Preface

The Machine Learning Tsunami

In 2006, Geoffrey Hinton et al. published a paper [1] showing how to train a deep neural network capable of recognizing handwritten digits with state-of-the-art precision (>98%). They branded this technique “deep learning”. A deep neural network is a (very) simplified model of our cerebral cortex, composed of a stack of layers of artificial neurons. Training a deep neural net was widely considered impossible at the time, [2] and most researchers had abandoned the idea in the late 1990s. This paper revived the interest of the scientific community, and before long many new papers demonstrated that deep learning was not only possible, but capable of mind-blowing achievements that no other machine learning (ML) technique could hope to match (with the help of tremendous computing power and great amounts of data). This enthusiasm soon extended to many other areas of machine learning.

A decade later, machine learning had conquered the industry, and today it is at the heart of much of the magic in high-tech products, ranking your web search results, powering your smartphone’s speech recognition, recommending videos, and perhaps even driving your car.

Machine Learning in Your Projects

So, naturally you are excited about machine learning and would love to join the party!

Perhaps you would like to give your homemade robot a brain of its own? Make it recognize faces? Or learn to walk around?

Or maybe your company has tons of data (user logs, financial data, production data, machine sensor data, hotline stats, HR reports, etc.), and more than likely you could unearth some hidden gems if you just knew where to look. With machine learning, you could accomplish the following and much more:

- Segment customers and find the best marketing strategy for each group.
- Recommend products for each client based on what similar clients bought.
- Detect which transactions are likely to be fraudulent.
- Forecast next year’s revenue.

Whatever the reason, you have decided to learn machine learning and implement it in your projects. Great idea!

Objective and Approach

This book assumes that you know close to nothing about machine learning. Its goal is to give you the concepts, tools, and intuition you need to implement programs capable of *learning from data*.

We will cover a large number of techniques, from the simplest and most commonly used (such as linear regression) to some of the deep learning techniques that regularly win competitions. For this, we will be using production-ready Python frameworks:

- Scikit-Learn is very easy to use, yet it implements many machine learning algorithms efficiently, so it makes for a great entry point to learning machine learning. It was created by David Cournapeau in 2007, and is now led by a team of researchers at the French Institute for Research in Computer Science and Automation (Inria).
- TensorFlow is a more complex library for distributed numerical computation. It makes it possible to train and run very large neural networks efficiently by distributing the computations across potentially hundreds of multi-GPU (graphics processing unit) servers. TensorFlow (TF) was created at Google and supports many of its large-scale machine learning applications. It was open sourced in November 2015, and version 2.0 was released in September 2019.
- Keras is a high-level deep learning API that makes it very simple to train and run neural networks. Keras comes bundled with TensorFlow, and it relies on TensorFlow for all the intensive computations.

The book favors a hands-on approach, growing an intuitive understanding of machine learning through concrete working examples and just a little bit of theory.

TIP

While you can read this book without picking up your laptop, I highly recommend you experiment with the code examples.

Code Examples

All the code examples in this book are open source and available online at *https://github.com/ageron/handson-ml3*, as Jupyter notebooks. These are interactive documents containing text, images, and executable code snippets (Python in our case). The easiest and quickest way to get started is to run these notebooks using Google Colab: this is a free service that allows you to run any Jupyter notebook directly online, without having to install anything on your machine. All you need is a web browser and a Google account.

> **NOTE**
>
> In this book, I will assume that you are using Google Colab, but I have also tested the notebooks on other online platforms such as Kaggle and Binder, so you can use those if you prefer. Alternatively, you can install the required libraries and tools (or the Docker image for this book) and run the notebooks directly on your own machine. See the instructions at *https://homl.info/install*.

Prerequisites

This book assumes that you have some Python programming experience. If you don't know Python yet, *https://learnpython.org* is a great place to start. The official tutorial on Python.org is also quite good.

This book also assumes that you are familiar with Python's main scientific libraries—in particular, NumPy, Pandas, and Matplotlib. If you have never used these libraries, don't worry; they're easy to learn, and I've created a tutorial for each of them. You can access them online at *https://homl.info/tutorials*.

Moreover, if you want to fully understand how the machine learning algorithms work (not just how to use them), then you should have at least a basic understanding of a few math concepts, especially linear algebra. Specifically, you should know what vectors and matrices are, and how to perform some simple operations like adding vectors, or transposing and multiplying matrices. If you need a quick introduction to linear algebra (it's really not rocket science!), I provide a tutorial at *https://homl.info/tutorials*. You will also find a tutorial on differential calculus, which may be helpful to understand how neural networks are trained, but it's not entirely essential to grasp the important concepts. This book also uses other mathematical concepts occasionally, such as exponentials and logarithms, a bit of probability theory, and some basic statistics concepts, but nothing too advanced. If you need help on any of these, please check out *https://khanacademy.org*, which offers many excellent and free math courses online.

Roadmap

This book is organized in two parts. Part I, "The Fundamentals of Machine Learning", covers the following topics:

- What machine learning is, what problems it tries to solve, and the main categories and fundamental concepts of its systems
- The steps in a typical machine learning project
- Learning by fitting a model to data
- Optimizing a cost function
- Handling, cleaning, and preparing data
- Selecting and engineering features
- Selecting a model and tuning hyperparameters using cross-validation
- The challenges of machine learning, in particular underfitting and overfitting (the bias/variance trade-off)
- The most common learning algorithms: linear and polynomial regression, logistic regression, *k*-nearest neighbors, support vector machines, decision trees, random forests, and ensemble methods
- Reducing the dimensionality of the training data to fight the "curse of dimensionality"
- Other unsupervised learning techniques, including clustering, density estimation, and anomaly detection

Part II, "Neural Networks and Deep Learning", covers the following topics:

- What neural nets are and what they're good for
- Building and training neural nets using TensorFlow and Keras
- The most important neural net architectures: feedforward neural nets for

tabular data, convolutional nets for computer vision, recurrent nets and long short-term memory (LSTM) nets for sequence processing, encoder–decoders and transformers for natural language processing (and more!), autoencoders, generative adversarial networks (GANs), and diffusion models for generative learning

- Techniques for training deep neural nets
- How to build an agent (e.g., a bot in a game) that can learn good strategies through trial and error, using reinforcement learning
- Loading and preprocessing large amounts of data efficiently
- Training and deploying TensorFlow models at scale

The first part is based mostly on Scikit-Learn, while the second part uses TensorFlow and Keras.

> **CAUTION**
>
> Don’t jump into deep waters too hastily: while deep learning is no doubt one of the most exciting areas in machine learning, you should master the fundamentals first. Moreover, most problems can be solved quite well using simpler techniques such as random forests and ensemble methods (discussed in Part I). deep learning is best suited for complex problems such as image recognition, speech recognition, or natural language processing, and it requires a lot of data, computing power, and patience (unless you can leverage a pretrained neural network, as you will see).

Changes Between the First and the Second Edition

If you have already read the first edition, here are the main changes between the first and the second edition:

- All the code was migrated from TensorFlow 1.x to TensorFlow 2.x, and I replaced most of the low-level TensorFlow code (graphs, sessions, feature columns, estimators, and so on) with much simpler Keras code.
- The second edition introduced the Data API for loading and preprocessing large datasets, the distribution strategies API to train and deploy TF models at scale, TF Serving and Google Cloud AI Platform to productionize models, and (briefly) TF Transform, TFLite, TF Addons/Seq2Seq, TensorFlow.js, and TF Agents.
- It also introduced many additional ML topics, including a new chapter on unsupervised learning, computer vision techniques for object detection and semantic segmentation, handling sequences using convolutional neural networks (CNNs), natural language processing (NLP) using recurrent neural networks (RNNs), CNNs and transformers, GANs, and more.

See *https://homl.info/changes2* for more details.

Changes Between the Second and the Third Edition

If you read the second edition, here are the main changes between the second and the third edition:

- All the code was updated to the latest library versions. In particular, this third edition introduces many new additions to Scikit-Learn (e.g., feature name tracking, histogram-based gradient boosting, label propagation, and more). It also introduces the *Keras Tuner* library for hyperparameter tuning, Hugging Face's *Transformers* library for natural language processing, and Keras's new preprocessing and data augmentation layers.
- Several vision models were added (ResNeXt, DenseNet, MobileNet, CSPNet, and EfficientNet), as well as guidelines for choosing the right one.
- Chapter 15 now analyzes the Chicago bus and rail ridership data instead of generated time series, and it introduces the ARMA model and its variants.
- Chapter 16 on natural language processing now builds an English-to-Spanish translation model, first using an encoder–decoder RNN, then using a transformer model. The chapter also covers language models such as Switch Transformers, DistilBERT, T5, and PaLM (with chain-of-thought prompting). In addition, it introduces vision transformers (ViTs) and gives an overview of a few transformer-based visual models, such as data-efficient image transformers (DeiTs), Perceiver, and DINO, as well as a brief overview of some large multimodal models, including CLIP, DALL·E, Flamingo, and GATO.
- Chapter 17 on generative learning now introduces diffusion models, and shows how to implement a denoising diffusion probabilistic model (DDPM) from scratch.
- Chapter 19 migrated from Google Cloud AI Platform to Google Vertex

AI, and uses distributed Keras Tuner for large-scale hyperparameter search. The chapter now includes TensorFlow.js code that you can experiment with online. It also introduces additional distributed training techniques, including PipeDream and Pathways.

- In order to allow for all the new content, some sections were moved online, including installation instructions, kernel principal component analysis (PCA), mathematical details of Bayesian Gaussian mixtures, TF Agents, and former appendices A (exercise solutions), C (support vector machine math), and E (extra neural net architectures).

See *https://homl.info/changes3* for more details.

Other Resources

Many excellent resources are available to learn about machine learning. For example, Andrew Ng's ML course on Coursera is amazing, although it requires a significant time investment.

There are also many interesting websites about machine learning, including Scikit-Learn's exceptional User Guide. You may also enjoy Dataquest, which provides very nice interactive tutorials, and ML blogs such as those listed on Quora.

There are many other introductory books about machine learning. In particular:

- Joel Grus's *Data Science from Scratch*, 2nd edition (O'Reilly), presents the fundamentals of machine learning and implements some of the main algorithms in pure Python (from scratch, as the name suggests).
- Stephen Marsland's *Machine Learning: An Algorithmic Perspective*, 2nd edition (Chapman & Hall), is a great introduction to machine learning, covering a wide range of topics in depth with code examples in Python (also from scratch, but using NumPy).
- Sebastian Raschka's *Python Machine Learning*, 3rd edition (Packt Publishing), is also a great introduction to machine learning and leverages Python open source libraries (Pylearn 2 and Theano).
- François Chollet's *Deep Learning with Python*, 2nd edition (Manning), is a very practical book that covers a large range of topics in a clear and concise way, as you might expect from the author of the excellent Keras library. It favors code examples over mathematical theory.
- Andriy Burkov's *The Hundred-Page Machine Learning Book* (self-published) is very short but covers an impressive range of topics, introducing them in approachable terms without shying away from the math equations.

- Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin's *Learning from Data* (AMLBook) is a rather theoretical approach to ML that provides deep insights, in particular on the bias/variance trade-off (see Chapter 4).
- Stuart Russell and Peter Norvig's *Artificial Intelligence: A Modern Approach*, 4th edition (Pearson), is a great (and huge) book covering an incredible amount of topics, including machine learning. It helps put ML into perspective.
- Jeremy Howard and Sylvain Gugger's *Deep Learning for Coders with fastai and PyTorch* (O'Reilly) provides a wonderfully clear and practical introduction to deep learning using the fastai and PyTorch libraries.

Finally, joining ML competition websites such as Kaggle.com will allow you to practice your skills on real-world problems, with help and insights from some of the best ML professionals out there.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

Constant width bold

Shows commands or other text that should be typed literally by the user.

Constant width italic

Shows text that should be replaced with user-supplied values or by values determined by context.

Punctuation

To avoid any confusion, punctutation appears outside of quotes throughout the book. My apologies to the purists.

> **TIP**
>
> This element signifies a tip or suggestion.

> **NOTE**
>
> This element signifies a general note.

WARNING

This element indicates a warning or caution.

O’Reilly Online Learning

> **NOTE**
>
> For more than 40 years, *O’Reilly Media* has provided technology and business training, knowledge, and insight to help companies succeed.

Our unique network of experts and innovators share their knowledge and expertise through books, articles, and our online learning platform. O’Reilly’s online learning platform gives you on-demand access to live training courses, in-depth learning paths, interactive coding environments, and a vast collection of text and video from O’Reilly and 200+ other publishers. For more information, visit *https://oreilly.com*.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O’Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

800-998-9938 (in the United States or Canada)

707-829-0515 (international or local)

707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *https://homl.info/oreilly3*.

Email *bookquestions@oreilly.com* to comment or ask technical questions about this book.

For news and information about our books and courses, visit *https://oreilly.com*.

Find us on LinkedIn: *https://linkedin.com/company/oreilly-media*

Follow us on Twitter: *https://twitter.com/oreillymedia*

Watch us on YouTube: *https://youtube.com/oreillymedia*

Acknowledgments

Never in my wildest dreams did I imagine that the first and second editions of this book would get such a large audience. I received so many messages from readers, many asking questions, some kindly pointing out errata, and most sending me encouraging words. I cannot express how grateful I am to all these readers for their tremendous support. Thank you all so very much! Please do not hesitate to file issues on GitHub if you find errors in the code examples (or just to ask questions), or to submit errata if you find errors in the text. Some readers also shared how this book helped them get their first job, or how it helped them solve a concrete problem they were working on. I find such feedback incredibly motivating. If you find this book helpful, I would love it if you could share your story with me, either privately (e.g., via LinkedIn) or publicly (e.g., tweet me at @aureliengeron or write an Amazon review).

Huge thanks as well to all the wonderful people who offered their time and expertise to review this third edition, correcting errors and making countless suggestions. This edition is so much better thanks to them: Olzhas Akpambetov, George Bonner, François Chollet, Siddha Gangju, Sam Goodman, Matt Harrison, Sasha Sobran, Lewis Tunstall, Leandro von Werra, and my dear brother Sylvain. You are all amazing!

I am also very grateful to the many people who supported me along the way, by answering my questions, suggesting improvements, and contributing to the code on GitHub: in particular, Yannick Assogba, Ian Beauregard, Ulf Bissbort, Rick Chao, Peretz Cohen, Kyle Gallatin, Hannes Hapke, Victor Khaustov, Soonson Kwon, Eric Lebigot, Jason Mayes, Laurence Moroney, Sara Robinson, Joaquín Ruales, and Yuefeng Zhou.

This book wouldn’t exist without O’Reilly’s fantastic staff, in particular Nicole Taché, who gave me insightful feedback and was always cheerful, encouraging, and helpful: I could not dream of a better editor. Big thanks to Michele Cronin as well, who cheered me on through the final chapters and managed to get me past the finish line. Thanks to the whole production team,

in particular Elizabeth Kelly and Kristen Brown. Thanks as well to Kim Cofer for the thorough copyediting, and to Johnny O'Toole, who managed the relationship with Amazon and answered many of my questions. Thanks to Kate Dullea for greatly improving my illustrations. Thanks to Marie Beaugureau, Ben Lorica, Mike Loukides, and Laurel Ruma for believing in this project and helping me define its scope. Thanks to Matt Hacker and all of the Atlas team for answering all my technical questions regarding formatting, AsciiDoc, MathML, and LaTeX, and thanks to Nick Adams, Rebecca Demarest, Rachel Head, Judith McConville, Helen Monroe, Karen Montgomery, Rachel Roumeliotis, and everyone else at O'Reilly who contributed to this book.

I'll never forget all the wonderful people who helped me with the first and second editions of this book: friends, colleagues, experts, including many members of the TensorFlow team. The list is long: Olzhas Akpambetov, Karmel Allison, Martin Andrews, David Andrzejewski, Paige Bailey, Lukas Biewald, Eugene Brevdo, William Chargin, François Chollet, Clément Courbet, Robert Crowe, Mark Daoust, Daniel "Wolff" Dobson, Julien Dubois, Mathias Kende, Daniel Kitachewsky, Nick Felt, Bruce Fontaine, Justin Francis, Goldie Gadde, Irene Giannoumis, Ingrid von Glehn, Vincent Guilbeau, Sandeep Gupta, Priya Gupta, Kevin Haas, Eddy Hung, Konstantinos Katsiapis, Viacheslav Kovalevskyi, Jon Krohn, Allen Lavoie, Karim Matrah, Grégoire Mesnil, Clemens Mewald, Dan Moldovan, Dominic Monn, Sean Morgan, Tom O'Malley, James Pack, Alexander Pak, Haesun Park, Alexandre Passos, Ankur Patel, Josh Patterson, André Susano Pinto, Anthony Platanios, Anosh Raj, Oscar Ramirez, Anna Revinskaya, Saurabh Saxena, Salim Sémaoune, Ryan Sepassi, Vitor Sessak, Jiri Simsa, Iain Smears, Xiaodan Song, Christina Sorokin, Michel Tessier, Wiktor Tomczak, Dustin Tran, Todd Wang, Pete Warden, Rich Washington, Martin Wicke, Edd Wilder-James, Sam Witteveen, Jason Zaman, Yuefeng Zhou, and my brother Sylvain.

Last but not least, I am infinitely grateful to my beloved wife, Emmanuelle, and to our three wonderful children, Alexandre, Rémi, and Gabrielle, for encouraging me to work hard on this book. Their insatiable curiosity was

priceless: explaining some of the most difficult concepts in this book to my wife and children helped me clarify my thoughts and directly improved many parts of it. Plus, they keep bringing me cookies and coffee, who could ask for more?

---

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

2 Despite the fact that Yann LeCun’s deep convolutional neural networks had worked well for image recognition since the 1990s, although they were not as general-purpose.