Deep Learning with PyTorch 1.x

Second Edition

Implement deep learning techniques and neural network architecture variants using Python

Laura Mitchell, Sri. Yogesh K.
and Vishnu Subramanian

Packt
www.packt.com

Deep Learning with PyTorch 1.x
Second Edition

Implement deep learning techniques and neural network architecture variants using Python

Laura Mitchell
Sri. Yogesh K.
Vishnu Subramanian

BIRMINGHAM - MUMBAI

Deep Learning with PyTorch 1.x
Second Edition

Commissioning Editor: Sunith Shetty
Acquisition Editor: Devika Battike
Content Development Editor: Athikho Sapuni Rishana
Senior Editor: Sofi Rogers
Technical Editor: Joseph Sunil
Copy Editor: Safis Editing
Project Coordinator: Aishwarya Mohan
Proofreader: Safis Editing
Indexer: Pratik Shirodkar
Production Designer: Jyoti Chauhan

First published: February 2018
Second edition: November 2019

Production reference: 1291119

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-83855-300-5

www.packt.com

Packt.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.packt.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at customercare@packtpub.com for more details.

At www.packt.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

About the authors

Laura Mitchell graduated with a degree in mathematics from the University of Edinburgh. With 15 years of experience in the tech and data science space, Laura is the lead data scientist at MagicLab whose brands have connected the lives of over 500 million people through dating, social and business. Laura has hands-on experience in the delivery of projects surrounding natural language processing, image classification and recommender systems, from initial conception to production. She has a passion for learning new technologies and keeping herself up to date with industry trends.

Sri. Yogesh K. is an experienced data scientist with a history of working in higher education. He is skilled in Python, Apache Spark, deep learning, Hadoop, and machine learning. He is a strong engineering professional with a Certificate of Engineering Excellence from the International School of Engineering (INSOFE) and is focused on big data analytics. Sri has trained over 500 working professionals in data science and deep learning from companies including Flipkart, Honeywell, GE, and Rakuten. Additionally, he has worked on various projects that involved deep learning and PyTorch.

Vishnu Subramanian has experience in leading, architecting, and implementing several big data analytical projects using artificial intelligence, machine learning, and deep learning. He specializes in machine learning, deep learning, distributed machine learning, and visualization. He has experience in retail, finance, and travel domains. Also, he is good at understanding and coordinating between businesses, AI, and engineering teams.

About the reviewers

Mingfei Ma is a senior deep learning software engineer from Intel Asia-Pacific Research & Development Ltd and he has plenty of experience in high-performance computation. Mingfei contributed extensively to the CPU performance optimization of PyTorch and its predecessor, Torch. He also has expertise in computer graphics, heterogeneous computing, microarchitecture detection, high-performance computation libraries, and more.

Ajit Pratap Kundan is at the forefront of innovative technologies in the world of IT. He's worked with HPE, VMware, Novell, Redington, and PCS to help their customers in transforming their data centers through software-defined services. Ajit is an innovative pre-sales tech enthusiast with over 19 years of experience in technologies such as Lotus, SUSE Linux, Platespin, and all VMware solutions. Ajit is a valued author on cloud technologies and has authored two books, *VMware Cross-Cloud Architecture* and *Intelligent Automation with VMware*, published by Packt.

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Table of Contents

Section 3: Understanding Modern Architectures in Deep Learning

Preface

PyTorch is grabbing the attention of deep learning researchers and data science professionals due to its accessibility, efficiency, and the fact of it being more native to the Python way of development. This book will get you up and running with one of the most cutting-edge deep learning libraries—PyTorch.

In this second edition, you'll learn about the various fundamental building blocks that power modern deep learning using the new features and offerings of the PyTorch 1.x library. You will learn how to solve real-world problems using **convolutional neural networks** (**CNNs**), **recurrent neural networks** (**RNNs**), and **long short-term memory** (**LSTM**) networks. You will then get to grips with the concepts of various state-of-the-art modern deep learning architectures, such as ResNet, DenseNet, and Inception. You will learn how to apply neural networks to various domains, such as computer vision, **natural language processing** (**NLP**), and more. You will see how to build, train, and scale a model with PyTorch and dive into complex neural networks such as generative networks and autoencoders for producing text and images. Furthermore, you will learn about GPU computing and how GPUs can be used to perform heavy computations. Lastly, you will learn how to work with deep learning-based architectures for transfer learning and reinforcement learning problems.

By the end of the book, you'll be able to implement deep learning applications in PyTorch with ease.

Who this book is for

This book is for data scientists and machine learning engineers who are looking to explore deep learning algorithms using PyTorch 1.x. Those who wish to migrate to PyTorch 1.x will find this book insightful. To make the most out of this book, working knowledge of Python programming and some knowledge of machine learning will be helpful.

What this book covers

`Chapter 1`, *Getting Started with Deep Learning Using PyTorch*, introduces you to the history of deep learning, machine learning, and AI. This chapter covers how they are related to neuroscience and other areas of science, such as statistics, information theory, probability, and linear algebra.

`Chapter 2`, *Building Blocks of Neural Networks*, covers the various math concepts that are required to understand and appreciate neural networks using PyTorch.

`Chapter 3`, *Diving Deep into Neural Networks*, shows you how to apply neural networks to various real-world scenarios.

`Chapter 4`, *Deep Learning for Computer Vision*, covers the various building blocks of modern CNN architectures.

`Chapter 5`, *Natural Language Processing with Sequence Data*, shows you how to handle sequence data, particularly text data, and teaches you how to create a network model.

`Chapter 6`, *Implementing Autoencoders*, introduces the idea of semi-supervised learning algorithms through an introduction of autoencoders. It also covers how to use restricted Boltzmann machines to understand the probability distribution of data.

`Chapter 7`, *Working with Generative Adversarial Networks*, shows you how to build generative models capable of producing text and images.

`Chapter 8`, *Transfer Learning with Modern Network Architectures*, covers modern architectures such as ResNet, Inception, DenseNet, and Seq2Seq, and also shows you how to use pre-trained weights for transfer learning.

`Chapter 9`, *Deep Reinforcement Learning*, starts with a basic introduction to reinforcement learning, including coverage of agents, state, action, reward, and policy. It also contains hands-on code for deep learning-based architectures for reinforcement learning problems, such as Deep Q networks, policy gradient methods, and actor-critic models.

`Chapter 10`, *What Next?*, gives you a quick overview of what the book covered along with information on how you can keep up to date with the latest advances in the field.

To get the most out of this book

Working knowledge of Python will be useful.

Download the example code files

You can download the example code files for this book from your account at `www.packt.com`. If you purchased this book elsewhere, you can visit `www.packtpub.com/support` and register to have the files emailed directly to you.

You can download the code files by following these steps:

1. Log in or register at `www.packt.com`.
2. Select the **Support** tab.
3. Click on **Code Downloads**.
4. Enter the name of the book in the **Search** box and follow the onscreen instructions.

Once the file is downloaded, please make sure that you unzip or extract the folder using the latest version of:

- WinRAR/7-Zip for Windows
- Zipeg/iZip/UnRarX for Mac
- 7-Zip/PeaZip for Linux

The code bundle for the book is also hosted on GitHub at `https://github.com/PacktPublishing/Deep-Learning-with-PyTorch-1.x`. In case there's an update to the code, it will be updated on the existing GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at `https://github.com/PacktPublishing/`. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: `http://www.packtpub.com/sites/default/files/downloads/9781838553005_ColorImages.pdf`.

Conventions used

There are a number of text conventions used throughout this book.

`CodeInText`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "Let's use simple Python functions, such as `split` and `list`, to convert the text into tokens."

A block of code is set as follows:

```
toy_story_review = "Just perfect. Script, character, animation....this
manages to break free of the yoke of 'children's movie' to simply be one of
the best movies of the 90's, full-stop."

print(list(toy_story_review))
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
['J', 'u', 's', 't', ' ', 'p', 'e', 'r', 'f', 'e', 'c', 't', '.', ' ', 'S',
'c', 'r', 'i', 'p', 't', ',', ' ', 'c', 'h', 'a', 'r', 'a', 'c', 't', 'e',
'r', ',', ' ', 'a', 'n', 'i', 'm', 'a', 't', 'i', 'o', 'n', '.', '.', '.',
'.', 't', 'h', 'i', 's', ' ', 'm', 'a', 'n', 'a', 'g', 'e', 's', ' ', 't',
'o', ' ', 'b', 'r', 'e', 'a', 'k', ' ', 'f', 'r', 'e', 'e', ' ', 'o', 'f',
' ', 't', 'h', 'e', ' ', 'y', 'o', 'k', 'e', ' ', 'o', 'f', ' ', "'", 'c',
'h', 'i', 'l', 'd', 'r', 'e', 'n', "'", 's', ' ', 'm', 'o', 'v', 'i', 'e',
"'", ' ', 't', 'o', ' ', 's', 'i', 'm', 'p', 'l', 'y', ' ', 'b', 'e', ' ',
'o', 'n', 'e', ' ', 'o', 'f', ' ', 't', 'h', 'e', ' ', 'b', 'e', 's', 't',
' ', 'm', 'o', 'v', 'i', 'e', 's', ' ', 'o', 'f', ' ', 't', 'h', 'e', ' ',
'9', '0', "'", 's', ',', ' ', 'f', 'u', 'l', 'l', '-', 's', 't', 'o', 'p',
'.']
```

Any command-line input or output is written as follows:

```
pip install torchtext
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For example, words in menus or dialog boxes appear in the text like this. Here is an example: "We will be helping you to understand **recurrent neural networks** (**RNNs**). "

Warnings or important notes appear like this.

Tips and tricks appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, mention the book title in the subject of your message and email us at `customercare@packtpub.com`.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/support/errata`, selecting your book, clicking on the Errata Submission Form link, and entering the details.

Piracy: If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `authors.packtpub.com`.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit `packt.com`.

1

Section 1: Building Blocks of Deep Learning with PyTorch 1.x

In this section, you will be introduced to the concepts of deep learning and the various deep learning frameworks.

This section contains the following chapters:

- `Chapter 1`, *Getting Started with Deep Learning Using PyTorch*
- `Chapter 2`, *Building Blocks of Neural Networks*

1
Getting Started with Deep Learning Using PyTorch

Deep learning (**DL**) has revolutionized industry after industry. It was once famously described by Andrew Ng on Twitter as follows:

> *"Artificial intelligence is the new electricity!"*

Electricity transformed countless industries; now, **artificial intelligence** (**AI**) will do the same.

AI and DL are used as synonyms, but there are substantial differences between the two. Let's demystify the terminology that's used in the industry so that you, as a practitioner, will be able to differentiate between signal and noise.

In this chapter, we will cover the following different parts of AI:

- Exploring artificial intelligence
- Machine learning in the real world
- Applications of deep learning
- Deep learning frameworks
- Setting up PyTorch 1.x

Exploring artificial intelligence

Countless articles discussing AI are published every day. The trend has increased in the last 2 years. There are several definitions of AI floating around the web, with my favorite being *the automation of intellectual tasks normally performed by humans*.

The history of AI

Since you've picked up this book, you may be well aware of the recent hype in AI. But it all started when John McCarthy, then a young assistant professor at Dartmouth, coined the term *artificial intelligence* in 1995, which he defined as a field pertaining to the science and engineering of intelligent machines. This kick-started the first wave of AI, which was primarily driven by symbolic reasoning; its outcomes were astonishing, to say the least. AI that was developed during this time was capable of reading and solving high-school Algebra problems [STUDENT], proving theorems in Geometry [SAINT], and learning the English language [SHRDLU]. Symbolic reasoning is the use of complex rules nested in if-then statements.

The most promising work in this era, though, was the perceptron, which was introduced in 1958 by Frank Rosenblatt. The perceptron, when combined with intelligent optimization techniques that were discovered later, laid the foundations for deep learning as we know it today.

It wasn't plain sailing for AI, though, since the funding in the field significantly reduced during lean periods, mostly due to overpromising initial discoveries and, as we were yet to discover, a lack of data and compute power. The rise in prominence of **machine learning** (**ML**) in the early nineties bucked the trend and created significant interest in the field. First, we need to understand the paradigm of ML and its relationship with DL.

Machine learning in the real world

ML is a subfield of AI that uses algorithms and statistical techniques to perform a task without the use of any explicit instructions. Instead, it relies on underlying statistical patterns in the data.

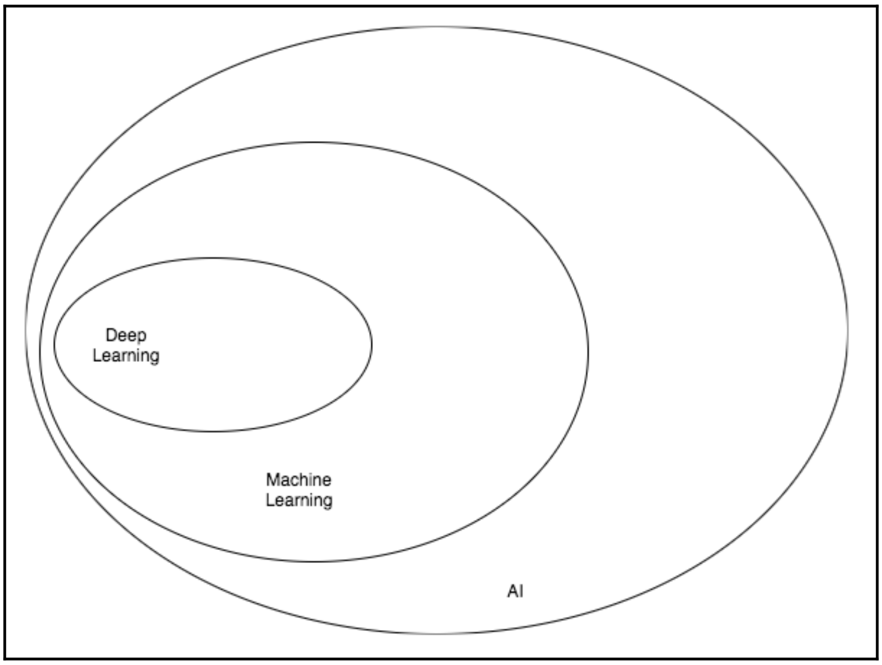
To build successful machine learning models, we need to provide ML algorithms with labeled data. The success of this approach was heavily dependent on the available data and compute power so that large amounts of data could be used.

So, why DL?

Most ML algorithms perform well on structured data, such as sales predictions, recommendation systems, and marketing personalization. An important factor for any ML algorithm is feature engineering and data scientists need to spend a lot of time exploring possible features with high predictive power for ML algorithms. In certain domains, such as computer vision and **natural language processing** (**NLP**), feature engineering is challenging as features that are important for one task may not hold up well for other tasks. This is where DL excels—the algorithm itself engineers features in a non-linear space so that they are important for a particular task.

Traditional ML algorithms still outperform DL methods when there is a paucity of data, but as data increases, the performance of traditional machine learning algorithms tends to plateau and deep learning algorithms tend to significantly outperform other learning strategies.

The following diagram shows the relationship DL has with ML and AI:



To summarize this, DL is a subfield of machine learning; feature engineering is where the algorithm non-linearly explores its space.