Berufsmaturitätsarbeit 2019-2020

Hardware-Neuron

Tendai Rondof / Malik Schneider







Inhalt

[Vorwort 2](#_Toc26308252)

[Tendai 2](#_Toc26308253)

[Malik 2](#_Toc26308254)

[Einleitung 2](#_Toc26308255)

[Blockschaltbild des gesamten 4](#_Toc26308256)

[Beispiel Komponente 4](#_Toc26308257)

[Theorie 4](#_Toc26308258)

[Grundlage der Problematik 4](#_Toc26308259)

[Ansatz 4](#_Toc26308260)

[Mathematischer Ansatz vorwärts Rechnung 5](#_Toc26308261)

[Fehler Berechnung 5](#_Toc26308262)

[*Problematik von Theorie und Praxis* 6](#_Toc26308263)

[*Adaption von der Theorie zur Machbarkeit* 6](#_Toc26308264)

[*Neue Bedingungen und Nachteile* 6](#_Toc26308265)

[Schema Blockschaltbild 6](#_Toc26308266)

[Beispiel Komponente 6](#_Toc26308267)

[Hardware 7](#_Toc26308268)

[Schema 7](#_Toc26308269)

[Verbindung Blockschaltbild Schema 7](#_Toc26308270)

[Layout/Bestückungsplan 7](#_Toc26308271)

[Messungen 7](#_Toc26308272)

[Benutzeranleitung 7](#_Toc26308273)

# Vorwort

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

## 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 und trainieren, was meinem späteren Berufsleben ein wenig zu gute kommen könnte.

## Danksagungen

Wir möchten unserem Abteilungsleiter, Oliver Schneider, in der MSW danken, welcher uns erlaubte an den Wochenenden in der MSW an unserer BMA zu arbeiten. Ohne seine Hilfe hätten wir unser Produkt sicher nicht rechtzeitig beenden können.

Daniel Büchler, Abteilungsleiter für die ersten beiden Lehrjahre der Elektroniker in der MSW hat uns freundlicherweise den Fräsbohrplotter zu Verfügung gestellt. Dieser Umstand hat unsere Arbeit ungeheuer beschleunigt.

# Abstract

# 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. Denn wir simulieren das menschliche Neuron mit einer Elektrischen Schaltung.

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.

Um die Printplatte herzustellen, auf welcher wir unsere Schaltung aufgebaut haben, verwendeten wir den MSW-Internen Fräsbohrplotter, welcher uns die ersten Prototypen in höchstgeschwindigkeit zur Verfügung stellen konnte.

Die Frage die wir uns mit dieser BMA beantworten wollen lautet:

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

Um diese Frage dreht sich die ganze Arbeit unserer BMA. Darum auch der Name Hardware-Neuron. Unser Endprodukt wird Ausschlisslich aus Elektronischer Hardware bestehen. Mit unserem Endprodukt werden wir uns den ersten Teil der Frage also selbst beantworten.

Zu Beginn des Hauptteils wird das Konzept zur Lösung des Problems in all ihren Details und mathematischen Formeln erklärt erklärt. Danach wird die Schaltung erklärt die wir erstellt haben

# Hauptteil

# Blockschaltbild des Versuchsaufbaus

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

Die Grösste Problematik ist, dass das Speichern von Analogen Spannungen nur sehr schlecht geht und sehr störungsanfällig ist. Mann es

### Adaption von der Theorie zur Machbarkeit

### Neue Bedingungen und Nachteile

## Schema Blockschaltbild

Rückrechnung Steuern

0V/5V

Schaltung

3 Anschlüsse

0V/5V

Stecker

Sollwert für Output

Output

Rückrechnung

Anzeige für Rückrechnung

## Beispiel Komponente

*Funktionsweise*

*Gleichung*

*Abweichung Theorie Praxis*

# Detailiertes Gesamtschema von EAGLE



# Erklärung der Funktion

## 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



Die Aktivierung ist dazu da, um zu bestimmen ob ein Signal weiter bzw. ausgegeben werden soll. Wenn die Spannung am Core (Kondensator) über die 2.5V steigt, schaltet der Vergleicher von 0V auf 5V über eine sehr kurze Zeit. Durch diese Rapide Änderung gibt es eine Spannungsspitze auf der anderen Seite des Kondensators. Diese Spannungsspitze wird über die zwei folgenden Transistoren verstärkt, damit das Ausgangssignal eine höhere Spannung erreichen kann und auch mehr Strom liefern kann.

Dieses Ausgangssignal wird über zurückgeführt und entlädt den Core (Kondensator) sehr schnell. Wodurch dieser wieder auf unter die 2.5V fällt und der Ausgang des Vergleichers wieder auf 0V schaltet. Danach beginnt der Vorgang von vorne.

Die Diode am Ausgang sorgt dafür das wen der Ausgang des Vergleicher auf 0V ist, der Ausgang der Ganzen Schaltung einem Tri-state (Das bedeutet das kein Strom in die Schaltung fliessen kann).

### Vergleicher OPV



Dies ist eine primitive Schaltung eines Komparators bzw. Vergleicher, welcher in der Elektronik eine Essenzielle Rolle spielt. Er vergleicht die Eingänge + und – und schaltet den Ausgang auf 5V wenn die Spannung am + Eingang Grösser ist als die am – Eingang. Ist das umgekehrte der Fall wird der Ausgang auf 0V gesetzt.

## DeltaCalc



Die Delta-Calculation (DeltaCalc) ist dafür zuständig zu entscheiden ob der Ausgang (Ist-Wert) mit dem Soll-Wert übereinstimmt. Der Ausgang «Delta» von diesem Abschnitt gibt an ob die Kondensatoren an den Eingängen bis geladen oder entladen werden sollen. Der Ausgang «En» (Kurz für Enable) von diesem Abschnitt sagt ob dieses Laden oder Entladen der Kondensatoren überhabt durchgeführt werden soll oder nicht.

In Digital-technischer Sprache:

Der Ausgang «Delta» entspricht einem Und Gatter mit den Eingängen und . Der Ausgang «En» ist eine XNOR Verknüpfung von und . Sprich wen dann sollen keine Änderungen an den Kondensatoren vorgenommen werden.

## Verbindung Blockschaltbild Schema

# Layout/Bestückungsplan von EAGLE



# Messungen

# Benutzeranleitung

# Abkürzungsverzeichnis

* EAGLE einfach anwendbarer Grafischer Layout Editor

Wir haben dieses Programm verwendet, um darin unser Schema zu zeichnen und den Print zu designen