

Inhaltsverzeichnis

	Über den Autor	17
	Über die Korrektoren	17
	Über den Fachkorrektor der deutschen Ausgabe	18
	Einleitung	19
Teil I	Grundlagen des Reinforcement Learnings	24
1	Was ist Reinforcement Learning?	25
1.1	Überwachtes Lernen	25
1.2	Unüberwachtes Lernen	26
1.3	Reinforcement Learning	26
1.4	Herausforderungen beim Reinforcement Learning	28
1.5	RL-Formalismen	28
1.5.1	Belohnung	29
1.5.2	Der Agent	31
1.5.3	Die Umgebung	31
1.5.4	Aktionen	31
1.5.5	Beobachtungen	32
1.6	Die theoretischen Grundlagen des Reinforcement Learnings	34
1.6.1	Markov-Entscheidungsprozesse	35
1.6.2	Markov-Prozess	35
1.6.3	Markov-Belohnungsprozess	39
1.6.4	Aktionen hinzufügen	42
1.6.5	Policy	44
1.7	Zusammenfassung	45
2	OpenAI Gym	47
2.1	Aufbau des Agenten	47
2.2	Anforderungen an Hard- und Software	50
2.3	OpenAI-Gym-API	51
2.3.1	Aktionsraum	52
2.3.2	Beobachtungsraum	52
2.3.3	Die Umgebung	54
2.3.4	Erzeugen der Umgebung	55
2.3.5	Die CartPole-Sitzung	57
2.4	Ein CartPole-Agent nach dem Zufallsprinzip	59

2.5	Zusätzliche Gym-Funktionalität: Wrapper und Monitor	60
2.5.1	Wrapper	61
2.5.2	Monitor	63
2.6	Zusammenfassung	66
3	Deep Learning mit PyTorch	67
3.1	Tensoren	67
3.1.1	Tensoren erzeugen	68
3.1.2	Skalare Tensoren	70
3.1.3	Tensor-Operationen	71
3.1.4	GPU-Tensoren	71
3.2	Gradienten	72
3.2.1	Tensoren und Gradienten	74
3.3	NN-Bausteine.	76
3.4	Benutzerdefinierte Schichten.	78
3.5	Verlustfunktionen und Optimierer	80
3.5.1	Verlustfunktionen.	81
3.5.2	Optimierer	81
3.6	Monitoring mit TensorBoard	83
3.6.1	Einführung in TensorBoard.	84
3.6.2	Plotten	85
3.7	Beispiel: GAN für Bilder von Atari-Spielen.	87
3.8	PyTorch Ignite	92
3.8.1	Konzepte	93
3.9	Zusammenfassung	97
4	Das Kreuzentropie-Verfahren.	99
4.1	Klassifikation von RL-Verfahren	99
4.2	Kreuzentropie in der Praxis	100
4.3	Kreuzentropie beim CartPole	102
4.4	Kreuzentropie beim FrozenLake	111
4.5	Theoretische Grundlagen des Kreuzentropie-Verfahrens	118
4.6	Zusammenfassung	119
Teil II	Wertebasierte Verfahren	120
5	Tabular Learning und das Bellman'sche Optimalitätsprinzip.	121
5.1	Wert, Zustand und Optimalität	121
5.2	Das Bellman'sche Optimalitätsprinzip	123
5.3	Aktionswert	126
5.4	Wertiteration	128
5.5	Wertiteration in der Praxis	130
5.6	Q-Learning in der FrozenLake-Umgebung.	136
5.7	Zusammenfassung	138

6	Deep Q-Networks	139
6.1	Wertiteration in der Praxis	139
6.2	Tabular Q-Learning	140
6.3	Deep Q-Learning	145
	6.3.1 Interaktion mit der Umgebung	147
	6.3.2 SGD-Optimierung	147
	6.3.3 Korrelation der Schritte	148
	6.3.4 Die Markov-Eigenschaft	148
	6.3.5 Die endgültige Form des DQN-Trainings	149
6.4	DQN mit Pong	150
	6.4.1 Wrapper	151
	6.4.2 DQN-Modell	156
	6.4.3 Training	158
	6.4.4 Ausführung und Leistung	167
	6.4.5 Das Modell in Aktion	170
6.5	Weitere Möglichkeiten	172
6.6	Zusammenfassung	173
7	Allgemeine RL-Bibliotheken	175
7.1	Warum RL-Bibliotheken?	175
7.2	Die PTAN-Bibliothek	176
	7.2.1 Aktionsselektoren	177
	7.2.2 Der Agent	179
	7.2.3 Quelle der Erfahrungswerte	183
	7.2.4 Replay Buffer für Erfahrungswerte	189
	7.2.5 Die TargetNet-Klasse	191
	7.2.6 Hilfsfunktionen für Ignite	193
7.3	Lösung der CartPole-Umgebung mit PTAN	194
7.4	Weitere RL-Bibliotheken	196
7.5	Zusammenfassung	197
8	DQN-Erweiterungen	199
8.1	Einfaches DQN	199
	8.1.1 Die Bibliothek common	200
	8.1.2 Implementierung	205
	8.1.3 Ergebnisse	207
8.2	N-Schritt-DQN	208
	8.2.1 Implementierung	211
	8.2.2 Ergebnisse	211
8.3	Double DQN	212
	8.3.1 Implementierung	213
	8.3.2 Ergebnisse	215
8.4	Verrauschte Netze	216
	8.4.1 Implementierung	217
	8.4.2 Ergebnisse	219

8.5	Priorisierter Replay Buffer	220
8.5.1	Implementierung	221
8.5.2	Ergebnisse	225
8.6	Rivalisierendes DQN	227
8.6.1	Implementierung	228
8.6.2	Ergebnisse	229
8.7	Kategoriales DQN	230
8.7.1	Implementierung	232
8.7.2	Ergebnisse	239
8.8	Alles miteinander kombinieren	241
8.8.1	Ergebnisse	242
8.9	Zusammenfassung	243
8.10	Quellenangaben	244
9	Beschleunigung von RL-Verfahren	245
9.1	Die Bedeutung der Geschwindigkeit	245
9.2	Der Ausgangspunkt	248
9.3	Der Berechnungsgraph in PyTorch	250
9.4	Mehrere Umgebungen	252
9.5	Spielen und Trainieren in separaten Prozessen	255
9.6	Optimierung der Wrapper	259
9.7	Zusammenfassung der Benchmarks	265
9.8	Atari-Emulation: CuLE	265
9.9	Zusammenfassung	266
9.10	Quellenangaben	266
10	Aktienhandel per Reinforcement Learning	267
10.1	Börsenhandel	267
10.2	Daten	268
10.3	Aufgabenstellungen und Grundsatzentscheidungen	269
10.4	Die Handelsumgebung	270
10.5	Modelle	279
10.6	Trainingscode	281
10.7	Ergebnisse	281
10.7.1	Das Feedforward-Modell	281
10.7.2	Das Faltungsmodell	287
10.8	Weitere Möglichkeiten	288
10.9	Zusammenfassung	289
Teil III	Policybasierte Verfahren	290
11	Eine Alternative: Policy Gradients	291
11.1	Werte und Policy	291
11.1.1	Warum Policy?	292

11.1.2	Repräsentation der Policy	292
11.1.3	Policy Gradients	293
11.2	Das REINFORCE-Verfahren	294
11.2.1	Das CartPole-Beispiel	295
11.2.2	Ergebnisse	299
11.2.3	Policybasierte und wertebasierte Verfahren	300
11.3	Probleme mit REINFORCE	301
11.3.1	Notwendigkeit vollständiger Episoden	301
11.3.2	Große Varianz der Gradienten.	302
11.3.3	Exploration.	302
11.3.4	Korrelation zwischen Beispielen	303
11.4	PG mit CartPole	303
11.4.1	Implementierung	303
11.4.2	Ergebnisse	306
11.5	PG mit Pong.	310
11.5.1	Implementierung	311
11.5.2	Ergebnisse	312
11.6	Zusammenfassung	313
12	Das Actor-Critic-Verfahren	315
12.1	Verringern der Varianz	315
12.2	Varianz der CartPole-Umgebung	317
12.3	Actor-Critic	320
12.4	A2C mit Pong.	322
12.5	A2C mit Pong: Ergebnisse	328
12.6	Optimierung der Hyperparameter	331
12.6.1	Lernrate	332
12.6.2	Beta.	333
12.6.3	Anzahl der Umgebungen	333
12.6.4	Batchgröße.	333
12.7	Zusammenfassung	333
13	Asynchronous Advantage Actor Critic	335
13.1	Korrelation und Stichprobeneffizienz	335
13.2	Ein weiteres A zu A2C hinzufügen	336
13.3	Multiprocessing in Python	339
13.4	A3C mit Datenparallelität	339
13.4.1	Implementierung	339
13.4.2	Ergebnisse	346
13.5	A3C mit Gradientenparallelität.	347
13.5.1	Implementierung	348
13.5.2	Ergebnisse	353
13.6	Zusammenfassung	354
14	Chatbot-Training per Reinforcement Learning	355
14.1	Chatbots – ein Überblick.	355

14.2	Chatbot-Training	356
14.3	Grundlagen der Verarbeitung natürlicher Sprache	357
	14.3.1 Rekurrente neuronale Netze	357
	14.3.2 Wort-Embeddings	359
	14.3.3 Encoder-Decoder	360
14.4	Seq2Seq-Training	361
	14.4.1 Log-Likelihood-Training	361
	14.4.2 Der BLEU-Score	363
	14.4.3 RL und Seq2Seq	364
	14.4.4 Self-critical Sequence Training	365
14.5	Das Chatbot-Beispiel	366
	14.5.1 Aufbau des Beispiels	366
	14.5.2 Module: cornell.py und data.py	367
	14.5.3 BLEU-Score und utils.py	368
	14.5.4 Modell	369
14.6	Daten überprüfen	376
14.7	Training: Kreuzentropie	378
	14.7.1 Implementierung	378
	14.7.2 Ergebnisse	382
14.8	Training: Self-critical Sequence Training (SCST)	385
	14.8.1 Implementierung	385
	14.8.2 Ergebnisse	392
14.9	Tests der Modelle mit Daten	395
14.10	Telegram-Bot	397
14.11	Zusammenfassung	401
15	Die TextWorld-Umgebung	403
15.1	Interactive Fiction	403
15.2	Die Umgebung	406
	15.2.1 Installation	407
	15.2.2 Spiel erzeugen	407
	15.2.3 Beobachtungs- und Aktionsräume	409
	15.2.4 Zusätzliche Informationen	411
15.3	Einfaches DQN	414
	15.3.1 Vorverarbeitung von Beobachtungen	416
	15.3.2 Embeddings und Encoder	421
	15.3.3 DQN-Modell und Agent	424
	15.3.4 Trainingscode	426
	15.3.5 Trainingsergebnisse	426
15.4	Das Modell für den Befehlsgenerator	431
	15.4.1 Implementierung	433
	15.4.2 Ergebnisse des Pretrainings	437
	15.4.3 DQN-Trainingscode	439
	15.4.4 Ergebnis des DQN-Trainings	441
15.5	Zusammenfassung	442

16	Navigation im Web	443
16.1	Webnavigation	443
	16.1.1 Browserautomatisierung und RL	444
	16.1.2 Mini World of Bits	445
16.2	OpenAI Universe	446
	16.2.1 Installation	447
	16.2.2 Aktionen und Beobachtungen	448
	16.2.3 Umgebung erzeugen	449
	16.2.4 MiniWoB-Stabilität	451
16.3	Einfaches Anklicken	451
	16.3.1 Aktionen auf dem Gitter	452
	16.3.2 Übersicht der Beispiele	453
	16.3.3 Modell	454
	16.3.4 Trainingscode	455
	16.3.5 Container starten	460
	16.3.6 Trainingsprozess	461
	16.3.7 Überprüfen der erlernten Policy	464
	16.3.8 Probleme mit einfachem Anklicken	465
16.4	Demonstrationen durch den Menschen	467
	16.4.1 Aufzeichnung von Demonstrationen	468
	16.4.2 Aufzeichnungsformat	470
	16.4.3 Training durch Demonstration	473
	16.4.4 Ergebnisse	474
	16.4.5 Tic-Tac-Toe	478
16.5	Hinzufügen von Beschreibungstext	480
	16.5.1 Implementierung	481
	16.5.2 Ergebnisse	486
16.6	Weitere Möglichkeiten	489
16.7	Zusammenfassung	489

Teil IV Fortgeschrittene Verfahren und Techniken

17	Stetige Aktionsräume	491
17.1	Wozu stetige Aktionsräume?	491
17.2	Aktionsraum	492
17.3	Umgebungen	492
17.4	Das A2C-Verfahren	495
	17.4.1 Implementierung	496
	17.4.2 Ergebnisse	499
	17.4.3 Modelle verwenden und Videos aufzeichnen	501
17.5	Deterministisches Policy-Gradienten-Verfahren	502
	17.5.1 Exploration	503
	17.5.2 Implementierung	504
	17.5.3 Ergebnisse	509

17.5.4	Videos aufzeichnen	511
17.6	Distributional Policy Gradients	511
17.6.1	Architektur	512
17.6.2	Implementierung	512
17.6.3	Ergebnisse	517
17.6.4	Videoaufzeichnung.	519
17.7	Weitere Möglichkeiten	519
17.8	Zusammenfassung	519
18	RL in der Robotik	521
18.1	Roboter und Robotik.	521
18.1.1	Komplexität von Robotern	523
18.1.2	Hardware.	524
18.1.3	Plattform	525
18.1.4	Sensoren	526
18.1.5	Aktuatoren.	528
18.1.6	Rahmen.	528
18.2	Ein erstes Trainingsziel.	532
18.3	Emulator und Modell	534
18.3.1	Definitionsdatei des Modells	535
18.3.2	Die robot-Klasse	539
18.4	DDPG-Training und Ergebnisse	545
18.5	Steuerung der Hardware	548
18.5.1	MicroPython	548
18.5.2	Handhabung von Sensoren	552
18.5.3	Servos ansteuern.	565
18.5.4	Einrichtung des Modells auf der Hardware	569
18.5.5	Alles kombinieren.	577
18.6	Experimente mit der Policy	580
18.7	Zusammenfassung	581
19	Trust Regions – PPO, TRPO, ACKTR und SAC.	583
19.1	Roboschool.	584
19.2	Standard-A2C-Verfahren	584
19.2.1	Implementierung	584
19.2.2	Ergebnisse	586
19.2.3	Videoaufzeichnungen.	590
19.3	Proximal Policy Optimization (PPO)	590
19.3.1	Implementierung	591
19.3.2	Ergebnisse	595
19.4	Trust Region Policy Optimization (TRPO)	597
19.4.1	Implementierung	597
19.4.2	Ergebnisse	599
19.5	Advantage Actor-Critic mit Kronecker-Factored Trust Region (ACKTR)	600
19.5.1	Implementierung	601

19.5.2	Ergebnisse	601
19.6	Soft-Actor-Critic (SAC)	602
19.6.1	Implementierung	603
19.6.2	Ergebnisse	605
19.7	Zusammenfassung	607
20	Blackbox-Optimierung beim Reinforcement Learning	609
20.1	Blackbox-Verfahren	609
20.2	Evolutionsstrategien (ES)	610
20.3	ES mit CartPole	611
20.3.1	Ergebnisse	616
20.4	ES mit HalfCheetah	617
20.4.1	Implementierung	618
20.4.2	Ergebnisse	622
20.5	Genetische Algorithmen (GA)	624
20.6	GA mit CartPole	624
20.6.1	Ergebnisse	626
20.7	GA-Optimierung	627
20.7.1	Deep GA	628
20.7.2	Novelty Search	628
20.8	GA mit HalfCheetah	628
20.8.1	Ergebnisse	631
20.9	Zusammenfassung	633
20.10	Quellenangaben	633
21	Fortgeschrittene Exploration	635
21.1	Die Bedeutung der Exploration	635
21.2	Was ist das Problem beim ϵ -Greedy-Ansatz?	636
21.3	Alternative Explorationsverfahren	639
21.3.1	Verrauschte Netze	639
21.3.2	Zählerbasierte Verfahren	640
21.3.3	Vorhersagebasierte Verfahren	641
21.4	MountainCar-Experimente	641
21.4.1	Das DQN-Verfahren mit ϵ -Greedy-Ansatz	643
21.4.2	Das DQN-Verfahren mit verrauschten Netzen	644
21.4.3	Das DQN-Verfahren mit Zustandszählern	646
21.4.4	Das PPO-Verfahren	649
21.4.5	Das PPO-Verfahren mit verrauschten Netzen	652
21.4.6	Das PPO-Verfahren mit zählerbasierter Exploration	654
21.4.7	Das PPO-Verfahren mit Netz-Destillation	656
21.5	Atari-Experimente	658
21.5.1	Das DQN-Verfahren mit ϵ -Greedy-Ansatz	659
21.5.2	Das klassische PPO-Verfahren	660
21.5.3	Das PPO-Verfahren mit Netz-Destillation	661
21.5.4	Das PPO-Verfahren mit verrauschten Netzen	662

21.6	Zusammenfassung	663
21.7	Quellenangaben.	663
22	Jenseits modellfreier Verfahren – Imagination	665
22.1	Modellbasierte Verfahren	665
22.1.1	Modellbasierte und modellfreie Verfahren.	665
22.2	Unzulänglichkeiten der Modelle	666
22.3	Imagination-augmented Agent	668
22.3.1	Das Umgebungsmodell	669
22.3.2	Die Rollout-Policy	670
22.3.3	Der Rollout-Encoder	670
22.3.4	Ergebnisse der Arbeit	670
22.4	I2A mit dem Atari-Spiel Breakout	670
22.4.1	Der Standard-A2C-Agent	671
22.4.2	Training des Umgebungsmodells	672
22.4.3	Der Imagination-Agent	675
22.5	Ergebnisse der Experimente.	681
22.5.1	Der Basis-Agent	681
22.5.2	Training der EM-Gewichte	683
22.5.3	Training mit dem I2A-Modell	685
22.6	Zusammenfassung	688
22.7	Quellenangaben.	688
23	AlphaGo Zero	689
23.1	Brettspiele	689
23.2	Das AlphaGo-Zero-Verfahren.	690
23.2.1	Überblick.	690
23.2.2	Monte-Carlo-Baumsuche	691
23.2.3	Self-Playing	693
23.2.4	Training und Bewertung	694
23.3	Vier-gewinnt-Bot	694
23.3.1	Spielmodell	695
23.3.2	Implementierung der Monte-Carlo-Baumsuche	697
23.3.3	Modell	702
23.3.4	Training.	705
23.3.5	Test und Vergleich	705
23.4	Vier gewinnt: Ergebnisse	706
23.5	Zusammenfassung	708
23.6	Quellenangaben.	708
24	RL und diskrete Optimierung	709
24.1	Die Reputation von Reinforcement Learnings	709
24.2	Zauberwürfel und kombinatorische Optimierung.	710
24.3	Optimalität und Gottes Zahl.	711
24.4	Ansätze zur Lösung	712

24.4.1	Datenrepräsentation	712
24.4.2	Aktionen.	712
24.4.3	Zustände	713
24.5	Trainingsvorgang.	717
24.5.1	Architektur des neuronalen Netzes	717
24.5.2	Training	718
24.6	Anwendung des Modells	719
24.7	Ergebnisse der Arbeit	721
24.8	Code	722
24.8.1	Würfel-Umgebungen	723
24.8.2	Training	727
24.8.3	Suchvorgang	729
24.9	Ergebnisse des Experiments	729
24.9.1	Der 2x2-Würfel	731
24.9.2	Der 3x3-Würfel	733
24.9.3	Weitere Verbesserungen und Experimente	734
24.10	Zusammenfassung	735
25	RL mit mehreren Agenten	737
25.1	Mehrere Agenten	737
25.1.1	Kommunikationsformen	738
25.1.2	Der RL-Ansatz	738
25.2	Die MAgent-Umgebung	738
25.2.1	Installation	739
25.2.2	Überblick	739
25.2.3	Eine zufällige Umgebung	739
25.3	Deep Q-Networks für Tiger	745
25.3.1	Training und Ergebnisse	748
25.4	Zusammenarbeit der Tiger	750
25.5	Training der Tiger und Hirsche	754
25.6	Der Kampf ebenbürtiger Akteure	755
25.7	Zusammenfassung	756
	Stichwortverzeichnis	757