Berufsmaturitätsarbeit 2019-2020

Hardware-Neuron

Tendai Rondof / Malik Schneider







# Vorwort

## Tendai

Ich habe dieses Thema ausgewählt, weil Künstliche Intelligenzen mich seit Jahren interessieren. Durch dieses Projekt kann ich den Wunsch ein Neuron auf Hardwarebasis zu erstellen, endlich erfüllen.

## Malik

Ich habe das Thema gewählt, weil darin zwei meiner grössten Interessensgebiete vereinigt sind: Die Elektronik und die Neurowissenschaft. Vor allem bei letzterem kann ich hiermit mein Wissen erweitern und vertiefen. In der Elektronik kann ich vor allem mein bereits bestehendes Wissen anwenden.

Seit einiger Zeit boomt das Interesse an künstlichen Intelligenzen und neue Ideen wie sie umgesetzt werden können, sind hoch gefragt. Wir erwarten nicht dass wir damit das Rad neu erfinden, jedoch gelingt es uns einen Weg zu beschreiten der bis jetzt noch niemand gedacht hat.

<Danksagungen>

# Einleitung

was ist es?

was macht es?

was bringt es?

Das Ziel unserer BMA ist es, den Vorgang des Lernens mit greifbaren elektrischen Bauteilen zu simulieren. Dieser Aspekt allein hat schon eine gewisse Faszination an sich. Unter Machine-Learning versteht man etwas extrem Komplexes, was sich auf Softwareebene abspielt. Für eine Künstliche Intelligenz sind riesige Mengen an Daten Nötig um ihr etwas beizubringen. Aber wie so vieles lässt sich auch eine Intelligenz, ob sie jetzt künstlich ist oder nicht, herunterbrechen auf ihre Elementarsten Teile. Beim Menschen sind es die Neuronen, bei der Maschine elektrische Schaltkreise. Unser Produkt ist ein wenig von beidem.

Aufgrund unseres Lehrberufs, Elektroniker, setzten wir den Fokus auf die Funktionalität und nicht auf Biologische Richtigkeit. Das bedeutet, dass wir versuchten die chemischen Prozesse elektronisch zu imitieren. Es fielen zwar viele der Biologischen Aspekte eines Neurons weg, jedoch bleibt der Kern des ganzen schlussendlich bestehen: das Lernen.

Wegen der grossen Menge an kleinen Bauteilen in unserer Schaltung, verwenden wir eine Bestückungsmaschine anstelle des einfachen Lötkolbens verwenden. Das wird die Arbeit um einiges erleichtern.

Wie ist es möglich ein Neuron (Nervenzelle) mittels Elektronischen teilen zu realisieren und kann damit ein selbstlernendes Netzwerk (“Neuronales Netzwerk”) erstellt werden?

Inhalt

[Vorwort 2](#_Toc29817700)

[Tendai 2](#_Toc29817701)

[Malik 2](#_Toc29817702)

[Einleitung 2](#_Toc29817703)

[Blockschaltbild des gesamten 4](#_Toc29817704)

[Beispiel Komponente 4](#_Toc29817705)

[Theorie 4](#_Toc29817706)

[Grundlage der Problematik 4](#_Toc29817707)

[Ansatz 4](#_Toc29817708)

[Mathematischer Ansatz vorwärts Rechnung 5](#_Toc29817709)

[Fehler Berechnung 5](#_Toc29817710)

[*Problematik von Theorie und Praxis* 6](#_Toc29817711)

[*Adaption von der Theorie zur Machbarkeit* 6](#_Toc29817712)

[*Neue Bedingungen und Nachteile* 6](#_Toc29817713)

[Schema Blockschaltbild 6](#_Toc29817714)

[Beispiel Komponente 6](#_Toc29817715)

[Hardware 7](#_Toc29817716)

[Schema 7](#_Toc29817717)

[Ganzes Neuron 7](#_Toc29817718)

[Beispiel von Trainingsdaten 7](#_Toc29817719)

[Input 8](#_Toc29817720)

[DeltaApply 9](#_Toc29817721)

[Core 9](#_Toc29817722)

[Activation 10](#_Toc29817723)

[Vergleicher OPV 10](#_Toc29817724)

[DeltaCalc 11](#_Toc29817725)

[Verbindung Blockschaltbild Schema 11](#_Toc29817726)

[Layout/Bestückungsplan 11](#_Toc29817727)

[Messungen 11](#_Toc29817728)

[Benutzeranleitung 11](#_Toc29817729)

# Blockschaltbild des gesamten

Lichtsensor Wert

Schaltung

Schaltungs Ausgang

User Feedback (Sollwert)

A/D Wandlung

Licht- steuer-ung

## Beispiel Komponente

wofür ist sie zuständig:

verweise auf andere stellen im dokument:

# Theorie

## Grundlage der Problematik

was soll erreicht werden:

Es soll eine Schaltung erstellt werden, die Mathematischen Berechnungen eines Perceptron imitiert. Dadurch kann eine lernende Schaltung entwickelt werden die Dynamische bzw. lernende Schaltkreise ermöglichen. Zudem wird die Rechenzeit für ein Perceptron reduziert.

Bedingungen/Einschränkungen:

Es gibt gewisse Probleme bei der Übersetzung vom Model in die reale Welt. Zum einen ist die Praxis nie perfekt. Bauteile haben Toleranzen, Unendlichkeiten gibt es nicht und es ist nicht möglich Spannungen auf eine einfache Art und Weise zu multiplizieren.

Die gewünschte Formel wird auch nicht in einer reinen Form vorkommen.

## Ansatz

Typ des Perceptrons (Vorteile gegenüber anderen Methoden)

Als Grundprinzip benutzen wir das Model eines Spiking-Preceprons, was ein einzelnes Neuron mit einem pulsierenden, digitalen Ausgang darstellt.

Der Vorteil gegenüber anderen Methoden ist, es kommt der Biologie am nächsten und die Rückberechnung ist mit einzelnen Transistoren machbar.

Der Nachteil ist, es ist zeitabhängig. Was bedeutet, dass zeitlich alles gut aufeinander abgestimmt sein muss.

### Mathematischer Ansatz vorwärts Rechnung

Die Grundformen für den Kern des Neurons ist:

Wenn der Kern-Kondensator eine Gewisse Spannung erreicht wird ein «Spike» erzeugt, was am Ausgang ausgegeben wird. Dieser «Spike» wird durch eine Aktivierung Funktion erzeugt.

Sobald der Spike erzeugt wird, wird über den Discharge-Transitor auf 0V gezogen. Was zur folge hat, dass wenn der Input dauerhaft auf 5V gestellt ist, sich ein «Spike Train» mit einer Festen Frequenz bildet.

### Fehler Berechnung

Damit das Neuron überhaupt etwas lernt, muss eine Art Feedback vom Nutzer oder von der Umgebung kommen. Dafür wird ein Rückgabewert () verwendet. Mit diesem Rückgabewert kann die Gewichtung angepasst werden, in dem verändert wird.

: Kondensator am Gate des Transistors

: Spannung am Eingang

: Widerstand der die Ent-/Lade Geschwindigkeit des Kondensators angibt (Lern- Schrittgeschwindigkeit)

: Ziel Spike-Train, dass das Neuron erreichen soll

Da und zeitbegrenzte Pulse sind, wird der Kondensator nur ein klein wenig entladen/geladen was den Transistor am jeweiligen Eingang etwas mehr sperrt oder leitet. Dadurch können alle inputgewichte über die Zeit so eingestellt werden, dass dem Ausgang entspricht.

Dies ist natürlich nur bis zu einer gewissen Komplexität möglich. Vor allem da die eingangsströme nicht negativ sein können.

### Problematik von Theorie und Praxis

### Adaption von der Theorie zur Machbarkeit

### Neue Bedingungen und Nachteile

## Schema Blockschaltbild

Schaltung

3 Anschlüsse

0V/5V

Stecker

Sollwert

Output

## Beispiel Komponente

*Funktionsweis*

*Gleichung*

*Abweichung Theorie Praxis*

# Hardware

# Schema

## Ganzes Neuron



Der Eingang (Input) verarbeitet die Spannung an den Eingängen bis und summiert sie. Das Zentrum (Core) des Neurons speichert die Werte des Eingangs über die Zeit bis die Aktivierung (Activation) einsetzt und das Zentrum zurückgesetzt wird.

Der Eingang bei der Fehlerberechnung (DeltaCalcV2) entspricht dem Sollwert zu Trainingszeiten. Der Ausgang bei der Aktivierung wird zu jedem Zeitpunkt beim Training mit dem Sollwert mittels der Fehlerberechnung verarbeitet und Der Eingang angepasst.

Das Neuron lernt somit über Zeit und der Eingang wird bei jedem Fehler um einen konstanten minimalen Schritt-Wert angepasst um ein Schwingen um den Idealwert zu vermeiden und um das Neuron robuster gegen einzelne Fehler in den Trainingsdaten zu machen.

## Input



Die Eingangsspannungen bis werden über die, zum Teil leitenden, Transistoren in spezifische Ströme umgewandelt und am Ausgang summiert.

Die Spannungen an den Kondensatoren Bestimmen wie stark die Transistoren durchsteuern. So kann der Widerstand der Transistoren kontrolliert werden. Die Spannung an den Kondensatoren wird über die Fehlerberechnung verändert.

### DeltaApply



Dieser Teilschaltung ist für das anpassen der Spannungen am Kondensator zuständig. Jeder Eingang besitzt eine solche Teilschaltung. Der Eingang Delta bestimmt ob der Kondensator geladen oder entladen wird, Enable steuert mit der Spannung am

## Core



Das Core besteht hauptsächlich aus einem Kondensator welcher über den Eingang aufgeladen wird. Für den Fall, dass keine Signale über die Eingänge bis kommen, sorgt der Widerstand dafür, dass sich der Kondensator langsam entlädt und nach einer gewissen Zeit der Kondensator wieder zurückgesetzt ist.

## Activation



### Vergleicher OPV



## DeltaCalc



## Verbindung Blockschaltbild Schema

# Layout/Bestückungsplan

# Messungen

# Benutzeranleitung