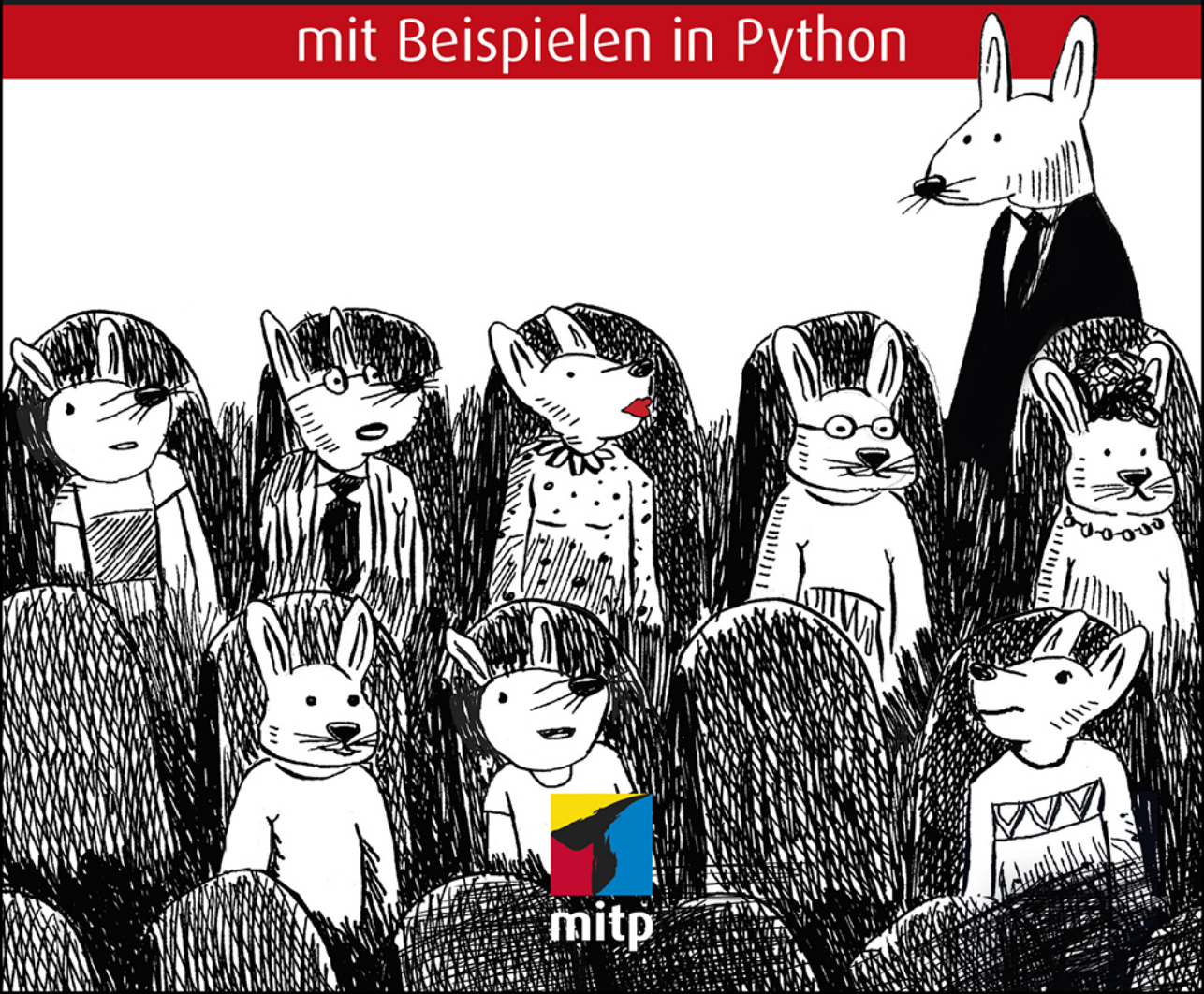
ANDREW W. TRASK

NEURONALE NETZE UND DEEP LEARNING KAPIEREN

DER EINFACHE PRAXISEINSTIEG

mit Beispielen in Python



mitp

Inhaltsverzeichnis

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 *Systhesis* 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.

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:

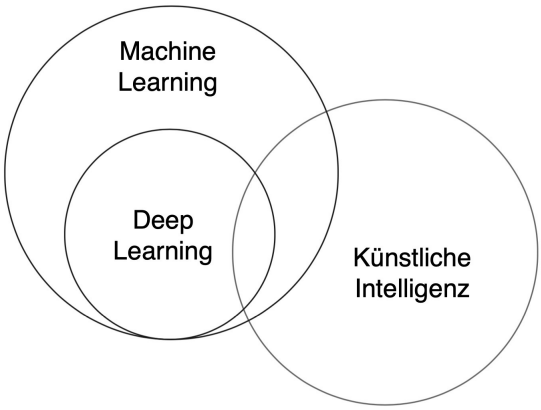
- Deep Learning, Machine Learning und künstliche Intelligenz
- Parametrische und nichtparametrische Modelle
- Überwachtes und unüberwachtes Lernen
- Wie können Maschinen lernen?

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 **Nachrichtenmeldungen**, 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 *Ausgabe*datenmenge nur anhand der *Eingabe*datenmenge 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

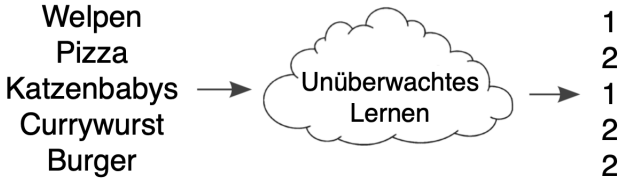
Unüberwachtes Lernen gruppiert deine Daten.

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-

ten wird also in eine Menge von Bezeichnungen umgewandelt. Und warum sind die Bezeichnungen Zahlen? Der Algorithmus sagt nichts darüber aus, was die Cluster eigentlich sind. Er stellt lediglich fest: »Hey, Forscher! Ich habe eine Struktur entdeckt. Sieht ganz danach aus, als ob es Gruppen in den Daten gibt – und zwar diese hier!«

Eine gute Nachricht: Dieses Prinzip des Clusterings kannst du dir als Definition des unüberwachten Lernens merken. Es gibt zwar eine Vielzahl verschiedener Arten, aber *alle Formen des unüberwachten Lernens können als eine Form des Clusterings betrachtet werden*. Mehr zu diesem Thema später im Buch.

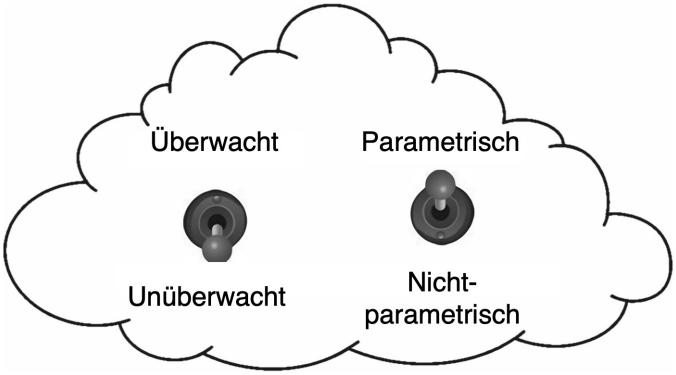


Sieh dir das Beispiel an. Der Algorithmus gibt zwar nicht an, wie die Cluster heißen, aber du kannst sicher erkennen, wonach er die Wörter gruppiert hat. (Antwort: 1 == niedlich, 2 == lecker.) Später werden wir uns damit befassen, dass andere Formen des unüberwachten Lernens auch nur eine Art Clustering sind und weshalb sich diese Cluster für das überwachte Lernen als nützlich erweisen.

2.5 Parametrisches und nichtparametrisches Lernen

Grob vereinfacht: Lernen durch Trial and Error vs. Zählen und Wahrscheinlichkeit

Auf den letzten Seiten wurden alle Machine-Learning-Algorithmen einer von zwei Gruppen zugeordnet: überwachtem und unüberwachtem Lernen. Jetzt werden wir eine andere Variante erörtern, die gleichen Machine-Learning-Algorithmen in zwei Gruppen zu unterteilen, nämlich in parametrische und nichtparametrische. Wenn wir unsere kleine Machine-Learning-Wolke genau betrachten, stellen wir fest, dass sie über zwei Einstellungen verfügt:



Wie du siehst, gibt es eigentlich vier verschiedene Arten von Algorithmen. Ein Algorithmus ist entweder unüberwacht oder überwacht und entweder parametrisch oder nichtparametrisch. Im vorangegangenen Abschnitt über überwachtes Lernen ging es um *die Art des erlernten Musters*, bei der Parametrisierung hingegen geht es darum, wie das Gelernte *gespeichert* wird, und im weiteren Sinne oft um die *Lernmethode*. Sehen wir uns zunächst einmal die formale Definition von parametrischen und nichtparametrischen Algorithmen an. Man hat sich übrigens noch nicht auf einen genauen Unterschied einigen können.

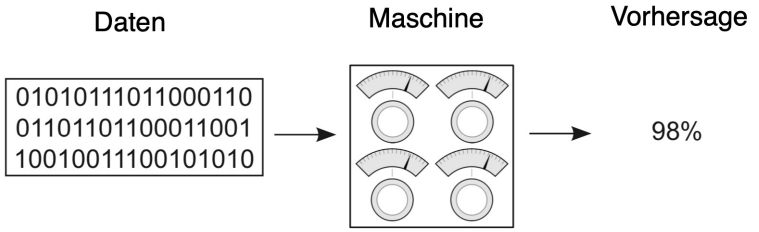
Ein parametrisches Modell ist dadurch gekennzeichnet, dass es eine feste Anzahl von Parametern besitzt, die Anzahl der Parameter eines nichtparametrischen Modells hingegen ist unbegrenzt (sie wird durch die Daten bestimmt).

Betrachten wir als Beispiel die Aufgabe, einen quadratischen Zapfen in eine passende (quadratische) Öffnung zu stecken. Manche Menschen (etwa Kleinkinder) stopfen den Zapfen einfach in alle Öffnungen, bis er irgendwo passt (parametrisch). Ein Teenager hingegen würde vielleicht die Anzahl der Seiten (vier) zählen und nach einer Öffnung suchen, die ebenso viele Seiten besitzt (nichtparametrisch). Parametrische Modelle verwenden tendenziell das Prinzip von Trial and Error, nichtparametrische dagegen versuchen eher, zu zählen. Das sehen wir uns genauer an.

2.6 Überwachtes parametrisches Lernen

Grob vereinfacht: Lernen durch Trial and Error mithilfe von Drehschaltern

Überwachte parametrisch lernende Maschinen besitzen eine festgelegte Anzahl von Drehschaltern (das ist der parametrische Teil), wobei das Lernen durch das Drehen der Schalter erfolgt. Vorliegende Eingabedaten werden entsprechend der Winkel, in dem die Drehschalter stehen, verarbeitet und in eine *Vorhersage* umgewandelt.



Das Lernen vollzieht sich in der Form, dass die Drehschalter in verschiedenen Winkeln positioniert werden. Wenn du versuchst, vorherzusagen, dass Bayern München die Champions League gewinnt, dann würde dieses Modell zuerst Daten entgegennehmen (wie Statistiken über Siege/Niederlagen oder die durchschnittliche Anzahl der Zehen pro Spieler) und eine Vorhersage treffen (etwa eine Chance von 98 Prozent). Anschließend würde das Modell beobachten, ob die Bayern tatsächlich gewinnen. Wenn feststeht, dass sie gewonnen haben, würde der Lernalgorithmus die *Position der Drehschalter aktualisieren*, um beim nächsten Mal eine genauere Vorhersage zu treffen, wenn die *gleichen oder ähnliche Eingabedaten* vorliegen.

Wenn sich der Schalter für »Siege/Niederlagen« bei der Vorhersage als nützlich erwiesen hat, würde das Modell ihn vielleicht etwas »aufdrehen«. Umgekehrt würde es den Schalter für die »durchschnittliche Anzahl der Zehen pro Spieler« »zurückdrehen«, wenn dieser Datenpunkt nicht besonders hilfreich war. Auf diese Weise lernen parametrische Modelle!

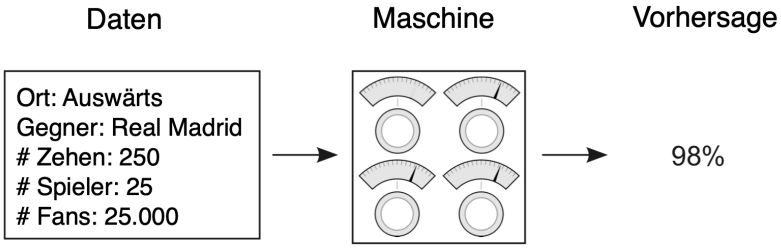
Hier ist zu beachten, dass alles, was das Modell gelernt hat, zu jedem Zeitpunkt in Form der Positionen der Drehschalter erfasst werden kann. Du kannst dir diese Art Lernmodell auch wie einen Suchalgorithmus vorstellen. Du »suchst« nach den geeigneten Stellungen der Drehschalter, indem du Konfigurationen ausprobierst, anpasst und wieder testest.

Außerdem musst du bedenken, dass das Prinzip von Trial and Error nicht der formalen Definition entspricht, es wird in parametrischen Modellen aber (mit Ausnahmen) häufig verwendet. Wenn es eine beliebige (aber feststehende) Anzahl von Drehschaltern gibt, muss eine Suche durchgeführt werden, um die optimale Konfiguration zu finden. Das ist beim nichtparametrischen Lernen anders, das oft auf Zählvorgängen beruht und (hin und wieder) neue Drehschalter hinzufügt, wenn etwas Neues gefunden wird, das gezählt werden kann. Wir unterteilen überwachtes parametrisches Lernen in drei Schritte.

Schritt 1: Vorhersage treffen

Zur Veranschaulichung des überwachten parametrischen Lernens bleiben wir bei der Analogie, vorherzusagen, ob Bayern München die Champions League gewinnt. Der erste Schritt besteht wie erwähnt darin, Statistiken zusammenzutragen, sie in

die Maschine einzuspeisen und eine Vorhersage über die Wahrscheinlichkeit zu treffen, dass die Bayern gewinnen.



Schritt 2: Vergleich mit Faktum

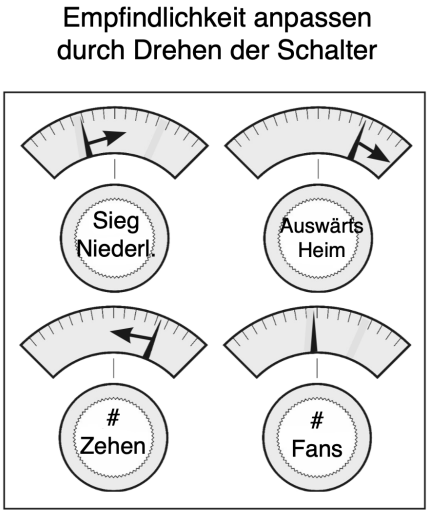
Der zweite Schritt besteht darin, die Vorhersage (98 Prozent) mit dem Faktum zu vergleichen, das von Bedeutung ist (ob Bayern München gewonnen hat). Leider haben sie verloren, somit ergibt sich:

Vorhersage: 98 % > Faktum: 0 %

Dieser Schritt erfasst, dass das Modell die bevorstehende Niederlage genau vorhergesehen hätte, wenn die Vorhersage 0 Prozent Gewinnchance gelautet hätte. Die Maschine soll möglichst genau arbeiten, was zu Schritt 3 führt.

Schritt 3: Erlernen des Musters

Dieser Schritt passt die Position der Drehschalter an und berücksichtigt dabei, wie sehr die Vorhersage daneben gelegen hat (98 Prozent) und was die Eingabedaten zum Zeitpunkt der Vorhersage waren (die Statistiken über Siege/Niederlagen). Auf diese Weise soll anhand der Eingabedaten eine genauere Vorhersage erfolgen.



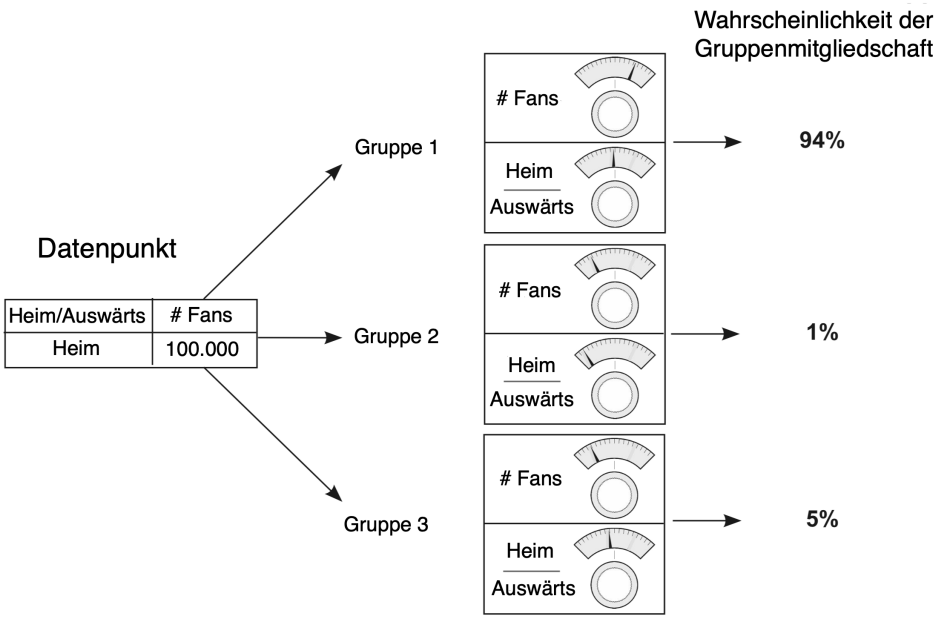
Theoretisch sollte dieser Schritt beim nächsten Vorliegen der gleichen Statistiken über Siege/Niederlagen weniger als 98 Prozent vorhersagen. Beachte auch, dass jeder Drehschalter die *Empfindlichkeit der Vorhersage für verschiedene Typen von Eingabedaten* repräsentiert. Das ist das, was du änderst, wenn die Maschine »lernt«.

2.7 Unüberwachtes parametrisches Lernen

Heim/Auswärts	# Fans
Heim	**100.000**
Auswärts	50.000
Heim	**100.000**
Heim	**99.000**
Auswärts	50.000
Auswärts	*10.000*
Auswärts	*11.000*

Das unüberwachte parametrische Lernen verwendet einen sehr ähnlichen Ansatz. Wir gehen die Schritte einmal der Reihe nach durch. Wie du weißt, geht es beim unüberwachten Lernen immer um das Gruppieren von Daten. Das unüberwachte *parametrische* Lernen verwendet die Drehschalter, um Daten zu gruppieren. Aber in diesem Fall gibt es für jede Gruppe mehrere Drehschalter, die jeweils die Stärke der Zugehörigkeit der Eingabedaten zu einer bestimmten Gruppe angeben (mit Ausnahmen – das ist eine allgemeine Beschreibung). Betrachten wir ein Beispiel, bei dem du die Daten in drei Gruppen unterteilen möchtest.

Die Datenmenge enthält drei Cluster, die das parametrische Modell entdecken soll. Sie sind durch Textformatierung als **Gruppe 1**, Gruppe 2 und *Gruppe 3* gekennzeichnet. Nun übergeben wir einem trainierten unüberwachten Modell den ersten Datenpunkt. Die Zuordnung (auch *Mapping* genannt) zur **Gruppe 1** ist am stärksten.



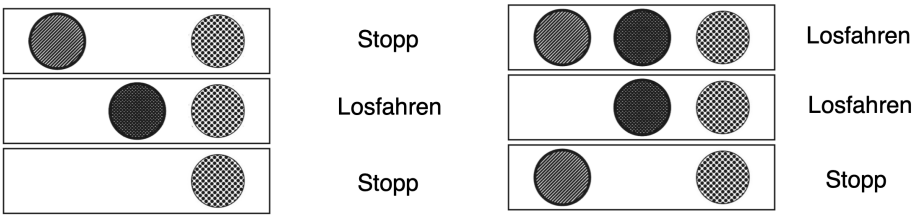
Die Maschine jeder Gruppe versucht, die Eingabedaten in eine Zahl zwischen 0 und 1 umzuwandeln und somit die Wahrscheinlichkeit anzugeben, dass die Einga-

bedaten zu dieser Gruppe gehören. Das Training für die Modelle unterscheidet sich deutlich und führt zu unterschiedlichen Eigenschaften, aber ganz allgemein werden die Parameter angepasst, um die Eingabedaten auf die zugehörige/n Gruppe/n abzubilden.

2.8 Nichtparametrisches Lernen

Grob vereinfacht: Auf Zählvorgängen beruhende Verfahren

Nichtparametrisches Lernen ist eine Algorithmenklasse, bei der die Anzahl der Parameter von den Daten abhängt (und nicht vorab festgelegt ist). Daher werden im Allgemeinen Verfahren verwendet, die auf die eine oder andere Weise zählen und so die Anzahl der Parameter anhand der in den Daten vorhandenen Objekte erhöhen. Bei einem überwachten nichtparametrischen Modell könnte beispielsweise gezählt werden, wie oft das Auftreten einer bestimmten Farbe einer speziellen Ampel dafür sorgt, dass Autos »losfahren«. Schon nach dem Zählen einiger weniger Beispiele wäre das Modell in der Lage vorherzusagen, dass das Aufleuchten der *mittleren* Lampe immer (in 100 Prozent der Fälle) dafür sorgt, dass Autos losfahren und das Aufleuchten der *rechten* Lampe nur manchmal (in 50 Prozent der Fälle).



Dieses Modell hätte drei Parameter: drei Zähler, die angeben, wie oft eine der Lampen aufleuchtet und Autos losfahren (eventuell geteilt durch die Anzahl der Beobachtungen). Bei fünf Lampen gäbe es fünf Zähler (fünf Parameter). Dieses einfache Modell ist *nichtparametrisch*, weil es die Eigenschaft hat, dass die Anzahl der Parameter sich in Abhängigkeit von den Daten (hier die Anzahl der Lampen) ändert. Das ist bei parametrischen Modellen anders, bei denen die Anzahl der Parameter anfangs zwar festgelegt ist, diese aber nach Ermessen des Forschers, der das Modell trainiert, erhöht oder verringert werden kann (unabhängig von den Daten).

Ein kritischer Beobachter könnte das infrage stellen. Beim vorangegangenen parametrischen Modell gab es offenbar für jeden Eingabedatenpunkt einen Drehschalter. Die meisten parametrischen Modelle benötigen irgendeine Art *Eingabe*, die von der Anzahl der Klassen in den Daten abhängt. Es gibt also eine *Grauzone* zwischen parametrischen und nichtparametrischen Algorithmen. Auch parametrische Algo-

rithmen werden in gewisser Weise von der Anzahl der Klassen in den Daten beeinflusst, sogar wenn es sich überhaupt nicht um ein Zählmuster handelt.

Hier wird auch deutlich, dass *Parameter* ein allgemeiner Begriff ist, der sich lediglich auf die Menge der Zahlen bezieht, die zum Modellieren eines Musters verwendet werden (ohne Berücksichtigung, wie diese Zahlen verwendet werden). Zähler sind Parameter. Gewichte sind Parameter. Normalisierte Varianten von Zählern und Gewichten sind ebenfalls Parameter. Korrelationskoeffizienten können auch Parameter sein. Der Begriff bezeichnet die Menge der Zahlen, die zur Modellierung eines Musters verwendet werden. Und Deep Learning ist eine Klasse von parametrischen Modellen. Ich werde in diesem Buch nicht näher auf nichtparametrische Modelle eingehen, aber es handelt sich um eine hochinteressante und leistungsstarke Algorithmenklasse.

2.9 Zusammenfassung

In diesem Kapitel haben wir die verschiedenen Ausprägungen des Machine Learnings etwas genauer betrachtet. Du hast erfahren, dass ein Machine-Learning-Algorithmus entweder überwacht oder unüberwacht und entweder parametrisch oder nichtparametrisch ist. Darüber hinaus haben wir erkundet, was genau diese vier Gruppen von Algorithmen unterscheidet. Du hast gelernt, dass überwachtes Machine Learning zu einer Algorithmenklasse gehört, bei der eine Datenmenge anhand einer anderen vorhergesagt wird, und dass unüberwachtes Lernen im Allgemeinen eine einzelne Datenmenge zu verschiedenen Clustern gruppiert. Zudem hast du erfahren, dass parametrische Algorithmen eine festgelegte Anzahl von Parametern besitzen und dass nichtparametrische Algorithmen die Anzahl der Parameter den Daten entsprechend anpassen.

Deep Learning verwendet sowohl für überwachte als auch für unüberwachte Vorhersagen neuronale Netze. Bislang sind wir auf theoretischer Ebene verblieben, damit du die Bedeutung für das Fachgebiet als Ganzes erkennst. Im nächsten Kapitel wirst du dein erstes neuronales Netz entwickeln, und alle nachfolgenden Kapitel sind *projektbasiert*. Also starte dein Jupyter Notebook und los geht's!

Stichwortverzeichnis