

Saving and Loading Models That Contain Custom Components	377
Custom Activation Functions, Initializers, Regularizers, and Constraints	379
Custom Metrics	380
Custom Layers	383
Custom Models	386
Losses and Metrics Based on Model Internals	388
Computing Gradients Using Autodiff	389
Custom Training Loops	393
TensorFlow Functions and Graphs	396
Autograph and Tracing	398
TF Function Rules	400
13. Loading and Preprocessing Data with TensorFlow.....	403
The Data API	404
Chaining Transformations	405
Shuffling the Data	406
Preprocessing the Data	409
Putting Everything Together	410
Prefetching	411
Using the Dataset With tf.keras	413
The TFRecord Format	414
Compressed TFRecord Files	415
A Brief Introduction to Protocol Buffers	415
TensorFlow Protobufs	416
Loading and Parsing Examples	418
Handling Lists of Lists Using the SequenceExample Protobuf	419
The Features API	420
Categorical Features	421
Crossed Categorical Features	421
Encoding Categorical Features Using One-Hot Vectors	422
Encoding Categorical Features Using Embeddings	423
Using Feature Columns for Parsing	426
Using Feature Columns in Your Models	426
TF Transform	428
The TensorFlow Datasets (TFDS) Project	429
14. Deep Computer Vision Using Convolutional Neural Networks.....	431
The Architecture of the Visual Cortex	432
Convolutional Layer	434
Filters	436
Stacking Multiple Feature Maps	437
TensorFlow Implementation	439

Memory Requirements	441
Pooling Layer	442
TensorFlow Implementation	444
CNN Architectures	446
LeNet-5	449
AlexNet	450
GoogLeNet	452
VGGNet	456
ResNet	457
Xception	459
SENet	461
Implementing a ResNet-34 CNN Using Keras	464
Using Pretrained Models From Keras	465
Pretrained Models for Transfer Learning	467
Classification and Localization	469
Object Detection	471
Fully Convolutional Networks (FCNs)	473
You Only Look Once (YOLO)	475
Semantic Segmentation	478
Exercises	482

The Machine Learning Tsunami

In 2006, Geoffrey Hinton et al. published a paper¹ 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.” Training a deep neural net was widely considered impossible at the time,² and most researchers had abandoned the idea since the 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.

Fast-forward 10 years and Machine Learning has conquered the industry: it is now at the heart of much of the magic in today’s high-tech products, ranking your web search results, powering your smartphone’s speech recognition, recommending videos, and beating the world champion at the game of Go. Before you know it, it will be driving your car.

Machine Learning in Your Projects

So naturally you are excited about Machine Learning and you 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?

1 Available on Hinton’s home page at <http://www.cs.toronto.edu/~hinton/>.

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.

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; for example:

- 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
- **And more**

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, the intuitions, and the tools you need to actually 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.

Rather than implementing our own toy versions of each algorithm, we will be using actual 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 learn Machine Learning.
- **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 servers. TensorFlow was created at Google and supports many of their large-scale Machine Learning applications. It was open sourced in November 2015.
- **Keras** is a high level Deep Learning API that makes it very simple to train and run neural networks. It can run on top of either TensorFlow, Theano or Microsoft Cognitive Toolkit (formerly known as CNTK). TensorFlow comes with its own implementation of this API, called *tf.keras*, which provides support for some advanced TensorFlow features (e.g., to efficiently load data).

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. While you can read this book without picking up your laptop, we highly recommend

you experiment with the code examples available online as Jupyter notebooks at <https://github.com/ageron/handson-ml2>.

Prerequisites

This book assumes that you have some Python programming experience and that you are familiar with Python's main scientific libraries, in particular **NumPy**, **Pandas**, and **Matplotlib**.

Also, if you care about what's under the hood you should have a reasonable understanding of college-level math as well (calculus, linear algebra, probabilities, and statistics).

If you don't know Python yet, <http://learnpython.org/> is a great place to start. The official tutorial on python.org is also quite good.

If you have never used Jupyter, **Chapter 2** will guide you through installation and the basics: it is a great tool to have in your toolbox.

If you are not familiar with Python's scientific libraries, the provided Jupyter notebooks include a few tutorials. There is also a quick math tutorial for linear algebra.

Roadmap

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

- What is Machine Learning? What problems does it try to solve? What are the main categories and fundamental concepts of Machine Learning systems?
- The main 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 main challenges of Machine Learning, in particular underfitting and overfitting (the bias/variance tradeoff).
- Reducing the dimensionality of the training data to fight the curse of dimensionality.
- Other unsupervised learning techniques, including clustering, density estimation and anomaly detection.

- The most common learning algorithms: Linear and Polynomial Regression, Logistic Regression, k-Nearest Neighbors, Support Vector Machines, Decision Trees, Random Forests, and Ensemble methods.

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

- What are neural nets? What are they good for?
- Building and training neural nets using TensorFlow and Keras.
- The most important neural net architectures: feedforward neural nets, convolutional nets, recurrent nets, long short-term memory (LSTM) nets, autoencoders and generative adversarial networks (GANs).
- Techniques for training deep neural nets.
- Scaling neural networks for large datasets.
- Learning strategies with Reinforcement Learning.
- Handling uncertainty with Bayesian Deep Learning.

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



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, provided you have enough data, computing power, and patience.

Other Resources

Many resources are available to learn about Machine Learning. Andrew Ng's [ML course on Coursera](#) and Geoffrey Hinton's [course on neural networks and Deep Learning](#) are amazing, although they both require a significant time investment (think months).

There are also many interesting websites about Machine Learning, including of course 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](#). Finally, the [Deep Learning website](#) has a good list of resources to learn more.

Of course there are also many other introductory books about Machine Learning, in particular:

- Joel Grus, *Data Science from Scratch* (O'Reilly). This book presents the fundamentals of Machine Learning, and implements some of the main algorithms in pure Python (from scratch, as the name suggests).

- Stephen Marsland, *Machine Learning: An Algorithmic Perspective* (Chapman and Hall). This book 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, *Python Machine Learning* (Packt Publishing). Also a great introduction to Machine Learning, this book leverages Python open source libraries (Pylearn 2 and Theano).
- François Chollet, *Deep Learning with Python* (Manning). 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.
- Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin, *Learning from Data* (AMLBook). A rather theoretical approach to ML, this book provides deep insights, in particular on the bias/variance tradeoff (see [Chapter 4](#)).
- Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach, 3rd Edition* (Pearson). This is a great (and huge) book covering an incredible amount of topics, including Machine Learning. It helps put ML into perspective.

Finally, a great way to learn is to join ML competition websites such as [Kaggle.com](#) this 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.



This element signifies a tip or suggestion.



This element signifies a general note.



This element indicates a warning or caution.

Code Examples

Supplemental material (code examples, exercises, etc.) is available for download at <https://github.com/ageron/handson-ml2>. It is mostly composed of Jupyter notebooks.

Some of the code examples in the book leave out some repetitive sections, or details that are obvious or unrelated to Machine Learning. This keeps the focus on the important parts of the code, and it saves space to cover more topics. However, if you want the full code examples, they are all available in the Jupyter notebooks.

Note that when the code examples display some outputs, then these code examples are shown with Python prompts (`>>>` and `...`), as in a Python shell, to clearly distinguish the code from the outputs. For example, this code defines the `square()` function then it computes and displays the square of 3:

```
>>> def square(x):  
...     return x ** 2  
...  
>>> result = square(3)  
>>> result  
9
```

When code does not display anything, prompts are not used. However, the result may sometimes be shown as a comment like this:


```
def square(x):  
    return x ** 2  
  
result = square(3) # result is 9
```

Using Code Examples

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: “*Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* by Aurélien Géron (O'Reilly). Copyright 2019 Aurélien Géron, 978-1-492-03264-9.” If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

O'Reilly Safari

 **Safari**® *Safari* (formerly Safari Books Online) is a membership-based training and reference platform for enterprise, government, educators, and individuals.

Members have access to thousands of books, training videos, Learning Paths, interactive tutorials, and curated playlists from over 250 publishers, including O'Reilly Media, Harvard Business Review, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Adobe, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, and Course Technology, among others.

For more information, please visit <http://oreilly.com/safari>.

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)