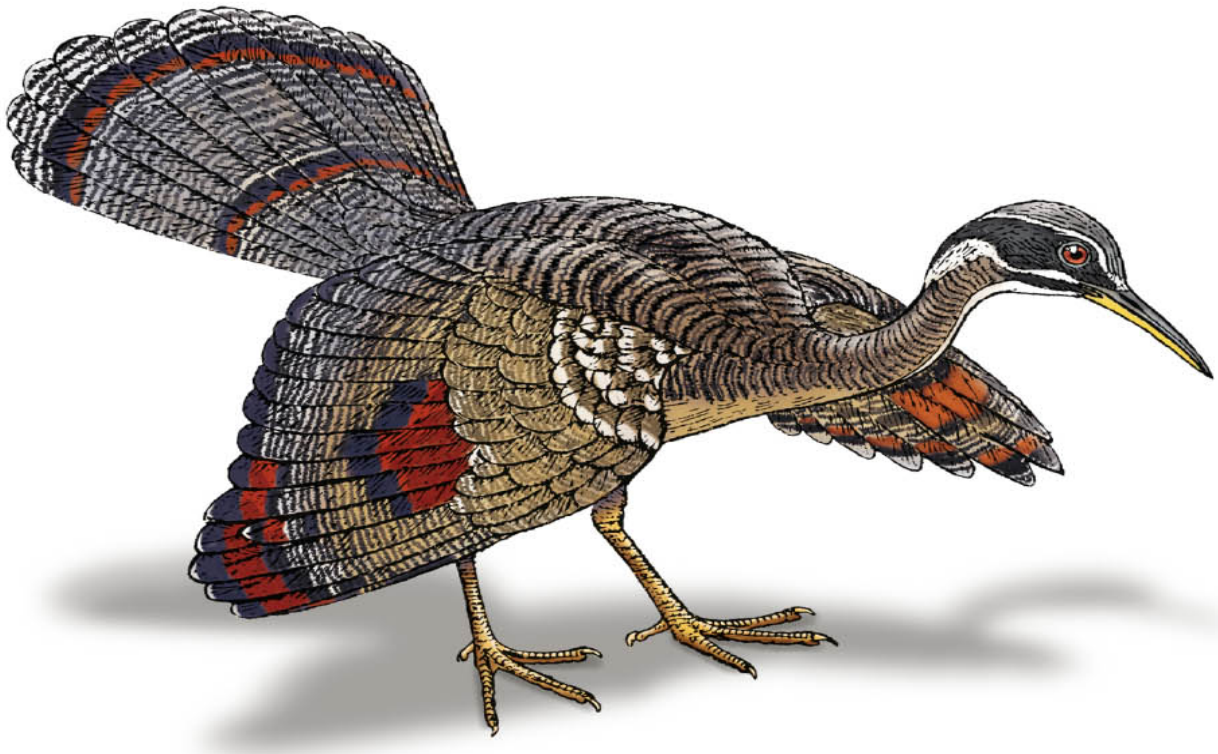
O'REILLY®

Best Practices für die gesamte ML-Pipeline

Design Patterns für Machine Learning

Entwurfsmuster für Datenaufbereitung, Modellbildung und MLOps



Valliappa Lakshmanan,
Sara Robinson & Michael Munn

Übersetzung von Frank Langenau

Design Patterns für Machine Learning

Entwurfsmuster für Datenaufbereitung, Modellbildung und MLOps

Valliappa Lakshmanan, Sara Robinson & Michael Munn

Deutsche Übersetzung von Frank Langenau

O'REILLY®

Valliappa Lakshmanan, Sara Robinson und Michael Munn

Lektorat: Alexandra Follenius
Übersetzung: Frank Langenau
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*

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	978-3-96009-164-6
PDF	978-3-96010-596-1
ePub	978-3-96010-597-8
mobi	978-3-96010-598-5

1. Auflage 2022

Wieblinger Weg 17
69123 Heidelberg

Authorized German translation of the English edition of *Machine Learning Design Patterns*, ISBN 9781098115784 © 2021 Valliappa Lakshmanan, Sara Robinson, and Michael Munn. 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: *kommentar@oreilly.de*.

5 4 3 2 1 0

Inhalt

8 Verbundene Muster

Index

Vorwort

Für wen dieses Buch gedacht ist

Einführende Bücher zum Machine Learning konzentrieren sich normalerweise auf das *Was* und *Wie* des maschinellen Lernens (ML). Sie erläutern dann die mathematischen Aspekte neuer Methoden aus KI-Forschungseinrichtungen und lehren, wie sich diese Methoden mithilfe von KI-Frameworks implementieren lassen. Dieses Buch hingegen bringt die hart erarbeiteten Erfahrungen rund um das *Warum* zusammen, das den Tipps und Tricks zugrunde liegt, auf die erfahrene ML-Praktiker:innen setzen, wenn sie maschinelles Lernen auf reale Probleme anwenden.

Wir gehen davon aus, dass Sie über Vorkenntnisse zu maschinellem Lernen und Datenverarbeitung verfügen. Dies ist kein Grundlagenlehrbuch für maschinelles Lernen. Vielmehr richtet sich dieses Buch an Data Scientists oder ML Engineers, die nach einem zweiten Buch über praktisches Machine Learning suchen. Wenn Sie die Grundlagen bereits kennen, präsentiert Ihnen dieses Buch einen Katalog von Ideen, von denen Sie (als ML-Praktiker:in) vielleicht einige wiedererkennen, und gibt diesen Ideen einen Namen, damit Sie zielsicher nach ihnen greifen können.

Wenn Sie Informatikstudent:in sind und einen Job in der Industrie anstreben, wird dieses Buch Ihr Wissen abrunden und Sie auf die Berufswelt vorbereiten. Es wird Ihnen helfen, zu lernen, wie man hochwertige ML-Systeme aufbaut.

Was Sie nicht im Buch finden

In erster Linie richtet sich das Buch an ML Engineers in Unternehmen, nicht an ML-Wissenschaftler:innen in akademischen oder industriellen Forschungslabors.

Wir gehen absichtlich nicht auf Bereiche der aktiven Forschung ein – so werden Sie hier sehr wenig zur Modellarchitektur des maschinellen Lernens finden (wie zum Beispiel bidirektionale Encoder, den Aufmerksamkeitsmechanismus oder Kurzschlussschichten), da wir annehmen, dass Sie mit einer vorgefertigten Modellarchitektur (wie etwa ResNet-50 oder GRUCell) arbeiten und nicht Ihr eigenes Bildklassifizierungs- oder rekurrentes neuronales Netz schreiben werden.

Die folgenden Punkte listen einige konkrete Beispiele für Bereiche auf, von denen wir uns absichtlich fernhalten, da wir glauben, dass diese Themen eher für Hochschulkurse und ML-Forscher geeignet sind:

ML-Algorithmen

Zum Beispiel behandeln wir nicht die Unterschiede zwischen Random Forests und neuronalen Netzen. Damit beschäftigen sich Lehrbücher, die in das maschinelle Lernen einführen.

Bausteine

Verschiedene Arten von Optimierern für den Gradientenabstieg oder Aktivierungsfunktionen

behandeln wir ebenfalls nicht. Wir empfehlen hier Adam und ReLU – unserer Erfahrung nach ist das Potenzial für Verbesserungen der Performance durch verschiedene Auswahlen bei derartigen Dingen eher gering.

ML-Modellarchitekturen

Wenn Sie Bilder klassifizieren, verwenden Sie am besten ein Standardmodell wie ResNet oder was auch immer der letzte Schrei ist, wenn Sie dies lesen. Überlassen Sie den Entwurf neuer Modelle für die Bild- oder Textklassifizierung den Forschern, die sich auf derartige Probleme spezialisiert haben.

Modellebenen

In diesem Buch werden Sie keine Convolutional Neural Networks und keine rekurrenten neuronalen Netze finden, und zwar gleich aus zwei Gründen – erstens, weil sie Bausteine sind, und zweitens, weil sie zu den Dingen gehören, die bereits als Standardlösungen zur Verfügung stehen.

Benutzerdefinierte Trainingsschleifen

Der einfache Aufruf von `model.fit()` in Keras dürfte den Ansprüchen von Praktikerinnen und Praktikern genügen.

Wir haben versucht, in dieses Buch nur gängige Muster aufzunehmen – wie sie etwa ML Engineers in Unternehmen bei ihrer täglichen Arbeit verwenden.

Nehmen Sie als Analogie dazu Datenstrukturen. Während ein Hochschulkurs über Datenstrukturen eingehend die Implementierungen der verschiedenen Datenstrukturen erläutert und ein Forscher zu Datenstrukturen lernen muss, wie man ihre mathematischen Eigenschaften formal darstellt, können Fachleute pragmatischer herangehen. Entwickler:innen von Unternehmenssoftware müssen einfach wissen, wie sie effektiv mit Arrays, verketteten

Listen, Mengen und Bäumen arbeiten können. Dieses Buch ist für pragmatische Fachleute des maschinellen Lernens geschrieben.

Codebeispiele

Wir stellen Code für maschinelles Lernen (manchmal in Keras/TensorFlow, manchmal in scikit-learn oder BigQuery ML) und Datenverarbeitung (in SQL) zur Verfügung, um zu zeigen, wie die im Buch beschriebenen Techniken in der Praxis umgesetzt werden. Der gesamte Code, auf den im Buch verwiesen wird, ist Teil unseres GitHub-Repositorys (*https://github.com/GoogleCloudPlatform/ml-design-patterns*), in dem Sie voll funktionsfähige ML-Modelle finden. Diese Codebeispiele sollten Sie unbedingt ausprobieren.

Gegenüber den behandelten Konzepten und Techniken spielt der Code nur eine untergeordnete Rolle. Unser Ziel ist es gewesen, dass Thema und Prinzipien unabhängig von Änderungen an TensorFlow oder Keras relevant bleiben. Wir können uns durchaus vorstellen, das GitHub-Repository zum Beispiel mit anderen ML-Frameworks zu ergänzen, während der Buchtext unverändert bleibt. Daher sollte das Buch für Sie genauso informativ sein, wenn Ihr primäres ML-Framework PyTorch oder sogar ein Nicht-Python-Framework wie H20.ai oder R ist. Darüber hinaus begrüßen wir es, wenn Sie die Implementierung eines Musters (es dürfen auch mehrere sein) in Ihrem bevorzugten ML-Framework als Beitrag für das GitHub-Repository bereitstellen.

Dieses Buch soll Ihnen bei Ihrer täglichen Arbeit helfen. Falls Beispielcode zum Buch angeboten wird, dürfen Sie ihn im Allgemeinen in Ihren Programmen und für

Dokumentationen verwenden. Sie müssen uns nicht um Erlaubnis bitten, es sei denn, Sie kopieren einen erheblichen Teil des Codes. Wenn Sie zum Beispiel ein Programm schreiben, das einige Codeblöcke aus diesem Buch verwendet, benötigen Sie keine Erlaubnis. Sollten Sie aber Beispiele aus O'Reilly-Büchern verkaufen oder verbreiten, ist eine Erlaubnis erforderlich. Wenn Sie eine Frage beantworten und dabei dieses Buch oder Beispielcode aus diesem Buch zitieren, brauchen Sie wiederum keine Erlaubnis. Aber wenn Sie große Teile des Beispielcodes aus diesem Buch in die Dokumentation Ihres Produkts einfließen lassen, ist eine Erlaubnis einzuholen.

Wir schätzen eine Quellenangabe, verlangen sie aber nicht. Eine Quellenangabe umfasst in der Regel Titel, Autor, Verlag und ISBN. Zum Beispiel: »*Design Patterns für Machine Learning* von Valliappa Lakshmanan, Sara Robinson und Michael Munn (O'Reilly). Copyright 2022 dpunkt.verlag, ISBN 978-3-96009-164-6.« Wenn Sie der Meinung sind, dass Sie die Codebeispiele in einer Weise verwenden, die über die oben erteilte Erlaubnis hinausgeht, kontaktieren Sie uns bitte unter *kommentar@oreilly.de*.

Typografischen Konventionen

In diesem Buch folgen wir diesen typografischen Konventionen:

Kursiv
: Kennzeichnet neue Begriffe, URLs, E-Mail-Adressen, Dateinamen und Dateierweiterungen.

`Schreibmaschinenschrift`
: Wird in Programmlistings verwendet und im Fließtext für Programmelemente wie zum Beispiel Variablen- oder

Funktionsnamen, Datenbanken, Datentypen, Umgebungsvariablen, Anweisungen und Schlüsselwörter.

`Schreibmaschinenschrift fett`

Kennzeichnet Befehle oder andere Texte, die vom Benutzer buchstäblich eingegeben werden sollen.

`Schreibmaschinenschrift kursiv`

Zeigt Text, der ersetzt werden soll, durch Werte, die der Benutzer bereitstellt, oder Werte, die sich aus dem Kontext ergeben.

Dieses Element kennzeichnet einen Tipp oder Vorschlag.

Dieses Element kennzeichnet einen allgemeinen Hinweis.

Dieses Element kennzeichnet eine Warnung oder einen Achtungshinweis.

Danksagungen

Ein Buch wie dieses wäre nicht möglich ohne die Großzügigkeit zahlreicher Googler, insbesondere unserer Kolleg:innen aus den Teams Cloud AI, Solution

Engineering, Professional Services und Developer Relations. Wir sind ihnen dankbar, dass wir ihre Lösungen für die herausfordernden Probleme beim Training, bei der Verbesserung und der Operationalisierung von ML-Modellen beobachten, analysieren und hinterfragen durften. Wir danken unseren Managern Karl Weinmeister, Steve Cellini, Hamidou Dia, Abdul Razack, Chris Hallenbeck, Patrick Cole, Louise Byrne und Rochana Golani dafür, dass sie den Geist der Offenheit bei Google fördern und uns die Freiheit geben, diese Muster zu katalogisieren und dieses Buch zu veröffentlichen.

Salem Haykal, Benoit Dherin und Khalid Salama haben jedes Muster und jedes Kapitel durchgesehen. Sal hat uns auf Feinheiten hingewiesen, die wir übersehen hatten, Benoit hat unsere Behauptungen eingegrenzt, und Khalid hat uns auf relevante Forschungsarbeiten aufmerksam gemacht. Ohne ihre Beiträge wäre dieses Buch bei Weitem nicht so gut geworden. Vielen Dank dafür! Amy Unruh, Rajesh Thallam, Robbie Haertel, Zhitao Li, Anusha Ramesh, Ming Fang, Parker Barnes, Andrew Zaldivar, James Wexler, Andrew Sellergren und David Kanter haben die Teile dieses Buchs, die in ihre Fachgebiete fallen, überprüft und zahlreiche Vorschläge dazu unterbreitet, wie die kurzfristige Roadmap unsere Empfehlungen beeinflussen würde. Nitin Aggarwal und Matthew Yeager haben das Auge des Lesers in das Manuskript eingebracht und dessen Klarheit verbessert. Besonderer Dank gebührt Rajesh Thallam, der den Prototyp für das Design der allerletzten Abbildung in Kapitel 8 erstellt hat. Alle verbliebenen Fehler gehen natürlich auf unsere Kappe.

O’Reilly ist der Verlag der Wahl für technische Bücher, und die Professionalität unseres Teams zeigt, warum das so ist. Rebecca Novak hat uns an die Hand genommen, um eine

überzeugende Gliederung zusammenzustellen, Kristen Brown hat die gesamte inhaltliche Entwicklung souverän gemeistert, Corbin Collins hat uns in jeder Phase hilfreiche Tipps gegeben, die Zusammenarbeit mit Elizabeth Kelly während der Produktion war eine Freude, und Charles Roumeliotis hat das Lektorat mit scharfem Blick begleitet. Danke an alle für diese Hilfe!

Michael: Ich danke meinen Eltern, die immer an mich geglaubt und meine Interessen gefördert haben, nicht nur meine akademischen. Sie werden das Cover, auf dem mein Name steht, genauso zu schätzen wissen wie ich. An Phil: Danke, dass du meinen kaum akzeptablen Zeitplan während der Arbeit an diesem Buch geduldig ertragen hast. Und jetzt werde ich schlafen gehen.

Sara: Jon – du bist ein wichtiger Grund dafür, dass es dieses Buch gibt. Danke, dass du mich ermutigt hast, es zu schreiben, dass du immer gewusst hast, wie du mich zum Lachen bringst, dass du meine Verrücktheit zu schätzen weißt und dass du an mich geglaubt hast, vor allem als ich es nicht tat. Meinen Eltern danke ich, dass sie vom ersten Tag, seit ich mich erinnern kann, meine größten Fans waren und meine Liebe zur Technik und zum Schreiben gefördert haben. An Ally, Katie, Randi und Sophie – danke, dass ihr in diesen unsicheren Zeiten eine ständige Quelle des Lichts und des Lachens seid.

Lak: Ich habe dieses Buch in Angriff genommen in der Annahme, ich könnte daran arbeiten, um Wartezeiten auf Flughäfen sinnvoll zu nutzen. COVID-19 hat es »möglich gemacht«, dass ich einen Großteil der Arbeit zu Hause erledigen konnte. Danke Abirami, Sidharth und Sarada für all eure Nachsicht, als ich mich wieder einmal zum Schreiben hingehockt habe. Mehr Wanderungen an den Wochenenden jetzt!

Wir drei spenden 100 % der Tantiemen aus diesem Buch an »Girls Who Code« (*https://girlswhocode.com/*), eine Organisation, deren Mission es ist, die weltweit größte Pipeline an zukünftigen Ingenieurinnen aufzubauen. Vielfalt, Gleichberechtigung und Inklusion sind beim maschinellen Lernen besonders wichtig, um sicherzustellen, dass KI-Modelle bestehende Vorurteile in der menschlichen Gesellschaft nicht noch untermauern.

KAPITEL 1

Der Bedarf an Entwurfsmustern für maschinelles Lernen

In technischen Disziplinen erfassen Entwurfsmuster Best Practices und Lösungen für häufig auftretende Problemstellungen. Sie kodifizieren das Wissen und die Erfahrung von Expertinnen und Experten in Ratschlägen, die alle Praktiker befolgen können. Dieses Buch ist ein Katalog von Entwurfsmustern, auch Design Patterns genannt, die wir im Laufe unserer Arbeit mit Hunderten von Teams für Machine Learning beobachtet haben.

Was sind Entwurfsmuster?

Christopher Alexander und fünf Mitautoren haben die Idee der Muster (engl. *Patterns*) und einen Katalog bewährter Muster im Bereich der Architektur in einem weltweit anerkannten Buch mit dem Titel *A Pattern Language* (Oxford University Press, 1977) eingeführt. In ihrem Buch stellen sie 253 Muster folgendermaßen vor:

> Jedes Muster beschreibt zum einen ein Problem, das in unserer Umgebung immer wieder auftritt, und zum anderen das Prinzip der

> Problemlösung. Das geschieht so, dass man diese Lösung unzählige Male nutzen kann, ohne sie jemals auf exakt dieselbe Weise anzuwenden.
>
> …
>
> Jede Lösung gibt das wesentliche Beziehungsfeld an, das für die Lösung des Problems erforderlich ist, aber in einer sehr allgemeinen und abstrakten Form. Damit können Sie das Problem in eigener Regie und auf Ihre eigene Art und Weise angehen, indem Sie die Lösung an Ihre Vorstellungen und die vor Ort herrschenden Bedingungen anpassen.

Beispielsweise lauten verschiedene Muster, die persönliche Besonderheiten beim Bau von Wohnungen einbeziehen, *Tageslicht auf zwei Seiten jedes Raums* und *Zwei-Meter-Loggia*. Denken Sie an Ihr Lieblingszimmer in Ihrem Haus und an den Raum, den Sie am wenigsten mögen. Hat Ihr Lieblingsraum zwei Fenster in zwei Wänden? Wie steht es mit Ihrem unbeliebtesten Raum? Nach Alexander:

> In Räumen, in denen von zwei Seiten natürliches Licht einfallen kann, ist die Blendwirkung um Personen und Objekte herum geringer, und vor allem können wir die Mimik in den Gesichtern der Menschen in allen Einzelheiten erkennen …

Ein Name für dieses Muster erspart einem Architekten, dieses Prinzip ständig neu entdecken zu müssen. Doch wo und wie man zwei Lichtquellen in einer bestimmten örtlichen Situation herbekommt, bleibt dem Geschick des Architekten überlassen. Ähnlich verhält es sich beim Entwurf einer Loggia: Wie groß sollte sie sein? Alexander empfiehlt eine Größe von 2 × 2 Metern als ausreichend für zwei (nicht unbedingt zusammenpassende) Stühle und einen Beistelltisch, während die Loggia 4 × 4 Meter groß sein sollte, wenn Sie sowohl einen überdeckten Sitzplatz haben als auch in der Sonne sitzen möchten.

Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides übertrugen die Idee auf Software, indem sie 1994 im Buch *Design Patterns: Elements of Reusable Object-Oriented Software* (Addison-Wesley, 1995) 23

objektorientierte Entwurfsmuster katalogisierten. Die in ihrem Katalog enthaltenen Muster wie Proxy, Singleton und Decorator haben das Gebiet der objektorientierten Programmierung nachhaltig beeinflusst. Im Jahr 2005 verlieh die *Association of Computing Machinery* (ACM) ihren jährlichen Programming Languages Achievement Award an die Autoren und würdigte damit den Einfluss ihrer Arbeit »auf die Programmierpraxis und das Design von Programmiersprachen.«

Modelle für maschinelles Lernen in der Produktion zu erstellen, wird zunehmend zu einer Engineering-Disziplin. Man greift dabei auf bewährte ML-Methoden aus Forschungsumgebungen zurück und wendet sie auf Geschäftsprobleme an. Da maschinelles Lernen immer mehr zum Mainstream wird, sollten Praktiker unbedingt die Vorteile bewährter Methoden nutzen, um damit wiederkehrende Probleme zu lösen.

Unsere Arbeit im kundenorientierten Teil von Google Cloud hat den Vorteil, dass wir mit unterschiedlichsten Teams für maschinelles Lernen und Data Science sowie einzelnen Entwickler:innen aus der ganzen Welt in Kontakt kommen. Gleichzeitig arbeitet jeder von uns eng mit internen Google-Teams zusammen, die hochmoderne Probleme des maschinellen Lernens lösen. Schließlich sind wir in der glücklichen Lage, mit den Teams von TensorFlow, Keras, BigQuery ML, TPU und Cloud AI Platform zusammenzuarbeiten, die die Demokratisierung der Forschung und Infrastruktur für maschinelles Lernen vorantreiben. Dies alles gibt uns eine ziemlich einzigartige Perspektive, von der aus wir die Best Practices katalogisieren können, die wir bei diesen Teams beobachtet haben.

Dieses Buch ist ein Katalog von Entwurfsmustern oder wiederholbaren Lösungen für häufig auftretende Probleme im ML-Engineering. Zum Beispiel erzwingt das Muster *Transformation* (Kapitel 6) die Trennung von Eingaben, Features und Transformationen. Außerdem macht es die Transformationen persistent, um die Überführung eines ML-Modells in die Produktion zu vereinfachen. In ähnlicher Weise ist *Keyed Predictions* in Kapitel 5 ein Muster, das die Verteilung von Batch-Vorhersagen im großen Maßstab ermöglicht, wie zum Beispiel für Empfehlungsmodelle.

Für jedes Muster beschreiben wir das häufig auftretende Problem, das angesprochen wird, gehen dann verschiedenartige mögliche Lösungen für das Problem durch, erläutern Kompromisse dieser Lösungen und geben Empfehlungen für die Auswahl zwischen diesen Lösungen. Der Implementierungscode für diese Lösungen ist angegeben in SQL (was sinnvoll ist, wenn Sie Vorverarbeitungen und andere ETL[1]-Operationen in Spark SQL, BigQuery usw. ausführen), scikit-learn und/oder Keras mit einem TensorFlow-Backend.

Wie Sie dieses Buch verwenden

Vor Ihnen liegt ein Katalog von Entwurfsmustern, die wir in der Praxis beobachtet haben, und zwar bei mehreren Teams. In einigen Fällen sind die zugrunde liegenden Konzepte schon seit vielen Jahren bekannt. Wir erheben nicht den Anspruch, diese Muster erfunden oder entdeckt zu haben. Vielmehr hoffen wir, einen gemeinsamen Bezugsrahmen und einen Satz von Werkzeugen für ML-Praktiker:innen bereitzustellen. Das ist uns dann gelungen, wenn dieses Buch Ihnen und Ihrem Team ein Vokabular an

die Hand gibt, um über Konzepte zu sprechen, die Sie in Ihren ML-Projekten bereits intuitiv umgesetzt haben.

Wir gehen nicht davon aus, dass Sie dieses Buch der Reihe nach durchlesen (obwohl nichts dagegenspricht!). Stattdessen nehmen wir an, dass Sie das Buch überfliegen, einige Abschnitte eingehender als andere lesen, die Ideen in Gesprächen mit Kolleginnen und Kollegen erwähnen und auf das Buch zurückgreifen, wenn Sie mit Problemen konfrontiert werden, von denen Sie hier bereits gelesen haben. Falls Sie so vorgehen möchten, empfehlen wir, mit Kapitel 1 und Kapitel 8 zu beginnen, bevor Sie sich einzelnen Mustern zuwenden.

Zu jedem Muster gehört eine kurze Problemaussage, eine kanonische Lösung und eine Erklärung dazu, warum die Lösung funktioniert, sowie eine mehrteilige Diskussion über Kompromisse und Alternativen. Wir empfehlen, den Diskussionsabschnitt zu lesen und dabei die kanonische Lösung fest im Hinterkopf zu behalten, um zu vergleichen und gegenüberzustellen. Die Musterbeschreibung enthält Codefragmente aus der Implementierung der kanonischen Lösung. Den vollständigen Code finden Sie in unserem GitHub-Repository (*https://github.com/GoogleCloudPlatform/ml-design-patterns*). Es empfiehlt sich, den Code durchzugehen, während Sie die Musterbeschreibung lesen.

Terminologie für maschinelles Lernen

Da Praktikerinnen und Praktiker im Bereich des maschinellen Lernens heutzutage aus den unterschiedlichsten Fachgebieten – Softwaretechnik,

Datenanalyse, DevOps oder Statistik – stammen können, gibt es subtile Unterschiede in der Verwendung bestimmter Begriffe. In diesem Abschnitt definieren wir die Terminologie, die wir im gesamten Buch verwenden.

Modelle und Frameworks

In seinem Kern ist *maschinelles Lernen* ein Prozess, der Modelle erstellt, die aus Daten lernen. Dies steht im Gegensatz zur herkömmlichen Programmierung, bei der wir explizite Regeln schreiben, die den Programmen sagen, wie sie sich verhalten sollen. *Modelle für maschinelles Lernen* sind Algorithmen, die Muster aus Daten lernen. Diesen Punkt wollen wir anhand einer Firma veranschaulichen, die Umzugskosten für potenzielle Kunden abschätzen muss. In der herkömmlichen Programmierung könnten wir dies mit einer `if`-Anweisung lösen:

```
if num_bedrooms == 2 and num_bathrooms == 2:

estimate = 1500

elif num_bedrooms == 3 and sq_ft > 2000:

estimate = 2500
```

Man kann sich vorstellen, wie schnell dies kompliziert wird, wenn wir weitere Variablen (Anzahl großer Möbelstücke, Umfang der Kleidung, zerbrechliche Gegenstände usw.) hinzufügen und versuchen, Sonderfälle zu behandeln. Und wenn man all diese Informationen im Voraus von den Kunden abfragt, kann das vor allem dazu führen, dass die Firma den Schätzprozess aufgibt. Stattdessen können wir ein maschinelles Lernmodell trainieren, um die

Umzugskosten basierend auf den Daten früherer Umzüge unseres Unternehmens zu schätzen.

In den Beispielen, die das Buch vorstellt, verwenden wir hauptsächlich neuronale Feedforward-Netze, doch ziehen wir auch Modelle der linearen Regression, Entscheidungsbäume, Clustering-Modelle und andere heran. *Neuronale Feedforward-Netze*, die wir üblicherweise kurz als *neuronale Netze* bezeichnen, stellen einen Algorithmentyp für maschinelles Lernen dar, bei dem mehrere Schichten (engl. *Layers*) mit jeweils vielen Neuronen Informationen analysieren und verarbeiten und dann diese Informationen an die nächste Schicht senden, wobei schließlich die letzte Schicht eine Vorhersage als Ausgabe produziert. Obwohl sie keineswegs identisch sind, werden neuronale Netze oft mit den Neuronen in unserem Gehirn verglichen, und zwar aufgrund der Konnektivität zwischen den Knoten und der Art und Weise, wie sie verallgemeinern und neue Vorhersagen aus den verarbeiteten Daten bilden können. Neuronale Netze mit mehr als einem *Hidden Layer* (einer versteckten Schicht, d. h. einer Schicht, die weder Eingabe- noch Ausgabeschicht ist) werden als *Deep Learning* klassifiziert (siehe Abbildung 1-1).

Modelle für maschinelles Lernen sind – unabhängig davon, wie man sie visuell darstellt – mathematische Funktionen und lassen sich demzufolge mit einem numerischen Softwarepaket von Grund auf neu erstellen. Allerdings greifen ML Engineers in der Industrie gern zu einem von mehreren Open-Source-Frameworks, die konzeptionell intuitive APIs für das Erstellen von Modellen anbieten. Die Mehrheit unserer Beispiele verwendet *TensorFlow*, ein Open-Source-Framework für maschinelles Lernen, das von Google mit Schwerpunkt auf Deep-Learning-Modelle

geschaffen wurde. Innerhalb der TensorFlow-Bibliothek verwenden wir für unsere Beispiele die *Keras*-API, die sich über `tensorflow.keras` importieren lässt. Bei Keras handelt es sich um eine Higher-Level-API zum Erstellen von neuronalen Netzen. Von den verschiedenen Backends, die Keras unterstützt, haben wir uns für Tensor-Flow entschieden. Andere Beispiele arbeiten mit den ebenfalls beliebten Open-Source-Frameworks *scikit-learn, XGBoost* und *PyTorch*, die neben APIs für das Erstellen von linearen und tiefen Modellen auch Hilfsprogramme enthalten, mit denen Sie Ihre Daten vorbereiten können. Maschinelles Lernen wird immer zugänglicher, und eine spannende Entwicklung ist die Verfügbarkeit von Modellen für maschinelles Lernen, die sich in SQL ausdrücken lassen. Als Beispiel hierfür setzen wir *BigQuery ML* ein, insbesondere in Situationen, in denen wir Datenvorverarbeitung und Modellerstellung kombinieren möchten.

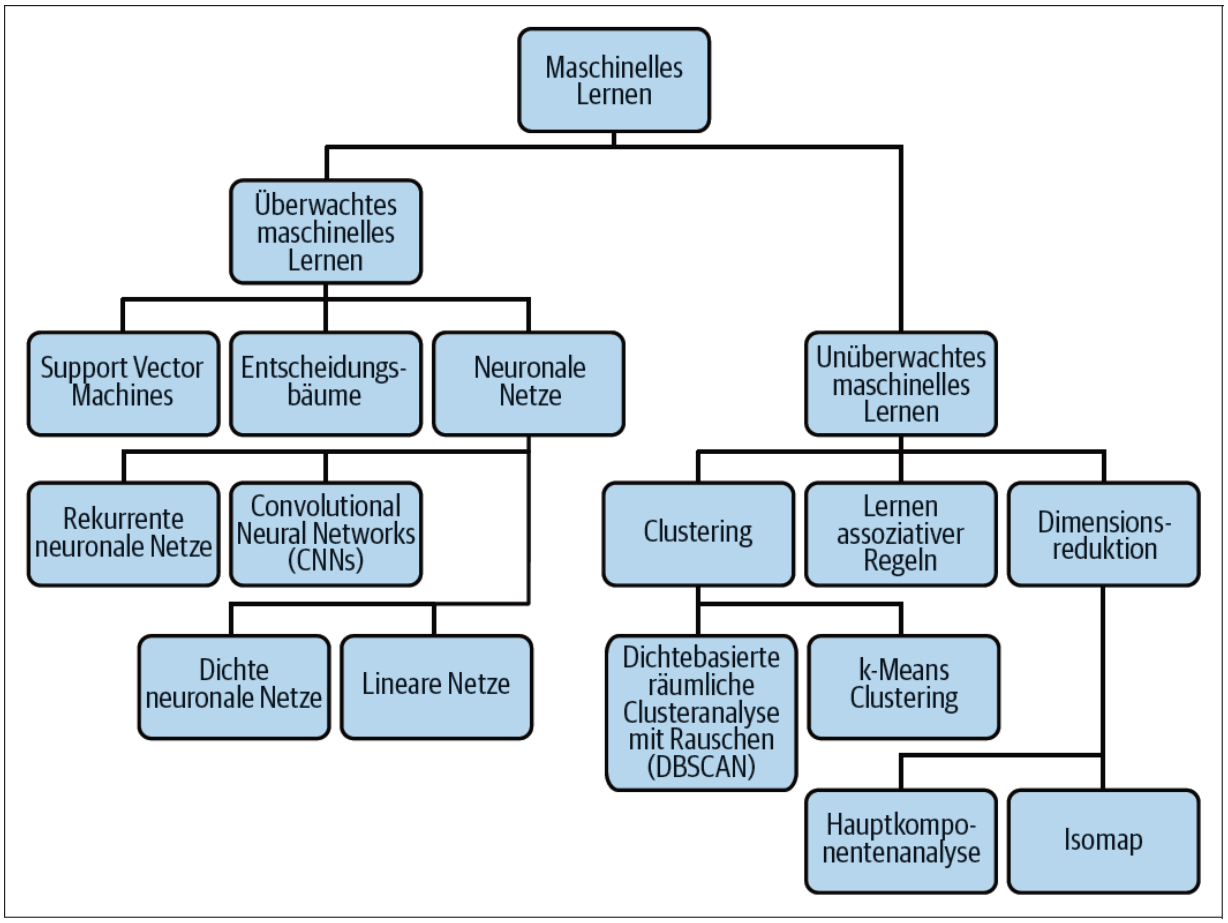


Abbildung 1-1: Eine Aufschlüsselung der verschiedenen Arten von maschinellem Lernen mit jeweils einigen Beispielen. Obwohl sie nicht in der Darstellung enthalten sind, können auch neuronale Netze wie Autoencoder für unüberwachtes Lernen eingesetzt werden.

Umgekehrt bilden neuronale Netze mit nur einer Eingabe- und einer Ausgabeschicht eine andere Teilmenge des maschinellen Lernens, die sogenannten *linearen Modelle*. Diese stellen die aus den Daten gelernten Muster mithilfe einer linearen Funktion dar. *Entscheidungsbäume* sind Modelle des maschinellen Lernens, die Ihre Daten verwenden, um eine Teilmenge von Pfaden mit verschiedenen Verzweigungen zu erzeugen. Diese Verzweigungen stellen eine Annäherung an die Ergebnisse verschiedener Ausgaben aus Ihren Daten dar. Schließlich suchen *Clustering*-Modelle nach Ähnlichkeiten zwischen