

Grundbegriffe des maschinellen Lernens

Clemens Dautermann

17. November 2019

Inhaltsverzeichnis

1 Was ist maschinelles Lernen?	3
1.1 Einsatzgebiete maschinellen Lernens	3
2 Neuronale Netze	3
2.1 Maschinelles Lernen und menschliches Lernen	3
2.2 Der Aufbau eines neuronalen Netzes	4
2.3 Berechnung des Ausgabevektors	7
2.4 Der Lernprozess	7
2.4.1 Backpropagation	7
2.4.2 Fehlerfunktionen	7
2.4.3 SGD	7
2.4.4 Zusammenfassung	7
2.5 Verschiedene Layerarten	7
2.5.1 Fully connected Layers	7
2.5.2 Convolutional Layers	7
2.5.3 Pooling Layers	7
3 PyTorch	7
3.1 Datenvorbereitung	7
3.2 Definieren des Netzes	7
3.3 Trainieren des Netzes	7
4 Fallbeispiel I: Ein Klassifizierungsnetzwerk für handgeschriebene Ziffern	7
4.1 Aufgabe	7
4.2 Der MNIST Datensatz	7
4.3 Fragmentbasierte Erkennung	7
4.4 Ergebnis	7
5 Fallbeispiel II: Eine selbsttrainierende KI für Tic-Tac-Toe	7
5.1 Das Prinzip	7
5.2 Chance-Tree Optimierung	7
5.3 Lösung mittels eines neuronalen Netzes	7
5.4 Vergleich	7
6 Schlusswort	7

1 Was ist maschinelles Lernen?

Die wohl bekannteste und am häufigsten zitierte Definition des maschinellen Lernens stammt von Arthur Samuel aus dem Jahr 1959. Er war Pionier auf diesem Gebiet und rief den Begriff „machine learning“ ins Leben. So sagte er:

[Machine learning is the] field of study that gives computers the ability to learn without being explicitly programmed[1].

—Arthur Samuel, 1959

Beim maschinellen Lernen werden Computer also nicht mit einem bestimmten Algorithmus programmiert um eine Aufgabe zu lösen, sondern lernen eigenständig diese Aufgabe zu bewältigen. Dies geschieht zumeist, indem das Programm aus einer großen, bereits „gelabelten“, Datenmenge mit Hilfe bestimmter Methoden, die im Folgenden weiter erläutert werden sollen, lernt, gewisse Muster abzuleiten um eine ähnliche Datenmenge selber „labeln“ zu können. Als Label bezeichnet man in diesem Fall die gewünschte Ausgabe des Programmes. Dies kann beispielsweise eine Klassifikation sein. Soll das Programm etwa handgeschriebene Ziffern erkennen können, so bezeichnet man das (bearbeitete) Bild der Ziffer als „Input Vector“ und die Information welche Ziffer der Computer hätte erkennen sollen, als „Label“. Soll jedoch maschinell erlernt werden, ein simuliertes Auto zu fahren, so bestünde der Input Vector aus Sensorinformationen und das Label würde aussagen, in welche Richtung das Lenkrad hätte gedreht werden sollen, wie viel Gas das Programm hätte geben sollen oder andere Steuerungsinformationen. Der Input Vector ist also immer die Eingabe, die der Computer erhält um daraus zu lernen und das Label ist die richtige Antwort, die vom Programm erwartet wurde. Für maschinelles Lernen wird also vor allem eins benötigt: Ein enormer Datensatz, der bereits gelabelt wurde, damit das Programm daraus lernen kann.

Natürlich werden für maschinelles Lernen trotzdem Algorithmen benötigt. Diese Algorithmen sind jedoch keine problemspezifischen Algorithmen, sondern Algorithmen für maschinelles Lernen. Eine der populärsten Methoden des maschinellen Lernens ist das sogenannte „Neuronale Netz“.

1.1 Einsatzgebiete maschinellen Lernens

2 Neuronale Netze

bei Neuronalen Netzen handelt es sich um eine programminterne Struktur, die für das maschinelle Lernen genutzt wird. Wie der Name bereits vermuten lässt, ist diese Methode ein Versuch das menschliche Lernen nachzuahmen.

2.1 Maschinelles Lernen und menschliches Lernen

Das menschliche Gehirn ist aus sogenannten „Neuronen“ aufgebaut. Ein Neuron ist eine Nervenzelle, die elektrische oder chemische Impulse annimmt, und gegebenen-

falls einen elektrischen oder chemischen Impuls weitergibt. Die Nervenzellen berühren sich nicht direkt sondern sind nur über die sogenannten Synapsen verbunden, über die diese Signale übertragen werden, sodass sich ein hoch komplexes Netzwerk von milliarden von Neuronen ergibt.¹ Ein neuronales Netz ist ähnlich aufgebaut. Es

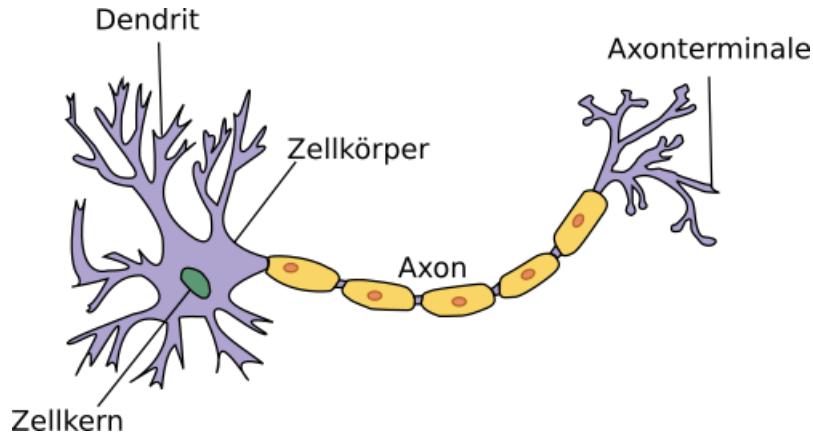


Abbildung 1: Ein Neuron wie es im Gehirn vorliegt

besteht aus „Neuronen“, die eine theoretisch beliebige Anzahl von Eingaben annehmen können und mit einer entsprechenden Ausgabe reagieren, sowie Verbindungen zwischen den Neuronen. Auch das Lernprinzip entspricht dem eines Menschen. Das Netz nimmt immer Zahlen zwischen 0 und 1 als Eingabe an und berechnet eine entsprechende Ausgabe. Es erhält anschließend die Information, wie die richtige Lösung gelautet hätte und lernt dann aus seinen Fehlern, indem es gewisse Werte, die in die Berechnung einfließen, anpasst. Analog lernt ein Mensch, indem er ausprobiert, gegebenenfalls scheitert, anschließend die richtige Antwort durch eine externe Quelle erhält und somit aus seinem Fehler lernt. Im Menschlichen Gehirn verknüpfen sich Dabei oft genutzte neuronale Verbindungen stärker und weniger benutzte Verbindungen bauen sich ab[2]. Die Verstärkung und der Abbau entsprechen dem Ändern der Gewichtung einer Verbindung im neuronalen Netz. Die Gewichtung ist eine Eigenschaft der Verbindung, die eine zentrale Rolle in der Berechnung spielt und soll im folgenden weiter erläutert werden. Diese Ähnlichkeiten sind kein Zufall, sondern viel mehr Intention. Ein neuronales Netz ist nämlich der gezielte Versuch das menschliche Lernen nachzuahmen um maschinelles Lernen zu ermöglichen.

2.2 Der Aufbau eines neuronalen Netzes

Ein neuronales Netz besteht aus Neuronen und Verbindungen zwischen diesen. Es gibt einen sogenannten „Input Layer“, der die Daten annimmt, eine beliebige Anzahl von sogenannten „Hidden Layers“, in denen das eigentliche Lernen statt findet, und

¹ Diese Definition ist stark vereinfacht. Sie enthält ausschließlich die wesentlichen Komponenten um das menschliche Gehirn mit einem neuronalen Netz vergleichen zu können.

einen sogenannten „Output Layer“, der für die Datenausgabe verantwortlich ist. Die Anzahl der Neuronen ist nach oben nicht begrenzt, wird jedoch zumeist der Aufgabe angepasst. Im Input Layer ist meist ein Neuron pro Pixel des Eingabebildes vorhanden und im Output Layer ein Neuron pro möglicher Ausgabe. Sollen also 28×28 Pixel große Bilder handgeschriebener Ziffern klassifiziert werden, so gibt es 784 Eingabeneuronen, da jedes Bild 784 Pixel groß ist, und 10 Ausgabeneuronen, da es 10 Ziffern gibt. Jedes Neuron hat außerdem eine sogenannte Aktivierungsfunktion, die sich von

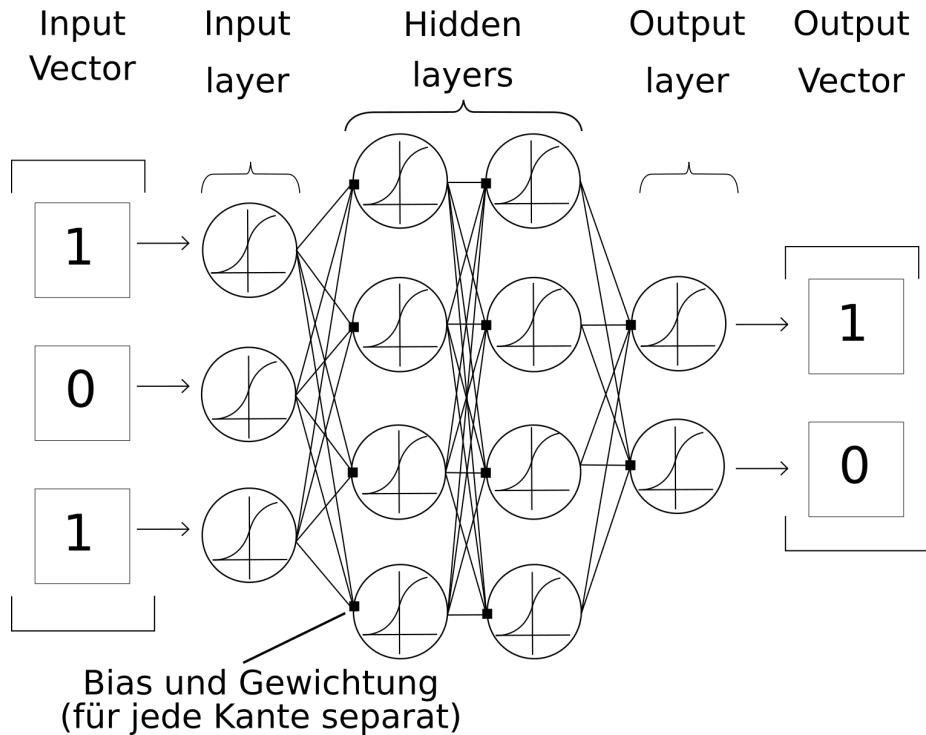


Abbildung 2: Ein einfaches neuronales Netz

Neuron zu Neuron unterscheiden kann, und jede Kante eine assozierte Gewichtung und einen Bias. Ein neuronales Netz besteht also aus:

1. Neuronen mit gegebenenfalls verschiedenen Aktivierungsfunktionen, aufgeteilt in ein Input-, beliebig viele Hidden- und ein Output-Layer.
2. Verbindungen zwischen diesen Neuronen, die jeweils einen eigenen Bias und eine Gewichtung besitzen.

Sind alle Neuronen eines Layers jeweils mit allen Neuronen des nächsten Layers verbunden, wird das Layer als „fully connected layer“ bezeichnet.

2.3 Berechnung des Ausgabevektors**2.4 Der Lernprozess****2.4.1 Backpropagation****2.4.2 Fehlerfunktionen****2.4.3 SGD****2.4.4 Zusammenfassung****2.5 Verschiedene Layerarten****2.5.1 Fully connected Layers****2.5.2 Convolutional Layers****2.5.3 Pooling Layers****3 PyTorch****3.1 Datenvorbereitung****3.2 Definieren des Netzes****3.3 Trainieren des Netzes****4 Fallbeispiel I:**

Ein Klassifizierungsnetzwerk für handgeschriebene Ziffern

4.1 Aufgabe**4.2 Der MNIST Datensatz****4.3 Fragmentbasierte Erkennung****4.4 Ergebnis****5 Fallbeispiel II:**

Eine selbsttrainierende KI für Tic-Tac-Toe

5.1 Das Prinzip**5.2 Chance-Tree Optimierung****5.3 Lösung mittels eines neuronalen Netzes****5.4 Vergleich****6 Schlusswort**

Literatur

[1] Hands-On Machine Learning with Scikit-Learn and TensorFlow
von Aurélien Géron
Veröffentlicht: March 2017 O'Reilly Media, Inc
ISBN: 9781491962282

[2] Die Logistik des Lernens eine Studie der LMU München
Quelle: [www.uni-muenchen.de/forschung/news/2013/
f-71-13_kiebler_nervenzellen.html](http://www.uni-muenchen.de/forschung/news/2013/f-71-13_kiebler_nervenzellen.html) –abgerufen am 16.11.2019

Abbildungsverzeichnis

1	Neuron	4
	Quelle: simple.wikipedia.org/wiki/File:Neuron.svg	
	Copyright: CC Attribution-Share Alike von Nutzer Dhp1080, bearbeitet	
2	Ein einfaches neuronales Netz	5