EXPERT INSIGHT

Deep Reinforcement Learning Hands-On

Apply modern RL methods to practical problems of chatbots, robotics, discrete optimization, web automation, and more

Second edition – Includes multi-agent methods and advanced exploration techniques

Maxim Lapan

Packt

Deep Reinforcement Learning Hands-On

Second Edition

Apply modern RL methods to practical problems of chatbots, robotics, discrete optimization, web automation, and more

Maxim Lapan

Packt

BIRMINGHAM - MUMBAI

Deep Reinforcement Learning Hands-On

Second Edition

Copyright © 2020 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.

Producer: Jonathan Malysiak

Acquisition Editor – Peer Reviews: Suresh Jain

Content Development Editors: Joanne Lovell and Chris Nelson

Technical Editor: Saby D'silva

Project Editor: Kishor Rit

Proofreader: Safis Editing

Indexer: Rekha Nair

Presentation Designer: Sandip Tadge

First published: June 2018

Second edition: January 2020

Production reference: 1300120

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

ISBN 978-1-83882-699-4

`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
- Learn better 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.

Contributors

About the authors

Maxim Lapan is a deep learning enthusiast and independent researcher. His background and 15 years' work expertise as a software developer and a systems architect covers everything from low-level Linux kernel driver development to performance optimization and the design of distributed applications working on thousands of servers. With extensive work experience in big data, machine learning, and large parallel distributed HPC and non-HPC systems, he has the ability to explain complicated things using simple words and vivid examples. His current areas of interest surround the practical applications of deep learning, such as deep natural language processing and deep reinforcement learning.

Maxim lives in Moscow, Russia, with his family.

I'd like to thank my family: my wife, Olga, and my children, Ksenia, Julia, and Fedor, for their patience and support. It was a challenging time writing this book and it wouldn't have been possible without you, so thanks! Julia and Fedor did a great job of gathering samples for MiniWoB (*Chapter 16*, *Web Navigation*) and testing the Connect 4 agent's playing skills (*Chapter 23*, *AlphaGo Zero*).

About the reviewers

Mikhail Yurushkin holds a PhD. His areas of research are high-performance computing and optimizing compiler development. Mikhail is a senior lecturer at SFEDU university, Rostov-on-Don, Russia. He teaches advanced deep learning courses on computer vision and NLP. Mikhail has worked for over eight years in cross-platform native C++ development, machine learning, and deep learning. He is an entrepreneur and founder of several technological start-ups, including BroutonLab – Data Science Company, which specializes in the development of AI-powered software products.

Per-Arne Andersen is a PhD student in deep reinforcement learning at the University of Agder, Norway. He has authored several technical papers on reinforcement learning for games and received the best student award from the British Computer Society for his research into model-based reinforcement learning. Per-Arne is also an expert on network security, having worked in the field since 2012. His current research interests include machine learning, deep learning, network security, and reinforcement learning.

Sergey Kolesnikov is an industrial and academic research engineer with over five years' experience in machine learning, deep learning, and reinforcement learning. He's currently working on industrial applications that deal with CV, NLP, and RecSys, and is involved in reinforcement learning academic research. He is also interested in sequential decision making and psychology. Sergey is a NeurIPS competition winner and an open source evangelist. He is also the creator of Catalyst – a high-level PyTorch ecosystem for accelerated deep learning/reinforcement learning research and development.

Table of Contents

Preface **xiii**

Chapter 1: What Is Reinforcement Learning? **1**

Supervised learning **2**
Unsupervised learning **2**
Reinforcement learning **3**
RL's complications **4**
RL formalisms **5**
Reward 6
The agent 8
The environment 9
Actions 9
Observations 9
The theoretical foundations of RL **12**
Markov decision processes 12
The Markov process 13
Markov reward processes 17
Adding actions 20
Policy 23
Summary **24**

Chapter 2: OpenAI Gym **25**

The anatomy of the agent **25**
Hardware and software requirements **28**
The OpenAI Gym API **30**
The action space 31
The observation space 31
The environment 33
Creating an environment 35
The CartPole session 37

The random CartPole agent **40**
Extra Gym functionality – wrappers and monitors **41**
Wrappers 41
Monitor 45
Summary **48**
Chapter 3: Deep Learning with PyTorch 49
Tensors **50**
The creation of tensors 50
Scalar tensors 53
Tensor operations 53
GPU tensors 53
Gradients **55**
Tensors and gradients 56
NN building blocks **59**
Custom layers **61**
The final glue – loss functions and optimizers **64**
Loss functions 64
Optimizers 65
Monitoring with TensorBoard **67**
TensorBoard 101 68
Plotting stuff 69
Example – GAN on Atari images **71**
PyTorch Ignite **76**
Ignite concepts 77
Summary **81**
Chapter 4: The Cross-Entropy Method 83
The taxonomy of RL methods **84**
The cross-entropy method in practice **85**
The cross-entropy method on CartPole **87**
The cross-entropy method on FrozenLake **96**
The theoretical background of the cross-entropy method **104**
Summary **106**
Chapter 5: Tabular Learning and the Bellman Equation 107
Value, state, and optimality **107**
The Bellman equation of optimality **110**
The value of the action **112**
The value iteration method **115**
Value iteration in practice **117**
Q-learning for FrozenLake **124**
Summary **126**

Chapter 6: Deep Q-Networks 127

Real-life value iteration 127

Tabular Q-learning 129

Deep Q-learning 134

Interaction with the environment 135

SGD optimization 136

Correlation between steps 137

The Markov property 137

The final form of DQN training 138

DQN on Pong 139

Wrappers 140

The DQN model 145

Training 147

Running and performance 156

Your model in action 159

Things to try 161

Summary 162

Chapter 7: Higher-Level RL Libraries 163

Why RL libraries? 163

The PTAN library 164

Action selectors 166

The agent 167

DQNAgent 169

PolicyAgent 170

Experience source 171

Toy environment 172

The ExperienceSource class 173

ExperienceSourceFirstLast 176

Experience replay buffers 177

The TargetNet class 179

Ignite helpers 181

The PTAN CartPole solver 182

Other RL libraries 184

Summary 185

Chapter 8: DQN Extensions 187

Basic DQN 188

Common library 188

Implementation 193

Results 195

N-step DQN 197

Implementation 199

Results 200

Double DQN 201

Implementation 202

Results 204

Noisy networks 205

Implementation 206

Results 208

Prioritized replay buffer 210

Implementation 211

Results 215

Dueling DQN 216

Implementation 218

Results 219

Categorical DQN 220

Implementation 223

Results 229

Combining everything 232

Results 232

Summary 234

References 235

Chapter 9: Ways to Speed up RL 237

Why speed matters 237

The baseline 240

The computation graph in PyTorch 242

Several environments 245

Play and train in separate processes 247

Tweaking wrappers 252

Benchmark summary 257

Going hardcore: CuLE 257

Summary 258

References 258

Chapter 10: Stocks Trading Using RL 259

Trading 259

Data 260

Problem statements and key decisions 261

The trading environment 263

Models 271

Training code 273

Results 273

The feed-forward model 273

The convolution model 280
Things to try **281**
Summary **282**
Chapter 11: Policy Gradients – an Alternative **283**
Values and policy **283**
Why the policy? 284
Policy representation 285
Policy gradients 286
The REINFORCE method **286**
The CartPole example 288
Results 291
Policy-based versus value-based methods 293
REINFORCE issues **294**
Full episodes are required 294
High gradients variance 295
Exploration 295
Correlation between samples 296
Policy gradient methods on CartPole **296**
Implementation 296
Results 300
Policy gradient methods on Pong **304**
Implementation 305
Results 306
Summary **308**
Chapter 12: The Actor-Critic Method **309**
Variance reduction **309**
CartPole variance **311**
Actor-critic **315**
A2C on Pong **317**
A2C on Pong results **323**
Tuning hyperparameters **327**
Learning rate 327
Entropy beta 328
Count of environments 328
Batch size 329
Summary **329**
Chapter 13: Asynchronous Advantage Actor-Critic **331**
Correlation and sample efficiency **331**
Adding an extra A to A2C **333**
Multiprocessing in Python **336**

A3C with data parallelism **336**
Implementation 336
Results 343
A3C with gradients parallelism **344**
Implementation 345
Results 350
Summary **352**
Chapter 14: Training Chatbots with RL **353**
An overview of chatbots **353**
Chatbot training **354**
The deep NLP basics **356**
RNNs 356
Word embedding 358
The Encoder-Decoder architecture 359
Seq2seq training **360**
Log-likelihood training 360
The bilingual evaluation understudy (BLEU) score 362
RL in seq2seq 363
Self-critical sequence training 364
Chatbot example **365**
The example structure 366
Modules: cornell.py and data.py 367
BLEU score and utils.py 368
Model 369
Dataset exploration **376**
Training: cross-entropy **378**
Implementation 378
Results 383
Training: SCST **385**
Implementation 386
Results 393
Models tested on data **396**
Telegram bot **398**
Summary **402**
Chapter 15: The TextWorld Environment **403**
Interactive fiction **403**
The environment **407**
Installation 407
Game generation 408
Observation and action spaces 410

Extra game information 412
Baseline DQN **415**
Observation preprocessing 417
Embeddings and encoders 422
The DQN model and the agent 425
Training code 427
Training results 428
The command generation model **433**
Implementation 435
Pretraining results 440
DQN training code 442
The result of DQN training 443
Summary **445**
Chapter 16: Web Navigation **447**
Web navigation **447**
Browser automation and RL 448
The MiniWoB benchmark 449
OpenAI Universe **451**
Installation 452
Actions and observations 453
Environment creation 454
MiniWoB stability 456
The simple clicking approach **456**
Grid actions 457
Example overview 459
The model 459
The training code 460
Starting containers 465
The training process 467
Checking the learned policy 470
Issues with simple clicking 471
Human demonstrations **473**
Recording the demonstrations 474
The recording format 477
Training using demonstrations 480
Results 481
The tic-tac-toe problem 486
Adding text descriptions **489**
Implementation 490
Results 495

Things to try **498**
Summary **499**
Chapter 17: Continuous Action Space **501**
Why a continuous space? **501**
The action space 502
Environments 503
The A2C method **505**
Implementation 506
Results 510
Using models and recording videos 512
Deterministic policy gradients **512**
Exploration 514
Implementation 515
Results 520
Recording videos 522
Distributional policy gradients **522**
Architecture 523
Implementation 524
Results 528
Video recordings 530
Things to try **530**
Summary **530**
Chapter 18: RL in Robotics **531**
Robots and robotics **531**
Robot complexities 534
The hardware overview 535
The platform 536
The sensors 537
The actuators 539
The frame 540
The first training objective **544**
The emulator and the model **546**
The model definition file 548
The robot class 552
DDPG training and results **558**
Controlling the hardware **561**
MicroPython 561
Dealing with sensors 565
The I^{2}C bus 566
Sensor initialization and reading 569
Sensor classes and timer reading 573