



alexander ZAI  
brandon BROWN

# EINSTIEG IN DEEP REINFORCEMENT LEARNING

KI-Agenten mit Python und  
PyTorch programmieren



[GitHub-Repository](#) zum Buch

HANSER

HANSER

Alexander Zai  
Brandon Brown

# **Einstieg in Deep Reinforcement Learning**

KI-Agenten mit Python und  
PyTorch programmieren

Titel der Originalausgabe: „Deep Reinforcement Learning“, © 2020 by Manning Publications Co.

Authorized translation of the English edition. This translation is published and sold by permission of Manning Publications, the owner of all rights to publish and sell the same.

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autoren, Übersetzer und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autoren, Übersetzer und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Copyright für die deutsche Ausgabe:

© 2020 Carl Hanser Verlag, [www.hanser-fachbuch.de](http://www.hanser-fachbuch.de)

Lektorat: Sylvia Hasselbach

Copy editing: Christian Schneider, Traunstein

Fachlektorat: Peter Seeberg, Neubiberg

Layout: Manuela Treindl, Fürth

Umschlagdesign: Marc Müller-Bremer, München, [www.rebranding.de](http://www.rebranding.de)

Umschlagrealisation: Max Kostopoulos

Titelmotiv: © [shutterstock.com/ioa](http://shutterstock.com/ioa)

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Print-ISBN: 978-3-446-45900-7

E-Book-ISBN: 978-3-446-46608-1

E-Pub-ISBN: 978-3-446-46609-8

**Inhalt**

**Titelei**

**Impressum**

**Inhalt**

**Vorwort**

**Danksagung**

**Die Autoren**

**Über dieses Buch**

**Teil I: Grundlagen**

**1 Was ist Reinforcement Learning?**

## 1.1 Das „Tiefe“ beim Deep Reinforcement Learning

## 1.2 Reinforcement Learning

## 1.3 Dynamische Programmierung versus Monte Carlo

## 1.4 Das Reinforcement-Learning-Framework

## 1.5 Was kann ich mit Reinforcement Learning anfangen?

## 1.6 Warum Deep Reinforcement Learning?

## 1.7 Unser didaktisches Werkzeug: String-Diagramme

## 1.8 Wie geht es weiter?

## 1.9 Zusammenfassung

# 2 Modellierung von Reinforcement-Learning-Problemen: Markov Decision Processes

## 2.1 String-Diagramme und unsere Lehrmethoden

## 2.2 Multi-Armed-Bandit-Problem

### 2.2.1 Erkundung und Ausnutzung

2.2.2 Epsilon-greedy-Strategie

2.2.3 Softmax-Auswahlverfahren

## **2.3 Anwendung von Banditen zur Optimierung von Anzeigenplatzierungen**

2.3.1 Kontextabhängige Banditen

2.3.2 Zustände, Aktionen, Belohnungen

## **2.4 Netzwerke mit PyTorch erstellen**

2.4.1 Automatische Differenzierung

2.4.2 Modelle erstellen

## **2.5 Lösen kontextabhängiger Banditen**

## **2.6 Die Markov-Eigenschaft**

## **2.7 Vorhersage zukünftiger Belohnungen: Value- und Policy-Funktionen**

2.7.1 Policy-Funktionen

2.7.2 Optimale Policy

2.7.3 Wert-Funktionen

## 2.8 Zusammenfassung

# 3 Vorhersage der besten Zustände und Aktionen: Deep Q-Networks

## 3.1 Die Q-Funktion

## 3.2 Navigieren mit Q-Learning

### 3.2.1 Was ist Q-Learning?

### 3.2.2 Gridworld bewältigen

### 3.2.3 Hyperparameter

### 3.2.4 Diskontierungsfaktor

### 3.2.5 Aufbau des Netzwerks

### 3.2.6 Einführung in die Gridworld-Spiele-Engine

### 3.2.7 Ein neuronales Netz als Q-Funktion

## 3.3 Verhinderung katastrophalen Vergessens: Auf Erfahrung basiertes Wiederholungsspiel

### 3.3.1 Katastrophales Vergessen

### 3.3.2 Auf Erfahrung basiertes Wiederholungsspiel



## **3.4 Verbesserung der Stabilität mit einem Target Network**

### **3.4.1 Lerninstabilität**

## **3.5 Rückblick**

## **3.6 Zusammenfassung**

# **4 Lernen, die beste Policy auszuwählen: Policy-Gradient-Methoden**

## **4.1 Policy-Funktion mit neuronalen Netzen**

### **4.1.1 Neuronales Netz als Policy-Funktion**

### **4.1.2 Stochastischer Policy-Gradient**

### **4.1.3 Erkundung**

## **4.2 Verstärkung guter Aktionen: Der Policy-Gradient-Algorithmus**

### **4.2.1 Definieren eines Ziels**

### **4.2.2 Verstärkung der Aktionen**

### **4.2.3 Log-Wahrscheinlichkeit**

### **4.2.4 Anerkennungszuweisung**

## **4.3 Arbeiten mit OpenAI Gym**

### **4.3.1 CartPole**

### **4.3.2 Die OpenAI Gym-API**

## **4.4 Der REINFORCE-Algorithmus**

### **4.4.1 Erstellen des Policy-Netzwerks**

### **4.4.2 Interaktion des Agenten mit der Umgebung**

### **4.4.3 Das Modell trainieren**

### **4.4.4 Die vollständige Trainingsschleife**

### **4.4.5 Schlussfolgerung**

## **4.5 Zusammenfassung**

# **5 Bewältigung komplexerer Probleme mit Actor-Critic-Methoden**

## **5.1 Die Kombination von Wert- und Policy-Funktion**

## **5.2 Verteiltes Training**

## **5.3 Advantage-Actor-Critic**

## 5.4 N-step-Actor-Critic

## 5.5 Zusammenfassung

# Teil II: Darüber hinaus

## 6 Alternative Optimierungsmethoden: Evolutionäre Algorithmen

### 6.1 Ein anderer Ansatz für das Reinforcement Learning

### 6.2 Reinforcement Learning mit Evolutionsstrategien

#### 6.2.1 Evolution in der Theorie

#### 6.2.2 Entwicklung in der Praxis

### 6.3 Ein genetischer Algorithmus für CartPole

### 6.4 Vor- und Nachteile von evolutionären Algorithmen

#### 6.4.1 Evolutionäre Algorithmen erforschen mehr

#### 6.4.2 Evolutionäre Algorithmen sind unglaublich testintensiv

#### 6.4.3 Simulatoren

## **6.5 Evolutionäre Algorithmen als skalierbare Alternative**

**6.5.1 Skalierung evolutionärer Algorithmen**

**6.5.2 Parallele vs. serielle Verarbeitung**

**6.5.3 Skalierungseffizienz**

**6.5.4 Kommunikation zwischen Knoten**

**6.5.5 Lineare Skalierung**

**6.5.6 Auf Gradienten basierte Ansätze zur Skalierung**

## **6.6 Zusammenfassung**

# **7 Verteilungs-DQN: Die ganze Geschichte**

**7.1 Was ist falsch an Q-Learning?**

**7.2 Wahrscheinlichkeitsrechnung und Statistik**

**7.2.1 A priori und A posteriori**

**7.2.2 Erwartung und Varianz**

**7.3 Die Bellman-Gleichung**

**7.4 Verteilungs-Q-Learning**

7.4.1 Darstellung einer Wahrscheinlichkeitsverteilung in Python

7.4.2 Implementierung des Dist-DQN

7.5 Vergleich von Wahrscheinlichkeitsverteilungen

7.6 Dist-DQN auf simulierten Daten

7.7 Verwendung von Verteilungs-Q-Learning zum Spielen von Freeway

7.8 Zusammenfassung

## 8 Von Neugierde getriebene Erkundung

8.1 Mit prädiktiver Kodierung gegen spärliche Belohnungen vorgehen

8.2 Vorhersage der inversen Dynamik

8.3 Implementierung von Super Mario Bros.

8.4 Vorverarbeitung und das Q-Netz

8.5 Einrichten des Q-Netz und der Policy-Funktion

8.6 Intrinsisches Neugierde-Modul

## 8.7 Alternative intrinsische Belohnungsmechanismen

## 8.8 Zusammenfassung

# 9 Lernen mit Multi-Agent Reinforcement Learning

## 9.1 Von einem zu vielen Agenten

## 9.2 Nachbarschafts-Q-Learning

## 9.3 Das 1D-Ising-Modell

## 9.4 Mean-Field-Q-Learning und das 2D-Ising-Modell

## 9.5 Gemischte kooperativ-konkurrierende Spiele

## 9.6 Zusammenfassung

# 10 Interpretierbares Reinforcement Learning: Aufmerksamkeitsmodelle und relationale Modelle

## 10.1 Interpretierbarkeit von Machine Learning mit Aufmerksamkeit und relationalen Verzerrungen

10.1.1 Invarianz und Äquivarianz

## **10.2 Relationale Argumentation mit Aufmerksamkeit**

10.2.1 Aufmerksamkeitsmodelle

10.2.2 Relationale Argumentation

10.2.3 Self-Attention-Modelle

## **10.3 Implementierung der Self-Attention für MNIST**

10.3.1 Transformiertes MNIST

10.3.2 Das relationale Modul

10.3.3 Tensorverjüngungen und Einstein-Notation

10.3.4 Training des relationalen Moduls

## **10.4 Multi-Head Attention und relationales DQN**

## **10.5 Double Q-Learning**

## **10.6 Training und Aufmerksamkeitsvisualisierung**

10.6.1 Maximum Entropy Learning

10.6.2 Curriculum Learning

10.6.3 Visualisierung von Aufmerksamkeitsgewichten

## 10.7 Zusammenfassung

# 11 Abschließend: Rückblick und Fahrplan

## 11.1 Was haben wir gelernt?

## 11.2 Die unergründeten Themen des Deep Reinforcement Learning

### 11.2.1 Priorisiertes auf Erfahrung basiertes Wiederholungsspiel

### 11.2.2 Proximal-Policy-Optimierung (PPO)

### 11.2.3 Hierarchisches Reinforcement Learning und das Options Framework

### 11.2.4 Modellbasierte Planung

### 11.2.5 Monte-Carlo-Baumsuche (MCTS)

## 11.3 Das Ende

# Anhang: Mathematik, Deep Learning, PyTorch

## A.1 Lineare Algebra

## A.2 Analysis



A.3 Deep Learning

A.4 PyTorch

Literaturverzeichnis

# Vorwort

Deep Reinforcement Learning rückte 2015 ins Rampenlicht, als DeepMind einen Algorithmus entwickelte, der in der Lage ist, eine Reihe von Atari-2600-Spielen besser als der Mensch zu spielen. Die künstliche Intelligenz schien endlich echte Fortschritte zu machen, und wir wollten unseren Teil dazu beitragen.

Wir haben beide einen Hintergrund im Software-Engineering und Interesse an den Neurowissenschaften, und wir interessieren uns schon seit Langem für das breitere Feld der künstlichen Intelligenz (tatsächlich hat einer von uns sein erstes neuronales Netz noch vor der High School in C# geschrieben). Diese frühen Erfahrungen führten zu keinem anhaltenden Interesse, da sie in die Zeit vor der Revolution des Deep Learning um das Jahr 2012 fielen, als die überragende Leistung des Deep Learning deutlich wurde. Aber nachdem wir die erstaunlichen Erfolge des Deep Learning gesehen hatten, wandten wir uns wieder den aufregenden und aufblühenden Bereichen des Deep Learning und dann des Deep Reinforcement Learning zu, und wir beide haben das Machine Learning auf die eine oder andere Weise stärker in unsere berufliche Laufbahn integriert. Alex wechselte in eine Karriere als Machine Learning Engineer und machte sich

an wenig bekannten Orten wie Amazon einen Namen, und Brandon begann, Machine Learning in der akademischen neurowissenschaftlichen Forschung einzusetzen. Als wir uns in Deep Reinforcement Learning vertieften, mussten wir uns durch Dutzende von Lehrbüchern und Forschungsartikeln kämpfen, wobei wir uns in fortgeschrittene Mathematik und die Theorie des maschinellen Lernens vertieften. Wir stellten jedoch fest, dass die Grundlagen des Deep Reinforcement Learning durchaus zugänglich sind, wenn man einen Hintergrund im Software-Engineering hat. Die gesamte Mathematik lässt sich leicht in eine Sprache übersetzen, die für jeden Programmierer gut lesbar ist.

Wir begannen, über die Dinge zu bloggen, die wir in der Welt des Machine Learning lernten, und über Projekte, die wir bei unserer Arbeit verwendeten. Am Ende bekamen wir eine ganze Menge positives Feedback, was uns auf die Idee brachte, gemeinsam an diesem Buch zu arbeiten. Wir sind der Meinung, dass die meisten Ressourcen, die es zum Lernen komplexer Dinge gibt, entweder zu einfach sind und die fesselndsten Aspekte des Themas auslassen oder für Menschen ohne fortgeschrittenen mathematischen Hintergrund unzugänglich sind. Mit diesem Buch haben wir uns bemüht, ein für Experten geschriebenes Kompendium in einen Kurs für diejenigen zu übersetzen, die nichts weiter als einen Programmierhintergrund und einige Grundkenntnisse über neuronale Netze haben. Wir wenden einige neuartige Lehrmethoden an, von denen wir glauben, dass sie unser Buch auszeichnen und zu einem viel schnelleren Verständnis führen. Wir fangen bei den Grundlagen an, und am Ende werden Sie innovative Algorithmen implementieren, die von industriellen Forschungsprojekten wie DeepMind und OpenAI sowie von leistungsstarken akademischen Einrichtungen

wie dem Berkeley Artificial Intelligence Research (BAIR) Lab und dem University College London entwickelt wurden.

## Danksagung

Dieses Buch hat viel länger gedauert, als wir erwartet hatten, und wir haben unseren Lektorinnen Candace West und Susanna Kline viel zu verdanken. Sie haben uns in jeder Phase des Prozesses geholfen und uns auf Kurs gehalten. Wenn man ein Buch schreibt, muss man viele Details im Auge behalten, und ohne das professionelle und unterstützende Lektorenteam wären wir ins Schwimmen geraten.

Unser Dank gilt auch unseren technischen Redakteuren Marc-Philippe Huget und Al Krinker sowie allen Gutachtern, die sich die Zeit genommen haben, unser Manuskript zu lesen und uns entscheidendes Feedback zu geben. Unser besonderer Dank gilt den Gutachtern Al Rahimi, Ariel Gamiño, Claudio Bernardo Rodriguez, David Krief, Dr. Brett Pennington, Ezra Joel Schroeder, George L. Gaines, Godfred Asamoah, Helmut Hauschild, Ike Okonkwo, Jonathan Wood, Kalyan Reddy, M. Edward (Ed) Borasky, Michael Haller, Nadia Noori, Satyajit Sarangi und Tobias Kaatz. Wir möchten auch allen bei Manning danken, die an diesem Projekt mitgearbeitet haben: Karen Miller (Developmental Editor), Ivan Martinović (Review Editor), Deirdre Hiam (Project Editor), Andy Carroll (Copy Editor) und Jason Everett (Proofreader).

In der heutigen Zeit werden viele Bücher über verschiedene Online-Dienste im Selbstverlag herausgegeben, und diese Option hat uns zunächst gereizt. Nachdem wir jedoch diesen ganzen

Prozess durchlaufen haben, können wir den enormen Wert eines professionellen Lektorats erkennen. Insbesondere danken wir dem Copy Editor Andy Carroll für sein aufschlussreiches Feedback, das die Klarheit des Textes wesentlich verbessert hat.

Alex bedankt sich bei seinem PI Jamie, der ihn bereits während seines Grundstudiums an Machine Learning herangeführt hat.

Brandon dankt seiner Frau Xinzhu dafür, dass sie seine langen Nächte am Schreibtisch und seine Abwesenheit von der Familie erduldet und ihm zwei wunderbare Kinder, Isla und Avin, geschenkt hat.

## Die Autoren

ALEX ZAI arbeitete als Chief Technology Officer bei Codesmith, einem immersiven Coding-Bootcamp, wo er weiterhin als Technical Advisor tätig ist, sowie als Software-Ingenieur bei Uber und als Machine Learning Engineer bei Banjo und Amazon. Darüber hinaus wirkt er am Open-Source Deep-Learning-Framework Apache MXNet mit. Er hat zwei Unternehmen mitbegründet, von denen eines durch Y Combinator gefördert wurde.

BRANDON BROWN beschäftigte sich von klein auf mit dem Programmieren und arbeitete während des Studiums als Teilzeit-Software-Ingenieur, entschied sich aber schließlich für eine Karriere in der Medizin; nebenbei arbeitete er als Software-Ingenieur im Bereich der Gesundheitstechnologie. Heute ist er Arzt und verfolgt seine Forschungsinteressen im Bereich der

Computational Psychiatry, inspiriert durch Deep Reinforcement Learning.

# Über dieses Buch

## Wer dieses Buch lesen sollte

*Einstieg in Deep Reinforcement Learning* ist ein Kurs, der entwickelt wurde, um Sie von den grundlegenden Konzepten des Reinforcement Learning bis hin zur Implementierung der neuesten Algorithmen zu führen. Als Kurs konzentriert sich jedes Kapitel auf ein Hauptprojekt, das das jeweilige Thema oder Konzept veranschaulichen soll. Wir haben jedes Projekt so konzipiert, dass es effizient auf einem neueren Laptop ausgeführt werden kann; wir erwarten nicht, dass Sie Zugriff auf teure GPUs oder Cloud-Computing-Ressourcen haben (auch wenn der Zugriff auf diese Ressourcen einiges beschleunigt).

Dieses Buch richtet sich an Personen mit einem Programmierhintergrund, insbesondere mit Grundkenntnissen in Python, und an Personen, die zumindest ein grundlegendes Verständnis von neuronalen Netzen haben (auch bekannt als Deep Learning). Mit „grundlegendem Verständnis“ meinen wir, dass Sie schon einmal versucht haben, ein einfaches neuronales Netz in Python zu implementieren, auch wenn Sie nicht ganz verstanden haben, was tief im Inneren vor sich geht. Obwohl sich dieses Buch auf die Verwendung von neuronalen Netzen in Bezug

auf das Reinforcement Learning konzentriert, werden Sie wahrscheinlich auch viel Neues über Deep Learning im Allgemeinen lernen, das sich auf andere Probleme außerhalb des Reinforcement Learning anwenden lässt, sodass Sie kein Experte für Deep Learning sein müssen, bevor Sie sich in das Deep Reinforcement Learning stürzen.

## Wie dieses Buch organisiert ist: Ein Fahrplan

Das Buch hat zwei Teile mit insgesamt [11 Kapiteln](#).

**Teil I** erläutert die Grundlagen des Deep Reinforcement Learning.

- [Kapitel 1](#) bietet eine Einführung in Deep Learning, Reinforcement Learning und die Verbindung von beidem zum Deep Reinforcement Learning.
- [Kapitel 2](#) führt in die grundlegenden Konzepte des Reinforcement Learning ein, die im weiteren Verlauf des Buches immer wieder auftauchen werden. Wir implementieren auch unseren ersten angewandten Reinforcement-Learning-Algorithmus.
- [Kapitel 3](#) stellt Deep Q-Learning vor, eine der beiden Hauptklassen von Deep-Reinforcement-Algorithmen. Dabei handelt es sich um den Algorithmus, mit dem DeepMind im Jahr 2015 bei vielen Atari 2600 Spielen die Leistung von Menschen übertroffen hat.
- [Kapitel 4](#) beschreibt die andere Hauptklasse von Deep-Reinforcement-Learning-Algorithmen, die Policy-Gradient-Methoden. Wir verwenden sie, um einen Algorithmus für ein einfaches Spiel zu trainieren.



- [Kapitel 5](#) zeigt, wie wir Deep Q-Learning aus [Kapitel 3](#) und Policy-Gradient-Methoden aus [Kapitel 4](#) zu einer kombinierten Klasse von Algorithmen, den sogenannten Actor-Critic-Algorithmen, verbinden können.

**Teil II** baut auf den Grundlagen auf, die wir in [Teil I](#) gelegt haben, um die größten Fortschritte der letzten Jahre im Bereich des Deep Reinforcement Learning zu behandeln.

- [Kapitel 6](#) zeigt, wie evolutionäre Algorithmen, die Prinzipien der biologischen Evolution nutzen, implementiert werden können, um neuronale Netze zu trainieren.
- [Kapitel 7](#) beschreibt eine Methode zur signifikanten Verbesserung der Leistung des Deep Q-Learning durch die Einbeziehung probabilistischer Konzepte.
- [Kapitel 8](#) stellt eine Methode vor, wie man Reinforcement-Learning-Algorithmen ein Gefühl der Neugierde vermittelt, damit sie ihre Umgebung ohne externe Hinweise erforschen können.
- [Kapitel 9](#) zeigt, wie wir das, was wir beim Trainieren von Reinforcement-Learning-Algorithmen mit einem Agenten gelernt haben, auf Systeme mit mehreren interagierenden Agenten ausweiten können.
- [Kapitel 10](#) beschreibt, wie man Deep-Reinforcement-Learning-Algorithmen durch den Einsatz von Aufmerksamkeitsmechanismen besser interpretierbar und effizienter machen kann.
- [Kapitel 11](#) schließt das Buch mit einer Erörterung all der spannenden Bereiche des Deep Reinforcement Learning, die wir aus Platzgründen nicht behandeln konnten, die Sie aber vielleicht interessieren.

Die Kapitel in [Teil I](#) sollten der Reihe nach gelesen werden, da jedes Kapitel auf den Konzepten des vorhergehenden Kapitels aufbaut. Die Kapitel in [Teil II](#) können mehr oder weniger in beliebiger Reihenfolge angegangen werden, obwohl wir auch hier empfehlen, sie der Reihe nach zu lesen.

## Über den Code

Wie bereits erwähnt, ist dieses Buch ein Kurs, daher haben wir den gesamten Code, der für die Durchführung der Projekte erforderlich ist, in den Haupttext des Buches aufgenommen. Grundsätzlich fügen wir kürzere Codeblöcke als Inline-Code ein, der in `dieser Schriftart` formatiert ist, und stellen größere Codeblöcke in separaten, nummerierten Code-Listings dar.



## Hinweis

Zum Zeitpunkt der Drucklegung sind wir zuversichtlich, dass der gesamte In-Text-Code funktioniert, aber wir können nicht garantieren (insbesondere für diejenigen unter Ihnen, die dies in gedruckter Form lesen), dass der Code auf lange Sicht fehlerfrei bleibt, da sich das Gebiet des Deep Learning und folglich auch seine Bibliotheken schnell weiterentwickeln. Auch wurde der In-Text-Code auf das für das Funktionieren der Projekte notwendige Minimum reduziert, sodass wir Ihnen dringend empfehlen, die Projekte in diesem Buch mit dem Code aus dem GitHub-Repository zu diesem Buch zu verfolgen: <http://mng.bz/JzKp>. Wir beabsichtigen, den Code auf GitHub für die nächste Zeit auf dem neuesten Stand zu halten, und er enthält auch zusätzliche Kommentare und Code, den wir zur Generierung vieler Abbildungen im Buch verwendet haben. Daher ist es am besten, wenn Sie das Buch zusammen mit dem entsprechenden Code in den Jupyter-Notebooks lesen, die auf dem GitHub-Repository zu finden sind.

Wir sind zuversichtlich, dass dieses Buch Ihnen die Konzepte des Deep Reinforcement Learning vermitteln wird und nicht nur, wie man Dinge in Python effektiv programmiert. Wenn Python irgendwie verschwinden würde, nachdem Sie dieses Buch gelesen haben, wären Sie immer noch in der Lage, die Algorithmen in einer anderen Sprache oder einem anderen Framework zu implementieren, da Sie die Grundlagen verstanden haben.

## liveBook-Diskussionsforum

Mit dem Kauf von *Einstieg in Deep Reinforcement Learning* erhalten Sie Zugang zu einem privaten Web-Forum von Manning Publications, in dem Sie Kommentare zum Buch abgeben, technische Fragen stellen und Hilfe von den Autoren und anderen Benutzern erhalten können. Um auf das Forum zuzugreifen, nutzen Sie den folgenden Link:

<https://livebook.manning.com/#!/book/deep-reinforcement-learning-in-action/discussion>. Mehr über die Manning-Foren und die Verhaltensregeln erfahren Sie unter <https://livebook.manning.com/#!/discussion>.

Mannings Engagement für unsere Leser besteht darin, einen Ort zu schaffen, an dem ein sinnvoller Dialog zwischen einzelnen Lesern sowie zwischen Lesern und Autoren stattfinden kann. Es handelt sich dabei nicht um eine Verpflichtung zu einem bestimmten Umfang der Teilnahme seitens der Autoren, deren Beitrag zum Forum freiwillig (und unbezahlt) bleibt. Wir schlagen deshalb vor, dass Sie versuchen, den Autoren einige herausfordernde Fragen zu stellen, damit ihr Interesse geweckt wird! Das Forum und die Archive früherer Diskussionen sind auf der Website des Verlags zugänglich, solange das Buch im Handel erhältlich ist.

# I Teil I: Grundlagen

[Teil I](#) besteht aus fünf Kapiteln, die die grundlegendsten Aspekte des Deep Reinforcement Learning vermitteln. Nachdem Sie [Teil I](#) gelesen haben, können Sie die Kapitel in [Teil II](#) in beliebiger Reihenfolge verstehen.

- [Kapitel 1](#) beginnt mit einer Einführung in das Deep Reinforcement Learning, in der die wichtigsten Konzepte und ihr Nutzen erläutert werden.
- In [Kapitel 2](#) beginnen wir mit dem Aufbau angewandter Projekte, die die grundlegenden Ideen des Reinforcement Learning veranschaulichen.
- In [Kapitel 3](#) implementieren wir ein Deep Q-Network – die gleiche Art von Algorithmus, den DeepMind verwendete, um Atari-Spiele auf übermenschlichen Leveln zu spielen.
- [Kapitel 4 und 5](#) runden die gebräuchlichsten Reinforcement-Learning-Algorithmen ab, nämlich Policy-Gradient-Methoden und Actor-Critic-Methoden. Wir werden die Vor- und Nachteile dieser Ansätze im Vergleich zu Deep Q-Networks betrachten.

# 1 Was ist Reinforcement Learning?



## Dieses Kapitel beinhaltet:

- einen kurzen Überblick über Machine Learning (maschinelles Lernen),
- die Einführung von Reinforcement Learning (bestärkendes Lernen – RL) als Teilbereich,
- das grundlegende Framework (Programmiergerüst) des Reinforcement Learning.

*Computersprachen der Zukunft werden sich mehr mit Zielen und weniger mit vom Programmierer vorgegebenen Prozeduren beschäftigen.*

Marvin Minsky, 1970 ACM-Turing-Vorlesung

Wenn Sie dieses Buch lesen, sind Sie wahrscheinlich damit vertraut, wie tiefe neuronale Netze für Dinge wie Bildklassifikation oder Vorhersagen verwendet werden (und