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." 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, and recommending videos, 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
- Predict 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 using data flow graphs. It makes it possible to train and run very large neural networks efficiently by distributing the computations across potentially thousands 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.

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-ml*.

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