

ADDISON WESLEY DATA & ANALYTICS SERIES



FOUNDATIONS OF DEEP REINFORCEMENT LEARNING

Theory and Practice in Python



**LAURA GRAESSER
WAH LOON KENG**

Praise for *Foundations of Deep Reinforcement Learning*

“This book provides an accessible introduction to deep reinforcement learning covering the mathematical concepts behind popular algorithms as well as their practical implementation. I think the book will be a valuable resource for anyone looking to apply deep reinforcement learning in practice.”

—*Volodymyr Mnih, lead developer of DQN*

“An excellent book to quickly develop expertise in the theory, language, and practical implementation of deep reinforcement learning algorithms. A limpid exposition which uses familiar notation; all the most recent techniques explained with concise, readable code, and not a page wasted in irrelevant detours: it is the perfect way to develop a solid foundation on the topic.”

—*Vincent Vanhoucke, principal scientist, Google*

“As someone who spends their days trying to make deep reinforcement learning methods more useful for the general public, I can say that Laura and Keng’s book is a welcome addition to the literature. It provides both a readable introduction to the fundamental concepts in reinforcement learning as well as intuitive explanations and code for many of the major algorithms in the field. I imagine this will become an invaluable resource for individuals interested in learning about deep reinforcement learning for years to come.”

—*Arthur Juliani, senior machine learning engineer, Unity Technologies*

“Until now, the only way to get to grips with deep reinforcement learning was to slowly accumulate knowledge from dozens of different sources. Finally, we have a book bringing everything together in one place.”

—*Matthew Rahtz, ML researcher, ETH Zürich*

Foundations of Deep Reinforcement Learning: Theory and Practice in Python

Table of Contents

Cover

Title Page

Copyright Page

Contents

Foreword

Preface

Acknowledgments

About the Authors

1 Introduction to Reinforcement Learning

1.1 Reinforcement Learning

1.2 Reinforcement Learning as MDP

1.3 Learnable Functions in Reinforcement Learning

1.4 Deep Reinforcement Learning Algorithms

1.4.1 Policy-Based Algorithms

1.4.2 Value-Based Algorithms

1.4.3 Model-Based Algorithms

1.4.4 Combined Methods

1.4.5 Algorithms Covered in This Book

1.4.6 On-Policy and Off-Policy Algorithms

1.4.7 Summary

1.5 Deep Learning for Reinforcement Learning

Table of Contents

1.6 Reinforcement Learning and Supervised Learning

1.6.1 Lack of an Oracle

1.6.2 Sparsity of Feedback

1.6.3 Data Generation

1.7 Summary

Part I: Policy-Based and Value-Based Algorithms

2 REINFORCE

2.1 Policy

2.2 The Objective Function

2.3 The Policy Gradient

2.3.1 Policy Gradient Derivation

2.4 Monte Carlo Sampling

2.5 REINFORCE Algorithm

2.5.1 Improving REINFORCE

2.6 Implementing REINFORCE

2.6.1 A Minimal REINFORCE Implementation

2.6.2 Constructing Policies with PyTorch

2.6.3 Sampling Actions

2.6.4 Calculating Policy Loss

2.6.5 REINFORCE Training Loop

2.6.6 On-Policy Replay Memory

2.7 Training a REINFORCE Agent

2.8 Experimental Results

2.8.1 Experiment: The Effect of Discount Factor

2.8.2 Experiment: The Effect of Baseline

2.9 Summary

2.10 Further Reading

2.11 History

3 SARSA

3.1 The Q- and V-Functions

Table of Contents

3.2 Temporal Difference Learning

3.2.1 Intuition for Temporal Difference Learning

3.3 Action Selection in SARSA

3.3.1 Exploration and Exploitation

3.4 SARSA Algorithm

3.4.1 On-Policy Algorithms

3.5 Implementing SARSA

3.5.1 Action Function: -Greedy

3.5.2 Calculating the Q-Loss

3.5.3 SARSA Training Loop

3.5.4 On-Policy Batched Replay Memory

3.6 Training a SARSA Agent

3.7 Experimental Results

3.7.1 Experiment: The Effect of Learning Rate

3.8 Summary

3.9 Further Reading

3.10 History

4 Deep Q-Networks (DQN)

4.1 Learning the Q-Function in DQN

4.2 Action Selection in DQN

4.2.1 The Boltzmann Policy

4.3 Experience Replay

4.4 DQN Algorithm

4.5 Implementing DQN

4.5.1 Calculating the Q-Loss

4.5.2 DQN Training Loop

4.5.3 Replay Memory

4.6 Training a DQN Agent

4.7 Experimental Results

4.7.1 Experiment: The Effect of Network Architecture

4.8 Summary

Table of Contents

4.9 Further Reading

4.10 History

5 Improving DQN

5.1 Target Networks

5.2 Double DQN 106

5.3 Prioritized Experience Replay (PER) 109

5.3.1 Importance Sampling

5.4 Modified DQN Implementation

5.4.1 Network Initialization

5.4.2 Calculating the Q-Loss

5.4.3 Updating the Target Network

5.4.4 DQN with Target Networks

5.4.5 Double DQN

5.4.6 Prioritized Experienced Replay

5.5 Training a DQN Agent to Play Atari Games

5.6 Experimental Results

5.6.1 Experiment: The Effect of Double DQN and PER

5.7 Summary

5.8 Further Reading

Part II: Combined Methods

6 Advantage Actor-Critic (A2C)

6.1 The Actor

6.2 The Critic

6.2.1 The Advantage Function

6.2.2 Learning the Advantage Function

6.3 A2C Algorithm

6.4 Implementing A2C

6.4.1 Advantage Estimation

6.4.2 Calculating Value Loss and Policy Loss

6.4.3 Actor-Critic Training Loop

6.5 Network Architecture

Table of Contents

6.6 Training an A2C Agent

6.6.1 A2C with n-Step Returns on Pong

6.6.2 A2C with GAE on Pong

6.6.3 A2C with n-Step Returns on BipedalWalker

6.7 Experimental Results

6.7.1 Experiment: The Effect of n-Step Returns

6.7.2 Experiment: The Effect of of GAE

6.8 Summary

6.9 Further Reading

6.10 History

7 Proximal Policy Optimization (PPO)

7.1 Surrogate Objective

7.1.1 Performance Collapse

7.1.2 Modifying the Objective

7.2 Proximal Policy Optimization (PPO)

7.3 PPO Algorithm

7.4 Implementing PPO

7.4.1 Calculating the PPO Policy Loss

7.4.2 PPO Training Loop

7.5 Training a PPO Agent

7.5.1 PPO on Pong

7.5.2 PPO on BipedalWalker

7.6 Experimental Results

7.6.1 Experiment: The Effect of of GAE

7.6.2 Experiment: The Effect of Clipping Variable

7.7 Summary

7.8 Further Reading

8 Parallelization Methods

8.1 Synchronous Parallelization

8.2 Asynchronous Parallelization

8.2.1 Hogwild!

Table of Contents

8.3 Training an A3C Agent

8.4 Summary

8.5 Further Reading

9 Algorithm Summary

Part III: Practical Details

10 Getting Deep RL to Work

10.1 Software Engineering Practices

10.1.1 Unit Tests

10.1.2 Code Quality

10.1.3 Git Workflow

10.2 Debugging Tips

10.2.1 Signs of Life

10.2.2 Policy Gradient Diagnoses

10.2.3 Data Diagnoses

10.2.4 Preprocessor

10.2.5 Memory

10.2.6 Algorithmic Functions

10.2.7 Neural Networks

10.2.8 Algorithm Simplification

10.2.9 Problem Simplification

10.2.10 Hyperparameters

10.2.11 Lab Workflow

10.3 Atari Tricks

10.4 Deep RL Almanac

10.4.1 Hyperparameter Tables

10.4.2 Algorithm Performance Comparison

10.5 Summary

11 SLM Lab

11.1 Algorithms Implemented in SLM Lab

11.2 Spec File

11.2.1 Search Spec Syntax

Table of Contents

11.3 Running SLM Lab

11.3.1 SLM Lab Commands

11.4 Analyzing Experiment Results

11.4.1 Overview of the Experiment Data

11.5 Summary

12 Network Architectures

12.1 Types of Neural Networks

12.1.1 Multilayer Perceptrons (MLPs)

12.1.2 Convolutional Neural Networks (CNNs)

12.1.3 Recurrent Neural Networks (RNNs)

12.2 Guidelines for Choosing a Network Family

12.2.1 MDPs vs. POMDPs

12.2.2 Choosing Networks for Environments

12.3 The Net API

12.3.1 Input and Output Layer Shape Inference

12.3.2 Automatic Network Construction

12.3.3 Training Step

12.3.4 Exposure of Underlying Methods

12.4 Summary

12.5 Further Reading

13 Hardware

13.1 Computer

13.2 Data Types

13.3 Optimizing Data Types in RL

13.4 Choosing Hardware

13.5 Summary

Part IV: Environment Design

14 States

14.1 Examples of States

14.2 State Completeness

Table of Contents

14.3 State Complexity

14.4 State Information Loss

14.4.1 Image Grayscale

14.4.2 Discretization

14.4.3 Hash Conflict

14.4.4 Metainformation Loss

14.5 Preprocessing

14.5.1 Standardization

14.5.2 Image Preprocessing

14.5.3 Temporal Preprocessing

14.6 Summary

15 Actions

15.1 Examples of Actions

15.2 Action Completeness

15.3 Action Complexity

15.4 Summary

15.5 Further Reading: Action Design in Everyday Things

16 Rewards

16.1 The Role of Rewards

16.2 Reward Design Guidelines

16.3 Summary

17 Transition Function

17.1 Feasibility Checks

17.2 Reality Check

17.3 Summary

Epilogue

A: Deep Reinforcement Learning Timeline

B: Example Environments

B.1 Discrete Environments

Table of Contents

B.1.1 CartPole-v0

B.1.2 MountainCar-v0

B.1.3 LunarLander-v2

B.1.4 PongNoFrameskip-v4

B.1.5 BreakoutNoFrameskip-v4

B.2 Continuous Environments

B.2.1 Pendulum-v0

B.2.2 BipedalWalker-v2

References

Index