

Maschinelles Lernen im Kontext der Programmierung natürlicher Sprachen

Seminar Paper of

Weinmann Philipp

At the Department of Informatics
Institut für Programmstrukturen
und Datenorganisation (IPD)

Advisor: Dipl. Inform. Alexander Wachtel

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

I followed the rules for securing a good scientific practice of the Karlsruhe Institute of Technology (Regeln zur Sicherung guter wissenschaftlicher Praxis im Karlsruher Institut für Technologie (KIT)).

Karlsruhe, [FILL OUT DATE HERE]

.....
(Weinmann Philipp)

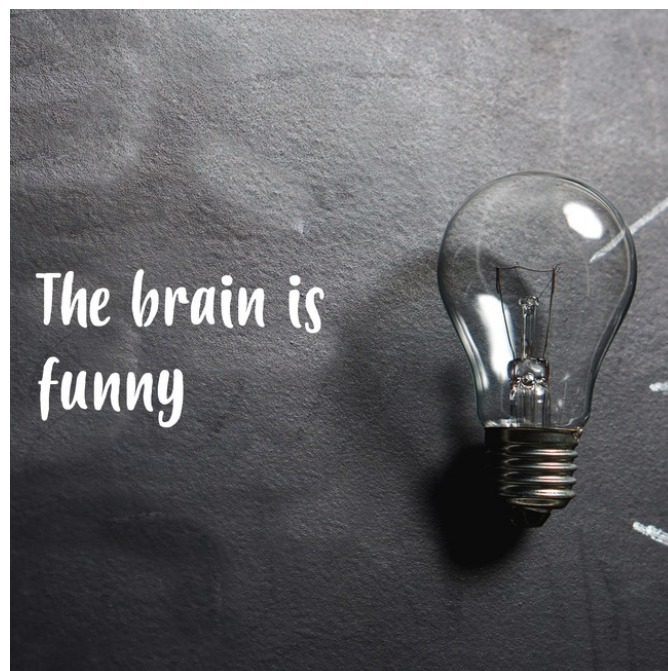


Abbildung 1: Quote generated through an AI [int18]

Inhaltsverzeichnis

Abstract	ix
1 Was ist Maschinelles Lernen	x
1.1 Anwendungen im Bereich der Informatik	x
1.2 Ein erstes Beispiel	xi
2 Aufkommen von Maschinellern Lernen	xii
3 Künstliche Neuronale Netze	xiii
3.1 Einführung	xiii
3.2 propagation function	xv
3.3 sigmoidfunction	xv
3.3.1 Weights and Biases	xv
3.4 Fully Connected Layers	xv
3.5 Convolution	xv
3.6 Pooling	xv
3.7 Reccurence	xv
3.8 Training of a NN	xv
3.9 Backpropagation	xv
3.9.1 Gradient descent	xv
3.10 Limitationen von Neuronalen Netzen	xv
3.10.1 Computational Power	xv
3.10.2 Debugging	xv
3.10.3 Kontrollverlust(Skynet)	xv
4 Maschine Translation	xv
4.1 Verschiedene methoden der Maschine Translation	xv
5 Neuronale Maschinenübersetzung	xvi
5.1 In der Industrie	xvi
5.2 Kuriositäten	xvi
6 Bewertung	xvi
Literaturverzeichnis	3

Abstract

[TODO] Ein Abstract ist eine prägnante Inhaltsangabe, ein Abriss ohne Interpretation und Wertung einer wissenschaftlichen Arbeit.

1 Was ist Maschinelles Lernen

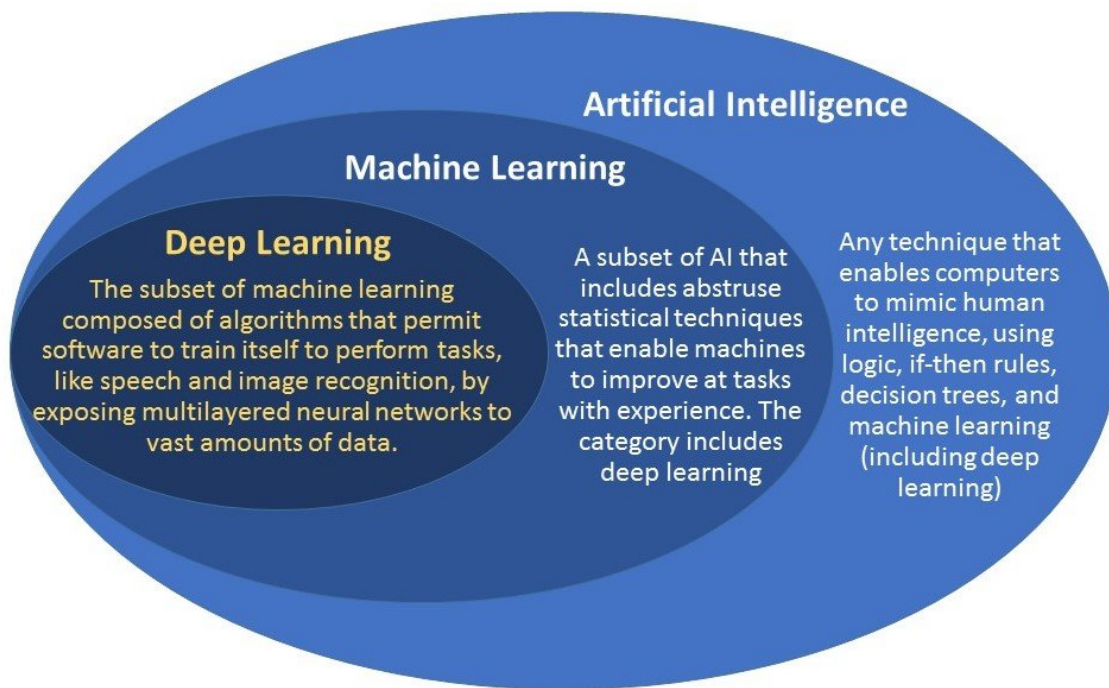


Abbildung 1: Veranschaulichung, wie Maschinelles Lernen einzuordnen ist. [Dha17]

Maschinelles Lernen ist ein Teilbereich der Künstlichen Intelligenz. Es handelt sich also um eine Methode, die es Maschinen ermöglicht auf ihre Umwelt zu reagieren. Es ist jedoch oft schwierig wenn nicht unmöglich von Hand zu erkennen welche Reaktion das beste Ergebniss liefert, hier kommt Maschinelles Lernen ins Spiel. Dank statistischer Auswertungen können Algorithmen entstehen, die (meistens) korrekte Ergebnisse liefern, ohne das der Programmierer sich Gedanken machen muss, wie der Algorithmus letztendlich aufgebaut ist. Während diese Methode der Datenauswertung schon lange bekannt ist

1.1 Anwendungen im Bereich der Informatik

In jedem Gebiet in dem große Mengen an Daten zur Verfügung stehen bzw. generiert werden können, ist Maschine Learning theoretisch anwendbar. Weil dies auf so ziemlich jeden Bereich der Informatik zutrifft, wird so viel Hoffnung in diese Art von Algorithmen gesteckt.

Obwohl die Theorie hinter dieser Art der Datenauswertung seit langem bekannt ist, so finden mächtigere Algorithmen die meist auf Neuronalen Netzen basieren erst seit kurzem verbreitete Anwendung dank verbesserter Rechenleistung. [CITATION NEEDED?]

Anwendungsbereiche sind zum Beispiel:

- Gesichtserkennung
- Spamerkennung (Email)
- Spracherkennung
- Handschrifterkennung
- automatische Medikamentenentwicklung
- Maschinelle Übersetzungen

In dieser Ausarbeitung werden wir uns insbesondere für Maschinelle Übersetzungen interessieren.

1.2 Ein erstes Beispiel

Hören sie sich dieses Beispiel an: audiofile

Sie erkennen sofort, das es sich hier um spielende kinder handelt. Unser Gehirn schafft es mit extremer Genauigkeit Komplexe geräusche zu erkennen und zu analysieren. Auch wenn wir nicht erkennen was jedes einzelne Kind ruft, so wissen wir instinktiv das es sich um Kinder handelt. Schauen wir uns einmal die Wellenfunktion eines Abschnittes dieses Audiofile an:

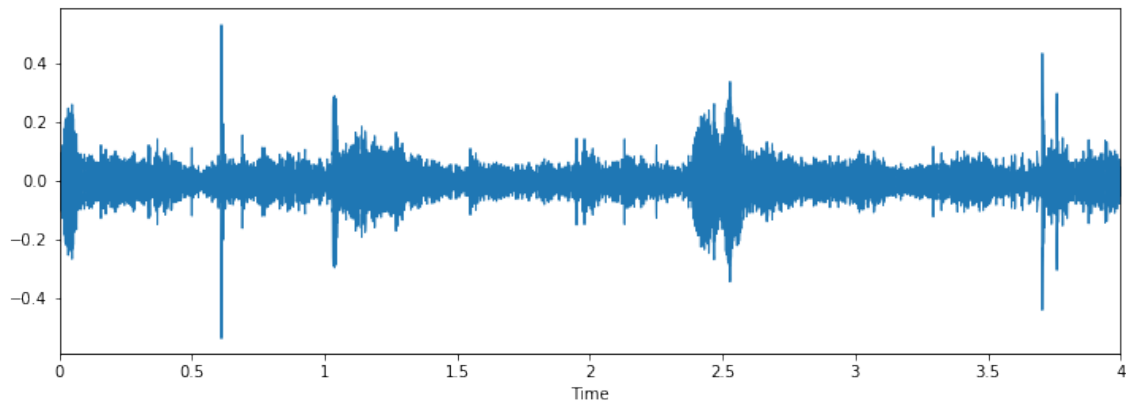


Abbildung 2: Audiofile Spielende Kinder [Shaa]

Vergleichen wir dieses mit der Wellenfunktion eines PressluftHammers:

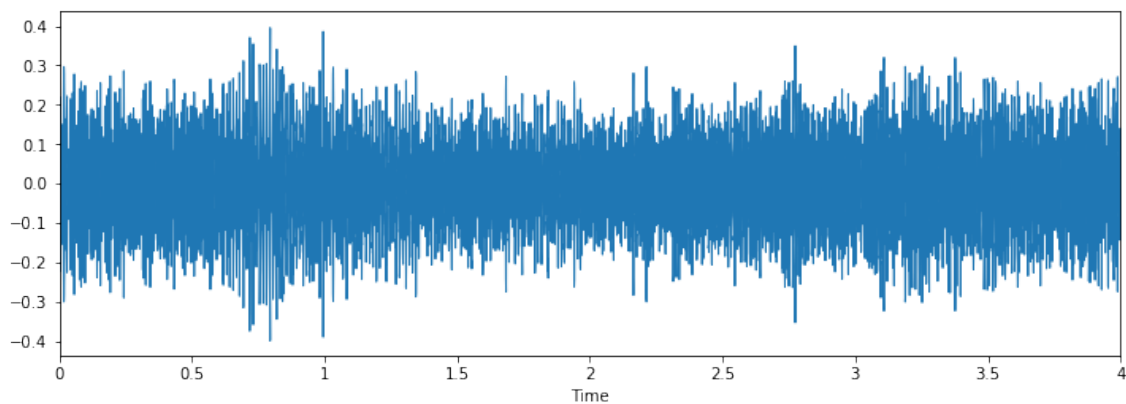


Abbildung 3: Audiofile eines pressluftHammers [Shab]

Wir bemerken einige klare Unterschiede. Falls wir nun ein Programm schreiben wollen, das erkennt welches erkennen kann ob ein audiofile eher spielende Kinder oder das Geräusch eines Presslufthammers enthält, so können wir Anhand der Wellenfunktion dieser Geräusche einige Ansätze herausfolgern. Wir bemerken zum Beispiel, das es bei kreischenden Kindern deutlich mehr ausreißer gibt als bei dem Geräusch eines Presslufthammers. Wir könnten also ein Programm schreiben, das den Durchschnitt aller Varianzen berechnet. Dies würde etwa so aussehen:

```
String classify(int[] audiofile) {
    int varianz = getVarianz(audiofile);
    if (varianz > (spielkinderDurchschnittsVarianz
        + pressluftHammerDurchschnittsVarianz) / 2)
    ){
        return "SpielendeKinder";
    } else {
        return "PressluftHammer";
    }
}
```

Man bemerkt jedoch, das man hierfür schon wissen muss, welchen Wert die Varianzen von spielenden Kindern bzw von Presslufthammern im Durchschnitt annehmen. Dazu braucht man einen großen Datensatz an AudioDaten, bei denen man weiß um welche Geräusche es sich handelt. Falls diese Vorhanden sind, kann man folgendes Programm schreiben, welches den Durchschnittswert der Varianz für die jeweilige Kategorie ermittelt:

```
int mittelwertVarianz = (spielkinderDurchschnittVarianz +
    pressluftHammerDurchschnittVarianz) / 2);
String classify(int[] audiofile, String category) {
    int varianz = getVarianz(audiofile);
    if (varianz > mittelwertVarianz){
        updateVarianzen(varianz, category);

        return "SpielendeKinder";
    } else {
        updateVarianzen(varianz, category);
        return "PressluftHammer";
    }
}

void updateVarianzen(int varianz, String category) {
    if (category.equals(spielendeKinder)) {
        spielendeKindervarianzen.add(varianz);
    } else {
        pressluftHammerarianzen.add(varianz);
    }
}
```

Man bemerkt, das dieses Programm noch immer ermittelt, welcher Kategorie die Audiofiles angehören. Daher kann es, falls bekannt ist um welches der beiden Kategorien es sich handelt weiterverwendet werden. Die Durschnittsvarianzen werden damit mit jedem neuen Audiofile präziser, das programm lernt mit der Zeit. Dies ist ein Beispiel sehr rudimentären Maschinellern.

Während dieses Programm sehr einfach ist, so gibt es weitaus mächtigere programme, in denen der Algorithmus nicht nur den Wert einer Variablen erkennt, sondern auch de Kriterien zur Unterscheidung zwischen Kategorien oder sogar Kategorien selber ermittelt.

2 Aufkommen von Maschinellern Lernen

Maschinelles Lernen generell und insbesondere Neuronale Netze erfreuen sich seit einigen Jahren (stand 2018) großer Aufmerksamkeit. Der Grundstein für diese Verfahren wurde jedoch schon Ende des 18. Jahrhunderts von Thomas Bayes gelegt[BPC63].

Während die ersten Anwendungen zum Größteil (meißt abstrakter) mathematischer natur

waren[Leg05], so befasst sich schon 1913 das erste Paper mit der Analyse von Gedichten[Mar06]. Dort analysiert Markov ein Gedicht und bemerkt, dass man Wahrscheinlichkeiten formulieren kann, welche Eigenschaften weitere Teile vom Text haben ohne dass man die Gesamtheit des Gedichtes in betracht ziehen muss.

Ab der Zweiten Hälfte des 20. Jahrhunderts wurden Entdeckungen im Bereich des Maschinellen Lernens gemacht, die heute jedem der mit dem Fach zu tun hat bekannt sind. 1950 formuliert Alan Turing die Turing Learning Machine [Mac50], ein Jahr später entwickeln und bauen zwei Wissenschaftler das erste Neuronale Netz[MM51]. 1957 entwickelt Frank Rosenblatt den "perceptron"[Ros58], ein erstes Modell eines fully connected neural network.

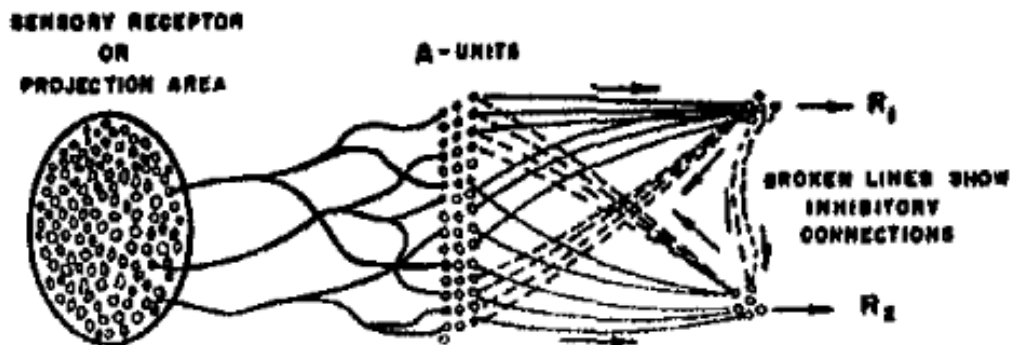


FIG. 2A. Schematic representation of connections in a simple perceptron.

Abbildung 4: original draft of a perceptron in the original paper

Nachdem [CONTINUE HERE]

3 Künstliche Neuronale Netze

3.1 Einführung

[UNINTERESSANT? NIMMT VIEL ZU VIEL PLATZ EIN; JEDER DER DAS LIEST WEIß WIE EIN NN IM PRINZIP AUFGEBAUT IST] Ein technischer Durchbruch ist oft "nur" die gelungene Nachahmung eines in der Natur vorkommenden Phänomens. Neuronale Netze kann man Vergleichen mit dem Versuch das menschliche Gehirn nachzubauen.[CITATION?] Während diese Unterfangen nicht oder nur teilweise gelungen sind, haben sich einige Nebenprodukte dieser Forschung als sehr nützlich erwiesen. Ein Neuronales Netz besteht aus Knoten die "künstliche Neuronen" genannt. Diese können auf verschiedene Weise miteinander verbunden sein. Diese Verbindungen werden oft Synapsen genannt. Jedes Neuron kann Signale empfangen und weiterleiten. Der Graph der gebildet wird nennt man die Topologie des Neuronalen Netzes.

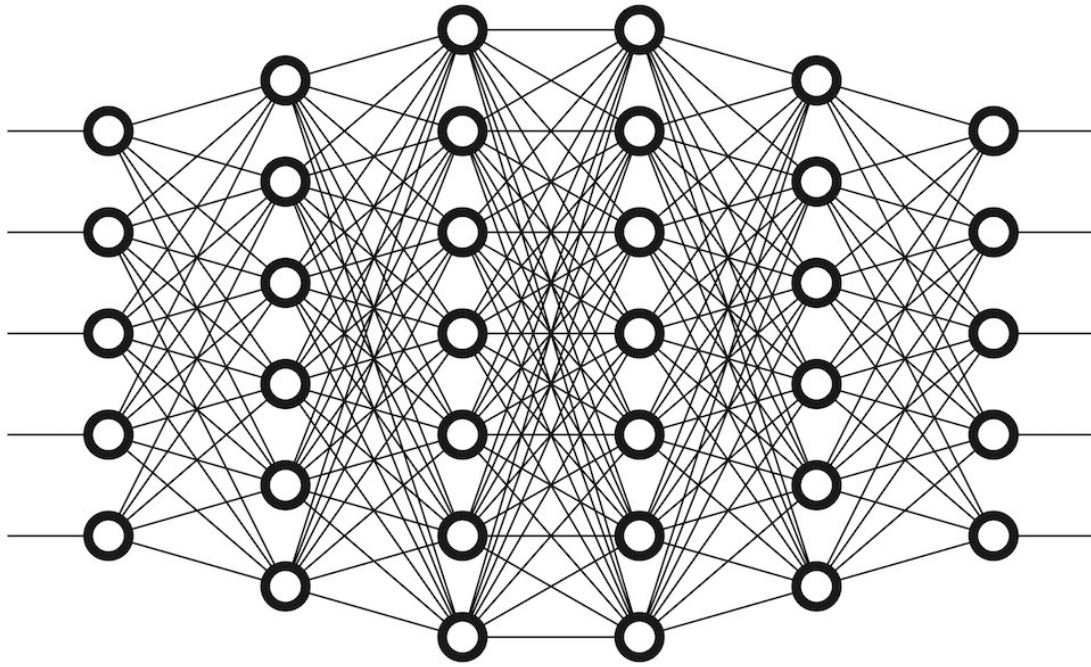


Abbildung 5: Topologie eines Fully Connected Feed Forward Neuronalen Netzes. [Neu]

Neuronen werden meist in Schichten zusammengefasst, die Stufen des Neuronalen Netzes darstellt, die mithilfe der Breitensuche bestimmt werden können.

[AB HIER WIEDER?] Wie in unserem Beispiel erwähnt, ist das Ziel von Neuronalen Netzen Eigenschaften zu erkennen, die gewisse Objekte gemeinsam haben. Neuronale Netze sollen:

- Klassifizierung
- Zusammenhänge erkennen

3.2 propagation function

3.3 sigmoidfunction

3.3.1 Weights and Biases

3.4 Fully Connected Layers

3.5 Convolution

3.6 Pooling

3.7 Reccurence

3.8 Training of a NN

3.9 Backpropagation

3.9.1 Gradient descent

3.10 Limitationen von Neuronalen Netzen

3.10.1 Computational Power

3.10.2 Debugging

3.10.3 Kontrollverlust(Skynet)

Diese Eigenschaften machen Maschine Learning zu einem Vielversprechenden Werkzeug für maschinelle Übersetzungen.

4 Maschine Translation

Die Anwendung von Software um Text von einer Sprache zur anderen zu übersetzen.

4.1 Verschiedene methoden der Maschine Translation

- Regelbasiert
- Statistische Übersetzung
- Neuronale Maschinenübersetzung

5 Neuronale Maschinenübersetzung

- Recurrent Neural Networks [Warum gerade diese?] [Elman Network, Jordan Netzwerk, Hopfield Network,...]
-

5.1 In der Industrie

- Google
- Microsoft
- Yahoo

5.2 Kuriositäten

- Facebook chatbots haben eine eigene Sprache erfinden um zu kommunizieren.
- Google Translate hat eine Sprache erfunden die als Zwischensprache dient.

Fazit: Es ist schwer bzw unmöglich die Funktionsweise von Programmen die anhand von NN netzen entstanden sind zu verstehen oder zu kontrollieren.

6 Bewertung

Test der Arbeit IA [Goo17]

Test02 [int18]

Literaturverzeichnis

- [BPC63] T. Bayes, R. Price, and J. Canton, “An essay towards solving a problem in the doctrine of chances,” 1763.
- [Dha17] M. Dhande, 2017, aufgerufen: 03/12/18. [Online]. Available: <https://www.geospatialworld.net/blogs/difference-between-ai%E2%80%9Fmachine-learning-and-deep-learning/>
- [Goo17] Google Inc., “Google assistant,” 2017, aufgerufen: 09.01.2017. [Online]. Available: <https://assistant.google.com/>
- [int18] A. intelligence, 2018, generated on 27.11.18. [Online]. Available: <http://inspirobot.me/>
- [Leg05] A. M. Legendre, *Nouvelles méthodes pour la détermination des orbites des comètes*. F. Didot, 1805.
- [Mac50] C. Machinery, “Computing machinery and intelligence-am turing,” *Mind*, vol. 59, no. 236, p. 433, 1950.
- [Mar06] A. A. Markov, “An example of statistical investigation of the text eugene onegin concerning the connection of samples in chains,” *Science in Context*, vol. 19, no. 4, pp. 591–600, 2006.
- [MM51] D. E. Marvin Minsky, 2051, haven’t been able to locate their original thesis, Website opened 09.12.18. [Online]. Available: <http://cyberneticzoo.com/mazesolvers/1951-maze-solver-minsky-edmonds-american/>
- [Neu] “Neuronales netz bild,” aufgerufen: 09.01.2017. [Online]. Available: <https://machine-learning-blog.de/2017/11/02/was-ist-deep-learning/>
- [Ros58] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain.” *Psychological review*, vol. 65, no. 6, p. 386, 1958.
- [Shaa] F. Shaikh, “audiofile kids playing,” aufgerufen: 09.01.2017. [Online]. Available: <https://s3-ap-south-1.amazonaws.com/av-blog-media/wp-content/uploads/2017/08/23193324/02.png>
- [Shab] —, “audiofile presslufthammer,” aufgerufen: 09.01.2017. [Online]. Available: <https://s3-ap-south-1.amazonaws.com/av-blog-media/wp-content/uploads/2017/08/23194100/jack.png>

