Book Review

Python Machine Learning:

Machine Learning and Deep Learning With Python, Scikit-Learn, and TensorFlow 2, Third Edition

Reviewed by Yuan Ren, Shanghai Dianji University, Shanghai, China

Python Machine Learning: Machine Learning and Deep Learning With Python, Scikit-Learn, and TensorFlow 2, Third Edition
Sebastian Raschka, Vahid Mirjalili
Copyright 2019 Packt Publishing Ltd., Birmingham, UK
770 pages
$35.99
ISBN 978 1 78995 575 0

Python is one of the most popular programming languages for data science and thanks to its developer and open source community, a large number of useful libraries for scientific computing and machine learning have been developed. This book is a comprehensive introduction to machine learning (ML) with Python. It is both a tutorial and a reference for ML system designer. With detailed explanations, visualizations, and code examples, this edition covers important ML techniques and principles. Updated for TensorFlow 2, this edition introduces the new Keras package and scikit-learn (sklearn) package. This book covers state-of-the-art reinforcement learning (RL) techniques, and introduces generative adversarial networks (GANs). This book also investigates sentiment analysis.

This book is divided into 18 chapters. Each chapter centers on an essential aspect of ML, and is totally worth deep reading for all learners.

The first chapter raises a basic question "how to give computers the ability to learn from data". Common terminology used in ML is explain at the beginning of this chapter. This chapter discusses the importance of giving computers the ability to learn from data, and three types of ML techniques. The first type is learning that contains label data, direct feedback, and prediction of the outcome. The second type is learning that needs no labels or feedback and can find hidden structures within data. The third type is learning with decision process, reward system, and series of action. Each type of ML can be applied using different machines at a time.

Training simple ML algorithms for classification is the main topic of Chapter 2. This is a classic question that goes back to the origin of ML. Chapter 2 introduces binary perceptron classifiers and adaptive linear neurons for supervised learning. This chapter is a gentle introduction to the fundamentals of pattern classification, and focuses on how to use the sklearn ML library. This package enables you to access more advanced learning classifiers.

A tour of machine learning classifiers using sklearn is given in the next chapter. Chapter 3 describes essential ML algorithms for classification, and provides practical examples using one of the most popular and comprehensive open source ML libraries, sklearn. This chapter introduces robust and popular algorithms for classification, such as logistic regression, support vector machines, and decision trees. This chapter gives examples and explanations using the sklearn which provides various ML algorithms via a user friendly application programming interface (API). This chapter also discusses about the advantages and disadvantages of classifiers with linear and nonlinear decision boundaries.

How to build good training datasets is an important question for data preprocessing. Chapter 4 discusses how to deal with the most common problems in unprocessed datasets, such as missing data. Support Vector Machines are the powerful linear models that can be extended to nonlinear models. This means that they have many parameters that need to be all in tuned in order to make good predictions. K-nearest neighbor is seen as a lazy way of classification which allows us to make predictions without any model training. This chapter also discusses several approaches of identifying the most informative features in datasets and preparing different types of variables as proper input for ML algorithms. This chapter reminds the readers that no algorithm can make good predictions if the informative and discriminatory features are not in place.

Compressing data via dimensionality reduction is an essential technique for lowering model training time. The author talks about three different fundamental dimensionality reduction techniques for feature extraction standard in Chapter 5. They are principal component analysis (PCA), linear discriminant analysis (LDA), and kernel PCA (KPCA). Using PCA, data was projected onto a lower-dimensional subspace. On the contrast, LDA is a technique for supervised dimensionality reduction. And KPCA is the nonlinear feature extractor, able to compress datasets consisting of nonlinear features onto a lower-dimensional subspace where the classes became linearly separable.

Learning best practices for model evaluation and hyperparameter tuning is then discussed for various models. Chapter 6 introduces different transformation techniques and classifiers in convenient model pipelines and fine-tune ML models, and explains how to deal with imbalanced data. This chapter also discusses how to evaluate the performance of ML models, how to diagnose the common problems of machine learning algorithms, and how to evaluate predictive models using different performance metrics.

Chapter 7 combines different models for ensemble learning. The author investigates ensemble methods that combine multiple models and classification algorithms to boost the predictive performance of an ML system even further. This chapter explores how to build ensembles of experts to overcome the weaknesses of individual learners. This chapter also discusses how to make predictions based on majority voting, and how to use bagging to reduce overfitting.

MLs are applied for sentiment analysis in Chapter 8. This chapter focuses on how to encode a document as a feature vector using the bag-of-words model, and how to weight the term frequency by relevance using term frequency-inverse document frequency. Topics covered in this chapter also include: preparing text, building feature vectors from text, training an ML model to classify movie reviews, working with large scale text, and inferring topics from documents for categorization.

An ML model can also embedded into a web application. ML techniques are not only offline applications, but also the predictive engine of web services. Chapter 9 shows how to embed a document classifier into a web application that can not only classify, but also learn from data in real time. This chapter introduces approaches of how to save the state of a trained model, how to use databases for data storage, how to developing web applications, as well as how to deploy an ML application to a server.

Predicting continuous target variables with regression analysis is discussed in the next chapter. Chapter 10 focuses on another subcategory of supervised learning: regression learning, and discusses the main concepts of regression models. This chapter presents an explanatory data analysis technique to look at patterns and anomalies in data which is an important first step in predictive modeling tasks. It covers algorithms from three fundamental families of clustering that share a certain degree of similarity. Knowledge covered in this chapter enables you to organize data into logically meaningful structures.

Clustering analysis is a key technique when working with unlabeled data. Chapter 11 switch from supervised learning to unsupervised learning techniques that allows to discover hidden structures in data. This chapter explores clustering analysis which is a category of unsupervised learning, and studies how to use cluster analysis to find hidden structures in data in the absence of target variables. The popular k-means algorithm is applied to find centers of similarity.

A multilayer artificial neural network (NN) is implemented from Scratch. In Chapter 12, connections are made from multiple neurons to a powerful NN architecture to solve complex problems such as handwritten digit recognition. This chapter introduces basic concepts of artificial NNs. After learning this chapter, you will be ready for subsequent chapters which talks about deep NN (DNN) architectures. DNNs are widely use for image and text analyses.

Moving on from ML and DL, how to parallelize NN training with TensorFlow is analyzed in Chapter 13. The author focuses on TensorFlow which is one of the most popular DL libraries. With TensorFlow, programmers are able to implement NNs much more efficiently and quickly than using NumPy alone. This chapter teaches how to use the new TensorFlow 2, including its installation and low-level API. This chapter also introducing Keras, a high-level API for TensorFlow, which can be used to implement common DL conveniently.

The discussion then goes deeper to the mechanics of TensorFlow. Chapter 14 dives much deeper into TensorFlow, where they find themselves working with TensorFlow function decoration and TensorFlow estimators. This chapter introduces the key features and development of TensorFlow, and covers the migration from TensorFlow v1.x to v2. Various ways of model building using the Keras API are illustrated by considering the classic exclusive or (XOR) problem.

Deep convolutional NNs (CNNs) are employed to classify images. Chapter 15 introduces CNNs, and the concept behind CNN architectures for DL. CNNs are powerful models and have shown great performance in the field of computer vision. This chapter covers the following aspect of CNN: building blocks of CNN architectures, implementing deep CNNs in TensorFlow, data augmentation techniques. As an application, readers are guided to implement a face image-based CNN for gender detection.

Recurrent NNs (RNNs) are used to model sequential data in Chapter 16. This chapter focuses on different type of data such as structured data and images. As a warm-up exercise, this chapter introduces RNNs for predicting the sentiment of movie reviews. Long short-term memory and truncated backpropagation through time are discussed with details. A multilayer RNN is implemented for sequence modeling. Two interesting projects are accomplished to illustrate the applications of RNN: sentiment analysis and character-level language modeling.

Chapter 17 deals with the question how to make use of GANs for synthesizing new data. The author introduces a popular adversarial training regime for NNs that can be used to generate new, realistic-looking images. GANs have shown remarkable results in the computer vision community for various vision tasks such as synthesizing new images. Autoencoders, variational autoencoders, and the relationship to GANs are discussed with detailed explanations. As an application, a simple GAN is implemented to generate handwritten digits. This chapter helps readers to understanding transposed convolution and batch normalization. This chapter also covers two improved GANs: deep convolutional GANs and GANs using the Wasserstein distance.

RL for decision making in complex environments is described in the last chapter of the book. Chapter 18 explains the essential concepts in RL, starting from the very foundations how RL can support decision making in complex environments. Readers will learn about agent-environment interactions and Markov decision processes, and will understand three main approaches for solving RL problems: dynamic programming, Monte Carlo learning, and temporal difference learning.

Overall, this book is a very useful companion for machine learning beginners all the way to advanced readers. If you already know the Python language and want to start learning ML, or want to extend your knowledge, this new edition is ideally an essential resource for you.

Ren Yuan is an instructor in Shanghai Dianji University. He was born in 1984. He got his bachelor’s degree in mathematics from Jilin University in 2007, and doctor’s degree in computer software from Fudan University in 2013. His multidisciplinary research interests include image understanding, artificial intelligence, and data science.