

Künstliche Intelligenz verstehen

Eine spielerische Einführung

» Hier geht's
direkt
zum Buch

DAS VORWORT

Kapitel 1

Einleitung

Es fällt nicht schwer, Namen von historischen Persönlichkeiten zu nennen, die besonders intelligent waren. Wir denken spontan an Köpfe wie Albert Einstein, Hannah Arendt, Leonardo da Vinci oder Simone de Beauvoir. Vielleicht haben wir auch einen Freund – nennen wir ihn Karsten –, der herausragend Schach spielt, oder eine Freundin – nennen wir sie Sabine –, die im Studium immer alles als Erste kapiert hat und im Alltag regelmäßig schlaue Lösungen für vertrackte Probleme aus dem Hut zaubert.

Weitaus schwieriger, als intelligente Menschen zu identifizieren, ist es, zu erklären, was Intelligenz eigentlich ist. Eine allgemein anerkannte Definition fehlt. Somit ist es wenig überraschend, dass auch der Begriff *Künstliche Intelligenz* (KI) höchst umstritten ist.

Den herausragenden Schachspieler Karsten werden wir ohne Umschweife intelligent nennen. Aber viele von uns würden zögern, ein Schachprogramm als intelligent zu bezeichnen, selbst wenn dieses weitaus besser spielen kann als unser Freund. Woher kommt diese Zurückhaltung?

Das dürfte unter anderem daran liegen, dass sich viele Menschen generell schwer damit tun, Maschinen menschliche Eigenschaften zuzuschreiben. Ein weiterer und aus unserer Sicht der entscheidende Grund: Alle Anwendungen, die aktuell unter dem Namen »künstliche Intelligenz« firmieren, sind jeweils auf *eine einzige Klasse von Aufgaben* spezialisiert. Versuchen Sie einmal, mit einem Schachprogramm über Literatur zu diskutieren, probieren Sie, eine Bilderkennungssoftware zu einer Partie Halma zu überreden, oder bitten Sie Ihr Navigationssystem, eine Knobelaufgabe zu lösen oder eine Geschichte zu einem vorgegebenen Thema zu improvisieren.



Diese Spezialisierung markiert einen wesentlichen Unterschied zwischen menschlicher Intelligenz und dem, was Computerprogramme aktuell zu leisten imstande sind: Unsere Freundin Sabine vermag sich auf sehr unterschiedliche Probleme und Aufgaben im kürzester Zeit einzustellen. Auch von Freund Karsten dürfen wir erwarten, dass er seine im Schachspiel erworbenen Fähigkeiten zum strategischen und weit vorausschauenden Denken auch auf andere Brettspiele oder gar auf Alltagssituationen übertragen kann.

Zwar können Sprachmodelle wie ChatGPT im Medium der natürlichen Sprache sehr unterschiedliche Aufgaben mehr oder weniger erfolgreich bewältigen. Sie sind aber von einer universell anwendbaren Intelligenz, die wir bei Sabine und Karsten beobachten können, noch weit entfernt. Es ist unter Fachleuten umstritten, ob eine solche *Allgemeine Künstliche Intelligenz* (AKI) jemals erreicht werden kann – und wenn ja, auf welchem Wege wir dorthin gelangen könnten. Wir halten uns in dieser Frage mit einer Einschätzung zurück, sind uns aber in einem Punkt sicher: Es genügt nicht, einen Haufen von spezialisierten Programmen zu versammeln und dann mal das eine, mal das andere aufzurufen, je nachdem, was gerade anliegt.

Fassen wir also zusammen: Die Kritik am Begriff *Künstliche Intelligenz* halten wir für berechtigt. Trotzdem lässt sich kaum bestreiten, dass unter diesem Begriff viele erstaunliche Anwendungen versammelt sind, die unser aller Leben bereits maßgeblich bestimmen und verändern – vielleicht sogar mehr, als uns lieb ist. Ziel dieses Buchs ist es, allen interessierten Leserinnen und Lesern ein grundlegendes Verständnis von KI zu vermitteln, ganz unabhängig davon, welche Vorkenntnisse diese mitbringen.

1.1 Worum es uns in diesem Buch geht

Stellen Sie sich vor, Sie besuchten eine Stadt, in der Sie nie zuvor gewesen sind. Sie planen einen längeren Aufenthalt oder wollen diese Stadt in Zukunft öfter besuchen. Es lohnt sich also, Orientierung zu gewinnen! Wie gehen Sie vor? Werden Sie sämtliche Straßen systematisch von Norden nach Süden und von Osten nach Westen durchkämmen? Werden Sie ausgehend von den Hauptverkehrsadern alle Nebenstraßen durchstreifen oder gar den Stadtplan auswendig lernen? Wohl eher nicht! Wenn Sie eine Stadt erkunden, werden Sie sich vermutlich zunächst einige Orientierungspunkte einprägen: den Bahnhof, das gemütliche Café, das Kunstmuseum, das Restaurant mit dem leckeren chinesischen Essen ... Ausgehend von diesen Punkten werden Sie im Laufe der Zeit die Stadt immer besser kennenlernen.

Dieses Buch nimmt sich das Thema KI in ähnlicher Weise vor. Wir erheben nicht den Anspruch, eine systematische oder gar vollständige Darstellung zu liefern. Vielmehr haben

wir einige Verfahren herausgepickt, die wir besonders lehrreich, interessant oder unterhaltsam finden.

Beim Schreiben dieses Buchs haben wir großen Wert auf anschauliche Beispiele und eine kurzweilige Darstellung gelegt. Um es mit den Worten einer befreundeten Informatikstudentin zu sagen: »An der Uni sind wir in das Thema Q-Learning mit einem Haufen mathematischer Definitionen eingestiegen – euer Buch startet mit einem Eichhörnchen, das seinen Nussvorrat verbummelt hat.«

Sehr viel Mühe haben wir darauf verwendet, die vorgestellten Verfahren praktisch erfahrbar zu machen: Auf der Webseite zum Buch finden Sie zahlreiche Beispielprogramme, die Sie mit nur einem Mausklick im Browser starten können. Die Programme laden ein zum Probieren und Experimentieren und stellen eine direkte Verbindung zwischen Theorie und Praxis her.

1.2 Für wen wir dieses Buch geschrieben haben

Wir sprechen mit diesem Buch einen großen Kreis von Leserinnen und Lesern an. Wenn Sie sich in einer der folgenden Beschreibungen wiederfinden, dann haben wir dieses Buch auch für Sie geschrieben:

- ▶ Sie haben noch nie programmiert und möchten es vielleicht auch nicht lernen, wollen aber mehr über KI erfahren.
- ▶ Sie sind Hobbyprogrammiererin oder Freizeit-Coder und freuen sich über Anregungen für eigene Projekte.
- ▶ Sie stehen am Beginn einer Ausbildung oder eines Studiums in einem Fachgebiet, in dem Sie mit Code und KI in Berührung kommen werden, und möchten sich einen ersten Überblick über KI-Verfahren verschaffen.
- ▶ Sie haben beruflich mit Programmierung zu tun, wissen aber noch wenig über KI und wollen ihr Wissen ausbauen.
- ▶ Sie lehren an einer Schule oder einer Hochschule und sind auf der Suche nach anschaulichen Beispielen für Ihren eigenen Unterricht.

Bei allen Verfahren, die wir vorstellen, war unser erstes Anliegen, ein grundlegendes Verständnis der entscheidenden Ideen zu vermitteln. Wir haben mehr Wert auf ein intuitives Verständnis als auf mathematische Präzision gelegt. Denn wir haben uns gedacht: Ausführliche Darstellungen von technischen Details und Spezialfällen nützen wenig, wenn das Wesentliche unklar bleibt.

Das bedeutet aber nicht, dass Sie irgendetwas verlernen müssen, wenn Sie mehr erfahren und über das Darstellungsniveau dieses Buchs hinausgehen wollen. Wann immer wir einen Sachverhalt grob vereinfacht haben, weisen wir deutlich darauf hin.

1.3 Aufbau der einzelnen Kapitel

Die einzelnen Kapitel sind weitgehend einheitlich aus folgenden Bausteinen zusammengesetzt:

- ▶ Wir stellen eine Klasse von Problemen vor, die wir lösen wollen.
- ▶ Wir erläutern ein Verfahren, das zur Lösung des vorgestellten Problems geeignet ist.
- ▶ Wir präsentieren ein Beispielprogramm, in dem das vorgestellte Verfahren zur Anwendung kommt. Das Programm beschreiben wir zunächst aus der Sicht von Nutzer:innen und Nutzern.
- ▶ Wir stellen den Programmcode vor, der das jeweilige Verfahren umsetzt.
- ▶ Wir geben einen Ausblick auf benachbarte Themen, skizzieren Punkte, die wir weglassen haben, schlagen Modifikationen des Beispielprogramms sowie Codeexperimente vor und geben Denkanstöße.

1.4 Ein Wort an die Programmierunkundigen

Wir richten uns mit diesem Buch ganz bewusst auch an Personen, die noch nie in ihrem Leben programmiert haben. Trotzdem enthält es einige Passagen, in denen wir Code im Detail diskutieren. In diesen Passagen mussten wir die Kenntnis von Grundbegriffen wie *Schleife*, *Funktion*, *Objekt* oder *Klasse* voraussetzen. Falls Ihnen solche Begriffe noch fremd sind, schlagen wir Ihnen eine dieser drei Vorgehensweisen vor:

- ▶ Sie überspringen unsere Codeerläuterungen komplett – das wäre zwar schade, aber zu verschmerzen: Die Passagen, die sich auf Code beziehen, sind für das angestrebte allgemeine Verständnis und die weitere Lektüre nicht zwingend notwendig.
- ▶ Sie überfliegen die Codeabschnitte und beachten dabei vor allem die Tabellen und die Infografiken. Auf diese Weise bekommen Sie immerhin eine Ahnung davon, aus welchen Komponenten der Code besteht und wie diese ineinandergreifen.

- ▶ Wenn Sie es genauer wissen wollen, finden Sie in Anhang A eine knappe Einführung in die verwendete Programmiersprache JavaScript und die p5.js-Bibliothek. Diese Einführung fängt bei null an, setzt also überhaupt keine Vorkenntnisse voraus. Das Studium dieser Einführung sollte Sie in die Lage versetzen, unseren Codeerläuterungen weitgehend zu folgen.

1.5 Beispielprogramme und die Webseite zum Buch

Alle Beispielprogramme können Sie ohne Download, Installation und Anmeldung direkt im Browser starten. Die Links finden Sie auf der Webseite zum Buch:

<https://maschinennah.de/ki-buch>.

The screenshot displays a grid of 10 interactive example programs, each with a unique illustration and a set of controls:

- Kapitel 2**: Texte bauen mit Markow. Illustration: A book character with a face and antennae. Example program: Nonsense-Texter. Controls: STARTEN, im Editor öffnen.
- Kapitel 3**: Schreibfehler automatisch korrigieren. Illustration: A grid of numbers with arrows indicating corrections. Example program: Wortvergleich. Controls: STARTEN, im Editor öffnen.
- Kapitel 4**: Wörter gruppieren. Illustration: A diagram showing items and transactions. Example program: Assoziationsanalyse. Controls: STARTEN, im Editor öffnen, > mehr...
- Kapitel 5**: Spiele für eine Person lösen. Illustration: A Tetris-like game board. Example program: Fruchtkräsch. Controls: STARTEN, im Editor öffnen, > mehr...
- Kapitel 6**: Spiele für zwei Personen gewinnen. Illustration: A Go board with black and white stones.
- Kapitel 7**: Q-Learning. Illustration: A book character holding a sword.
- Kapitel 8**: K-nächste-Nachbarn. Illustration: A diagram showing connections between nodes. Example program: Tiere erkennen. Controls: STARTEN, im Editor öffnen, > mehr...
- Kapitel 9**: K-means-Clustering. Illustration: A character holding a scale and a funnel.

Abbildung 1.1 Die Webseite zum Buch

Profis haben darüber hinaus die Möglichkeit, den Quellcode direkt bei GitHub herunterzuladen: <https://github.com/MaschinenNah/ki-buch>.

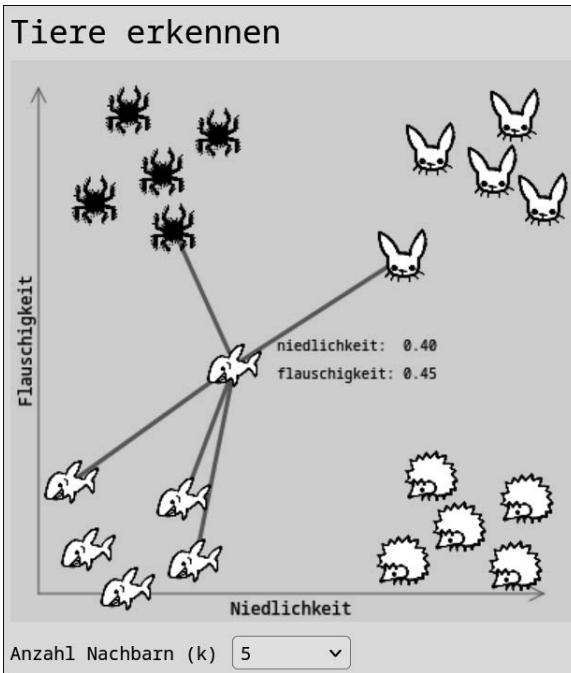


Abbildung 1.2 Das Beispielprogramm »Tiere erkennen«

Experimentierfreudige Leserinnen und Leser können den Code unserer Beispielprogramme eigenhändig modifizieren und das Ergebnis direkt im Browserfenster sehen. Der Online-Editor von p5.js macht es möglich. Abbildung 1.3 zeigt einen Screenshot des Editors: Links sehen Sie den Code, rechts die Ausführung des Programms. Mehr über Nutzung und Funktionen des Online-Editors erfahren Sie in Anhang A in Abschnitt A.3, »Der p5.js-Online-Editor«.

Im Buch werden wir ausschließlich diejenigen Codeabschnitte diskutieren, die für die Umsetzung der vorgestellten KI-Verfahren relevant sind. So stellen wir zum Beispiel in Kapitel 6, »Spiele für zwei Personen gewinnen«, nicht dar, wie wir das Spiel Reversi programmiert haben. Die Funktionsweise der grafischen Benutzeroberfläche, die Repräsentation des Spielfelds oder die Ermittlung der jeweils gültigen Züge besprechen wir nicht. Es geht uns allein darum, die Funktionsweise der KI zu erläutern, die gegen uns Reversi spielt.

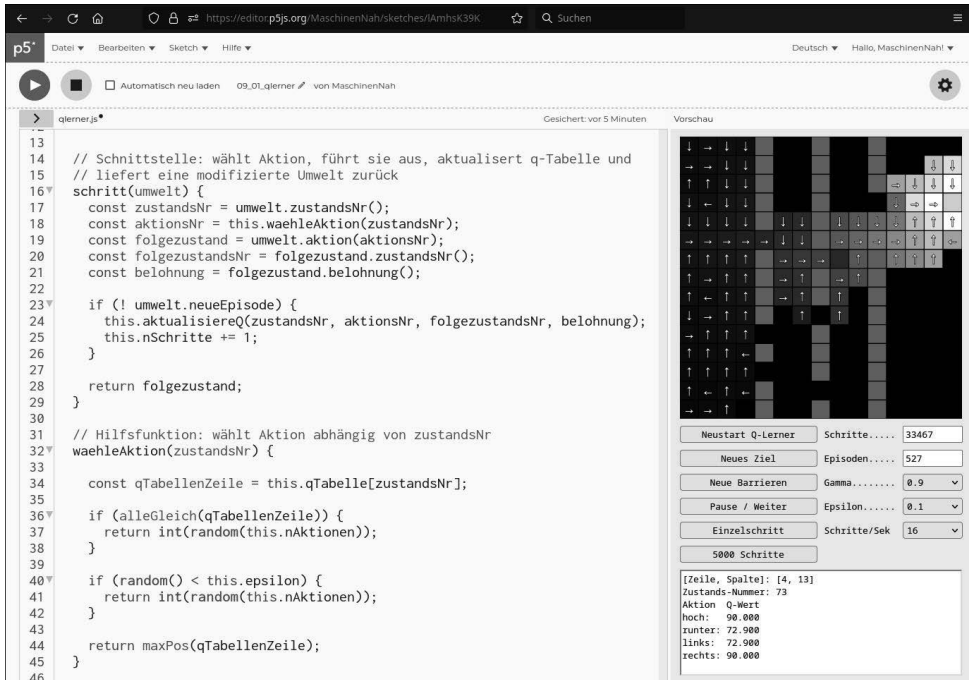


Abbildung 1.3 Im Online-Editor modifizieren Sie auf der linken Seite den Code, auf der rechten Seite sehen Sie das Ergebnis; dafür ist keine Anmeldung notwendig.

Wenn Sie allerdings mehr über das Zusammenspiel der einzelnen Komponenten unserer Beispielprogramme erfahren wollen, stehen Sie nicht mit leeren Händen da: Den online verfügbaren Quellcode haben wir mit erläuternden Kommentaren versehen. Da wir uns mit diesem Buch auch an Anfängerinnen und Einsteiger richten, sind die Kommentare mitunter etwas ausführlicher geraten, als es professionelle Softwareentwicklerinnen und -entwickler für angebracht halten dürften.

1.6 Warum wir JavaScript und p5.js verwendet haben

Programmierprofis werden sich fragen, warum wir für unsere KI-Einführung ausgerechnet JavaScript und p5.js nutzen. Schließlich werden professionelle KI-Anwendungen meist in Python geschrieben. Und in der Tat: Wenn Sie nach der Lektüre dieses Buchs auf den Geschmack gekommen sind und größere KI-Projekte in Angriff nehmen wollen, werden Sie an Python und Python-Umgebungen wie Keras oder PyTorch nicht vorbeikommen.

Das liegt zum einen daran, dass wir waschechte p5.js-Fans sind und diese Sprache seit vielen Jahren erfolgreich in Kursen für jugendliche und erwachsene Programmierneulinge verwenden.

Und auch sonst war unsere Entscheidung didaktisch motiviert: Wir wollen möglichst viele Leserinnen und Leser ansprechen und den Einstieg so unkompliziert wie möglich gestalten. Mit JavaScript, p5.js und dem Online-Editor ist jedes Beispielprogramm nur einen Mausklick entfernt. Die Notwendigkeit eines Downloads, einer Softwareinstallation oder auch nur einer Registrierung würde viele potenziell interessierte Menschen abschrecken.

Zudem programmieren wir in unseren Beispielen vieles »von Hand«, was Keras, PyTorch & Co. als fertige Funktion anbieten. Auch hier gilt: Wenn Sie später auf eine andere Programmiersprache umsteigen, müssen Sie nichts davon verlernen, was Sie in diesem Buch gelernt haben. Im Gegenteil: Die Fachbegriffe, die wir hier vorstellen, bleiben dieselben, ganz unabhängig davon, welche Programmiersprache Sie nutzen. Zudem bieten unsere handprogrammierten Beispiele die Möglichkeit, nachzuvollziehen, was bei Keras und PyTorch hinter den Kulissen passiert.

1.7 Begriffliche Abgrenzung und Fachbegriffe

Wir haben bei der Auswahl der Themen den Begriff *KI* bewusst sehr weit gefasst. Viele der vorgestellten Verfahren werden im strengen Sinne nicht – oder nicht mehr – zum Bereich der *KI* gezählt. So setzen einige Kapitel einen Schwerpunkt auf *Data-Mining*, also auf die Extraktion von Zusammenhängen aus Datenmengen, deren Umfang das menschliche Fassungsvermögen überschreitet. Die Grenzen zwischen *Data-Mining* und *KI* sind aber fließend. Kapitel 9, »K-means-Clustering«, zeigt mit dem k-means-Clustering-Algorithmus ein klassisches *Data-Mining*-Verfahren. Kapitel 12, »Neuronale Netze III: Fehler zurückverfolgen mit dem Neuronentrainer«, stellt die Verbindung zwischen k-means-Clustering und neuronalen Netzen her: Letztlich geht es in beiden Fällen um die Ziehung von Grenzlinien zwischen Datenpunktmengen.

Beim Schreiben dieses Buchs haben wir uns sehr viele Gedanken darüber gemacht, welche Fachbegriffe wir vorstellen und welche wir unerwähnt lassen. Einerseits helfen Fachbegriffe, Ordnung in ein Themengebiet zu bringen und Unterschiede zu markieren. Andererseits kann eine zu große Genauigkeit beim Einstieg eher abschrecken und den Blick auf das Wesentliche verstellen.

1.8 Inhalte, Themen, Kapitel

Im Folgenden skizzieren wir, was Sie in den einzelnen Kapiteln erwartet. Viele der Kapitel können Sie unabhängig voneinander lesen. Wo diese inhaltlich aufeinander aufbauen, haben wir das markiert.

In **Kapitel 2 bis Kapitel 4** stellen wir drei Verfahren vor, die sich besonders gut auf Texte anwenden lassen. Sie stehen in einem losen Zusammenhang, können aber auch unabhängig voneinander gelesen werden:

- ▶ Mittels sogenannter *Markow-Prozesse* lassen sich Texte im Hinblick auf die Häufigkeit bestimmter Buchstaben- und Wortfolgen untersuchen. Anschließend können wir diese Eigenschaften im Sinne einer Parodie reproduzieren, um abhängig vom Ausgangsmaterial etwa deutsch-, englisch- oder französischsprachigen Unsinn zu produzieren. Wir zeigen auch, wie Sie den Markow-Prozess zur Programmierung einer einfachen Textvervollständigung nutzen können, wie wir sie von unseren Smartphones kennen.
- ▶ Der *Levenshtein-Algorithmus* berechnet die Ähnlichkeit zweier Wörter, indem er die kürzeste Schrittfolge ermittelt, die nötig ist, um ein Wort in ein anderes zu verwandeln. Wir nutzen dieses Verfahren zur Programmierung einer einfachen Rechtschreibprüfung, die etwa das Wort »Schampingjong« zu »Champignon« korrigiert.
- ▶ Das Data-Mining-Verfahren *Assoziationsanalyse* findet inhaltliche Zusammenhänge in Playlisten, Warenkörben oder Texten. Unser Beispielprogramm »Begriffsnetz« navigiert durch ein Netz von Assoziationen, die wir aus Sendungsmanuskripten des Deutschlandfunks gewonnen haben. Das Programm lernt etwa, dass die Begriffe »Urknall«, »Weltformel« und »Relativitätstheorie« zusammengehören, ohne eine Idee von Astrophysik oder von überhaupt irgendeinem Thema zu haben.

Kapitel 5 und Kapitel 6 sind der klassischen Spiele-KI gewidmet. Sie bauen inhaltlich aufeinander auf und sollten daher unbedingt hintereinander gelesen werden:

- ▶ Wir haben uns extra für Sie, liebe Leserinnen und Leser, das Ein-Personen-Spiel »Fruchtkräsch« ausgedacht und programmiert. Dabei haben wir uns von Puzzlespielklassikern wie »Candy Crush« oder »Collapse« inspirieren lassen. Wir zeigen, wie eine KI mittels eines Verfahrens namens *Breitensuche* zukünftige Spielzustände vorausrechnen und auf diese Weise den vielversprechendsten Zug ermitteln kann. Sie können Ihre eigenen Fruchtkräsch-Skills direkt mit jenen unserer KI messen.
- ▶ Bei Spielen für zwei Personen reicht es nicht, die denkbaren Spielverläufe per *Breitensuche* zu berechnen, um den jeweils besten Zug zu finden. Die KI muss die den eigenen Interessen entgegengesetzten Absichten einer Gegnerin oder eines Gegners berücksichtigen. Dies leisten die *Tiefensuche* und der *Minimax-Algorithmus*. Im Bei-

spielprogramm können Sie »Reversi« gegen unsere selbst programmierte KI spielen, die Minimax und eine Optimierung namens *Alpha-Beta-Pruning* nutzt.

Kapitel 7 stellt einen grundlegenden Vertreter des *bestärkenden Lernens* vor:

- ▶ *Q-Learning* ist ein automatisches Lernverfahren, bei dem ein Agent (in unserem Fall ein Eichhörnchen) mit einer Umwelt (dem Garten) interagiert, um ein Ziel (das veressene Nussdepot) zu erreichen. Wir führen den Spezialfall des Q-Learnings mittels *Q-Tabelle* im Detail vor.

In **Kapitel 8** und **Kapitel 9** zeigen wir zwei Verfahren, die auf Geometrie beruhen. Sie bauen aufeinander auf und sollten daher gemeinsam gelesen werden:

- ▶ Der *k-nächste-Nachbarn-Algorithmus* ist ein grundlegendes Verfahren zur *Klassifikation* von Daten. Wir lernen, Daten als geometrische Punkte zu interpretieren. Dann können wir nicht klassifizierte *Datenpunkte* einer Klasse zuordnen, indem wir die Entfernungen zu anderen Datenpunkten berechnen, die bereits klassifiziert wurden. Wir führen das praktisch vor mit einer Anwendung, bei der wir anhand wissenschaftlich objektiv gemessener Werte für »Flauschigkeit« und »Niedlichkeit« entscheiden, ob wir es mit einem Häschen, einem Hai, einem Igel oder einer Vogelspinne zu tun haben.
- ▶ Beim *k-means-Clustering-Algorithmus* geht es ebenfalls um die Klassifizierung von Datenpunkten. Allerdings starten wir in diesem Fall mit einer vollständig unklassifizierten Punktmenge. Der Algorithmus ordnet die Datenpunkte in Klassen ein, indem er Gruppen von Punkten findet, die besonders nahe beieinanderstehen. Die praktische Anwendung gruppiert auf diese Weise Messdaten von Wetterstationen.

In **Kapitel 10 bis Kapitel 13** geben wir eine Einführung in die Programmierung *neurona-ler Netze*. Dieser Abschnitt stellt inhaltliche Bezüge zu Kapitel 8 und Kapitel 9 her, die Sie also vorher gelesen haben sollten:

- ▶ Sie lernen, was ein *künstliches Neuron* ist, wie Sie aus Neuronen ein Netz spinnen und wie ein solches *künstliches neuronales Netz* Berechnungen durchführt. Bei dieser Gelegenheit treffen wir das »Häschenproblem« aus Kapitel 8 wieder und lösen es mittels eines neuronalen Netzes.
- ▶ Wir erklären, wie Netze per *überwachtem Lernen* darauf trainiert werden, ein bestimmtes Problem zu lösen, und stellen das *Gradientenabstiegsverfahren* vor, das wir für dieses Training benötigen.
- ▶ Das Beispielprogramm »Neuronentrainer« erlaubt es Ihnen, einem neuronalen Netz beim Lernen zuzuschauen, und lädt zu eigenen Experimenten ein.
- ▶ In Kapitel 13 bieten wir eine flotte Übersicht weiterer Typen und Bauarten neuronaler Netze. Wir erklären Ihnen, was *Faltungsnetze* sind, und führen die Arbeit von *Filterker-*

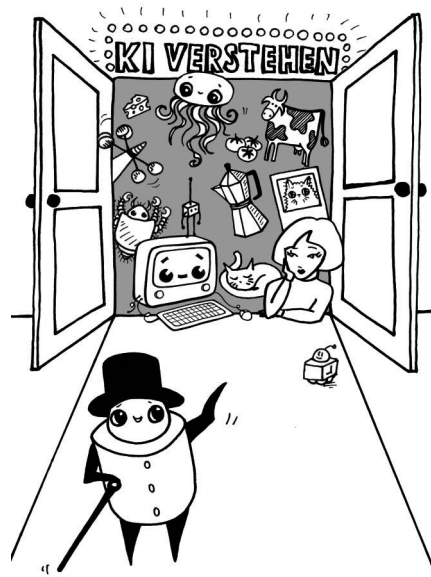
neln anhand eines praktischen Beispiels vor. Zudem skizzieren wir, was es mit *Autoencodern*, *GANs* und *Deep Q-Learning* auf sich hat. Zuletzt kommen wir auf die in dieser Einleitung aufgeworfene Frage nach einer allgemeinen künstlichen Intelligenz zurück.

Das abschließende **Kapitel 14** schlägt die Brücke zu den aktuellsten technischen Entwicklungen.

Es stellt zwei Verfahren vor, die eine unabdingbare Voraussetzung dafür sind, dass *Transformer* wie ChatGPT überhaupt Texte verarbeiten und produzieren können: *Wort- und Sequenzeinbettungen* ordnen Wörter und Sätze so in hochdimensionalen Räumen an, dass sinnhafte und grammatikalische Beziehungen zwischen ihnen als mathematische Relationen berechenbar werden. Es bezieht sich auf fast sämtliche vorangegangene Kapitel und sollte als letztes gelesen werden.

Das **Nachwort** nähert sich dem Thema KI aus einer eher philosophischen und kunstwissenschaftlichen Richtung. Es befasst sich mit der Frage, was es zu bedeuten hat, wenn Computerprogramme Geschichten schreiben, Bilder malen und Musikstücke komponieren.

Jetzt bleibt uns nur noch, Ihnen unterhaltsame und lehrreiche Stunden mit unserem Buch und den Beispielprogrammen zu wünschen. Möge Ihnen das eine oder andere Licht aufgehen! Wenn Sie bei der Lektüre auch nur annähernd so viel Spaß haben, wie uns das Schreiben gebracht hat, und wenn Sie nur einen Bruchteil von dem lernen, was wir beim Schreiben gelernt haben – dann dürfen wir uns sehr glücklich schätzen.



1.9 Dank

An der Entstehung dieses Buchs haben sehr viele Personen mitgewirkt, bei denen wir uns ganz herzlich bedanken:

Stellvertretend für die immer freundlichen und hilfsbereiten Mitarbeiterinnen und Mitarbeiter des Rheinwerk Verlags bedanken wir uns bei der Lektorin Almut Poll. Die Zusammenarbeit mit ihr war eine überaus glückliche Konstellation: Frau Poll hat das Projekt von Anfang an mit Herz und Verstand unterstützt. Es war immer ein Vergnügen, mit ihr didaktische, fachliche und technische Aspekte des Buchs zu diskutieren. Auch für ihre Geduld angesichts unserer künstlerischen Auffassung von Kapitelplanungen und Abgabefristen schulden wir ihr unseren Dank.

Der geschätzte Autorenkollege Torsten T. Will war so freundlich, das komplette Manuskript durchzusehen. Er hat durch seine kompetenten Fragen, Anregungen und Kommentare ganz erheblich zur Verbesserung der Qualität dieses Buchs beigetragen. Marco Rudolph vom Institut für Informationsverarbeitung an der Leibniz Universität Hannover und Lea Reinhart haben weite Teile des Manuskripts gelesen, auf fachliche Korrektheit geprüft, viele wertvolle Hinweise gegeben und Verbesserungsvorschläge gemacht. Torsten, Marco und Lea waren eine unschätzbare Hilfe bei der Realisierung unseres Vorhabens, fachliche Präzision mit anschaulichen Erklärungen zu verbinden. Trotzdem gilt selbstverständlich, dass alle verbleibenden Fehler, Ungereimtheiten und Ungenauigkeiten ganz allein auf unsere Kappe gehen.

Claudia Ley hat in den letzten zwei Jahren einen guten Teil ihrer knappen Freizeit geopfert, um uns bei der Entwicklung der Beispielprogramme und der Texte zu unterstützen. Ihre Geduld und ihr niemals ermüdendes Interesse an dem Projekt haben geholfen, so manche Durststrecke zu überwinden.

Herbert Schmid hat uns bei der Konzeption des Buchs beraten und uns die eine oder andere Perle aus seinem enzyklopädischen IT-Wissen zukommen lassen.

Der Grafiker Ralf Rohde hat auf den letzten Metern bei der Korrektur und Ergänzung von Infografiken sehr wertvolle Dienste geleistet.

Pit Noack bedankt sich zudem bei der Illustratorin Sophia Sanner für die überaus angenehme Zusammenarbeit und bei Lea Reinhart für die Mitarbeit an Kapitel 7 und dem Glossar. Er dankt Stefan Dissmer, Karsten Fecht, Sibylle Feldmann, Vincent Franke, Lukas Greipel, Heiko Idensen, Pina Merkert, Tobias Prinz, Jonas Rahm und Martin Reche – ihr wisst schon, wofür – sowie allen Teilnehmerinnen und Teilnehmern seiner Programmierkurse, vor allem Karin Gaerner und Olivia, Patrik, Rebecca und Tarik Ley.