

EXPERT INSIGHT

Deep Reinforcement Learning with Python

Master classic RL, deep RL, distributional RL, inverse RL,
and more with OpenAI Gym and TensorFlow

Second Edition



Sudharsan Ravichandiran

Packt>

Deep Reinforcement Learning with Python

Second Edition

Master classic RL, deep RL, distributional RL, inverse RL, and more with OpenAI Gym and TensorFlow

Sudharsan Ravichandiran



BIRMINGHAM - MUMBAI

Deep Reinforcement Learning with Python

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.

Producers: Ben Renow-Clarke and Aarthi Kumaraswamy

Acquisition Editor – Peer Reviews: Divya Mudaliar

Content Development Editor: Bhavesh Amin

Technical Editor: Aniket Shetty

Project Editor: Janice Gonsalves

Copy Editor: Safis Editing

Proofreader: Safis Editing

Indexer: Pratik Shirodkar

Presentation Designer: Pranit Padwal

First published: June 2018

Second edition: September 2020

Production reference: 1300920

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham B3 2PB, UK.

ISBN 978-1-83921-068-6

www.packt.com

To my adorable mom, Kasthuri, and to my beloved dad, Ravichandiran.

- Sudharsan Ravichandiran



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 author

Sudharsan Ravichandiran is a data scientist, researcher, best-selling author, and YouTuber (search for *Sudharsan reinforcement learning*). He completed his bachelor's in information technology at Anna University. His area of research focuses on practical implementations of deep learning and reinforcement learning, which includes natural language processing and computer vision. He is an open source contributor and loves answering questions on Stack Overflow. He also authored a best-seller, *Hands-On Reinforcement Learning with Python, 1st edition*, published by Packt Publishing.

I would like to thank my most amazing parents and my brother, Karthikeyan, for inspiring and motivating me. My huge thanks to the producer of the book, Aarthi, and the editors, Bhavesh, Aniket, and Janice. Special thanks to the reviewers, Sujit Pal and Valerii Babushkin, for providing their valuable insights and feedback. Without all their support, it would have been impossible to complete this book.

About the reviewers

Sujit Pal is a Technology Research Director at Elsevier Labs, an advanced technology group within the Reed-Elsevier Group of companies. His areas of interests include semantic search, natural language processing, machine learning, and deep learning. At Elsevier, he has worked on several initiatives involving search quality measurement and improvement, image classification and duplicate detection, and annotation and ontology development for medical and scientific corpora. He has co-authored a book on deep learning and writes about technology on his blog *Salmon Run*.

Valerii Babushkin is the senior director of data science at X5 Retail Group, where he leads a team of 100+ people in the area of natural language processing, machine learning, computer vision, data analysis, and A/B testing. Valerii is a Kaggle competitions Grand Master, ranking globally in the top 30. He studied cybernetics at Moscow Polytechnical University and mechatronics at Karlsruhe University of Applied Sciences and has worked with Packt as an author of the *Python Machine Learning Tips, Tricks, and Techniques* course and a technical reviewer for some books on reinforcement learning.

Table of Contents

Preface	xv
Chapter 1: Fundamentals of Reinforcement Learning	1
Key elements of RL	2
Agent	2
Environment	2
State and action	2
Reward	3
The basic idea of RL	3
The RL algorithm	5
RL agent in the grid world	5
How RL differs from other ML paradigms	9
Markov Decision Processes	10
The Markov property and Markov chain	11
The Markov Reward Process	13
The Markov Decision Process	13
Fundamental concepts of RL	16
Math essentials	16
Expectation	16
Action space	18
Policy	19
Deterministic policy	20
Stochastic policy	20
Episode	23
Episodic and continuous tasks	25
Horizon	25
Return and discount factor	26
Small discount factor	27

Large discount factor	28
What happens when we set the discount factor to 0?	28
What happens when we set the discount factor to 1?	29
The value function	29
Q function	33
Model-based and model-free learning	35
Different types of environments	36
Deterministic and stochastic environments	36
Discrete and continuous environments	37
Episodic and non-episodic environments	38
Single and multi-agent environments	38
Applications of RL	38
RL glossary	39
Summary	41
Questions	41
Further reading	42
Chapter 2: A Guide to the Gym Toolkit	43
Setting up our machine	44
Installing Anaconda	44
Installing the Gym toolkit	45
Common error fixes	46
Creating our first Gym environment	47
Exploring the environment	50
States	50
Actions	51
Transition probability and reward function	52
Generating an episode in the Gym environment	56
Action selection	56
Generating an episode	58
More Gym environments	62
Classic control environments	62
State space	64
Action space	66
Cart-Pole balancing with random policy	67
Atari game environments	69
General environment	70
Deterministic environment	70
No frame skipping	71
State and action space	71
An agent playing the Tennis game	75
Recording the game	77
Other environments	79
Box2D	79
MuJoCo	80
Robotics	80

Toy text	81
Algorithms	81
Environment synopsis	82
Summary	82
Questions	83
Further reading	83
Chapter 3: The Bellman Equation and Dynamic Programming	85
The Bellman equation	86
The Bellman equation of the value function	86
The Bellman equation of the Q function	90
The Bellman optimality equation	93
The relationship between the value and Q functions	95
Dynamic programming	97
Value iteration	97
The value iteration algorithm	99
Solving the Frozen Lake problem with value iteration	107
Policy iteration	115
Algorithm – policy iteration	118
Solving the Frozen Lake problem with policy iteration	125
Is DP applicable to all environments?	129
Summary	130
Questions	131
Chapter 4: Monte Carlo Methods	133
Understanding the Monte Carlo method	134
Prediction and control tasks	135
Prediction task	135
Control task	135
Monte Carlo prediction	136
MC prediction algorithm	140
Types of MC prediction	144
First-visit Monte Carlo	145
Every-visit Monte Carlo	146
Implementing the Monte Carlo prediction method	147
Understanding the blackjack game	147
The blackjack environment in the Gym library	158
Every-visit MC prediction with the blackjack game	160
First-visit MC prediction with the blackjack game	166
Incremental mean updates	167
MC prediction (Q function)	168
Monte Carlo control	170
MC control algorithm	172
On-policy Monte Carlo control	174

Monte Carlo exploring starts	174
Monte Carlo with the epsilon-greedy policy	176
Implementing on-policy MC control	179
Off-policy Monte Carlo control	184
Is the MC method applicable to all tasks?	188
Summary	189
Questions	190
Chapter 5: Understanding Temporal Difference Learning	191
TD learning	192
TD prediction	193
TD prediction algorithm	196
Predicting the value of states in the Frozen Lake environment	202
TD control	206
On-policy TD control – SARSA	206
Computing the optimal policy using SARSA	211
Off-policy TD control – Q learning	213
Computing the optimal policy using Q learning	218
The difference between Q learning and SARSA	220
Comparing the DP, MC, and TD methods	221
Summary	222
Questions	222
Further reading	223
Chapter 6: Case Study – The MAB Problem	225
The MAB problem	226
Creating a bandit in the Gym	228
Exploration strategies	229
Epsilon-greedy	230
Softmax exploration	234
Upper confidence bound	240
Thompson sampling	245
Applications of MAB	254
Finding the best advertisement banner using bandits	255
Creating a dataset	256
Initialize the variables	256
Define the epsilon-greedy method	257
Run the bandit test	257
Contextual bandits	259
Summary	260
Questions	260
Further reading	261

Chapter 7: Deep Learning Foundations	263
Biological and artificial neurons	264
ANN and its layers	266
Input layer	267
Hidden layer	267
Output layer	267
Exploring activation functions	267
The sigmoid function	268
The tanh function	269
The Rectified Linear Unit function	269
The softmax function	270
Forward propagation in ANNs	271
How does an ANN learn?	274
Putting it all together	281
Building a neural network from scratch	282
Recurrent Neural Networks	285
The difference between feedforward networks and RNNs	287
Forward propagation in RNNs	288
Backpropagating through time	290
LSTM to the rescue	292
Understanding the LSTM cell	293
What are CNNs?	295
Convolutional layers	297
Strides	302
Padding	303
Pooling layers	304
Fully connected layers	305
The architecture of CNNs	306
Generative adversarial networks	307
Breaking down the generator	309
Breaking down the discriminator	310
How do they learn, though?	311
Architecture of a GAN	313
Demystifying the loss function	314
Discriminator loss	314
Generator loss	316
Total loss	317
Summary	317
Questions	318
Further reading	318

Chapter 8: A Primer on TensorFlow	319
What is TensorFlow?	320
Understanding computational graphs and sessions	321
Sessions	322
Variables, constants, and placeholders	323
Variables	323
Constants	324
Placeholders and feed dictionaries	324
Introducing TensorBoard	325
Creating a name scope	327
Handwritten digit classification using TensorFlow	330
Importing the required libraries	330
Loading the dataset	330
Defining the number of neurons in each layer	331
Defining placeholders	332
Forward propagation	333
Computing loss and backpropagation	334
Computing accuracy	334
Creating a summary	335
Training the model	336
Visualizing graphs in TensorBoard	338
Introducing eager execution	343
Math operations in TensorFlow	344
TensorFlow 2.0 and Keras	348
Bonjour Keras	348
Defining the model	348
Compiling the model	350
Training the model	351
Evaluating the model	351
MNIST digit classification using TensorFlow 2.0	351
Summary	353
Questions	353
Further reading	353
Chapter 9: Deep Q Network and Its Variants	355
What is DQN?	356
Understanding DQN	358
Replay buffer	358
Loss function	361
Target network	364
Putting it all together	366
The DQN algorithm	367
Playing Atari games using DQN	368

Architecture of the DQN	368
Getting hands-on with the DQN	369
Preprocess the game screen	370
Defining the DQN class	371
Training the DQN	375
The double DQN	377
The double DQN algorithm	380
DQN with prioritized experience replay	380
Types of prioritization	381
Proportional prioritization	382
Rank-based prioritization	383
Correcting the bias	383
The dueling DQN	384
Understanding the dueling DQN	384
The architecture of a dueling DQN	386
The deep recurrent Q network	388
The architecture of a DRQN	389
Summary	391
Questions	392
Further reading	392
Chapter 10: Policy Gradient Method	393
Why policy-based methods?	394
Policy gradient intuition	396
Understanding the policy gradient	400
Deriving the policy gradient	403
Algorithm – policy gradient	407
Variance reduction methods	408
Policy gradient with reward-to-go	409
Algorithm – Reward-to-go policy gradient	411
Cart pole balancing with policy gradient	412
Computing discounted and normalized reward	413
Building the policy network	413
Training the network	415
Policy gradient with baseline	417
Algorithm – REINFORCE with baseline	420
Summary	421
Questions	422
Further reading	422
Chapter 11: Actor-Critic Methods – A2C and A3C	423
Overview of the actor-critic method	424
Understanding the actor-critic method	425
The actor-critic algorithm	428

Advantage actor-critic (A2C)	429
Asynchronous advantage actor-critic (A3C)	430
The three As	431
The architecture of A3C	432
Mountain car climbing using A3C	434
Creating the mountain car environment	435
Defining the variables	435
Defining the actor-critic class	436
Defining the worker class	441
Training the network	445
Visualizing the computational graph	446
A2C revisited	448
Summary	448
Questions	449
Further reading	449
Chapter 12: Learning DDPG, TD3, and SAC	451
Deep deterministic policy gradient	452
An overview of DDPG	452
Actor	453
Critic	453
DDPG components	454
Critic network	454
Actor network	459
Putting it all together	461
Algorithm – DDPG	463
Swinging up a pendulum using DDPG	464
Creating the Gym environment	464
Defining the variables	464
Defining the DDPG class	465
Training the network	472
Twin delayed DDPG	473
Key features of TD3	474
Clipped double Q learning	475
Delayed policy updates	477
Target policy smoothing	479
Putting it all together	480
Algorithm – TD3	482
Soft actor-critic	484
Understanding soft actor-critic	486
V and Q functions with the entropy term	486
Components of SAC	487
Critic network	487
Actor network	492
Putting it all together	493

Algorithm – SAC	495
Summary	496
Questions	497
Further reading	497
Chapter 13: TRPO, PPO, and ACKTR Methods	499
Trust region policy optimization	500
Math essentials	501
The Taylor series	502
The trust region method	507
The conjugate gradient method	508
Lagrange multipliers	509
Importance sampling	511
Designing the TRPO objective function	512
Parameterizing the policies	514
Sample-based estimation	515
Solving the TRPO objective function	516
Computing the search direction	517
Performing a line search in the search direction	521
Algorithm – TRPO	522
Proximal policy optimization	524
PPO with a clipped objective	525
Algorithm – PPO-clipped	528
Implementing the PPO-clipped method	529
Creating the Gym environment	529
Defining the PPO class	530
Training the network	535
PPO with a penalized objective	537
Algorithm – PPO-penalty	538
Actor-critic using Kronecker-factored trust region	538
Math essentials	540
Block matrix	540
Block diagonal matrix	540
The Kronecker product	542
The vec operator	543
Properties of the Kronecker product	543
Kronecker-Factored Approximate Curvature (K-FAC)	543
K-FAC in actor-critic	546
Incorporating the trust region	549
Summary	549
Questions	550
Further reading	550
Chapter 14: Distributional Reinforcement Learning	551
Why distributional reinforcement learning?	552
Categorical DQN	555

Predicting the value distribution	557
Selecting an action based on the value distribution	559
Training the categorical DQN	562
Projection step	564
Putting it all together	571
Algorithm – categorical DQN	573
Playing Atari games using a categorical DQN	574
Defining the variables	575
Defining the replay buffer	575
Defining the categorical DQN class	576
Quantile Regression DQN	583
Math essentials	584
Quantile	584
Inverse CDF (quantile function)	584
Understanding QR-DQN	586
Action selection	591
Loss function	592
Distributed Distributional DDPG	595
Critic network	596
Actor network	598
Algorithm – D4PG	600
Summary	601
Questions	602
Further reading	602
Chapter 15: Imitation Learning and Inverse RL	603
Supervised imitation learning	604
DAgger	605
Understanding DAgger	606
Algorithm – DAgger	607
Deep Q learning from demonstrations	608
Phases of DQfD	609
Pre-training phase	609
Training phase	610
Loss function of DQfD	610
Algorithm – DQfD	611
Inverse reinforcement learning	612
Maximum entropy IRL	613
Key terms	613
Back to maximum entropy IRL	614
Computing the gradient	615
Algorithm – maximum entropy IRL	617
Generative adversarial imitation learning	617
Formulation of GAIL	619

Summary	622
Questions	623
Further reading	623
Chapter 16: Deep Reinforcement Learning with Stable Baselines	625
Installing Stable Baselines	626
Creating our first agent with Stable Baselines	626
Evaluating the trained agent	627
Storing and loading the trained agent	627
Viewing the trained agent	628
Putting it all together	629
Vectorized environments	629
SubprocVecEnv	630
DummyVecEnv	631
Integrating custom environments	631
Playing Atari games with a DQN and its variants	632
Implementing DQN variants	633
Lunar lander using A2C	634
Creating a custom network	635
Swinging up a pendulum using DDPG	636
Viewing the computational graph in TensorBoard	637
Training an agent to walk using TRPO	639
Installing the MuJoCo environment	640
Implementing TRPO	643
Recording the video	646
Training a cheetah bot to run using PPO	648
Making a GIF of a trained agent	649
Implementing GAIL	651
Summary	652
Questions	652
Further reading	653
Chapter 17: Reinforcement Learning Frontiers	655
Meta reinforcement learning	656
Model-agnostic meta learning	657
Understanding MAML	660
MAML in a supervised learning setting	663
MAML in a reinforcement learning setting	665
Hierarchical reinforcement learning	668
MAXQ value function Decomposition	668
Imagination augmented agents	672
Summary	676

Questions	677
Further reading	677
Appendix 1 – Reinforcement Learning Algorithms	679
Reinforcement learning algorithm	679
Value Iteration	679
Policy Iteration	680
First-Visit MC Prediction	680
Every-Visit MC Prediction	681
MC Prediction – the Q Function	681
MC Control Method	682
On-Policy MC Control – Exploring starts	683
On-Policy MC Control – Epsilon-Greedy	683
Off-Policy MC Control	684
TD Prediction	685
On-Policy TD Control – SARSA	685
Off-Policy TD Control – Q Learning	686
Deep Q Learning	686
Double DQN	687
REINFORCE Policy Gradient	688
Policy Gradient with Reward-To-Go	688
REINFORCE with Baseline	689
Advantage Actor Critic	689
Asynchronous Advantage Actor-Critic	690
Deep Deterministic Policy Gradient	690
Twin Delayed DDPG	691
Soft Actor-Critic	692
Trust Region Policy Optimization	693
PPO-Clipped	694
PPO-Penalty	695
Categorical DQN	695
Distributed Distributional DDPG	697
Dagger	698
Deep Q learning from demonstrations	698
MaxEnt Inverse Reinforcement Learning	699
MAML in Reinforcement Learning	700
Appendix 2 – Assessments	701
Chapter 1 – Fundamentals of Reinforcement Learning	701
Chapter 2 – A Guide to the Gym Toolkit	702
Chapter 3 – The Bellman Equation and Dynamic Programming	702
Chapter 4 – Monte Carlo Methods	703

Chapter 5 – Understanding Temporal Difference Learning	704
Chapter 6 – Case Study – The MAB Problem	705
Chapter 7 – Deep Learning Foundations	706
Chapter 8 – A Primer on TensorFlow	707
Chapter 9 – Deep Q Network and Its Variants	708
Chapter 10 – Policy Gradient Method	709
Chapter 11 – Actor-Critic Methods – A2C and A3C	709
Chapter 12 – Learning DDPG, TD3, and SAC	710
Chapter 13 – TRPO, PPO, and ACKTR Methods	711
Chapter 14 – Distributional Reinforcement Learning	712
Chapter 15 – Imitation Learning and Inverse RL	713
Chapter 16 – Deep Reinforcement Learning with Stable Baselines	714
Chapter 17 – Reinforcement Learning Frontiers	714
Other Books You May Enjoy	717
Index	721

Preface

With significant enhancement in the quality and quantity of algorithms in recent years, this second edition of *Hands-On Reinforcement Learning with Python* has been revamped into an example-rich guide to learning state-of-the-art **reinforcement learning (RL)** and deep RL algorithms with TensorFlow 2 and the OpenAI Gym toolkit.

In addition to exploring RL basics and foundational concepts such as the Bellman equation, Markov decision processes, and dynamic programming, this second edition dives deep into the full spectrum of value-based, policy-based, and actor-critic RL methods. It explores state-of-the-art algorithms such as DQN, TRPO, PPO and ACKTR, DDPG, TD3, and SAC in depth, demystifying the underlying math and demonstrating implementations through simple code examples.

The book has several new chapters dedicated to new RL techniques including distributional RL, imitation learning, inverse RL, and meta RL. You will learn to leverage Stable Baselines, an improvement of OpenAI's baseline library, to implement popular RL algorithms effortlessly. The book concludes with an overview of promising approaches such as meta-learning and imagination augmented agents in research.

Who this book is for

If you're a machine learning developer with little or no experience with neural networks interested in artificial intelligence and want to learn about reinforcement learning from scratch, this book is for you. Basic familiarity with linear algebra, calculus, and Python is required. Some experience with TensorFlow would be a plus.

What this book covers

Chapter 1, Fundamentals of Reinforcement Learning, helps you build a strong foundation on RL concepts. We will learn about the key elements of RL, the Markov decision process, and several important fundamental concepts such as action spaces, policies, episodes, the value function, and the Q function. At the end of the chapter, we will learn about some of the interesting applications of RL and we will also look into the key terms and terminologies frequently used in RL.

Chapter 2, A Guide to the Gym Toolkit, provides a complete guide to OpenAI's Gym toolkit. We will understand several interesting environments provided by Gym in detail by implementing them. We will begin our hands-on RL journey from this chapter by implementing several fundamental RL concepts using Gym.

Chapter 3, The Bellman Equation and Dynamic Programming, will help us understand the Bellman equation in detail with extensive math. Next, we will learn two interesting classic RL algorithms called the value and policy iteration methods, which we can use to find the optimal policy. We will also see how to implement value and policy iteration methods for solving the Frozen Lake problem.

Chapter 4, Monte Carlo Methods, explains the model-free method, Monte Carlo. We will learn what prediction and control tasks are, and then we will look into Monte Carlo prediction and Monte Carlo control methods in detail. Next, we will implement the Monte Carlo method to solve the blackjack game using the Gym toolkit.

Chapter 5, Understanding Temporal Difference Learning, deals with one of the most popular and widely used model-free methods called **Temporal Difference (TD)** learning. First, we will learn how the TD prediction method works in detail, and then we will explore the on-policy TD control method called SARSA and the off-policy TD control method called Q learning in detail. We will also implement TD control methods to solve the Frozen Lake problem using Gym.

Chapter 6, Case Study – The MAB Problem, explains one of the classic problems in RL called the **multi-armed bandit (MAB)** problem. We will start the chapter by understanding what the MAB problem is and then we will learn about several exploration strategies such as epsilon-greedy, softmax exploration, upper confidence bound, and Thompson sampling methods for solving the MAB problem in detail.

Chapter 7, Deep Learning Foundations, helps us to build a strong foundation on deep learning. We will start the chapter by understanding how artificial neural networks work. Then we will learn several interesting deep learning algorithms, such as recurrent neural networks, LSTM networks, convolutional neural networks, and generative adversarial networks.

Chapter 8, A Primer on TensorFlow, deals with one of the most popular deep learning libraries called TensorFlow. We will understand how to use TensorFlow by implementing a neural network to recognize handwritten digits. Next, we will learn to perform several math operations using TensorFlow. Later, we will learn about TensorFlow 2.0 and see how it differs from the previous TensorFlow versions.

Chapter 9, Deep Q Network and Its Variants, enables us to kick-start our deep RL journey. We will learn about one of the most popular deep RL algorithms called the **Deep Q Network (DQN)**. We will understand how DQN works step by step along with the extensive math. We will also implement a DQN to play Atari games. Next, we will explore several interesting variants of DQN, called Double DQN, Dueling DQN, DQN with prioritized experience replay, and DRQN.

Chapter 10, Policy Gradient Method, covers policy gradient methods. We will understand how the policy gradient method works along with the detailed derivation. Next, we will learn several variance reduction methods such as policy gradient with reward-to-go and policy gradient with baseline. We will also understand how to train an agent for the Cart Pole balancing task using policy gradient.

Chapter 11, Actor-Critic Methods – A2C and A3C, deals with several interesting actor-critic methods such as advantage actor-critic and asynchronous advantage actor-critic. We will learn how these actor-critic methods work in detail, and then we will implement them for a mountain car climbing task using OpenAI Gym.

Chapter 12, Learning DDPG, TD3, and SAC, covers state-of-the-art deep RL algorithms such as deep deterministic policy gradient, twin delayed DDPG, and soft actor, along with step by step derivation. We will also learn how to implement the DDPG algorithm for performing the inverted pendulum swing-up task using Gym.

Chapter 13, TRPO, PPO, and ACKTR Methods, deals with several popular policy gradient methods such as TRPO and PPO. We will dive into the math behind TRPO and PPO step by step and understand how TRPO and PPO helps an agent find the optimal policy. Next, we will learn to implement PPO for performing the inverted pendulum swing-up task. At the end, we will learn about the actor-critic method called actor-critic using Kronecker-Factored trust region in detail.

Chapter 14, Distributional Reinforcement Learning, covers distributional RL algorithms. We will begin the chapter by understanding what distributional RL is. Then we will explore several interesting distributional RL algorithms such as categorical DQN, quantile regression DQN, and distributed distributional DDPG.

Chapter 15, Imitation Learning and Inverse RL, explains imitation and inverse RL algorithms. First, we will understand how supervised imitation learning, DAgger, and deep Q learning from demonstrations work in detail. Next, we will learn about maximum entropy inverse RL. At the end of the chapter, we will learn about generative adversarial imitation learning.

Chapter 16, Deep Reinforcement Learning with Stable Baselines, helps us to understand how to implement deep RL algorithms using a library called Stable Baselines. We will learn what Stable Baselines is and how to use it in detail by implementing several interesting Deep RL algorithms such as DQN, A2C, DDPG, TRPO, and PPO.

Chapter 17, Reinforcement Learning Frontiers, covers several interesting avenues in RL, such as meta RL, hierarchical RL, and imagination augmented agents in detail.

To get the most out of this book

You need the following software for this book:

- Anaconda
- Python
- Any web browser

Download the example code files

You can download the example code files for this book from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://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 <http://www.packtpub.com>.
2. Select the **SUPPORT** tab.
3. Click on **Code Downloads & Errata**.
4. Enter the name of the book in the **Search** box and follow the on-screen 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-Reinforcement-Learning-with-Python>. 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: https://static.packt-cdn.com/downloads/9781839210686_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. For example: "epsilon_greedy computes the optimal policy."

A block of code is set as follows:

```
def epsilon_greedy(epsilon):  
    if np.random.uniform(0,1) < epsilon:  
        return env.action_space.sample()  
    else:  
        return np.argmax(Q)
```

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

```
def epsilon_greedy(epsilon):  
    if np.random.uniform(0,1) < epsilon:  
        return env.action_space.sample()  
    else:  
        return np.argmax(Q)
```

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

```
source activate universe
```

Bold: Indicates a new term, an important word, or words that you see on the screen, for example, in menus or dialog boxes, also appear in the text like this. For example: "The **Markov Reward Process (MRP)** is an extension of the Markov chain with the reward function."



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: Email feedback@packtpub.com, and mention the book's title in the subject of your message. If you have questions about any aspect of this book, please email us at questions@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, <http://www.packtpub.com/submit-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@packtpub.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 <http://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 packtpub.com.

