan*: [Packaging Products] Feature-based cost estimation for packaging products using neural networks. In: *Computers in Industry* (32) 1996, S. 95–113.

- Zhong, Haoming, Chunyan Miao, Zhiqi Shen und Yuhong Feng*: [Effectiveness] Comparing the learning effectiveness of BP, ELM, I-ELM, and SVM for corporate credit ratings. In: *Neurocomputing* (128) 2014, S. 285–295.
- Zhou, Zhi-Hua, Jianxin Wu und Wei Tang*: [Ensembling] Ensembling Neural Networks: Many could be better than all. In: *Artificial Intelligence* (137) 2002, S. 239–263.
- Zimmermann, Volker*: [Geschäftsprozessmanagement] Objektorientiertes Geschäftsprozessmanagement. Wiesbaden 1999.