

Inhalt

Vorwort	XI
Danksagung	XII
Die Autoren	XII
Über dieses Buch	XIII
Teil I: Grundlagen	1
1 Was ist Reinforcement Learning?	3
1.1 Das „Tiefe“ beim Deep Reinforcement Learning	4
1.2 Reinforcement Learning	6
1.3 Dynamische Programmierung versus Monte Carlo	9
1.4 Das Reinforcement-Learning-Framework	11
1.5 Was kann ich mit Reinforcement Learning anfangen?	15
1.6 Warum Deep Reinforcement Learning?	17
1.7 Unser didaktisches Werkzeug: String-Diagramme	20
1.8 Wie geht es weiter?	22
1.9 Zusammenfassung	23
2 Modellierung von Reinforcement-Learning-Problemen: Markov Decision Processes	25
2.1 String-Diagramme und unsere Lehrmethoden	25
2.2 Multi-Armed-Bandit-Problem	30
2.2.1 Erkundung und Ausnutzung	31
2.2.2 Epsilon-greedy-Strategie	32
2.2.3 Softmax-Auswahlverfahren	37
2.3 Anwendung von Banditen zur Optimierung von Anzeigenplatzierungen	40
2.3.1 Kontextabhängige Banditen	41
2.3.2 Zustände, Aktionen, Belohnungen	42
2.4 Netzwerke mit PyTorch erstellen	43
2.4.1 Automatische Differenzierung	44
2.4.2 Modelle erstellen	44

2.5	Lösen kontextabhängiger Banditen.....	45
2.6	Die Markov-Eigenschaft.....	51
2.7	Vorhersage zukünftiger Belohnungen: Value- und Policy-Funktionen.....	53
2.7.1	Policy-Funktionen.....	54
2.7.2	Optimale Policy.....	55
2.7.3	Wert-Funktionen.....	56
2.8	Zusammenfassung.....	57
3	Vorhersage der besten Zustände und Aktionen: Deep Q-Networks.....	59
3.1	Die Q-Funktion.....	59
3.2	Navigieren mit Q-Learning.....	61
3.2.1	Was ist Q-Learning?.....	62
3.2.2	Gridworld bewältigen.....	63
3.2.3	Hyperparameter.....	65
3.2.4	Diskontierungsfaktor.....	65
3.2.5	Aufbau des Netzwerks.....	66
3.2.6	Einführung in die Gridworld-Spiele-Engine.....	68
3.2.7	Ein neuronales Netz als Q-Funktion.....	71
3.3	Verhinderung katastrophalen Vergessens: Auf Erfahrung basiertes Wiederholungsspiel.....	81
3.3.1	Katastrophales Vergessen.....	81
3.3.2	Auf Erfahrung basiertes Wiederholungsspiel.....	83
3.4	Verbesserung der Stabilität mit einem Target Network.....	87
3.4.1	Lerninstabilität.....	88
3.5	Rückblick.....	93
3.6	Zusammenfassung.....	96
4	Lernen, die beste Policy auszuwählen: Policy-Gradient-Methoden.....	97
4.1	Policy-Funktion mit neuronalen Netzen.....	98
4.1.1	Neuronales Netz als Policy-Funktion.....	98
4.1.2	Stochastischer Policy-Gradient.....	99
4.1.3	Erkundung.....	102
4.2	Verstärkung guter Aktionen: Der Policy-Gradient-Algorithmus.....	102
4.2.1	Definieren eines Ziels.....	103
4.2.2	Verstärkung der Aktionen.....	104
4.2.3	Log-Wahrscheinlichkeit.....	106
4.2.4	Anerkennungszuweisung.....	107
4.3	Arbeiten mit OpenAI Gym.....	108
4.3.1	CartPole.....	110
4.3.2	Die OpenAI Gym-API.....	111
4.4	Der REINFORCE-Algorithmus.....	112

4.4.1	Erstellen des Policy-Netzwerks	112
4.4.2	Interaktion des Agenten mit der Umgebung	113
4.4.3	Das Modell trainieren	114
4.4.4	Die vollständige Trainingsschleife	116
4.4.5	Schlussfolgerung	118
4.5	Zusammenfassung	119

5 Bewältigung komplexerer Probleme mit Actor-Critic-Methoden .. 121

5.1	Die Kombination von Wert- und Policy-Funktion	123
5.2	Verteiltes Training	128
5.3	Advantage-Actor-Critic	134
5.4	N-step-Actor-Critic	143
5.5	Zusammenfassung	148

Teil II: Darüber hinaus 151

6 Alternative Optimierungsmethoden:

Evolutionäre Algorithmen 153

6.1	Ein anderer Ansatz für das Reinforcement Learning	154
6.2	Reinforcement Learning mit Evolutionsstrategien	155
6.2.1	Evolution in der Theorie	155
6.2.2	Entwicklung in der Praxis	159
6.3	Ein genetischer Algorithmus für CartPole	164
6.4	Vor- und Nachteile von evolutionären Algorithmen	170
6.4.1	Evolutionäre Algorithmen erforschen mehr	170
6.4.2	Evolutionäre Algorithmen sind unglaublich testintensiv	171
6.4.3	Simulatoren	172
6.5	Evolutionäre Algorithmen als skalierbare Alternative	172
6.5.1	Skalierung evolutionärer Algorithmen	173
6.5.2	Parallele vs. serielle Verarbeitung	174
6.5.3	Skalierungseffizienz	175
6.5.4	Kommunikation zwischen Knoten	176
6.5.5	Lineare Skalierung	178
6.5.6	Auf Gradienten basierte Ansätze zur Skalierung	178
6.6	Zusammenfassung	179

7 Verteilungs-DQN: Die ganze Geschichte 181

7.1	Was ist falsch an Q-Learning?	182
7.2	Wahrscheinlichkeitsrechnung und Statistik	187
7.2.1	A priori und A posteriori	189
7.2.2	Erwartung und Varianz	190
7.3	Die Bellman-Gleichung	194

7.4	Verteilungs-Q-Learning	196
7.4.1	Darstellung einer Wahrscheinlichkeitsverteilung in Python.....	197
7.4.2	Implementierung des Dist-DQN	206
7.5	Vergleich von Wahrscheinlichkeitsverteilungen	208
7.6	Dist-DQN auf simulierten Daten	213
7.7	Verwendung von Verteilungs-Q-Learning zum Spielen von Freeway	218
7.8	Zusammenfassung	224
8	Von Neugierde getriebene Erkundung.....	225
8.1	Mit prädiktiver Kodierung gegen spärliche Belohnungen vorgehen.....	227
8.2	Vorhersage der inversen Dynamik	230
8.3	Implementierung von Super Mario Bros.....	234
8.4	Vorverarbeitung und das Q-Netz.....	236
8.5	Einrichten des Q-Netz und der Policy-Funktion.....	238
8.6	Intrinsisches Neugierde-Modul	242
8.7	Alternative intrinsische Belohnungsmechanismen	255
8.8	Zusammenfassung	258
9	Lernen mit Multi-Agent Reinforcement Learning.....	259
9.1	Von einem zu vielen Agenten	259
9.2	Nachbarschafts-Q-Learning	264
9.3	Das 1D-Ising-Modell	267
9.4	Mean-Field-Q-Learning und das 2D-Ising-Modell	278
9.5	Gemischte kooperativ-konkurrierende Spiele	288
9.6	Zusammenfassung	299
10	Interpretierbares Reinforcement Learning: Aufmerksamkeitsmodelle und relationale Modelle	301
10.1	Interpretierbarkeit von Machine Learning mit Aufmerksamkeit und relationalen Verzerrungen	302
10.1.1	Invarianz und Äquivarianz	304
10.2	Relationale Argumentation mit Aufmerksamkeit	306
10.2.1	Aufmerksamkeitsmodelle	306
10.2.2	Relationale Argumentation	308
10.2.3	Self-Attention-Modelle	314
10.3	Implementierung der Self-Attention für MNIST	317
10.3.1	Transformiertes MNIST	317
10.3.2	Das relationale Modul	319
10.3.3	Tensorverjüngungen und Einstein-Notation	322
10.3.4	Training des relationalen Moduls	326
10.4	Multi-Head Attention und relationales DQN	330
10.5	Double Q-Learning	338
10.6	Training und Aufmerksamkeitsvisualisierung	339

10.6.1	Maximum Entropy Learning	343
10.6.2	Curriculum Learning	344
10.6.3	Visualisierung von Aufmerksamkeitsgewichten	344
10.7	Zusammenfassung	348
11	Abschließend: Rückblick und Fahrplan	351
11.1	Was haben wir gelernt?	351
11.2	Die unergründeten Themen des Deep Reinforcement Learning	353
11.2.1	Priorisiertes auf Erfahrung basiertes Wiederholungsspiel	354
11.2.2	Proximal-Policy-Optimierung (PPO)	354
11.2.3	Hierarchisches Reinforcement Learning und das Options Framework	355
11.2.4	Modellbasierte Planung	356
11.2.5	Monte-Carlo-Baumsuche (MCTS)	357
11.3	Das Ende	358
	Anhang: Mathematik, Deep Learning, PyTorch	359
A.1	Lineare Algebra	359
A.2	Analysis	361
A.3	Deep Learning	366
A.4	PyTorch	367
	Literaturverzeichnis	371
	Stichwortverzeichnis	375