

Silas Hoffmann, inf103088

5. Fachsemester

6. Verwaltungssemester

15. März 2020

Thema 2

Seminar

im Sommersemester 2020

Dozent: Prof. Dr. Dennis Säring
Fachbereich Informatik

Fachhochschule Wedel

Inhaltsverzeichnis

I. Einführung	4
II. Geschichtliche Entwicklung	5
1. McCulloch-Pitts-Neuron	5
1.1. Funktionsweise	5
1.2. Nachteile bzw. Verbesserungspotenzial	8
2. Perceptron	9
3. Adeline	9
III. Aktuelle Entwicklung	10
A. Anhang	ii
B. McCulloch-Pitts-Zelle	ii
C. McCulloch-Pitts-Zelle	ii

Abbildungsverzeichnis

1.	McCulloch-Pitts-Zellne - Genereller Aufbau	5
2.	Biologische Neuronen - Genereller Aufbau	6
3.	Biologische Neuronen - Notation	7
4.	McCulloch-Pitts-Zelle - AND Gatter	7
5.	McCulloch-Pitts-Zelle - OR Gatter	8
6.	McCulloch-Pitts-Zelle - OR Gatter	ii
7.	McCulloch-Pitts-Zelle - OR Gatter	ii
8.	McCulloch-Pitts-Zelle - OR Gatter	iii

Teil I.

Einführung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Teil II.

Geschichtliche Entwicklung

Im folgenden Abschnitt werde ich etwas auf die geschichtlichen Aspekte von neuronalen Netzen eingehen. Hierbei werden insbesondere die generellen Aspekte der generellen Funktionsweise von älteren Modellen bis hin zur aktuellen Entwicklung verfolgt. Ich werde versuchen die folgenden Leitfragen in diesem Abschnitt zu beantworten:

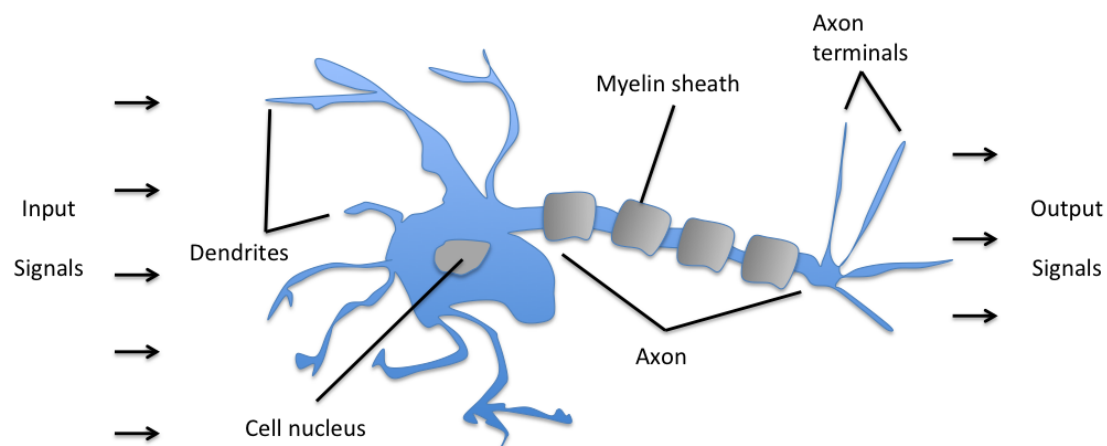
- Woher kommt Deep Learning und wie ist dieser Begriff im Kontext zur künstlichen Intelligenz einzuordnen?
- Welche Entwicklungen hat das Neuronale Netz von damals zu heute durchgemacht?

Um darauf näher eingehen zu können werden folgende Meilensteine behandelt:
todo Inhaltsverzeichnis (Sections)

1. McCulloch-Pitts-Neuron

1.1. Funktionsweise

Im Jahr 1943 entwickelten Warren McCulloch und Walter Pitts ein Modell welches die Funktionalität eines biologischen Neurons imitieren sollte. In der folgenden Abbildung 1 ist der grobe Aufbau eines Neurons zu sehen.



Schematic of a biological neuron.

Abbildung 1: McCulloch-Pitts-Zellne - Genereller Aufbau [1]

Die sogenannten *Dendriten* (englisch *dendrites*) nehmen Informationen auf. Sie besitzen Rezeptoren welche in der Lage sind Signale anderer Neuronen aufzunehmen. Diese Signale bewirken elektrische Veränderungen in dem Neuron welche vom Zellkörper (*Soma*) interpretiert / verarbeitet werden. Dieser Zellkörper sammelt alle Informationen und speichert diese im sogenannten *Axonhügel* (engl. Axonhillock) welcher die Ursprungsstelle des *Axons* beziehungsweise *Neuriten* beschreibt. Wenn das gebündelte Signal stark genug sein sollte wird es an den nächsten Teil des Neurons, dem *Axon*, weitergeleitet. Ab diesem Zeitpunkt wird das Signal als *Aktionspotential* bezeichnet und wird über die *Axon* übertragen. Am Ende wird das Signal an diverse *Axonterminale* weitergeleitet welche per Neurotransmitter mit den jeweils nächsten Dendriten verbunden sind.

Dieser biologische Aufbau dient als Grundlage für die Entwicklung des Modells von McCulloch und Pitts. Das Augenmerk ihres Modells liegt in erster Linie darauf logische Gatter mittels eines Neuronen ähnlichen Modells zu definieren. Der grobe Aufbau eines sogenannten *McCulloch-Pitts-Zelle* ist in 2 zu sehen.

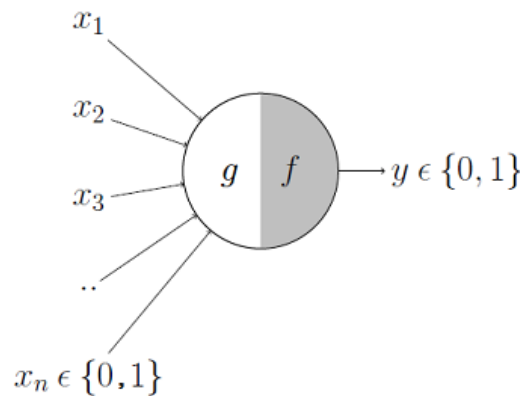


Abbildung 2: Biologische Neuronen - Genereller Aufbau [1]

Das Modell kann beliebig viele Input-werte aufweisen. Wichtig hierbei: Sie dürfen nur boolescher Natur sein (nur falsch oder wahr). Bei gegebenen Werten führt das Neuron selbst zwei Arbeitsschritte durch:

1. Erst werden alle Werte aufaddiert (in der Abbildung dargestellt durch die Funktion g). Dies imitiert das Verhalten der *Dendriten* in einem biologischen Neuron.
2. Anschließend überprüft die Funktion f ob ein gegebener Schwellwert überschritten wurde oder nicht (gibt dies entsprechend in Form einer booleschen Ausgabe weiter). Das biologische Neuron tut dies mittels des *Axonhügels*.

Die übliche Notation dieses Modells gibt vor, dass der jeweilige Schwellwert jeweils in die linke Seite des Kreises geschrieben wird während die rechte Seite ausgegraut wird. Im folgenden seien einmal beispielhaft das *AND* und das *OR* Gatter dargestellt. (Für weitere Beispiele siehe Anhang C)

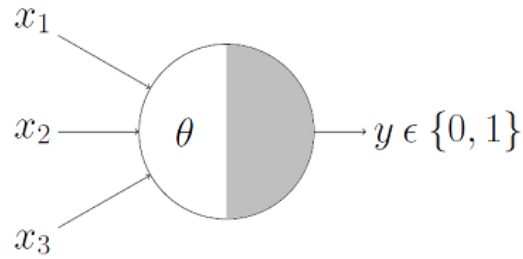


Abbildung 3: Biologische Neuronen - Notation [1]

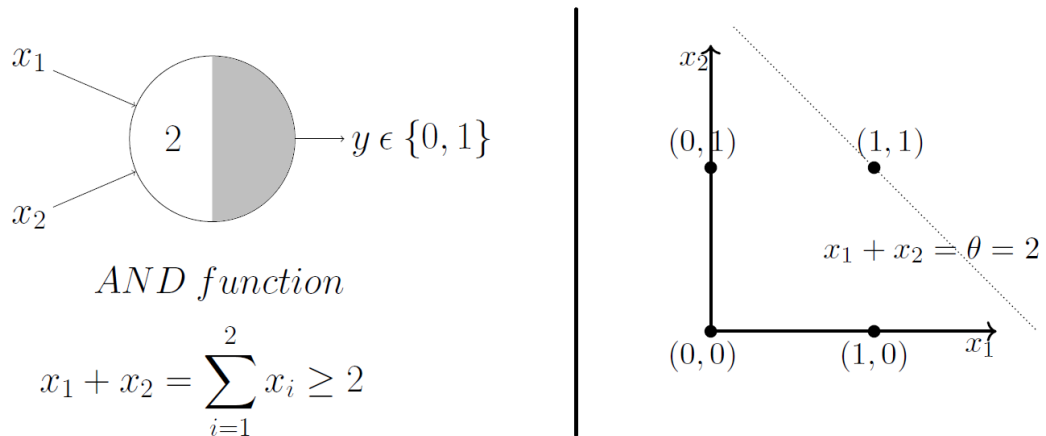


Abbildung 4: McCulloch-Pitts-Zelle - AND Gatter [1]

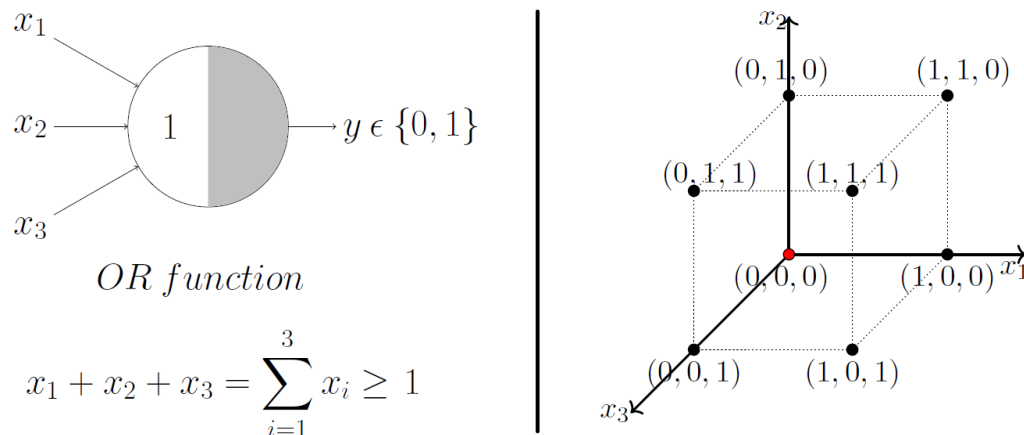


Abbildung 5: McCulloch-Pitts-Zelle - OR Gatter [1]

1.2. Nachteile bzw. Verbesserungspotenzial

- Dieses Modell erlaubt wie gesagt nur boolesche Eingabewerte, viele Modelle erfordern allerdings kontinuierliche Werte. Mit diesen wäre es zum Beispiel deutlich einfach ein Bild oder Ähnliches zu analysieren.
- Die Schwelle (Theta) muss stets manuell bestimmt werden. Einen Trainingsalgorithmus wie man ihn von heutigen Ansätzen her kennt gibt es in diesem Modell nicht.
- Es gibt keinerlei Priorisierungsmöglichkeit zwischen den einzelnen Eingabewerten. Jeder hat einen gleichgroßen Einfluss auf das Ergebnis, so etwas wie ein Auschlusskriterium gibt es hierbei also nicht.
- Es ist nicht möglich *gedeckt* Gatter wie zum Beispiel ein *XOR* abzubilden. Bei einem Neuron mit zwei Input müsste zum Beispiel ein Schwellwert von 1 genau getroffen werden. Dieses Modell ist allerdings nur in der Lage zu entscheiden ob ein Schwellwert *überschritten* wurde oder nicht.

2. Perceptron

Im Jahre 1958

3. Adeline

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Teil III.

Aktuelle Entwicklung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Literatur

- [1] McCulloch-Pitts Neuron. <https://towardsdatascience.com/mcculloch-pitts-model-5fdf65ac5dd1>. Aufgerufen am: 14-03-2020.
- [2] Single-Layer Neural Networks and Gradient Descent. https://sebastianraschka.com/Articles/2015_singlelayer_neurons.html. Aufgerufen am: 14-03-2020.
- [3] M.A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

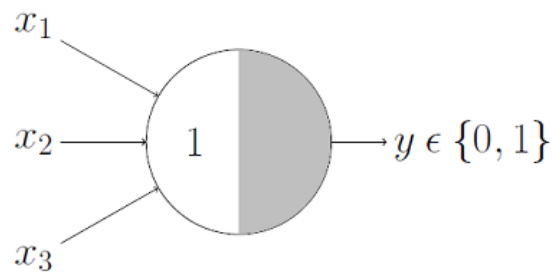
A. Anhang**B. McCulloch-Pitts-Zelle****C. McCulloch-Pitts-Zelle**

Abbildung 6: McCulloch-Pitts-Zelle - OR Gatter [1]

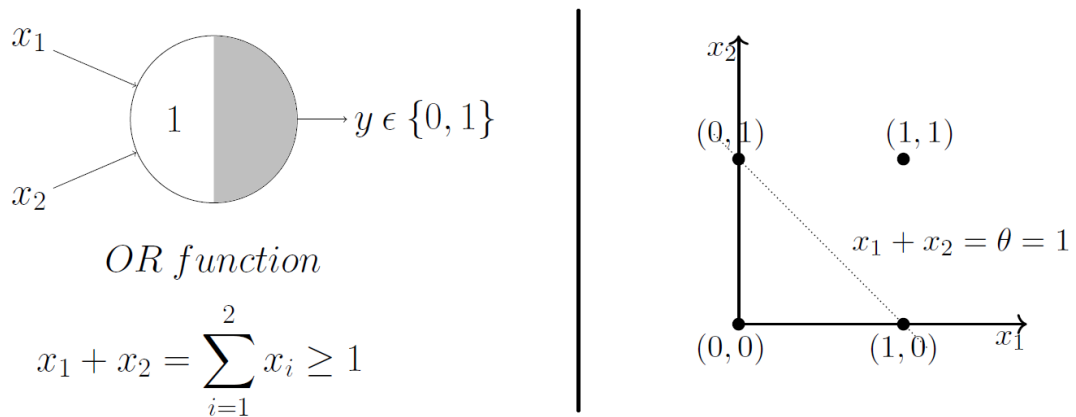


Abbildung 7: McCulloch-Pitts-Zelle - OR Gatter [1]

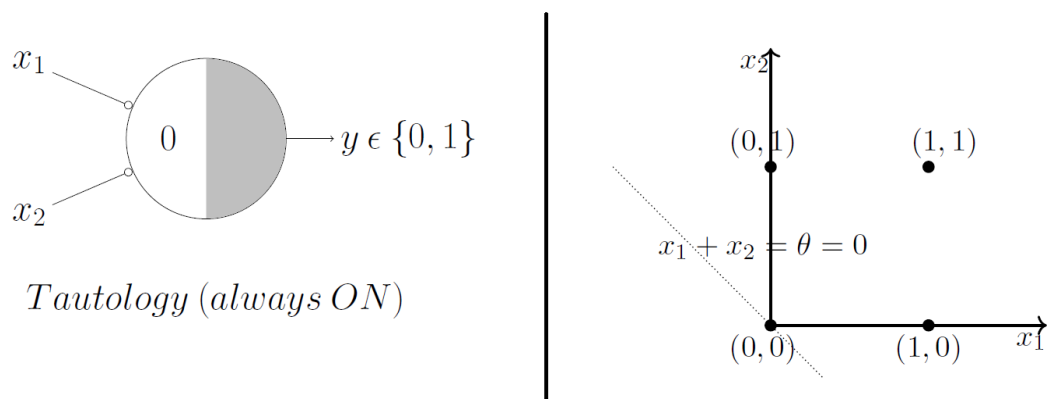


Abbildung 8: McCulloch-Pitts-Zelle - OR Gatter [1]