

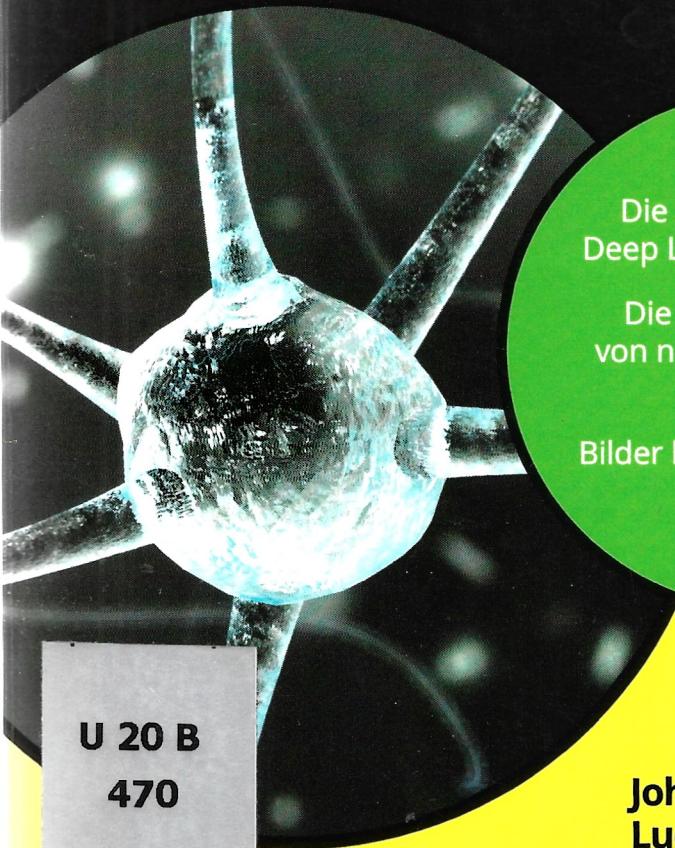
LERNEN EINFACH GEMACHT



Deep Learning

kompakt

für
dummies®



Die Grundlagen von
Deep Learning verstehen

Die Funktionsweise
von neuronalen Netzen
begreifen

Bilder klassifizieren, Texte
verarbeiten

U 20 B

470

John Paul Mueller
Luca Massaron

Definition von Deep Learning

Zur Vermeidung von Missverständnissen müssen wir zunächst wichtige Begriffe klar definieren. Wie schon erwähnt, ist Deep Learning eine Form des maschinellen Lernens und dies wiederum ein Teilgebiet der KI (Abbildung 1.1). In den folgenden Abschnitten arbeiten wir uns daher Stück für Stück bis zum Kern der Sache vor, damit Sie Deep Learning künftig korrekt von anderen Verfahren abgrenzen können.

Künstliche
Intelligenz

Maschinelles
Lernen

Deep
Learning

Abbildung 1.1: Deep Learning ist eine Unterkategorie des maschinellen Lernens und dies wiederum ein Teilgebiet der KI.

Künstliche Intelligenz als Oberbegriff

Was ist »künstliche Intelligenz«? Diese Frage könnte man damit beantworten, dass es sich um ein intelligentes System oder Wesen handelt, das künstlich erschaffen wurde. Aber was genau meinen wir eigentlich mit »intelligent«? Hier gehen die Definitionen weit auseinander. Wir können aber festhalten, dass Intelligenz bestimmte kognitive Funktionen und Fähigkeiten voraussetzt, nämlich:

- ✓ **Lernen:** Neue Informationen werden aufgenommen und verarbeitet.
- ✓ **Schlussfolgern:** Informationen werden auf verschiedene Weise ausgewertet.

- ✓ **Verstehen:** Die Bedeutung der ausgewerteten Informationen wird begriffen.
- ✓ **Überprüfen:** Die Gültigkeit der Informationen wird geprüft und mit konstistenten, belegbaren Quellen verglichen.
- ✓ **Vorausschauen:** Zusammenhänge zwischen geprüften Daten und anderen Daten werden vorhergesagt.
- ✓ **Beurteilen:** Spezifische Situationen werden gemäß den identifizierten Zusammenhängen analysiert.

Intelligenz beruht also auf bestimmten Denkprozessen, die ein Computersystem in einer Simulation recht gut nachahmen kann:

1. Es wird ein zu erreichendes Ziel festgelegt.
2. Alle bisher verfügbaren Daten zu diesem Ziel werden beurteilt.
3. Zusätzliche Daten, die beim Erreichen des Ziels helfen könnten, werden erfasst.
4. Die Daten werden so manipuliert und arrangiert, dass sie in ihrer Form mit den vorhandenen Daten übereinstimmen.
5. Die Beziehungen und Wahrheitswerte zwischen den vorhandenen und neuen Daten werden definiert.
6. Es wird geprüft, ob das Ziel mithilfe der aufbereiteten Daten erreicht wird.
7. Falls nicht, wird der Prozess an die neuen Daten und die sich ändernden Erfolgsschancen angepasst.
8. Die Schritte 2 bis 7 werden so oft wiederholt, bis entweder das Ziel erreicht ist (das Ergebnis ist wahr) oder alle Möglichkeiten erfolglos ausprobiert wurden (das Ergebnis ist falsch).



Ein Computer hat nur sehr eingeschränkte Möglichkeiten, sich zu einem wahrlich intelligenten System zu entwickeln. Da er die Daten durch maschinelle Prozesse automatisch und auf rein mathematische Weise manipuliert und auswertet, kann von echtem »Begreifen« und »Verstehen« wohl keine Rede sein. Bisher sind Computer nicht in der Lage, die kognitiven Fähigkeiten, auf denen Intelligenz basiert, vollständig zu implementieren.

Methoden zur Realisierung von
maschinellem Lernen unterscheiden

Maschinelles Lernen aus den richtigen
Gründen anwenden

Kapitel 2

Maschinelles Lernen

Wie in Kapitel 1 erläutert, lernen Menschen und Computer auf unterschiedliche Weise. Das maschinelle Lernen, also die Lernmethodik von Computern, lässt sich als Kombination aus Mathematik, Mustererkennung und Datenspeicherung beschreiben. In diesem Kapitel schauen wir uns nun genauer an, wie maschinelles Lernen eigentlich funktioniert und wie aus Daten Wissen abgeleitet wird.

Für die Datenanalyse gibt es dabei keine wirklich perfekten Methoden. Oft müssen Sie experimentieren, bis Sie die gewünschten Ergebnisse erhalten. Zudem hat jede Methode beim maschinellen Lernen ihre eigenen Vor- und Nachteile.

Zum Abschluss des Kapitels erfahren Sie, wie sich die Theorie in der Praxis umsetzen lässt und wie Sie geeignete von ungeeigneten Anwendungsszenarien unterscheiden. Manchmal wäre maschinelles Lernen nämlich der falsche Ansatz, und Sie hätten auch nach langem Herumprobieren keine verwertbaren Resultate. Nur wenn Sie wissen, wo und wie sich maschinelles Lernen sinnvoll einsetzen kann, können Sie konsistente Ergebnisse erzielen und spannende Aufgaben lösen.

Definition von maschinellem Lernen

Von maschinellem Lernen sprechen wir, wenn ein KI-System automatisch aus Erfahrungen lernt und sein Verhalten optimiert, ohne dass ihm dies vorher explizit einprogrammiert wurde. Der Lernprozess beruht dabei auf der Auswertung großer Datensmengen. Die Parameter des zugrunde liegenden Algorithmus werden bei der Berechnung so lange angepasst, bis ein zufriedenstellendes Ergebnis

gefunden wird. So einfach, wie dies klingt, ist es natürlich in der Praxis nicht. Die folgenden Abschnitte erklären wesentliche Konzepte des maschinellen Lernens, und Sie erfahren, wo genau in diesem Themenfeld Deep Learning einzurorden ist.



Maschinelles Lernen wird mithilfe von *Algorithmen* realisiert. Ein Algorithmus ist eine Reihe von abzuarbeitenden Operationen (zumeist Berechnungen), die ein vorgegebenes Problem in einer begrenzten Anzahl von Schritten lösen. Beim maschinellen Lernen umfassen diese Schritte die Analyse von Daten, die Ableitung von Wissen und darauf basierend die eigentliche Lösung der Aufgabenstellung.

Funktionsweise von maschinellem Lernen

Wie Sie schon wissen, »lernen« Computer nicht wie Menschen, sondern mithilfe von Mathematik und Daten. Präzise definieren lässt sich dieser Lernprozess nicht, weil der Computer je nach Funktionsweise des Algorithmus aus den Daten auf unterschiedliche Weise Wissen extrahiert. Fast immer sind aber riesige Datenmengen nötig, die dem Computer vermitteln, welche Ausgabe bei bestimmten Eingaben erwartet wird. Jedes Eingabe/Ausgabe-Paar stellt ein *Beispiel* dar. Je mehr Beispiele der Computer zur Verfügung hat, desto besser kann er lernen, indem er in den Daten nach Linien, Clustern oder anderen statistischen Mustern sucht, die den Problemraum definieren. Im Grunde optimiert der Computer mithilfe des Algorithmus ein Modell, das die Daten mathematisch zusammenfasst und es ermöglicht, auch für neue Eingabewerte die korrekte Ausgabe zu berechnen. Diese *Anpassung* des Modells wird als *Training* bezeichnet, weil der Computer mit den Daten so lange »trainiert«, bis er möglichst jeder erhaltenen Eingabe die korrekte Ausgabe zuordnen kann. Je genauer das Modell die Ausgaben vorhersagt, desto besser hat der Computer aus den bereits bekannten Daten gelernt.

Abbildung 2.1 zeigt vereinfacht, wie maschinelles Lernen funktioniert. Gegeben sind die Eingabewerte 1, 4, 5, 8 und 10 sowie die zugehörigen Ausgabewerte 7, 13, 15, 21 und 25. Der ML-Algorithmus erkennt, dass der Zusammenhang zwischen den Eingaben und Ausgaben am besten durch die Formel $2x + 5$ beschrieben wird. Diese Formel definiert das *Modell*, mit dessen Hilfe sich auch für neue Eingaben die zugehörigen Ausgaben berechnen lassen. Die Trendlinie zeigt das vom Algorithmus gefundene Muster. Für den neuen Eingabewert 3 wäre somit der Ausgabewert 11 zu erwarten. Diese Ausgabe muss nicht mehr explizit programmiert werden, sondern wird vom Computer selbstständig berechnet. In der Praxis sind die Szenarien natürlich viel komplizierter, und nicht immer findet der Computer eine Formel, die präzise jeder Eingabe eine Ausgabe zuordnet.

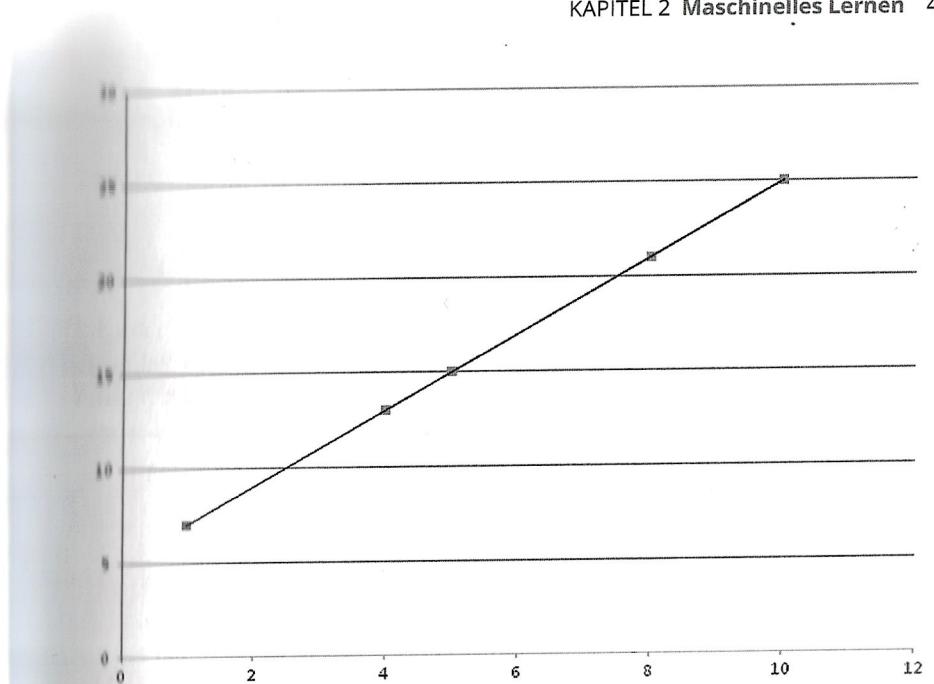


Abbildung 2.1: Visualisierung eines einfachen ML-Szenarios

Maschinelles Lernen basiert also auf der Idee, dass sich die Realität durch mathematische Funktionen abbilden lässt, die der Computer vorab nicht kennt, aber mithilfe von Beispieldaten herausfinden kann. Auch sehr komplexe Szenarien können durch mathematische Formeln ausgedrückt werden. Der Computer muss nur den Zusammenhang zwischen den beispielhaften Eingaben und Ausgaben erkennen. Wie schon in Kapitel 1 erklärt, versteht der Computer eigentlich gar nicht, was er da lernt. Für ihn ist alles reine Mathematik.

Lernen durch unterschiedliche Strategien

Beim maschinellen Lernen werden die Daten mithilfe unterschiedlicher Ansätze ausgewertet. Je nach Art der verfügbaren Eingaben und erwarteten Ausgaben lassen sich die Algorithmen bestimmten Lernstrategien zuordnen. Bei Ihren Projekten müssen Sie sich später überlegen, was für Daten Sie zur Verfügung haben und welche Ergebnisse Sie sich erhoffen. Wir unterscheiden vier Lernstrategien:

- ✓ Überwachtes Lernen
- ✓ Selbstüberwachtes Lernen

- ✓ Verstärkendes Lernen
- ✓ Unüberwachtes Lernen

Die folgenden Abschnitte stellen diese vier Strategien im Detail vor.

Überwachtes Lernen

Die Eingabedaten für die Algorithmen beim überwachten Lernen werden durch Menschen klar beschriftet und erwarten ein spezifisches Ergebnis. Im Rahmen des Trainings wird ein Modell erstellt, das der Algorithmus dann an die Daten anpasst. Je länger das Training fortgeführt wird, desto besser werden die Vorhersagen. Zu den überwacht lernenden Algorithmen zählen zum Beispiel:

- ✓ Lineare oder logistische Regression
- ✓ Support Vector Machines (SVM)
- ✓ Naiver Bayes-Klassifikator
- ✓ K-Nearest Neighbors (KNN)

Wir unterscheiden zwischen Regressionsproblemen, bei denen das Ziel ein numerischer Wert ist, und Klassifikationsproblemen, bei denen das Ziel eine qualitative Variable ist (ein Klassenname oder ein Schlagwort). So wäre die Bestimmung der durchschnittlichen Immobilienpreise im Großraum Boston ein Regressionsproblem und die Unterscheidung zwischen Schwertlilienarten anhand ihrer Kelch- und Blütenblätter ein Klassifikationsproblem. Die folgende Tabelle enthält weitere Beispiele für das überwachte Lernen:

| Dateneingabe (x) | Datenausgabe (y) | Alltagsbeispiel |
|-----------------------------------|---|--|
| Bisherige Käufe eines Kunden | Liste mit Produkten, die vom Kunden noch nie gekauft wurden | Empfehlungssystem |
| Bilder | Beschriftete Kästchen mit Objekten im Bild | Bilderkennung und -analyse |
| Deutscher Text in Form von Fragen | Deutscher Text in Form von Antworten | Chatbot (Softwareanwendung, die mit Kunden kommuniziert) |
| Englischer Text | Deutscher Text | Maschinelle Übersetzung |
| Audioinhalte | Niederschriften | Spracherkennung |
| Bilder, Sensordaten | Befehle zum Lenken, Bremsen oder Beschleunigen | Verhaltensplanung für autonome Fahrzeuge |

Selbstüberwachtes Lernen

Das selbstüberwachte Lernen beschreiben manche Experten auch als autonomes überwachtes Lernen – man profitiert von den Vorteilen des überwachten Lernens, erspart sich aber den Aufwand für das Beschriften aller Trainingsdaten. Der Algorithmus findet stattdessen Korrelationen, eingebettete Metadaten oder relevante Informationen in den Eingabedaten und kann sich daraus die Beschriftung der Ausgabedaten selbst herleiten.

Verstärkendes Lernen

Verstärkendes Lernen ist im Prinzip eine erweiterte Form des selbstüberwachten Lernens, denn beide Strategien versuchen, anhand unbeschrifteter Trainingsdaten ähnliche Ziele zu erreichen. Beim verstärkenden Lernen gibt es aber zusätzlich eine Feedbackschleife. Wenn der Algorithmus eine Aufgabe korrekt löst, erhält er positives Feedback für sein Modell. Er wird sich bei seinen Berechnungen sicherer. Bei Fehlern hingegen erhält er negatives Feedback und seine Berechnungen werden unsicherer. Sie können sich dies in etwa so vorstellen, als ob Sie einem Hund mithilfe von Belohnungen und Bestrafungen das richtige Verhalten beibringen.

Unüberwachtes Lernen

Beim unüberwachten Lernen sind die Eingabedaten für die Algorithmen nicht beschriftet und die zugehörigen Ausgaben nicht bekannt. Der Algorithmus muss selbstständig Strukturen in den Daten erkennen und sich ein geeignetes Modell erarbeiten. Bei dieser strukturellen Analyse könnte das Ziel darin bestehen, Redundanzen zu eliminieren oder ähnliche Daten zu gruppieren. Beispiele für unüberwachtes Lernen sind:

- ✓ Cluster-Bildung
- ✓ Erkennung von Anomalien

Training, Validierung und Test

Maschinelles Lernen ist ein fortlaufender Prozess aus drei Arbeitsabschnitten, die Sie so oft wiederholen müssen, bis Ihr System wie gewünscht arbeitet:

- ✓ **Training:** Beim Training wird das Modell mithilfe des jeweiligen Algorithmus an spezifische Trainingsdaten angepasst, die den später zu erwartenden neuen Daten gut entsprechen sollten. Ist dies nicht der Fall, wird das entwickelte

- Elastic Net
- LARS (Least-Angle Regression)

✓ **Support Vector Machine (SVM):** Diese überwacht lernenden Algorithmen stellen die Trainingsdaten als Vektoren dar und lösen Klassifikations- und Regressionsprobleme, indem sie die Daten mithilfe einer sogenannten *Kernelfunktion* in Klassen trennen. Die Funktion berechnet eine Trennebene und versucht, den Abstand zu den am nächsten liegenden Daten (die sogenannten Stützvektoren, engl. *support vectors*) möglichst zu maximieren. Die Zuordnung eines neuen Beispiels zu einer Klasse fällt sodann leichter. Beispiele für SVM-Algorithmen sind:

- Lineare SVM
- RBF-SVM
- SVM mit nur einer Klasse (für unüberwachtes Lernen)

✓ **Deep Learning:** Und damit sind wir bei der Kategorie, um die es in diesem Buch hauptsächlich geht. DL-Algorithmen sind eine Weiterentwicklung künstlicher neuronaler Netze und erlauben die Nutzung von viel mehr Schichten und somit die Entwicklung viel komplexerer neuronaler Netze zur Auswertung noch größerer Datensätze. Derartige Algorithmen funktionieren gut beim semiüberwachten Lernen, wenn nur ein Minimum an beschrifteten Trainingsdaten verfügbar ist. Vertreter dieser Kategorie sind:

- Deep Boltzmann Machine (DBM)
- Deep Belief Network (DBN)
- Konvolutionsnetz (CNN)
- Rekurrentes neuronales Netz (RNN)
- Geschichtete Autoencoder

Neben den aufgelisteten wichtigsten Algorithmengruppen stehen noch zahlreiche andere Optionen zur Wahl. Nicht aufgeführt sind zum Beispiel Algorithmen, die zur Merkmalsselektion, zur Verbesserung der Genauigkeit, zur Leistungsbeurteilung oder für andere spezialisierte ML-Zwecke genutzt werden. So gibt es unter anderem eine ganze Reihe von Algorithmen, die sich dem maschinellen Sehen (*Computer Vision, CV*) oder der Verarbeitung natürlicher Sprache (*Natural Language Processing, NLP*) widmen.

Ein ziemliches Wirrwarr an Algorithmen, nicht wahr? Aber keine Sorge. Je mehr Sie sich mit dem Thema befassen, desto leichter fällt es, den Überblick zu behalten.

Sinnvolle Einsatzbereiche für maschinelles Lernen

Wie bereits erwähnt, stehen bei der KI verschiedene Methoden zur Wahl. Das heißt, maschinelles Lernen (und damit auch Deep Learning) ist nicht für jedes Problem die beste Lösung. Um zu verstehen, für welche Arten von Problemen sich maschinelles Lernen gut eignet, sollten Sie sich zunächst in Erinnerung rufen, wie die verschiedenen Algorithmen lernen. Dann können Sie entscheiden, ob sich Ihr spezifisches Problem damit lösen lässt. Da beim maschinellen Lernen Verallgemeinerungen anhand spezifischer Beispiele gefunden werden sollen, ist es für die folgenden Szenarien eher ungeeignet:

- ✓ Als Antwort werden präzise Ergebnisse erwartet, beispielsweise Berechnungen für eine Reise zum Mars.
- ✓ Das Problem ließe sich zwar durch Verallgemeinerung lösen, aber andere Methoden wären einfacher (zum Beispiel das Entwickeln einer Software zur Berechnung der Fakultät einer Zahl).
- ✓ Das Problem lässt sich nicht gut verallgemeinern, da es missverstanden wird oder zu komplex ist oder keine spezifischen Zusammenhänge zwischen Eingabe- und Ausgabedaten bestehen.

Die folgenden Abschnitte stellen einige sehr sinnvolle Einsatzbereiche für maschinelles Lernen vor.

Anwendungsszenarien für maschinelles Lernen

Ob sich maschinelles Lernen in einem bestimmten Anwendungsszenario vorteilhaft nutzen lässt, hängt teils von der Problemstellung und teils von Ihren Erwartungen ab. Wenn Sie beispielsweise häufig bei einem großen Onlinehändler einkaufen, erwarten Sie sicher, dass der ML-Algorithmus Ihnen nach einer gewissen Zeit gute Produktempfehlungen basierend auf Ihren bisherigen Käufen gibt. Dabei möchten Sie natürlich Produkte empfohlen bekommen, die Sie wirklich brauchen und bisher übersehen haben. Wenn der Algorithmus Ihnen

nutzen. Außerdem schauen wir uns die vorgefertigte Implementierung [keras-rl](https://github.com/keras-rl/keras-rl) an (<https://github.com/keras-rl/keras-rl>), mit der zum Beispiel ein moderner RL-Algorithmus wie *Deep Q-Network* (DQN) von Google klassische Computerspiele für Atari 2600 erlernen und auf menschlichem Niveau spielen kann.



Wie immer müssen Sie den Beispielcode nicht manuell abtippen. Die Quelltexte befinden sich in der Datei DL4D_17_Reinforcement_Learning.ipynb.

Strategische Spiele mit neuronalen Netzen

Als Kleinkind haben Sie bestimmt gern Ihre Umwelt erkundet und die Grenzen Ihrer kindlichen Fähigkeiten unter den besorgten (oder genervten) Augen Ihrer Eltern ausgetestet. Erst später haben Sie begonnen, sich neues Wissen anhand der Erfahrungen Ihrer Mitmenschen anzueignen, statt alles selbst auszuprobieren. Während ein überwachtes Lernverfahren dem Schüler dank Lehrern und Büchern neues Wissen erlangt, ist ein RL-Algorithmus mit einem neugierigen Kleinkind zu vergleichen – der Algorithmus beginnt bei null und sammelt neues Wissen durch Ausprobieren verschiedener Aktionen, die entweder eine Belohnung oder eine Bestrafung zur Folge haben.

RL bietet somit eine kompakte Methodik, die das Lernen auch ohne vorbereitete Sammeln riesiger Datenmassen ermöglicht, dabei aber trotzdem komplexe Interaktionen mit der externen Welt zulässt. Da RL ohne vorbereitete Daten beginnt, muss der Algorithmus zwangsläufig mit der Umwelt interagieren und Feedback verschaffen, um an seine nötigen Daten zu gelangen. Mit diesem Ansatz könnten Sie beispielsweise einen Roboter in der physischen Welt oder einen Chatbot in einer digitalen Welt lernen lassen. RL-Verfahren eignen sich vor allem für Problemstellungen, die sich allein mit statischen (vorbereiteten) Daten nicht einfach lösen lassen. Dazu zählt das Erlernen von Spielen oder das Finden der bestmöglichen Strategie in ungewissen Szenarien, zum Beispiel bei der Optimierung von Onlinewerbekampagnen. Gerade die Werbebranche ist ein pischer Anwendungsbereich für derartige Lernverfahren, denn es muss eine richtige Kampagne für die jeweils richtige Zielgruppe gefunden werden, ohne vorherige Erfahrungswerte (weil die Werbekampagnen ja in der Regel neu sind).

Grundprinzip des verstärkenden Lernens

Beim verstärkenden Lernen gibt es einen *Agenten* (ein Roboter in der physischen Welt oder ein Bot in der digitalen Welt), der mit einer physikalischen oder

Umgebung interagiert. Dieser Agent erhält Informationen von der Umgebung (den sogenannten *Zustand*) und reagiert darauf (wodurch er manchmal auch die Umgebung verändert). Je nachdem, was der Agent tut oder nicht tut, bekommt er positive oder negative Rückmeldungen von der Umgebung. Dieses Feedback wird als *Belohnung* bezeichnet, auch wenn es negativ ausfällt. Ziel des Agenten ist es, selbstständig herauszufinden, wie sich die Gesamtsumme aller Belohnungen während der Interaktion mit der Umgebung maximieren lässt.

Abbildung 17.1 zeigt ein Schema dieses Zwischenspiels des Agenten und der Umgebung. Beachten Sie die tiefgestellte Variable t für die Zeiterfassung. Wenn der aktuelle Moment t ist, lässt sich der Moment direkt davor mit $t-1$ beschreiben. Zu diesem Zeitpunkt $t-1$ führte der Agent eine Handlung aus, die ihm neue Zustandsinformationen und eine Belohnung von der Umgebung beschert hat. Anhand dieser relativen Werte – Handlung zum Zeitpunkt t , Zustand zum Zeitpunkt $t-1$ und Belohnung zum Zeitpunkt t – lernt der Algorithmus, mit welchen Handlungen sich ein bestimmter Zustand der Umgebung herbeiführen lässt.

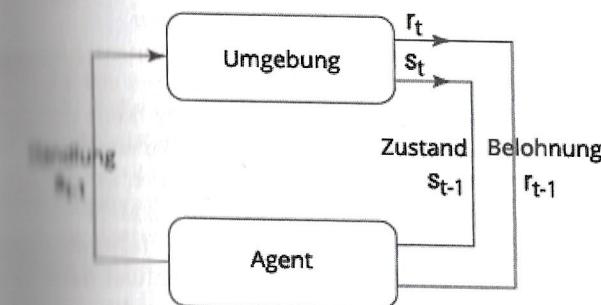


Abbildung 17.1: Ein Agent interagiert mit seiner Umgebung beim verstärkenden Lernen.

Ivan Goodfellow, der auch das GAN-Konzept entwickelte, hat die bessere Integration von RL und Deep Learning in den kommenden Jahren oberste Priorität. Dadurch lassen sich nämlich auch smartere Roboter entwickeln (<https://www.forbes.com/sites/quora/2017/07/21/whats-next-for-deep-learning/#36131b871002>). Die Idee, RL in andere Verfahren zu integrieren, ist nicht neu, aber bisher konzentrierte man sich mehr auf die Kombination mit der Statistik und Algorithmen statt mit neuronalen Netzen. Doch einige Vordenker, die sich schon eher an dieses Unterfangen gewagt haben, in den frühen 1990ern beispielsweise entwarf Gerald Tesauro am IBM Research Center ein Verfahren, um einem Computer das Spiel Backgammon beizubringen (mehr zum Spiel unter <https://www.brettspielnetz.de/spielregeln/backgammon.php> und einen (menschlichen) Champion zu besiegen. Dazu nutzte

In der Tiefe sieht man sie blitzen, die Intelligenz

Sie haben von Deep Learning gehört und wollen sich nun darüber informieren und vielleicht erste kleine Schritte machen, diese Technologie zu nutzen? Dann hilft Ihnen dieses Buch. Die Autoren erklären Ihnen, welchen Einfluss Deep Learning im Alltag hat und in welcher Beziehung es zu maschinellem Lernen steht. Anschließend sammeln Sie erste eigene Erfahrungen mit vorgegebenen Python-Programmen und verstehen so die Funktionsweise von neuronalen Netzen und wie Bilder klassifiziert und Sprache sowie Text verarbeitet werden. So ist dieses Buch ein schneller erster und verständlicher Einstieg in Deep Learning.

Sie erfahren

- Welche Frameworks Sie brauchen
- Was die mathematischen Grundlagen von Deep Learning sind
- Wie Sie grundlegende Vorteile von Deep Learning ausnutzen
- Wie Sie komplexere Aufgaben mit CNNs lösen



John Paul Mueller ist freier Autor und technischer Redakteur. Bis heute hat er 100 Bücher und mehr als 600 Zeitschriftenartikel geschrieben.

Luca Massaron ist Data Scientist und spezialisiert auf die Aufbereitung und Analyse von Big Data.

Mach dich schlau:
www.fuer-dummies.de

WILEY



für
dummies®

ISBN 978-3-527-71687-6



9 783527 716876

Coverfotos: © uotas - stock.adobe.com