

O'REILLY®

GANs
und VAEs
kreativ nutzen

Generatives Deep Learning

Maschinen das Malen, Schreiben
und Komponieren
beibringen



David Foster
Übersetzung von
Marcus Fraaß und
Konstantin Mack

Papier
plus⁺
PDF.

Zu diesem Buch – sowie zu vielen weiteren O'Reilly-Büchern – können Sie auch das entsprechende E-Book im PDF-Format herunterladen. Werden Sie dazu einfach Mitglied bei oreilly.plus⁺:

www.oreilly.plus

Generatives Deep Learning

*Maschinen das Malen, Schreiben
und Komponieren beibringen*

David Foster

*Deutsche Übersetzung von
Marcus Fraaß und Konstantin Mack*

O'REILLY®

David Foster

Lektorat: Alexandra Follenius

Übersetzung: Marcus Fraaß, Konstantin Mack

Korrektorat: Sibylle Feldmann, www.richtiger-text.de

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

Herstellung: Stefanie Weidner

Umschlaggestaltung: Karen Montgomery, Michael Oréal, www.oreal.de

Druck und Bindung: mediaprint solutions GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek

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

ISBN:

Print 978-3-96009-128-8

PDF 978-3-96010-356-1

ePub 978-3-96010-354-7

mobi 978-3-96010-355-4

1. Auflage

Translation Copyright für die deutschsprachige Ausgabe © 2020 dpunkt.verlag GmbH

Wieblinger Weg 17

69123 Heidelberg

Authorized German translation of the English edition of *Generative Deep Learning*, ISBN 9781492041948 © 2019 Applied Data Science Partners Ltd. This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«. O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Hinweis:

Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.



Schreiben Sie uns:

Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: komentar@oreilly.de.

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag noch Übersetzer können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Vorwort	XI
----------------------	-----------

Teil I Einführung ins Generative Deep Learning

1 Generative Modellierung	3
Was ist generative Modellierung?	3
Vergleich der generativen und diskriminativen Modellierung	4
Fortschritte im maschinellen Lernen	6
Der Vormarsch der generativen Modellierung	7
Das generative Modellierungskonzept	10
Wahrscheinlichkeitsbasierte generative Modelle	12
Hallo Wrodl!	15
Ihr erstes wahrscheinlichkeitsbasiertes generatives Modell	16
Das naive Bayes-Modell	19
Hallo Wrodl! – Fortsetzung	22
Die Herausforderungen der generativen Modellierung	23
Representation Learning	25
Einrichten Ihrer Arbeitsumgebung	28
Zusammenfassung	31
2 Deep Learning	33
Strukturierte und unstrukturierte Daten	33
Tiefe neuronale Netzwerke	35
Keras und TensorFlow	36
Ihr erstes tiefes neuronales Netzwerk	37
Einlesen der Daten	37
Erstellen des Modells	39
Kompilierung des Modells	43

Trainieren des Modells.	45
Bewertung des Modells	46
Verbesserung des Modells	48
Konvolutionsschichten.	49
Batch-Normalisierung	54
Drop-out-Schichten	56
Zusammenführen aller Teilelemente	58
Zusammenfassung	61
3 Variational Autoencoder	63
Die Kunstaustellung	63
Autoencoder	66
Ihr erster Autoencoder	67
Der Encoder	68
Der Decoder	69
Verbinden von Encoder und Decoder	72
Analyse des Autoencoders	73
Die »Variationale« Kunstaustellung	76
Erstellen eines Variational Autoencoders	78
Der Encoder	78
Die Verlustfunktion	84
Analyse des Variational Autoencoders.	85
Erzeugen von Gesichtern mithilfe von VAEs	86
Trainieren des VAE	87
Analyse des VAE	89
Erzeugen neuer Gesichter	90
Arithmetik im latenten Raum	91
Verschmelzen von Gesichtern	93
Zusammenfassung	94
4 Generative Adversarial Networks	95
Ganimals	95
Einführung in GANs.	97
Ihr erstes GAN	98
Der Diskriminator	99
Der Generator	101
Trainieren des GAN.	104
GAN-Herausforderungen	109
Oszillierender Verlust	109
Mode-Collapse.	110
Uninformativer Verlust	111

Hyperparameter	111
Bewältigung der GAN-Herausforderungen	111
Wasserstein-GAN	111
Wasserstein-Verlust	112
Die Lipschitz-Bedingung	114
Gewichte beschränken	115
Training des WGAN	115
Analyse des WGAN	116
WGAN-GP	117
Der Gradient-Penalty-Verlust	118
Analyse des WGAN-GP	121
Zusammenfassung	122

Teil II **Maschinen das Malen, Schreiben, Komponieren und Spielen beibringen**

5 Malen	127
Äpfel und Orangen	128
CycleGAN	130
Ihr erstes CycleGAN	132
Überblick	132
Die Generatoren (U-Net)	133
Die Diskriminatoren	137
Kompilieren des CycleGAN	138
Trainieren des CycleGAN	140
Analyse des CycleGAN	141
Erstellen eines CycleGAN, das wie Monet malt	143
Die Generatoren (ResNet)	144
Analyse des CycleGAN	145
Neuronaler Stiltransfer	147
Inhaltsverlust	148
Stilverlust	150
Gesamt-Varianz-Verlust	153
Ausführen des neuronalen Stiltransfers	154
Analyse des neuronalen Stiltransfer-Modells	155
Zusammenfassung	156
6 Schreiben	157
Die literarische Gesellschaft für lästige Bösewichte e. V.	158
Long-Short-Term-Memory-Netzwerk	159

Ihr erstes LSTM-Netzwerk	160
Tokenisierung	160
Erstellen des Datensatzes	163
Die Architektur des LSTM-Netzwerks	164
Die Embedding-Schicht	164
Die LSTM-Schicht	166
Die Zelle eines LSTM-Netzwerks	167
Einen neuen Text erzeugen	170
Erweiterungen von RNNs	174
Hintereinandergeschaltete rekurrente Netzwerke	174
Gated Recurrent Units	174
Bidirektionale Zellen	177
Encoder-Decoder-Modelle	178
Ein Frage-Antwort-Generator	180
Ein Frage-Antwort-Datensatz	181
Die Modellarchitektur	182
Inferenz	186
Modellergebnisse	187
Zusammenfassung	189
7 Komponieren	191
Vorabklärungen	192
Musiknotation	192
Ihr erstes Musik erzeugendes RNN	195
Der Aufmerksamkeitsmechanismus	196
Erstellen eines Aufmerksamkeitsmechanismus in Keras	198
Analyse des RNN mit Aufmerksamkeitsmechanismus	203
Aufmerksamkeit in Encoder-Decoder-Netzwerken	207
Erzeugen mehrstimmiger Musiktitel	210
Das »Musikorgan«	211
Ihr erstes MuseGAN	212
Der Generator des MuseGAN	215
Akkorde, Stil, Melodie und Groove	216
Der Taktgenerator	218
Zusammenführen aller Elemente	219
Der Kritiker	221
Analyse des MuseGAN	222
Zusammenfassung	224
8 Spielen	227
Reinforcement Learning	228
OpenAI Gym	229

Architektur von World Models	231
Der Variational Autoencoder	232
Das MDN-RNN	232
Der Controller.	233
Setup	234
Überblick über den Trainingsprozess	234
Sammeln zufälliger Roll-out-Daten	235
Trainieren des VAE	238
Die Architektur des VAE.	239
Untersuchung des VAE.	239
Sammeln von Daten zum Trainieren des RNN	243
Trainieren des MDN-RNN	244
Die Architektur des MDN-RNN.	245
Ziehen des nächsten z und der Belohnung aus dem MDN-RNN.	246
Die MDN-RNN-Verlustfunktion	247
Trainieren des Controllers.	249
Die Architektur des Controllers	249
CMA-ES	250
Parallelisierung von CMA-ES	252
Ausgabe des Controller-Trainings	254
In-Traum-Training	255
In-Traum-Training des Controllers	257
Herausforderungen des In-Traum-Trainings	258
Zusammenfassung.	259
9 Die Zukunft der generativen Modellierung	261
Fünf Jahre Fortschritt	261
Der Transformer	263
Positionscodierung	264
Mehrköpfige Aufmerksamkeit	266
Der Decoder	268
Analyse des Transformers.	268
BERT.	270
GPT-2	270
MuseNet.	272
Fortschritte bei der Bilderzeugung.	272
ProGAN	272
GAN mit Selbstaufmerksamkeitsmechanismus (SAGAN)	274
BigGAN.	275
StyleGAN	276

Anwendungen der generativen Modellierung.	279
Durch KI geschaffene Kunst.	280
Durch KI geschaffene Musik.	281
10 Schlussbemerkung	283
Index	287

Vorwort

Das, was ich nicht erschaffen kann, verstehe ich nicht.

– Richard Feynman

Ein essenzieller Teil des menschlichen Wesens ist unsere Fähigkeit, Dinge zu erschaffen. Seit unseren frühesten Tagen als Höhlenbewohner haben wir nach Möglichkeiten gesucht, originelle und schöne Kreationen zu schaffen. Die Urmenschen taten dies in Form von Höhlenmalereien von wilden Tieren und abstrakten Mustern, die sie mit Pigmenten sorgfältig und methodisch auf Felsen anbrachten. Der Zeit der Romantik entstammen die meisterhaften Tschaikowsky-Symphonien, die Triumph- und Tragödiengefühle mithilfe von Klangwellen wecken, die zu schönen Melodien und Harmonien verschmelzen. Heutzutage laufen wir um Mitternacht in die Buchhandlungen, um Geschichten über einen fiktiven Zauberer zu kaufen. Denn die Kombination von Buchstaben bringt eine Geschichte hervor, die uns so sehr fesselt, dass wir unbedingt die nächsten Abenteuer unseres Helden erfahren möchten.

Daher ist es nicht verwunderlich, dass die Menschheit nun beginnt, die ultimative Frage nach der Kreativität zu stellen: Können wir etwas erzeugen, das selbst über Kreativität verfügt?

Genau diese Frage soll die generative Modellierung beantworten. Dank neuester Fortschritte in Methodik und Technologie sind wir heute dazu in der Lage, Maschinen zu bauen, die originelle Kunstwerke in einem vorgegebenen Stil malen, zusammenhängende Absätze mit übergreifender Struktur schreiben, angenehm klingende Musik komponieren und Siegerstrategien für komplexe Spiele entwickeln können, indem sie imaginäre Zukunftsszenarien generieren. Dies ist nur der Beginn einer generativen Revolution, die uns keine andere Wahl lässt, als Antworten auf einige der größten Fragen zur Wirkungsweise von Kreativität zu finden und dadurch letztlich auch zu beleuchten, was es bedeutet, ein Mensch zu sein.

Kurz gesagt, nie war die Zeit besser als jetzt, um mehr über generative Modellierung zu lernen – also lassen Sie uns loslegen!

Ziel und Ansatz

Dieses Buch behandelt die wichtigsten Methoden, die die generative Modellierungslandschaft in den letzten Jahren geprägt haben und durch die wir beeindruckende Fortschritte bei kreativen Aufgaben erzielen konnten. Zusätzlich zur Behandlung der Grundlagen der generativen Modellierung werden wir funktionsfähige Beispiele einiger der Schlüsselmodelle der Fachliteratur erstellen und bei jedem Modell Schritt für Schritt zusammen durch die Codebasis gehen.

Im gesamten Buch finden Sie kurze, sinnbildliche Geschichten, die dabei helfen sollen, die Funktionsweise einiger der Modelle zu erklären. Meiner Meinung nach besteht eine der besten Möglichkeiten, eine neue abstrakte Theorie zu lehren, darin, sie zuerst in etwas nicht ganz so Abstraktes umzuwandeln, etwa in eine Geschichte, bevor man in die technischen Details eintaucht. Dadurch wird die Theorie in diesem Zusammenhang klarer, da sie Menschen, Handlungen und Emotionen einbezieht, die alle gut nachvollziehbar sind, statt neuronaler Netze, Backpropagation und Verlustfunktionen, die allesamt abstrakte Konstrukte darstellen.

Die fiktive Geschichte und die Erklärung der Modelltheorie beschreiben beide die gleiche Funktionsweise, aber auf zwei verschiedene Arten. Sie werden es daher vielleicht nützlich finden, beim Lernen der Modelltheorie auf die jeweilige Geschichte zurückzugreifen. Sollten Sie bereits mit einer bestimmten Methode vertraut sein, werden Sie bestimmt Spaß daran finden, die Parallelen der einzelnen Modellelemente innerhalb der Geschichte zu finden!

Im ersten Teil dieses Buchs werde ich Ihnen die wichtigsten Methoden vorstellen, die wir bei der Erstellung generativer Modelle verwenden, einschließlich eines Überblicks über Deep Learning, Variational Autoencoder und Generative Adversarial Networks. Im zweiten Teil werden wir diese Methoden verwenden, um verschiedene kreative Aufgaben wie Malen, Schreiben und Komponieren von Musik mithilfe von Modellen wie CycleGAN, Encoder-Decoder-Modellen und MuseGAN zu bewältigen. Darüber hinaus werden wir sehen, wie die generative Modellierung zur Optimierung der Spielstrategie für ein Spiel eingesetzt werden kann (World Models). Außerdem werfen wir einen Blick auf topaktuelle generative Architekturen wie StyleGAN, BigGAN, BERT, GPT-2 und MuseNet.

Voraussetzungen

Dieses Buch setzt voraus, dass Sie eine gewisse Erfahrung in der Programmierung mit Python mitbringen. Wenn Sie mit Python nicht vertraut sind, gelingt der Einstieg über LearnPython.org (<https://www.learnpython.org>) am besten. Darüber hinaus gibt es unzählige kostenlose Onlinere Ressourcen, die es Ihnen ermöglichen, sich genügend Python-Wissen anzueignen, um mit den Beispielen in diesem Buch arbeiten zu können.

Da einige der Modelle in mathematischer Notation beschrieben werden, ist es sinnvoll, grundlegende Kenntnisse in linearer Algebra (z.B. Matrixmultiplikation etc.) und der Wahrscheinlichkeitstheorie zu haben.

Schließlich benötigen Sie noch eine Rechenumgebung, in der Sie die Codebeispiele aus dem GitHub-Repository (https://github.com/davidADSP/GDL_code) des Buchs ausführen können. Ich habe bewusst darauf geachtet, dass alle Beispiele in diesem Buch eine nicht zu große Menge an Rechenressourcen für das Trainieren der Modelle benötigen. Es gibt den Mythos, dass Sie eine GPU benötigen, um Deep-Learning-Modelle zu trainieren – dies ist natürlich hilfreich und beschleunigt das Trainieren, aber es ist keineswegs unbedingt notwendig. Wenn Sie noch neu im Deep Learning unterwegs sind, ermutige ich Sie, sich zunächst mit den Grundlagen zu befassen, indem Sie mit kleinen Beispielen auf Ihrem Laptop experimentieren, bevor Sie Geld und Zeit für die Recherche von Hardware ausgeben, um das Trainieren der Modelle zu beschleunigen.

Weitere Ressourcen

Die folgenden zwei Bücher kann ich Ihnen wärmstens empfehlen als allgemeine Einführung in Machine Learning und Deep Learning:

- *Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme* von Aurelien Geron, Übersetzung von Kristian Rother und Thomas Demmig (O'Reilly)
- *Deep Learning mit Python und Keras* von Francois Chollet (mitp)

Die meisten der Forschungsartikel in diesem Buch kommen von arXiv (<https://arxiv.org>), einem kostenlosen Archiv für wissenschaftliche Forschungsarbeiten. Es ist heutzutage üblich, dass Autoren ihre Forschungsarbeiten in arXiv posten, bevor sie vollständig begutachtet werden. Indem Sie dort die jüngsten Veröffentlichungen verfolgen, können Sie sich hervorragend über die neuesten Entwicklungen in diesem Bereich informieren.

Darüber hinaus kann ich Ihnen auch die Webseite Papers with Code (<https://paperswithcode.com>) empfehlen, auf der Sie die neuesten Ergebnisse in einer Vielzahl von Aufgaben des maschinellen Lernens finden können, sowie Links zu den Forschungspapieren und offiziellen GitHub-Repositories. Dies ist eine ausgezeichnete Ressource, um herauszufinden, welche Methoden derzeit die besten Ergebnisse in einer Reihe von Aufgaben erzielen, und sie hat mir bei der Entscheidung geholfen, welche Methoden in diesem Buch behandelt werden sollen.

Zu guter Letzt ist Google Colaboratory (<https://colab.research.google.com>) eine nützliche Ressource für das Trainieren von Deep-Learning-Modellen auf schnellerer Hardware. Dies ist eine kostenlose Jupyter-Notebook-Umgebung, die keine Einrichtung erfordert und vollständig in der Cloud läuft. Sie können dem Notebook sagen, dass es auf einer kostenlos zur Verfügung gestellten GPU laufen soll, und

das für bis zu zwölf Stunden Laufzeit. Es ist zwar nicht unbedingt notwendig, die Beispiele in diesem Buch auf einem Grafikprozessor auszuführen, es kann aber helfen, den Trainingsprozess zu beschleunigen. So oder so, Colab bietet Ihnen die großartige Möglichkeit, kostenlos auf GPU-Ressourcen zuzugreifen.

In diesem Buch verwendete Konventionen

Die folgenden typografischen Konventionen werden in diesem Buch verwendet:

Kursiv

Kennzeichnet neue Begriffe, URLs, E-Mail-Adressen, Dateinamen und Dateiendungen.

Konstante Zeichenbreite

Wird für Programmlistings und für Programmelemente in Textabschnitten wie Namen von Variablen und Funktionen, Datenbanken, Datentypen und Umgebungsvariablen sowie für Anweisungen und Schlüsselwörter verwendet.

Konstante Zeichenbreite, fett

Kennzeichnet Befehle oder anderen Text, den der Nutzer wörtlich eingeben sollte.

Konstante Zeichenbreite, kursiv

Kennzeichnet Text, den der Nutzer je nach Kontext durch entsprechende Werte ersetzen sollte.



Dieses Symbol steht für einen allgemeinen Hinweis.

Verwenden von Codebeispielen

Zusätzliche Materialien (Codebeispiele, Übungen und so weiter) können Sie von der Adresse https://github.com/davidADSP/GDL_code herunterladen. Der Repository-Branch *master* beruht auf der TensorFlow-Backendversion 1.14. Es gibt einen zusätzlichen Branch namens *tensorflow_2*, in dem der Code mit der Ende September 2019 erschienenen TensorFlow-Version 2.0 kompatibel ist.

Dieses Buch dient dazu, Ihnen beim Erledigen Ihrer Arbeit zu helfen. Im Allgemeinen dürfen Sie die Codebeispiele aus diesem Buch in Ihren eigenen Programmen und der dazugehörigen Dokumentation verwenden. Sie müssen uns dazu nicht um Erlaubnis fragen, solange Sie nicht einen beträchtlichen Teil des Codes reproduzieren. Beispielsweise benötigen Sie keine Erlaubnis, um ein Programm zu schreiben, in dem mehrere Codefragmente aus diesem Buch vorkommen. Wollen Sie dagegen eine CD-ROM mit Beispielen aus Büchern von O'Reilly verkaufen oder verteilen, benötigen Sie eine Erlaubnis. Eine Frage zu beantworten, indem Sie aus diesem

Buch zitieren und ein Codebeispiel wiedergeben, benötigt keine Erlaubnis. Eine beträchtliche Menge Beispielcode aus diesem Buch in die Dokumentation Ihres Produkts aufzunehmen, bedarf hingegen einer Erlaubnis.

Wir freuen uns über Zitate, verlangen diese aber nicht. Ein Zitat enthält Titel, Autor, Verlag und ISBN, beispielsweise: »*Generatives Deep Learning* von David Foster (O'Reilly). Copyright 2019 Applied Data Science Partners Ltd., ISBN 978-3-96009-128-8«.

Wenn Sie glauben, dass Ihre Verwendung von Codebeispielen über die übliche Nutzung hinausgeht oder außerhalb der oben vorgestellten Nutzungsbedingungen liegt, kontaktieren Sie uns bitte unter komentar@oreilly.de.

Danksagungen

Es gibt so viele Menschen, denen ich für ihre Hilfe beim Schreiben dieses Buchs danken möchte.

Zunächst möchte ich allen danken, die sich die Zeit genommen haben, das Buch fachlich zu begutachten – insbesondere Luba Elliott, Darren Richardson, Eric George, Chris Schon, Sigurður Skúli Sigurgeirsson, Hao-Wen Dong, David Ha und Lorna Barclay.

Ein großes Dankeschön geht auch an meine Kollegen bei Applied Data Science Partners, Ross Witeszczak, Chris Schon, Daniel Sharp und Amy Bull. Ihre Geduld mit mir, während ich mir die Zeit genommen habe, das Buch fertigzustellen, habe ich sehr geschätzt, und ich freue mich sehr auf all die Machine-Learning-Projekte, die wir in Zukunft gemeinsam angehen werden! Besonderer Dank gilt Ross – hätten wir uns nicht entschieden, zusammen ein Unternehmen zu gründen, wäre dieses Buch vielleicht nie entstanden. Danke also, dass du an mich als deinen Geschäftspartner geglaubt hast!

Ich möchte auch jedem danken, der mir jemals etwas Mathematisches beigebracht hat – ich hatte das große Glück, fantastische Mathematiklehrer in der Schule zu haben, die mein Interesse an diesem Fach geweckt und mich dazu ermutigt haben, es an der Universität weiterzuerfolgen. Ich möchte Ihnen für Ihr großes Engagement danken und dafür, Ihr Wissen über das Thema mit mir geteilt zu haben.

Ein großes Dankeschön geht auch an die Mitarbeiter von O'Reilly, die mich durch den Prozess des Schreibens dieses Buchs geführt haben. Ein besonderer Dank geht an Michele Cronin, die bei jedem Schritt dabei war, nützliches Feedback gab und mich freundlich daran erinnerte, die Kapitel weiter zu vervollständigen! Außerdem möchte ich Katie Tozer, Rachel Head und Melanie Yarbrough danken, weil sie das Buch in Produktion gebracht haben, und Mike Loukides, weil er sich als Erstes an mich gewandt hatte, um mich zu fragen, ob ich nicht ein Buch schreiben wolle. Sie alle haben dieses Projekt von Anfang an so toll unterstützt, und ich möchte Ihnen dafür danken, dass Sie mir eine Plattform zur Verfügung gestellt haben, auf der ich über etwas schreiben kann, das ich liebe.

Während des gesamten Schreibprozesses war meine Familie eine ständige Quelle der Ermutigung und Unterstützung. Ein großes Dankeschön geht an meine Mutter, Gillian Foster, für die Überprüfung jeder einzelnen Textzeile auf Tippfehler und dafür, dass sie mir damals überhaupt erst das Addieren beigebracht hatte! Deine Liebe zum Detail war sehr hilfreich beim Korrekturlesen dieses Buchs, und ich bin überaus dankbar für all die Möglichkeiten, die du und Dad mir gegeben haben. Mein Vater, Clive Foster, hat mir erst mal überhaupt gezeigt, wie man einen Computer programmiert – dieses Buch ist voller praktischer Beispiele, und das dank seiner Geduld damals, während ich in BASIC herumfummelte und versuchte, als Teenager Fußballspiele zu programmieren. Mein Bruder Rob Foster ist das bescheidenste Genie, das Sie je finden werden, besonders in der Linguistik – Gespräche mit ihm über KI und die Zukunft des textbasierten maschinellen Lernens waren extrem hilfreich für mich. Zum Schluss möchte ich meiner Oma danken, die eine ständige Quelle der Inspiration und des Spaßes für uns alle ist. Ihre Liebe zur Literatur ist einer der Gründe dafür, dass ich mich überhaupt dafür entschieden habe, dieses Buch zu schreiben.

Abschließend möchte ich meiner Verlobten (und bald Frau) Lorna Barclay danken. Neben der fachlichen Überprüfung jedes Worts in diesem Buch hat sie mich während des gesamten Schreibprozesses endlos unterstützt, sei es, dass sie mir Tee zubereitet oder verschiedene Snacks gebracht hat oder mir generell dabei half, dieses Buch durch ihre akribische Liebe zum Detail und ihr unglaubliches Expertenwissen über Statistik und maschinelles Lernen zu einem besseren Leitfaden für die generative Modellierung zu machen. Ohne dich hätte ich dieses Projekt mit Sicherheit nicht fertigstellen können, und ich bin dir sehr dankbar für die Zeit, die du investiert hast, um mir bei der Umstrukturierung und Erweiterung der Teile des Buchs zu helfen, die noch das gewisse Etwas benötigten. Ich verspreche dir, dass ich mindestens für ein paar Wochen nach der Veröffentlichung am Esstisch nicht mehr über generative Modellierung sprechen werde.

Einführung ins Generative Deep Learning

Die ersten vier Kapitel dieses Buchs stellen Ihnen die wichtigsten Grundtechniken vor, die Sie benötigen, um generative Deep-Learning-Modelle zu erzeugen.

In Kapitel 1 werfen wir zunächst einen detaillierten Blick auf das Themenfeld der generativen Modellierung und erfassen die Problemstellung aus wahrscheinlichkeitstheoretischer Sicht. Wir werden dann unser erstes Beispiel eines einfachen probabilistischen generativen Modells untersuchen und analysieren, warum wir Deep-Learning-Techniken einsetzen müssen, wenn die Komplexität der generativen Aufgabe steigt.

Kapitel 2 bietet einen Leitfaden zu den Werkzeugen und Methoden des Deep Learning, die Sie für komplexere generative Modelle benötigen. Dies soll mehr ein praktischer Deep-Learning-Leitfaden sein als eine theoretische Analyse des Fachgebiets. Im Besonderen werde ich Ihnen hier Keras vorstellen, ein Framework zur Erzeugung neuronaler Netze, mit dem einige der modernsten neuronalen Netzwerkarchitekturen aus der Literatur konstruiert und trainiert werden können.

In Kapitel 3 werfen wir einen Blick auf unser erstes generatives Deep-Learning-Modell, den Variational Autoencoder. Diese effektive Methode ermöglicht es uns, nicht nur realistische Gesichter zu erzeugen, sondern auch bestehende Bilder zu verändern – zum Beispiel durch das Hinzufügen eines Lächelns oder die Änderung der Haarfarbe.

Kapitel 4 untersucht eine der erfolgreichsten generativen Modellierungstechniken der letzten Jahre, das Generative Adversarial Network. Dieses elegante Framework für eine Modellierungsaufgabe ist der Motor für die meisten modernen generativen Modelle. Wir werden sehen, wie es verfeinert und angepasst wurde, um die Grenzen der generativen Modellierung kontinuierlich zu erweitern.

Generative Modellierung

Dieses Kapitel bietet eine allgemeine Einführung in die generative Modellierung. Zunächst verdeutlichen wir, was es bedeutet, wenn ein Modell *generativ* ist, und stellen die Unterschiede zu den weiter verbreiteten *diskriminativen* Modellen heraus. Im Anschluss werden der Rahmen und die mathematischen Kernideen vorgestellt, mit denen wir unseren allgemeinen Ansatz für Fragestellungen, die eine generative Lösung erfordern, strukturieren können.

Auf dieser Grundlage entwickeln wir anschließend unser erstes Beispiel für ein wahrscheinlichkeitsbasiertes generatives Modell (Naive Bayes). Wir werden feststellen, dass dadurch neue Beispiele erzeugt werden, die nicht unserem Trainingsdatensatz angehören. Zugleich untersuchen wir die Gründe dafür, dass diese Art von Modellierung mit zunehmender Größe und Komplexität möglicher Gestaltungsvarianten fehlschlagen kann.

Was ist generative Modellierung?

Ein generatives Modell kann im Wesentlichen wie folgt definiert werden:

Ein generatives Modell beschreibt, wie ein Datensatz im Rahmen eines Wahrscheinlichkeitsmodells erzeugt wird. Die aus diesem Modell gewonnenen Stichproben ermöglichen uns, neue Daten zu generieren.

Angenommen, wir hätten einen Datensatz mit Bildern von Pferden. Wir möchten ein Modell entwickeln, das ein neues Bild von einem Pferd erzeugen kann, das nie existiert hat, aber trotzdem echt aussieht, weil das Modell die allgemeinen Prinzipien gelernt hat, die das Aussehen eines Pferdes bestimmen. Exakt das ist die Art von Aufgabenstellung, die sich mithilfe der generativen Modellierung lösen lässt. Abbildung 1-1 veranschaulicht den generativen Modellierungsprozess an unserem Beispiel.

Zuerst benötigen wir einen Datensatz, der aus zahlreichen Beispielen der zu erzeugenden Entität besteht. Wir sprechen in diesem Zusammenhang vom *Trainingsdatensatz*, bei dem jeder dieser Datenpunkte als *Beobachtung* bezeichnet wird.

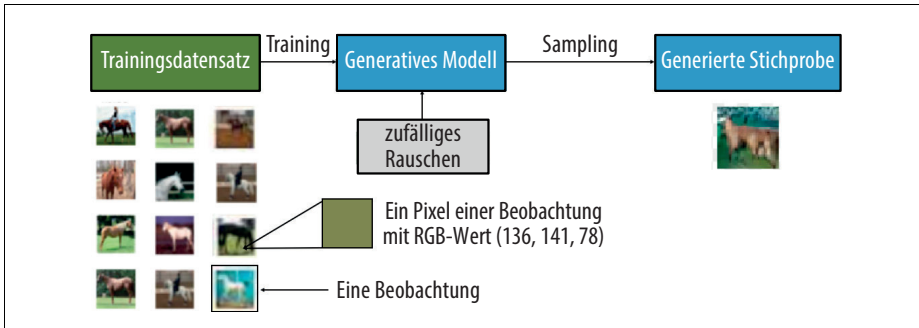


Abbildung 1-1: Der generative Modellierungsprozess

Jede Beobachtung besteht aus vielen *Merkmalen* – bei einer Bilderzeugungsaufgabe sind die Merkmale üblicherweise die jeweiligen Werte der einzelnen Pixel. Unser Ziel ist es, ein Modell zu erstellen, das neue Merkmale erzeugen kann, die so aussehen, als wären sie nach den gleichen Regeln wie die Originaldaten erstellt worden. Rein konzeptionell ist dies für die Bilderzeugung eine unglaublich schwierige Aufgabe – bedenkt man, wie vielfältig die einzelnen Pixelwerte zugeordnet werden können und wie vergleichsweise gering dagegen die Anzahl der Anordnungen ist, die ein Bild der Entität erzeugen, die wir reproduzieren wollen.

Ein generatives Modell sollte zudem *probabilistisch* und nicht *deterministisch* sein. Wenn unser Modell nur eine fest vorgegebene Berechnung umfasst, wie z. B. die Berechnung des Durchschnittswerts jedes Pixels im Datensatz, ist es nicht generativ, da das Modell jedes Mal die gleiche Ausgabe erzeugt. Folglich muss das Modell ein *stochastisches* (zufälliges) Element beinhalten, das die von ihm erzeugten Ausgaben beeinflusst.

Wir können uns das so vorstellen: Es gibt eine unbekannte Wahrscheinlichkeitsverteilung, die erklärt, warum einige Bilder wahrscheinlich im Trainingsdatensatz zu finden sind und andere Bilder hingegen nicht. Unsere Aufgabe ist es, ein Modell zu entwickeln, das diese Verteilung so genau wie möglich nachahmt, um daraus neue, einzigartige Beobachtungen zu generieren, die so aussehen, als würden sie dem ursprünglichen Trainingsdatensatz entstammen.

Vergleich der generativen und diskriminativen Modellierung

Um genau zu verstehen, was generative Modellierung leisten soll und warum sie wichtig ist, ergibt es Sinn, sie mit ihrem Gegenstück, der *diskriminativen Modellierung*, zu vergleichen. Wenn Sie sich in Ihrem Studium mit maschinellem Lernen befassen haben, werden die meisten Fragestellungen, mit denen Sie konfrontiert wurden, höchstwahrscheinlich diskriminativer Natur gewesen sein. Um den Unterschied zu verstehen, schauen wir uns ein Beispiel an.

Angenommen, wir hätten einen Datensatz von mehreren Gemälden. Einige sind von van Gogh und einige von anderen Künstlern. Mit genügend Daten könnten wir ein diskriminatives Modell trainieren, um vorherzusagen, ob ein bestimmtes Bild tatsächlich von van Gogh gemalt wurde. Unser Modell würde lernen, dass bestimmte Farben, Formen und Texturen eher darauf hinweisen, dass ein Gemälde von diesem niederländischen Meister stammt. Für Gemälde mit diesen Merkmalen würde das Modell seine Vorhersage entsprechend höher bewerten. Abbildung 1-2 zeigt den diskriminativen Modellierungsprozess – achten Sie insbesondere darauf, wie er sich von dem in Abbildung 1-1 dargestellten generativen Modellierungsprozess unterscheidet.

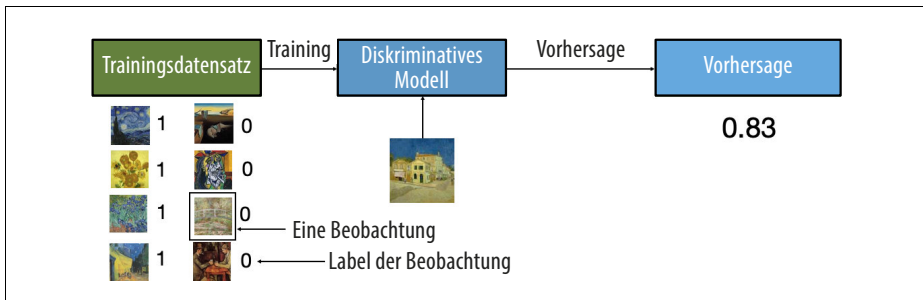


Abbildung 1-2: Der diskriminative Modellierungsprozess

Ein wesentlicher Unterschied besteht darin, dass bei der diskriminativen Modellierung jede Beobachtung in den Trainingsdaten mit einem *Label* versehen ist. Im Fall unserer binären Klassifikationsaufgabe wären Gemälde von van Gogh mit einer *1* und alle anderen mit einer *0* gekennzeichnet. Unser Modell lernt dann, beide Gruppen zu unterscheiden, und gibt die Wahrscheinlichkeit aus, dass eine neue Beobachtung das Label *1* trägt – d. h., dass sie von van Gogh gemalt wurde.

Deshalb ist diskriminative Modellierung gleichbedeutend mit *überwachtem Lernen* beziehungsweise dem Erlernen einer Funktion, die eine Eingabe mithilfe eines mit Labeln versehenen Datensatzes auf eine Ausgabe abbildet. Für die generative Modellierung bedarf es für gewöhnlich nur ungelabelter Datensätze (als Form des unüberwachten Lernens). Sie kann jedoch auch auf einen mit Labeln versehenen Datensatz angewendet werden, um zu lernen, wie man Beobachtungen aus den einzelnen Kategorien erzeugt.

Werfen wir einen Blick auf einige mathematische Notationen, um den Unterschied zwischen generativer und diskriminativer Modellierung zu verdeutlichen.

Diskriminative Modellierung schätzt $p(y|\mathbf{x})$ – die Wahrscheinlichkeit des Labels y , gegeben der Beobachtung \mathbf{x} .

Generative Modellierung schätzt $p(\mathbf{x})$ – die Wahrscheinlichkeit, Beobachtung \mathbf{x} zu beobachten.

Sofern der Datensatz mit Labels versehen ist, können wir auch ein generatives Modell entwickeln, das die Verteilung $p(\mathbf{x}|y)$ schätzt.

Anders ausgedrückt, versucht die diskriminative Modellierung, die Wahrscheinlichkeit abzuschätzen, dass eine Beobachtung \mathbf{x} der Kategorie y angehört. In der generativen Modellierung ist es nicht von Bedeutung, Beobachtungen mit Labels zu versehen. Stattdessen zielt sie darauf ab, die Wahrscheinlichkeit abzuschätzen, dass die Beobachtung überhaupt auftritt.

Der entscheidende Punkt ist: Selbst wenn wir in der Lage wären, ein perfektes diskriminatives Modell zur Identifizierung von Gemälden von van Gogh zu entwickeln, hätte das Modell immer noch keine Ahnung, wie es ein Gemälde erstellt, das wie eines von van Gogh aussieht. Es kann nur Wahrscheinlichkeiten für bereits existente Bilder ausgeben, da es ausschließlich dazu trainiert wurde. Stattdessen müssten wir ein generatives Modell trainieren, das Pixelgruppen ermitteln kann, die eine hohe Wahrscheinlichkeit besitzen, zum ursprünglichen Trainingsdatensatz zu gehören.

Fortschritte im maschinellen Lernen

Um zu verstehen, warum die generative Modellierung als die kommende Herausforderung für das maschinelle Lernen angesehen werden kann, gilt es zu ergründen, warum die diskriminative Modellierung in den letzten zwei Jahrzehnten die treibende Kraft für die meisten Fortschritte in der Methodik des maschinellen Lernens war, sowohl in der Wissenschaft als auch in der Industrie.

Aus akademischer Sicht ist der Fortschritt der diskriminativen Modellierung sicherlich leichter zu überblicken. Mithilfe von Leistungsmetriken können wir für eine Reihe anspruchsvoller Klassifikationsaufgaben messen, welche Methode derzeit am besten abschneidet. Generative Modelle sind oft schwieriger zu bewerten, insbesondere wenn die Qualität der Ergebnisse weitgehend subjektiv ist. Daher wurde in den letzten Jahren vermehrt Wert auf die Entwicklung diskriminativer Modelle gelegt, um menschliche oder gar übermenschliche Leistungen bei einer Vielzahl von Bild- oder Textklassifikationsaufgaben zu erreichen.

Im Jahr 2012 gelang beispielsweise in der Klassifikation von Bildern der entscheidende Durchbruch, als ein Team um Geoff Hinton von der Universität Toronto mit einem tiefen neuronalen Konvolutionsnetz die *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) gewann. Der Wettbewerb umfasst die Klassifikation von Bildern in eine von 1.000 Kategorien und dient als Benchmark für den Vergleich neuester Methoden. Das Deep-Learning-Modell hatte eine Fehlerquote

von 16% – eine massive Verbesserung gegenüber dem nächstbesten Modell, das lediglich eine Fehlerquote von 26,2% erreichte. Dies löste einen Deep-Learning-Boom aus, der dazu führte, dass die Fehlerquote Jahr für Jahr noch weiter zurückging. Der Gewinner des Jahres 2015 erreichte mit einer Fehlerquote von 4% erstmals eine übermenschliche Leistung, und das aktuell modernste Modell weist eine Fehlerquote von nur 2% auf. Für viele erscheint die Aufgabenstellung nun als gelöst.

Abgesehen von der Tatsache, dass es einfacher ist, messbare Ergebnisse in einem akademischen Umfeld zu veröffentlichen, war die diskriminative im Vergleich zur generativen Modellierung in der Vergangenheit leichter auf Geschäftsprobleme anzuwenden. Im Regelfall interessiert es uns in einem Geschäftsumfeld nicht, wie die Daten generiert wurden. Stattdessen wollen wir wissen, wie ein neues Beispiel kategorisiert oder bewertet werden kann, zum Beispiel:

- Bei einem Satellitenbild würde sich ein Verteidigungsfunktionär nur um die Wahrscheinlichkeit kümmern, dass es feindliche Einheiten enthält, und nicht, wie wahrscheinlich es ist, dass dieses spezielle Bild erscheint.
- Ein Kundenbetreuer wäre nur daran interessiert, zu wissen, ob die Wahrnehmung einer eingehenden E-Mail positiv oder negativ ist, und würde aus einem generativen Modell, das Beispiele für noch nicht existierende Kunden-E-Mails ausgeben könnte, keinen großen Nutzen ziehen.
- Ein Arzt möchte wissen, mit welcher Wahrscheinlichkeit ein bestimmtes Netzhautbild auf Grünen Star hinweist, und nicht auf ein Modell zurückgreifen, das neue Bilder von der Rückseite eines Auges erzeugen kann.

Da die meisten von Unternehmen benötigten Lösungen dem Gebiet der diskriminativen Modellierung entstammen, ist die Anzahl der *Machine-Learning-as-a-Service-Tools* (MLaaS) gestiegen. Sie zielen darauf ab, den Einsatz diskriminativer Modellierung in der Industrie zu kommerzialisieren, indem sie die Erstellungs-, Validierungs- und Überwachungsprozesse, die in fast allen diskriminativen Modellierungsaufgaben vorkommen, weitgehend automatisieren.

Der Vormarsch der generativen Modellierung

Obwohl die diskriminative Modellierung bisher der größte Impulsgeber für Entwicklungen auf dem Gebiet des maschinellen Lernens war, sind in den letzten drei bis fünf Jahren viele der bedeutendsten Fortschritte durch neue Anwendungen des Deep Learning in generativen Modellierungsaufgaben erzielt worden.

Die mediale Aufmerksamkeit stieg insbesondere durch generative Modellierungsprojekte wie StyleGAN von NVIDIA,¹ das in der Lage ist, hyperrealistische Bilder von menschlichen Gesichtern zu erzeugen, und durch das Sprachmodell GPT-2

1 Tero Karras, Samuli Laine und Timo Aila, »A Style-Based Generator Architecture for Generative Adversarial Networks«, 12. Dezember 2018, <https://arxiv.org/abs/1812.04948>.

von OpenAI², das eine Textpassage auf Basis eines kurzen einleitenden Absatzes vervollständigen kann.

Abbildung 1-3 zeigt die bemerkenswerten Fortschritte, die seit dem Jahr 2014 bei der Erzeugung von Gesichtsbildern bereits erzielt wurden.³ Hier gibt es sicherlich nützliche Anwendungen für Branchen wie etwa die Spielegestaltung oder die Filmproduktion. Verbesserungen in der automatischen Erzeugung von Musik werden sicherlich auch in diesen Bereichen immer mehr Beachtung finden. Es wird sich zeigen, ob wir eines Tages Nachrichtenartikel oder Romane lesen werden, die von einem generativen Modell geschrieben wurden. Die jüngsten Fortschritte in diesem Bereich sind enorm. Deshalb ist es sicherlich nicht vermessen, zu behaupten, dass dies eines Tages durchaus der Fall sein könnte. Aber auch wenn es reizvoll erscheint, so wirft es durchaus ethische Fragen rund um die Verbreitung von gefälschten Inhalten im Internet auf, und es könnte immer schwieriger werden, dem zu vertrauen, was wir in den öffentlichen Kommunikationskanälen wahrnehmen und lesen.



Abbildung 1-3: Die Gesichtsbilderzeugung mittels generativer Modellierung hat sich in den letzten vier Jahren deutlich verbessert.⁴

Neben den praktischen Einsatzmöglichkeiten der generativen Modellierung (von denen viele noch unentdeckt sind) gibt es drei wesentliche Gründe dafür, dass sie als Schlüssel zur Erschließung einer weitaus anspruchsvolleren Form der künstlichen Intelligenz angesehen werden kann, die über das hinausgeht, was die diskriminative Modellierung zu leisten vermag.

Erstens sollten wir uns rein theoretisch nicht damit zufriedengeben, dass wir uns lediglich bei der Kategorisierung von Daten auszeichnen können, sondern auch ein umfassenderes Verständnis dafür anstreben, wie die Daten überhaupt generiert wurden. Das ist zweifelsohne ein schwerer zu lösendes Problem, da etliche Ausgaben denkbar sind, jedoch nur wenige davon als zum Datensatz gehörend eingestuft werden würden. Wie wir sehen werden, können viele der Methoden, die bereits die

2 Alec Radford et al., »Language Models Are Unsupervised Multitask Learners«, 2019, <https://paperswithcode.com/paper/language-models-are-unsupervised-multitask>.

3 Miles Brundage et al., »The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation«, Februar 2018, https://www.eff.org/files/2018/02/20/malicious_ai_report_final.pdf.

4 Quelle: Brundage et al., 2018.

Entwicklung in der diskriminativen Modellierung vorangetrieben haben, wie z.B. Deep Learning, auch von generativen Modellen genutzt werden.

Zweitens ist es sehr wahrscheinlich, dass die generative Modellierung künftig auch bei Entwicklungen in anderen Bereichen des maschinellen Lernens von zentraler Bedeutung sein wird, wie z.B. beim Reinforcement Learning (der Analyse von Agenten, die in einer bestimmten Umgebung mittels Trial-and-Error ihr Ziel optimieren). Beispielsweise könnten wir das Reinforcement Learning nutzen, um einen Roboter so zu trainieren, dass er über ein bestimmtes Gelände läuft. Der Grundansatz besteht darin, eine Computersimulation des Geländes zu erstellen und dann viele Experimente durchzuführen, bei denen der Agent verschiedene Strategien ausprobiert. Im Laufe der Zeit würde der Agent lernen, welche Strategien erfolgreicher als andere sind, und sich sukzessiv verbessern. Ein typisches Problem bei diesem Ansatz ist, dass die physikalischen Gegebenheiten der Umgebung oft sehr komplex sind und bei jedem Teilschritt berechnet werden müssten, um die Informationen an den Agenten zurückzugeben, auf deren Basis er dann über seinen nächsten Zug entscheiden kann. Wenn der Agent jedoch in der Lage wäre, seine Umgebung durch ein generatives Modell zu simulieren, müsste er die Strategie nicht in der Computersimulation oder in der realen Welt testen, sondern könnte stattdessen in seiner eigenen *imaginären* Umgebung lernen. In Kapitel 8 setzen wir diese Idee in die Praxis um. Wir trainieren ein Auto darauf, so schnell wie möglich auf einer Strecke zu fahren, indem es selbst aus seiner eigenen halluzinierten Umgebung lernen kann.

Nicht zuletzt muss die generative Modellierung sicherlich Teil der Lösung sein, um eine Maschine bauen zu können, die eine Form der Intelligenz erlangt, die der eines Menschen gleicht. Eines der besten Beispiele für ein real existierendes generatives Modell ist die Person, die dieses Buch liest – Sie! Nehmen Sie sich einen Moment Zeit, um zu überlegen, was für ein unglaubliches generatives Modell Sie sind. Sie vermögen die Augen zu schließen und sich vorzustellen, wie ein Elefant aus jedem möglichen Blickwinkel aussehen würde. Sie können sich eine Reihe von unterschiedlichen plausiblen Ausgängen Ihrer Lieblingsserie vorstellen. Und Sie können Ihre Woche im Voraus planen, indem Sie verschiedene Zukunftsszenarien vor Ihrem geistigen Auge durcharbeiten und sich für entsprechende Handlungen entscheiden. Die aktuelle neurowissenschaftliche Theorie legt nahe, dass unsere Wahrnehmung der Realität kein hochkomplexes, auf unseren Sinneseindrücken basierendes diskriminatives Modell ist, um Vorhersagen über das, was wir erleben, zu treffen. Vielmehr gleicht sie einem generativen Modell, das von Geburt an trainiert wird, um Simulationen unserer Umgebung zu liefern, die der Zukunft entsprechen. Einige Theorien legen sogar nahe, dass die Ergebnisse dieses generativen Modells das sind, was wir direkt als Realität wahrnehmen. Offensichtlich ist ein umfassendes Verständnis dafür, wie wir Maschinen bauen können, die diese Fähigkeit erwerben, von zentraler Bedeutung für unser weiteres Verständnis der Funktionsweise des Gehirns und der allgemeinen künstlichen Intelligenz.

Beginnen wir vor diesem Hintergrund unsere Reise in die spannende Welt der generativen Modellierung. Zunächst schauen wir uns die einfachsten Beispiele für generative Modelle und einige der Ideen an, die uns dabei helfen, die komplexeren Architekturen, denen wir später im Buch begegnen werden, durchleuchten zu können.

Das generative Modellierungskonzept

Spielen wir als Erstes ein kleines Spiel mit einem generativen Modell, das nur zwei Dimensionen besitzt. Hierzu habe ich eine Regel gewählt, mit der die Punkte \mathbf{X} in Abbildung 1-4 erzeugt wurden. Nennen wir diese Regel p_{data} . Ihre Aufgabe besteht darin, einen anderen Punkt $\mathbf{x} = (x_1, x_2)$ in diesem Raum zu wählen, der so erscheint, als wäre er von der gleichen Regel erzeugt worden.

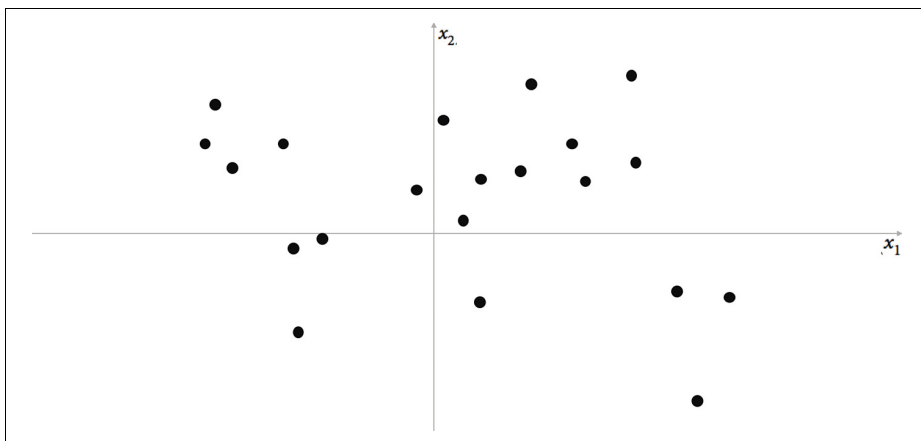


Abbildung 1-4: Eine Punktmenge im zweidimensionalen Raum, die durch die unbekannte Regel p_{data} erzeugt wurde

Für welchen Punkt haben Sie sich entschieden? Wahrscheinlich haben Sie Ihre Kenntnisse über die vorhandenen Datenpunkte genutzt, um vor Ihrem geistigen Auge ein Modell, p_{model} , zu konstruieren, mit dem Sie abschätzen, an welchen Stellen die Punkte eher zu finden sind. In diesem Zusammenhang stellt p_{model} eine Schätzung von p_{data} dar. Vielleicht haben Sie entschieden, dass p_{model} wie Abbildung 1-5 aussehen sollte – eine rechteckige Fläche, in der Punkte aufzufinden sind, und ein Bereich außerhalb dieser Fläche, in dem keine Wahrscheinlichkeit besteht, einen Punkt zu finden. Um eine neue Beobachtung zu erzeugen, können Sie einfach einen zufälligen Punkt innerhalb der Fläche wählen – oder, formeller ausgedrückt, aus der Verteilung p_{model} eine Stichprobe ziehen beziehungsweise *samplen*. Herzlichen Glückwunsch, Sie haben gerade Ihr erstes generatives Modell entwickelt!

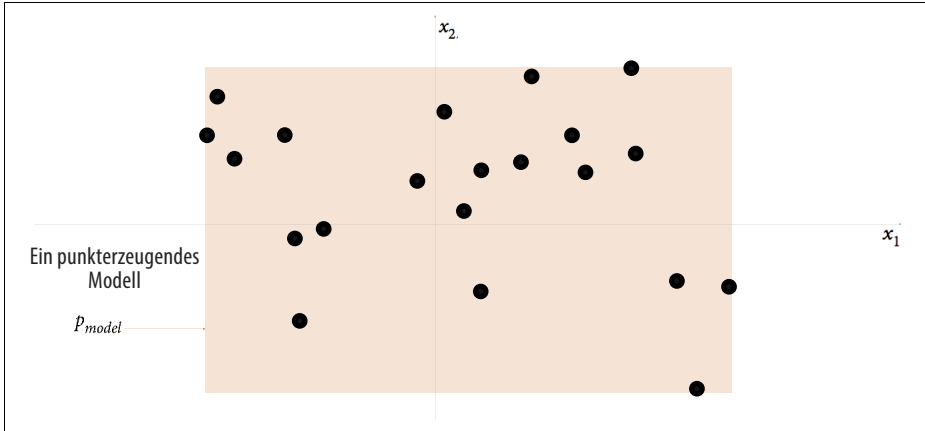


Abbildung 1-5: Die orangefarbene Fläche, p_{model} , ist eine Schätzung der wahren zugrunde liegenden datengenerierenden Verteilung, p_{data} .

Wenn dies auch nicht das komplexeste Beispiel gewesen sein mag, so haben wir dennoch einen ersten Einblick darin gewinnen können, worauf generative Modellierung abzielt. Fassen wir die Grundprinzipien noch einmal zusammen.

Die Grundprinzipien der generativen Modellierung

- Wir haben einen Datensatz an Beobachtungen X .
- Wir gehen davon aus, dass die Beobachtungen auf der Grundlage einer unbekanntem Verteilung, p_{data} , generiert wurden.
- Ein generatives Modell p_{model} versucht, p_{data} nachzubilden. Wenn wir dies erreichen, können wir aus p_{model} Stichproben ziehen, um Beobachtungen zu erzeugen, die auch p_{data} entstammen könnten.
- Uns überzeugt p_{model} , falls folgende Regeln gelten:
 - Regel 1: Es kann Beispiele erzeugen, die ebenfalls aus p_{data} stammen könnten.
 - Regel 2: Es kann Beispiele generieren, die sich in ausreichendem Maße von den Beobachtungen in X unterscheiden. Mit anderen Worten, das Modell sollte nicht einfach Dinge reproduzieren, die es zuvor gesehen hat.

Ergünden wir nun die wahre datengenerierende Verteilung p_{data} und übertragen wir darauf die Grundprinzipien.

Wie wir in Abbildung 1-6 sehen können, ist der zugrunde liegende Datengenerierungsprozess schlicht eine Gleichverteilung der Punkte über die Kontinente hinweg, wobei kein Punkt im Meer zu finden ist.

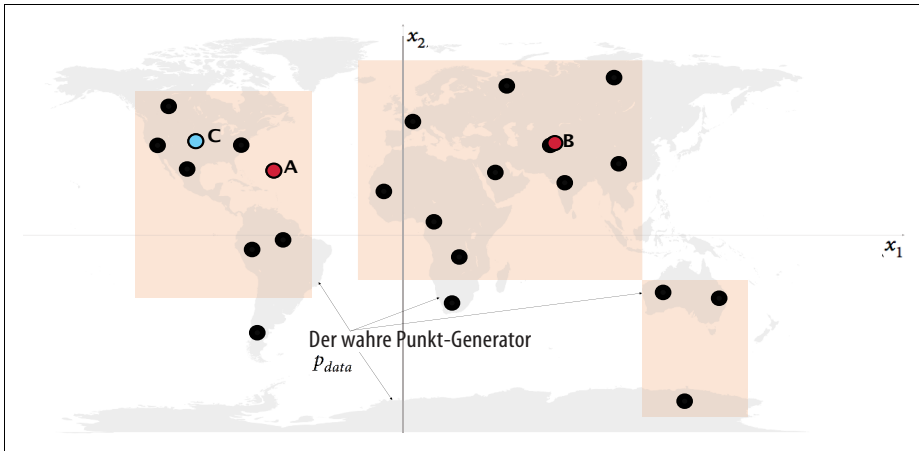


Abbildung 1-6: Die orangefarbenen Flächen, p_{model} , stellen eine Schätzung der tatsächlichen datenerzeugenden Verteilung, p_{data} , dar (die grauen Flächen).

Offensichtlich ist unser Modell p_{model} eine zu starke Vereinfachung von p_{data} . Die Punkte A, B und C heben drei Beobachtungen hervor, die von p_{model} mit unterschiedlichem Erfolg erzeugt wurden:

- **Punkt A** verstößt gegen Regel 1 – er scheint nicht von p_{data} generiert worden zu sein, da er sich mitten auf dem Meer befindet.
- **Punkt B** ist so nah an einem Punkt im Datensatz, dass wir nicht damit zufrieden sein sollten, wenn unser Modell einen solchen Punkt erzeugt. Wenn alle vom Modell erzeugten Beispiele so wären, würde dies gegen Regel 2 verstoßen.
- **Punkt C** kann als Erfolg gewertet werden, da er von p_{data} generiert worden sein könnte und sich in ausreichendem Maße von jedem Punkt im ursprünglichen Datensatz unterscheidet.

Das Anwendungsfeld der generativen Modellierung ist vielfältig, und es sind unterschiedlichste Aufgabenstellungen denkbar. In den meisten Anwendungsfällen bieten die Grundprinzipien der generativen Modellierung jedoch einen Leitfaden dazu, wie wir im Großen und Ganzen über die Lösung der Aufgabe nachdenken sollten.

Befassen wir uns nun mit einem weniger trivialen Beispiel für ein generatives Modell.

Wahrscheinlichkeitsbasierte generative Modelle

Eines vorweg – auch wenn Sie nie Statistik studiert haben, sollten Sie sich keine Sorgen machen! Um die meisten Deep-Learning-Modelle, die wir später in diesem Buch sehen werden, zu entwickeln und zu verwenden, ist es nicht notwendig, ein

ausgeprägtes Wissen über die dahinterstehende statistische Theorie zu besitzen. Um jedoch eine umfassende Einschätzung der Entwicklungen zu gewinnen, lohnt es sich, ein generatives Modell zu entwickeln, das nicht auf Methoden des Deep Learning basiert, sondern ausschließlich auf der Wahrscheinlichkeitstheorie. Dieser Ansatz vermittelt Ihnen die Grundlagen, um alle generativen Modelle, ob basierend auf Deep Learning oder nicht, aus der gleichen wahrscheinlichkeitstheoretischen Sichtweise verstehen zu können.



Wenn Sie bereits über profunde statistische Kenntnisse verfügen, ist das großartig, und ein Großteil des folgenden Abschnitts dürfte Ihnen geläufig sein. Es sei jedoch auf ein lustiges Beispiel in der Mitte dieses Kapitels hingewiesen, das sich zu lesen lohnt!

Als Erstes definieren wir vier Grundbegriffe: *Stichprobenraum*, *Wahrscheinlichkeitsdichtefunktion*, *parametrische Modellierung* und *Maximum-Likelihood-Schätzung*.

Der Stichprobenraum

Der *Stichprobenraum* ist der gesamte Wertebereich, den eine Beobachtung \mathbf{x} annehmen kann.

In unserem vorherigen Beispiel besteht der Stichprobenraum aus allen auf der Weltkarte möglichen Kombinationen aus Breiten- und Längengrad $\mathbf{x} = (x_1, x_2)$.

$\mathbf{x} = (40,7306^\circ, -73,9352^\circ)$ ist z.B. ein Punkt im Stichprobenraum (New York City).

Die Wahrscheinlichkeitsdichtefunktion

Eine *Wahrscheinlichkeitsdichtefunktion* (oder *Dichtefunktion*), $p(\mathbf{x})$, ist eine Funktion, die einen Punkt \mathbf{x} des Stichprobenraums auf einen Wertebereich zwischen 0 und 1 abbildet. Die Summe⁵ der Dichtefunktion aller Punkte im Stichprobenraum muss 1 betragen, damit die Wahrscheinlichkeitsverteilung eindeutig definiert ist.⁶

Im unserem Weltkartenbeispiel ist die Dichtefunktion unseres Modells außerhalb der orangefarbenen Bereiche 0 und innerhalb konstant.

Obwohl es nur eine wahre Dichtefunktion p_{data} gibt, von der angenommen wird, dass sie den beobachtbaren Datensatz erzeugt hat, gibt es unendlich viele Dichte-

⁵ Beziehungsweise das Integral, wenn der Stichprobenraum kontinuierlich ist.

⁶ Wenn der Stichprobenraum diskret ist, entspricht $p(\mathbf{x})$ der dem Beobachtungspunkt \mathbf{x} zuzuordnenden Wahrscheinlichkeit.

funktionen p_{model} , mit denen wir p_{data} schätzen können. Um unseren Ansatz zur Suche nach einem geeigneten $p_{\text{model}}(\mathbf{X})$ zu strukturieren, können wir eine Technik verwenden, die als *parametrische Modellierung* bekannt ist.

Parametrische Modellierung

Ein *parametrisches Modell*, $p_{\theta}(\mathbf{x})$, stellt eine Gruppe von Dichtefunktionen dar, die mit einer endlichen Anzahl an Parametern, θ , beschrieben werden können.

Die Gruppe aller möglichen Flächen, die man in Abbildung 1-5 zeichnen kann, stellen ein Beispiel eines parametrischen Modells dar. In diesem Fall gibt es vier Parameter: die Koordinaten der linken unteren (θ_1, θ_2) und der rechten oberen (θ_3, θ_4) Ecke der Fläche.

In diesem parametrischen Modell kann somit jede Dichtefunktion $p_{\theta}(\mathbf{x})$ (d. h. jede Fläche) eindeutig durch vier Zahlen, $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$, dargestellt werden.

Die Wahrscheinlichkeitsfunktion

Die *Wahrscheinlichkeitsfunktion* oder auch *Likelihood* $\mathcal{L}(\theta|\mathbf{x})$ eines Parametersatzes θ ist eine Funktion, die die Wahrscheinlichkeit von θ unter Berücksichtigung eines beobachteten Punkts \mathbf{x} misst.

Sie ist wie folgt definiert:

$$\mathcal{L}(\theta|\mathbf{x}) = p_{\theta}(\mathbf{x})$$

Das bedeutet, dass die Wahrscheinlichkeit von θ bei einem Beobachtungspunkt \mathbf{x} durch den Wert der durch θ parametrisierten Dichtefunktion an dem Punkt \mathbf{x} definiert ist.

Wenn wir einen kompletten Datensatz \mathbf{X} unabhängiger Beobachtungen haben, dann können wir formulieren:

$$\mathcal{L}(\theta|\mathbf{X}) = \prod_{\mathbf{x} \in \mathbf{X}} p_{\theta}(\mathbf{x})$$

Da die Berechnung dieses Produkts ziemlich rechenintensiv sein kann, verwenden wir stattdessen oft die *Log-Likelihood* ℓ :

$$\ell(\theta|\mathbf{X}) = \sum_{\mathbf{x} \in \mathbf{X}} \log p_{\theta}(\mathbf{x})$$

Es gibt statistische Gründe dafür, dass die Wahrscheinlichkeit auf diese Weise definiert wird. Zu unseren Zwecken genügt es, die Idee dahinter nachvollziehen zu können. Wir definieren einfach die Wahrscheinlichkeit, dass ein Parametersatz θ gleich der Wahrscheinlichkeit ist, die Daten unter dem von θ parametrisierten Modell zu erfassen.