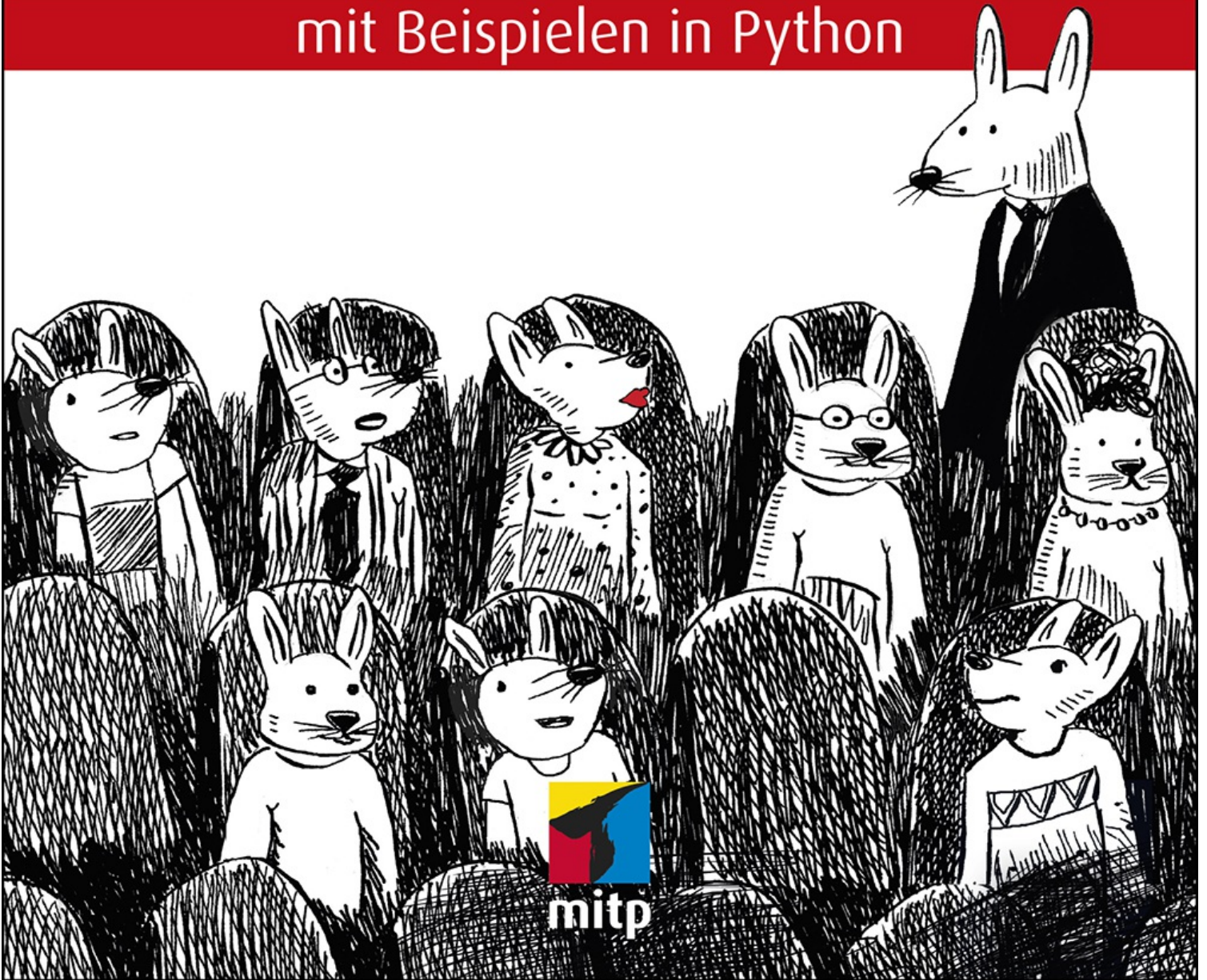


ANDREW W. TRASK

NEURONALE NETZE UND DEEP LEARNING KAPIEREN

DER EINFACHE PRAXISEINSTIEG

mit Beispielen in Python



Leistungsfähigkeit der Technologie kein Zurück mehr gibt. Ich kann nur hoffen, dass du und deine Bekannten mithilfe dieses Buchs von einer vor Umwälzungen stehenden Branche in ein Fachgebiet wechseln können, dem Wachstum und Wohlstand bevorstehen: Deep Learning.

Was es bedeutet, ein Mensch zu sein, kannst du herausfinden, indem du versuchst, Intelligenz und Kreativität zu simulieren.

Ich persönlich bin zum Deep Learning gekommen, weil ich es faszinierend finde. Es handelt sich um eine erstaunliche Schnittmenge von Mensch und Maschine. Genau aufzuschlüsseln, was es eigentlich bedeutet, zu denken, zu schlussfolgern und Neues zu erschaffen, ist erhellend, fesselnd und – für mich jedenfalls – inspirierend. Stell dir vor, dass du über eine Datenmenge verfügst, die sämtliche Bilder enthält, die jemals gemalt worden sind, und sie dazu verwendest, einer Maschine beizubringen, wie Monet zu malen. Es ist verrückt, aber das ist möglich – und es ist unfassbar cool, zu beobachten, wie das funktioniert.

1.3 Ist es schwierig, Deep Learning zu verstehen?

Wie hart musst du arbeiten, bis es eine »ansehnliche« Belohnung gibt?

Das ist meine Lieblingsfrage. Mit einer »ansehnlichen« Belohnung meine ich, zu beobachten, dass etwas von mir Erstelltes von sich aus *lernt*. Dabei zuzusehen, wie etwas mit eigenen Händen Erschaffenes das vollbringt, ist etwas ganz Besonderes. Wenn du das auch so siehst, ist die Antwort einfach. Am Anfang von [Kapitel 3](#) wirst du dein erstes neuronales Netz erstellen. Die einzige damit verbundene Arbeit ist das Lesen der Seiten von hier bis dorthin.

Nach [Kapitel 3](#) folgt die *nächste* ansehnliche Belohnung, wenn du dir ein paar Zeilen Code gemerkt hast und weiterliest bis zur Mitte von [Kapitel 4](#). Das läuft in allen Kapiteln so ab: Du merkst dir einen kleinen Codeabschnitt aus dem vorangegangenen Kapitel, liest das nachfolgende und erhältst dann die Belohnung in Form eines neuen *lernenden* neuronalen Netzes.

1.4 Warum du dieses Buch lesen solltest

Das Buch ermöglicht einen besonders einfachen Einstieg.

Der Grund dafür, dass du dieses Buch lesen solltest, ist der gleiche, aus dem ich es geschrieben habe. Mir ist keine andere Ressource bekannt (Buch, Kurs, Blogbeitragsreihen), die Deep Learning lehrt, *ohne fortgeschrittene mathematische Kenntnisse* (einen Hochschulabschluss in einem mathematischen Bereich) *vorauszusetzen*.

Versteh das nicht falsch: Es gibt gute Gründe dafür, dabei auf Mathematik zurückzugreifen. Mathematik ist letzten Endes eine Sprache, die zweifelsohne ein *effizienteres* Lehren von Deep Learning ermöglicht, aber ich halte es nicht für unbedingt notwendig, fortgeschrittene Mathematikkenntnisse zu besitzen, um ein fähiger und kenntnisreicher Entwickler zu werden, der das »Wie« beim Deep Learning gründlich verstanden hat.

Und weshalb solltest du nun Deep Learning mithilfe dieses Buchs erlernen? Weil ich davon ausgehe, dass du über Mathematikkenntnisse auf Gymnasialniveau verfügst (die auch nicht mehr ganz frisch sind), und ich *alles andere erklären werde*, wenn es vorkommt. Kannst du dich an Multiplikation und x-y-Koordinatensysteme erinnern (die Quadrate mit den Kurven darin)? Perfekt! Das genügt.

Das Buch hilft dir zu verstehen, was *innerhalb* eines Frameworks (wie Torch, TensorFlow usw.) vor sich geht.

Es gibt im Wesentlichen zwei Gruppen von Lehrmaterialien (wie Bücher oder Kurse) über Deep Learning. Die eine Gruppe konzentriert sich auf die Verwendung verbreiteter Frameworks und Codebibliotheken wie Torch, TensorFlow, Keras und andere. Die zweite Gruppe konzentriert sich auf das Deep Learning selbst, also auf die *wissenschaftlichen Aspekte hinter den Kulissen* der großen Frameworks.

Letzten Endes sind Kenntnisse über beide Gruppen wichtig. Wenn du ein NASCAR-Fahrer werden willst, musst du nicht nur dein Automodell kennen (das Framework), sondern auch wissen, wie man fährt (die Wissenschaft und Kenntnisse). Sich nur mit dem Framework zu befassen, ist damit vergleichbar, die Vor- und Nachteile eines Chevrolet SS der 6. Generation auswendig zu lernen, bevor man überhaupt weiß, was ein Schaltknüppel ist. In diesem Buch geht es darum, zu lehren, was Deep Learning ist, damit du darauf vorbereitet bist, ein Framework zu erlernen.

Alles, was mit Mathematik zu tun hat, wird durch anschauliche Analogien erklärt.

Wenn ich auf mathematische Formeln stoße, unternehme ich die beiden folgenden Schritte: Zuerst übersetze ich das Verfahren in eine anschauliche Analogie. Ich nehme eine Formel nur selten einfach so hin, wie sie ist: Ich zerlege sie in Teile, die für sich selbst genommen etwas aussagen. Diesen Ansatz verfolgt auch das Buch. Wenn es um mathematische Konzepte geht, biete ich eine Analogie an, die erläutert, was eine Formel eigentlich leistet.

Man muss die Dinge so einfach wie möglich machen, aber nicht einfacher.
– Albert Einstein zugeschrieben

Nach den einführenden Kapiteln ist alles projektbasiert.

Eines kann ich beim Lernen gar nicht leiden: wenn ich mich fragen muss, ob das, was ich lerne, nützlich oder von Bedeutung ist. Wenn mir jemand alles über einen Hammer beibringen möchte, ohne mir ein paar Nägel in die Hand zu drücken, die ich dann einschlagen kann, dann bringt man mir nicht wirklich bei, wie man einen Hammer benutzt. Ich weiß, dass noch irgendetwas fehlt, und wenn ich dann mit einem Hammer und einer Schachtel Nägel vor einem Stapel Kantholz stehe, bin ich auf Mutmaßungen angewiesen.

In diesem Buch geht es darum, dir Holz, Nägel und Hammer bereitzustellen, *bevor* erklärt wird, was man damit anfangen kann. Bei jeder Übung stehen Werkzeug und Arbeitsmaterial zur Verfügung, und wie etwas funktioniert, wird erklärt, während es erledigt wird. Auf diese Weise stehst du am Ende nicht nur mit einer Liste der vorhandenen Deep-Learning-Tools da, sondern verfügst über die Fähigkeit, sie einzusetzen, um Aufgaben zu lösen. Darüber hinaus lernst du das, was am wichtigsten ist, nämlich wann und weshalb ein Werkzeug für die Aufgaben, die du lösen möchtest, geeignet ist. Diese Kenntnisse versetzen dich in die Lage, eine Laufbahn in der Forschung oder der Branche einzuschlagen.

1.5 Was du brauchst, um loszulegen

Installiere Jupyter Notebook und die Python-Bibliothek NumPy.

Am liebsten verwende ich Jupyter Notebooks. Beim Erlernen von Deep Learning gehört es – jedenfalls für mich – zu den wichtigsten Dingen, das Training eines Netzes abbrechen zu können, um es bis ins kleinste Detail zu untersuchen. Hier erweisen sich Jupyter Notebooks als außerordentlich nützlich.

Was NumPy betrifft, werden wir nur eine Matrix-Bibliothek verwenden. Auf diese Weise lernst du, wie alles *funktioniert*, nicht nur, wie man ein Framework benutzt. Dieses Buch

lehrt Deep Learning von Grund auf, von A bis Z.

Installationsanleitungen für die beiden Tools sind unter <http://jupyter.org> und <http://numpy.org> zu finden. Ich habe die Beispiele mit Python 2.7 erstellt, sie aber auch unter Python 3 getestet. Zur einfachen Installation empfehle ich außerdem das Framework Anaconda: <https://docs.continuum.io/anaconda/install>.

Mathematik auf Gymnasialniveau

Ein Teil der Mathematik geht über den Rahmen dieses Buchs hinaus, aber mein Ziel ist es, Deep Learning nur mithilfe von grundlegenden Algebra-Kenntnissen zu lehren.

Suche nach einer Aufgabe, an der du persönlich interessiert bist.

Um loszulegen, ist das eigentlich keine echte Notwendigkeit. Aber im Ernst: Ich kann nur dringend empfehlen, sich eine solche Aufgabe zu suchen. Alle mir bekannten Leute, die auf diesem Gebiet Erfolg hatten, wollten eine ganz bestimmte Aufgabe lösen. Sich mit Deep Learning zu befassen, war eigentlich nur ein Mittel zum Zweck, um eine andere interessante Aufgabe zu lösen.

Ich wollte Twitter verwenden, um Aktienkurse vorherzusagen. Für mich war das einfach faszinierend und mein Antrieb, das nächste Kapitel zu lesen und den nächsten Prototyp zu entwickeln.

Wie sich herausstellte, ist dieses Fachgebiet noch *so neu* und ändert sich *so schnell*, dass man eher, als man denkt, zu einem führenden Experten für eine bestimmte Aufgabe wird, wenn man sich die nächsten Jahre damit befasst, sein Ziel zu verfolgen. In meinem Fall führte das dazu, dass ich innerhalb von 18 Monaten von jemandem, der praktisch nichts über Programmierung wusste, zu einem Forschungsstipendiaten mit dem Schwerpunkt Hedgefonds wurde, der das Gelernte anwenden konnte. Beim Deep Learning kann eine faszinierende Aufgabe, die eine Datenmenge nutzt, um eine andere vorherzusagen, ein entscheidender Katalysator sein. Such dir also eine solche Aufgabe!

1.6 Python-Kenntnisse sind nützlich

Python ist meine bevorzugte Programmierumgebung, aber ich stelle online auch einige andere Sprachen zur Verfügung.

Python ist eine erstaunlich intuitive Sprache. Ich denke, Python dürfte die verbreitetste intuitiv verständliche Programmiersprache sein, die es gibt. Darüber hinaus hat die Python-Community ein Faible für Einfachheit, das nicht zu schlagen ist. Aus diesen Gründen werde ich für alle Beispiele Python verwenden (Python 2.7, um genau zu sein, die Beispiele sind aber auch unter Python 3 getestet). Der Quellcode ist unter <https://github.com/iamtrask/Grokking-Deep-Learning> verfügbar.

Wie viel Erfahrung mit der Programmierung ist erforderlich?

Wirf doch einmal einen Blick auf den englischen Kurs *Python Codecademy* (<http://www.codecademy.com/learn/python>). Wenn du das Inhaltsverzeichnis verstehst und die genannten Begriffe dir geläufig sind, bist du bereit! Falls nicht, dann absolviere den Kurs und lies weiter, wenn du damit fertig bist. Der Kurs ist für Einsteiger gedacht und sehr gut gestaltet.

1.7 Zusammenfassung

Wenn du Jupyter Notebooks verwenden kannst und dir die Grundlagen von Python vertraut sind, bist du bereit für das nächste Kapitel! Es ist das letzte Kapitel, das vornehmlich dialogorientiert ist (ohne Programme zu erstellen). Es soll dir einen Eindruck vom Vokabular der Sprache, von Konzepten und Fachgebieten der künstlichen Intelligenz, des Machine Learnings und – am wichtigsten – des Deep Learnings verschaffen.