

Auf einen Blick

1	Einleitung	17
2	Texte bauen mit Markow	29
3	Schreibfehler automatisch korrigieren	51
4	Wörter gruppieren	67
5	Spiele für eine Person lösen	91
6	Spiele für zwei Personen gewinnen	107
7	Q-Learning	131
8	K-nächste-Nachbarn	167
9	K-means-Clustering	181
10	Neuronale Netze I: Das Häschenproblem	197
11	Neuronale Netze II: Auf dem Weg ins Tal	213
12	Neuronale Netze III: Fehler zurückverfolgen mit dem Neuronentrainer	229
13	Neuronale Netze IV: Faltungsnetze, Autoencoder, GANs und DQL	249

Inhalt

Materialien zum Buch	16
1 Einleitung	17
1.1 Worum es uns in diesem Buch geht	18
1.2 Für wen wir dieses Buch geschrieben haben	19
1.3 Aufbau der einzelnen Kapitel	20
1.4 Ein Wort an die Programmierunkundigen	20
1.5 Beispielprogramme und die Webseite zum Buch	21
1.6 Warum wir JavaScript und p5.js verwendet haben	23
1.7 Begriffliche Abgrenzung und Fachbegriffe	24
1.8 Inhalte, Themen, Kapitel	25
1.9 Dank	28
2 Texte bauen mit Markow	29
2.1 Das Beispielprogramm Nonsense-Texter	33
2.2 Der Code des Nonsense-Texters unter der Lupe	35
2.2.1 Die Komponenten des Markow-Objekts	35
2.2.2 Übergänge lernen	36
2.2.3 Nonsense-Texte produzieren	38
2.3 Das Beispielprogramm Wörter vorschlagen	41
2.3.1 Die Komponenten des Markow-Objekts	41
2.3.2 Übergänge und Häufigkeiten lernen	42
2.4 Wörter vorschlagen	44
2.5 Gewichteter Zufall	46
2.6 Ideen zum Weitermachen	48
2.7 Zusammenfassung und Ausblick	49

3 Schreibfehler automatisch korrigieren 51

3.1	Das Beispielprogramm Wortvergleich	52
3.2	Die Matrix befüllen	55
3.2.1	Die Füllung der oberen Zeile und der linken Spalte	55
3.2.2	Die Füllung der verbleibenden Zellen	56
3.2.3	Die drei Schritte unter der Lupe	60
3.3	Die Umsetzung im Beispielprogramm	60
3.3.1	Das Levenshtein-Objekt	60
3.3.2	Die Funktion matrix()	61
3.4	Das Beispielprogramm Korrekturvorschläge	63
3.5	Ideen zum Weitermachen	65
3.6	Zusammenfassung und Ausblick	66

4 Wörter gruppieren 67

4.1	Items und Transaktionen	69
4.2	Kenngößen der Assoziationsanalyse	70
4.2.1	Support	70
4.2.2	Confidence	71
4.2.3	Lift	72
4.3	Ein Beispiel von Hand gerechnet	74
4.4	Das Beispielprogramm Begriffsnetz	77
4.4.1	Die Datenquelle	79
4.4.2	Beschränkung der Anwendung auf Item-Paare	79
4.5	Eine Tour durch den Code	80
4.5.1	Die Klasse Begriffsnetz	80
4.5.2	Enkodierung der Transaktionen	81
4.5.3	Befüllung der Arrays für Support	83
4.5.4	Befüllung der Arrays für Confidence und Lift	85
4.5.5	Die Funktion assoziationen()	85
4.6	Ideen zum Weitermachen	86

4.7	Zusammenfassung und Ausblick	88
4.7.1	Der Apriori-Algorithmus	88
4.7.2	Transaktionstabellen	88
4.7.3	Eine Übersicht aller Fachbegriffe aus diesem Kapitel	89

5 Spiele für eine Person lösen 91

5.1	Das Spiel Fruchtkräsch	91
5.2	Wie findet die KI den besten Zug?	93
5.3	Eine vielseitig einsetzbare Spiel-KI	96
5.4	Die Klasse Spielzustand	97
5.4.1	Züge liefern modifizierte Spielzustände	97
5.4.2	Die möglichen Züge	98
5.4.3	Die Bewertung eines Spielzustands	98
5.4.4	Ein Gedächtnis für Züge	99
5.4.5	Die Schnittstelle im Überblick	99
5.5	Die Klasse KI	100
5.5.1	Alle Folgezustände eines Spielzustands berechnen	100
5.5.2	Alle Spielverläufe per Warteschlangenverfahren berechnen	101
5.5.3	Die Spielzustände nach Bewertung sortieren	104
5.5.4	Die Funktion besterZug()	104
5.5.5	Die Funktionen des KI-Objekts im Überblick	105
5.6	Ideen zum Weitermachen	105
5.7	Zusammenfassung und Ausblick	106

6 Spiele für zwei Personen gewinnen 107

6.1	Das Spiel Reversi	108
6.2	Das Beispielprogramm Reversi KI	109
6.3	Der Minimax-Algorithmus	110
6.3.1	Anwendungsgebiete und Grenzen des Minimax-Algorithmus	112

6.4	Tiefensuche und Rekursion	113
6.4.1	Breitensuche und Tiefensuche	114
6.4.2	Die Paradoxie der Rekursion	115
6.4.3	Verzweigte Rekursion	119
6.5	Die Klasse Spielzustand	121
6.5.1	Die Bewertungsfunktion	121
6.5.2	Die Schnittstelle im Überblick	123
6.6	Die Klasse KI	123
6.7	Beschleunigung mit Alpha-Beta-Pruning	128
6.8	Ideen zum Weitermachen	129
6.9	Zusammenfassung und Ausblick	130

7 Q-Learning 131

7.1	Das Eichhörnchen und das Nussversteck	132
7.2	Umwelt, Agent, Aktion und Belohnung	137
7.2.1	Das Verhältnis von Agent und Umwelt	138
7.3	Die Q-Tabelle	139
7.3.1	Q steht für Qualität	140
7.4	Das Beispielprogramm Q-Lerner	140
7.5	Die Q-Tabelle befüllen	145
7.5.1	Warum funktioniert das?	147
7.6	Der Code unter der Lupe	148
7.6.1	Die Umwelt	148
7.6.2	Der Q-Lerner	149
7.7	Gamma bestimmt die Weitsicht	150
7.8	Epsilon: Erforschung oder Anwendung	153
7.9	Ein zweiter Blick auf den Code	155
7.10	Alpha	157
7.11	Was wir weggelassen haben	158
7.11.1	Komplexere Umwelten	159
7.11.2	Kosten für Aktionen	159

7.11.3	Belohnungen mit Zustands-Aktions-Paaren verknüpfen	159
7.11.4	Mehrere Belohnungszustände	159
7.12	Ideen zum Weitermachen	160
7.12.1	OpenAI Gym	161
7.12.2	Das Buch von Sutton und Barto	162
7.13	Zusammenfassung und Ausblick	162
7.13.1	Menschliches Lernen vs. Q-Learning	162
7.13.2	Die Grenzen des Verfahrens	163

8 K-nächste-Nachbarn 167

8.1	Häschen, Igel, Vogelspinne oder Hai?	168
8.2	Das Beispielprogramm Tiere erkennen	169
8.3	Entfernungen bestimmen mit Pythagoras	172
8.4	Der Code im Detail	174
8.5	Ideen zum Weitermachen	178
8.6	Zusammenfassung und Ausblick	179

9 K-means-Clustering 181

9.1	Clusterbildung in Aktion	183
9.1.1	Mittelwert, Zentrum, Schwerpunkt	183
9.1.2	Die Schrittfolge des k-means-Clustering-Algorithmus	185
9.2	Das Beispielprogramm Wetterdaten gruppieren	186
9.3	Der Code	188
9.3.1	Zentren zufällig setzen	189
9.3.2	Datenpunkte zuordnen	189
9.3.3	Zentren neu berechnen	190
9.4	Grenzen des Verfahrens	191
9.4.1	Unsinnige Gruppierungen	191
9.4.2	Zu viele Dimensionen	192
9.4.3	Linear nicht trennbare Datenpunktmengen	194

9.5	Ideen zum Weitermachen	195
9.6	Zusammenfassung und Ausblick	195

10 Neuronale Netze I: Das Häschenproblem 197

10.1	Bildererkennung: ein klassisches Problem	198
10.2	Was ist ein Modell?	199
10.3	Der Aufbau eines neuronalen Netzes	201
10.4	Das Häschenneuron und seine Kollegen	204
10.4.1	Die biologische Nervenzelle als Vorbild	205
10.4.2	Das künstliche Neuron	207
10.4.3	b steht für Bias	208
10.4.4	Die Aktivierungsfunktion	208
10.5	Das Beispielprogramm Tiere erkennen II	209
10.6	Der Code	211
10.7	Ideen zum Weitermachen	211
10.8	Zusammenfassung und Ausblick	212

11 Neuronale Netze II: Auf dem Weg ins Tal 213

11.1	Das überwachte Lernen	214
11.2	Die schrittweise Justierung des Modells	216
11.2.1	Die grundlegende Idee	217
11.2.2	Steigung	218
11.2.3	Tangente	219
11.2.4	Ableitung	219
11.2.5	Der Gradientenabstieg	220
11.2.6	Die Lernrate	222
11.3	Das Beispielprogramm Gradientenabstieg	223
11.4	Der Code	225

11.5	Tipps zum Weitermachen	226
11.6	Zusammenfassung und Ausblick	226

12 Neuronale Netze III: Fehler zurückverfolgen mit dem Neuronentrainer 229

12.1	Was ist Backpropagation?	230
12.2	Das Beispielprogramm Neuronentrainer	231
12.2.1	Aufgaben und Netzarchitekturen	232
12.2.2	Ein Wiedersehen mit dem Häschenproblem	235
12.2.3	Lineare Trennbarkeit	236
12.3	Validierungsdaten, Überanpassung, Generatoren	237
12.3.1	Validierungsdaten und Überanpassung	238
12.3.2	Generatoren	239
12.4	Weitere Beispielaufgaben	240
12.4.1	Kreis und Hintergrund	240
12.4.2	Quadrat und Hintergrund	241
12.4.3	Farbtunnel sieben Farben	242
12.5	Die Anzahlen der verdeckten Schichten und der Neuronen	244
12.5.1	Viel hilft viel?	244
12.6	Was wir weggelassen haben	245
12.6.1	Dynamisierung der Lernrate	245
12.6.2	Batch und Epoche	245
12.6.3	Verlustfunktionen und Softmax	245
12.7	Ideen zum Weitermachen	246
12.8	Zusammenfassung und Ausblick	248

13 Neuronale Netze IV: Faltungsnetze, Autoencoder, GANs und DQL 249

13.1	Faltungsnetze	249
13.1.1	Das Beispielprogramm Filterlabor I	250

13.1.2	Der Filterkernel in Aktion	251
13.1.3	Padding und Striding	253
13.1.4	Das Beispielprogramm Filterlabor II	254
13.1.5	Eine Filterkombination, die den Buchstaben K erkennt	255
13.1.6	Die Struktur eines Faltungsnetzes	256
13.1.7	Faltungsnetze trainieren	257
13.2	Modelle, die Bilder erzeugen	258
13.3	Autoencoder	260
13.3.1	Dimensionsreduktion	260
13.3.2	Daten ausdenken	261
13.4	Generative Adversarial Networks	261
13.5	Deep Q-Learning	264
13.5.1	Wie kommt die Umwelt ins Modell?	265
13.6	Zusammenfassung und Ausblick	265
13.7	Tipps zum Weitermachen	268

Anhang 269

A	Eine kurze Einführung in JavaScript und p5.js	271
A.1	JavaScript	271
A.2	Was ist p5.js?	272
A.3	Der p5.js-Online-Editor	272
A.4	Text ausgeben	273
A.5	Bezeichner und Berechnungen	274
A.6	Die Struktur eines p5.js Programms	276
A.7	p5.js-Grafik-Grundlagen	278
A.8	Arrays	282
A.9	Bedingungen	284
A.10	Schleifen	288
A.11	Zufall in p5.js	291
A.12	Funktionen	294
A.13	Funktionen höherer Ordnung	297
A.14	Dateien laden und speichern in p5.js	301
A.15	Gültigkeitsbereiche von Bezeichnern	302
A.16	Objekte und Klassen	303

A.17	Bedienelemente in p5.js	306
A.18	Details zum Thema Strings	308
A.19	Sauber programmieren	312
B	Glossar	315
C	Quellen und Literaturhinweise	323
C.1	JavaScript	323
C.2	p5.js	323
C.3	Algorithmen	323
C.4	Künstliche Intelligenz & Co.	323
C.5	Onlinequellen	324
D	Abbildungsverzeichnis	325
	Index	329