

# Table of Contents

<b>Preface</b>	<b>xi</b>
<b>Chapter 1: What is Reinforcement Learning?</b>	<b>1</b>
<b>Learning – supervised, unsupervised, and reinforcement</b>	<b>2</b>
<b>RL formalisms and relations</b>	<b>5</b>
Reward	5
The agent	7
The environment	8
Actions	8
Observations	8
<b>Markov decision processes</b>	<b>11</b>
Markov process	12
Markov reward process	16
Markov decision process	19
<b>Summary</b>	<b>23</b>
<b>Chapter 2: OpenAI Gym</b>	<b>25</b>
<b>The anatomy of the agent</b>	<b>25</b>
<b>Hardware and software requirements</b>	<b>28</b>
<b>OpenAI Gym API</b>	<b>30</b>
Action space	30
Observation space	31
The environment	33
Creation of the environment	34
The CartPole session	36
<b>The random CartPole agent</b>	<b>39</b>

---

<b>The extra Gym functionality – wrappers and monitors</b>	<b>40</b>
Wrappers	40
Monitor	43
<b>Summary</b>	<b>47</b>
<b>Chapter 3: Deep Learning with PyTorch</b>	<b>49</b>
<hr/>	
<b>Tensors</b>	<b>50</b>
Creation of tensors	50
Scalar tensors	53
Tensor operations	53
GPU tensors	54
<b>Gradients</b>	<b>55</b>
Tensors and gradients	56
<b>NN building blocks</b>	<b>59</b>
<b>Custom layers</b>	<b>60</b>
<b>Final glue – loss functions and optimizers</b>	<b>63</b>
Loss functions	63
Optimizers	64
<b>Monitoring with TensorBoard</b>	<b>66</b>
TensorBoard 101	67
Plotting stuff	68
<b>Example – GAN on Atari images</b>	<b>70</b>
<b>Summary</b>	<b>75</b>
<b>Chapter 4: The Cross-Entropy Method</b>	<b>77</b>
<hr/>	
<b>Taxonomy of RL methods</b>	<b>78</b>
<b>Practical cross-entropy</b>	<b>79</b>
<b>Cross-entropy on CartPole</b>	<b>81</b>
<b>Cross-entropy on FrozenLake</b>	<b>90</b>
<b>Theoretical background of the cross-entropy method</b>	<b>95</b>
<b>Summary</b>	<b>97</b>
<b>Chapter 5: Tabular Learning and the Bellman Equation</b>	<b>99</b>
<hr/>	
<b>Value, state, and optimality</b>	<b>99</b>
<b>The Bellman equation of optimality</b>	<b>102</b>
<b>Value of action</b>	<b>104</b>
<b>The value iteration method</b>	<b>106</b>
<b>Value iteration in practice</b>	<b>109</b>
<b>Q-learning for FrozenLake</b>	<b>114</b>
<b>Summary</b>	<b>117</b>

---

<b>Chapter 6: Deep Q-Networks</b>	<b>119</b>
<b>Real-life value iteration</b>	<b>119</b>
<b>Tabular Q-learning</b>	<b>121</b>
<b>Deep Q-learning</b>	<b>125</b>
Interaction with the environment	127
SGD optimization	128
Correlation between steps	128
The Markov property	129
The final form of DQN training	129
<b>DQN on Pong</b>	<b>130</b>
Wrappers	132
DQN model	137
Training	139
Running and performance	148
Your model in action	150
<b>Summary</b>	<b>153</b>
<b>Chapter 7: DQN Extensions</b>	<b>155</b>
<b>The PyTorch Agent Net library</b>	<b>156</b>
Agent	157
Agent's experience	158
Experience buffer	159
Gym env wrappers	160
<b>Basic DQN</b>	<b>160</b>
<b>N-step DQN</b>	<b>168</b>
Implementation	170
<b>Double DQN</b>	<b>172</b>
Implementation	173
Results	176
<b>Noisy networks</b>	<b>178</b>
Implementation	179
Results	182
<b>Prioritized replay buffer</b>	<b>184</b>
Implementation	185
Results	190
<b>Dueling DQN</b>	<b>191</b>
Implementation	193
Results	194
<b>Categorical DQN</b>	<b>195</b>
Implementation	197
Results	205

---

<b>Combining everything</b>	<b>207</b>
Implementation	208
Results	213
<b>Summary</b>	<b>214</b>
<b>References</b>	<b>214</b>
<b>Chapter 8: Stocks Trading Using RL</b>	<b>217</b>
<b>Trading</b>	<b>217</b>
<b>Data</b>	<b>218</b>
<b>Problem statements and key decisions</b>	<b>219</b>
<b>The trading environment</b>	<b>221</b>
<b>Models</b>	<b>229</b>
<b>Training code</b>	<b>231</b>
<b>Results</b>	<b>231</b>
The feed-forward model	231
The convolution model	237
<b>Things to try</b>	<b>239</b>
<b>Summary</b>	<b>240</b>
<b>Chapter 9: Policy Gradients – An Alternative</b>	<b>241</b>
<b>Values and policy</b>	<b>241</b>
Why policy?	242
Policy representation	243
Policy gradients	244
<b>The REINFORCE method</b>	<b>244</b>
The CartPole example	246
Results	250
Policy-based versus value-based methods	251
<b>REINFORCE issues</b>	<b>252</b>
Full episodes are required	252
High gradients variance	253
Exploration	253
Correlation between samples	254
<b>PG on CartPole</b>	<b>254</b>
Results	257
<b>PG on Pong</b>	<b>259</b>
Results	261
<b>Summary</b>	<b>262</b>

---

<b>Chapter 10: The Actor-Critic Method</b>	<b>263</b>
<b>Variance reduction</b>	<b>263</b>
<b>CartPole variance</b>	<b>265</b>
<b>Actor-critic</b>	<b>268</b>
<b>A2C on Pong</b>	<b>270</b>
<b>A2C on Pong results</b>	<b>276</b>
<b>Tuning hyperparameters</b>	<b>279</b>
Learning rate	279
Entropy beta	280
Count of environments	280
Batch size	281
<b>Summary</b>	<b>281</b>
<b>Chapter 11: Asynchronous Advantage Actor-Critic</b>	<b>283</b>
<b>Correlation and sample efficiency</b>	<b>283</b>
<b>Adding an extra A to A2C</b>	<b>285</b>
<b>Multiprocessing in Python</b>	<b>288</b>
<b>A3C – data parallelism</b>	<b>288</b>
Results	294
<b>A3C – gradients parallelism</b>	<b>295</b>
Results	301
<b>Summary</b>	<b>301</b>
<b>Chapter 12: Chatbots Training with RL</b>	<b>303</b>
<b>Chatbots overview</b>	<b>303</b>
<b>Deep NLP basics</b>	<b>305</b>
Recurrent Neural Networks	306
Embeddings	307
Encoder-Decoder	309
<b>Training of seq2seq</b>	<b>309</b>
Log-likelihood training	310
Bilingual evaluation understudy (BLEU) score	312
RL in seq2seq	313
Self-critical sequence training	315
<b>The chatbot example</b>	<b>316</b>
The example structure	316
Modules: cornell.py and data.py	317
BLEU score and utils.py	318

---

Model	319
Training: cross-entropy	326
Running the training	330
Checking the data	332
Playing with the trained model	334
Training: SCST	336
Running the SCST training	343
Results	344
Telegram bot	345
<b>Summary</b>	<b>349</b>
<b>Chapter 13: Web Navigation</b>	<b>351</b>
<b>Web navigation</b>	<b>351</b>
Browser automation and RL	352
Mini World of Bits benchmark	353
<b>OpenAI Universe</b>	<b>356</b>
Installation	357
Actions and observations	357
Environment creation	358
MiniWoB stability	361
<b>Simple clicking approach</b>	<b>362</b>
Grid actions	362
Example overview	364
Model	365
Training code	366
Starting containers	371
Training process	374
Checking the learned policy	375
Issues with simple clicking	377
<b>Human demonstrations</b>	<b>379</b>
Recording the demonstrations	380
Recording format	383
Training using demonstrations	386
Results	387
TicTacToe problem	388
<b>Adding text description</b>	<b>390</b>
Results	396
<b>Things to try</b>	<b>397</b>
<b>Summary</b>	<b>398</b>

---

<b>Chapter 14: Continuous Action Space</b>	<b>399</b>
<b>Why a continuous space?</b>	<b>399</b>
<b>Action space</b>	<b>400</b>
<b>Environments</b>	<b>401</b>
<b>The Actor-Critic (A2C) method</b>	<b>403</b>
Implementation	404
Results	408
Using models and recording videos	409
<b>Deterministic policy gradients</b>	<b>410</b>
Exploration	411
Implementation	412
Results	417
Recording videos	418
<b>Distributional policy gradients</b>	<b>419</b>
Architecture	419
Implementation	420
Results	425
<b>Things to try</b>	<b>425</b>
<b>Summary</b>	<b>426</b>
<b>Chapter 15: Trust Regions – TRPO, PPO, and ACKTR</b>	<b>427</b>
<b>Introduction</b>	<b>427</b>
<b>Roboschool</b>	<b>428</b>
<b>A2C baseline</b>	<b>429</b>
Results	431
Videos recording	432
<b>Proximal Policy Optimization</b>	<b>432</b>
Implementation	433
Results	437
<b>Trust Region Policy Optimization</b>	<b>438</b>
Implementation	438
Results	440
<b>A2C using ACKTR</b>	<b>440</b>
Implementation	441
Results	442
<b>Summary</b>	<b>442</b>

---

<b>Chapter 16: Black-Box Optimization in RL</b>	<b>443</b>
<b>Black-box methods</b>	<b>443</b>
<b>Evolution strategies</b>	<b>444</b>
<b>ES on CartPole</b>	<b>445</b>
Results	450
<b>ES on HalfCheetah</b>	<b>451</b>
Results	456
<b>Genetic algorithms</b>	<b>457</b>
<b>GA on CartPole</b>	<b>458</b>
Results	460
<b>GA tweaks</b>	<b>461</b>
Deep GA	461
Novelty search	461
<b>GA on Cheetah</b>	<b>462</b>
Results	464
<b>Summary</b>	<b>465</b>
<b>References</b>	<b>466</b>
<b>Chapter 17: Beyond Model-Free – Imagination</b>	<b>467</b>
<b>Model-based versus model-free</b>	<b>467</b>
<b>Model imperfections</b>	<b>469</b>
<b>Imagination-augmented agent</b>	<b>470</b>
The environment model	472
The rollout policy	472
The rollout encoder	473
Paper results	473
<b>I2A on Atari Breakout</b>	<b>473</b>
The baseline A2C agent	474
EM training	475
The imagination agent	478
The I2A model	478
The Rollout encoder	482
Training of I2A	483
<b>Experiment results</b>	<b>484</b>
The baseline agent	484
Training EM weights	485
Training with the I2A model	487
<b>Summary</b>	<b>489</b>
<b>References</b>	<b>489</b>

---

---

<b>Chapter 18: AlphaGo Zero</b>	<b>491</b>
<b>Board games</b>	<b>491</b>
<b>The AlphaGo Zero method</b>	<b>493</b>
Overview	493
Monte-Carlo Tree Search	494
Self-play	496
Training and evaluation	497
<b>Connect4 bot</b>	<b>497</b>
Game model	498
Implementing MCTS	500
Model	506
Training	508
Testing and comparison	509
<b>Connect4 results</b>	<b>509</b>
<b>Summary</b>	<b>512</b>
<b>References</b>	<b>512</b>
<b>Book summary</b>	<b>512</b>
<b>Other Books You May Enjoy</b>	<b>513</b>
<b>Index</b>	<b>517</b>

---