

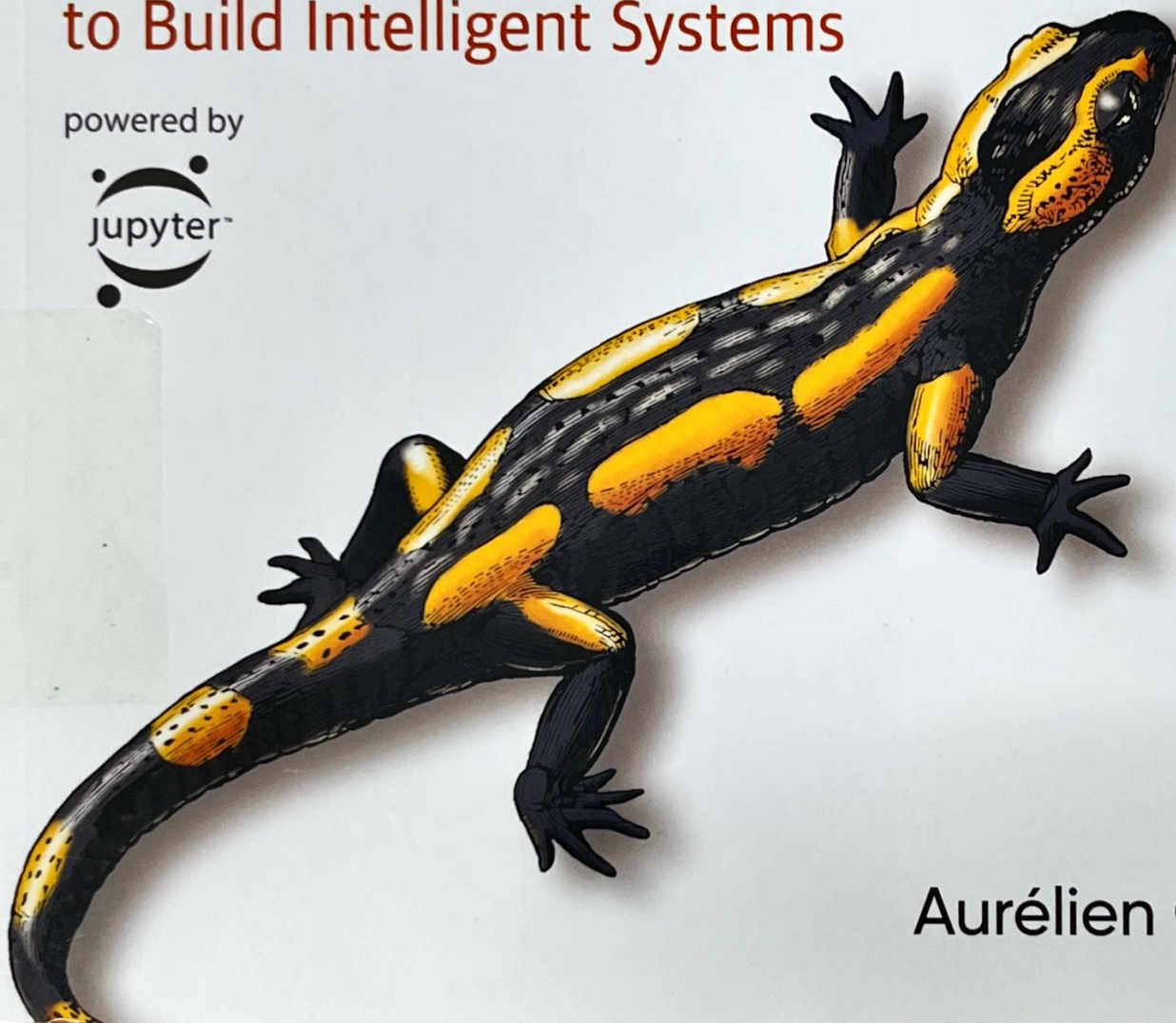
O'REILLY®

2nd Edition
Updated for
TensorFlow 2

Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow

Concepts, Tools, and Techniques
to Build Intelligent Systems

powered by

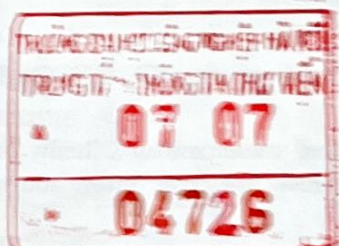


Aurélien Géron

SECOND EDITION

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

*Concepts, Tools, and Techniques to
Build Intelligent Systems*



Aurélien Geron

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

Table of Contents

Preface.....	xv
--------------	----

Part I. The Fundamentals of Machine Learning

1. The Machine Learning Landscape.....	1
What Is Machine Learning?	2
Why Use Machine Learning?	2
Examples of Applications	5
Types of Machine Learning Systems	7
Supervised/Unsupervised Learning	7
Batch and Online Learning	14
Instance-Based Versus Model-Based Learning	17
Main Challenges of Machine Learning	23
Insufficient Quantity of Training Data	23
Nonrepresentative Training Data	25
Poor-Quality Data	26
Irrelevant Features	27
Overfitting the Training Data	27
Underfitting the Training Data	29
Stepping Back	30
Testing and Validating	30
Hyperparameter Tuning and Model Selection	31
Data Mismatch	32
Exercises	33
2. End-to-End Machine Learning Project.....	35
Working with Real Data	35

Look at the Big Picture	37
Frame the Problem	37
Select a Performance Measure	39
Check the Assumptions	42
Get the Data	42
Create the Workspace	42
Download the Data	46
Take a Quick Look at the Data Structure	47
Create a Test Set	51
Discover and Visualize the Data to Gain Insights	56
Visualizing Geographical Data	56
Looking for Correlations	58
Experimenting with Attribute Combinations	61
Prepare the Data for Machine Learning Algorithms	62
Data Cleaning	63
Handling Text and Categorical Attributes	65
Custom Transformers	68
Feature Scaling	69
Transformation Pipelines	70
Select and Train a Model	72
Training and Evaluating on the Training Set	72
Better Evaluation Using Cross-Validation	73
Fine-Tune Your Model	75
Grid Search	76
Randomized Search	78
Ensemble Methods	78
Analyze the Best Models and Their Errors	78
Evaluate Your System on the Test Set	79
Launch, Monitor, and Maintain Your System	80
Try It Out!	83
Exercises	84
3. Classification.....	85
MNIST	85
Training a Binary Classifier	88
Performance Measures	88
Measuring Accuracy Using Cross-Validation	89
Confusion Matrix	90
Precision and Recall	92
Precision/Recall Trade-off	93
The ROC Curve	97
Multiclass Classification	100

Error Analysis	102
Multilabel Classification	106
Multioutput Classification	107
Exercises	108
4. Training Models.....	111
Linear Regression	112
The Normal Equation	114
Computational Complexity	117
Gradient Descent	118
Batch Gradient Descent	121
Stochastic Gradient Descent	124
Mini-batch Gradient Descent	127
Polynomial Regression	128
Learning Curves	130
Regularized Linear Models	134
Ridge Regression	135
Lasso Regression	137
Elastic Net	140
Early Stopping	141
Logistic Regression	142
Estimating Probabilities	143
Training and Cost Function	144
Decision Boundaries	145
Softmax Regression	148
Exercises	151
5. Support Vector Machines.....	153
Linear SVM Classification	153
Soft Margin Classification	154
Nonlinear SVM Classification	157
Polynomial Kernel	158
Similarity Features	159
Gaussian RBF Kernel	160
Computational Complexity	162
SVM Regression	162
Under the Hood	164
Decision Function and Predictions	165
Training Objective	166
Quadratic Programming	167
The Dual Problem	168
Kernelized SVMs	169

Online SVMs	172
Exercises	174
6. Decision Trees	175
Training and Visualizing a Decision Tree	175
Making Predictions	176
Estimating Class Probabilities	178
The CART Training Algorithm	179
Computational Complexity	180
Gini Impurity or Entropy?	180
Regularization Hyperparameters	181
Regression	183
Instability	185
Exercises	186
7. Ensemble Learning and Random Forests	189
Voting Classifiers	189
Bagging and Pasting	192
Bagging and Pasting in Scikit-Learn	194
Out-of-Bag Evaluation	195
Random Patches and Random Subspaces	196
Random Forests	197
Extra-Trees	198
Feature Importance	198
Boosting	199
AdaBoost	200
Gradient Boosting	203
Stacking	208
Exercises	211
8. Dimensionality Reduction	213
The Curse of Dimensionality	214
Main Approaches for Dimensionality Reduction	215
Projection	215
Manifold Learning	218
PCA	219
Preserving the Variance	219
Principal Components	220
Projecting Down to d Dimensions	221
Using Scikit-Learn	222
Explained Variance Ratio	222
Choosing the Right Number of Dimensions	223

PCA for Compression	224
Randomized PCA	225
Incremental PCA	225
Kernel PCA	226
Selecting a Kernel and Tuning Hyperparameters	227
LLE	230
Other Dimensionality Reduction Techniques	232
Exercises	233

9. Unsupervised Learning Techniques..... 235

Clustering	236
K-Means	238
Limits of K-Means	248
Using Clustering for Image Segmentation	249
Using Clustering for Preprocessing	251
Using Clustering for Semi-Supervised Learning	253
DBSCAN	255
Other Clustering Algorithms	258
Gaussian Mixtures	260
Anomaly Detection Using Gaussian Mixtures	266
Selecting the Number of Clusters	267
Bayesian Gaussian Mixture Models	270
Other Algorithms for Anomaly and Novelty Detection	274
Exercises	275

Part II. Neural Networks and Deep Learning

10. Introduction to Artificial Neural Networks with Keras..... 279

From Biological to Artificial Neurons	280
Biological Neurons	281
Logical Computations with Neurons	283
The Perceptron	284
The Multilayer Perceptron and Backpropagation	289
Regression MLPs	292
Classification MLPs	294
Implementing MLPs with Keras	295
Installing TensorFlow 2	296
Building an Image Classifier Using the Sequential API	297
Building a Regression MLP Using the Sequential API	307
Building Complex Models Using the Functional API	308
Using the Subclassing API to Build Dynamic Models	313

Saving and Restoring a Model	314
Using Callbacks	315
Using TensorBoard for Visualization	317
Fine-Tuning Neural Network Hyperparameters	320
Number of Hidden Layers	323
Number of Neurons per Hidden Layer	325
Learning Rate, Batch Size, and Other Hyperparameters	325
Exercises	327
11. Training Deep Neural Networks.....	331
The Vanishing/Exploding Gradients Problems	332
Glorot and He Initialization	333
Nonsaturating Activation Functions	335
Batch Normalization	338
Gradient Clipping	345
Reusing Pretrained Layers	345
Transfer Learning with Keras	347
Unsupervised Pretraining	349
Pretraining on an Auxiliary Task	350
Faster Optimizers	351
Momentum Optimization	351
Nesterov Accelerated Gradient	353
AdaGrad	354
RMSProp	355
Adam and Nadam Optimization	356
Learning Rate Scheduling	359
Avoiding Overfitting Through Regularization	364
ℓ_1 and ℓ_2 Regularization	364
Dropout	365
Monte Carlo (MC) Dropout	368
Max-Norm Regularization	370
Summary and Practical Guidelines	371
Exercises	373
12. Custom Models and Training with TensorFlow.....	375
A Quick Tour of TensorFlow	376
Using TensorFlow like NumPy	379
Tensors and Operations	379
Tensors and NumPy	381
Type Conversions	381
Variables	382
Other Data Structures	383

Customizing Models and Training Algorithms	384
Custom Loss Functions	384
Saving and Loading Models That Contain Custom Components	385
Custom Activation Functions, Initializers, Regularizers, and Constraints	387
Custom Metrics	388
Custom Layers	391
Custom Models	394
Losses and Metrics Based on Model Internals	397
Computing Gradients Using Autodiff	399
Custom Training Loops	402
TensorFlow Functions and Graphs	405
AutoGraph and Tracing	407
TF Function Rules	409
Exercises	410
13. Loading and Preprocessing Data with TensorFlow.....	413
The Data API	414
Chaining Transformations	415
Shuffling the Data	416
Preprocessing the Data	419
Putting Everything Together	420
Prefetching	421
Using the Dataset with tf.keras	423
The TFRecord Format	424
Compressed TFRecord Files	425
A Brief Introduction to Protocol Buffers	425
TensorFlow Protobufs	427
Loading and Parsing Examples	428
Handling Lists of Lists Using the SequenceExample Protobuf	429
Preprocessing the Input Features	430
Encoding Categorical Features Using One-Hot Vectors	431
Encoding Categorical Features Using Embeddings	433
Keras Preprocessing Layers	437
TF Transform	439
The TensorFlow Datasets (TFDS) Project	441
Exercises	442
14. Deep Computer Vision Using Convolutional Neural Networks.....	445
The Architecture of the Visual Cortex	446
Convolutional Layers	448
Filters	450
Stacking Multiple Feature Maps	451

TensorFlow Implementation	453
Memory Requirements	456
Pooling Layers	456
TensorFlow Implementation	458
CNN Architectures	460
LeNet-5	463
AlexNet	464
GoogLeNet	466
VGGNet	470
ResNet	471
Xception	474
SENet	476
Implementing a ResNet-34 CNN Using Keras	478
Using Pretrained Models from Keras	479
Pretrained Models for Transfer Learning	481
Classification and Localization	483
Object Detection	485
Fully Convolutional Networks	487
You Only Look Once (YOLO)	489
Semantic Segmentation	492
Exercises	496
15. Processing Sequences Using RNNs and CNNs.....	497
Recurrent Neurons and Layers	498
Memory Cells	500
Input and Output Sequences	501
Training RNNs	502
Forecasting a Time Series	503
Baseline Metrics	505
Implementing a Simple RNN	505
Deep RNNs	506
Forecasting Several Time Steps Ahead	508
Handling Long Sequences	511
Fighting the Unstable Gradients Problem	512
Tackling the Short-Term Memory Problem	514
Exercises	523
16. Natural Language Processing with RNNs and Attention.....	525
Generating Shakespearean Text Using a Character RNN	526
Creating the Training Dataset	527
How to Split a Sequential Dataset	527
Chopping the Sequential Dataset into Multiple Windows	528

Building and Training the Char-RNN Model	530
Using the Char-RNN Model	531
Generating Fake Shakespearean Text	531
Stateful RNN	532
Sentiment Analysis	534
Masking	538
Reusing Pretrained Embeddings	540
An Encoder–Decoder Network for Neural Machine Translation	542
Bidirectional RNNs	546
Beam Search	547
Attention Mechanisms	549
Visual Attention	552
Attention Is All You Need: The Transformer Architecture	554
Recent Innovations in Language Models	563
Exercises	565
17. Representation Learning and Generative Learning Using Autoencoders and GANs.	567
Efficient Data Representations	569
Performing PCA with an Undercomplete Linear Autoencoder	570
Stacked Autoencoders	572
Implementing a Stacked Autoencoder Using Keras	572
Visualizing the Reconstructions	574
Visualizing the Fashion MNIST Dataset	574
Unsupervised Pretraining Using Stacked Autoencoders	576
Tying Weights	577
Training One Autoencoder at a Time	578
Convolutional Autoencoders	579
Recurrent Autoencoders	580
Denoising Autoencoders	581
Sparse Autoencoders	582
Variational Autoencoders	586
Generating Fashion MNIST Images	590
Generative Adversarial Networks	592
The Difficulties of Training GANs	596
Deep Convolutional GANs	598
Progressive Growing of GANs	601
StyleGANs	604
Exercises	607
18. Reinforcement Learning.....	609
Learning to Optimize Rewards	610
Policy Search	612

Introduction to OpenAI Gym	613
Neural Network Policies	617
Evaluating Actions: The Credit Assignment Problem	619
Policy Gradients	620
Markov Decision Processes	625
Temporal Difference Learning	629
Q-Learning	630
Exploration Policies	632
Approximate Q-Learning and Deep Q-Learning	633
Implementing Deep Q-Learning	634
Deep Q-Learning Variants	639
Fixed Q-Value Targets	639
Double DQN	640
Prioritized Experience Replay	640
Dueling DQN	641
The TF-Agents Library	642
Installing TF-Agents	643
TF-Agents Environments	643
Environment Specifications	644
Environment Wrappers and Atari Preprocessing	645
Training Architecture	649
Creating the Deep Q-Network	650
Creating the DQN Agent	652
Creating the Replay Buffer and the Corresponding Observer	654
Creating Training Metrics	655
Creating the Collect Driver	656
Creating the Dataset	658
Creating the Training Loop	661
Overview of Some Popular RL Algorithms	662
Exercises	664
19. Training and Deploying TensorFlow Models at Scale.	667
Serving a TensorFlow Model	668
Using TensorFlow Serving	668
Creating a Prediction Service on GCP AI Platform	677
Using the Prediction Service	682
Deploying a Model to a Mobile or Embedded Device	685
Using GPUs to Speed Up Computations	689
Getting Your Own GPU	690
Using a GPU-Equipped Virtual Machine	692
Colaboratory	693
Managing the GPU RAM	694

Placing Operations and Variables on Devices	697
Parallel Execution Across Multiple Devices	699
Training Models Across Multiple Devices	701
Model Parallelism	701
Data Parallelism	704
Training at Scale Using the Distribution Strategies API	709
Training a Model on a TensorFlow Cluster	711
Running Large Training Jobs on Google Cloud AI Platform	714
Black Box Hyperparameter Tuning on AI Platform	716
Exercises	717
Thank You!	718
A. Exercise Solutions.....	719
B. Machine Learning Project Checklist.....	755
C. SVM Dual Problem.....	761
D. Autodiff.....	765
E. Other Popular ANN Architectures.....	773
F. Special Data Structures.....	783
G. TensorFlow Graphs.....	791
Index.....	801

Preface

The Machine Learning Tsunami

In 2006, Geoffrey Hinton et al. published a paper (<https://homl.info/136>)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 or so later, Machine Learning has conquered the industry: it is 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 would love to join the party!

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.

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 more (<https://homl.info/usecases>):

- 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.

Rather than implementing our own toy versions of each algorithm, we will be using production-ready Python frameworks:

- Scikit-Learn (<http://scikit-learn.org/>) 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 (<https://tensorflow.org/>) 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 (<https://keras.io/>) 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., the ability 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, I 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 (<http://numpy.org/>), pandas (<http://pandas.pydata.org/>), and Matplotlib (<http://matplotlib.org/>).

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 (<https://docs.python.org/3/tutorial/>) is also quite good.

If you have never used Jupyter, Chapter 2 will guide you through installation and the basics: it is a powerful 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 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, autoencoders and generative adversarial networks (GANs) 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.



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.