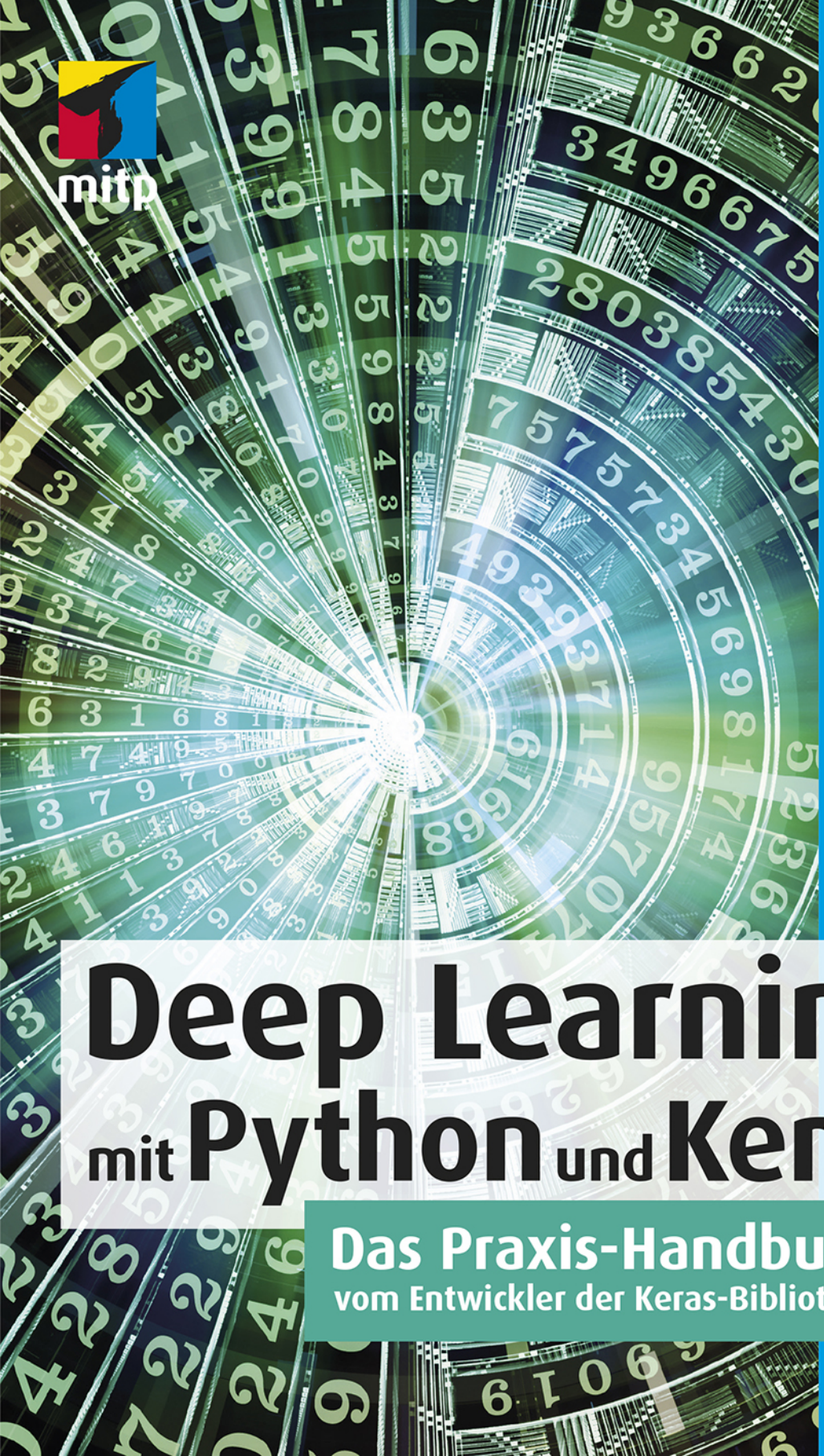

mitp

François Chollet

Deep Learning mit Python und Keras

Das Praxis-Handbuch

vom Entwickler der Keras-Bibliothek

Inhaltsverzeichnis

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 Entwicklergemeinde 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.

Kapitel 1

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?

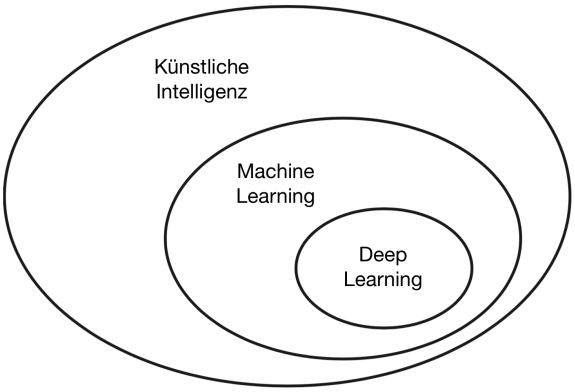

Abb. 1.1: Künstliche Intelligenz, Machine Learning und Deep Learning

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.

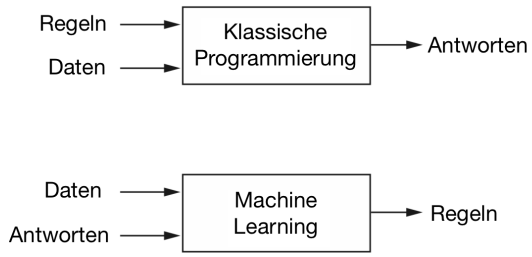

Abb. 1.2: Machine Learning, ein neues Programmierparadigma

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 Bilderkennung 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).

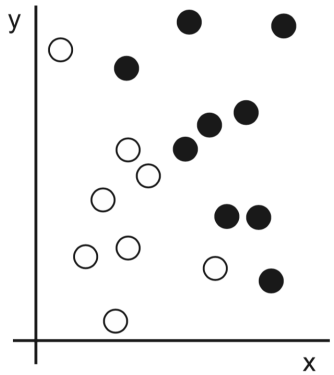

Abb. 1.3: Einige Beispieldaten

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.

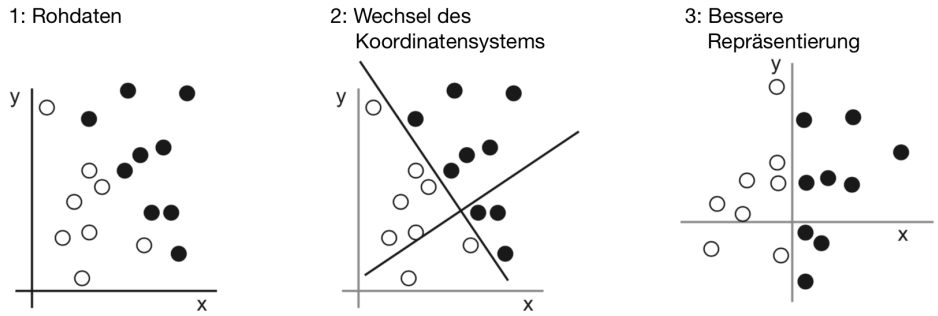

Abb. 1.4: Wechsel des Koordinatensystems

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).

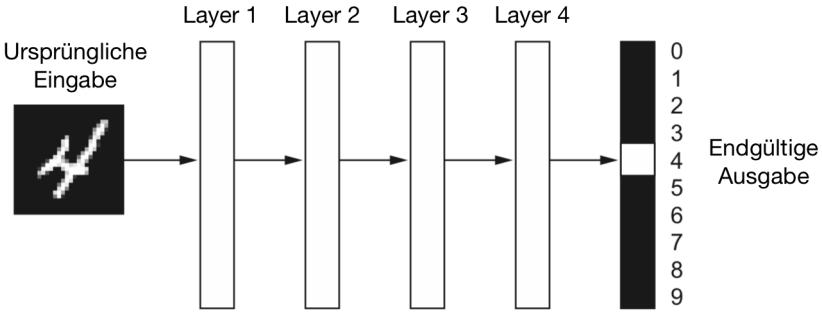

Abb. 1.5: Ein DNN zur Klassifizierung von Ziffern

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.

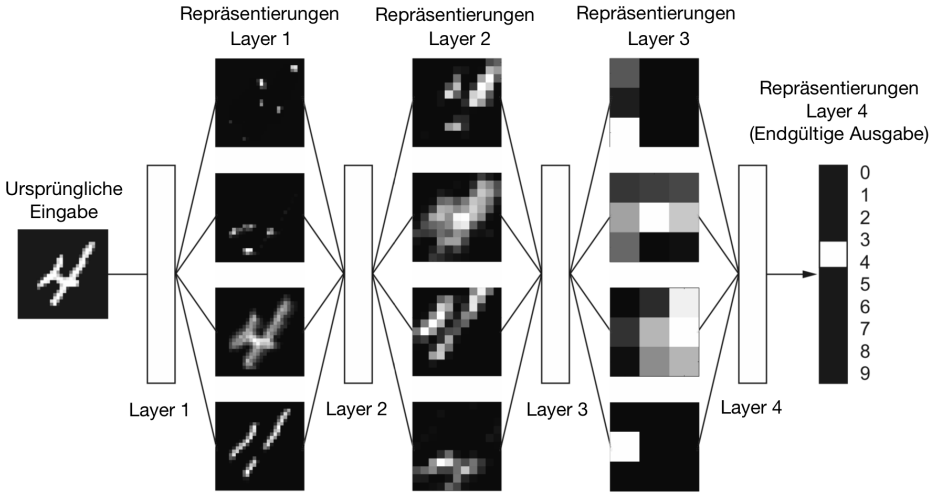

Abb. 1.6: Die von einem Klassifizierungsmodell erlernten Repräsentationen

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 Layern) 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!

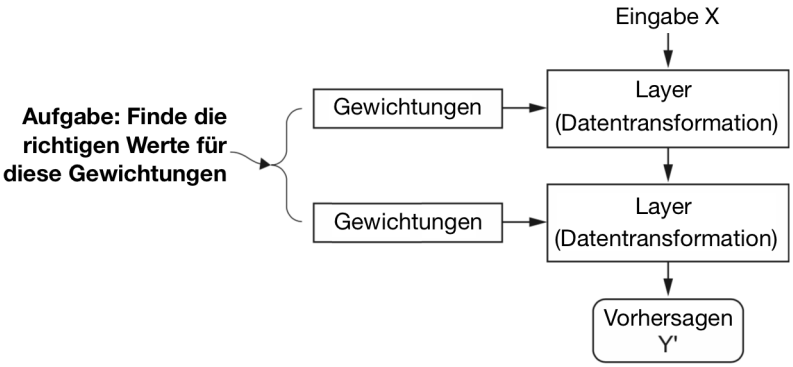

Abb. 1.7: Ein neuronales Netz wird durch seine Gewichtungen parametrisiert.

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).

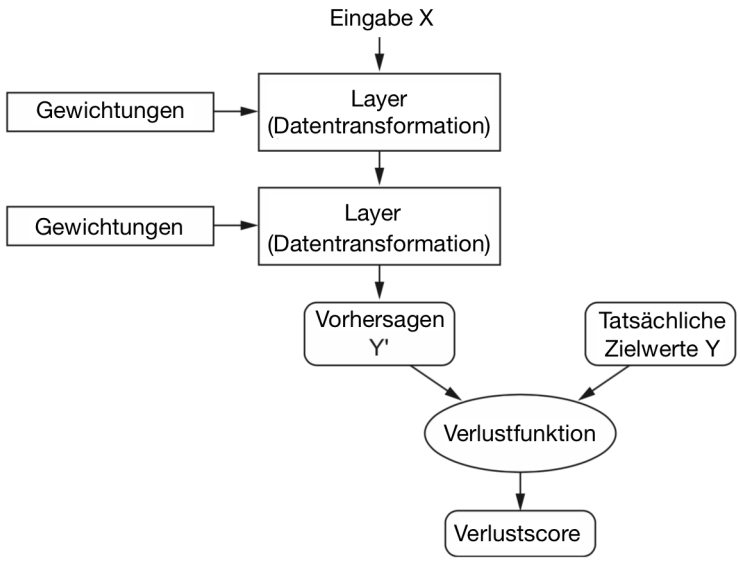

Abb. 1.8: Die Verlustfunktion bemisst die Qualität der Ausgabe eines neuronalen Netzes.

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.

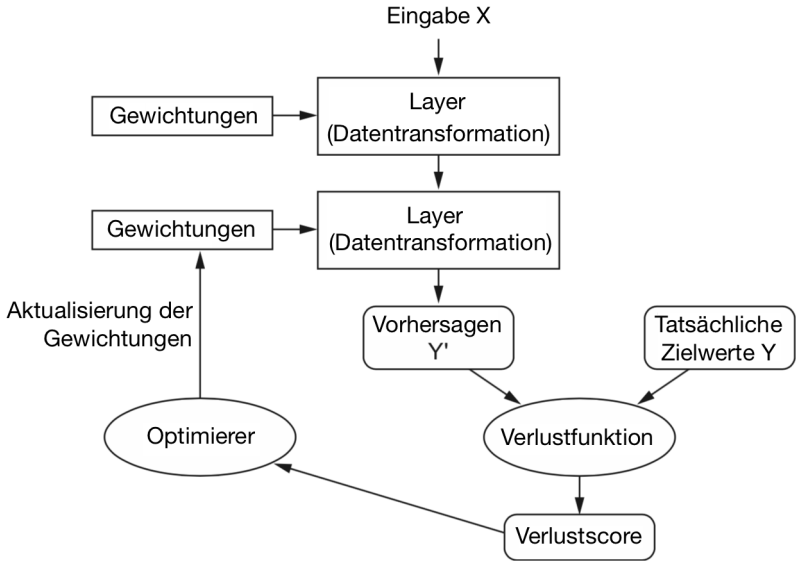

Abb. 1.9: Der Verlustscore dient als Feedback-Signal zur Anpassung der Gewichtungen.

Zunächst werden den Gewichtungen zufällige Werte zugewiesen – das neuronale Netz implementiert also eine Reihe zufälliger Transformationen. Die Ausgabe ist natürlich weit von den idealen Werten entfernt, und der Verlustscore ist anfangs dementsprechend groß. Doch mit jedem weiteren vom neuronalen Netz verarbeiteten Beispiel werden die Gewichtungen so angepasst, dass der Verlustscore abnimmt. Hierbei handelt es sich um die *Trainingsschleife*, die, nachdem sie hinreichend oft durchlaufen wurde (typischerweise etwa zehn Mal mit jeweils Tausenden von Beispielen), Gewichtungen liefert, die die Verlustfunktion minimieren. Bei einem neuronalen Netz mit minimaler Verlustfunktion liegen die Ausgabewerte so nah wie möglich an den Zielwerten: Man spricht in diesem Fall von einem trainierten neuronalen Netz. Tatsächlich ist es dieser einfache Mechanismus, der wahre Wunder bewirken kann, wenn er nur hinreichend oft durchlaufen wird.

1.1.6 Was Deep Learning heute schon leisten kann

Deep Learning ist zwar ein vergleichsweise altes Teilgebiet des Machine Learnings, erlangte aber erst Anfang der 2010er-Jahre größere Bedeutung. In den wenigen seither vergangenen Jahren sorgte es jedoch für eine regelrechte Revolution auf diesem Fachgebiet und erzielte erstaunliche Leistungen bei Aufgaben der Sinneswahrnehmung wie Hören und Sehen – für Menschen natürliche und intuitive Fähigkeiten, die für Maschinen jedoch lange unerreichbar waren.

Mit Deep Learning konnte auf den folgenden traditionell schwierigen Gebieten des Machine Learnings Durchbrüche erzielt werden:

- Bildklassifizierung auf nahezu menschlichem Niveau
- Spracherkennung auf nahezu menschlichem Niveau
- Handschriftenerkennung auf nahezu menschlichem Niveau
- Verbesserung der Übersetzung von Fremdsprachen
- Verbesserung der Sprachsynthese
- Digitale Assistenten wie Google Now oder Amazon Alexa
- Selbstfahrende Autos auf nahezu menschlichem Niveau
- Verbesserung gezielter Werbung, wie sie Google, Baidu und Bing einsetzen
- Verbesserung der Suchergebnisse im Web
- Beantwortung von in natürlicher Sprache gestellten Fragen
- Ein Programm schlägt den besten menschlichen Go-Spieler

Wir sind noch immer damit beschäftigt, das Ausmaß dessen zu erkunden, was mit Deep Learning erreicht werden kann. Inzwischen wird es für eine Vielzahl von Aufgaben jenseits der Sinneswahrnehmung und des Verstehens natürlicher Sprache eingesetzt, wie etwa für das formale Schließen. Wenn sich diese Ansätze als erfolgreich erweisen, könnte das der Beginn eines neuen Zeitalters sein, in dem Deep Learning Menschen bei der Forschung, der Softwareentwicklung und vielem anderen unterstützt.

1.1.7 Schenken Sie dem kurzfristigen Hype keinen Glauben

Deep Learning hat zwar in den letzten Jahren bemerkenswerte Fortschritte erzielt, allerdings sind die Erwartungen in Bezug darauf, was auf diesem Gebiet im kommenden Jahrzehnt erreicht werden kann, viel zu hoch gesteckt. Auch wenn weltbewegende Anwendungen wie selbstfahrende Autos schon in greifbarer Nähe sind, werden viele andere wahrscheinlich noch lange unerreichbar bleiben, wie etwa glaubwürdige Sprachdialogsysteme, Übersetzungen zwischen beliebigen Sprachen oder das Verständnis natürlicher Sprache auf menschlichem Niveau. Vor allem die Berichte über allgemeine Intelligenz auf menschlichem Niveau sollten nicht allzu ernst genommen werden. Hohe Erwartungen, die kurzfristig nicht erfüllt werden, weil die Technologie noch nicht so weit ist, bringen das Risiko mit sich, dass weniger in die Forschung investiert wird und sich der Fortschritt auf diese Weise nachhaltig verlangsamt.

Es wäre nicht das erste Mal. In der Vergangenheit kam es schon zwei Mal vor, dass bezüglich der KI erst immenser Optimismus herrschte, dem Enttäuschung und Skepsis folgten, was zu einem Mangel an Fördergeldern führte. Den Anfang machte die symbolische KI in den 1960er-Jahren. In diesen frühen Tagen wurden hochgesteckte Prognosen bezüglich der KI geäußert. Marvin Minsky war einer der bekanntesten Pioniere und Verfechter der symbolischen KI. 1967 behauptete er:

»Innerhalb der nächsten Generation [...] wird die Aufgabe, eine ›Künstliche Intelligenz‹ zu erschaffen, im Wesentlichen gelöst sein.« Drei Jahre später, also 1970, traf er eine genauere Vorhersage: »In den nächsten drei bis acht Jahren wird es eine Maschine mit der allgemeinen Intelligenz eines durchschnittlichen Menschen geben.« Auch 2017 scheint das Erreichen dieses Ziels noch in ferner Zukunft zu liegen – so fern, dass wir nicht vorhersagen können, wie lange es noch dauern wird. Doch in den 1960er- und frühen 1970er-Jahren waren einige Experten (so wie viele Menschen heutzutage) davon überzeugt, dass der Durchbruch kurz bevorsteht. Nachdem sich die hohen Erwartungen einige Jahre später nicht erfüllt hatten, wendeten sich die Wissenschaftler von diesem Forschungsgebiet ab, und die Regierung strich die Fördergelder. Das war der Anfang des ersten *KI-Winters* (eine Anspielung auf den nuklearen Winter, denn diese Ereignisse fanden kurz nach dem Höhepunkt des Kalten Kriegs statt).

Es sollte nicht der letzte KI-Winter gewesen sein. In den 1980er-Jahren entwickelte sich ein neuer Ansatz für die symbolische KI, die sogenannten Expertensysteme, die in größeren Unternehmen allmählich Fahrt aufnahmen. Einige wenige anfängliche Erfolgsgeschichten lösten eine Investitionswelle aus. Unternehmen rund um den Globus gründeten ihre eigenen KI-Abteilungen, um Expertensysteme zu entwickeln. Mitte der 1980er-Jahre gaben die Unternehmen jedes Jahr mehr als eine Milliarde Dollar für diese Technologie aus. Anfang der 1990er-Jahre stellte sich dann heraus, dass der Unterhalt dieser Systeme kostspielig war, dass sie sich nur schwer skalieren ließen und dass sie lediglich in wenigen Bereichen einsetzbar waren – das Interesse verlief im Sande. Somit begann der zweite KI-Winter.

Möglichweise sind wir gerade Zeugen eines dritten Zyklus von KI-Hype und nachfolgender Enttäuschung – und befinden uns noch in der Phase des immensen Optimismus. Am besten mäßigen wir unsere kurzfristigen Erwartungen und vergewissern uns, dass die mit den technischen Aspekten weniger vertrauten Menschen die richtige Vorstellung davon haben, was mit Deep Learning möglich ist und was nicht.

1.1.8 Das Versprechen der KI

Auch wenn die kurzfristigen Erwartungen an die KI unrealistisch sind, sieht die Zukunft langfristig doch rosig aus. Wir haben gerade erst damit angefangen, Deep Learning auf viele wichtige Aufgabenstellungen anzuwenden, die sich als umwälzend erweisen könnten – von medizinischen Diagnoseverfahren bis zum digitalen Assistenten. Die KI-Forschung hat in den vergangenen fünf Jahren, größtenteils aufgrund einer in der kurzen Geschichte der KI beispiellosen Summe von Fördergeldern, erstaunlich schnell Fortschritte erzielt. Allerdings ist nur relativ wenig von diesem Fortschritt in Form von Produkten oder Verfahren in unserem Alltag angekommen. Die meisten Forschungsergebnisse finden noch keine Anwendung

oder zumindest keine Anwendung auf sämtliche Aufgabenstellungen, die sie in allen Industriezweigen lösen könnten. Ihr Arzt setzt noch keine KI ein, ebenso wenig wie Ihr Buchhalter. Und Sie selbst werden im Alltag vermutlich auch keine KI nutzen. Sie können natürlich Ihrem Smartphone einfache Fragen stellen und vernünftige Antworten erhalten. Auch die Produktempfehlungen von Amazon können ziemlich nützlich sein. Oder Sie geben bei Google Fotos den Suchbegriff »Geburtstag« ein, und augenblicklich werden Ihnen die Bilder von der Geburtstagsfeier Ihrer Tochter im letzten Monat angezeigt. Diese Technologien haben sich schon deutlich weiterentwickelt.

Aber diese Tools sind noch immer nur Beiwerk des täglichen Lebens. Der Übergang dahin, dass die KI die Art und Weise bestimmt, wie wir arbeiten, denken und leben, hat noch nicht stattgefunden.

Heutzutage ist es schwer vorstellbar, dass die KI große Auswirkungen auf unsere Welt haben wird, weil sie bislang noch kaum im Einsatz ist. Auch 1995 wäre es schwer zu glauben gewesen, welchen Einfluss das Internet in der Zukunft haben würde. Damals konnten sich die meisten Menschen nicht vorstellen, welche Bedeutung das Internet für sie haben könnte und wie es ihr Leben verändern würde. Gleiches gilt heute für Deep Learning und KI. Aber machen wir uns nichts vor: Die KI wird sich unaufhaltsam durchsetzen. In nicht allzu ferner Zukunft wird eine KI Ihr Assistent sein, vielleicht sogar Ihr Freund. Sie wird Ihre Fragen beantworten, bei der Erziehung der Kinder zur Hand gehen und auf Ihre Gesundheit achten. Sie wird Ihnen Lebensmittel bis vor die Haustür bringen und Sie von A nach B befördern. Sie wird Ihre Schnittstelle zu einer immer komplexeren und informationsintensiveren Welt sein. Noch wichtiger ist, dass die KI der gesamten Menschheit Fortschritte ermöglichen wird, indem sie menschlichen Wissenschaftlern bei bahnbrechenden Entdeckungen auf allen Forschungsgebieten assistiert, von der Genetik bis hin zur Mathematik.

Bis es so weit ist, wird es womöglich einige Rückschläge oder vielleicht einen neuen KI-Winter geben, auf ähnliche Weise wie das Internet in den Jahren 1998 und 1999 übertrieben hochgejubelt wurde und schließlich unter einem Zusammenbruch zu leiden hatte, der zu einem Rückgang der Investitionen führte, der bis Anfang der 2000er-Jahre anhielt. Aber irgendwann wird es so weit sein. Die KI wird auf nahezu alle Vorgänge angewendet werden, die unsere Gesellschaft und unseren Alltag ausmachen, ganz so wie heutzutage das Internet.

Schenken Sie dem kurzfristigen Hype keinen Glauben, aber vertrauen Sie auf die langfristige Vision. Es wird wohl noch eine Weile dauern, bis das volle Potenzial der KI ausgeschöpft werden kann – ein Potenzial, von dem noch niemand auch nur zu träumen gewagt hat. Aber die KI wird sich unaufhaltsam durchsetzen und unsere Welt auf fantastische Weise verändern.

1.2 Vor Deep Learning: eine kurze Geschichte des Machine Learnings

Die öffentliche Aufmerksamkeit für Deep Learning und die von der Industrie getätigten Investitionen haben ein Ausmaß angenommen, das in der Geschichte der KI beispielslos ist, dennoch handelt es sich nicht um die erste erfolgreiche Form des Machine Learnings. Man kann mit Sicherheit sagen, dass die meisten heutzutage in der Industrie eingesetzten Machine-Learning-Algorithmen keine Deep-Learning-Algorithmen sind. Deep Learning ist keineswegs immer das geeignete Tool für eine gegebene Aufgabe – manchmal sind nicht genügend Daten für die Anwendung von Deep Learning vorhanden, in anderen Fällen kann ein anderer Algorithmus die Aufgabe besser lösen. Falls Sie beim Deep Learning erstmals mit Machine Learning in Berührung kommen, laufen Sie womöglich Gefahr, dass Sie ausschließlich auf den »Deep-Learning-Hammer« zurückgreifen und dass plötzlich alle Machine-Learning-Aufgaben wie Nägel aussehen. Die Kenntnis anderer geeigneter Ansätze und Verfahren ist die einzige Möglichkeit, nicht in diese Falle zu tappen.

Eine ausführliche Erörterung der klassischen Ansätze des Machine Learnings geht über den Rahmen dieses Buchs hinaus, wir werden sie dennoch kurz betrachten und die Umstände beschreiben, unter denen sie entwickelt wurden. Auf diese Weise können wir Deep Learning in den Kontext des Machine Learnings einordnen und besser verstehen, wie Deep Learning entstand und warum das von Bedeutung ist.

1.2.1 Probabilistische Modellierung

Probabilistische Modellierung ist die Anwendung statistischer Prinzipien auf die Datenanalyse. Dabei handelt es sich um eine der ersten Formen des Machine Learnings, die auch heute noch sehr gebräuchlich ist. Zu den bekanntesten Algorithmen dieser Kategorie gehört der *naive Bayes-Klassifikator*.

Der naive Bayes-Klassifikator ist ein Machine-Learning-Algorithmus, der auf der Anwendung des Satzes von Bayes beruht, bei der vorausgesetzt wird, dass die Merkmale der Eingabedaten alle voneinander unabhängig sind (eine »naive« Annahme, die für die Bezeichnung namensgebend war). Diese Art der Datenanalyse ist sehr viel älter als der Computer und wurde schon Jahrzehnte vor der ersten Computerimplementierung (vermutlich in den 1950er-Jahren) von Hand angewendet. Der Satz von Bayes und die Grundlagen der Statistik entstammen dem 18. Jahrhundert, und mehr müssen Sie für die Verwendung des naiven Bayes-Klassifikators gar nicht wissen.

Die *logistische Regression* ist ein eng verwandtes Modell, das manchmal als das »Hallo Welt«-Pendant des modernen Machine Learnings bezeichnet wird. Lassen

Sie sich nicht von der Bezeichnung täuschen: Die logistische Regression ist kein Regressionsalgorithmus, sondern ein Klassifizierungsalgorithmus. Ebenso wie der naive Bayes-Klassifikator ist die logistische Regression sehr viel älter als der Computer, sie ist aber dessen ungeachtet dank ihrer Einfachheit und Vielseitigkeit auch heute noch nützlich. Sie ist oft das Erste, was ein Data Scientist auf eine Datenmenge anwendet, um ein Gespür für die gegebene Klassifizierungsaufgabe zu erlangen.

1.2.2 Die ersten neuronalen Netze

Die ersten Formen neuronaler Netze sind vollständig von den in diesem Buch beschriebenen modernen Varianten ersetzt worden. Es ist jedoch aufschlussreich, zu wissen, welche Rolle sie bei der Entstehung des Deep Learnings spielten.

Die grundlegenden Konzepte neuronaler Netze wurden zwar schon in den 1950er-Jahren in Form von einfachen Modellen untersucht, es sollte jedoch noch Jahrzehnte dauern, bis dieser Ansatz richtig in Schwung kam. Es fehlte lange die Möglichkeit, große neuronale Netze effizient zu trainieren. Mitte der 1980er-Jahre änderte sich das, als mehrere Forscher voneinander unabhängig den Backpropagation-Algorithmus wiederentdeckten – eine Möglichkeit, miteinander verknüpfte parametrische Operationen mithilfe eines auf dem Gradientenabstiegsverfahren beruhenden Optimierungsansatzes zu trainieren – und ihn auf neuronale Netze anwendeten. (Wir werden diese Konzepte später im Buch noch genau definieren.)

Die erste praktische Anwendung eines *Convolutional Neural Networks* (konvolutionales neuronales Netz, kurz CNN) wurde 1989 von den Bell Labs vorgestellt. Yann LeCun kombinierte die älteren Konzepte CNN und Backpropagation und wendete sie auf die Klassifizierung handgeschriebener Ziffern an. Das so entstandene Netz, das den Spitznamen *LeNet* erhielt, wurde in den 1990er-Jahren vom United States Postal Service zur Automatisierung des Lesens von Postleitzahlen auf Briefumschlägen eingesetzt.

1.2.3 Kernel-Methoden

Nachdem neuronale Netze unter den Forschern dank dieses ersten Erfolgs in den 1990er-Jahren allmählich Anerkennung fanden, erlangte ein neuer Machine-Learning-Ansatz Berühmtheit und sorgte dafür, dass neuronale Netze schnell wieder in Vergessenheit gerieten: *Kernel-Methoden*. Kernel-Methoden bilden eine Gruppe von Klassifizierungsalgorithmen, von denen die *Support Vector Machine* (SVM) am bekanntesten ist. Die moderne Beschreibung einer SVM wurde Anfang der 1990er-Jahre in den Bell Labs von Vladimir Vapnik und Corinna Cortes entwickelt und im Jahr 1995 veröffentlicht.[2] Allerdings gibt es auch eine ältere, lineare Be-

2 Vladimir Vapnik und Corinna Cortes, *Support-Vector Networks*, Machine Learning 20, No. 3 (1995): Seiten 273–297.

schreibung, die bereits 1963 von Vapnik und Alexey Chervonenkis veröffentlicht wurde.[3]

SVMs versuchen Klassifizierungsaufgaben zu lösen, indem sie eine geeignete *Entscheidungsgrenze* (siehe Abbildung 1.10) zwischen zwei zu verschiedenen Kategorien gehörenden Punktmengen aufspüren. Eine Entscheidungsgrenze kann man sich wie eine Linie oder Fläche vorstellen, die Ihre Trainingsdaten in zwei Gebiete unterteilt, die jeweils einer der Kategorien zugeordnet sind. Zur Klassifizierung neuer Datenpunkte müssen Sie lediglich prüfen, auf welcher Seite der Entscheidungsgrenze sie sich befinden.

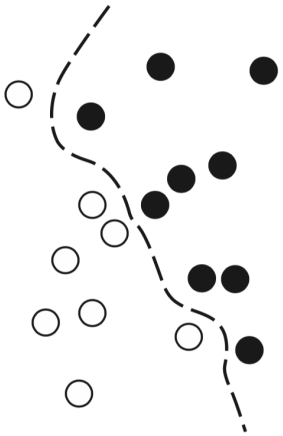

Abb. 1.10: Eine Entscheidungsgrenze

Bei der Suche nach der Entscheidungsgrenze führen SVMs zwei Schritte durch:

1. Die Daten werden auf eine hochdimensionale Repräsentation abgebildet, in der die Entscheidungsgrenze als *Hyperebene* formuliert werden kann. (Wenn die Daten, wie in Abbildung 1.10, zweidimensional sind, ist die Hyperebene eine einfache Linie.)
2. Eine geeignete Entscheidungsgrenze (eine trennende Hyperebene) wird berechnet, indem man versucht, den Abstand zwischen Hyperebene und den am nächsten gelegenen Punkten der beiden Klassen zu maximieren. Dieser Schritt wird als *Maximierung des Randbereichs* bezeichnet. Auf diese Weise lässt sich die Entscheidungsgrenze gut auf neue Punkte verallgemeinern, die nicht Teil der Trainingsdatenmenge sind.

3 Vladimir Vapnik und Alexey Chervonenkis, *A Note on One Class of Perceptrons*, Automation and Remote Control 25 (1964).

Das Verfahren, Daten auf eine hochdimensionale Repräsentation abzubilden, in der die Klassifizierungsaufgabe einfacher wird, mag theoretisch gut klingen, in der Praxis sind die erforderlichen Berechnungen jedoch oft kaum beherrschbar. An dieser Stelle kommt der *Kernel-Trick* ins Spiel (das entscheidende Konzept, nach dem die Kernel-Methoden benannt sind). Er funktioniert im Wesentlichen folgendermaßen: Um im neuen Repräsentationsraum eine geeignete Entscheidungsgrenze zu finden, muss man nicht unbedingt die Koordinaten sämtlicher Punkte in diesem neuen Raum berechnen. Es genügt, den Abstand von Punktepaaren in diesem Raum zu ermitteln, was sich effizient mit einer *Kernel-Funktion* erledigen lässt. Eine Kernel-Funktion ist eine Operation mit überschaubarem Rechenaufwand, die allen Punktepaaren im ursprünglichen Raum den Abstand dieser Punktepaare im neuen Repräsentationsraum zuordnet und dabei die explizite Berechnung der neuen Repräsentation vollständig umgeht. Kernel-Funktionen werden typischerweise von Hand erstellt und nicht anhand der Daten erlernt. Im Fall einer SVM wird lediglich die trennende Hyperebene erlernt.

Zum Zeitpunkt ihrer Entwicklung boten SVMs bei einfachen Klassifizierungsaufgaben eine Leistung, die dem Stand der Technik entsprach, und gehörten zu den wenigen Methoden des Machine Learnings, die auf einer umfassenden Theorie beruhten und die einer ernsthaften mathematischen Analyse zugänglich waren. Dadurch waren sie gut zu verstehen und leicht interpretierbar. Aufgrund dieser angenehmen Eigenschaften wurden SVMs im Fachgebiet Machine Learning sehr beliebt und blieben es auch lange.

Aber es stellte sich heraus, dass SVMs bei großen Datenbanken schlecht skalieren und bei Aufgaben der Sinneswahrnehmung, wie z. B. der Bildklassifizierung, keine besonders guten Ergebnisse lieferten. Da eine SVM dem Shallow Learning zuzuordnen ist, müssen zunächst manuell sinnvolle Repräsentationen der Daten erzeugt werden, bevor man eine SVM für Aufgaben der Sinneswahrnehmung verwenden kann. Dieser Schritt wird als *Merkmalserstellung* (engl. *Feature Engineering*) bezeichnet und ist oft schwierig und fehleranfällig.

1.2.4 Entscheidungsbäume, Random Forests und Gradient Boosting Machines

Entscheidungsbäume sind Strukturen, die Flussdiagrammen ähneln und es ermöglichen, Datenpunkte zu klassifizieren oder Ausgabewerte für gegebene Eingabewerte vorherzusagen (siehe Abbildung 1.11). Sie lassen sich leicht visualisieren und interpretieren. In den 2000er-Jahren weckten anhand von Daten erlernte Entscheidungsbäume das Interesse der Forschung. Seit 2010 werden sie oftmals anstelle von Kernel-Methoden eingesetzt.

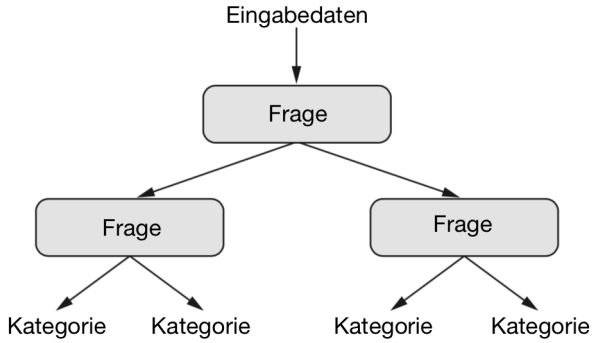

Abb. 1.11: Ein Entscheidungsbaum: Die erlernten Parameter sind die Fragen zu den Daten. Eine solche Frage könnte beispielsweise »Ist Koeffizient 2 in den Daten größer als 3,5?« lauten.

Insbesondere der *Random-Forest*-Algorithmus bot einen robusten und praxistauglichen Ansatz für das Trainieren von Entscheidungsbäumen, bei dem eine große Anzahl spezialisierter Entscheidungsbäume zum Einsatz kommt, deren Ausgaben zusammengefasst werden. Random Forests sind für ein breites Aufgabenspektrum geeignet – man könnte sagen, dass sie fast immer der zweitbeste Algorithmus für Aufgaben des Shallow Learnings sind. Als 2010 die beliebte Website für Machine-Learning-Wettbewerbe Kaggle (`http://kaggle.io`) ans Netz ging, wurden Random Forests schnell zum Favoriten dieser Plattform – bis 2014, als *Gradient Boosting Machines* diese Stelle einnahmen. Eine Gradient Boosting Machine ist, ähnlich wie ein Random Forest, ein Machine-Learning-Verfahren, das auf der Zusammenfassung vieler schwacher Vorhersagemodelle (üblicherweise Entscheidungsbäume) beruht. Es basiert auf Gradient Boosting, einer Methode zur Verbesserung beliebiger Machine-Learning-Modelle durch verschachteltes Trainieren neuer Modelle, die darauf spezialisiert sind, die Schwachpunkte der vorangegangenen Modelle zu korrigieren. Die Anwendung des Gradient-Boosting-Verfahrens auf Entscheidungsbäume führt zu Modellen, die Random Forests meist weit überlegen sind, aber ansonsten ähnliche Eigenschaften besitzen. Hierbei handelt es sich derzeit um einen der besten, wenn nicht sogar um *den* besten Algorithmus zur Handhabung von Daten jenseits der Sinneswahrnehmung. Dieses Verfahren wird, neben Deep Learning, bei Kaggle-Wettbewerben am häufigsten eingesetzt.

1.2.5 Zurück zu neuronalen Netzen

Obwohl weite Teile des Wissenschaftsbetriebs um das Jahr 2010 herum neuronale Netze nahezu vollständig mieden, arbeiteten ein paar Forscher weiter daran, und es gelangen einige wichtige Durchbrüche. Daran beteiligt waren die Forschungsgruppen um Geoffrey Hinton an der University of Toronto, Yoshua Bengio an der University of Montreal, Yann LeCun an der New York University und das IDSIA

(ital. *Istituto Dalle Molle di Studi sull'Intelligenza Artificiale*, Dalle-Molle-Forschungsinstitut für Künstliche Intelligenz) in der Schweiz.

Dan Ciresan vom IDSIA konnte 2011 mit GPU-trainierten DNNs mehrere akademische Bildklassifizierungswettwerbe gewinnen – die ersten in der Praxis erzielten Erfolge des modernen Deep Learnings. Der entscheidende Schritt vorwärts gelang jedoch 2012, als Hintons Forschungsgruppe sich am jährlichen Bildklassifizierungswettbewerb der ImageNet-Datenbank beteiligte. (ImageNet ist eine aus mehr als 1,4 Millionen großformatigen Bildern bestehende Bilddatenbank für Forschungszwecke.) Der ImageNet-Wettbewerb war damals bekanntermaßen äußerst schwierig, denn die zu klassifizierenden hochaufgelösten Farbfotos mussten nach dem Training mit 1,4 Millionen Bildern 1.000 verschiedenen Kategorien zugeordnet werden. Das Gewinnermodell im Jahr 2011, das auf klassischen Ansätzen des maschinellen Sehens beruhte, erzielte eine Korrektklassifizierungsrate (engl. Accuracy, auch Genauigkeit) von lediglich 74,3%. 2012 erreichte ein von Alex Krizhevsky geleitetes Team, dem Geoffrey Hinton beratend zur Seite stand, eine Korrektklassifizierungsrate von 86,3% – ein bedeutender Durchbruch. Der Wettbewerb wird seither von *Deep Convolutional Neural Networks* (tiefe konvolutionale neuronale Netze, kurz DCNNs) dominiert. Im Jahr 2015 erzielte der Gewinner eine Korrektklassifizierungsrate von 96,4%, und die Klassifizierung der ImageNet-Datenbank wurde als vollständig gelöste Aufgabe eingestuft.

Seit 2012 sind DCNNs für Aufgaben des maschinellen Sehens zum Algorithmus der Wahl geworden. Allgemeiner gesagt, sind solche Algorithmen grundsätzlich für alle Aufgaben der Sinneswahrnehmung geeignet. Bei den größeren Konferenzen zum Thema maschinelles Sehen in den Jahren 2015 und 2016 war es fast unmöglich, irgendeinen Vortrag zu finden, bei dem es nicht in irgendeiner Form um CNNs ging. Gleichzeitig erschloss sich Deep Learning diverse weitere Aufgabenstellungen, wie z. B. die Verarbeitung natürlicher Sprache. In vielen Anwendungsbereichen hat Deep Learning SVMs und Entscheidungsbäume vollständig verdrängt. Das CERN, die europäische Organisation für Kernforschung, verwendete beispielsweise einige Jahre lang auf Entscheidungsbäumen beruhende Verfahren zur Analyse der Partikeldaten des ATLAS-Detektors am LHC (*Large Hadron Collider*). Inzwischen werden dort Keras-basierte DNNs eingesetzt, weil diese eine bessere Leistung erzielen und sich leichter mit großen Datenmengen trainieren lassen.

1.2.6 Das Besondere am Deep Learning

Der Hauptgrund dafür, dass sich Deep Learning so schnell durchgesetzt hat, ist die verbesserte Leistung, die es bei vielen Aufgabenstellungen bietet. Das ist jedoch nicht der einzige Grund. Deep Learning vereinfacht das Lösen von Aufgaben so sehr, weil es einen der entscheidenden Schritte des Machine-Learning-Workflows automatisiert: die Merkmalserstellung.

Bei früheren Machine-Learning-Verfahren, wie dem Shallow Learning, mussten die Eingabedaten in ein oder zwei aufeinanderfolgende Repräsentationsräume transformiert werden, für gewöhnlich durch einfache Transformationen wie hochdimensionale nicht lineare Projektionen (SVMs) oder Entscheidungsbäume. Aber die bei komplexeren Aufgabenstellungen erforderlichen ausgefeilteren Repräsentationen lassen sich durch diese Verfahren im Allgemeinen nicht erzielen. Ein menschlicher Benutzer musste also große Anstrengungen unternehmen, um die ursprünglichen Eingabedaten so aufzubereiten, dass sie mit diesen Verfahren verarbeitet werden konnten. Die für die Repräsentation der Daten geeigneten Layer mussten manuell konstruiert werden. Man spricht hier von der *Merkmalserstellung* (engl. *Feature Engineering*). Beim Deep Learning ist dieser Schritt hingegen komplett automatisiert. Die Merkmale werden in einem einzigen Durchgang vollständig erlernt und müssen nicht manuell erstellt werden. Diese Vorgehensweise hat den Machine-Learning-Workflow sehr vereinfacht, und in vielen Fällen wurden umständliche mehrstufige Pipelines durch ein einzelnes, einfaches und alles umfassendes Deep-Learning-Modell ersetzt.

Aber wenn das Vorhandensein mehrerer aufeinanderfolgender Layer von so entscheidender Bedeutung ist, könnte man dann nicht einfach wiederholt Shallow-Learning-Methoden anwenden, um die Ergebnisse des Deep Learnings zu simulieren? In der Praxis zeigt sich, dass die Ergebnisse mehrfach angewendeter Shallow-Learning-Methoden sehr schnell deutlich schlechter ausfallen, *weil der optimale erste Repräsentations-Layer eines dreischichtigen Modells als erster Layer für ein ein- oder zweischichtiges Modell nicht optimal ist*. Beim Deep Learning ist entscheidend, dass es dem Modell ermöglicht wird, alle Repräsentations-Layer zusammen und gleichzeitig zu erlernen (man bezeichnet das als *greedy*, also gierig) anstatt der Reihe nach. Durch dieses gleichzeitige Erlernen der Merkmale werden bei der Anpassung des Modells an eins der internen Merkmale alle anderen davon abhängigen Merkmale ebenfalls automatisch angepasst, ohne dass ein menschliches Eingreifen nötig wäre. Alles wird durch ein einziges Feedback-Signal gesteuert: Alle Änderungen am Modell tragen zum eigentlichen Ziel bei. Dieses Verfahren ist sehr viel leistungsfähiger als ein massenhaftes Aufreihen von Shallow-Learning-Modellen, weil es ermöglicht, komplexe, abstrakte Repräsentationen zu erlernen, indem sie in eine lange Reihe dazwischenliegender Räume (Layer) unterteilt werden: Jeder dieser Räume unterscheidet sich von dem vorhergehenden nur durch eine einfache Transformation.

Das Lernen aus den Daten ist beim Deep Learning durch zwei wesentliche Eigenschaften gekennzeichnet: die *inkrementelle, schichtweise Vorgehensweise, bei der zunehmend komplexere Repräsentationen entwickelt werden*, und die Tatsache, dass die *dazwischenliegenden inkrementellen Repräsentationen zusammen erlernt werden*, wobei jeder Layer so aktualisiert wird, dass er den Ansprüchen der Repräsentationen der vorhergehenden und der nachfolgenden Layer genügt. Diese beiden

Eigenschaften sind gemeinsam dafür verantwortlich, dass Deep Learning so viel erfolgreicher als frühere Machine-Learning-Ansätze geworden ist.

1.2.7 Der Stand des modernen Machine-Learnings

Die Machine-Learning-Wettbewerbe bei Kaggle näher zu betrachten, stellt eine hervorragende Möglichkeit dar, sich ein Bild vom aktuellen Stand der Machine-Learning-Algorithmen und der verfügbaren Tools zu machen. Dank der starken Konkurrenz (bei manchen Wettbewerben gibt es mehrere Tausend Teilnehmer und Geldpreise in Millionenhöhe) und der großen Vielfalt der Machine-Learning-Aufgaben bietet Kaggle eine realitätsnahe Möglichkeit, zu beurteilen, was tatsächlich funktioniert und was nicht. Hier finden Sie die Antwort auf die Frage, welche Art Algorithmus zuverlässig Wettbewerbe gewinnt und welche Tools die besten Teilnehmer einsetzen.

In den Jahren 2016 und 2017 standen bei Kaggle zwei Ansätze im Vordergrund: Gradient Boosting Machines und Deep Learning. Gradient Boosting kommt bei Aufgaben zum Einsatz, bei denen strukturierte Daten zur Verfügung stehen. Deep Learning hingegen wird für Aufgaben der Sinneswahrnehmung genutzt, wie z. B. der Bildklassifizierung. Anwender des erstgenannten Verfahrens verwenden fast immer die ausgezeichnete XGBoost-Bibliothek, die Unterstützung für die beiden in der Data Science beliebtesten Programmiersprachen bietet: Python und R. Die meisten Kaggle-Teilnehmer, die Deep Learning einsetzen, verwenden inzwischen die Keras-Bibliothek, denn sie ist leicht nutzbar, flexibel und unterstützt Python.

Mit diesen beiden Verfahren sollten Sie vertraut sein, wenn Sie heutzutage Machine Learning erfolgreich anwenden möchten: Gradient Boosting Machines für Shallow-Learning-Aufgaben und Deep Learning für Aufgaben der Sinneswahrnehmung. Rein technisch betrachtet, bedeutet das, dass Ihnen XGBoost und Keras vertraut sein sollten – die beiden Bibliotheken, die derzeit die Kaggle-Wettbewerbe dominieren. Mit diesem Buch sind Sie Ihrem Ziel schon einen großen Schritt näher gekommen.

1.3 Warum Deep Learning? Und warum jetzt?

Die beiden grundlegenden Konzepte des Deep Learnings für maschinelles Sehen – CNNs und Backpropagation – waren bereits 1989 etabliert. Der LSTM-Algorithmus (*Long Short-Term Memory*, zu Deutsch etwa »langes Kurzzeitgedächtnis«), der beim Deep Learning für Zeitreihen von fundamentaler Bedeutung ist, wurde 1997 entwickelt und hat sich seither kaum verändert. Warum aber kam Deep Learning dann erst nach 2012 so richtig in Schwung? Was hat sich in den anderthalb Jahrzehnten geändert?

Es gibt drei allgemeine Faktoren, die den Fortschritt beim Machine Learning beeinflussen:

- Hardware
- Datenmengen und Benchmarks
- Verbesserte Algorithmen

Dieses Fachgebiet wird nicht durch theoretische Überlegungen, sondern durch experimentelle Entdeckungen vorangetrieben, daher sind verbesserte Algorithmen nur dann machbar, wenn die entsprechenden Daten und die nötige Hardware zur Verfügung stehen, um neue Ideen auszuprobieren (oder, wie es häufig der Fall ist, ältere Ideen in größerem Maßstab umzusetzen). Anders als in der Mathematik oder der Physik ist es beim Machine Learning nicht möglich, nur mit Papier und Bleistift entscheidende Fortschritte zu erzielen. Machine Learning ist eine Ingenieurwissenschaft.

In den 1990er- und 2000er-Jahren stellten die Daten und die Hardware den eigentlichen Engpass dar. Aber in diesem Zeitraum erlebte das Internet seinen Durchbruch, und für den Bedarf des Spielemarkts wurden Hochleistungsgrafikchips entwickelt.

1.3.1 Hardware

Zwischen 1990 und 2010 sind handelsübliche CPUs etwa um den Faktor 5.000 schneller geworden. Dementsprechend ist es heutzutage möglich, kleine Deep-Learning-Modelle auf einem Laptop auszuführen. Vor 25 Jahren wäre das noch undenkbar gewesen.

Allerdings benötigen die beim maschinellen Sehen oder bei der Spracherkennung typischerweise eingesetzten Deep-Learning-Modelle Rechnenleistung in einer Größenordnungen, die ein Laptop nicht zu liefern imstande ist. In den 2000er-Jahren investierten Unternehmen wie NVIDIA und AMD Milliarden an Dollars in die Entwicklung schneller, parallel arbeitender Chips (*Graphical Processing Units*, Grafikprozessoren oder kurz GPUs), die zur Darstellung immer fotorealistischerer Videospiele dienten – preiswerte Einzweck-Supercomputer, die dafür ausgelegt sind, komplexe 3-D-Szenen in Echtzeit auf dem Bildschirm anzuzeigen. Diese Investitionen kamen schließlich auch der Wissenschaftsgemeinde zugute, als NVIDIA 2007 CUDA (`https://developer.nvidia.com/about-cuda`) vorstellte, eine Programmierschnittstelle für ihre GPU-Baureihe. Bei verschiedenen hochgradig parallelisierbaren Anwendungen, wie etwa physikalischen Simulationen, konnten einige wenige GPUs riesige CPU-Cluster ersetzen. DNNs, für die vornehmlich viele kleine Matrizenmultiplikationen erforderlich sind, lassen sich ebenfalls in hohem Maße parallelisieren. Und seit etwa 2011 programmieren

einige Forscher CUDA-Implementierungen ihrer neuronalen Netze. Zu den ersten gehörten Dan Ciresan[4] und Alex Krizhevsky.[5]

Faktisch subventionierte der Spielemarkt das Supercomputing der nächsten Generation von KI-Anwendungen. Manchmal wird aus einem Spiel tatsächlich eine bedeutende Entwicklung. Eine NVIDIA TITAN X, eine Grafikkarte für Spiele, die Ende 2015 1.000 Dollar kostete, kann bis zu 6,6 TFLOPS liefern (einfache Genauigkeit): 6,6 Billionen `float32`-Operationen pro Sekunde. Das ist etwa das 350-Fache dessen, was ein moderner Laptop leistet. Mit einer TITAN X dauert es nur ein paar Tage, ein ImageNet-Modell zu trainieren, mit dem Sie vor einigen Jahren noch den ILSVRC-Wettbewerb (*ImageNet Large Scale Visual Recognition Challenge*) gewonnen hätten. Inzwischen trainieren große Unternehmen Deep-Learning-Modelle mit Clustern aus Hunderten von GPUs, die speziell auf die Anforderungen des Deep Learnings zugeschnitten sind, wie z. B. die NVIDIA Tesla K80. Die Rechenleistung, die solche Cluster bieten, wäre ohne moderne GPUs nicht möglich.

Darüber hinaus ging die Deep-Learning-Industrie noch einen Schritt weiter und investierte in zunehmend spezialisiertere effiziente Chips für Deep Learning. Google hat auf seiner alljährlich stattfindenden Entwicklerkonferenz Google I/O 2016 das Projekt TPU (*Tensor Processing Unit*) vorgestellt, ein neues Chipdesign, das von Grund auf für die Anforderungen von DNNs ausgelegt ist und Berichten zufolge zehnmal schneller und sehr viel energiesparender ist als die leistungsfähigsten GPUs.

1.3.2 Daten

Mitunter heißt es, dass die KI eine neue industrielle Revolution einläutet. Wenn Deep Learning bei dieser Revolution die Rolle der Dampfmaschine einnimmt, dann sind die Daten die Kohle, also das Rohmaterial, das unsere intelligenten Maschinen antreibt und ohne das es nicht geht. Was die Daten betrifft, hat nicht nur das exponentielle Wachstum der Speicherkapazität in den letzten 20 Jahren (nach dem Moore'schen Gesetz), sondern auch der Aufstieg des Internets für veränderte Bedingungen gesorgt, denn nun ist es möglich, für das Machine Learning sehr große Datenmengen zu sammeln oder zu verteilen. Heutzutage verwenden große Unternehmen Datenmengen (Bilder, Videos, Sprachaufnahmen), die ohne das Internet nicht zustande gekommen wären. Die Verschlagwortung der bei Flickr gespeicherten Bilder durch die Benutzer hat sich für das maschinelle Sehen als wahre Datengoldgrube erwiesen. Gleiches gilt auch für YouTube-Videos. Und

4 Siehe *Flexible, High Performance Convolutional Neural Networks for Image Classification*, Proceedings of the 22nd International Joint Conference on Artificial Intelligence (2011), `http://www.ijcai.org/Proceedings/11/Papers/210.pdf`.

5 Siehe *ImageNet Classification with Deep Convolutional Neural Networks*, Advances in Neural Information Processing Systems 25 (2012), `http://mng.bz/2286`.

die Wikipedia ist für die Verarbeitung natürlicher Sprache eine Datenmenge von entscheidender Bedeutung.

Wenn man eine Datenmenge nennen sollte, die als Katalysator für den Erfolg des Deep Learnings gedient hat, dann die ImageNet-Datenbank, die mehr als 1,4 Millionen Bilder enthält, die von Hand einer von 1.000 Kategorien zugewiesen worden sind (eine Kategorie pro Bild). Das Besondere an der ImageNet-Datenbank ist jedoch nicht nur ihre Größe, sondern auch der dazugehörige alljährlich stattfindende Wettbewerb.[6]

Wie Kaggle seit 2010 gezeigt hat, sind öffentliche Wettbewerbe eine ausgezeichnete Möglichkeit, Wissenschaftler und Entwickler zu motivieren, ihre Forschungen voranzutreiben. Die allgemein bekannten Benchmarks, die alle Forscher gern übertreffen möchten, haben viel zu den jüngsten Erfolgen des Deep Learnings beigetragen.

1.3.3 Algorithmen

Neben der Hardware und den Daten fehlte bis Ende der 2000er-Jahre auch eine zuverlässige Möglichkeit, sehr große DNNs zu trainieren. Deshalb waren sie noch immer nicht sehr tief und bestanden aus nur einem oder zwei Repräsentations-Layern. Dementsprechend waren sie auch nicht in der Lage, im Vergleich mit ausgefeilteren Shallow-Learning-Methoden wie SVMs und Random Forests zu glänzen. Der entscheidende Punkt war die Ausbreitung des Gradienten (engl. *Gradient Propagation*) in den tieferen Layern. Das zum Trainieren der NNs verwendete Feedback-Signal wurde mit steigender Zahl der Layer immer schwächer.

Das sollte sich 2009/2010 durch einige einfache, aber bedeutende Verbesserungen der Algorithmen ändern, die nun eine bessere Ausbreitung des Gradienten ermöglichten:

- verbesserte *Aktivierungsfunktionen* für die Layer
- verbesserte *Verfahren zur Initialisierung der Gewichtungen*, zunächst durch schichtweises Vortrainieren, was aber bald wieder aufgegeben wurde
- verbesserte *Optimierungsverfahren*, wie etwa RMSProp und Adam

Erst nachdem diese Verbesserungen es ermöglichten, Modelle mit zehn oder mehr Layern zu trainieren, zeigten sich die Vorteile des Deep Learnings.

In den Jahren 2014, 2015 und 2016 wurden weitere sogar noch fortschrittlichere Verfahren zur Verbesserung der Ausbreitung des Gradienten entdeckt, wie z. B. die Normierung der Stapel (engl. Batch Normalization, dt. auch als Batch-Normalisierung bezeichnet), residuale Verbindungen oder kanalweise trennbare Faltun-

6 Der vorhin schon kurz erwähnte ILSVRC-Wettbewerb (*ImageNet Large Scale Visual Recognition Challenge*). Siehe auch `http://www.image-net.org/challenges/LSVRC`.

gen. Heutzutage können wir problemlos Modelle trainieren, die aus mehreren Tausend Layern bestehen.

1.3.4 Eine neue Investitionswelle

Nachdem sich Deep Learning in den Jahren 2012 und 2013 zum neuen Stand der Technik für das maschinelle Sehen und schließlich auch für alle anderen Aufgaben der Sinneswahrnehmung entwickelt hatte, nahmen die Wirtschaftsführer das ebenfalls zur Kenntnis. Daraufhin folgte eine allmählich wachsende Investitionswelle, die alles übertraf, was es in der Geschichte der KI bisher gegeben hatte.

Im Jahr 2011, unmittelbar bevor Deep Learning die Aufmerksamkeit auf sich zog, betrug das gesamte in die KI investierte Risikokapital 19 Millionen Dollar, die fast vollständig für praktische Anwendungen von Shallow-Learning-Methoden verwendet wurden. 2014 war dieser Wert auf atemberaubende 394 Millionen Dollar angewachsen. Während dieser drei Jahre wurden Dutzende Start-ups gegründet, die versuchten, vom Deep-Learning-Hype zu profitieren. Große Unternehmen wie Google, Facebook, Baidu und Microsoft hatten zwischenzeitlich Beträge in ihre internen Forschungsabteilungen investiert, die den Fluss des Risikokapitals höchstwahrscheinlich in den Schatten stellen. Es sind nur wenige konkrete Zahlen verfügbar: 2013 übernahm Google das Deep-Learning-Start-up DeepMind für angeblich 500 Millionen Dollar – die kostspieligste Übernahme eines KI-Unternehmens der Geschichte. 2014 eröffnete Baidu im Silicon Valley ein Deep-Learning-Forschungszentrum und investierte 300 Millionen Dollar in das Projekt. Und das Start-up Nervana Systems, das Deep-Learning-Hardware entwickelt, wurde 2016 für mehr als 400 Millionen Dollar von Intel übernommen.

Machine Learning, insbesondere Deep Learning, ist für die Produktstrategie dieser Tech-Giganten inzwischen von entscheidender Bedeutung. Ende 2015 sagte Googles CEO Sundar Pichai: »Machine Learning ist ein zentrales, umwälzendes Verfahren, das wir beim Überdenken unserer Vorgehensweise zur Grundlage gemacht haben. Wir wenden es wohlüberlegt auf alle unsere Produkte an, ob Internetsuche, Anzeigen, YouTube oder Play. Und wir stehen erst am Anfang, aber Sie werden sehen, dass wir Machine Learning systematisch in all diesen Bereichen einsetzen werden.«[7]

Diese Investitionswelle hatte zur Folge, dass die Zahl der Beschäftigten, die an Deep Learning arbeiten, in nur fünf Jahren von einigen Hundert auf mehrere Zehntausende stieg und der Fortschritt der Forschung ein aberwitziges Tempo erreicht hat. Gegenwärtig gibt es keinerlei Anzeichen dafür, dass sich diese Entwicklung in absehbarer Zeit verlangsamen wird.

7 Sundar Pichai bei der Bekanntgabe von Alphabets Geschäftszahlen am 22. Oktober 2015.

1.3.5 Die Demokratisierung des Deep Learnings

Zu den entscheidenden Faktoren für die Zunahme der an Deep Learning arbeitenden Beschäftigten gehört die Demokratisierung der in diesem Fachgebiet verwendeten Tools. In der Frühphase waren beträchtliche Fachkenntnisse in C++ und CUDA erforderlich, über die kaum jemand verfügte, um Deep Learning betreiben zu können. Heutzutage reichen grundlegende Python-Kenntnisse aus, um Deep-Learning-Forschung zu betreiben. Dafür verantwortlich sind vor allem die Entwicklung von Theano und später TensorFlow – zwei Python-Frameworks zur Bearbeitung von Tensoren, die automatisches Differenzieren unterstützen und die Implementierung neuer Modelle enorm vereinfachen – und die Verbreitung benutzerfreundlicher Bibliotheken wie Keras, die das Deep Learning so einfach wie das Spielen mit Legosteinen machen. Nach der Veröffentlichung Anfang 2015 wurde Keras für viele neue Start-ups, Doktoranden und Forscher, die auf das neue Fachgebiet drängten, schnell zur Deep-Learning-Lösung der Wahl.

1.3.6 Bleibt es so?

Was ist das Besondere an DNNs? Weshalb sind sie offenbar das »richtige« Verfahren, in das Unternehmen investieren? Wieso befassen sich scharenweise Forscher damit? Oder ist Deep Learning nur eine kurzfristige Modeerscheinung, die nicht von Dauer ist? Werden wir in 20 Jahren noch immer DNNs verwenden?

Deep Learning besitzt verschiedene Eigenschaften, die es rechtfertigen, von einer KI-Revolution zu sprechen, und wird Bestand haben. Wir werden in zwei Jahrzehnten vielleicht keine neuronalen Netze mehr verwenden, aber das, was wir dann benutzen, wird unmittelbar vom modernen Deep Learning und dessen Kernkonzepten abgeleitet sein. Diese wichtigen Eigenschaften lassen sich grob in drei Kategorien einteilen:

- *Einfachheit* – Deep Learning macht die Merkmalserstellung überflüssig und ersetzt komplexe, fehleranfällige und umständliche Pipelines durch ein einfaches alles umfassendes trainierbares Modell, das typischerweise in Form von nur fünf oder sechs Tensoroperationen realisiert wird.
- *Skalierbarkeit* – Deep Learning lässt sich sehr gut für die Ausführung auf GPUs oder TPUs parallelisieren und nutzt so das Moore'sche Gesetz zu seinem Vorteil. Weil das Trainieren von Deep-Learning-Modellen in Form einer Stapelverarbeitung erfolgt, können zudem Datenmengen beliebiger Größe verwendet werden. (Der einzige Engpass ist die gleichzeitig verfügbare Rechenleistung, die aber dank des Moore'schen Gesetzes schnell zunimmt.)
- *Vielseitigkeit und Wiederverwendbarkeit* – Im Gegensatz zu vielen älteren Machine-Learning-Ansätzen können Deep-Learning-Modelle mit zusätzlichen Daten trainiert werden, ohne dass man wieder ganz von vorn anfangen müsste. Damit sind diese Modelle bestens für kontinuierliches Online Learning geeig-

net – eine für produktiv eingesetzte sehr große Modelle wichtige Eigenschaft. Darüber hinaus lassen sich bereits trainierte Deep-Learning-Modelle für andere Zwecke erneut einsetzen und sind somit wiederverwendbar. So ist es beispielsweise möglich, ein für die Bildklassifizierung trainiertes Deep-Learning-Modell als Teil einer Videoverarbeitungs-Pipeline einzusetzen. Auf diese Weise ist die geleistete Arbeit auch in zunehmend komplexeren und leistungsfähigeren Modellen von Nutzen. Außerdem ist das Deep Learning dadurch auch auf vergleichsweise kleine Datenmengen anwendbar.

Deep Learning steht erst seit einigen wenigen Jahren im Mittelpunkt des Interesses, und wir haben noch gar nicht vollständig ausgelotet, was es alles zu leisten vermag. Mit jedem Monat, der verstreicht, kommen neue Anwendungsfälle und Verbesserungen hinzu, die früher gültige Beschränkungen aufheben. Nach einer wissenschaftlichen Revolution folgt der weitere Fortschritt im Allgemeinen einer Sigmoidfunktion: Am Anfang steht eine Phase schnellen Fortschritts, die sich allmählich stabilisiert, wenn die Forscher an unüberwindbare Grenzen stoßen und die weiteren Verbesserungen nur noch in kleinen Schritten erfolgen. Das Deep Learning befindet sich Ende 2017 offenbar in der ersten Hälfte des Sigmoids, sodass in den kommenden Jahren mit weiteren bedeutenden Fortschritten zu rechnen ist.

Stichwortverzeichnis