

Inhaltsverzeichnis



Vorwort	13
Einleitung	15
Über den Autor	17
Danksagung	17
1 Deep Learning: Weshalb man sich damit befassen sollte	19
1.1 Willkommen bei »Deep Learning kapieren«	19
1.2 Weshalb du dich mit Deep Learning befassen solltest	19
1.3 Ist es schwierig, Deep Learning zu verstehen?	21
1.4 Warum du dieses Buch lesen solltest	21
1.5 Was du brauchst, um loszulegen	23
1.6 Python-Kenntnisse sind nützlich	24
1.7 Zusammenfassung	24
2 Grundlegende Konzepte: Wie lernen Maschinen?	25
2.1 Was ist Deep Learning?	25
2.2 Was ist Machine Learning?	26
2.3 Überwachtes Machine Learning	27
2.4 Unüberwachtes Machine Learning	28
2.5 Parametrisches und nichtparametrisches Lernen	29
2.6 Überwachtes parametrisches Lernen	30
2.7 Unüberwachtes parametrisches Lernen	33
2.8 Nichtparametrisches Lernen	34
2.9 Zusammenfassung	35

3	Vorhersage mit neuronalen Netzen: Forward Propagation	37
3.1	Vorhersage	37
3.2	Ein einfaches neuronales Netz trifft eine Vorhersage.	39
3.3	Was ist ein neuronales Netz?	40
3.4	Wie funktioniert das neuronale Netz?	41
3.5	Eine Vorhersage mit mehreren Eingaben treffen	44
3.6	Mehrere Eingaben: Wie verhält sich das neuronale Netz?	45
3.7	Mehrere Eingaben: vollständiger ausführbarer Code	51
3.8	Eine Vorhersage mit mehreren Ausgaben treffen	52
3.9	Vorhersagen mit mehreren Eingaben und mehreren Ausgaben treffen	54
3.10	Mehrere Ein- und Ausgaben: Wie funktioniert das?	56
3.11	Vorhersagen über Vorhersagen	58
3.12	Kurzeinführung in NumPy.	60
3.13	Zusammenfassung	64
4	Lernen in neuronalen Netzen: Gradientenabstieg	65
4.1	Vorhersagen, Vergleichen und Erlernen	65
4.2	Vergleichen.	66
4.3	Erlernen	66
4.4	Vergleichen: Trifft das Netz gute Vorhersagen?	67
4.5	Warum Fehler messen?	68
4.6	Was ist die einfachste Form des Lernens in neuronalen Netzen?	69
4.7	Hot und Cold Learning	71
4.8	Hot und Cold Learning: Eigenschaften	72
4.9	Richtung und Betrag anhand des Fehlers berechnen	73
4.10	Eine Iteration des Gradientenabstiegs	76
4.11	Lernen bedeutet nur, den Fehler zu verringern	78
4.12	Mehrere Schritte des Lernens	80
4.13	Weshalb funktioniert das? Was ist <code>weight_delta</code> eigentlich?	82
4.14	Tunnelblick auf ein Konzept.	84
4.15	Eine Kiste, aus der zwei Stangen ragen	85
4.16	Ableitungen: Zweiter Ansatz	86
4.17	Was du wirklich wissen musst	87
4.18	Was man eigentlich nicht wissen muss	88
4.19	Wie man Ableitungen zum Lernen verwendet	89
4.20	Wirkt das vertraut?	90
4.21	Den Gradientenabstieg stören	91
4.22	Visualisierung der übermäßigen Korrektur.	92

4.23	Divergenz	93
4.24	Einführung von alpha	94
4.25	Alpha im Code	95
4.26	Auswendig lernen	96
5	Mehrere Gewichte gleichzeitig erlernen: Generalisierung des Gradientenabstiegs	99
5.1	Lernen durch Gradientenabstieg mit mehreren Eingaben	99
5.2	Gradientenabstieg mit mehreren Eingaben erklärt	102
5.3	Mehrere Lernschritte	106
5.4	Einfrieren eines Gewichts: Was bewirkt das?	109
5.5	Lernen mittels Gradientenabstieg mit mehreren Ausgaben	111
5.6	Lernen mittels Gradientenabstieg mit mehreren Eingaben und mehreren Ausgaben	114
5.7	Was erlernen die Gewichte?	116
5.8	Visualisierung der Werte von Gewichten	118
5.9	Visualisierung von Skalarprodukten (gewichtete Summen)	119
5.10	Zusammenfassung	120
6	Das erste tiefe neuronale Netz: Einführung in Backpropagation . . .	121
6.1	Das Ampelproblem	121
6.2	Vorbereitung der Daten	123
6.3	Matrizen	124
6.4	Erstellen von Matrizen in Python	128
6.5	Erstellen eines neuronalen Netzes	129
6.6	Erlernen der gesamten Datenmenge	130
6.7	Vollständiger, Batch- und stochastischer Gradientenabstieg	131
6.8	Neuronale Netze erlernen eine Korrelation	132
6.9	Druck nach oben und unten	133
6.10	Grenzfall: Überanpassung	135
6.11	Grenzfall: Widersprüchliche Druckkräfte	137
6.12	Erlernen indirekter Korrelation	139
6.13	Korrelation erzeugen	140
6.14	Stapeln neuronaler Netze: Überblick	141
6.15	Backpropagation: Fehlerattribution	142
6.16	Backpropagation: Weshalb funktioniert das?	143
6.17	Linear vs. nichtlinear	144
6.18	Weshalb das neuronale Netz noch nicht funktioniert	145
6.19	Bedingte Korrelation	146

6.20	Eine kurze Pause	148
6.21	Das erste tiefe neuronale Netz	148
6.22	Backpropagation im Code	150
6.23	Eine Iteration der Backpropagation	152
6.24	Alles zusammenbringen	154
6.25	Warum sind tiefe Netze von Bedeutung?	155
7	Neuronale Netze abbilden: im Kopf und auf Papier	157
7.1	Zeit für Vereinfachungen	157
7.2	Korrelationszusammenfassung	158
7.3	Die alte, zu komplizierte Visualisierung	160
7.4	Die vereinfachte Visualisierung	161
7.5	Weitere Vereinfachung	162
7.6	Ein Netz bei der Vorhersage beobachten	164
7.7	Visualisierung mit Buchstaben statt Bildern	165
7.8	Variablen verknüpfen	166
7.9	Alles zusammen.	166
7.10	Die Bedeutung von Visualisierungstools	167
8	Signale erlernen, Rauschen ignorieren: Einführung in Regularisierung und Batching	169
8.1	Ein dreischichtiges Netz mit dem MNIST-Datensatz	169
8.2	Das war einfach	171
8.3	Auswendiglernen vs. Generalisierung	173
8.4	Überanpassung in neuronalen Netzen	174
8.5	Wie Überanpassung entsteht	175
8.6	Die einfachste Regularisierung: früher Abbruch	176
8.7	Der Industriestandard: das Dropout-Verfahren	177
8.8	Weshalb das Dropout-Verfahren funktioniert: Ensembles.	178
8.9	Das Dropout-Verfahren im Code	179
8.10	Bewertung des Dropout-Verfahrens mit dem MNIST-Datensatz	182
8.11	Batch-Gradientenabstieg	183
8.12	Zusammenfassung	186
9	Modellierung von Wahrscheinlichkeiten und Nichtlinearitäten: Aktivierungsfunktionen	187
9.1	Was ist eine Aktivierungsfunktion?	187
9.2	Standardaktivierungsfunktionen der verdeckten Schicht.	191
9.3	Standardaktivierungsfunktionen der Ausgabeschicht	192

9.4	Das grundlegende Problem: Eingaben ähneln einander	194
9.5	softmax-Berechnung	195
9.6	Anleitung für die Nutzung von Aktivierungsfunktionen	197
9.7	Multiplikation von delta mit der Steigung	199
9.8	Ausgabe in Steigung umwandeln (Ableitung)	200
9.9	Verbesserung des MNIST-Netzes	201
10	Einführung in Convolutional Neural Networks	205
10.1	Wiederverwendung von Gewichten an verschiedenen Stellen	205
10.2	Die Faltungsschicht.	206
10.3	Eine einfache Implementierung in NumPy	209
10.4	Zusammenfassung	214
11	Neuronale Netze, die Sprache verstehen:	
	Knecht – Mann + Frau == ?	217
11.1	Was bedeutet es, Sprache zu verstehen?	217
11.2	Verarbeitung natürlicher Sprache	218
11.3	Überwachte Verarbeitung natürlicher Sprache.	219
11.4	Die IMDB-Filmbewertungsdatenbank.	220
11.5	Wortkorrelationen in Eingabedaten erfassen	221
11.6	Vorhersage von Filmbewertungen	223
11.7	Kurz vorgestellt: Embedding-Schichten.	224
11.8	Interpretation der Ausgabe	227
11.9	Architektur neuronaler Netze.	228
11.10	Wort-Embeddings vergleichen	230
11.11	Welche Bedeutung hat ein Neuron?	232
11.12	Ausfüllen der Lücke	233
11.13	Bedeutung ergibt sich aus dem Verlust.	236
11.14	Knecht – Mann + Frau \approx Magd	239
11.15	Wortanalogien	240
11.16	Zusammenfassung	241
12	Rekurrente Schichten für Daten variabler Größe	243
12.1	Die Herausforderungen von variabler Größe	243
12.2	Spielen Vergleiche tatsächlich eine Rolle?	244
12.3	Die erstaunliche Leistungsstärke gemittelter Wortvektoren	246
12.4	Wie werden Informationen in diesen Embeddings gespeichert?	247
12.5	Wie verwendet ein neuronales Netz Embeddings?	248
12.6	Die Beschränkungen von Bag-of-words-Vektoren	249

12.7	Verwendung von Einheitsmatrizen zum Summieren von Wort-Embeddings	251
12.8	Matrizen, die überhaupt nichts verändern.	252
12.9	Erlernen der Übergangsmatrizen.	253
12.10	Lernen, nützliche Satzvektoren zu erzeugen.	254
12.11	Forward Propagation in Python	256
12.12	Wie funktioniert die Backpropagation?	257
12.13	Training durchführen	258
12.14	Einrichtung.	259
12.15	Forward Propagation beliebiger Größe	261
12.16	Backpropagation beliebiger Größe	262
12.17	Aktualisierung von Gewichten beliebiger Größe	263
12.18	Ausführung und Analyse der Ausgabe	264
12.19	Zusammenfassung	267
13	Automatische Optimierung: Entwicklung eines Deep-Learning-Frameworks	269
13.1	Was ist ein Deep-Learning-Framework?	269
13.2	Einführung in Tensoren	271
13.3	Einführung in Automatic Gradient Computation (Autograd)	272
13.4	Eine kurze Bestandsaufnahme	274
13.5	Mehrfach verwendete Tensoren	275
13.6	Autograd um mehrmals verwendbare Tensoren erweitern	276
13.7	Wie funktioniert Backpropagation durch Addition?	279
13.8	Negation hinzufügen.	280
13.9	Weitere Funktionen hinzufügen	281
13.10	Autograd zum Trainieren eines neuronalen Netzes verwenden	286
13.11	Automatische Optimierung hinzufügen	288
13.12	Schichttypen hinzufügen	290
13.13	Schichten, die weitere Schichten enthalten	291
13.14	Verlustfunktions-Schichten.	292
13.15	Erlernen eines Frameworks	293
13.16	Nichtlineare Schichten	294
13.17	Die Embedding-Schicht.	296
13.18	Indizierung zu Autograd hinzufügen	297
13.19	Zurück zur Embedding-Schicht	299
13.20	Die Kreuzentropie-Schicht	300
13.21	Die rekurrente Schicht eines neuronalen Netzes	302
13.22	Zusammenfassung	307

14	Lernen, wie Shakespeare zu schreiben:	
	Long Short-Term Memory	309
14.1	Zeichenbasierte Sprachmodellierung	309
14.2	Die Notwendigkeit einer verkürzten Backpropagation.	310
14.3	Truncated Backpropagation	312
14.4	Eine Stichprobe von der Ausgabe.	315
14.5	Verschwindende und explodierende Gradienten	317
14.6	Ein Testbeispiel für Backpropagation in RNNs.	318
14.7	LSTM-Zellen	320
14.8	Veranschaulichung von LSTM-Gates.	321
14.9	Die LSTM-Schicht	322
14.10	Erweiterung des zeichenbasierten Sprachmodells	324
14.11	Training des zeichenbasierten LSTM-Sprachmodells	325
14.12	Optimierung des zeichenbasierten LSTM-Sprachmodells.	327
14.13	Zusammenfassung	328
15	Deep Learning mit unbekanntem Daten: Federated Learning	329
15.1	Das Problem der Privatsphäre beim Deep Learning.	329
15.2	Federated Learning	330
15.3	Spam-Erkennung erlernen	332
15.4	Federation	334
15.5	Federated Learning hacken.	336
15.6	Sichere Aggregation	337
15.7	Homomorphe Verschlüsselung	338
15.8	Homomorph verschlüsseltes Federated Learning	339
15.9	Zusammenfassung	341
16	Wie geht es weiter? Ein kurzer Leitfaden	343
16.1	Glückwunsch!	343
16.2	Schritt 1: Lerne PyTorch	344
16.3	Schritt 2: Nimm an einem weiteren Deep-Learning-Kurs teil	344
16.4	Schritt 3: Lies ein mathematisch anspruchsvolles Lehrbuch	345
16.5	Schritt 4: Schreibe ein Blog und unterrichte Deep Learning.	345
16.6	Schritt 5: Twitter	347
16.7	Schritt 6: Verweise auf wissenschaftliche Arbeiten	347
16.8	Schritt 7: Verschaffe dir Zugriff auf eine GPU (oder mehrere).	347
16.9	Schritt 8: Lass dich für praktische Erfahrungen bezahlen	348
16.10	Schritt 9: Beteilige dich an einem Open-Source-Projekt	348
16.11	Schritt 10: Fördere deine Community vor Ort.	349
	Stichwortverzeichnis	351