

Mastering PyTorch

Build powerful neural network architectures
using advanced PyTorch 1.x features

Ashish Ranjan Jha

Foreword by Dr. Gopinath Pillai, Head of Department, Electrical Engineering, IIT Roorkee



Mastering PyTorch

Build powerful neural network architectures using advanced PyTorch 1.x features

Ashish Ranjan Jha

Packt

BIRMINGHAM—MUMBAI

Mastering PyTorch

Copyright © 2021 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Group Product Manager: Kunal Parikh

Publishing Product Manager: Devika Battike

Senior Editor: Roshan Kumar

Content Development Editor: Tazeen Shaikh

Technical Editor: Sonam Pandey

Copy Editor: Safis Editing

Project Coordinator: Aishwarya Mohan

Proofreader: Safis Editing

Indexer: Priyanka Dhadke

Production Designer: Nilesh Mohite

First published: February 2021

Production reference: 1140121

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham

B3 2PB, UK.

ISBN 978-1-78961-438-1

www.packt.com

*To my mother and best-friend - Rani Jha, my father and idol - Bibhuti Bhushan Jha,
for their sacrifices and constant support and for being the driving forces of my life and career.
Without their love, none of this would matter. To my sisters, Sushmita, Nivedita, and Shalini,
for teaching me what and what not to do in life.*



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 [packt . com](http://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](mailto:customercare@packtpub.com) for more details.

At [www . packt . com](http://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.

Foreword

I am happy to know that Ashish, who was my student on the artificial neural networks course 8 years ago at IIT Roorkee, has now authored this hands-on book that covers a range of deep learning topics in reasonable depth.

Learning by coding is something every deep learning enthusiast wants to undertake, but tends to leave half way through. The effort needed to go through documentation and extract useful information to run deep learning projects is cumbersome. I have seen far too many students become frustrated during the process. There are tons of resources available for any beginner to become an expert. However, it is easy for any beginner to lose sight of the learning task while trying to strike a balance between concept-oriented courses and the coding-savvy approach of many academic programs.

PyTorch is uniquely placed as being pythonic and very flexible. It is appealing both to beginners who have just started coding machine learning models and to experts who like to meddle in the finer parameters of model designing and training. PyTorch is one library I am happy to recommend to any enthusiast, regardless of their level of expertise.

The best way to learn machine learning and deep learning models is by practicing coding in PyTorch. This book navigates the world of deep learning through PyTorch in a very engaging way. It starts from the basic building blocks of deep learning. The visual appeal of learning the data pipeline is one of its strong points. The PyTorch modules used for model building and training are introduced in the simplest of ways. Any student will appreciate the hands-on approach of this book. Every concept is explained through codes, and every step of the code is well documented. It should not be assumed that this book is just for beginners. Instead, any beginner can become an expert by following this book.

Starting from basic model building, such as the popular VGG16 or ResNet, to advanced topics, such as AutoML and distributed learning, all these aspects are covered here. The book further encompasses concepts such as AI explainability, deep reinforcement learning, and GANs. The exercises in this book range from building an image captioning model to music generation and neural style transfer models, as well as building PyTorch model servers in production systems. This helps you to prepare for any niche deep learning ventures.

I recommend this book to anyone who wants to master PyTorch for deploying deep learning models with the latest libraries.

Dr. Gopinath Pillai
Head Of Department, Electrical Engineering, IIT Roorkee

Contributors

About the author

Ashish Ranjan Jha received his bachelor's degree in electrical engineering from IIT Roorkee (India), his master's degree in computer science from EPFL (Switzerland), and an MBA degree from the Quantic School of Business (Washington). He received distinctions in all of his degrees. He has worked for a variety of tech companies, including Oracle and Sony, and tech start-ups, such as Revolut, as a machine learning engineer.

Aside from his years of work experience, Ashish is a freelance ML consultant, an author, and a blogger (datashines). He has worked on products/projects ranging from using sensor data for predicting vehicle types to detecting fraud in insurance claims. In his spare time, Ashish works on open source ML projects and is active on StackOverflow and kaggle (arj7192).

About the reviewer

Javier Abascal Carrasco has a master's degree in telecommunication engineering from the University of Seville (Spain). He also studied abroad at TU Dresden (Germany) and Thomas College (ME, USA), where he obtained his MBA. Since his career started, Javier has been passionate about the world of data and analytics. He has had the chance to work with and help all manner of companies, ranging from small start-ups to big corporations, including the consulting firm EY and Facebook. In addition, for the last 3 years, he has been a part-time lecturer on the data science space. He truly believes that PyTorch is bringing a new, fresh style to programming and work involving deep learning, generating a friendly competitor landscape in relation to TensorFlow.

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

Preface

Section 1: PyTorch Overview

1

Overview of Deep Learning using PyTorch

Technical requirements	17	PyTorch modules	33
A refresher on deep learning	18	Tensor modules	35
Activation functions	25	Training a neural network using PyTorch	40
Optimization schedule	28		
Exploring the PyTorch library	32	Summary	48

2

Combining CNNs and LSTMs

Technical requirements	50	Preprocessing caption (text) data	55
Building a neural network with CNNs and LSTMs	50	Preprocessing image data	58
		Defining the image captioning data loader	59
Text encoding demo	51	Defining the CNN-LSTM model	61
Building an image caption generator using PyTorch	52	Training the CNN-LSTM model	65
		Generating image captions using the trained model	69
Downloading the image captioning datasets	53	Summary	71

Section 2: Working with Advanced Neural Network Architectures

3

Deep CNN Architectures

Technical requirements	76	Exploring GoogLeNet and Inception v3	108
Why are CNNs so powerful?	76	Inception modules	109
Evolution of CNN architectures	80	1x1 convolutions	111
Developing LeNet from scratch	82	Global average pooling	111
Using PyTorch to build LeNet	84	Auxiliary classifiers	112
Training LeNet	90	Inception v3	113
Testing LeNet	91	Discussing ResNet and DenseNet architectures	115
Fine-tuning the AlexNet model	94	DenseNet	118
Using PyTorch to fine-tune AlexNet	97	Understanding EfficientNets and the future of CNN architectures	122
Running a pre-trained VGG model	104	Summary	124

4

Deep Recurrent Model Architectures

Technical requirements	129	Gated orthogonal recurrent units	138
Exploring the evolution of recurrent networks	129	Training RNNs for sentiment analysis	138
Types of recurrent neural networks	129	Loading and preprocessing the text dataset	139
RNNs	132	Instantiating and training the model	145
Bidirectional RNNs	133	Building a bidirectional LSTM	151
LSTMs	134	Loading and preprocessing text dataset	151
Extended and bidirectional LSTMs	135		
Multi-dimensional RNNs	136		
Stacked LSTMs	136		
GRUs	137		
Grid LSTMs	138		

Instantiating and training the LSTM model	152	GRUs and PyTorch	155
		Attention-based models	156
Discussing GRUs and attention-based models	155	Summary	158

5

Hybrid Advanced Models

Technical requirements	160	model from scratch	173
Building a transformer model for language modeling	160	Understanding RandWireNNs	174
Reviewing language modeling	161	Developing RandWireNNs using PyTorch	174
Understanding the transformer model architecture	161	Summary	187

Developing a RandWireNN

Section 3: Generative Models and Deep Reinforcement Learning

6

Music and Text Generation with PyTorch

Technical requirements	192	Out-of-the-box text generation with GPT-2	196
Building a transformer-based text generator with PyTorch	193	Text generation strategies using PyTorch	197
Training the transformer-based language model	193	Generating MIDI music with LSTMs using PyTorch	204
Saving and loading the language model	194	Loading the MIDI music data	205
Using the language model to generate text	194	Defining the LSTM model and training routine	209
Using a pre-trained GPT-2 model as a text generator	196	Training and testing the music generation model	211
		Summary	215

7

Neural Style Transfer

Technical requirements	218	Loading and trimming the pre-trained VGG19 model	224
Understanding how to transfer style between images	218	Building the neural style transfer model	227
Implementing neural style transfer using PyTorch	222	Training the style transfer model	228
Loading the content and style images	222	Experimenting with the style transfer system	232
		Summary	236

8

Deep Convolutional GANs

Technical requirements	240	Defining the generator	244
Defining the generator and discriminator networks	241	Defining the discriminator	247
Understanding the DCGAN generator and discriminator	242	Loading the image dataset	248
Training a DCGAN using PyTorch	244	Training loops for DCGANs	249
		Using GANs for style transfer	254
		Understanding the pix2pix architecture	255
		Summary	263

9

Deep Reinforcement Learning

Technical requirements	266	Building a DQN model in PyTorch	279
Reviewing reinforcement learning concepts	267	Initializing the main and target CNN models	279
Types of reinforcement learning algorithms	269	Defining the experience replay buffer	282
Discussing Q-learning	272	Setting up the environment	283
Understanding deep Q-learning	277	Defining the CNN optimization function	284
Using two separate DNNs	277	Managing and running episodes	286
Experience replay buffer	278	Training the DQN model to learn Pong	289
		Summary	294

Section 4: PyTorch in Production Systems

10

Operationalizing PyTorch Models into Production

Technical requirements	298	Understanding the utility of TorchScript	325
Model serving in PyTorch	299	Model tracing with TorchScript	326
Creating a PyTorch model inference pipeline	299	Model scripting with TorchScript	330
Building a basic model server	306	Running a PyTorch model in C++	332
Creating a model microservice	313	Using ONNX to export PyTorch models	337
Serving a PyTorch model using TorchServe	319	Serving PyTorch models in the cloud	340
Installing TorchServe	320	Using PyTorch with AWS	340
Launching and using a TorchServe server	320	Serving PyTorch model on Google Cloud	344
Exporting universal PyTorch models using TorchScript and ONNX	324	Serving PyTorch models with Azure	346
		Summary	348
		References	349

11

Distributed Training

Technical requirements	352	Training the MNIST model in a distributed fashion	356
Distributed training with PyTorch	352	Distributed training on GPUs with CUDA	364
Training the MNIST model in a regular fashion	353	Summary	367

12

PyTorch and AutoML

Technical requirements	370	Using Auto-PyTorch for optimal MNIST model search	371
Finding the best neural architectures with AutoML	370	Using Optuna for hyperparameter search	379

Defining the model architecture and loading dataset	380	Running Optuna's hyperparameter search	384
Defining the model training routine and optimization schedule	383	Summary	387

13

PyTorch and Explainable AI

Technical requirements	390	Visualizing the feature maps of the model	396
Model interpretability in PyTorch	390	Using Captum to interpret models	399
Training the handwritten digits classifier – a recap	391	Setting up Captum	399
Visualizing the convolutional filters of the model	393	Exploring Captum's interpretability tools	400
		Summary	405

14

Rapid Prototyping with PyTorch

Technical requirements	408	Training models on any hardware using PyTorch Lightning	416
Using fast.ai to set up model training in a few minutes	409	Defining the model components in PyTorch Lightning	416
Setting up fast.ai and loading data	409	Training and evaluating the model using PyTorch Lightning	418
Training a MNIST model using fast.ai	412		
Evaluating and interpreting the model using fast.ai	414	Summary	422

Other Books You May Enjoy

Index
