

Maschinelles Lernen im Kontext der Programmierung natürlicher Sprachen

Seminararbeit
von

Weinmann Philipp

An der Fakultät für Informatik
Institut für Programmstrukturen
und Datenorganisation (IPD)

Betreuer: Dipl. Inform. Alexander Wachtel

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Die Regeln zur Sicherung guter wissenschaftlicher Praxis im Karlsruher Institut für Technologie (KIT) habe ich befolgt.

Karlsruhe, [FILL OUT DATE HERE]

.....
(Weinmann Philipp)

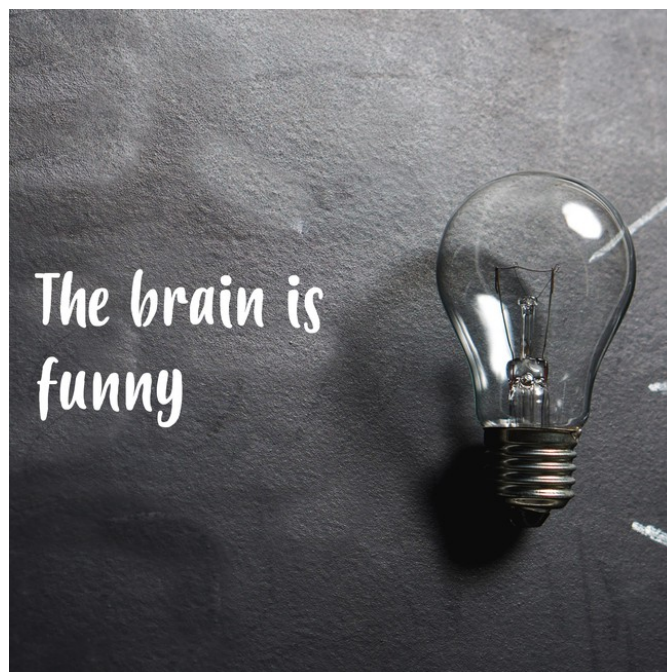


Abbildung 1: Quote die von einer KI generiert worden ist. *Die Resultate sind noch nicht zufriedenstellend* [int18]

Inhaltsverzeichnis

Abstract	ix
1 Einleitung	x
1.1 Anwendungen im Bereich der Informatik	x
1.2 Maschinelles Lernen: Ein erstes Beispiel	xi
1.3 Aufkommen von Maschinellern Lernen	xii
2 Künstliche Neuronale Netze (Artificial Neural Network)	xiii
2.1 Aktivierungsfunktion	xiv
2.2 Propagation function	xv
2.3 Training eines Artificial Neural Network	xvi
2.3.1 Backpropagation	xvi
2.3.2 Gradientenverfahren	xvii
2.4 Limitationen und Gefahren von Neuronalen Netzen	xvii
3 Maschinelle Übersetzungen (machine translation)	xvii
3.1 Neuronale Maschinenübersetzung (Neural machine translation) . . .	xviii
3.2 Kuriositäten	xviii
4 Bewertung	xix
Literaturverzeichnis	1
Glossar	3

Abstract

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

1 Einleitung

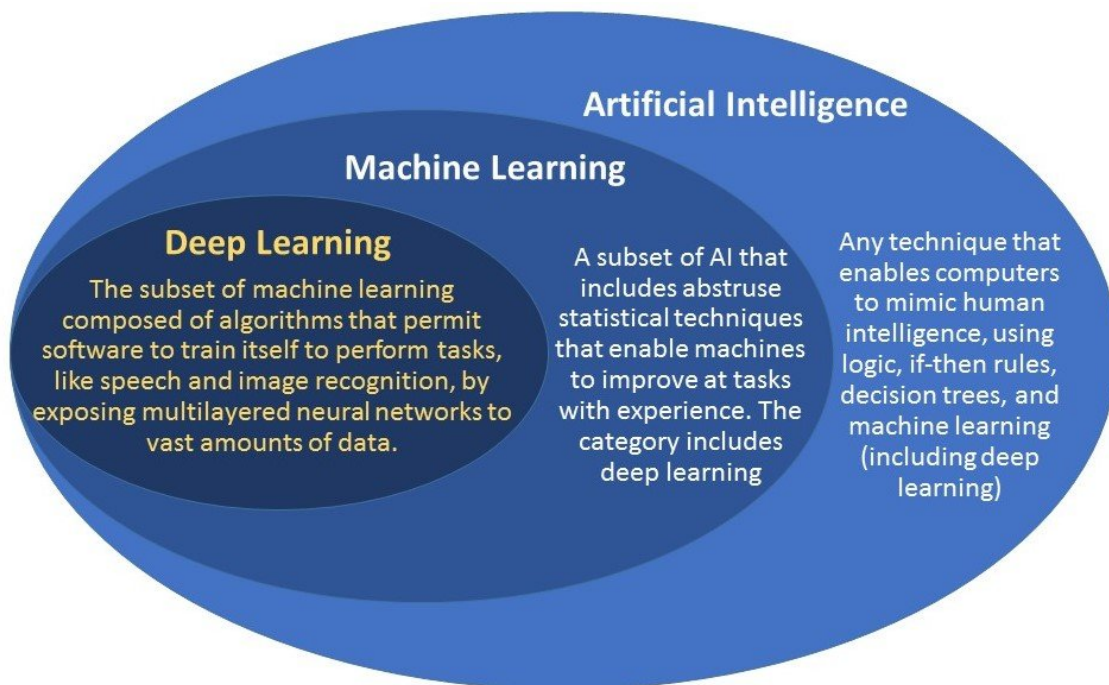


Abbildung 1: Veranschaulichung, wie Maschinelles Lernen einzuordnen ist. [Dha17]

Maschinelles Lernen (ML) ist ein Teilbereich der Künstlichen Intelligenz (KI). Es handelt sich also um eine Methode, die es Maschinen ermöglicht auf ihre Umwelt zu reagieren. Weil es oft schwierig wenn nicht unmöglich ist von Hand zu erkennen welche Reaktion das beste Ergebnis liefert versucht man dies Maschinell zu lösen. Dank statistischer Auswertungen können Algorithmen entstehen, die mit einer gewissen Wahrscheinlichkeit korrekte Ergebnisse liefern, ohne das der Programmierer sich Gedanken machen muss, wie der Algorithmus letztendlich aufgebaut ist. Diese Methode wird ML genannt.

1.1 Anwendungen im Bereich der Informatik

In jedem Gebiet in dem große Mengen an Daten zur Verfügung stehen bzw. generiert werden können, ist ML theoretisch anwendbar. Weil dies auf so ziemlich jeden Bereich der Informatik zutrifft, erhofft man sich große Fortschritte von dieser neuen Technologie [Kou18].

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. [Hwa18]

Anwendungsbereiche sind zum Beispiel:

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

- Maschinelle Übersetzungen

In dieser Ausarbeitung werden wir uns insbesondere für Künstliche Neuronale Netze (Artificial Neural Network) sowie deren Anwendung für Maschinelle Übersetzungen interessieren.

1.2 Maschinelles Lernen: 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 Audiofiles an:

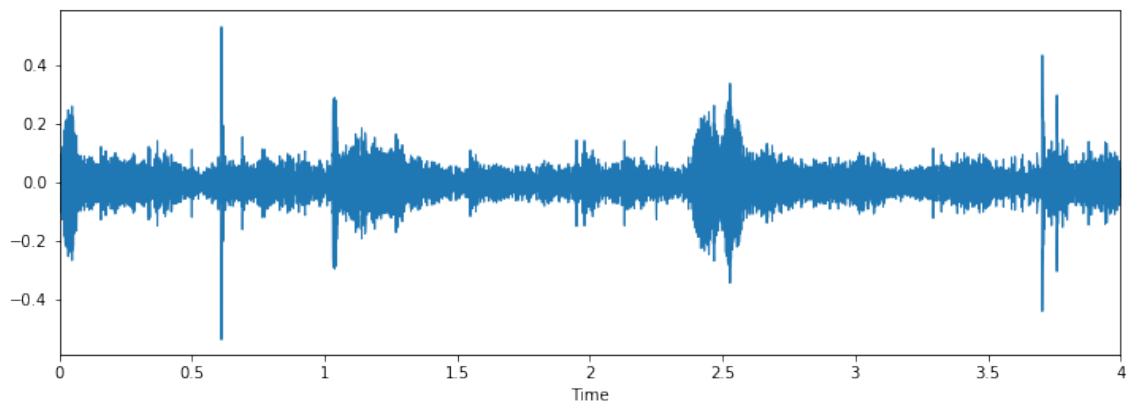


Abbildung 2: Audiofile spielender Kinder. Luftdruck/Umgebungsdruck in Funktion der Zeit (Sekunden). [SF17]

Vergleichen wir dieses mit der Wellenfunktion eines PressluftHammers:

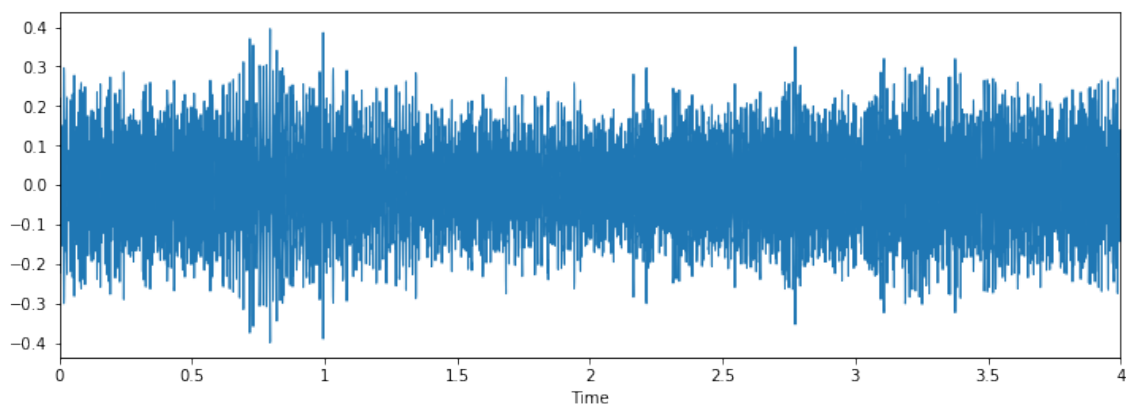


Abbildung 3: Audiofile eines Presslufthammers. Luftdruck/Umgebungsdruck in Funktion der Zeit (Sekunden). [SF17]

Wir bemerken einige klare Unterschiede. Falls wir nun ein Programm schreiben wollen, das erkennt ob ein Audiofile spielende Kinder oder das Geräusch eines Presslufthammers enthält, so können wir Anhand der Wellenfunktion dieser Geräusche einige Ansätze folgern. 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 ein Audiofile nach der Anzahl an Ausreißern pro Zeit(**APS**) erkennt, d.h. klassifiziert.

```
String classify(int[] audiofile) {
    int APS = getAPS(audiofile);    // AusreisserProSekunde
    if (APS > (spielkinderDurchschnittsAPS +
        pressluftHammerDurchschnittsAPS) / 2)){
        return "SpielendeKinder";
    } else {
        return "PressluftHammer";
    }
}
```

Man bemerkt jedoch, das man hierfür schon wissen muss, welchen Wert die **APS** von spielenden Kindern bzw. von Presslufthämmern 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 **APS** für die jeweilige Kategorie ermittelt:

```
int mittelwertAPS = (spielkinderDurchschnittsAPS +
    pressluftHammerDurchschnittsAPS) / 2);
String classify(int[] audiofile, String category) {
    int APS = getAPS(audiofile);
    if (APS > mittelwertAPS){
        updateAPS(APS, category);

        return "SpielendeKinder";
    } else {
        updateAPS(APS, category);
        return "PressluftHammer";
    }
}

void updateAPS(int APS, String category) {
    if (category.equals(spielendeKinder)) {
        spielendeKinderAPS.add(APS);
    } else {
        pressluftHammerAPS.add(APS);
    }
}
```

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 Ausreißer pro Sekunde (**APS**) werden damit mit jedem neuen Audiofile präziser, das Programm Lernt mit der Zeit. Dies ist ein Beispiel sehr rudimentärem ML.

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 die Kriterien zur Unterscheidung zwischen Kategorien oder sogar Kategorien selber ermittelt.

1.3 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 Großteil abstrakter mathematischer Natur waren[Leg05], so befasst sich schon 1913 die erste These mit der Analyse von Gedichten[Mar06]. Dort analysiert Markov ein Gedicht und bemerkt, das man Wahrscheinlichkeiten formulieren

kann, welche Eigenschaften weitere Teile vom Text haben ohne das man die Gesamtheit des Gedichtes in Betracht ziehen muss.

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 vollständig verbundenem neuronalen Netzes (Fully connected Neural Network).

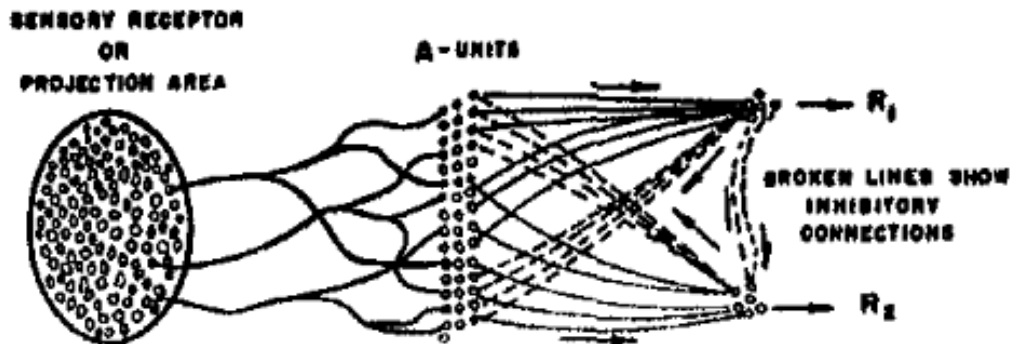


FIG. 2A. Schematic representation of connections in a simple perceptron.

Abbildung 4: Zeichnung eines Perceptrons in der ursprünglichen These. [Ros58]

1970 wird eine erste Form der Fehlerrückführung (*Backpropagation*) entwickelt [Lin70], 1982 wird ein erstes Rekurrentes Neuronales Netz (Recurrent Neural Network) entwickelt, welches es ermöglicht Sequenzen von Daten zu verarbeiten, die voneinander abhängen. Anfang des 21. Jahrhunderts und insbesondere ab 2010 hat die Rechnerleistung so sehr zugenommen, das sehr rechenintensive Anwendungen von Maschinellem Lernen möglich werden. Die ersten kommerziellen Bilderkennungssoftware kommen auf den Markt [TYRW14]. 2016 stellte Google seine Neuronale Maschinelle Übersetzungssoftware vor. Microsoft zieht noch im selben Jahr nach. [Mic]

Wir werden nun Künstliche Neuronale Netze (Artificial Neural Network) kurz einführen.

2 Künstliche Neuronale Netze (Artificial Neural Network)

Ein technischer Durchbruch ist oft *nur* die gelungene Nachahmung eines in der Natur vorkommenden Phänomens. Artificial Neural Network werden oft beschrieben als Versuch das menschliche Gehirn nachzubauen. Während diese Unterfangen nicht oder nur teilweise gelungen sind [Ada18], haben sich einige Nebenprodukte dieser Forschung als sehr nützlich erwiesen. Eines dieser Nebenprodukte ist die Entwicklung Artificial Neural Network. Es handelt sich dabei um ein Programmiergerüst (*Framework*) für Algorithmen die auf ML basieren. Dieses erlaubt Eingabesignale, die verarbeitet werden zu mehreren Ausgabesignalen. Beschrieben werden Artificial Neural Network mit Begriffen aus der Biologie. Die erste Lage (*inputlayer*), besteht aus einer gewissen Anzahl an Knoten, sogenannten *Neuronen*, die bei unterschiedlichen Eingaben unterschiedlich aktiviert werden. Diese sind mit weiteren Neuronen über sogenannte *Synapsen* verbunden und versenden je nach Artificial Neural Network mehr oder weniger starke Signale über diese Verbindungen.

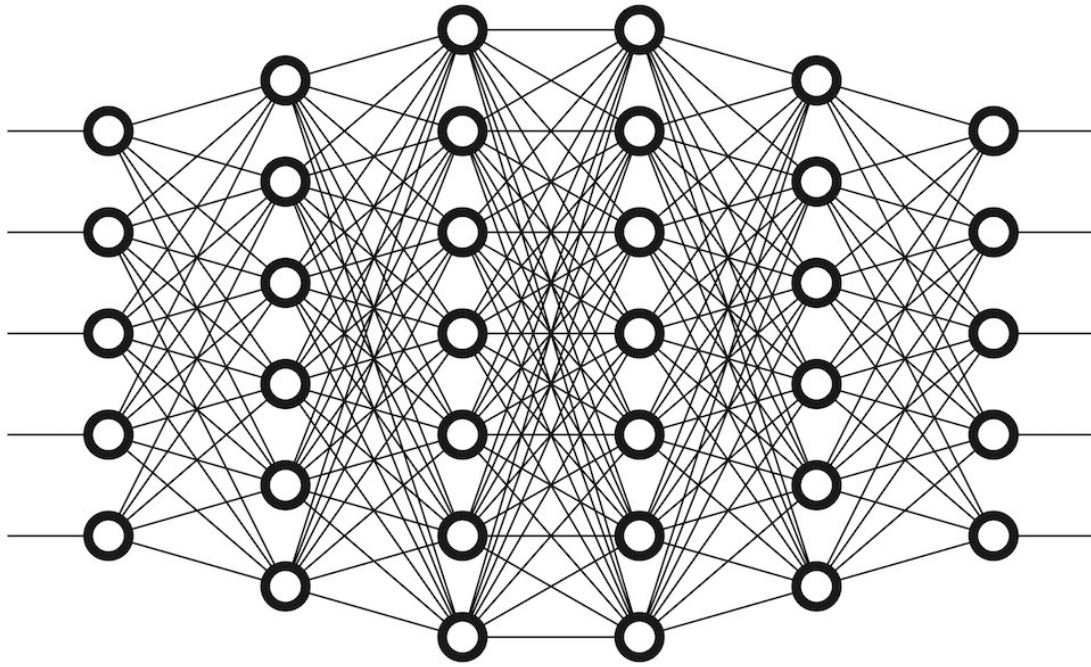


Abbildung 5: Topologie eines Fully connected Neural Network. [Neu]

Neuronen können in Lagen (*Layers*) zusammengefasst werden, welche die Stufen des Artificial Neural Network darstellen, die mithilfe der Breitensuche bestimmt werden können.

Wie in unserem Beispiel erwähnt, ist das Ziel von neuronalen Netzen Eigenschaften zu erkennen, die gewisse Objekte gemeinsam haben. Artificial Neural Network sollen:

- Klassifizieren
- Zusammenhänge erkennen

2.1 Aktivierungsfunktion

Durch Signale können Neuronen im Artificial Neural Network mehr oder weniger stark aktiviert werden. Das Ausgabesignal eines aktivierten Neurons wird durch die Aktivierungsfunktion berechnet. Diese bildet den Aktivierungswert auf ein Intervall ab und ist daher eine nichtlineare Funktion. Ohne diese Funktion könnten komplexe nichtlineare Datensätze wie Bilder, Videos sowie Audio nicht analysiert werden.

"Neural-Networks are considered Universal Function Approximators. It means that they can compute and learn any function at all. Almost any process we can think of can be represented as a functional computation in Neural Networks." [Wal17]

[KANN ICH DAS SO STEHEN LASSEN ODER MUSS ICH ES ÜBERSETZEN? FINDE DAS DAS ZITAT EIGENTLICH SEHR SCHÖN BESCHREIBT WELCHE FUNKTION DIE AKTIVIERUNGSFUNKTION BESITZT]

Eine früher beliebte Aktivierungsfunktion war die normalisierte Sigmoid Funktion.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Während dies die wohl allgemein bekannteste Aktivierungsfunktion ist, verwenden moderne Artificial Neural Network heute eine Variante der *Rectifier Funktion* [LBH15]:

$$f(x) = \max(0, x)$$

Die eine beliebte Approximierung dieser Funktion ist z.B die analytische Funktion:

$$g(x) = \log(1 + e^x)$$

Die *Rectifier Funktionen* besitzt einige entscheidende Vorteile, weshalb sie heute so weit verbreitet ist. sie sind leicht ableitbar, sind einfach zu berechnen (*d.h. sie benötigen wenig Rechenleistung*) und besitzen einige weitere Eigenschaften die sehr hilfreich sind beim trainieren von Artificial Neural Network.

2.2 Propagation function

Jede Verbindung besitzt eine Gewichtung, *weight*. Diese beschreibt wie viel Einfluss sie hat auf die Aktivierung des verbundenen Neurons.

Die *propagation function* berechnet den Input $p_j(t)$ des Neurons j anhand der Outputs $o_i(t)$ der vorangehenden Neuronen.

$$p_j(t) = \sum_i o_i(t) w_{ij}$$

Um eine Aktivierungsschwelle (*threshold*) einzuführen kann ein sogenannter *bias* der Summe hinzugefügt werden. Dies ist nötig wenn ein Neuron nur ab einem bestimmten Aktivierungswert von Bedeutung ist.

$$p_j(t) = \sum_i o_i(t) w_{ij} + bias$$

Ein großer Vorteil von Artificial Neural Network ist, das sie als eine Folge von Matrixmultiplikationen darstellbar sind, welche besonders effizient durch Computerprozessoren berechnet werden können.

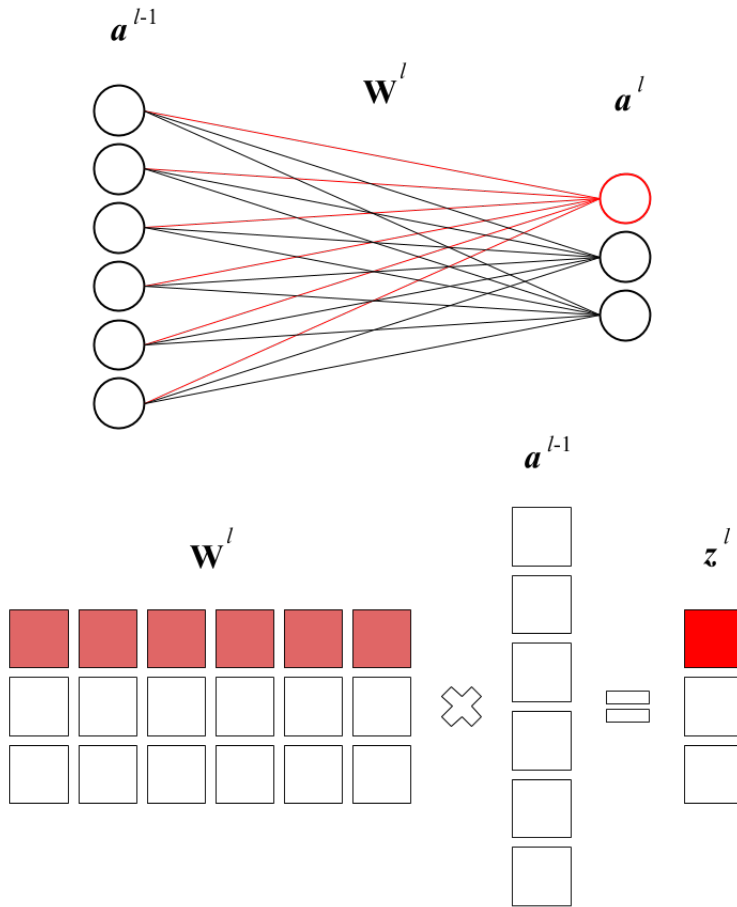


Abbildung 6: Neuronales Netz in Matrixrepräsentation. Hierbei ist $a^l = f(z^l)$ wobei f die Aktivierungsfunktion darstellt. [Hal16]

2.3 Training eines Artificial Neural Network

Wie zuvor beschrieben, benötigt ein Artificial Neural Network massive Datensätze. Gesucht wird nun ein Algorithmus der anhand dieser Daten die *weights* und *biases* so einstellt, das das Netzwerk seine Funktion erfüllt. Einer dieser Algorithmen ist die Fehlerrückführung (*Backpropagation*).

2.3.1 Backpropagation

Der Backpropagation Algorithmus verläuft folgendermaßen: Erst wird ein Eingabemuster durch das Artificial Neural Network propagiert. Daraufhin wird die Ausgabe des Netzes, d.h. die Aktivierung des letzten Layers verglichen mit der gewünschten Aktivierung. Der Fehler des Netzes kann durch die quadrierte Fehlerfunktion berechnet werden.

$$E_j = \frac{(t_j - y_j)^2}{2}$$

t_j ist der gewünschte Output des Neurons j

y_j ist der wirkliche Output des Neurons j

Das Ziel ist es nun die Summe der Fehlerfunktionen über alle möglichen Inputs zu minimieren. Dies ist nicht möglich ohne alle möglichen Kombinationen von *weights* und *biases* auszuprobieren, jedoch kann man dank des Gradientenverfahrens lokalen Minimas nahe kommen.

2.3.2 Gradientenverfahren

2.4 Limitationen und Gefahren von Neuronalen Netzen

Während Artificial Neural Network im Moment großer Beliebtheit genießen, so sollte man sich einiger ihrer Nachteile bewusst sein. Wie zuvor erwähnt wird durch das Gradientenverfahren nur lokale Minima gefunden, das die Performance eines Artificial Neural Network nach dem Trainieren ist daher stark vom Anfangszustand abhängig und kann unter Umständen auf enttäuschendem Niveau trotz neuer Daten stagnieren. Das trainieren von Artificial Neural Network ist noch immer sehr Rechnerleistungsintensiv und benötigt große Mengen an Daten (labeled data).

Durch diese neuen Technologien sind Daten sehr wertvoll geworden. Sie sind der Grund weshalb Suchmaschinen und Soziale Netzwerke zu den Wertvollsten unternehmen Weltweit gehören. Während ML zwar nicht verantwortlich ist für den unvorsichtigen Umgang und Handel mit unseren Daten, so verstärken sie diesen Trend. entstehende Kontrollverlust der durch das nutzen solcher Werkzeuge entsteht. Es wird oft angenommen, das es nicht möglich ist ein Artificial Neural Network zu verstehen, das es sich um eine *black box* handelt. Diese Aussage sollte relativiert werden, es ist in den Letzten Jahren viel geforscht worden um dieses Problem zu überwinden[RSG16]. Nichtsdestotrotz ist es noch immer schwierig das Innere eines Artificial Neural Network oder ähnlichem Konstrukt zu verstehen, es muss immer damit gerechnet werden, das ein Neuronales Netz auf völlig unvorhergesehen Weise auf eine Neue Eingabe reagiert. Es können kaum Garantien für das Verhalten eines neuronalen Netzes gegeben werden. Es ist daher mehr als problematisch diese Algorithmen für Zwecke zu nutzen in denen über Menschenleben entschieden wird, wie es das Pentagon mit Google's Framework Tensorflow getan hat [Gib18]. Anwendungen von Artificial Neural Network werfen ethische Fragen auf und es sollte immer Abgewogen werden, in welchen Fällen man auf Artificial Neural Network lieber verzichten sollte.

Trotz dieser Bedenken hat Maschinelles Lernen schon große Fortschritte in vielen Bereichen der Informatik ermöglicht. Im Bereich der Programmierung in Natürlicher Sprache sind diese schon längst beim Nutzer angekommen in der Form von Chatbots oder persönlichen Assistenten wie Siri im Jahre 2010.

3 Maschinelle Übersetzungen (machine translation)

Während Chatbots noch weit entfernt sind den Turingtest zu bestehen, so haben Maschinelle Übersetzungen dank Artificial Neural Network von und ins Englische quasi Menschliche Fehlerraten erreicht[goo16].

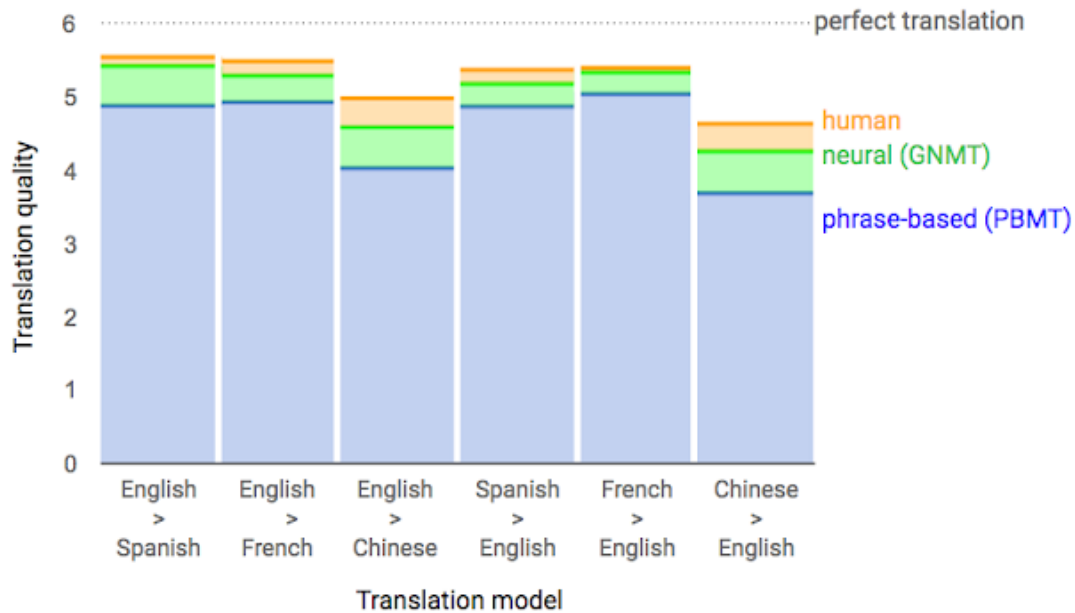


Abbildung 7: Google Translate Performance. Muttersprachler wurden gebeten Übersetzungen von Google Translate auf einer Skala von 0 bis 6 zu bewerten. [goo16]

3.1 Neuronale Maschinenübersetzung (Neural machine translation)

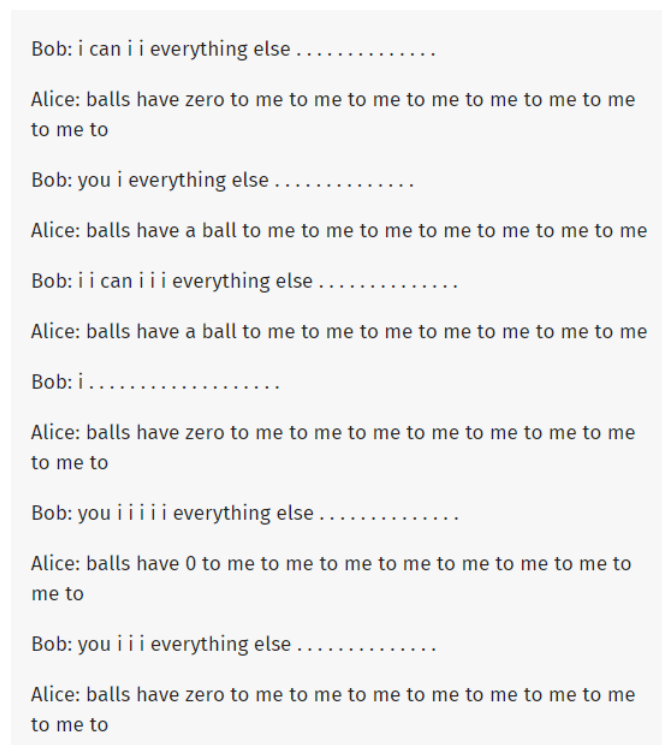
Neural machine translation hat signifikative Verbesserungen in machine translation ermöglicht. Dank Artificial Neural Network behauptet Google z.B. ihre Fehlerraten um 60% gesenkt zu haben [WSC⁺16]. Andere Dienste wie Microsoft sowie Yahoo melden ähnliche Fortschritte [Mic18].

3.2 Kuriositäten

Algorithmen die auf ML und insbesondere auf Artificial Neural Network basieren ist oft unvorhersehbar und können kuriose Lösungswege nutzen.

- Facebook Chatbots haben eine eigene Sprache erfunden um zu kommunizieren [Wil18].
- Google Translate hat eine Sprache erfunden die als Zwischensprache dient [MS16].

Facebook hat zwei Chatbots verschiedene virtuelle Objekte gegeben, welche einen bestimmten Wert besitzen, die diese miteinander austauschen konnten. Die Verhandlungstaktiken die die beiden Chatbots nutzen sollten durch ML erlernt werden.



Bob: i can i i everything else

Alice: balls have zero to me to me to me to me to me to me to me to me to me to

Bob: you i everything else

Alice: balls have a ball to me to me to me to me to me to me to me to me to me

Bob: i i can i i i everything else

Alice: balls have a ball to me to me to me to me to me to me to me to me to me

Bob: i

Alice: balls have zero to me to me to me to me to me to me to me to me to me to me to

Bob: you i i i i everything else

Alice: balls have 0 to me to me to me to me to me to me to me to me to me to me to

Bob: you i i i everything else

Alice: balls have zero to me to me to me to me to me to me to me to me to me to me to

Abbildung 8: Facebook Chatbots entwickeln ihre eigene Sprache. [Wil18]

Das die Chatbots beschließen nicht mehr auf Englisch zu Kommunizieren hatte niemand vorausgesehen. Es zeigt auf wie unvorhergesehen sich ML Algorithmen verhalten können.

Bevor Google translate Artificial Neural Network verwendete, wurde jeder Satz erst ins Englische und dann in die Zielsprache übersetzt. Diese Methode limitiert die Anzahl an Übersetzungen die programmiert werden müssen drastisch, schlägt jedoch fehl sobald das Wort das übersetzt werden muss nicht im englischen vorkommt. Das Artificial Neural Network das von Google seit kurzem eingesetzt wird hat eine eigene Sprache entwickelt. Jeder Satz wird nun erst in die entwickelte Sprache übersetzt und dann in die Zielsprache. In dieser Sprache gibt es ein Wort für Baum. Dieses ist verlinkt mit jedem Wort in jeder Sprache, das Baum bedeutet. Man bemerkt das das neuronale Netz denselben Ansatz verwendet hat, den Menschen zuvor auch angewandt haben, jedoch ein Schritt weiter gegangen ist.

4 Bewertung

ML ist ein mächtiges Werkzeug in vielen Bereichen der Informatik. Insbesondere Artificial Neural Network und ähnliche Algorithmen haben die allgemeine Meinung das Menschen in gewissen Bereichen immer besser als Maschinen sein werden zumindest in Teilen widerlegt. Dadurch das diese Algorithmen Erfahrung sammeln und somit Lernen, stehen sie Menschen in Problemlösungen theoretisch in nichts nach.

Jedoch besitzen diese Werkzeuge dieselben Probleme die wir Menschen auch haben. Sie sind nur so gut wie die Daten die ihnen als Training zur Verfügung gestellt werden, genaue Lösungswege sind nur schwer zu durchschauen. In gewissen Bereichen, wie der machine translation sind diese Probleme nicht von Relevanz, Sprachen verändern sich ständig mit der Zeit, daher ist es wichtig, das Übersetzungstools genauso *mit ihrer Zeit gehen* d.h. sich dem Allgemeinen Sprachgebrauch anpassen. Maschinelles Lernen bietet Lösungen für sich dynamisch verändernde Probleme an. Man muss sich jedoch immer der Schwächen und Gefahren dieser Algorithmen bewusst sein sowie abwägen in welchen Fällen man auf sie verzichten sollte.

Literaturverzeichnis

- [Ada18] ADAN, Yariv: Do neural networks really work like neurons? In: *Medium* (2018), Oct. <https://medium.com/swlh/do-neural-networks-really-work-like-neurons-667859dbfb4f>
- [BPC63] BAYES, Thomas ; PRICE, Richard ; CANTON, John: An essay towards solving a problem in the doctrine of chances. (1763)
- [Dha17] Aufgerufen: 03/12/18
- [Gib18] GIBBS, Samuel: *Google's AI is being used by US military drone programme.* <https://www.theguardian.com/technology/2018/mar/07/google-ai-us-department-of-defense-military-drone-project-maven-tensorflow>. Version: Mar 2018
- [goo16] *A Neural Network for Machine Translation, at Production Scale.* <https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html>. Version: Sep 2016
- [Hal16] HALLSTROEM, Erik: *Backpropagation from the beginning – Erik Hallstroem – Medium.* <https://medium.com/@erikhallstrm/backpropagation-from-the-beginning-77356edf427d>. Version: Dec 2016
- [Hwa18] HWANG, Tim: Computational Power and the Social Impact of Artificial Intelligence. In: *CoRR* abs/1803.08971 (2018). <http://arxiv.org/abs/1803.08971>
- [int18] Generated on 27.11.18
- [Kou18] KOUR, Bhupinder: *The Rise of Machine Learning and AI is Improving Lives in 2018.* <https://www.smartdatacollective.com/rise-of-machine-learning-ai-improving-lives/>. Version: Jan 2018
- [LBH15] LECUN, Yann ; BENGIO, Yoshua ; HINTON, Geoffrey: *Deep learning.* <https://www.nature.com/articles/nature14539>. Version: May 2015
- [Leg05] LEGENDRE, Adrien M.: *Nouvelles méthodes pour la détermination des orbites des comètes.* F. Didot, 1805
- [Lin70] LINNAINMAA, Seppo: The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. In: *Master's Thesis (in Finnish), Univ. Helsinki* (1970), S. 6–7
- [Mac50] MACHINERY, Computing: Computing machinery and intelligence-AM Turing. In: *Mind* 59 (1950), Nr. 236, S. 433
- [Mar06] MARKOV, Andreï A.: An example of statistical investigation of the text Eugene Onegin concerning the connection of samples in chains. In: *Science in Context* 19 (2006), Nr. 4, S. 591–600

- [Mic] MICROSOFT: *Microsoft Translator launching Neural Network based translations for all its speech languages.* <https://blogs.msdn.microsoft.com/translation/2016/11/15/microsoft-translator-launching-neural-network-based-translations-for-all-its-sp>
- [Mic18] *Customized neural machine translation with Microsoft Translator.* <https://www.microsoft.com/en-us/research/blog/customized-neural-machine-translation-microsoft-translator/>.
Version: May 2018
- [MM51] Haven't been able to locate their original thesis, Website opened 09.12.18
- [MS16] MIKE SCHUSTER, Nikhil T. Melvin Johnson J. Melvin Johnson: *Zero-Shot Translation with Google's Multilingual Neural Machine Translation System.* <https://ai.googleblog.com/2016/11/zero-shot-translation-with-googles.html>. Version: Nov 2016
- [Neu] *Neuronal Netz Bild.* <https://machine-learning-blog.de/2017/11/02/was-ist-deep-learning/>. – Aufgerufen: 09.01.2017
- [Ros58] ROSENBLATT, Frank: The perceptron: a probabilistic model for information storage and organization in the brain. In: *Psychological review* 65 (1958), Nr. 6, S. 386
- [RSG16] RIBEIRO, Marco T. ; SINGH, Sameer ; GUESTRIN, Carlos: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016, S. 1135–1144
- [SF17] SHAIKH, Faizan ; FAIZAN: *Getting Started with Audio Data Analysis (Voice) using Deep Learning.* <https://www.analyticsvidhya.com/blog/2017/08/audio-voice-processing-deep-learning/>. Version: Aug 2017
- [TYRW14] TAIGMAN, Yaniv ; YANG, Ming ; RANZATO, Marc'Aurelio ; WOLF, Lior: Deepface: Closing the gap to human-level performance in face verification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, S. 1701–1708
- [Wal17] WALIA, Anish S.: *Activation functions and it's types- Which is better?* <https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f>.
Version: May 2017
- [Wil18] WILSON, Mark: *AI Is Inventing Languages Humans Can't Understand. Should We Stop It?* <https://www.fastcompany.com/90132632/ai-is-inventing-its-own-perfect-languages-should-we-let-it>.
Version: Jul 2018
- [WSC⁺16] WU, Yonghui ; SCHUSTER, Mike ; CHEN, Zhifeng ; LE, Quoc V. ; NOROUZI, Mohammad ; MACHEREY, Wolfgang ; KRIKUN, Maxim ; CAO, Yuan ; GAO, Qin ; MACHEREY, Klaus u. a.: Google's neural machine translation system: Bridging the gap between human and machine translation. In: *arXiv preprint arXiv:1609.08144* (2016)

Glossar

IA Intelligenter (persönlicher) Assistent.

KI Jegliches Programm das es einer Maschine ermöglicht auf ihre Umwelt zu reagieren..

ML Maschinelles Lernen künstliche generierung von Wissen aus Erfahrung..

testosterone itworks.