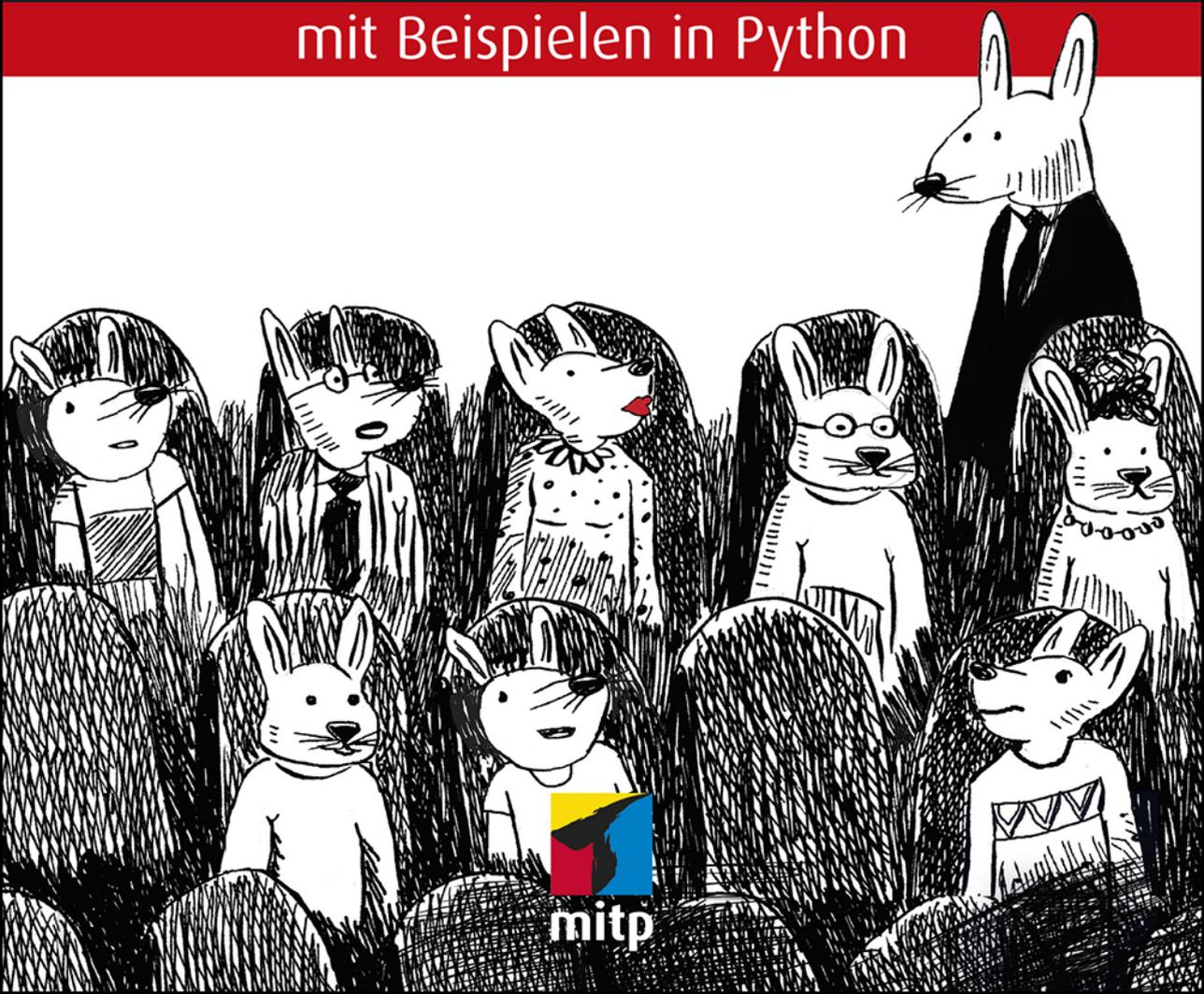


ANDREW W. TRASK

NEURONALE NETZE UND DEEP LEARNING KAPIEREN

DER EINFACHE PRAXISEINSTIEG

mit Beispielen in Python





Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Der Verlag räumt Ihnen mit dem Kauf des ebooks das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und Einspeicherung und Verarbeitung in elektronischen Systemen.

Der Verlag schützt seine ebooks vor Missbrauch des Urheberrechts durch ein digitales Rechtemanagement. Bei Kauf im Webshop des Verlages werden die ebooks mit einem nicht sichtbaren digitalen Wasserzeichen individuell pro Nutzer signiert.

Bei Kauf in anderen ebook-Webshops erfolgt die Signatur durch die Shopbetreiber. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Neuerscheinungen, Praxistipps, Gratiskapitel,
Einblicke in den Verlagsalltag –
gibt es alles bei uns auf Instagram und Facebook



[instagram.com/mitp_verlag](https://www.instagram.com/mitp_verlag)



[facebook.com/mitp.verlag](https://www.facebook.com/mitp.verlag)

*Für meine Mutter. Du hast so viel Zeit geopfert,
um Tara und mir eine gute Ausbildung zu ermöglichen.
Dein Wirken hat dieses Buch erst ermöglicht.*

*Und für meinen Vater. Danke, dass du uns so lieb hast und dass
du dir die Zeit genommen hast, mir in so jungen Jahren das Program-
mieren beizubringen. Ohne dich gäbe es dieses Buch nicht.*

Es ist eine große Ehre, euer Sohn zu sein.

Andrew W. Trask

Neuronale Netze und Deep Learning kapiieren

Der einfache Praxiseinstieg
mit Beispielen in Python

Übersetzung aus dem Amerikanischen
von Knut Lorenzen



Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-7475-0016-3

1. Auflage 2020

www.mitp.de

E-Mail: mitp-verlag@sigloch.de

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

© 2020 mitp Verlags GmbH & Co. KG, Frechen

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Übersetzung der amerikanischen Originalausgabe

Andrew W. Trask: Grokking Deep Learning

ISBN 978-1617293702

Original English language edition published by Manning Publications USA © 2019 by Manning Publications. German-language edition copyright © 2020 by mitp-Verlag. All rights reserved.

Lektorat: Janina Bahlmann

Sprachkorrektorat: Petra Heubach-Erdmann

Coverbild: Leslie Haimes

Satz: III-satz, Husby, www.drei-satz.de

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

Vorwort



Deep Learning kapieren ist das Ergebnis von drei Jahren harter Arbeit. Ich habe mindestens doppelt so viele Seiten geschrieben, wie dieses Buch in der vorliegenden Form aufweist. Ein halbes Dutzend Kapitel mussten drei oder vier Mal völlig neu verfasst werden, bevor sie reif für die Veröffentlichung waren, und zwischenzeitlich kamen wichtige Kapitel hinzu, die ursprünglich gar nicht vorgesehen waren.

Von viel größerer Bedeutung ist allerdings, dass ich frühzeitig zwei Entschlüsse gefasst habe, die dieses Buch einzigartig machen: Bis auf grundlegende Arithmetik sind keine mathematischen Kenntnisse erforderlich, und das Buch beruht nicht auf einer der allgemeinen Bibliotheken, die oftmals verbergen können, was eigentlich vor sich geht. Mit anderen Worten: Jeder kann dieses Buch lesen und verstehen, wie Deep Learning tatsächlich funktioniert. Um das zu erreichen, musste ich mir neue Methoden ausdenken, um die grundlegenden Konzepte und Verfahren zu beschreiben, ohne auf höhere Mathematik zurückzugreifen oder ausgeklügelten Code zu verwenden, den jemand anderes geschrieben hat.

Mein Ziel beim Schreiben des Buchs war, den praktischen Einstieg in Deep Learning so einfach wie möglich zu machen. Du liest nicht einfach nur über die Theorie des Deep Learnings, du entdeckst es selbst. Um dich dabei zu unterstützen, habe ich eine Menge Code geschrieben und mir große Mühe gegeben, ihn in der richtigen Reihenfolge zu erklären, damit die für die lauffähigen Demos erforderlichen Codeschnipsel einen Sinn ergeben.

Zusammen mit der Theorie, dem Code und den Beispielen, die du in diesem Buch erkunden wirst, ermöglicht dir dieses Wissen, viel schneller zu experimentieren. Du wirst rasch Erfolge erzielen, dir stehen bessere Karrieremöglichkeiten offen und du wirst fortgeschrittenere Deep-Learning-Konzepte schneller verstehen.

In den vergangenen drei Jahren habe ich nicht nur dieses Buch geschrieben, sondern auch an einem Doktorandenprogramm in Oxford teilgenommen, bin dem Team bei Google beigetreten und habe an der Entwicklung von OpenMined mitgewirkt, einer dezentralisierten Plattform für künstliche Intelligenz. Dieses Buch fasst also Jahre des Denkens, des Lernens und des Lehrens zusammen.

Es gibt eine Vielzahl von Ressourcen, um Deep Learning zu lernen. Es freut mich, dass du dich für diese hier entschieden hast.

Einleitung



Dieses Buch soll dir die Grundlagen des Deep Learnings vermitteln, damit du eines der großen Deep-Learning-Frameworks verwenden kannst. Wir konzentrieren uns zunächst auf elementare neuronale Netze und werfen anschließend einen Blick auf komplexere Schichten und Architekturen.

Wer sollte dieses Buch lesen?

Ich habe das Buch bewusst so geschrieben, dass es möglichst leicht verständlich ist und keine Kenntnisse der linearen Algebra, der Infinitesimalrechnung, konvexer Optimierungen oder auch nur des Machine Learnings voraussetzt. Alles aus diesen Themenbereichen, was für das Verständnis des Deep Learnings erforderlich ist, wird erklärt, sobald wir uns damit befassen. Wenn du in der Schule mit Mathematik zurechtgekommen bist und schon ein wenig in Python programmiert hast, bist du bereit für dieses Buch.

Überblick

Das Buch hat 16 Kapitel:

- Kapitel 1 konzentriert sich darauf, weshalb du dich mit Deep Learning befassen solltest und was du für den Anfang brauchst.
- Kapitel 2 untersucht grundlegende Konzepte, wie Machine Learning, parametrische und nichtparametrische Modelle oder überwachtes und unüberwachtes Lernen. Hier wird auch das Paradigma »Vorhersagen, Vergleichen, Erlernen« vorgestellt, das in den nachfolgenden Kapiteln immer wieder vorkommt.
- Kapitel 3 führt vor, wie du einfache Netze dazu verwenden kannst, Vorhersagen zu treffen, und kommt erstmals zu neuronalen Netzen.

- Kapitel 4 zeigt dir, wie du die in Kapitel 3 getroffenen Vorhersagen bewerten kannst und Fehler aufspürst, um im nächsten Schritt Modelle zu trainieren.
- Kapitel 5 konzentriert sich auf das »Erlernen« aus dem Paradigma »Vorhersagen, Vergleichen, Erlernen«. Anhand eines umfassenden Beispiels untersuchen wir den Lernvorgang.
- In Kapitel 6 wirst du dein erstes »tiefes« neuronales Netz erstellen – inklusive Code.
- Kapitel 7 betrachtet neuronale Netze aus der Vogelperspektive und soll deine Vorstellung davon vereinfachen.
- Kapitel 8 stellt Überanpassung, Dropout-Verfahren und Gradientenabstieg vor und zeigt dir, wie du deine Datenmenge in dem gerade erstellten Netz klassifizierst.
- Kapitel 9 erklärt Aktivierungsfunktionen und wie man sie verwendet, wenn man Wahrscheinlichkeiten modelliert.
- Kapitel 10 stellt neuronale Faltungsnetze oder CNNs (*Convolutional Neural Networks*) vor und stellt den Nutzen von Struktur bei der Vermeidung von Überanpassung heraus.
- Kapitel 11 befasst sich mit der Verarbeitung natürlicher Sprache (*Natural Language Processing*, NLP) und erläutert grundlegende Terminologie und Konzepte des Fachgebiets Deep Learning.
- Kapitel 12 erörtert rekurrente neuronale Netze (RNNs), einen modernen Ansatz, der bei Sequenzmodellierungen in fast allen Fachgebieten aktuell und in der Branche mit am weitesten verbreitet ist.
- Kapitel 13 bietet einen Schnellkurs für das Erstellen eines Deep-Learning-Frameworks von Grund auf.
- Kapitel 14 verwendet dein rekurrentes neuronales Netz, um eine schwierigere Aufgabe in Angriff zu nehmen: Sprachmodellierung.
- In Kapitel 15 geht es um den Datenschutz und um grundlegende Konzepte wie Federated Learning, homomorphe Verschlüsselung und weitere Konzepte, die mit Differential Privacy und sicheren Berechnungen, wenn mehrere Parteien beteiligt sind, zu tun haben.
- Kapitel 16 stellt die Tools und Ressourcen bereit, die dir beim weiteren Erkunden des Deep Learnings gute Dienste leisten werden.

Konventionen und Downloads

Für den im Buch abgedruckten Code wird eine nicht-proportionale Schrift verwendet, um ihn vom Fließtext zu unterscheiden. Einige der Listings enthalten Anmerkungen, die wichtige Konzepte hervorheben.

Die Codebeispiele kannst du unter <https://github.com/iamtrask/grokking-deep-learning> oder unter www.mitp.de/0015 herunterladen.

Über den Autor

Andrew Trask ist Gründungsmitglied von Digital Reasonings Machine Learning Lab, in dem Deep-Learning-Ansätze zur Verarbeitung natürlicher Sprache, Bilderkennung und Audiotranskription erforscht werden. Innerhalb weniger Monate gelang es Andrew und seinen Forschungskollegen, die besten veröffentlichten Ergebnisse bei der Stimmungsanalyse und der automatisierten Zuordnung von Wörtern eines Textes zu Wortgruppen (Part-of-Speech-Tagging) zu übertreffen. Er hat das größte künstliche neuronale Netz mit mehr als 160 Milliarden Parametern trainiert. Die Ergebnisse hat er zusammen mit seinem Koautor auf der internationalen Konferenz über Machine Learning vorgestellt. Sie wurden im *Journal of Machine Learning* veröffentlicht.

Derzeit ist er als Produktmanager für Text- und Audioanalytik bei Digital Reasoning für die Weiterentwicklung der kognitiven Computing-Plattform *Synthesis* verantwortlich, wobei Deep Learning zu den Kernkompetenzen gehört.

Danksagung

Ich bin allen außerordentlich dankbar, die an der Entstehung des Buchs mitgewirkt haben. An erster Stelle möchte ich mich bei dem tollen Team des amerikanischen Originalverlags Manning bedanken: Bert Bates, der mich lehrte, wie man schreibt, Christina Taylor, die mich drei Jahre lang geduldig zum Fortfahren anhielt, Michael Stephens, dessen Kreativität es dem Buch ermöglichte, schon vor der Veröffentlichung ein Erfolg zu sein, und Marjan Bace, deren Ermunterungen während der ganzen Verzögerungen den Ausschlag gaben.

Das Buch wäre ohne die vielen Beiträge von Lesern per E-Mail, Twitter und GitHub nicht das, was es ist. Ich bin folgenden Personen für ihre Hilfe bei der Verbesserung des Texts und des Codes zu großem Dank verpflichtet: Jascha Swisher, Varun Sudhakar, Francois Chollet, Frederico Vitorino, Cody Hammond, Mauricio Maroto Arrieta, Aleksandar Dragosavljevic, Alan Carter, Frank Hinek, Nicolas Benjamin Hocker, Hank Meisse, Wouter Hibma, Joerg Rosenkranz, Alex Vieira und Charlie Harrington.

Ich danke den Korrektoren, die sich die Zeit genommen haben, das Manuskript während verschiedener Phasen der Entwicklung zu lesen: Alexander A. Myltsev, Amit Lamba, Anand Saha, Andrew Hamor, Cristian Barrientos, Montoya, Eremey Valetov, Gerald Mack, Ian Stirk, Kalyan Reddy, Kamal Raj, Kelvin D. Meeks, Marco Paulo dos Santos Nogueira, Martin Beer, Massimo Ilario, Nancy W. Grady, Peter Hampton, Sebastian Maldonado, Shashank Gupta, Tymoteusz Wołodźko, Kumar Unnikrishnan, Vipul Gupta, Will Fuger und William Wheeler.

Ich danke Mat und Niko bei Udacity, die das Buch in das Nanodegree-Programm aufgenommen haben, was sehr dazu beigetragen hat, es unter angehenden Deep-Learning-Entwicklern bekannt zu machen.

Ich möchte Dr. William Hooper danken, der stets ein offenes Ohr für meine Fragen über Informatik hatte und mich ausnahmsweise an seinem eigentlich schon voll belegten Kurs »Programmieren 1« teilnehmen ließ. Er hat mich dazu inspiriert, eine Laufbahn im Deep Learning zu verfolgen. Ich bin außerordentlich dankbar für all die entgegengebrachte Geduld, die mir sehr geholfen hat.

Und schließlich möchte ich meiner Frau dafür danken, dass sie während all der Nächte und Wochenenden, die ich mit der Arbeit am Buch verbracht habe, so viel Geduld mit mir hatte und dass sie den gesamten Text des Buchs mehrmals eigenhändig redigiert hat und sich um die Erstellung und Pflege des Code-Repositorys bei GitHub gekümmert hat.

Deep Learning: Weshalb man sich damit befassen sollte

1

*Mach' dir keine Sorgen wegen deiner Schwierigkeiten mit der Mathematik.
Ich kann dir versichern, dass meine noch größer sind.*

– Albert Einstein



In diesem Kapitel:

- Weshalb du dich mit Deep Learning befassen solltest
- Warum du dieses Buch lesen solltest
- Was du brauchst, um loszulegen

1.1 Willkommen bei »Deep Learning kapieren«

Du bist im Begriff, dir einige der wichtigsten Fähigkeiten dieses Jahrhunderts anzueignen!

Es ist ein wirklich spannendes Thema, wie du hoffentlich bald selbst feststellen wirst. Deep Learning vereint Machine Learning und künstliche Intelligenz und wird in der Gesellschaft und der Industrie für beträchtliche Umwälzungen sorgen. Die in diesem Buch erörterten Verfahren verändern das alltägliche Leben. Die Optimierung des Motors in deinem Auto oder die Entscheidung, welche Inhalte du dir in sozialen Medien ansiehst – Deep Learning spielt überall eine Rolle, es ist leistungsstark und macht erfreulicherweise sogar Spaß!

1.2 Weshalb du dich mit Deep Learning befassen solltest

Deep Learning ist ein leistungsstarkes Werkzeug für die zunehmende Automatisierung von Intelligenz.

Seit es Menschen gibt, haben sie immer bessere Werkzeuge entwickelt, um ihre Umwelt in den Griff zu bekommen. Deep Learning schlägt in dieser Reihe von Innovationen ein neues Kapitel auf.

Dass dieses Kapitel so spannend ist, liegt vielleicht daran, dass es sich hier eher um eine geistige als um eine mechanische Innovation handelt. Deep Learning versucht, vergleichbar mit verwandten Fachgebieten wie Machine Learning, Intelligenz schrittweise zu automatisieren. In den letzten paar Jahren sind enorme Erfolge und Fortschritte erzielt worden, die frühere Rekorde im Bereich Computer Vision, bei der Spracherkennung, bei maschinellen Übersetzungen und vielen anderen Aufgaben in den Schatten stellen.

In Anbetracht der Tatsache, dass Deep Learning *weitestgehend den gleichen, vom Gehirn inspirierten Algorithmus* (neuronale Netze) *verwendet*, ist es besonders bemerkenswert, dass auf so vielen verschiedenen Gebieten Erfolge erreicht wurden. Auch wenn Deep Learning nach wie vor ein Fachgebiet mit vielen Herausforderungen ist, das aktiv weiterentwickelt wird, haben jüngste Entwicklungen für einige Aufsehen gesorgt: Vielleicht haben wir nicht nur ein mächtiges Werkzeug entdeckt, sondern auch einen Weg gefunden, unseren eigenen Verstand besser zu verstehen.

Deep Learning hat das Potenzial, Arbeitsabläufe in nahezu allen Bereichen zu automatisieren.

Es gibt eine Menge Wirbel um die potenziellen Auswirkungen von Deep Learning, wenn es weiterhin Fortschritte im gegenwärtigen Tempo gibt. Viele der Vorhersagen sind zwar übertrieben, aber eine verdient aus meiner Sicht Beachtung: die Verlagerung von Arbeitsplätzen. Diese Behauptungen sind meiner Meinung nach im Gegensatz zu anderen ernst zu nehmen, denn selbst wenn die mit Deep Learning einhergehenden Innovationen *sofort* zum Stillstand kämen, gäbe es bereits jetzt rund um den Globus enorme Auswirkungen auf diverse Arbeitsplätze. Callcenter-Mitarbeiter, Taxifahrer und Businessanalysten auf unterer Führungsebene sind überzeugende Beispiele für Berufe, für die Deep Learning eine kostengünstige Alternative bieten kann.

Glücklicherweise wandelt sich die Wirtschaft nicht von heute auf morgen, aber in vielerlei Hinsicht haben wir bereits einen Punkt überschritten, ab dem es angesichts der Leistungsfähigkeit der Technologie kein Zurück mehr gibt. Ich kann nur hoffen, dass du und deine Bekannten mithilfe dieses Buchs von einer vor Umwälzungen stehenden Branche in ein Fachgebiet wechseln können, dem Wachstum und Wohlstand bevorstehen: Deep Learning.

Was es bedeutet, ein Mensch zu sein, kannst du herausfinden, indem du versuchst, Intelligenz und Kreativität zu simulieren.

Ich persönlich bin zum Deep Learning gekommen, weil ich es faszinierend finde. Es handelt sich um eine erstaunliche Schnittmenge von Mensch und Maschine. Genau aufzuschlüsseln, was es eigentlich bedeutet, zu denken, zu schlussfolgern

und Neues zu erschaffen, ist erhellend, fesselnd und – für mich jedenfalls – inspirierend. Stell dir vor, dass du über eine Datenmenge verfügst, die sämtliche Bilder enthält, die jemals gemalt worden sind, und sie dazu verwendest, einer Maschine beizubringen, wie Monet zu malen. Es ist verrückt, aber das ist möglich – und es ist unfassbar cool, zu beobachten, wie das funktioniert.

1.3 Ist es schwierig, Deep Learning zu verstehen?

Wie hart musst du arbeiten, bis es eine »ansehnliche« Belohnung gibt?

Das ist meine Lieblingsfrage. Mit einer »ansehnlichen« Belohnung meine ich, zu beobachten, dass etwas von mir Erstelltes von sich aus *lernt*. Dabei zuzusehen, wie etwas mit eigenen Händen Erschaffenes das vollbringt, ist etwas ganz Besonderes. Wenn du das auch so siehst, ist die Antwort einfach. Am Anfang von Kapitel 3 wirst du dein erstes neuronales Netz erstellen. Die einzige damit verbundene Arbeit ist das Lesen der Seiten von hier bis dorthin.

Nach Kapitel 3 folgt die *nächste* ansehnliche Belohnung, wenn du dir ein paar Zeilen Code gemerkt hast und weiterliest bis zur Mitte von Kapitel 4. Das läuft in allen Kapiteln so ab: Du merkst dir einen kleinen Codeabschnitt aus dem vorangegangenen Kapitel, liest das nachfolgende und erhältst dann die Belohnung in Form eines neuen *lernenden* neuronalen Netzes.

1.4 Warum du dieses Buch lesen solltest

Das Buch ermöglicht einen besonders einfachen Einstieg.

Der Grund dafür, dass du dieses Buch lesen solltest, ist der gleiche, aus dem ich es geschrieben habe. Mir ist keine andere Ressource bekannt (Buch, Kurs, Blogbeitragsreihen), die Deep Learning lehrt, *ohne fortgeschrittene mathematische Kenntnisse* (einen Hochschulabschluss in einem mathematischen Bereich) *vorauszusetzen*.

Versteh das nicht falsch: Es gibt gute Gründe dafür, dabei auf Mathematik zurückzugreifen. Mathematik ist letzten Endes eine Sprache, die zweifelsohne ein *effizienteres* Lehren von Deep Learning ermöglicht, aber ich halte es nicht für unbedingt notwendig, fortgeschrittene Mathematikkenntnisse zu besitzen, um ein fähiger und kenntnisreicher Entwickler zu werden, der das »Wie« beim Deep Learning gründlich verstanden hat.

Und weshalb solltest du nun Deep Learning mithilfe dieses Buchs erlernen? Weil ich davon ausgehe, dass du über Mathematikkenntnisse auf Gymnasialniveau verfügst (die auch nicht mehr ganz frisch sind), und ich *alles andere erklären werde*, wenn es vorkommt. Kannst du dich an Multiplikation und x-y-Koordinatensysteme erinnern (die Quadrate mit den Kurven darin)? Perfekt! Das genügt.

Das Buch hilft dir zu verstehen, was *innerhalb* eines Frameworks (wie Torch, TensorFlow usw.) vor sich geht.

Es gibt im Wesentlichen zwei Gruppen von Lehrmaterialien (wie Bücher oder Kurse) über Deep Learning. Die eine Gruppe konzentriert sich auf die Verwendung verbreiteter Frameworks und Codebibliotheken wie Torch, TensorFlow, Keras und andere. Die zweite Gruppe konzentriert sich auf das Deep Learning selbst, also auf die *wissenschaftlichen Aspekte hinter den Kulissen* der großen Frameworks.

Letzten Endes sind Kenntnisse über beide Gruppen wichtig. Wenn du ein NASCAR-Fahrer werden willst, musst du nicht nur dein Automodell kennen (das Framework), sondern auch wissen, wie man fährt (die Wissenschaft und Kenntnisse). Sich nur mit dem Framework zu befassen, ist damit vergleichbar, die Vor- und Nachteile eines Chevrolet SS der 6. Generation auswendig zu lernen, bevor man überhaupt weiß, was ein Schaltknüppel ist. In diesem Buch geht es darum, zu lehren, was Deep Learning ist, damit du darauf vorbereitet bist, ein Framework zu erlernen.

Alles, was mit Mathematik zu tun hat, wird durch anschauliche Analogien erklärt.

Wenn ich auf mathematische Formeln stoße, unternehme ich die beiden folgenden Schritte: Zuerst übersetze ich das Verfahren in eine anschauliche Analogie. Ich nehme eine Formel nur selten einfach so hin, wie sie ist: Ich zerlege sie in Teile, die für sich selbst genommen etwas aussagen. Diesen Ansatz verfolgt auch das Buch. Wenn es um mathematische Konzepte geht, biete ich eine Analogie an, die erläutert, was eine Formel eigentlich leistet.

Man muss die Dinge so einfach wie möglich machen, aber nicht einfacher.
– Albert Einstein zugeschrieben

Nach den einführenden Kapiteln ist alles projektbasiert.

Eines kann ich beim Lernen gar nicht leiden: wenn ich mich fragen muss, ob das, was ich lerne, nützlich oder von Bedeutung ist. Wenn mir jemand alles über einen Hammer beibringen möchte, ohne mir ein paar Nägel in die Hand zu drücken, die ich dann einschlagen kann, dann bringt man mir nicht wirklich bei, wie man einen Hammer benutzt. Ich weiß, dass noch irgendetwas fehlt, und wenn ich dann mit einem Hammer und einer Schachtel Nägel vor einem Stapel Kantholz stehe, bin ich auf Mutmaßungen angewiesen.

In diesem Buch geht es darum, dir Holz, Nägel und Hammer bereitzustellen, *bevor* erklärt wird, was man damit anfangen kann. Bei jeder Übung stehen Werkzeug und Arbeitsmaterial zur Verfügung, und wie etwas funktioniert, wird erklärt, während es erledigt wird. Auf diese Weise stehst du am Ende nicht nur mit einer

Liste der vorhandenen Deep-Learning-Tools da, sondern verfügst über die Fähigkeit, sie einzusetzen, um Aufgaben zu lösen. Darüber hinaus lernst du das, was am wichtigsten ist, nämlich wann und weshalb ein Werkzeug für die Aufgaben, die du lösen möchtest, geeignet ist. Diese Kenntnisse versetzen dich in die Lage, eine Laufbahn in der Forschung oder der Branche einzuschlagen.

1.5 Was du brauchst, um loszulegen

Installiere Jupyter Notebook und die Python-Bibliothek NumPy.

Am liebsten verwende ich Jupyter Notebooks. Beim Erlernen von Deep Learning gehört es – jedenfalls für mich – zu den wichtigsten Dingen, das Training eines Netzes abbrechen zu können, um es bis ins kleinste Detail zu untersuchen. Hier erweisen sich Jupyter Notebooks als außerordentlich nützlich.

Was NumPy betrifft, werden wir nur eine Matrix-Bibliothek verwenden. Auf diese Weise lernst du, wie alles *funktioniert*, nicht nur, wie man ein Framework benutzt. Dieses Buch lehrt Deep Learning von Grund auf, von A bis Z.

Installationsanleitungen für die beiden Tools sind unter <http://jupyter.org> und <http://numpy.org> zu finden. Ich habe die Beispiele mit Python 2.7 erstellt, sie aber auch unter Python 3 getestet. Zur einfachen Installation empfehle ich außerdem das Framework Anaconda: <https://docs.continuum.io/anaconda/install>.

Mathematik auf Gymnasialniveau

Ein Teil der Mathematik geht über den Rahmen dieses Buchs hinaus, aber mein Ziel ist es, Deep Learning nur mithilfe von grundlegenden Algebra-Kenntnissen zu lehren.

Suche nach einer Aufgabe, an der du persönlich interessiert bist.

Um loszulegen, ist das eigentlich keine echte Notwendigkeit. Aber im Ernst: Ich kann nur dringend empfehlen, sich eine solche Aufgabe zu suchen. Alle mir bekannten Leute, die auf diesem Gebiet Erfolg hatten, wollten eine ganz bestimmte Aufgabe lösen. Sich mit Deep Learning zu befassen, war eigentlich nur ein Mittel zum Zweck, um eine andere interessante Aufgabe zu lösen.

Ich wollte Twitter verwenden, um Aktienkurse vorherzusagen. Für mich war das einfach faszinierend und mein Antrieb, das nächste Kapitel zu lesen und den nächsten Prototyp zu entwickeln.

Wie sich herausstellte, ist dieses Fachgebiet noch *so neu* und ändert sich *so schnell*, dass man eher, als man denkt, zu einem führenden Experten für eine bestimmte Aufgabe wird, wenn man sich die nächsten Jahre damit befasst, sein Ziel zu ver-

folgen. In meinem Fall führte das dazu, dass ich innerhalb von 18 Monaten von jemandem, der praktisch nichts über Programmierung wusste, zu einem Forschungsstipendiaten mit dem Schwerpunkt Hedgefonds wurde, der das Gelernte anwenden konnte. Beim Deep Learning kann eine faszinierende Aufgabe, die eine Datenmenge nutzt, um eine andere vorherzusagen, ein entscheidender Katalysator sein. Such dir also eine solche Aufgabe!

1.6 Python-Kenntnisse sind nützlich

Python ist meine bevorzugte Programmierumgebung, aber ich stelle online auch einige andere Sprachen zur Verfügung.

Python ist eine erstaunlich intuitive Sprache. Ich denke, Python dürfte die verbreitetste intuitiv verständliche Programmiersprache sein, die es gibt. Darüber hinaus hat die Python-Community ein Faible für Einfachheit, das nicht zu schlagen ist. Aus diesen Gründen werde ich für alle Beispiele Python verwenden (Python 2.7, um genau zu sein, die Beispiele sind aber auch unter Python 3 getestet). Der Quellcode ist unter <https://github.com/iamtrask/Grokking-Deep-Learning> verfügbar.

Wie viel Erfahrung mit der Programmierung ist erforderlich?

Wirf doch einmal einen Blick auf den englischen Kurs *Python Codecademy* (<http://www.codecademy.com/learn/python>). Wenn du das Inhaltsverzeichnis verstehst und die genannten Begriffe dir geläufig sind, bist du bereit! Falls nicht, dann absolviere den Kurs und lies weiter, wenn du damit fertig bist. Der Kurs ist für Einsteiger gedacht und sehr gut gestaltet.

1.7 Zusammenfassung

Wenn du Jupyter Notebooks verwenden kannst und dir die Grundlagen von Python vertraut sind, bist du bereit für das nächste Kapitel! Es ist das letzte Kapitel, das vornehmlich dialogorientiert ist (ohne Programme zu erstellen). Es soll dir einen Eindruck vom Vokabular der Sprache, von Konzepten und Fachgebieten der künstlichen Intelligenz, des Machine Learnings und – am wichtigsten – des Deep Learnings verschaffen.

Grundlegende Konzepte: Wie lernen Maschinen?

2

In fünf Jahren wird Machine Learning für jeden erfolgreichen Börsengang verantwortlich sein.

- Eric Schmidt, Vorstandsvorsitzender bei Google, in einer Rede auf der Cloud Computing Platform Conference 2016



In diesem Kapitel:

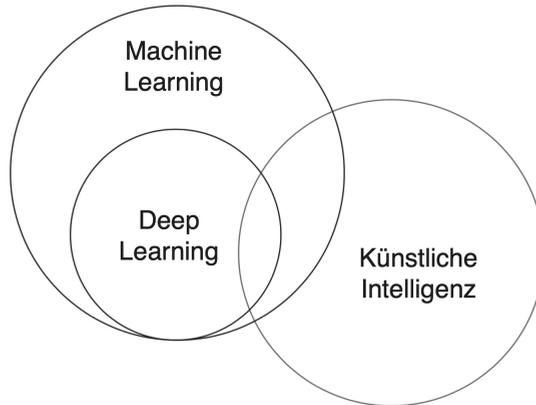
- Deep Learning, Machine Learning und künstliche Intelligenz
- Parametrische und nichtparametrische Modelle
- Überwachtes und unüberwachtes Lernen
- Wie können Maschinen lernen?

2.1 Was ist Deep Learning?

Deep Learning ist eine Teilmenge der Machine-Learning-Verfahren.

Deep Learning ist eine Teilmenge des Machine Learnings, das sich der Untersuchung und der Entwicklung von Maschinen widmet, die lernen können (manchmal auch mit dem Ziel, eine allgemeine künstliche Intelligenz zu erlangen).

In der Industrie wird Deep Learning in verschiedenen Bereichen zum Lösen praktischer Aufgaben eingesetzt, etwa bei Computer Vision (Bilder), bei der Verarbeitung natürlicher Sprache (Text) oder bei der automatischen Spracherkennung (Audio). Deep Learning ist also eine Teilmenge der *Verfahren*, die beim Machine Learning zum Einsatz kommen, wobei vor allem künstliche neuronale Netze verwendet werden, die zu einer Klasse von Algorithmen gehören, die mehr oder weniger durch das menschliche Gehirn inspiriert wurden.



Die Abbildung zeigt, dass es beim Deep Learning nicht nur um allgemeine künstliche Intelligenz geht (wie bei den empfindungsfähigen Maschinen in Spielfilmen). Viele Anwendungen dieser Technologie lösen die verschiedenartigsten Aufgaben. Dieses Buch soll sich auf die Grundlagen des Deep Learnings konzentrieren, das sowohl in der topaktuellen Forschung als auch in der Industrie zum Einsatz kommt, und dir diese Kenntnisse vermitteln.

2.2 Was ist Machine Learning?

Ein Fachgebiet, das Computern die Fähigkeit verleiht, zu lernen, ohne explizit programmiert zu werden.

– Arthur Samuel zugeschrieben

Wenn also Deep Learning eine Teilmenge des Machine Learnings ist, was ist dann eigentlich Machine Learning? Ganz allgemein ist es das, was der Name besagt. Machine Learning ist ein Teilgebiet der Informatik, das sich damit befasst, dass *Maschinen lernen*, Aufgaben zu erledigen, für die sie *nicht explizit programmiert* wurden. Oder kurz und bündig: Maschinen beobachten ein Muster und versuchen, es irgendwie zu imitieren, entweder direkt oder indirekt:

Machine Learning ~ = Affe sieht, Affe tut

Ich erwähne hier direktes und indirektes Imitieren, um die Analogie zu den beiden Haupttypen des Machine Learnings aufzuzeigen: *überwachtes* und *unüberwachtes* Lernen. Überwachtes Machine Learning ist das direkte Imitieren des Musters, das einem Datensatz zu eigen ist, bei einem anderen Datensatz. Es ist der Versuch, eine Eingabedatenmenge in eine Ausgabedatenmenge umzuwandeln. Das kann eine unglaublich mächtige und nützliche Fähigkeit sein. Sieh dir die folgenden Beispiele an (die Eingabedatenmengen sind fett gedruckt, die Ausgabedatenmengen kursiv):

- Verwendung der **Pixel** eines Bilds, um das *Vorhandensein* oder das *Fehlen einer Katze* zu erkennen
- Verwendung der **Filme, die dir gefallen haben**, um *Filme* vorherzusagen, *die dir wahrscheinlich gefallen werden*
- Verwendung der **Wörter**, die jemand benutzt, um festzustellen, ob die Person *fröhlich* oder *traurig* ist
- Verwendung der **Daten einer Wetterstation**, um die *Regenwahrscheinlichkeit* vorherzusagen
- Verwendung von **Motorsensoren**, um die optimalen *Einstellungen* zu ermitteln
- Verwendung von **Nachrichtmeldungen**, um die morgigen *Aktienkurse* vorherzusagen
- Verwendung eines **Eingabewerts** zur Berechnung des *doppelten Werts*
- Verwendung der Rohdaten einer **Audiodatei**, um eine *Transkription* des Inhalts zu erstellen

All diese Aufgaben gehören zum überwachten Lernen. Der Machine-Learning-Algorithmus versucht stets, das Muster, das die beiden Datenmengen verbindet, so zu imitieren, dass *die eine Datenmenge zur Vorhersage der anderen* verwendet werden kann. Stell dir vor, du könntest bei all diesen Beispielen die *Ausgabedatenmenge* nur anhand der *Eingabedatenmenge* vorhersagen – das wäre schon eine beachtliche Leistung.

2.3 Überwachtes Machine Learning

Überwachtes Lernen transformiert Datenmengen.

Überwachtes Lernen ist ein Verfahren, um eine Datenmenge in eine andere umzuwandeln. Wenn du beispielsweise eine Datenmenge namens *Aktienkurse am Montag* hättest, in der die Kurse aller Aktien an den Montagen der letzten zehn Jahre aufgezeichnet sind, und eine zweite, die die Aktienkurse an den Dienstagen desselben Zeitraums enthält, könnte ein überwachter Lernalgorithmus die eine Datenmenge verwenden, um die andere vorherzusagen.



Wenn du einen überwachten Machine-Learning-Algorithmus erfolgreich mit den Daten von zehn Jahren trainiert hast, bist du in der Lage, den Aktienkurs für einen

beliebigen Dienstag in der Zukunft vorherzusagen, wenn dir der Aktienkurs am direkt vorhergehenden Montag bekannt ist. Lege eine kurze Pause ein und denke einen Moment darüber nach.

Überwachtes Machine Learning gehört bei der angewandten künstlichen Intelligenz (der sogenannten schwachen KI) zum Alltag. Es erweist sich als nützlich, um das, *was du weißt*, als Eingabe zu verwenden und schnell in das umzuwandeln, *was du wissen möchtest*. Auf diese Weise können Machine-Learning-Algorithmen die Intelligenz und die Fähigkeiten des Menschen auf unzählige Arten erweitern.

Bei der Nutzung der Ergebnisse von Machine Learning besteht der größte Aufwand darin, einen überwachten Klassifikator zu trainieren. Bei der Entwicklung möglichst genauer überwachter Machine-Learning-Algorithmen kommt typischerweise sogar unüberwachtes Machine Learning (dazu gleich mehr) zum Einsatz.



Im folgenden Teil des Buchs wirst du Algorithmen entwickeln, die Eingabedaten mit bestimmten Eigenschaften entgegennehmen. Sie sind beobachtbar, lassen sich aufzeichnen, und sie sind *nachvollziehbar*. Außerdem lassen sie sich in wertvolle Ausgabedaten umwandeln, für die eine logische Analyse erforderlich ist. All das leistet überwachtes Machine Learning.

2.4 Unüberwachtes Machine Learning

Unüberwachtes Lernen gruppiert deine Daten.

Unüberwachtes und überwachtes Lernen haben etwas gemeinsam: Eine Datenmenge wird in eine andere Datenmenge umgewandelt. Beim unüberwachten Lernen ist die zu transformierende Datenmenge allerdings vorher *nicht bekannt*. Im Gegensatz zum überwachten Lernen gibt es keine »richtige Antwort«, die das Modell reproduzieren soll. Ein unüberwachter Lernalgorithmus sucht einfach nur nach Mustern in den Daten und zeigt an, was er gefunden hat.

Das Aufteilen einer Datenmenge in Gruppen, das sogenannte *Clustering*, ist eine Form des unüberwachten Lernens. Beim Clustering wird eine Menge von Datenpunkten in eine Sequenz von Clusterbezeichnungen transformiert. Wenn eine Aufteilung in zehn Cluster vorgenommen wird, verwendet man für die Bezeichnungen üblicherweise die Zahlen von 1 bis 10. Jedem Datenpunkt wird eine Zahl zugewiesen, die angibt, zu welchem Cluster er gehört. Die Menge aus Datenpunk-