Praxiseinstieg Machine Learning mit Scikit-Learn & TensorFlow

Konzepte, Tools und Techniken für intelligente Systeme

DAS VORWORT

Vorwort

Der Machine-Learning-Tsunami

Im Jahr 2006 erschien ein Artikel (*https://homl.info/136*) von Geoffrey Hinton et al.[1], in dem vorgestellt wurde, wie sich ein neuronales Netz zum Erkennen handgeschriebener Ziffern mit ausgezeichneter Genauigkeit (> 98 %) trainieren lässt. Ein Deep Neural Network ist ein (sehr) vereinfachtes Modell unseres zerebralen Kortex, und es besteht aus einer Folge von Schichten mit künstlichen Neuronen. Die Autoren nannten diese Technik »Deep Learning«. Zu dieser Zeit wurde das Trainieren eines Deep-Learning-Netzes im Allgemeinen als unmöglich angesehen,[2] und die meisten Forscher hatten die Idee in den 1990ern aufgegeben. Dieser Artikel ließ das Interesse der wissenschaftlichen Gemeinde wieder aufleben, und schon nach kurzer Zeit zeigten weitere Artikel, dass Deep Learning nicht nur möglich war, sondern umwerfende Dinge vollbringen konnte, zu denen kein anderes Machine-Learning-(ML-)Verfahren auch nur annähernd in der Lage war (mithilfe enormer Rechenleistung und riesiger Datenmengen). Dieser Enthusiasmus breitete sich schnell auf weitere Teilgebiete des Machine Learning aus.

Zehn Jahre später hat Machine Learning ganze Industriezweige erobert: Es ist zu einem Herzstück heutiger Spitzentechnologien geworden und dient dem Ranking von Suchergebnissen im Web, kümmert sich um die Spracherkennung Ihres Smartphones, gibt Empfehlungen für Videos und steuert vielleicht sogar Ihr Auto.

1 Geoffrey Hinton et al., »A Fast Learning Algorithm for Deep Belief Nets«, *Neural Computation* 18 (2006): 1527–1554.

2 Obwohl die Konvolutionsnetze von Yann Lecun bei der Bilderkennung seit den 1990ern gut funktioniert hatten, auch wenn sie nicht allgemein anwendbar waren.

Machine Learning in Ihren Projekten

Deshalb interessieren Sie sich natürlich auch für Machine Learning und möchten an der Party teilnehmen!

Womöglich möchten Sie Ihrem selbst gebauten Roboter einen eigenen Denkapparat geben? Ihn Gesichter erkennen lassen? Oder ihn lernen lassen, herumzulaufen?

Oder vielleicht besitzt Ihr Unternehmen Unmengen an Daten (Logdateien, Finanzdaten, Produktionsdaten, Sensordaten, Hotline-Statistiken, Personalstatistiken und so weiter), und Sie könnten vermutlich einige verborgene Schätze heben, wenn Sie nur wüssten, wo Sie danach suchen müssten. Mit Machine Learning können Sie das Folgende (und noch viel mehr (*https://homl.info/usecases*)) erreichen:

- Kundensegmente finden und für jede Gruppe die beste Marketingstrategie entwickeln.
- Jedem Kunden anhand des Kaufverhaltens ähnlicher Kunden Produktempfehlungen geben.
- Betrügerische Transaktionen mit hoher Wahrscheinlichkeit erkennen.
- Den Unternehmensgewinn im nächsten Jahr vorhersagen.

Was immer der Grund ist, Sie haben beschlossen, Machine Learning zu erlernen und in Ihren Projekten umzusetzen. Eine ausgezeichnete Idee!

Ziel und Ansatz

Dieses Buch geht davon aus, dass Sie noch so gut wie nichts über Machine Learning wissen. Unser Ziel ist es, Ihnen die Grundbegriffe, ein Grundverständnis und die Werkzeuge an die Hand zu geben, mit denen Sie Programme zum *Lernen aus Daten* entwickeln können.

Wir werden eine Vielzahl von Techniken besprechen, von den einfachsten und am häufigsten eingesetzten (wie der linearen Regression) bis zu einigen Deep-Learning-Verfahren, die regelmäßig Wettbewerbe gewinnen. Wir werden dazu für den Produktionsbetrieb geschaffene Python-Frameworks verwenden:

- Scikit-Learn (*http://scikit-learn.org/*) ist sehr einfach zu verwenden, enthält aber effiziente Implementierungen vieler Machine-Learning-Algorithmen. Damit ist es ein großartiger Ausgangspunkt, um Machine Learning zu erlernen. Scikit-Learn wurde von David Cournapeau im Jahr 2007 erstellt und wird mittlerweile von einem Forschungsteam am Nationalen Forschungsinstitut für Informatik und Automatisierung (INRIA) in Frankreich betreut.
- TensorFlow (*http://tensorflow.org/*) ist eine komplexere Bibliothek für verteiltes Rechnen. Mit ihr können Sie sehr große neuronale Netze effizient trainieren und ausführen, indem Sie die Berechnungen auf bis zu Hunderte von Servern mit mehreren GPUs (*Graphics Processing Units*) verlagern. TensorFlow

(TF) wurde von Google entwickelt und läuft in vielen umfangreichen Machine-Learning-Anwendungen. Die Bibliothek wurde im November 2015 als Open Source veröffentlicht, im September 2019 erschien Version 2.0.

- Keras (*https://keras.io/*) ist eine High-Level-Deep-Learning-API, die das Trainieren und Ausführen neuronaler Netze sehr einfach macht. Sie kommt zusammen mit TensorFlow und greift für alle rechenintensiven Aufgaben darauf zurück.

Dieses Buch verfolgt einen praxisorientierten Ansatz, bei dem Sie ein intuitives Verständnis von Machine Learning entwickeln, indem Sie sich mit konkreten Beispielen und ein klein wenig Theorie beschäftigen.

Auch wenn Sie dieses Buch lesen können, ohne Ihren Laptop in die Hand zu nehmen, empfehlen wir Ihnen, mit den Codebeispielen herumzuexperimentieren.

Codebeispiele

Alle Codebeispiele in diesem Buch sind Open Source und stehen online unter *https://github.com/ageron/handson-mls3* als Jupyter Notebooks zur Verfügung. Dabei handelt es sich um interaktive Dokumente mit Texten, Bildern und ausführbaren Codeabschnitten (in unserem Fall Python). Am einfachsten beginnen Sie damit, diese Notebooks mit Google Colab auszuführen. Das ist ein kostenloser Service, der es Ihnen ermöglicht, beliebige Jupyter Notebooks direkt online laufen zu lassen, ohne etwas auf Ihrem Rechner installieren zu müssen. Sie brauchen nur einen Webbrowser und einen Google-Account.

In diesem Buch gehe ich davon aus, dass Sie Google Colab nutzen, aber ich habe die Notebooks auch auf anderen Onlineplattformen wie Kaggle oder Binder getestet, sodass Sie sie dort ebenfalls einsetzen können, wenn Ihnen das lieber ist. Alternativ können Sie die erforderlichen Bibliotheken und Tools (oder das Docker-Image für dieses Buch) direkt auf Ihrem Computer installieren und die Notebooks dort laufen lassen. Die Anweisungen dazu finden Sie unter *https://homl.info/install*.

Verwenden von Codebeispielen

Dieses Buch ist dazu da, Ihnen beim Erledigen Ihrer Arbeit zu helfen. Wollen Sie über die Codebeispiele hinaus zusätzliche Inhalte in einem Umfang verwenden, der die Regeln eines fairen Einsatzes verletzen würde (zum Beispiel der Verkauf oder Vertrieb von Inhalten aus O'Reilly-Büchern oder das Einbinden eines beträchtlichen Teils dieses Buchs in die Dokumentation Ihres Produkts), wenden Sie sich bitte für eine Erlaubnis an *kommentar@oreilly.de*.

Wir freuen uns über Zitate, verlangen diese aber nicht. Ein Zitat enthält Titel, Autor, Verlag und ISBN. Beispiel: »*Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow* von Aurélien Géron, O'Reilly 2023, ISBN 978-3-96009-212-4.«

Voraussetzungen

Dieses Buch geht davon aus, dass Sie ein wenig Programmiererfahrung mit Python haben. Sollten Sie Python noch nicht kennen, ist *http://learnpython.org/* ein ausgezeichneter Ausgangspunkt. Auch das offizielle Tutorial auf python.org (*https://docs.python.org/3/tutorial/*) ist recht gut.

Es setzt ebenfalls voraus, dass Sie mit den wichtigsten wissenschaftlichen Bibliotheken in Python vertraut sind, insbesondere mit NumPy (*http://numpy.org/*), Pandas (*http://pandas.pydata.org/*) und Matplotlib (*http://matplotlib.org/*). Falls Sie diese Bibliotheken noch nie verwendet haben, ist das nicht schlimm – ihr Einsatz lässt sich leicht erlernen, und ich habe für jede von ihnen ein Tutorial erstellt. Diese finden Sie online unter *https://homl.info/tutorials*.

Wenn Sie vollständig verstehen wollen, wie die Algorithmen zum Machine Learning funktionieren (und nicht nur, wie man sie einsetzt), sollten Sie ein Grundverständnis von einigen mathematischen Konzepten haben – insbesondere von linearer Algebra. Sie sollten wissen, was Vektoren und Matrizen sind und wie Sie einige einfache Operationen mit ihnen ausführen, zum Beispiel das Addieren von Vektoren oder das Transponieren und Multiplizieren von Matrizen. Sollten Sie eine kurze Einführung in die lineare Algebra benötigen (sie ist wirklich kein Hexenwerk), habe ich unter *https://homl.info/tutorials* ein Tutorial bereitgestellt. Sie finden dort ebenfalls eines zur Differenzialrechnung, was hilfreich sein kann, um zu verstehen, wie neuronale Netze trainiert werden, aber für das grobe Verstehen der wichtigsten Konzepte ist es nicht unbedingt von entscheidender Bedeutung. Dieses Buch greift gelegentlich auch auf andere mathematische Konzepte zurück, beispielsweise auf Exponentialfunktionen und Logarithmen, ein bisschen Wahrscheinlichkeitstheorie und ein paar Statistikaspekte, aber alles nicht zu ausgefallen. Brauchen Sie dazu Hilfe, schauen Sie mal bei *https://khanacademy.org* vorbei, wo Sie online viele ausgezeichnete und kostenlose Mathematikkurse finden.

Wegweiser durch dieses Buch

Dieses Buch ist in zwei Teile aufgeteilt. Teil I behandelt folgende Themen:

- Was ist Machine Learning? Welche Aufgaben lassen sich damit lösen? Was sind die wichtigsten Kategorien und Grundbegriffe von Machine-Learning-Systemen?
- Die Schritte in einem typischen Machine-Learning-Projekt.

- Lernen durch Anpassen eines Modells an Daten.
- Optimieren einer Kostenfunktion.
- Bearbeiten, Säubern und Vorbereiten von Daten.
- Merkmale auswählen und entwickeln.
- Ein Modell auswählen und dessen Hyperparameter über Kreuzvalidierung optimieren.
- Die Herausforderungen beim Machine Learning, insbesondere Underfitting und Overfitting (das Gleichgewicht zwischen Bias und Varianz).
- Die verbreitetsten Lernalgorithmen: lineare und polynomielle Regression, logistische Regression, *k*-nächste Nachbarn, Support Vector Machines, Entscheidungsbäume, Random Forests und Ensemble-Methoden.
- Dimensionsreduktion der Trainingsdaten, um dem »Fluch der Dimensionalität« etwas entgegenzusetzen.
- Andere Techniken des unüberwachten Lernens, unter anderem Clustering, Dichteabschätzung und Anomalieerkennung.

Teil II widmet sich diesen Themen:

- Was sind neuronale Netze? Wofür sind sie geeignet?
- Erstellen und Trainieren neuronaler Netze mit TensorFlow und Keras.
- Die wichtigsten Architekturen neuronaler Netze: Feed-Forward-Netze für Tabellendaten, Convolutional Neural Networks zur Bilderkennung, rekurrente Netze und Long-Short-Term-Memory-(LSTM-)Netze zur Sequenzverarbeitung, Encoder/Decoder und Transformer für die Sprachverarbeitung (und mehr!), Autoencoder sowie Generative Adversarial Networks (GANs) und Diffusionsmodelle zum generativen Lernen.
- Techniken zum Trainieren von Deep-Learning-Netzen.
- Wie man einen Agenten erstellt (zum Beispiel einen Bot in einem Spiel), der durch Versuch und Irrtum gute Strategien erlernt und dabei Reinforcement Learning einsetzt.
- Effizientes Laden und Vorverarbeiten großer Datenmengen.
- Trainieren und Deployen von TensorFlow-Modellen im großen Maßstab.

Der erste Teil baut vor allem auf Scikit-Learn auf, der zweite Teil verwendet TensorFlow.

Springen Sie nicht zu schnell ins tiefe Wasser: Auch wenn Deep Learning zweifelsohne eines der aufregendsten Teilgebiete des Machine Learning ist, sollten Sie zuerst Erfahrungen mit den Grundlagen sammeln. Außerdem lassen sich die meisten Aufgabenstellungen recht gut mit einfacheren Techniken wie Random Forests und Ensemble-Methoden lösen (die in Teil I besprochen werden). Deep Learning ist am besten für komplexe Aufgaben

wie Bilderkennung, Spracherkennung und Sprachverarbeitung geeignet. Sie müssen aber genug Daten, Rechenleistung und Geduld haben (sofern Sie nicht ein vortrainiertes neuronales Netz nutzen können, wie Sie noch sehen werden).

Änderungen zwischen der ersten und der zweiten Auflage

Haben Sie schon die erste Auflage gelesen, finden Sie hier die wichtigsten Unterschiede zwischen der ersten und zweiten Auflage:

- Der gesamte Code wurde von TensorFlow 1.x auf TensorFlow 2.x gehoben, und ich habe einen Großteil des Low-Level-Codes mit TensorFlow (Graphen, Sessions, Feature Columns, Estimators und so weiter) durch viel einfacheren Keras-Code ersetzt.
- In der zweiten Auflage wurde die Data-API zum Laden und Vorverarbeiten großer Datasets aufgenommen, die Distribution Strategies API zum Trainieren und Deployen von TF-Modellen im großen Maßstab, TF Serving und die Google Cloud API Platform, mit denen Modelle in Produkte umgewandelt werden können, sowie (zumindest angerissen) TF Transform, TFLite, TF Addons/Seq2Seq, TensorFlow.js und TF Agents.
- Es wurden auch viele zusätzliche ML-Themen hinzugenommen, unter anderem ein neues Kapitel zu unüberwachtem Lernen, Computer-Vision-Techniken zur Objekterkennung und zur semantischen Segmentierung, außerdem das Verarbeiten von Sequenzen durch Convolutional Neural Networks (CNNs), die Verarbeitung natürlicher Sprache (Natural Language Processing, NLP) mit rekurrenten neuronalen Netzen (RNNs), CNNs und Transformer, GANs und vieles mehr.

Auf *https://homl.info/changes2* finden Sie mehr Details.

Änderungen zwischen der zweiten und der dritten Auflage

Haben Sie die zweite Auflage gelesen, finden Sie hier die wichtigsten Änderungen zwischen zweiter und dritter Auflage:

- Der gesamte Code wurde an die neuesten Bibliotheksversionen angepasst. Insbesondere wurden in dieser dritten Auflage viele neue Ergänzungen von Scikit-Learn berücksichtigt (zum Beispiel das Feature Name Tracking, histogrammbasiertes Gradient Boosting, Label Propagation und mehr). Zudem wurden die Bibliothek *Keras Tuner* für das Tuning von Hyperparametern, die

Transformers-Bibliothek von Hugging Face für die Verarbeitung natürlicher Sprache sowie die neuen Vorverarbeitungsschichten und Data Augmentation Layer mit aufgenommen.

- Es wurden diverse Vision-Modelle hinzugefügt (ResNeXt, DenseNet, MobileNet, CSPNet und EfficientNet) und dazu Entscheidungshilfen für das Wählen des richtigen Modells.
- In Kapitel 15 werden nun statt generierter Zeitserien die *Chicago Bus and Rail Ridership*-Daten analysiert, und das ARMA-Modell mit seinen Varianten wird vorgestellt.
- Kapitel 16 zur Verarbeitung natürlicher Sprache baut nun ein Modell zum Übersetzen aus dem Englischen ins Spanische, wobei zuerst ein Encoder-Decoder-RNN und dann ein Transformer-Modell zum Einsatz kommen. Das Kapitel behandelt zudem Sprachmodelle wie Switch Transformers, DistilBERT, T5 und PaLM (mit einem Chain-of-Thought Prompting), außerdem stellt es Vision Transformers (ViTs) vor und gibt einen Überblick über ein paar auf Transformern basierende visuelle Modelle, wie zum Beispiel Data-Efficient Image Transformers (DeiTs), Perceivers und DINO. Zudem gibt es einen kurzen Hinweis auf ein paar große, multimodale Modelle – unter anderem CLIP, DALL·E, Flamingo und GATO.
- In Kapitel 17 zu generativem Lernen werden nun Diffusionsmodelle vorgestellt, und es wird gezeigt, wie Sie ein Denoising Diffusion Probabilistic Model (DDPM) von Grund auf implementieren.
- Kapitel 19 ist von der Google Cloud AI Platform nach Google Vertex AI umgezogen und nutzt jetzt verteilte Keras Tuner für die Hyperparametersuche im großen Maßstab. Es führt nun TensorFlow.js-Code ein, mit dem Sie online experimentieren können. Zudem werden zusätzliche verteilte Trainingstechniken vorgestellt, unter anderem PipeDream und Pathways.
- Um all die neuen Inhalte unterbringen zu können, stehen manche Abschnitte nur noch online zur Verfügung, unter anderem die Installationsanweisungen, die Hauptkomponentenzerlegung (PCA), mathematische Details zu bayesschen gaußschen Mischverteilungsmodellen, TF Agents und die früheren Anhänge A (Lösungen zu den Übungsaufgaben), C (die Mathematik von Support Vector Machines) und E (weitere Architekturen für neuronale Netze). Diese drei früheren Anhänge finden Sie auf der deutschen Website zum Buch unter *https://dpunkt.de/produkt/praxiseinstieg-machine-learning-mit-scikit-learn-keras-und-tensorflow-2/* auf der Registerkarte *Zusatzmaterial*.

Auf *https://homl.info/changes3* finden Sie zusätzliche Informationen zu den Änderungen.

Ressourcen im Netz

Es gibt viele ausgezeichnete Ressourcen, mit deren Hilfe sich Machine Learning erlernen lässt. Der ML-Kurs auf Coursera (*https://homl.info/ngcourse*) von Andrew Ng ist faszinierend, auch wenn er einen beträchtlichen Zeitaufwand bedeutet.

Darüber hinaus finden Sie viele interessante Webseiten über Machine Learning, darunter natürlich den ausgezeichneten User Guide (*https://homl.info/skdoc*) von Scikit-Learn. Auch Dataquest (*https://www.dataquest.io/*), das sehr ansprechende Tutorials und ML-Blogs bietet, sowie die auf Quora (*https://homl.info/1*) aufgeführten ML-Blogs könnten Ihnen gefallen.

Natürlich bieten darüber hinaus viele andere Bücher eine Einführung in Machine Learning, insbesondere:

- Joel Grus, *Einführung in Data Science: Grundprinzipien der Datenanalyse mit Python (https://oreilly.de/produkt/einfuehrung-in-data-science-2/)* (O'Reilly, 2. Auflage). Dieses Buch stellt die Grundlagen von Machine Learning vor und implementiert die wichtigsten Algorithmen in reinem Python (von Grund auf).
- Stephen Marsland, *Machine Learning: An Algorithmic Perspective, 2nd Edition* (Chapman & Hall). Dieses Buch ist eine großartige Einführung in Machine Learning, die viele Themen ausführlich behandelt. Es enthält Codebeispiele in Python (ebenfalls von Grund auf, aber mit NumPy).
- Sebastian Raschka, *Machine Learning mit Python und Keras, TensorFlow 2 und Scikit-learn: Das umfassende Praxis-Handbuch für Data Science, Deep Learning und Predictive Analytics* (mitp Professional, 3. Auflage). Eine weitere ausgezeichnete Einführung in Machine Learning. Dieses Buch konzentriert sich auf Open-Source-Bibliotheken in Python (Pylearn 2 und Theano).
- François Chollet, *Deep Learning with Python* (Manning, 2nd Edition). Ein sehr praxisnahes Buch, das klar und präzise viele Themen behandelt – wie Sie es vom Autor der ausgezeichneten Keras-Bibliothek erwarten können. Es zieht Codebeispiele der mathematischen Theorie vor.
- Andriy Burkov, *Machine Learning kompakt: Alles, was Sie wissen müssen* (mitp Professional). Dieses sehr kurze Buch behandelt ein beeindruckendes Themenspektrum gut verständlich, scheut dabei aber nicht vor mathematischen Gleichungen zurück.
- Yaser S. Abu-Mostafa, Malik Magdon-Ismail und Hsuan-Tien Lin, *Learning from Data* (AMLBook). Als eher theoretische Abhandlung von ML enthält dieses Buch sehr tiefgehende Erkenntnisse, insbesondere zum Gleichgewicht zwischen Bias und Varianz (siehe Kapitel 4).
- Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach, 4th Edition* (Pearson). Dieses ausgezeichnete (und umfangreiche) Buch deckt

eine unglaubliche Stoffmenge ab, darunter Machine Learning. Es hilft dabei, ML in einem breiteren Kontext zu betrachten.

- Jeremy Howard and Sylvain Gugger, *Deep Learning for Coders with fastai and PyTorch* (O'Reilly). Eine wunderbar klare und praxisnahe Einführung in Deep Learning mithilfe der fastai- und PyTorch-Bibliotheken.

Eine gute Möglichkeit zum Lernen sind schließlich Webseiten mit ML-Wettbewerben wie Kaggle.com (*https://kaggle.com/*). Dort können Sie Ihre Fähigkeiten an echten Aufgaben üben und Hilfe und Tipps von einigen der besten ML-Profis erhalten.

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, Umgebungsvariablen, 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 Tipp oder eine Empfehlung.

Dieses Symbol steht für einen allgemeinen Hinweis.

Dieses Symbol steht für eine Warnung oder erhöhte Aufmerksamkeit.