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 *http://bit.ly/hands-on-machine-learning-with-scikit-learn-and-tensorflow* or *https://homl.info/oreilly*.

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com*.

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

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

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Changes in the Second Edition

This second edition has five main objectives:

1. Cover additional topics: additional unsupervised learning techniques (including clustering, anomaly detection, density estimation and mixture models), additional techniques for training deep nets (including self-normalized networks), additional computer vision techniques (including the Xception, SENet, object detection with YOLO, and semantic segmentation using R-CNN), handling sequences using CNNs (including WaveNet), natural language processing using RNNs, CNNs and Transformers, generative adversarial networks, deploying TensorFlow models, and more.
2. Update the book to mention some of the latest results from Deep Learning research.
3. Migrate all TensorFlow chapters to TensorFlow 2, and use TensorFlow's implementation of the Keras API (called tf.keras) whenever possible, to simplify the code examples.
4. Update the code examples to use the latest version of Scikit-Learn, NumPy, Pandas, Matplotlib and other libraries.
5. Clarify some sections and fix some errors, thanks to plenty of great feedback from readers.

Some chapters were added, others were rewritten and a few were reordered. Table P-1 shows the mapping between the 1st edition chapters and the 2nd edition chapters:

Table P-1. Chapter mapping between 1st and 2nd edition

1st Ed. chapter	2nd Ed. Chapter	% Changes	2nd Ed. Title
1	1	<10%	The Machine Learning Landscape
2	2	<10%	End-to-End Machine Learning Project
3	3	<10%	Classification
4	4	<10%	Training Models
5	5	<10%	Support Vector Machines
6	6	<10%	Decision Trees
7	7	<10%	Ensemble Learning and Random Forests
8	8	<10%	Dimensionality Reduction
N/A	9	100% new	Unsupervised Learning Techniques
10	10	~75%	Introduction to Artificial Neural Networks with Keras
11	11	~50%	Training Deep Neural Networks
9	12	100% rewritten	Custom Models and Training with TensorFlow
Part of 12	13	100% rewritten	Loading and Preprocessing Data with TensorFlow
13	14	~50%	Deep Computer Vision Using Convolutional Neural Networks
Part of 14	15	~75%	Processing Sequences Using RNNs and CNNs
Part of 14	16	~90%	Natural Language Processing with RNNs and Attention
15	17	~75%	Autoencoders and GANs
16	18	~75%	Reinforcement Learning
Part of 12	19	100% rewritten	Deploying your TensorFlow Models

More specifically, here are the main changes for each 2nd edition chapter (other than clarifications, corrections and code updates):

- Chapter 1
  - — Added a section on handling mismatch between the training set and the validation & test sets.
- Chapter 2
  - — Added how to compute a confidence interval.
  - — Improved the installation instructions (e.g., for Windows).
  - — Introduced the upgraded `OneHotEncoder` and the new `ColumnTransformer`.
- Chapter 4
  - — Explained the need for training instances to be Independent and Identically Distributed (IID).
- Chapter 7
  - — Added a short section about XGBoost.

- Chapter 9 – new chapter including:
  - Clustering with K-Means, how to choose the number of clusters, how to use it for dimensionality reduction, semi-supervised learning, image segmentation, and more.
  - The DBSCAN clustering algorithm and an overview of other clustering algorithms available in Scikit-Learn.
  - Gaussian mixture models, the Expectation-Maximization (EM) algorithm, Bayesian variational inference, and how mixture models can be used for clustering, density estimation, anomaly detection and novelty detection.
  - Overview of other anomaly detection and novelty detection algorithms.
- Chapter 10 (mostly new)
  - Added an introduction to the Keras API, including all its APIs (Sequential, Functional and Subclassing), persistence and callbacks (including the `TensorBoard` callback).
- Chapter 11 (many changes)
  - Introduced self-normalizing nets, the SELU activation function and Alpha Dropout.
  - Introduced self-supervised learning.
  - Added Nadam optimization.
  - Added Monte-Carlo Dropout.
  - Added a note about the risks of adaptive optimization methods.
  - Updated the practical guidelines.
- Chapter 12 – completely rewritten chapter, including:
  - A tour of TensorFlow 2
  - TensorFlow's lower-level Python API
  - Writing custom loss functions, metrics, layers, models
  - Using auto-differentiation and creating custom training algorithms.
  - TensorFlow Functions and graphs (including tracing and autograph).
- Chapter 13 – new chapter, including:
  - The Data API
  - Loading/Storing data efficiently using TFRecords
  - The Features API (including an introduction to embeddings).
  - An overview of TF Transform and TF Datasets
  - Moved the low-level implementation of the neural network to the exercises.

  - Removed details about queues and readers that are now superseded by the Data API.
- Chapter 14
  - Added Xception and SENet architectures.
  - Added a Keras implementation of ResNet-34.
  - Showed how to use pretrained models using Keras.
  - Added an end-to-end transfer learning example.
  - Added classification and localization.
  - Introduced Fully Convolutional Networks (FCNs).
  - Introduced object detection using the YOLO architecture.
  - Introduced semantic segmentation using R-CNN.
- Chapter 15
  - Added an introduction to Wavenet.
  - Moved the Encoder–Decoder architecture and Bidirectional RNNs to Chapter 16.
- Chapter 16
  - Explained how to use the Data API to handle sequential data.
  - Showed an end-to-end example of text generation using a Character RNN, using both a stateless and a stateful RNN.
  - Showed an end-to-end example of sentiment analysis using an LSTM.
  - Explained masking in Keras.
  - Showed how to reuse pretrained embeddings using TF Hub.
  - Showed how to build an Encoder–Decoder for Neural Machine Translation using TensorFlow Addons/seq2seq.
  - Introduced beam search.
  - Explained attention mechanisms.
  - Added a short overview of visual attention and a note on explainability.
  - Introduced the fully attention-based Transformer architecture, including positional embeddings and multi-head attention.
  - Added an overview of recent language models (2018).
- Chapters 17, 18 and 19: coming soon.

Acknowledgments

Never in my wildest dreams did I imagine that the first edition 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., in an Amazon review).

I am also incredibly thankful to all the amazing people who took time out of their busy lives to review my book with such care. In particular, I would like to thank François Chollet for reviewing all the chapters based on Keras & TensorFlow, and giving me some great, in-depth feedback. Since Keras is one of the main additions to this 2nd edition, having its author review the book was invaluable. I highly recommend François's excellent book Deep Learning with Python[3]: it has the conciseness, clarity and depth of the Keras library itself. Big thanks as well to Ankur Patel, who reviewed every chapter of this 2nd edition and gave me excellent feedback.

This book also benefited from plenty of help from members of the TensorFlow team, in particular Martin Wicke, who tirelessly answered dozens of my questions and dispatched the rest to the right people, including Alexandre Passos, Allen Lavoie, André Susano Pinto, Anna Revinskaya, Anthony Platanios, Clemens Mewald, Dan Moldovan, Daniel Dobson, Dustin Tran, Edd Wilder-James, Goldie Gadde, Jiri Simsa, Karmel Allison, Nick Felt, Paige Bailey, Pete Warden (who also reviewed the 1st edition), Ryan Sepassi, Sandeep Gupta, Sean Morgan, Todd Wang, Tom O'Malley, William Chargin, and Yuefeng Zhou, all of whom were tremendously helpful. A huge thank you to all of you, and to all other members of the TensorFlow team. Not just for your help, but also for making such a great library.

Big thanks to Haesun Park, who gave me plenty of excellent feedback and caught several errors while he was writing the Korean translation of the 1st edition of this book. He also translated the Jupyter notebooks to Korean, not to mention TensorFlow's documentation. I do not speak Korean, but judging by the quality of his feedback, all his translations must be truly excellent! Moreover, he kindly contributed some of the solutions to the exercises in this book.

3 "Deep Learning with Python," François Chollet (2017).

Many thanks as well to O'Reilly's fantastic staff, in particular Nicole Tache, who gave me insightful feedback, always cheerful, encouraging, and helpful: I could not dream of a better editor. Big thanks to Michele Cronin as well, who was very helpful (and patient) at the start of this 2nd edition. 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, and LaTeX, and thanks to Rachel Monaghan, Nick Adams, and all of the production team for their final review and their hundreds of corrections.

I would also like to thank my former Google colleagues, in particular the YouTube video classification team, for teaching me so much about Machine Learning. I could never have started the first edition without them. Special thanks to my personal ML gurus: Clément Courbet, Julien Dubois, Mathias Kende, Daniel Kitachewsky, James Pack, Alexander Pak, Anosh Raj, Vitor Sessak, Wiktor Tomczak, Ingrid von Glehn, Rich Washington, and everyone I worked with at YouTube and in the amazing Google research teams in Mountain View. All these people are just as nice and helpful as they are bright, and that's saying a lot.

I will never forget the kind people who reviewed the 1st edition of this book, including David Andrzejewski, Eddy Hung, Grégoire Mesnil, Iain Smears, Ingrid von Glehn, Justin Francis, Karim Matrah, Lukas Biewald, Michel Tessier, Salim Sémaoune, Vincent Guilbeau and of course my dear 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, as well as for their insatiable curiosity: 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 this book. Plus, they keep bringing me cookies and coffee! What more can one dream of?

PART I

The Fundamentals of Machine Learning

CHAPTER 1
The Machine Learning Landscape

With Early Release ebooks, you get books in their earliest form—the author's raw and unedited content as he or she writes—so you can take advantage of these technologies long before the official release of these titles. The following will be Chapter 1 in the final release of the book.

When most people hear "Machine Learning," they picture a robot: a dependable butler or a deadly Terminator depending on who you ask. But Machine Learning is not just a futuristic fantasy, it's already here. In fact, it has been around for decades in some specialized applications, such as *Optical Character Recognition* (OCR). But the first ML application that really became mainstream, improving the lives of hundreds of millions of people, took over the world back in the 1990s: it was the *spam filter*. Not exactly a self-aware Skynet, but it does technically qualify as Machine Learning (it has actually learned so well that you seldom need to flag an email as spam anymore). It was followed by hundreds of ML applications that now quietly power hundreds of products and features that you use regularly, from better recommendations to voice search.

Where does Machine Learning start and where does it end? What exactly does it mean for a machine to *learn* something? If I download a copy of Wikipedia, has my computer really "learned" something? Is it suddenly smarter? In this chapter we will start by clarifying what Machine Learning is and why you may want to use it.

Then, before we set out to explore the Machine Learning continent, we will take a look at the map and learn about the main regions and the most notable landmarks: supervised versus unsupervised learning, online versus batch learning, instance-based versus model-based learning. Then we will look at the workflow of a typical ML project, discuss the main challenges you may face, and cover how to evaluate and fine-tune a Machine Learning system.

This chapter introduces a lot of fundamental concepts (and jargon) that every data scientist should know by heart. It will be a high-level overview (the only chapter without much code), all rather simple, but you should make sure everything is crystal-clear to you before continuing to the rest of the book. So grab a coffee and let's get started!

If you already know all the Machine Learning basics, you may want to skip directly to Chapter 2. If you are not sure, try to answer all the questions listed at the end of the chapter before moving on.

What Is Machine Learning?

Machine Learning is the science (and art) of programming computers so they can *learn from data*.

Here is a slightly more general definition:

> [Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.
>
> —Arthur Samuel, *1959*

And a more engineering-oriented one:

> A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.
>
> —Tom Mitchell, *1997*

For example, your spam filter is a Machine Learning program that can learn to flag spam given examples of spam emails (e.g., flagged by users) and examples of regular (nonspam, also called "ham") emails. The examples that the system uses to learn are called the *training set*. Each training example is called a *training instance* (or *sample*). In this case, the task T is to flag spam for new emails, the experience E is the *training data*, and the performance measure P needs to be defined; for example, you can use the ratio of correctly classified emails. This particular performance measure is called *accuracy* and it is often used in classification tasks.

If you just download a copy of Wikipedia, your computer has a lot more data, but it is not suddenly better at any task. Thus, it is not Machine Learning.

Why Use Machine Learning?

Consider how you would write a spam filter using traditional programming techniques (Figure 1-1):