



François
Chollet

Deep Learning mit Python und Keras

Das Praxis-Handbuch
vom Entwickler der Keras-Bibliothek



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.

François Chollet

Deep Learning mit Python und Keras

Das Praxis-Handbuch

vom Entwickler der Keras-Bibliothek

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 978-3-95845-839-0

1. Auflage 2018

www.mitp.de

E-Mail: mitp-verlag@sigloch.de

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

© 2018 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

François Chollet: Deep Learning with Python

ISBN 978-1617294433

Original English language edition published by **Manning Publications** USA © 2017 by Manning Publications. German-language edition copyright © 2018 by mitp-Verlag. All rights reserved.

Lektorat: Sabine Schulz

Sprachkorrektorat: Sibylle Feldmann

Coverbild: © agsandrew / fotolia.com

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

Inhaltsverzeichnis

Einleitung	13
Über dieses Buch	14
Wer sollte dieses Buch lesen?	15
Überblick	15
Erforderliche Hard- und Software	16
Quellcode	17
Das Forum zum Buch	17
Über den Autor	17
Über den Fachkorrektor	17
Danksagungen	18
Teil I Grundlagen des Deep Learnings	19
<hr/>	
1 Was ist Deep Learning?	21
1.1 Künstliche Intelligenz, Machine Learning und Deep Learning.	21
1.1.1 Künstliche Intelligenz	22
1.1.2 Machine Learning	22
1.1.3 Die Repräsentation anhand der Daten erlernen	24
1.1.4 Das »Deep« in Deep Learning.	27
1.1.5 Deep Learning in drei Diagrammen erklärt.	29
1.1.6 Was Deep Learning heute schon leisten kann	31
1.1.7 Schenken Sie dem kurzfristigen Hype keinen Glauben	32
1.1.8 Das Versprechen der KI	33
1.2 Vor Deep Learning: eine kurze Geschichte des Machine Learnings	35
1.2.1 Probabilistische Modellierung.	35
1.2.2 Die ersten neuronalen Netze.	36
1.2.3 Kernel-Methoden	36
1.2.4 Entscheidungsbäume, Random Forests und Gradient Boosting Machines	38
1.2.5 Zurück zu neuronalen Netzen	39
1.2.6 Das Besondere am Deep Learning	40
1.2.7 Der Stand des modernen Machine-Learnings	42

1.3	Warum Deep Learning? Und warum jetzt?	42
1.3.1	Hardware	43
1.3.2	Daten	44
1.3.3	Algorithmen	45
1.3.4	Eine neue Investitionswelle	46
1.3.5	Die Demokratisierung des Deep Learnings	47
1.3.6	Bleibt es so?	47
2	Bevor es losgeht: die mathematischen Bausteine eines NNs	49
2.1	Ein erster Blick auf ein NN	49
2.2	Datenrepräsentationen	54
2.2.1	Skalare (0-D-Tensoren)	54
2.2.2	Vektoren (1-D-Tensoren)	55
2.2.3	Matrizen (2-D-Tensoren)	55
2.2.4	3-D-Tensoren und höherdimensionale Tensoren	55
2.2.5	Die wichtigsten Attribute	56
2.2.6	Bearbeiten von Tensoren mit Numpy	58
2.2.7	Datenstapel	58
2.2.8	Beispiele für Datentensoren aus der Praxis	59
2.2.9	Vektordaten	59
2.2.10	Zeitreihen oder sequenzielle Daten	60
2.2.11	Bilddaten	61
2.2.12	Videodaten	62
2.3	Das Getriebe von NNs: Tensoroperationen	62
2.3.1	Elementweise Operationen	63
2.3.2	Broadcasting	64
2.3.3	Tensorprodukt	66
2.3.4	Tensoren umformen	69
2.3.5	Geometrische Interpretation von Tensoroperationen	70
2.3.6	Eine geometrische Interpretation des Deep Learnings	72
2.4	Der Antrieb von NNs: gradientenbasierte Optimierung	73
2.4.1	Was ist eine Ableitung?	74
2.4.2	Ableitung einer Tensoroperation: der Gradient	76
2.4.3	Stochastisches Gradientenabstiegsverfahren	77
2.4.4	Ableitungen verketteten: der Backpropagation-Algorithmus	80
2.5	Zurück zum ersten Beispiel	81
2.6	Zusammenfassung Kapitel 2	83

3	Einführung in neuronale Netze	85
3.1	Aufbau eines NNs	86
3.1.1	Layer: Bausteine des Deep Learnings	86
3.1.2	Modelle: vernetzte Layer	88
3.1.3	Verlustfunktionen und Optimierer: Konfiguration des Lernvorgangs	88
3.2	Einführung in Keras	89
3.2.1	Keras, TensorFlow, Theano und CNTK	91
3.2.2	Mit Keras entwickeln: eine kurze Übersicht	92
3.3	Einrichtung eines Deep-Learning-Rechners	93
3.3.1	Die bevorzugte Methode zum Ausführen von Deep-Learning-Experimenten: Jupyter-Notebooks	94
3.3.2	Keras zum Laufen bringen: zwei Möglichkeiten	95
3.3.3	Pro und Kontra: Deep Learning in der Cloud ausführen	95
3.3.4	Für Deep Learning geeignete GPUs	96
3.4	Klassifizierung von Filmbewertungen: ein Beispiel für eine Binärklassifizierung	96
3.4.1	Die IMDb-Datensammlung	96
3.4.2	Daten vorbereiten	98
3.4.3	NN erzeugen	99
3.4.4	Validierung des Ansatzes	103
3.4.5	Vorhersagen über neue Daten mit einem trainierten NN treffen	107
3.4.6	Weitere Experimente	108
3.4.7	Zusammenfassung	108
3.5	Ein Beispiel für eine Mehrfachklassifizierung: Klassifizierung von Nachrichtenmeldungen	109
3.5.1	Die Reuters-Datensammlung	109
3.5.2	Daten vorbereiten	110
3.5.3	NN erzeugen	111
3.5.4	Validierung des Ansatzes	113
3.5.5	Vorhersagen über neue Daten treffen	116
3.5.6	Eine weitere Möglichkeit zur Handhabung der Klassenbezeichnungen und der Verlustfunktion	116
3.5.7	Hinreichend große zwischenliegende Layer sind wichtig	117
3.5.8	Weitere Experimente	117
3.5.9	Zusammenfassung	118

3.6	Ein Beispiel für eine Regression: Vorhersage der Kaufpreise von Häusern	118
3.6.1	Die Boston-Housing-Price-Datensammlung	119
3.6.2	Daten vorbereiten	120
3.6.3	NN erzeugen	120
3.6.4	K-fache Kreuzvalidierungen des Ansatzes	121
3.6.5	Zusammenfassung	127
3.7	Zusammenfassung Kapitel 3	127
4	Grundlagen des Machine Learnings	129
4.1	Vier Teilgebiete des Machine Learnings	129
4.1.1	Überwachtes Lernen	129
4.1.2	Unüberwachtes Lernen	130
4.1.3	Selbstüberwachtes Lernen	130
4.1.4	Verstärkendes Lernen	131
4.1.5	Glossar: Klassifizierung und Regression	132
4.2	Bewertung von Machine-Learning-Modellen	133
4.2.1	Trainings-, Validierungs- und Testmengen	133
4.2.2	Worauf zu achten ist	137
4.3	Datenvorverarbeitung, Merkmalerstellung und Erlernen von Merkmalen	138
4.3.1	Datenvorverarbeitung für NNs	138
4.3.2	Merkmalerstellung	140
4.4	Überanpassung und Unteranpassung	142
4.4.1	Das NN verkleinern	143
4.4.2	Regularisierung der Gewichtungen	147
4.4.3	Dropout-Regularisierung	149
4.5	Ein allgemeiner Machine-Learning-Workflow	151
4.5.1	Definition der Aufgabe und Zusammenstellen einer Datenmenge	152
4.5.2	Auswahl eines Erfolgskriteriums	153
4.5.3	Auswahl einer Bewertungsmethode	153
4.5.4	Daten vorbereiten	154
4.5.5	Entwicklung eines Modells, das besser funktioniert als zufälliges Raten	154
4.5.6	Hochskalieren: Entwicklung eines Modells mit Überanpassung	156
4.5.7	Regularisierung des Modells und Abstimmung der Hyperparameter	156
4.6	Zusammenfassung Kapitel 4	157

Teil II	Deep Learning in der Praxis	159
5	Deep Learning und maschinelles Sehen	161
5.1	Einführung in CNNs	161
5.1.1	Die Faltungsoperation	164
5.1.2	Die Max-Pooling-Operation	170
5.2	Ein CNN von Grund auf mit einer kleinen Datenmenge trainieren	172
5.2.1	Die Bedeutung des Deep Learnings für Aufgaben mit kleinen Datenmengen	173
5.2.2	Daten herunterladen	173
5.2.3	NN erzeugen	177
5.2.4	Datenvorverarbeitung	179
5.2.5	Datenaugmentation	184
5.3	Verwendung eines vortrainierten CNNs	189
5.3.1	Merkmalsextraktion	190
5.3.2	Feinabstimmung	202
5.3.3	Zusammenfassung	210
5.4	Visualisierung: Was CNNs erlernen können	210
5.4.1	Visualisierung zwischenliegender Aktivierungen	211
5.4.2	Visualisierung von CNN-Filtern	219
5.4.3	Visualisierung der Heatmaps der Klassenaktivierung	224
5.5	Zusammenfassung Kapitel 5	230
6	Deep Learning, Text und sequenzielle Daten	231
6.1	Textdaten	232
6.1.1	One-hot-Codierung von Wörtern und Zeichen	234
6.1.2	Worteinbettung	237
6.1.3	Zusammengefasst: von reinem Text zu Wort-einbettungen	243
6.1.4	Zusammenfassung	251
6.2	Rekurrente neuronale Netze	252
6.2.1	Ein rekurrenter Layer in Keras	255
6.2.2	LSTM- und GRU-Layer	260
6.2.3	Ein konkretes LSTM-Beispiel in Keras	263
6.2.4	Zusammenfassung	265
6.3	Erweiterte Nutzung rekurrenter neuronaler Netze	265
6.3.1	Temperaturvorhersage	266
6.3.2	Daten vorbereiten	269
6.3.3	Eine vernünftige Abschätzung ohne Machine Learning	272

6.3.4	Ein elementarer Machine-Learning-Ansatz	274
6.3.5	Ein erstes RNN	276
6.3.6	Rekurrentes Dropout-Verfahren zum Verhindern einer Überanpassung.	277
6.3.7	Hintereinanderschaltung rekurrenter Layer.	279
6.3.8	Bidirektionale RNNs.	281
6.3.9	Noch einen Schritt weiter gehen	286
6.3.10	Zusammenfassung.	286
6.4	Verarbeitung von Sequenzen mit CNNs	288
6.4.1	Eindimensionale Faltung sequenzieller Daten.	288
6.4.2	Eindimensionales Pooling sequenzieller Daten.	289
6.4.3	Implementierung eines eindimensionalen CNNs.	290
6.4.4	Lange Sequenzen mit einer Kombination aus CNNs und RNNs verarbeiten	293
6.4.5	Zusammenfassung.	297
6.5	Zusammenfassung Kapitel 6	298
7	Bewährte Verfahren des Deep Learnings.	299
7.1	Jenseits des Sequential-Modells: die funktionale Keras-API	299
7.1.1	Einführung in die funktionale API	303
7.1.2	Modelle mit mehreren Eingaben.	305
7.1.3	Modelle mit mehreren Ausgaben	308
7.1.4	Gerichtete azyklische Graphen von Layern.	311
7.1.5	Gemeinsam genutzte Gewichtungen von Layern	316
7.1.6	Modelle als Layer	318
7.1.7	Zusammenfassung.	319
7.2	Deep-Learning-Modelle mit Callbacks und TensorBoard untersuchen und überwachen	319
7.2.1	Beeinflussung eines Modells während des Trainings durch Callbacks.	320
7.2.2	Einführung in das Visualisierungs-Framework TensorBoard	325
7.2.3	Zusammenfassung.	332
7.3	Modelle richtig ausreizen	332
7.3.1	Erweiterte Architekturmuster	333
7.3.2	Hyperparameteroptimierung.	336
7.3.3	Ensemblemodelle	339
7.3.4	Zusammenfassung.	341
7.4	Zusammenfassung Kapitel 7	342

8	Generatives Deep Learning	343
8.1	Texterzeugung mit LSTM-Modellen	345
8.1.1	Eine kurze Geschichte generativer RNNs.	345
8.1.2	Wie erzeugt man sequenzielle Daten?	346
8.1.3	Die Bedeutung der Sampling-Strategie	347
8.1.4	Implementierung der LSTM-Texterzeugung für Zeichen	349
8.1.5	Zusammenfassung	356
8.2	DeepDream	356
8.2.1	DeepDream in Keras implementieren	357
8.2.2	Zusammenfassung	364
8.3	Stilübertragung mit dem Neural-Style-Algorithmus	365
8.3.1	Verlustfunktion für den Inhalt	366
8.3.2	Verlustfunktion für den Stil	366
8.3.3	Stilübertragung in Keras	367
8.3.4	Zusammenfassung	376
8.4	Bilderzeugung mit Variational Autoencoders	376
8.4.1	Sampling eines latenten Bilderraums.	376
8.4.2	Konzeptvektoren für das Bearbeiten von Bildern	377
8.4.3	Variational Autoencoders.	378
8.4.4	Zusammenfassung	385
8.5	Einführung in Generative Adversarial Networks	386
8.5.1	Eine schematische GAN-Implementierung	388
8.5.2	Einige nützliche Tricks	389
8.5.3	Der Generator	390
8.5.4	Der Diskriminator	391
8.5.5	Das gegnerische Netz.	392
8.5.6	Training des DCGAN.	393
8.5.7	Zusammenfassung	396
8.6	Zusammenfassung Kapitel 8	396
9	Schlussfolgerungen	397
9.1	Kernkonzepte im Überblick	397
9.1.1	Verschiedene Ansätze der KI.	398
9.1.2	Die Besonderheiten des Deep Learnings	398
9.1.3	Was ist vom Deep Learning zu halten?	399
9.1.4	Wichtige zugrunde liegende Technologien	401
9.1.5	Der allgemeine Machine-Learning-Workflow	402

9.1.6	Wichtige Netzarchitekturen	403
9.1.7	Der Raum der Möglichkeiten	407
9.2	Grenzen des Deep Learnings	409
9.2.1	Das Risiko der Vermenschlichung von Deep-Learning- Modellen	410
9.2.2	Lokale und extreme Verallgemeinerung.	413
9.2.3	Zusammenfassung.	414
9.3	Die Zukunft des Deep Learnings	415
9.3.1	Modelle als Programme	416
9.3.2	Jenseits von Backpropagation und differenzierbaren Layern	418
9.3.3	Automatisiertes Machine Learning	418
9.3.4	Beständiges Lernen und Wiederverwendung modularer Subroutinen	419
9.3.5	Langfristige Aussichten	421
9.4	Auf dem Laufenden bleiben	422
9.4.1	Praktische Erfahrungen sammeln mit Kaggle	423
9.4.2	Aktuelle Entwicklungen auf der arXiv-Website nachlesen.	423
9.4.3	Erkundung des Keras-Ökosystems	424
9.5	Schlusswort	424
A	Installation von Keras und der Erweiterungen unter Ubuntu	425
A.1	Installation der wissenschaftlichen Pakete	426
A.2	Einrichtung der GPU-Unterstützung.	427
A.3	Theano installieren (optional).	428
A.4	Keras installieren	429
B	Jupyter-Notebooks auf einer EC2-GPU-Instanz ausführen	431
B.1	Was sind Jupyter-Notebooks? – Gründe, sie auf AWS-GPUs ausführen	431
B.2	Gründe, auf AWS-Jupyter-Notebooks zu verzichten.	431
B.3	Einrichtung einer AWS-GPU-Instanz	432
B.3.1	Jupyter konfigurieren.	435
B.4	Keras installieren	436
B.5	Lokale Portweiterleitung einrichten	437
B.6	Jupyter mit dem lokalen Browser verwenden	437
	Stichwortverzeichnis	439

Einleitung

Da Sie dieses Buch lesen, ist Ihnen vermutlich bewusst, welchen außerordentlichen Fortschritt Deep Learning auf dem Gebiet der KI (der Künstlichen Intelligenz) in jüngster Zeit darstellt. In nur fünf Jahren hat sich die Bild- und Spracherkennung von einem nahezu unbrauchbaren Zustand zu einem Werkzeug mit fast übermenschlichen Fähigkeiten entwickelt.

Die mit diesem Fortschritt einhergehenden Folgen betreffen fast sämtliche Industriezweige. Für die Anwendung von Deep-Learning-Technologien auf alle erdenklichen Aufgaben müssen wir sie so vielen Menschen wie möglich zugänglich machen – auch Laien, die keine Forscher oder Doktoranden sind. Deep Learning muss radikal demokratisiert werden, damit es sein volles Potenzial ausschöpfen kann.

Als ich im März 2015 die erste Version des Deep-Learning-Frameworks Keras veröffentlichte, hatte ich eigentlich die Demokratisierung der KI gar nicht im Sinn. Ich hatte mehrere Jahre auf dem Gebiet des Machine Learnings geforscht und Keras für meine eigenen Experimente entwickelt. Aber in den Jahren 2015 und 2016 befassten sich immer mehr Menschen mit Deep Learning. Viele davon entschieden sich für Keras, weil es das für Einsteiger am einfachsten verwendbare Framework war und noch immer ist. Nachdem ich beobachten konnte, dass Scharen von Neulingen Keras auf unerwartete, leistungsstarke Weise einsetzten, wurde mir klar, dass ich mich um die Zugänglichkeit und Demokratisierung der KI bemühen sollte. Ich stellte fest, dass diese Technologien umso nützlicher und wertvoller wurden, je weiter sie sich verbreiteten. Bei der Entwicklung von Keras wurde die einfache Zugänglichkeit schnell zu einem ausdrücklichen Ziel, und die Entwicklergemeinschaft erzielte in einigen wenigen Jahren bemerkenswerte Erfolge auf diesem Gebiet. Wir ermöglichten Zehntausenden von Entwicklern, Deep Learning einzusetzen, die es ihrerseits zum Lösen bedeutsamer Probleme verwendeten, von denen wir bis vor Kurzem gar nicht wussten, dass es sie überhaupt gibt.

Das Buch, das Sie in Ihren Händen halten, ist ein weiterer Schritt auf dem Weg, Deep Learning für so viele Menschen wie möglich verfügbar zu machen. Es war schon lange überfällig, einen begleitenden Kurs für Keras zu erarbeiten, der nicht nur die Grundlagen des Deep Learnings behandelt, sondern gleichzeitig auch typische Anwendungsmuster und bewährte Verfahren in Keras aufzeigt. Dieses Buch stellt meinen Versuch dar, solch einen Lehrgang anzubieten. Ich habe mich beim Schreiben darauf konzentriert, die dem Deep Learning zugrunde liegenden Konzepte und deren Implementierung so verständlich wie möglich zu schildern. Dabei habe ich keinerlei Vereinfachungen vorgenommen – ich bin der festen Überzeugung, dass

beim Deep Learning keine wirklich schwierigen Konzepte auftreten. Ich hoffe, dass Ihnen dieses Buch von Nutzen sein wird und Ihnen dabei hilft, intelligente Anwendungen für die Aufgaben zu entwickeln, die für Sie von Bedeutung sind.

Über dieses Buch

Dieses Buch richtet sich an Leser, die Deep Learning von Grund auf erlernen oder ihre vorhandenen Kenntnisse vertiefen möchten. Es spielt dabei keine Rolle, ob Sie ein Machine-Learning-Entwickler im Praxiseinsatz, ein Programmierer oder ein Student sind: In diesem Buch ist auch für Sie etwas dabei.

Der Autor gibt Ihnen eine praxisorientierte, interaktive Einführung in das Deep Learning und verzichtet dabei weitgehend auf mathematische Formeln. Stattdessen werden anhand von Codebeispielen quantitative Konzepte verdeutlicht, um eine praktische Vorstellung von den grundlegenden Konzepten des Machine Learnings und des Deep Learnings zu vermitteln.

Sie werden anhand von mehr als 30 Codebeispielen – mit ausführlichen Kommentaren, praxisnahen Empfehlungen und zusammenfassenden Erklärungen – alles erfahren, was Sie benötigen, um Deep Learning zum Lösen konkreter Aufgabenstellungen einzusetzen.

Die Codebeispiele verwenden das auf TensorFlow aufbauende Deep-Learning-Framework Keras für Python. Keras ist eines der beliebtesten und am schnellsten wachsenden Deep-Learning-Frameworks und ist das am besten geeignete Tool zum Einstieg ins Deep Learning.

Nach der Lektüre dieses Buchs werden Sie über solide Kenntnisse des Deep Learnings verfügen und beurteilen können, wann es anwendbar ist und wo es an Grenzen stößt. Sie werden mit dem typischen Workflow bei der Bewältigung von für das Machine Learning geeigneten Aufgaben vertraut sein und wissen, wie sich häufig auftretende Probleme lösen lassen. Sie werden in der Lage sein, praxisnahe Aufgaben vom maschinellen Sehen bis zur Sprachverarbeitung mit Keras anzugehen: Bildklassifizierung, Vorhersage von Zeitreihen, Stimmungsanalyse, Erzeugung von Bildern und Texten und vieles andere mehr.

Wer sollte dieses Buch lesen?

Das Buch wendet sich an Leser, die bereits Programmiererfahrung mit Python haben und die ins Machine Learning und Deep Learning einsteigen möchten. Es kann jedoch auch für viele andere Leser von Nutzen sein:

- Falls Sie Data Scientist sind und Ihnen Machine Learning bereits vertraut ist, bietet das Buch eine solide und praxisnahe Einführung ins Deep Learning, das am schnellsten wachsende und bedeutendste Teilgebiet des Machine Learnings.

- Falls Sie Deep-Learning-Experte sind und sich mit dem Deep-Learning-Framework vertraut machen möchten, werden Sie feststellen, dass dieses Buch den besten verfügbaren Keras-Schnellkurs darstellt.
- Falls Sie Doktorand sind und sich in einer formalen Umgebung mit Deep Learning befassen, werden Sie feststellen, dass dieses Buch eine praxisorientierte Ergänzung Ihrer Ausbildung ist, die Ihnen eine Vorstellung vom Verhalten tiefer neuronaler Netze vermittelt und Sie mit den wichtigsten bewährten Verfahren vertraut macht.

Auch technisch Interessierte, die selbst nicht regelmäßig programmieren, werden das Buch als Einführung in die grundlegenden und weiterführenden Konzepte des Deep Learnings nützlich finden.

Für den Einsatz von Keras sind hinreichende Python-Kenntnisse unverzichtbar. Darüber hinaus ist es zwar keine Bedingung, dass Sie mit der Numpy-Bibliothek vertraut sind, aber durchaus hilfreich. Erfahrung mit Machine Learning oder Deep Learning wird nicht vorausgesetzt: Das Buch erläutert alle erforderlichen Grundlagen. Fortgeschrittene Mathematikkenntnisse sind ebenfalls nicht notwendig – Schulkenntnisse sollten für das Verständnis ausreichen.

Überblick

Das Buch besteht aus zwei Teilen. Sofern Sie noch keine Erfahrung mit Machine Learning haben, sollten Sie zunächst Teil I lesen, bevor Sie sich mit Teil II befassen. Wir werden anfangs einfache Beispiele behandeln und uns im weiteren Verlauf des Buchs zunehmend Verfahren annähern, die dem Stand der Technik entsprechen.

Teil I ist eine allgemeine Einführung in das Deep Learning und erläutert die grundlegenden Zusammenhänge und Begriffe sowie alle erforderlichen Konzepte, die für den Einstieg ins Deep Learning und in neuronale Netze wichtig sind:

- In Kapitel 1 wird das wesentliche Hintergrundwissen rund um die Themen KI, Machine Learning und Deep Learning erläutert.
- Kapitel 2 stellt die für das Verständnis von Deep Learning erforderlichen grundlegenden Konzepte vor: Tensoren, Tensoroperationen sowie Gradientenabstiegs- und Backpropagation-Verfahren. Dieses Kapitel enthält auch das erste Beispiel für ein funktionierendes neuronales Netz.
- Kapitel 3 enthält alles, was Sie wissen müssen, um neuronale Netze zu verwenden: eine Einführung in Keras, das Deep-Learning-Framework der Wahl, einen Leitfaden zum Einrichten Ihres Arbeitsplatzrechners und drei grundlegende Codebeispiele mit ausführlichen Erklärungen. Am Ende dieses Kapitels werden Sie in der Lage sein, einfache neuronale Netze für Klassifizierungs- und Regres-

sionsaufgaben zu trainieren. Außerdem werden Sie eine klare Vorstellung davon haben, was beim Trainieren eigentlich im Hintergrund vor sich geht.

- Kapitel 4 erörtert den typischen Workflow beim Machine Learning. Darüber hinaus lernen Sie einige häufig auftauchende Fallstricke kennen und erfahren, wie man diese umgeht.

Teil II befasst sich ausführlich mit den praktischen Anwendungen des Deep Learnings beim maschinellen Sehen (engl. Computer Vision) und bei der Verarbeitung natürlicher Sprache. Viele der hier vorgestellten Beispiele können Ihnen als Vorlage zum Lösen von Problemen dienen, die Ihnen in der Praxis des Deep Learnings begegnen werden:

- Kapitel 5 untersucht eine Reihe von Beispielen für das maschinelle Sehen und legt dabei einen Schwerpunkt auf die Klassifizierung von Bildern.
- In Kapitel 6 sammeln Sie Erfahrung mit Verfahren zur Verarbeitung sequenzieller Daten, wie etwa Texten und Zeitreihen.
- Kapitel 7 stellt einige erweiterte Verfahren zum Erstellen von Deep-Learning-Modellen vor, die dem gegenwärtigen Stand der Technik entsprechen.
- Kapitel 8 erläutert sogenannte generative Modelle: Deep-Learning-Modelle, die in der Lage sind, Bilder und Texte zu erzeugen, und manchmal erstaunlich künstlerisch wirkende Ergebnisse erzielen.
- Kapitel 9 fasst die im Buch erörterten Inhalte zusammen, wirft einen Blick auf die Beschränkungen des Deep Learnings und erkundet die voraussichtliche Zukunft dieses Fachgebiets.

Erforderliche Hard- und Software

Sämtliche Codebeispiele in diesem Buch verwenden das quelloffene und kostenlose Deep-Learning-Framework Keras (<http://keras.io>). Sie sollten über Zugang zu einem UNIX-System verfügen. Es ist zwar auch möglich, Windows zu verwenden, davon rate ich jedoch ab. In Anhang A ist die Einrichtung vollständig beschrieben.

Darüber hinaus empfehle ich, eine neuere NVIDIA-GPU wie z. B. die TITAN X zu verwenden. Das ist zwar nicht unbedingt notwendig, erleichtert die Arbeit aber ungemein, da die Codebeispiele erheblich schneller ausgeführt werden. Weitere Informationen zur Einrichtung eines Deep-Learning-Rechners finden Sie in Abschnitt 3.3.

Sollte Ihnen kein Zugang zu einem Arbeitsplatzrechner mit einer neueren NVIDIA-GPU zur Verfügung stehen, können Sie stattdessen eine Google-Cloud-Instanz (wie etwa eine n1-standard-8-Instanz mit zusätzlicher NVIDIA Tesla K80) oder von Amazon Web Services (AWS) bereitgestellte GPU-Instanzen (wie z. B.

eine p2.xlarge-Instanz) verwenden. In Anhang B ist ein möglicher Cloud-Workflow ausführlich beschrieben, der via Jupyter-Notebooks eine AWS-Instanz ausführt, auf die Sie per Webbrowser zugreifen können.

Quellcode

Sämtliche Codebeispiele stehen in Form von Jupyter-Notebooks zum Herunterladen bereit (unter <http://www.mitp.de/838> oder <https://github.com/fchollet/deep-learning-with-python-notebooks>).

Das Forum zum Buch

Der amerikanische Originalverlag Manning Publications betreibt unter <https://forums.manning.com/forums/deep-learning-with-python> ein Webforum zum Buch. Dort können Sie auf Englisch Kommentare schreiben, technische Fragen stellen und Hilfe vom Autor und anderen Forumsteilnehmern erhalten. Weitere Information und Hinweise zu den Verhaltensregeln finden Sie unter <https://forums.manning.com/forums/about>.

Über den Autor



François Chollet ist bei Google in Mountain View, Kalifornien, tätig und befasst sich mit Deep Learning. Er ist der Entwickler der Deep-Learning-Bibliothek Keras und hat bedeutende Beiträge zum Machine-Learning-Framework TensorFlow geleistet. Er forscht auf dem Gebiet des Deep Learnings mit den Schwerpunkten maschinelles Sehen und der Anwendung des Machine Learnings auf formales Schließen. Seine Forschungsergebnisse wurden auf

bedeutenden Veranstaltungen des Fachgebiets veröffentlicht, unter anderem auf der *Conference on Computer Vision and Pattern Recognition (CVPR)*, der *Conference on Neural Information Processing Systems (NIPS)*, der *International Conference on Learning Representations (ICLR)* und weiteren.

Über den Fachkorrektor

Alexander Bresk ist Unternehmer, Data Engineer, Machine Teacher, Autor und Basketball-Nerd. Er arbeitet als Senior Data Engineer bei der LOVOO GmbH. Dort beschäftigt er sich mit den Themen Recommendation und Online Segmentation. Seinen Master of Science erwarb Alexander an der HTW Dresden, wo er Angewandte Informationstechnologien studierte. Seit seinem Master im Bereich des

Question Answering forscht er aktiv an Verfahren zur Verarbeitung natürlicher Sprache (NLP), wobei er unter anderem Deep Learning Frameworks einsetzt. Außerdem organisiert er zusammen mit Kollegen und Freunden Meetups, Konferenzen und andere Veranstaltungen in den Bereichen Tech, Machine Learning und Startups. Mit seiner Firma Machine Rockstars berät er Unternehmen beim Einstieg in das Thema Digitalisierung sowie bei der Einführung von Machine Learning.

Alexander Bresks Homepage: <http://alexander.bre.sk>

Danksagungen

Ich danke der Keras-Community dafür, dass sie dieses Buch ermöglicht hat. Inzwischen gibt es mehrere Hundert Entwickler, die Beiträge leisten, und mehr als 200.000 Benutzer. Eure Beiträge und euer Feedback haben Keras zu dem gemacht, was es heute ist.

Außerdem danke ich Google für die Unterstützung des Keras-Projekts. Es war großartig zu erleben, dass Keras als TensorFlows High-Level-API übernommen wurde. Die reibungslose Zusammenführung von Keras und TensorFlow ist sowohl für TensorFlow- als auch für Keras-User von Vorteil und ermöglicht fast allen Entwicklern den Zugang zum Deep Learning.

Ich danke den Mitarbeitern des amerikanischen Originalverlags Manning, die dieses Buch ermöglicht haben: dem Herausgeber Marjan Bace und allen Redaktions- und Produktionsmitgliedern, insbesondere Christina Taylor, Janet Vail, Tiffany Taylor, Katie Tennant, Dottie Marsico und vielen anderen, die hinter den Kulissen tätig waren.

Großer Dank gebührt auch dem von Aleksandar Dragosavljevic geleiteten technischen Korrektorenteam – Diego Acuña Rozas, Geoff Barto, David Blumenthal-Barby, Abel Brown, Clark Dorman, Clark Gaylord, Thomas Heiman, Wilson Mar, Sumit Pal, Vladimir Pasman, Gustavo Patino, Peter Rabinovitch, Alvin Raj, Claudio Rodriguez, Srdjan Santic, Richard Tobias, Martin Verzilli, William E. Wheeler und Daniel Williams – sowie den Mitgliedern des Forums. Sie wiesen auf technische Fehler, unpassende Terminologie und Tippfehler hin und lieferten Themenvorschläge. Alle diese Überprüfungen und die Rückmeldungen zu den im Forum erörterten Themen haben das Manuskript mitgeformt und mitgestaltet.

Was technische Fragen betrifft, gebührt Jerry Gaines, der als technischer Redakteur tätig war, sowie den technischen Korrekturlesern Alex Ott und Richard Tobias besonderer Dank. Bessere technische Redakteure hätte ich mir nicht wünschen können.

Und abschließend möchte ich meiner Frau Maria meinen Dank für die unglaubliche Unterstützung während der Entwicklung von Keras und beim Schreiben dieses Buchs aussprechen.

Teil I

Grundlagen des Deep Learnings

Die Kapitel 1 bis 4 dieses Buchs vermitteln ein grundlegendes Verständnis dessen, was Deep Learning eigentlich ist, was sich damit erreichen lässt und wie es funktioniert. Außerdem lernen Sie den typischen Workflow beim Lösen von Aufgaben mittels Deep Learning kennen. Wenn Sie nicht schon über umfangreiche Kenntnisse auf dem Gebiet des Deep Learnings verfügen, sollten Sie zunächst unbedingt Teil I vollständig lesen, bevor Sie sich mit den praktischen Anwendungen in Teil II befassen.

In diesem Teil:

- **Kapitel 1**
Was ist Deep Learning? 21
- **Kapitel 2**
Bevor es losgeht: die mathematischen Bausteine eines NNs 49
- **Kapitel 3**
Einführung in neuronale Netze 85
- **Kapitel 4**
Grundlagen des Machine Learnings. 129

Was ist Deep Learning?

Die Themen in diesem Kapitel:

- Allgemeine Definitionen und grundlegende Konzepte
- Chronologie der Entwicklung des Machine Learnings
- Entscheidende Faktoren bei der zunehmenden Verbreitung des Deep Learnings und zukünftiges Potenzial

In den vergangenen Jahren sorgte die Künstliche Intelligenz (engl. *Artificial Intelligence* oder kurz AI) für einen weitreichenden Medienrummel. Machine Learning, Deep Learning und KI waren Gegenstand unzähliger Artikel, häufig auch jenseits technologieorientierter Publikationen. Man hat uns eine Zukunft mit intelligenten Chatbots, selbstfahrenden Autos und virtuellen Assistenten versprochen – eine Zukunft, die mitunter düster beschrieben wird, aber auch als eine Utopie, in der menschliche Arbeit selten ist und die meisten wirtschaftlichen Tätigkeiten von Robotern oder KI-Agenten erledigt werden. Für zukünftige und heutige Anwender des Machine Learnings ist es von großer Bedeutung, das Signal in all dem Rauschen zu erkennen, um wirklich weltbewegende Entwicklungen von hochgejubelten Pressemitteilungen unterscheiden zu können. Hier steht nicht weniger als unsere Zukunft auf dem Spiel – eine Zukunft, in der Sie eine aktive Rolle einnehmen müssen: Nach der Lektüre dieses Buchs gehören Sie zu denjenigen, die KI-Agenten entwickeln werden. Setzen wir uns also mit den folgenden Fragen auseinander: Was kann Deep Learning heute schon leisten? Wie bedeutsam ist es? Worauf steuern wir zu? Darf man dem Hype Glauben schenken?

Dieses Kapitel erläutert das wesentliche Hintergrundwissen rund um die Themen KI, Machine Learning und Deep Learning.

1.1 Künstliche Intelligenz, Machine Learning und Deep Learning

Zunächst einmal müssen wir eindeutig definieren, was wir eigentlich meinen, wenn wir von KI sprechen. Was sind KI, Machine Learning und Deep Learning (siehe Abbildung 1.1) eigentlich genau? In welcher Beziehung stehen sie zueinander?

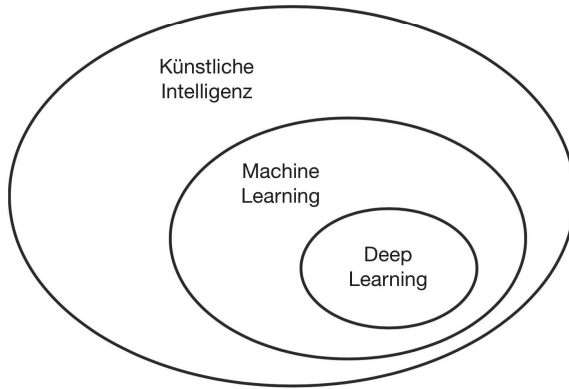


Abb. 1.1: Künstliche Intelligenz, Machine Learning und Deep Learning

1.1.1 Künstliche Intelligenz

Die Künstliche Intelligenz wurde in den 1950er-Jahren entwickelt, als sich einige Pioniere der damals aufblühenden Informatik fragten, ob es möglich sei, einem Computer das »Denken« beizubringen – eine Frage, mit deren Auswirkungen wir uns auch heute noch befassen. Eine kompakte Definition dieses Fachgebiets lautet folgendermaßen: *der Versuch, normalerweise von Menschen erledigte geistige Aufgaben automatisiert zu lösen*. In diesem Sinne ist die KI ein allgemeines Fachgebiet, das Machine Learning und Deep Learning einschließt, aber auch viele andere Ansätze umfasst, die nichts mit Lernen zu tun haben. So verwendeten beispielsweise die ersten Schachprogramme lediglich einen von den Programmierern fest vorgegebenen Regelsatz – dabei handelte es sich jedoch nicht um Machine Learning. Viele Experten gingen lange davon aus, dass sich eine KI auf menschlichem Niveau durch einen hinreichend großen Regelsatz zur Verarbeitung von Wissen erreichen ließe. Dieser Ansatz ist unter der Bezeichnung *symbolische KI* bekannt und war von Mitte der 1950er- bis Ende der 1980er-Jahre das vorherrschende Paradigma der KI. Diese Sichtweise erreichte Mitte der 1980er-Jahre während des Booms der sogenannten *Expertensysteme* ihren Höhepunkt.

Die symbolische KI erwies sich zwar als durchaus brauchbar, um wohldefinierte logische Aufgaben wie etwa Schachspielen zu lösen, es gelang jedoch nicht, explizite Regeln zur Lösung komplexer, weniger deutlich umrissener Aufgabenstellungen zu finden, wie z. B. die Klassifizierung von Bildern, die Erkennung natürlicher Sprache und die Übersetzung von Fremdsprachen. So ergab sich ein neuer Ansatz, der den Platz der symbolischen KI einnehmen sollte: *Machine Learning*.

1.1.2 Machine Learning

Lady Ada Lovelace war im viktorianischen England mit Charles Babbage, dem Erfinder der *Analytical Engine* (engl. für *analytische Maschine*), befreundet und arbeitete

mit ihm zusammen. Die Analytical Engine war visionär und ihrer Zeit weit voraus, allerdings war sie nicht als Allzweckcomputer gedacht, als sie während der 1830er- und 1840er-Jahre entworfen wurde, denn das Konzept eines Allzweckcomputers musste erst noch erfunden werden. Sie war lediglich dafür ausgelegt, bestimmte Berechnungen auf dem Fachgebiet der mathematischen Analyse durch mechanische Vorgänge zu automatisieren – daher auch der Name Analytical Engine. Ada Lovelace kommentierte die Erfindung 1843 folgendermaßen: »Die Analytical Engine beansprucht für sich in keinerlei Weise, Neues zu erschaffen. Sie kann das leisten, von dem wir wissen, wie wir es befehlen können [...] Sie dient dazu, uns dabei zu unterstützen, uns bereits Bekanntes bereitzustellen.«

Diese Anmerkung wurde 1950 von Alan Turing in seiner bahnbrechenden Arbeit *Computing Machinery and Intelligence*¹ zitiert und als »Lady Lovelaces Einspruch« bezeichnet. In dieser Arbeit schlug er den Turing-Test sowie weitere entscheidende die KI prägende Konzepte vor. Turing zitierte Ada Lovelace, während er darüber nachgrübelte, ob Allzweckcomputer in der Lage wären, zu lernen oder originell zu sein. Er kam zu dem Schluss, dass dies der Fall sei.

Machine Learning entstand aufgrund folgender Frage: Könnte ein Computer über das, »von dem wir wissen, wie wir es befehlen können«, hinausgehen und selbst erlernen, wie eine bestimmte Aufgabe erledigt wird? Könnte ein Computer uns überraschen? Könnte ein Computer automatisch Regeln erlernen, indem er Daten betrachtet, ohne dass Programmierer diese Datenverarbeitungsregeln von Hand erstellen müssen?

Diese Fragen öffneten einem neuen Programmierparadigma Tür und Tor. Bei der klassischen Programmierung, der symbolischen KI, geben Menschen Regeln (ein Programm) und die gemäß diesen Regeln zu verarbeitenden Daten vor, was zu Antworten führt (siehe Abbildung 1.2). Beim Machine Learning geben Menschen sowohl die Daten als auch die dazugehörigen Antworten vor, und heraus kommen die Regeln. Diese Regeln sind dann auf neue Daten anwendbar und liefern eigenständige Antworten.

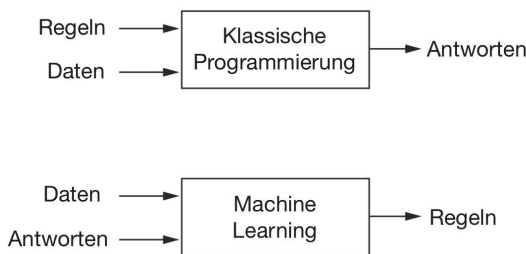


Abb. 1.2: Machine Learning, ein neues Programmierparadigma

1 A. M. Turing, *Computing Machinery and Intelligence*, Mind 59, Nr. 236 (1950), Seiten 433–460.

Ein Machine-Learning-System wird also nicht explizit programmiert, sondern vielmehr *trainiert*. Dem System werden viele für die zu lösende Aufgabe relevante Beispiele bereitgestellt, in denen es nach einer statistischen Struktur sucht, die ihm letztendlich erlaubt, Regeln für die Automatisierung der Aufgabe zu erstellen. Wenn Sie beispielsweise die Verschlagwortung Ihrer Urlaubsfotos automatisieren möchten, könnten Sie dem System Ihre bereits von Menschen verschlagworteten Bilder zur Verfügung stellen. Das System würde dann statistische Regeln erlernen, um bestimmten Fotos bestimmte Schlagwörter zuzuweisen.

So richtig blühte das Machine Learning zwar erst in den 1990er-Jahren auf, es wurde jedoch dank der Verfügbarkeit schnellerer Hardware und größerer Datenmengen rasch zum verbreitetsten und erfolgreichsten Teilgebiet der KI. Machine Learning ist eng mit der mathematischen Statistik verwandt, unterscheidet sich aber in einigen wichtigen Punkten. Im Gegensatz zur Statistik kommen beim Machine Learning tendenziell sehr große, komplexe Datenmengen zum Einsatz (wie z. B. eine Datenmenge, die aus mehreren Millionen Fotos mit jeweils Zehntausenden von Pixeln besteht), für die klassische statistische Verfahren wie eine Bayes'sche Analyse nicht praktikabel wären. Daher spielt die mathematische Theorie beim Machine Learning und insbesondere beim Deep Learning nur eine vergleichsweise kleine – vielleicht zu kleine – Rolle. In diesem praxisorientierten Fachgebiet werden Ideen häufiger empirisch erprobt als theoretisch vorhergesagt.

1.1.3 Die Repräsentation anhand der Daten erlernen

Um Deep Learning zu definieren und um den Unterschied zwischen Deep Learning und anderen Ansätzen des Machine Learnings zu verstehen, müssen wir zunächst einmal eine Vorstellung davon erlangen, wie Machine-Learning-Algorithmen eigentlich funktionieren. Ich habe soeben dargelegt, dass beim Machine Learning anhand von Beispielen für die zu erwartenden Ergebnisse Regeln gesucht werden, um die Verarbeitung von Daten zu erledigen. Für das Machine Learning sind also drei Dinge erforderlich:

- *Eingabedaten* – Bei einer Spracherkennung könnte es sich bei den Eingabedaten beispielsweise um Tondateien handeln, die Sprachaufnahmen enthalten, oder bei der Verschlagwortung von Fotos um Bilddateien.
- *Beispiele für die zu erwartende Ausgabe* – Bei einer Spracherkennung könnten die Beispiele in Form von durch Menschen erstellten Textdateien vorliegen, die den Inhalt der Tondateien wiedergeben. Bei einer Bildererkennung wären die zu erwartenden Ausgaben Kennzeichnungen wie »Hund«, »Katze« usw.
- *Eine Möglichkeit, zu messen, ob der Algorithmus gut funktioniert* – Diese Messung ist erforderlich, um die Abweichungen der aktuellen Ausgabe des Algorithmus von der zu erwartenden Ausgabe zu ermitteln. Das Messergebnis dient als Feedback-Signal zur Anpassung der Funktionsweise des Algorithmus.

Ein Machine-Learning-Modell wandelt die Eingabedaten in sinnvolle Ausgaben um. Dieser Vorgang wird anhand der bekannten Beispiele für Ein- und Ausgaben »erlernt«. Die grundsätzliche Aufgabe beim Machine Learning und beim Deep Learning besteht darin, *Daten sinnvoll umzuwandeln*. Mit anderen Worten: Es müssen sinnvolle *Repräsentationen* der gegebenen Eingabedaten erlernt werden – Repräsentationen, die uns der zu erwartenden Ausgabe näherbringen. Aber bevor wir fortfahren: Was genau ist eine Repräsentation? Im Grunde genommen handelt es sich um eine andere Art, Daten zu betrachten – Daten zu *repräsentieren* oder zu *codieren*. Ein Farbfoto kann beispielsweise im RGB-Format (Rot, Grün, Blau) oder im HSV-Format (*Hue*, *Saturation*, *Value* – Farbwert, Farbsättigung und Helligkeitswert) codiert sein. Dabei handelt es sich um zwei unterschiedliche Repräsentationen derselben Daten. Manche Aufgaben, die mit einer der Repräsentationen schwierig sind, können in einer anderen Repräsentation ganz einfach sein. Die Aufgabe, alle roten Pixel eines Bilds auszuwählen, ist im RGB-Format einfacher, die Aufgabe, die Farbsättigung zu verringern, ist hingegen im HSV-Format einfacher. Bei Machine-Learning-Modellen geht es immer auch darum, angemessene Repräsentationen der Eingabedaten zu finden – Transformationen der Daten, die eine gegebene Aufgabe vereinfachen, wie z. B. eine Klassifizierung.

Betrachten wir ein konkretes Beispiel, nämlich ein aus einer x- und einer y-Achse bestehendes Koordinatensystem sowie einige Punkte, die durch ihre (x, y) -Koordinaten definiert sind (siehe Abbildung 1.3).

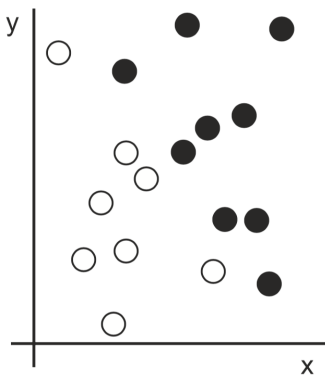


Abb. 1.3: Einige Beispieldaten

Wie Sie sehen, gibt es ein paar weiße und einige schwarze Punkte. Nehmen wir an, wir möchten einen Algorithmus entwickeln, der die (x, y) -Koordinaten eines Punkts entgegennimmt und ausgibt, ob der Punkt wahrscheinlich schwarz oder weiß ist. Hier gilt:

- Die Eingaben sind die Koordinaten der Punkte.
- Die zu erwartenden Ausgaben sind die Farben der Punkte.

- Eine Möglichkeit, zu messen, ob der Algorithmus gut funktioniert, wäre beispielsweise der Prozentsatz der richtig klassifizierten Punkte.

Wir benötigen hier eine neue Repräsentation der Daten, die weiße und schwarze Punkte eindeutig voneinander trennt. Eine von vielen möglichen Transformationen wäre beispielsweise ein Wechsel des Koordinatensystems, wie in Abbildung 1.4 gezeigt.

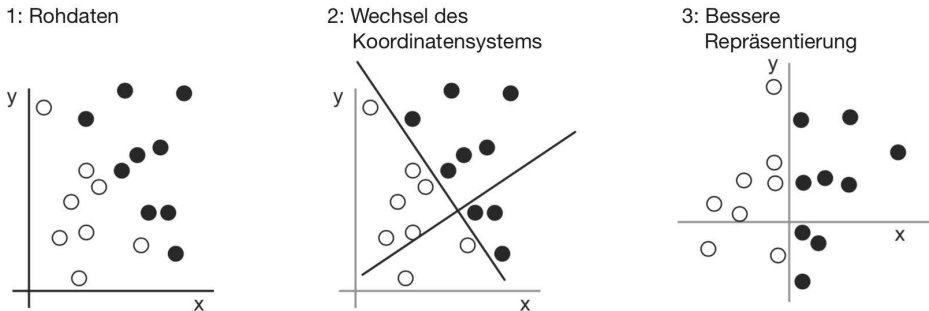


Abb. 1.4: Wechsel des Koordinatensystems

In diesem neuen Koordinatensystem stellen die Koordinaten der Punkte eine andere Repräsentation der Daten dar. Und zwar eine bessere! Bei dieser Repräsentation kann die Klassifizierung der Punkte als schwarz oder weiß durch eine einfache Regel formuliert werden: »Für schwarze Punkte gilt $x > 0$ « oder »Für weiße Punkte gilt $x < 0$.« Diese neue Repräsentation löst die Klassifizierungsaufgabe schon.

In diesem Fall haben wir das Koordinatensystem von Hand neu definiert. Wenn wir stattdessen systematisch nach möglichen Änderungen des Koordinatensystems suchen und den Prozentsatz der korrekt klassifizierten Punkte als Feedback verwenden, betreiben wir Machine Learning. Im Kontext des Machine Learnings beschreibt das *Learning* die automatische Suche nach besseren Repräsentationen.

Machine-Learning-Algorithmen bestehen stets aus der automatischen Suche nach solchen Transformationen, die für eine gegebene Aufgabe nützlichere Repräsentationen der Daten liefern. Bei diesen Operationen kann es sich, wie Sie gerade gesehen haben, um den Wechsel des Koordinatensystems, lineare Projektionen (bei denen Informationen verloren gehen können), Parallelverschiebungen, nicht lineare Operationen (wie etwa »Wähle alle Punkte aus, für die $x > 0$ gilt«) usw. handeln. Machine-Learning-Algorithmen sind bei der Suche nach solchen Transformationen für gewöhnlich nicht sonderlich kreativ, sondern durchsuchen einfach nur eine vorgegebene Menge von Operationen, die als *Hypothesenraum* bezeichnet wird.

Technisch betrachtet, ist Machine Learning also die Suche nach nützlichen Repräsentationen der Eingabedaten in einer vorgegebenen Menge von Möglichkeiten unter Berücksichtigung eines Feedback-Signals. Diese einfache Idee ermöglicht es, geistige Aufgaben von bemerkenswerter Bandbreite zu lösen, die von der Spracherkennung bis zu selbstfahrenden Autos reichen.

Nachdem wir nun geklärt haben, was mit *Learning* gemeint ist, wenden wir uns der Frage nach dem Besonderen des *Deep Learnings* zu.

1.1.4 Das »Deep« in Deep Learning

Deep Learning ist ein Teilgebiet des Machine Learnings: ein neuer Ansatz, die Repräsentationen anhand von Daten zu erkennen, der den Schwerpunkt auf das Erlernen aufeinanderfolgender *Layer* (Schichten) mit zunehmend sinnvolleren Repräsentationen legt. Das *Deep* in Deep Learning bezieht sich also nicht auf irgendein tiefer gehendes durch diesen Ansatz erzielbares Verständnis, sondern steht für das Konzept aufeinanderfolgender Repräsentations-Layer. Die Anzahl der zu einem Datenmodell beitragenden Layer wird als die *Tiefe* des Modells bezeichnet. Man hätte Deep Learning auch als *Lernen durch schichtweise Repräsentationen* oder *Lernen durch hierarchische Repräsentationen* bezeichnen können. Deep Learning umfasst heutzutage oft Dutzende oder sogar Hunderte aufeinanderfolgender Repräsentations-Layer – die alle durch die Bereitstellung der Trainingsdaten automatisch erlernt werden. Andere Ansätze des Machine Learnings konzentrieren sich tendenziell auf nur einen oder zwei Repräsentations-Layer und werden deshalb mitunter als *Shallow Learning* (»flaches« Lernen) bezeichnet.

Beim Deep Learning werden die Repräsentations-Layer (fast immer) durch ein Modell erlernt, das *neuronales Netz* genannt wird (*Neural Network*, ab sofort meistens mit *NN* bezeichnet). NNs sind aus buchstäblich übereinandergestapelten Layern aufgebaut. Der Begriff entstammt zwar der Neurobiologie, aber obwohl einige der grundlegenden Konzepte des Deep Learnings zum Teil durch unser Verständnis vom Gehirn inspiriert wurden, sind Deep-Learning-Modelle *keine* Nachbildungen des Gehirns. Es gibt keinerlei Hinweise darauf, dass das Gehirn irgendwelche Verfahren einsetzt, die den in modernen Deep-Learning-Modellen eingesetzten Lernmechanismen ähneln. In populärwissenschaftlichen Artikeln wird gelegentlich behauptet, Deep Learning funktioniere wie das Gehirn oder sei dem Gehirn nachgebildet worden, aber das stimmt nicht. Für Neulinge in diesem Fachgebiet wäre es verwirrend und kontraproduktiv, wenn sie annähmen, dass Deep Learning irgendetwas mit Neurobiologie zu tun hat. Vergessen Sie die Mythen und Rätsel, die um die Vorstellung »genau wie unser Gehirn« gesponnen wurden, und am besten auch alles, was Sie über hypothetische Zusammenhänge zwischen Deep Learning und Biologie gelesen haben. Für unsere Zwecke ist Deep Learning ein mathematisches Framework zum Erlernen der Repräsentationen anhand von Daten.

Wie sehen die von einem Deep-Learning-Algorithmus erlernten Repräsentationen eigentlich aus? Sehen wir uns doch einmal an, wie ein mehrere Layer umfassendes *Deep Neural Network* (tiefes neuronales Netz, kurz DNN) das Bild einer Ziffer umwandelt, um zu erkennen, um welche Ziffer es sich handelt (siehe Abbildung 1.5).

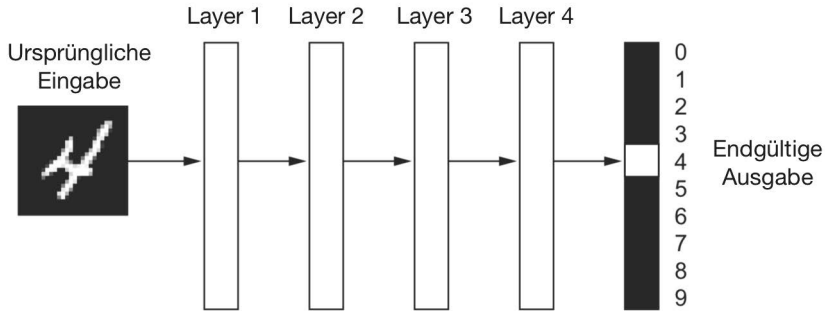


Abb. 1.5: Ein DNN zur Klassifizierung von Ziffern

Der folgenden Abbildung 1.6 können Sie entnehmen, dass das Netz das Bild der Ziffer in Repräsentationen transformiert, die sich zunehmend vom ursprünglichen Bild unterscheiden und bezüglich des Endergebnisses immer aussagekräftiger werden. Stellen Sie sich ein DNN wie eine mehrstufige Operation zum Herausdestillieren von Informationen vor, bei der die Informationen aufeinanderfolgende Filter passieren und dabei immer »reiner« werden, also immer aussagekräftiger bezüglich einer bestimmten Aufgabe.

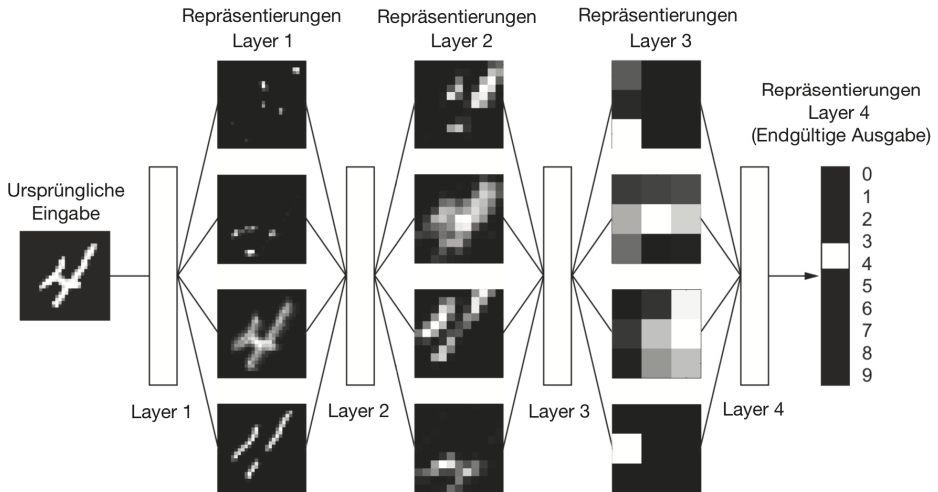


Abb. 1.6: Die von einem Klassifizierungsmodell erlernten Repräsentationen

Technisch betrachtet, ist Deep Learning ein mehrstufiges Verfahren, um Repräsentationen von Daten zu erlernen. Eigentlich eine ganz simple Idee – aber wie sich herausstellen wird, kann dieser einfache Mechanismus wahre Wunder bewirken, wenn er nur hinreichend oft durchlaufen wird.

1.1.5 Deep Learning in drei Diagrammen erklärt

Sie wissen bereits, dass es beim Machine Learning darum geht, Eingaben (wie z.B. Bilder) Zielen (wie etwa der Kennzeichnung »Katze«) zuzuordnen, indem viele Beispiele für Eingaben und Ziele betrachtet werden. Darüber hinaus wissen Sie, dass DNNs diese Zuordnung über eine Sequenz von einfachen Datentransformationen (den Layers) vornehmen und dass diese Transformationen anhand der Beispiele erlernt werden. Sehen wir uns doch einmal an, wie dieses Erlernen konkret stattfindet.

Die Angabe, was ein Layer mit den Eingabedaten anfängt, ist in den *Gewichtungen* des Layers gespeichert. Bei diesen handelt es sich im Wesentlichen um einen Haufen Zahlen. Man spricht hier davon, dass die von einem Layer implementierte Transformation durch die Gewichtungen *parametrisiert* ist (siehe Abbildung 1.7). (Die Gewichtungen werden manchmal auch als die *Parameter* des Layers bezeichnet.) In diesem Zusammenhang bedeutet *Learning*, einen Satz von Werten für die Gewichtungen aller Layer eines Neural Networks zu finden, der dafür sorgt, dass das Netz alle Eingabebeispiele den zugehörigen Zielen korrekt zuordnet. Und hier liegt das Problem: DNNs können zig Millionen Parameter besitzen. Den richtigen Wert für all diese Parameter zu finden, scheint eine entmutigende Aufgabe zu sein, insbesondere in Anbetracht der Tatsache, dass das Modifizieren eines Parameters Auswirkungen auf das Verhalten aller anderen hat!

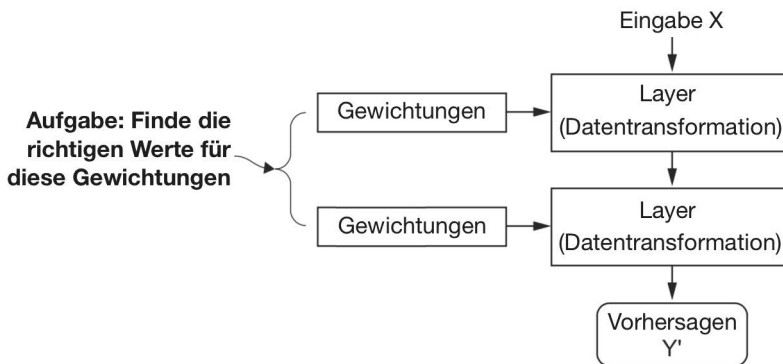


Abb. 1.7: Ein neuronales Netz wird durch seine Gewichtungen parametrisiert.

Man muss zunächst einmal in der Lage sein, etwas beobachten zu können, um es zu steuern. Zur Steuerung der Ausgabe eines neuronalen Netzes muss man messen können, wie stark die Ausgabe von dem erwarteten Wert abweicht. Diese Aufgabe erfüllt die *Verlustfunktion* des neuronalen Netzes, die manchmal auch als *Zielfunktion* bezeichnet wird. Die Verlustfunktion berechnet anhand der Vorhersage des Netzes und des tatsächlichen Zielwerts (des Werts, den das Netz ausgeben sollte) einen Verlustscore, der auf diese Weise erfasst, wie gut das Netz für dieses spezielle Beispiel funktioniert (siehe Abbildung 1.8).

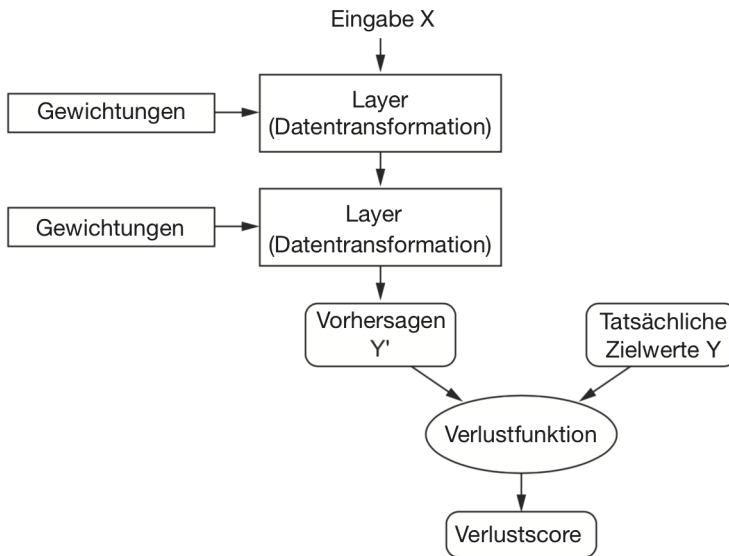


Abb. 1.8: Die Verlustfunktion bemisst die Qualität der Ausgabe eines neuronalen Netzes.

Beim Deep Learning besteht der eigentliche Trick darin, dass dieser Score als Feedback-Signal zur Feinabstimmung der Gewichtungen dient. Die Werte werden so geändert, dass sich der Verlustscore für das aktuelle Beispiel verringert (siehe Abbildung 1.9). Diese Anpassung ist die Aufgabe des *Optimierers*, der einen sogenannten *Backpropagation*-Algorithmus implementiert, den Hauptalgorithmus beim Deep Learning. Im nächsten Kapitel wird die Funktionsweise der Backpropagation ausführlicher erläutert.