

1

Einleitung

Wer die Berichterstattung zum Thema künstliche Intelligenz in den Medien verfolgt, dem wird aufgefallen sein, dass die erste Euphorie inzwischen deutlich abgeklungen ist. Das liegt nicht nur daran, dass sich aufgrund der Vielzahl an Berichten ein Gewöhnungseffekt eingestellt hat. Auch haben in vielen Bereichen die Entwicklungen den hohen Erwartungen nicht standgehalten.

Anstatt es uns auf den Rücksitzen autonomer Fahrzeuge bequem zu machen, wie von vielen ExpertInnen noch vor wenigen Jahren vollmundig angekündigt¹, steuern wir unsere Autos im Jahr 2022 noch immer gestresst und von Hand durch die Staus der Innenstädte. Auch um die allgemeine künstliche Intelligenz (*General AI*) ist es inzwischen verdächtig still geworden. Dabei jagte zuvor ein aufsehenerregendes KI-Ereignis das nächste. Man denke nur an IBMs *Watson*, der medienwirksam mehrfach in der Quizshow *Jeopardy* gegen die Champions gewinnen konnte [Franklin2014]. Oder auch Googles *Alpha Go*, das im gleichnamigen Brettspiel gegen den weltbesten Go-Spieler antrat und ihn besiegte. Viele glaubten, dass die Entwicklungen in diesem atemberaubenden Tempo weitergehen würden und dass die Zukunftsvisionen aus Science-Fiction-Filmen und Serien, in denen Humanoide von Menschen kaum noch zu unterscheiden sind, schon bald in greifbare Nähe rücken würden.

Tatsächlich ist der Alltag künstlicher Intelligenz heute wesentlich grauer geworden. Schlagzeilen machen KIs heute eher, weil sie von zweifelhaften Regimen zur umfassenden Überwachung von Menschen eingesetzt werden oder weil sie mit rassistischen oder sexistischen Verfehlungen auffallen [West2019]. Und die ersten Kontakte mit den kleinen Schwestern der Film-Supercomputer verlaufen auch nicht immer so, wie wir uns das vorgestellt haben. Wer hat nicht schon holprige Konversationen mit Alexa geführt, sich mit automatischen Antwortsystemen am Telefon herumgeschlagen, sinnlose Wortvervollständigungen auf Smartphones rückgängig gemacht oder ist mit dem Chatbot auf der Seite einer Bank oder Versicherung aneinandergeraten?

Womöglich hat Judea Pearl (ein Pionier der KI-Entwicklung) recht, wenn er etwas spöttisch zu Protokoll gibt, dass die eindrucksvollen Ergebnisse intelligenter Systeme im Wesentlichen auf *Curve Fitting* zurückgingen. Was er damit meint, ist, dass KI-Systeme heute zwar in der Lage sind, beliebig komplexe Regelmäßigkeiten in Daten aufzuspüren. Was sie dagegen nicht können, ist, zwischen kausalen Beziehungen und bloßen Korrelationen zu unterscheiden [Pearl2018]. Dadurch, so Pearl, seien die Möglichkeiten, weitere signifikante Fortschritte in diesem Bereich zu machen, eher begrenzt – die Revolution fällt also womöglich aus [Marcus2018].

¹ Vgl. <https://www.wired.com/story/self-driving-cars-challenges/> [Abgerufen am 11.09.2021]

Ob das stimmt oder nicht, sei dahingestellt. Gewiss ist allerdings, dass der Siegeszug künstlich intelligenter Systeme trotz aller Beschränktheit nicht aufzuhalten ist. Auch wenn die Anwendungen vielleicht nicht den Utopien oder Dystopien aus den Filmen nahekommen. Systeme, die Machine- und Deep-Learning-Verfahren implementieren, haben sich in den letzten Jahren in rasantem Tempo ausgebreitet. Kaum ein Geschäftsfeld, in dem maschinelle Lernalgorithmen nicht zum Einsatz kommen. Sei es im Marketing, im Journalismus, im Finanzwesen oder in der verarbeitenden Industrie [James201a].

Dabei handelt es sich meist um kleinteilige, problemzentrierte Anwendungen, um *Narrow AI* im Vergleich zu *General AI*. Solche Algorithmen erledigen spezifische Aufgaben für uns: sie übersetzen, produzieren Textzusammenfassungen, handeln an der Börse, warnen vor möglichen Ausfällen eines Aggregats, machen auf Hackerangriffe aufmerksam, generieren Antworten auf Fragen oder bestimmen, welche Videos und welche Werbung wir sehen und welche Posts in unseren Newsfeeds landen.

Wie umfassend diese Vereinnahmung bereits ist, lässt sich nur schwer sagen. Meist können wir von außen nicht sicher sein, ob eine Anwendung von einem maschinellen Lernsystem betrieben wird oder nicht. Sicher ist nur, dass die Technologie bereits heute Teil unseres Alltags ist und aus diesem Alltag nicht mehr verschwinden wird. Wir werden uns daran gewöhnen müssen, in Zukunft noch häufiger mit selbstlernenden Automaten zu interagieren und womöglich mit ihnen zu kooperieren. Viele Berufsbilder werden unter Anpassungsdruck geraten, sich wandeln, und Tätigkeiten, für die bis vor Kurzem noch menschliche Arbeitskraft und kognitive Fertigkeiten notwendig waren, werden ganz oder teilweise von Algorithmen übernommen [Acemoglu2018].

Dieses Buch handelt von dieser Art künstlicher Intelligenz. Von Anwendungen, die im Hintergrund operieren. Keine davon wird vermutlich bei Ihnen das Gefühl auslösen, dass Sie es mit einem künstlichen Bewusstsein à la *HAL* oder einem der Androiden aus *Westworld* zu tun haben. Das heißt aber nicht, dass diese Anwendungen nicht auf ihre Weise intelligent und faszinierend sind.

Automaten, die menschliche Sprache imitieren, hinterlassen bei den meisten Menschen einen etwas ambivalenten Eindruck. Dass ein Algorithmus einen Satz produziert, der aus der Feder eines Philosophen oder Dichters stammen könnte, ist zumindest gewöhnungsbedürftig. Je mehr man aber hinter die Kulissen blickt und versteht, wie solche Systeme arbeiten, was sie lernen und was sie nicht lernen und wovon ihre Leistung abhängt, umso weniger mysteriös wirken sie.

Was Ihnen dieses Buch bietet

Dieses Buch bietet eine Einführung in den Bereich der Verarbeitung natürlicher Sprachdaten (Natural Language Processing) mit Deep Learning-Verfahren. Es richtet sich an alle LeserInnen, die sich für diese Schnittmenge aus linguistischer Analyse und maschinellem Lernen interessieren. An Personen, die genauer wissen möchten, wie sich Textdaten mit den neuesten statistischen Verfahren verarbeiten lassen und wie Übersetzungen, Textvervollständigungen, Textzusammenfassungen, Sentiment-Analysen und andere textbasierte Anwendungen in der Theorie funktionieren und wie sie sich in der Praxis implementieren lassen.

Um mit diesem Buch arbeiten zu können, sollten Sie ein wenig Programmiererfahrung mit Python mitbringen. Sie sollten wissen, wie Sie eine virtuelle Umgebung mit *Anaconda* aufsetzen, wie sich Pakete installieren lassen, und Sie sollten mit der Arbeit mit Klassen, Funktionen, Schleifen und Standardbehältern wie Listen, Dictionaries, Tuples und Sets ver-

traut sein. Außerdem brauchen Sie ein paar grundlegende Konzepte aus der Statistik. So sollten Sie sich unter Begriffen wie *Variable*, *Verteilung*, *Mittelwert* und *Standardabweichung* etwas vorstellen können und generell keine größeren Phobien vor Zahlen haben. Viel mehr brauchen Sie nicht.

Dies ist kein Buch, in dem es in erster Linie um die mathematischen Grundlagen des Deep Learnings geht. Der Text kommt in weiten Teilen ohne Formeln aus, obwohl wir sie manchmal der Vollständigkeit halber abdrucken. Im Vordergrund stehen textbasierte Erklärungen der wichtigsten Konzepte und eine Umsetzung dieser Konzepte mithilfe gängiger Python-Frameworks wie *TensorFlow/Keras*, *Scikit-Learn* oder *Transformers*. Wenn Sie im Detail wissen möchten, wie die Algorithmen mathematisch implementiert sind, sollten Sie parallel ein anderes Buch lesen. Wenn Ihnen mathematische Formeln generell näherstehen als textbasierte Erklärungen, ist dieses Buch vermutlich eher nicht die richtige Wahl für Sie.

Wenn Sie sich aber dafür entscheiden, dieses Buch zu lesen, haben Sie am Ende eine ziemlich klare Vorstellung darüber, wie moderne Verfahren in der statistischen Sprachanalyse arbeiten. Sie lernen nicht nur die Funktionsweise, die Möglichkeiten und Grenzen dieser Verfahren kennen. Sie verstehen auch die Grundprinzipien rekurrenter und konvolutionaler Netze und wissen, wie moderne Architekturen wie Sequence-to-Sequence-Modelle oder Transformer arbeiten. Außerdem ist Ihnen klar, welche Verfahren für welche Aufgaben geeignet sind, und Sie können die meisten dieser Verfahren selbst implementieren. Sie können eigene neuronale Netze aufbauen und anlernen, um Texte zu klassifizieren, zu generieren oder in eine andere Sprache zu übersetzen.

Darüber hinaus bietet Ihnen dieses Buch einen Einstieg in die Verwendung neuronaler Netze, die von anderen vortrainiert und zur Verfügung gestellt werden. Sie lernen, wie Sie allgemeine Sprachmodelle, die über sehr große Datenmengen angelernt wurden, laden und einsetzen können, und Sie lernen, wie Sie diese Netze nachtrainieren können, um sie für spezifische Aufgaben nutzbar zu machen.

Welche Inhalte Sie erwarten

Dieses Buch bietet eine schrittweise und kompakte Einführung in den Themenkomplex Machine Learning/Natural Language Processing (NLP). Die Kapitel bauen inhaltlich aufeinander auf. Wenn Sie also eines der Kapitel lesen, sollten Sie die Inhalte aus den vorherigen Kapiteln kennen. Konzepte, Klassen oder Funktionen werden in der Regel nur an einer Stelle erklärt, und Sie können davon ausgehen, dass einmal eingeführtes Wissen später wieder gebraucht wird. Wenn Sie neu in diesem Gebiet sind, ist es daher ratsam, die Kapitel nacheinander zu lesen, damit das didaktische Konzept aufgeht. Wenn Sie schon über Kenntnisse im einen oder anderen Bereich verfügen, können Sie natürlich einzelne Kapitel oder auch mehrere Kapitel überspringen.

Wir starten unsere Reise in die Welt des Deep Learnings mit einer allgemeineren Einführung in den Bereich maschinellen Lernens. Bevor wir uns an komplexen Modellen abarbeiten, sollten wir eine konkrete Vorstellung davon haben, wie einfache Modelle funktionieren. Wir machen uns also mit den Grundlagen statistischer Lernverfahren und den Basics im Umgang mit Daten in Python vertraut.

Deshalb beginnen wir im Anschluss an die Einleitung im **zweiten Kapitel** mit einer Einführung in die Behälterklassen, die Python standardmäßig bietet, und in einige externe Pakete und Funktionen wie *NumPy*, die uns die Arbeit mit Textdaten und Matrizen erleichtern.

Das **dritte Kapitel** widmet sich den allgemeinen Grundlagen maschinellen Lernens. Wir schauen uns an, wie einfache Lernzellen arbeiten, wie man Beziehungen zwischen x - und y -Variablen modelliert, was es mit der Optimierung von Gewichten und dem Gradientenabstiegsverfahren auf sich hat und wie sich mithilfe von Aktivierungsfunktionen beliebige Zielwerte (stetige und kategoriale Werte) produzieren lassen.

In **Kapitel 4** steigen wir dann in die Analyse von Textdaten ein. Dabei geht es zunächst um bewährte Vorverarbeitungsverfahren wie den Bag-of-Words-Ansatz, der es ermöglicht, Textdaten auf sehr basale Art mit maschinellen Standardverfahren zu verarbeiten. **Kapitel 5** bietet im Anschluss einen Einstieg in das eigentliche Zielthema des Buches: in die Arbeit mit neuronalen Netzen. Auch hier müssen wir uns zuerst einiger grundlegender Verfahren versichern, ehe wir uns um die speziellen, auf die Interpretation und Produktion von Texten zugeschnittenen Architekturen kümmern. In diesem Kapitel geht es daher um die Basics, die für die praktische Arbeit mit neuronalen Netzen erforderlich sind. Angefangen von der Funktionsweise einzelner Neuronen, über Schichten von Neuronen, das Durchschleusen von Daten durch diese Schichten bis hin zur Verlustfunktion und der Optimierung der Gewichte mithilfe des Backpropagation-Mechanismus. Danach wissen Sie, wie ein neuronales Netz lernt, wie Gewichte initialisiert und optimiert werden und welche Aufgaben Ihnen bei der Zusammenstellung eines solchen Netzes zukommen.

Mit diesem Wissen ausgerüstet, erörtern wir in **Kapitel 6** eine erste, für die Verarbeitung von Textdaten konzipierte Netzarchitektur: rekurrente Netze. Dabei handelt es sich um ein Verfahren, das im Speziellen die zeitliche Abfolge der Präsentation von Zeichen – zum Beispiel von Buchstaben in Wörtern oder von Wörtern in Sätzen – berücksichtigt und auf dieser Grundlage Interpretationen erzeugt. In **Kapitel 7** geht es nahtlos mit konvolutionalen Netzen weiter. Zwar hat sich dieses Verfahren vor allem bei der Verarbeitung von Bild- und Audiodaten bewährt, es lässt sich aber ohne Weiteres auf Sequenzdaten übertragen. Die Vor- und Nachteile gegenüber rekurrenten Netzen sehen wir uns dabei anhand einer automatischen Rechtschreibkorrektur genauer an. Das folgende **Kapitel 8** thematisiert dann die Vorverarbeitung von Wörtern mit Worteinbettungsverfahren. Damit lassen sich Wörter bzw. Token als Vektoren in einem mehrdimensionalen Raum unter Berücksichtigung semantischer und grammatikalischer Beziehungen darstellen. Das Verfahren bildet heute die Grundlage für fast alle Deep Learning-Ansätze, die Textdaten als Rohmaterial verwenden. Wir beleuchten dabei nicht nur die Möglichkeiten, Worteinbettungen selbst anzulernen, sondern führen auch in die Verwendung vortrainierter Vektorräume wie *fastText* oder *Word2Vec* ein.

Beginnend mit **Kapitel 9** eruieren wir dann die Optionen, die komplexe neuronale Netze bei der Textinterpretation und Textgenerierung bieten. Insbesondere geht es um Encoder-Decoder-Modelle. Sie sind aus der modernen Textanalyse (sei es in Sequence-to-Sequence- oder in Transformer-Modellen) nicht mehr wegzudenken. Zunächst dreht sich aber alles um das Handwerkszeug, das wir brauchen, um solche Modelle praktisch umzusetzen: Es geht darum, wie wir Netze, die über zwei Eingänge verfügen oder in denen Datenströme bestimmte Schichten überspringen, aufsetzen können. Diese Kenntnisse sind notwendig, wenn wir im nachfolgenden **Kapitel 10** Sequence-to-Sequence-Modelle genauer unter die Lupe nehmen. Mit dieser Architektur lassen sich nicht nur einzelne Wörter, sondern ganze Sätze oder Textpassagen erzeugen. Solche Architekturen werden für Übersetzungen oder Textzusammenfassungen eingesetzt. Darüber hinaus lernen wir in diesem Kontext den Attention-Mechanismus kennen, der in einer Variante, der Self-Attention, auch in Transformer-Modellen Karriere gemacht hat. Das ist das Thema von **Kapitel 11**. Darin zeigen wir

nicht nur, wie Transformer-Modelle funktionieren, sondern wir sehen uns auch an, wie sich vortrainierte Modelle, die über den Transformer-Hub *Hugging Face* vertrieben werden, sich für verschiedene Aufgaben nachtrainieren lassen. Das Buch schließt im **zwölften Kapitel** mit einer Zusammenfassung einiger grundlegender Konzepte und einer Diskussion der Stärken und Schwächen neuronaler Netze.

Wie Sie mit den Codebeispielen arbeiten können

Die meisten Kapitel bestehen aus einem Mix aus Einführungen in Konzepte maschinellen Lernens und aus einem praxisorientierten Teil, in denen diese Konzepte mit Python umgesetzt werden. Wenn Sie die Beispiele auf Ihrem Rechner laufen lassen möchten, müssen Sie die Voraussetzungen dafür schaffen.

Wir verwenden Python in der *Version 3.7.6*, eingebettet in eine virtuelle Umgebung, die über die Data Science-Plattform *Anaconda* verwaltet wird. *Anaconda* bietet unter anderem den Vorteil, dass es bei der Installation externer Pakete sicherstellt, dass die Pakete untereinander harmonieren. Da wir einige Pakete installieren müssen und da diese Pakete in ihrer Arbeit wiederum auf weitere Unterpakete angewiesen sind, ist das ziemlich hilfreich.


Um also arbeitsfähig zu sein, benötigen Sie die folgenden Bibliotheken in Ihrer virtuellen Umgebung:

```
matplotlib, Version=3.4.2
nlTK, Version=3.6.2
numpy, Version=1.19.2
pandas, Version=1.0.3
scikit-learn, Version=0.22.1
tensorflow, Version=2.3.0
transformers, Version=4.10.2
```

Bis auf *Transformers* (das zum Zeitpunkt der Drucklegung mit *pip* installiert werden muss) sind alle Bibliotheken über *Anaconda* verwaltbar (installieren mit *conda install*). Sie können den Code vermutlich auch mit anderen, aktuelleren Versionen von Python und den genannten Bibliotheken ausführen. Allerdings ist es möglich, dass Sie dann an der einen oder anderen Stelle auf Warnungen, Fehlermeldungen oder auf andere Probleme stoßen, die wir nicht vorhersehen können.

Die Codebeispiele im Buch sind über *GitHub* als *Jupyter Notebooks* verfügbar. Sie können den Code entweder im Internet unter https://github.com/tplusone/hanser_deep_nlp abrufen und ansehen oder, wenn Sie die Software *Git* installiert haben, das gesamte Repository inklusive Codebeispielen und Beispieldaten über das Terminal mit dem folgenden Befehl auf Ihren Rechner ziehen:

```
git clone https://github.com/tplusone/hanser_deep_nlp.git
```

Diese Leseprobe haben Sie beim
 edv-buchversand.de heruntergeladen.
Das Buch können Sie online in unserem
Shop bestellen.

[Hier zum Shop](#)