

Inhaltsverzeichnis

1. EINFÜHRUNG	1
1.0 Kapitelübersicht	1
1.1 Neuronale Netzwerke und die Modellierung Kognitiver Prozesse	1
1.1.1 Neuronale Netze als komplexe dynamische Systeme	1
1.1.2 Kognitive Prozesse als Trajektorien in Zustandsräumen	2
1.2 Die Grundkomponenten eines Neuronalen Netzwerks	4
1.2.1 Die Struktur der elementaren Verarbeitungseinheiten	5
1.2.1.1 Die Inputfunktion	6
1.2.1.2 Aktivierungsfunktionen	7
1.2.1.2.1 Die Schrittfunktion	7
1.2.1.2.2 Partiiell lineare Aktivierungsfunktion	7
1.2.1.2.3 Die Sigmoidfunktion	8
1.2.2 Die Netzwerkarchitektur	10
1.2.2.1 Vollständig rekurrente Architektur	10
1.2.2.2 Feedforward-Architektur	11
1.2.2.3 Partiiell rekurrente Architektur: Feedforward-Architektur mit rekurrenten Verknüpfungen	11
1.2.3 Der Verarbeitungsalgorithmus	12
1.2.4 Die zu optimierende Funktion	12
1.3 Elementare Vektor- und Matrizenrechnung	13
1.3.1 Definition von Vektoren und Matrizen	13
1.3.1.1 Vektoren: Definition und Schreibweise	13
1.3.1.2 Graphische Darstellung von Vektoren	14
1.3.1.3 Matrizen: Definition und Schreibweise	15
1.3.2 Transponieren von Vektoren und Matrizen, symmetrische Matrizen	16
1.3.3 Addition und Subtraktion von Vektoren und Matrizen	17
1.3.3.1 Geometrische Darstellung der Addition und Subtraktion von Vektoren	17
1.3.4 Multiplikation eines Vektors und einer Matrix mit einem Skalar	18
1.3.4.1 Geometrische Darstellung der Multiplikation eines Vektors mit einem Skalar	19
1.3.4.2 Multiplikation von Vektoren und Matrizen	19
1.3.5 Komponentenweise Multiplikation von Vektoren und Matrizen	23
1.3.6 Die Inverse einer Matrix	24
1.4 Simulation Neuronaler Netzwerke mit Hilfe von Excel	24
1.4.1 Durchführung elementarer Matrizenoperationen mit Hilfe von Excel	25
1.4.1.1 Durchführung einer Matrizenmultiplikation	25
1.4.1.2 Zusammenstellung der elementaren Matrizenoperationen	31
1.4.2 Automatische Expansion von Skalaren und Vektoren	31
1.4.3 Die Spezifikation beliebiger Funktionen als Matrizenfunktionen	33
1.4.4 Zellenreferenzen (Zellenbezüge) und Namen	34
1.4.4.1 Variable und fixe Referenzen	35
1.4.4.2 Zirkuläre Referenzen	35
1.4.4.3 Spezifikation von Namen	36
1.4.4.3.1 Definition eines Namens mittels <i>Insert/Name/Define</i>	36
1.4.4.3.2 Erzeugen eines Namens mittels <i>Insert/Name/Create</i>	38
1.4.4.3.3 Die Verwendung von Namen in Berechnungsausdrücken	40
1.5 Zusammenfassung der zentralen Inhalte des Kapitels	40
1.6 Übungen zu Kapitel 1	41

2. PARALLEL CONSTRAINT SATISFACTION	45
2.0 Kapitelübersicht	45
2.1 Kognitive Verarbeitung als Erfüllung von »weichen« Beschränkungen	45
2.2 Ein paralleles und serielles Modell zur Simulation des Gestalteffekts beim Necker-Würfel	46
2.2.1 Gestalteffekt und Interpretationen des Necker-Würfels	46
2.2.2 Die Netzwerkarchitektur	46
2.2.3 Input- und Aktivierungsfunktion	49
2.2.4 Die Harmoniefunktion	51
2.2.5 Der Verarbeitungsalgorithmus	52
2.2.6 Simulation einer Miniversion des Necker-Würfels	53
2.2.6.1 Der Aufbau des Excel-Tabellenblatts zur Simulation der Miniversion	54
2.2.6.2 Berechnung des ersten Verarbeitungszyklus	56
2.2.6.2.1 Berechnung des Netzinputs	56
2.2.6.2.2 Berechnung des Aktivierungsvektors	57
2.2.6.2.3 Berechnung der Harmonie	59
2.2.6.3 Berechnung der restlichen Verarbeitungszyklen und die allgemeine Struktur der Verarbeitung	60
2.2.7 Struktur der Excel-Tabellenblätter zur Simulation von Parallel Constraint Satisfaction Modellen	61
2.2.7.1 Der Aufbau des Tabellenblatts zur Simulation des Necker-Würfels	61
2.2.7.2 Block mit Überschrift, Graphiken, Kommentare	62
2.2.7.3 Die Symboltabelle	62
2.2.7.4 Die Gewichtsmatrix	64
2.2.7.5 Der Verarbeitungsblock	66
2.2.8 Eine serielle Version des Necker-Würfels	66
2.2.8.1 Implementation des seriellen Necker-Würfels mittels Excel	66
2.2.8.1.1 Der Zufallsgenerator	66
2.2.8.1.2 Änderung des Tabellenblatts gegenüber der parallelen Version	68
2.2.8.1.3 Berechnung des Netzinputs	69
2.2.8.1.4 Berechnung der Aktivierung	71
2.3 Boltzmann-Maschinen: Eine probabilistische Version von Parallel Constraint Satisfaction	72
2.3.1 Spezifikation der Boltzmann-Maschine	72
2.3.2 Eine physikalische Interpretation von Boltzmann-Maschinen	74
2.3.3 Boltzmann-Maschinen und Kombinatorische Optimierung	75
2.3.4 Abkühlungsplan (Annealing Schedule) und Anzahl notwendiger Verarbeitungsschritte	80
2.3.4.1 Berechnung der Temperaturabnahme	80
2.3.4.2 Bestimmung der Anzahl Blöcke mit konstanter Temperatur und der Anzahl Iterationen pro Block	83
2.3.4.2.1 Wahl der Anzahl Iterationen innerhalb eines Blockes mit konstanter Temperatur	83
2.3.4.2.2 Wahl der Anzahl Blöcke mit konstanter Temperatur	84
2.3.5 Der Verarbeitungsalgorithmus	85
2.3.6 Implementation des Necker-Würfels als Boltzmann-Maschine mittels Excel	87
2.3.6.1 Die Symboltabelle	87
2.3.6.2 Aktivierungsvektor und Harmoniefunktion	88
2.3.6.3 Der Abkühlungsplan	89
2.4 Modellierung kognitiver Prozesse mittels Parallel Constraint Satisfaction Prozess	90
2.4.1 Ein Modell des Abrufs von schematischer Information: Die Jets & Sharks	90
2.4.1.1 Bemerkungen zum Konstrukt des Schemas	90
2.4.1.2 Ein Beispielmmodell: Die Jets und die Sharks	91
2.4.1.2.1 Die Netzwerkarchitektur	91

2.4.1.2.2	Berechnung des Netzeingangs	94
2.4.1.2.3	Berechnung der Aktivierung	94
2.4.1.2.4	Berechnung der Harmonie	95
2.4.1.2.5	Der Verarbeitungsalgorithmus des Modells der Jets & Sharks	97
2.4.1.2.6	Implementation der Jets und Sharks mittels Excel	99
2.4.2	Modellierung der Reduktion von Kognitiver Dissonanz mit Hilfe von Parallel Constraint Satisfaction	101
2.4.2.1	Das Konzept der Kognitiven Dissonanz	101
2.4.2.2	Das Experiment von Gerard & Mathewson (1966)	101
2.4.2.2.1	Methode	102
2.4.2.2.2	Ergebnisse	102
2.4.2.3	Dissonanzreduktion als Parallel Constraint Satisfaction Prozess	102
2.4.2.3.1	Struktur und Verknüpfung von Kognitionen	102
2.4.2.3.2	Netzeingang und Aktivierungsfunktion	103
2.4.2.3.3	Konsonanz- und Dissonanzfunktion	104
2.4.2.3.4	Netzwerkarchitektur	104
2.4.2.3.5	Der Verarbeitungsalgorithmus	107
2.4.2.4	Implementation des Modells der Dissonanzreduktion	109
2.5	Zusammenfassung der zentralen Inhalte des Kapitels	113
2.6	Übungen zu Kapitel 2	114
3.	LERNEN IN NEURONALEN NETZEN	119
3.0	Kapitelübersicht	119
3.1	Der Perzeptron-Lernalgorithmus	119
3.1.1	Die Netzwerkarchitektur eines Perzeptrons	120
3.1.2	Der Verarbeitungsalgorithmus	121
3.1.3	Die Funktionsweise des Perzeptronalgorithmus	122
3.1.4	Erweiterungen des Grundalgorithmus	129
3.1.4.1	Einführung einer Lernrate	129
3.1.4.2	Verwendung von Schwellen- oder Bias-Parameter	129
3.1.4.3	Funktionsweise des Algorithmus bei Mustern mit vielen Dimensionen	130
3.1.4.4	Musterklassifikation bei Vorliegen von mehr als zwei Kategorien	130
3.1.5	Implementation des Perzeptronalgorithmus	131
3.2	Das Gradientenabstiegsverfahren	133
3.2.1	Die Fehlerfunktion	133
3.2.2	Die Minimierung einer Funktion mit Hilfe der Methode des Gradientenabstiegs	135
3.2.3	Partielle Ableitung und Gradient	136
3.2.4	Anwendungsbeispiel: Die Berechnung der linearen Regressionskoeffizienten mittels Gradientenabstieg auf der Funktion der quadrierten Abweichung zwischen Modell und Daten	143
3.2.5	Implementation des Gradientenabstiegsverfahrens	144
3.2.6	Die Grenzen des Gradientenabstiegsverfahrens	147
3.2.7	Verfahren zur Beschleunigung des Gradientenabstiegs	150
3.2.7.1	Einführung eines Momentum-Terms	150
3.2.7.2	Die Delta-Bar-Delta – Regel	153
3.2.7.3	Implementation der Delta-Bar-Delta – Regel	156
3.3	Die Delta-Regel	158
3.3.1	Die Netzwerkarchitektur	158
3.3.2	Der Verarbeitungsalgorithmus	159
3.3.3	Anwendung der Delta-Regel als iterative Prozedur zur Schätzung der multiplen Regressionskoeffizienten	160

3.3.4	Implementation des Netzwerks zur Schätzung der linearen Regressionskoeffizienten	162
3.4	Das Rescorla-Wagner – Modell (RW-Modell)	164
3.4.1	Die Modellarchitektur	164
3.4.2	Der Verarbeitungsalgorithmus	165
3.4.3	Simulation empirischer Phänomene	166
3.4.3.1	Blockierung	166
3.4.3.2	Overexpectation	170
3.4.3.3	Konditionierte Hemmung	171
3.4.4	Empirische Phänomene, die das RW-Modell nicht erklären kann	173
3.4.4.1	Löschung eines konditionierten Hemmers	173
3.4.4.2	Gelernte Irrelevanz	173
3.4.4.3	Rückwärtsévaluation	173
3.4.4.4	Overexpectation und Alles-oder-Nichts – Effekte	174
3.4.5	Implementation des RW-Modells	174
3.4.5.1	Aufbau des Tabellenblatts »Symboltabelle und Input«	175
3.4.5.1.1	Der Block mit den Inputmustern und dem Feedback	175
3.4.5.1.2	Die Symboltabelle	175
3.4.5.2	Aufbau der Tabellenblätter mit den Verarbeitungsstrukturen	176
3.5	Konfigurable Modelle	179
3.5.1	Das Configural-Cue – Modell	180
3.5.2	Das Konfigurable Modell von Pearce	182
3.5.2.1	Die Modellarchitektur	182
3.5.2.2	Der Verarbeitungsalgorithmus (informelle Darstellung)	183
3.5.2.3	Der Verarbeitungsalgorithmus (formale Darstellung)	185
3.5.2.4	Implementation des Konfigurablen Modells von Pearce	187
3.5.2.4.1	Aufbau des Tabellenblatts: »Input, Symboltabelle, Ergebnisse«	188
3.5.2.4.2	Der Aufbau des Tabellenblatts: »A+BC+ABC-«	190
3.6	Einschränkungen der Lernfähigkeit Neuronaler Netze	192
3.6.1	Geometrische Darstellung der Kategoriestruktur	194
3.6.2	Einschränkungen aufgrund der Lage der Trennfläche	195
3.6.3	Einschränkung aufgrund der Linearität der Trennfläche	198
3.6.4	Repräsentation der XOR-Struktur durch Einführung konfigurabler Einheiten	199
3.6.5	Repräsentation der XOR-Struktur durch Verwendung einer Hidden-Lage	201
3.7	Lernen mit Hilfe der Verallgemeinerten Delta-Regel und der Backpropagation-Algorithmus	203
3.7.1	Die Netzwerkarchitektur des Beispielnetzes	203
3.7.2	Vorwärtspropagation der Aktivierung	204
3.7.3	Fehlerfunktion und Gradient	204
3.7.4	Fehlerpropagation und Gewichtsmodifikation	206
3.7.5	Der Backpropagation-Algorithmus (formale Darstellung)	209
3.7.6	Implementation des Backpropagation-Algorithmus	212
3.7.6.1	Der Aufbau des Tabellenblatts »Input und Symboltabelle«	212
3.7.6.2	Der Aufbau des Tabellenblatts »Verarbeitung«	214
3.7.6.2.1	Input- und Zielaktivierung	214
3.7.6.2.2	Strukturen der Hiddenlage	216
3.7.6.2.2.1	Vorwärtspropagation	216
3.7.6.2.2.2	Rückwärtspropagation	217
3.7.6.2.2.3	Gewichts- und Biasmodifikation	217
3.7.6.2.3	Strukturen der Outputlage	218
3.7.7	Geschwindigkeit des Erlernens der XOR-Struktur	218
3.7.8	Die Repräsentation komplexer Kategoriestrukturen und die Überwindung physikalischer Ähnlichkeiten	219
3.8	ALCOVE: Ein konfigurables Modell mit Aufmerksamkeitsgewichten und Verallgemeinerter Delta-Lernregel	220

3.8.1 Einführung	220
3.8.2 Die Netzwerkarchitektur	220
3.8.3 Der Lern- und Verarbeitungsalgorithmus von ALCOVE	
(informelle Darstellung)	221
3.8.3.1 Vorwärtspropagation der Aktivierungen	221
3.8.3.2 Rückwärtspropagation des Fehlersignals	228
3.8.4 Der Lern- und Verarbeitungsalgorithmus von ALCOVE	229
3.8.5 Implementation von ALCOVE	230
3.8.5.1 Ein Experiment von Medin & Schaffer (1978)	230
3.8.5.2 Der Aufbau des Tabellenblatts »Input, Symboltab., A-Gewichte«	232
3.8.5.3 Der Aufbau des Tabellenblatts »Hiddenlage«	236
3.8.5.4 Der Aufbau des Tabellenblatts »Outputlage«	239
3.8.5.5 Die Verwendung des SOLVER zur Schätzung der Modellparameter	241
3.9 Die Komplexität des Lernproblems und die Grenzen des Lernens mehr- lagiger Netzwerke	243
3.10 Zusammenfassung der zentralen Inhalte des Kapitels	248
3.11 Übungen zu Kapitel 3	249
4. PRINZIPIEN DER ASSOZIATIVEN SPEICHERUNG	259
4.0 Kapitelübersicht	259
4.1 Eigenschaften Assoziativer Gedächtnismodelle	259
4.2 Das Grundmodell	261
4.2.1 Beschreibung der Speichermengen	261
4.2.2 Die Netzwerkarchitektur	262
4.2.3 Der Verarbeitungsalgorithmus	263
4.3 Aufbau einer Speichermatrix mit Hilfe der Hebb'schen Regel	265
4.3.1 Das Prinzip der Hebb'schen Regel	265
4.3.2 Die Hebb'sche Regel in Matrizenform	266
4.3.3 Implementation der Hebb'schen Regel	268
4.3.3.1 Aufbau der Speichermatrix	268
4.3.3.2 Interpretation der Gewichte der Hebb'schen Speichermatrix	270
4.3.3.3 Informationsabruf	270
4.3.3.3.1 Abruf unter Verwendung der Speichervektoren als Abrufcues	270
4.3.3.3.2 Abruf mittels partieller Stimuli als Abrufcues	272
4.3.3.3.3 Rekonstruktion partieller Muster bei Verwendung eines non- linearen Abrufmechanismus	274
4.3.4 Die Hebb'schen Regel bei Verwendung non-orthogonaler Mustervektoren	275
4.3.5 Orthogonale Projektion und die Orthogonalisierung von Vektoren mit Hilfe des Verfahrens von Gram-Schmidt	277
4.3.5.1 Das Konzept der Orthogonalen Projektion	277
4.3.5.2 Die Berechnung der Orthogonalen Projektion	281
4.3.5.3 Optimalitätseigenschaft der Orthogonalen Projektion	283
4.3.5.4 Das Orthogonalisierungsverfahren von Gram-Schmidt	284
4.3.5.5 Implementation der Orthogonalisierung nach Gram-Schmidt	286
4.3.5.6 Lineare Unabhängigkeit und Orthogonalisierung von Vektoren	291
4.3.6 Die Hebb'sche Speichermatrix als Projektionsmatrix	298
4.3.7 Simulationsbeispiele mit Hebb'sche Speichermatrix	300
4.3.7.1 Simulation der Beziehung zwischen Kategoriegröße und prototypen- basierter Klassifikation	300
4.3.7.1.1 Erzeugung von Prototypen und Einzelexemplaren	301
4.3.7.1.2 Aufbau der Tabellenblätter zur Simulation	302
4.3.7.1.3 Ergebnis der Simulation	305

4.3.7.2 Brain-State-in-a-Box: Simulation kategorieller Wahrnehmung	306
4.3.7.2.1 Erzeugung der Stimuli	307
4.3.7.2.2 Das Modell: Brain-State-in-a-Box	310
4.3.7.2.3 Implementation des Modells	311
4.3.7.2.4 Ergebnisse der Simulation	312
4.4 Die Bildung einer Speichermatrix als Projektionsmatrix	313
4.4.1 Die allgemeine Form der Projektionsmatrix für linear unabhängige Speichervektoren	314
4.4.2 Berechnung der linearen Gewichte der Orthogonalen Projektion	318
4.4.3 Die Herleitung der Projektionsmatrix	319
4.4.4 Das Problem der Linearen Abhängigkeit	327
4.4.4.1 Geometrische Darstellung des Problems	327
4.4.4.2 Ansatz zu einer Lösung des Problems: Entfernung redundanter Vektoren	329
4.4.5 Lösung des Problems der linearen Abhängigkeit: Generalisierte Inverse	330
4.4.5.1 Die Bildung einer Inversen	330
4.4.5.2 Die Bildung einer Generalisierten Inversen	335
4.4.5.3 Die Moore-Penrose Pseudoinverse	340
4.4.6 Anwendungsbeispiel: Extraktion von Regularitäten aus Beispielen	343
4.4.6.1 »Regelbefolgung« versus »Regel-beschreibbares Verhalten«	343
4.4.6.2 Bildung einer Projektionsmatrix zur Heteroassoziation	345
4.4.6.3 Erlernen der 78-Regel	346
4.5 Die Verwendung der Delta-Regel zum Aufbau einer Speichermatrix	348
4.5.1 Eine modifizierte Form der Delta-Regel für non-lineare, autoassoziative Speichersysteme	348
4.5.2 Die Verwendung der Delta-Regel in non-linearen, autoassoziativen Speichersystemen	349
4.5.2.1 Modellbeschreibung	349
4.5.2.2 Implementation des Modells	352
4.5.2.3 Ergebnisse der Simulation	356
4.5.3 Anwendungsbeispiel: Simulation des Erwerbs von Kausalwissen	356
4.5.3.1 Simulation von Blockierung	356
4.5.3.1.1 Netzwerkarchitektur	356
4.5.3.1.2 Stimuli und Ablauf des Lernens	357
4.5.3.1.3 Parameterkonfiguration	357
4.5.3.1.4 Implementation des Modells	358
4.5.3.1.5 Simulationsergebnisse	358
4.5.3.2 Simulation des Erwerbs asymmetrischer Kausalrelationen	359
4.5.3.2.1 Netzwerkarchitektur und Parameterkonfiguration	359
4.5.3.2.2 Stimuli und Ablauf des Lernens	360
4.5.3.2.3 Simulationsergebnisse	360
4.6 Zusammenfassung der zentralen Inhalte des Kapitels	361
4.7 Übungen zu Kapitel 4	362
LITERATURVERZEICHNIS	369
ANHANG: LÖSUNGEN ZU DEN ÜBUNGSAUFGABEN	375
Lösungen zu den Übungsaufgaben von Kapitel 1	375
Lösungen zu den Übungsaufgaben von Kapitel 2	376
Übung 2.1 zur Miniversion des Necker-Würfels	376
Übung 2.2 zur parallelen Version des Necker-Würfels	376
Übung 2.3 zur seriellen Version des Necker-Würfels	378

Übung 2.4 zur Boltzmann-Maschine	378
Übung 2.5 zu den Jets und Sharks	379
Übung 2.6 zum Modell der Dissonanzreduktion	380
Lösungen zu den Übungsaufgaben von Kapitel 3	381
Übung 3.1 zum Perzeptron-Lernalgorithmus	381
Übung 3.2 zum Gradientenverfahren	382
Übung 3.3 zur Beschleunigung des Gradientenverfahrens	382
Übung 3.4 zur Delta-Regel	383
Übung 3.5 zum RW-Modell	383
Übung 3.6 zum Configural-Cue – Modell	384
Übung 3.7 zum Konfigurrellen Modell von Pearce	385
Übung 3.8 zum Backpropagation-Algorithmus	386
Übung 3.9 zu ALCOVE	386
Übung 3.10 zum SOLVER	387
Lösungen zu den Übungsaufgaben von Kapitel 4	388
Übung 4.1 zur Hebb'schen Speichermatrix	388
Übung 4.2 zur Projektionsmatrix	390
Übung 4.3 zur Delta-Regel	393
SACHINDEX	395