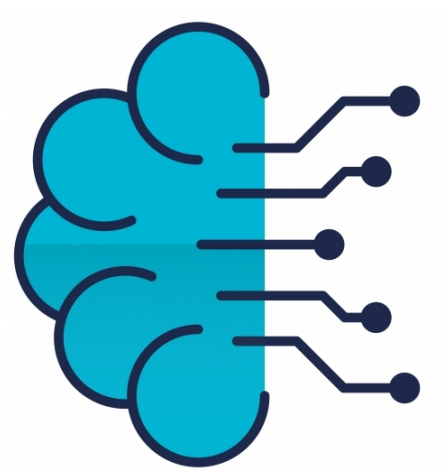
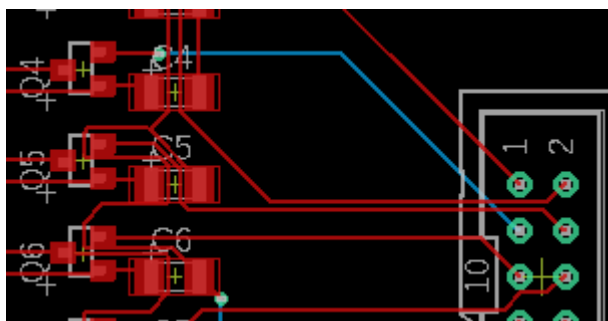
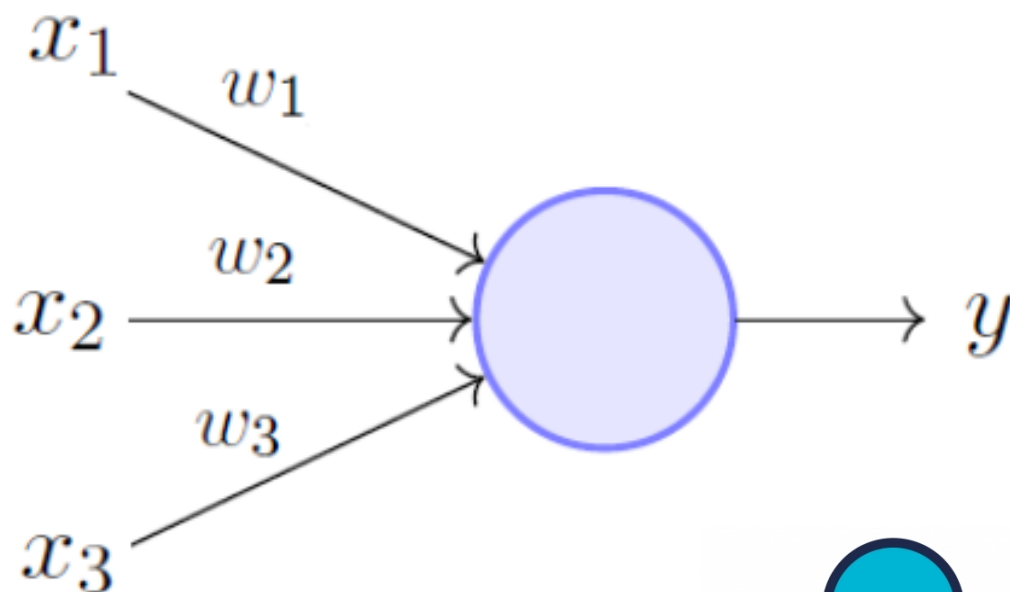
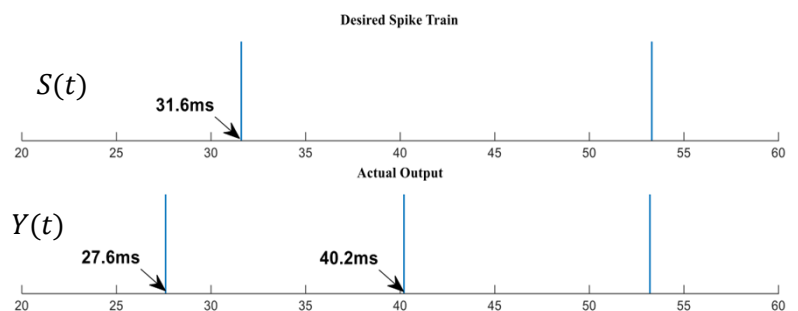


Berufsmaturitätsarbeit 2019-2020

Hardware-Neuron

Tendai Rondof / Malik Schneider



Betreuende Lehrperson: M.Mohr

I. Inhaltsverzeichnis

II.	Vorwort	4
A.	<i>Tendai</i>	4
B.	<i>Malik</i>	4
C.	<i>Danksagungen</i>	4
III.	Abstract	4
A.	<i>Deutsch</i>	4
B.	<i>English</i>	5
IV.	Einleitung	5
V.	Modelle eines Neurons	6
A.	<i>Biologisches Model</i>	6
B.	<i>Informatisches Model</i>	7
C.	<i>Anforderungen an unser Modell</i>	8
D.	<i>Überprüfung des Modells</i>	9
1.	Testmöglichkeiten	11
2.	Unmöglichkeiten	11
E.	<i>Unser Modell</i>	11
A.	<i>Mathematischer Ansatz</i>	12
1.	Spannung am Kernkondensator	12
2.	Aktivierung	13
3.	Fehlerberechnung	14
B.	<i>Problematik von Theorie und Praxis</i>	14
C.	<i>Bedingungen/Einschränkungen</i>	15
D.	<i>Blockschaltbild</i>	15
VI.	Praxis	16
A.	<i>Ganzes Neuron</i>	16
B.	<i>Input</i>	16
1.	DeltaApply	17
C.	<i>Core</i>	18

D.	<i>Activation</i>	18
1.	Vergleicher OPV	19
E.	<i>Delta Calc</i>	20
VII.	Messungen	21
A.	<i>Messung des Eingangs</i>	21
B.	<i>Messung der Aktivierung</i>	21
C.	<i>Messung der Delta Calculation</i>	22
D.	<i>Messung von DeltaApply</i>	22
VIII.	Fazit	23
IX.	Formelverzeichnis	24
X.	Abkürzungsverzeichnis	24
XI.	Abbildungsverzeichnis	25
XII.	Literaturverzeichnis	25
XIII.	Anhang	26
A.	<i>Layout/Bestückungsplan von EAGLE</i>	26
B.	<i>Schema von EAGLE</i>	26
C.	<i>Stückliste</i>	26
XIV.	Deklaration	26

II. 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 Neuron, dass aus einem elektronischen Schaltkreis besteht, das Rad neu erfinden, jedoch gelingt es uns vielleicht einen Weg zu beschreiten der bis jetzt noch niemand bedacht hat.

A. 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.

B. Malik

Ich habe das Thema gewählt, weil darin zwei grosse Interessensgebiete von mir 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 zugutekommen könnte.

C. 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. Vor allem für die endgültig letzte Version des Prints war diese Maschine unentbehrlich, weil der bestellte Print aus dem Ausland zu spät angekommen ist.

III. Abstract

A. Deutsch

In dieser BMA stellen wir, Tendai und Malik, uns die Frage, ob es möglich ist, mit simplen elektronischen Bauteilen eine Schaltung zu entwerfen, die bestimmte, elementare Logik erlernen und anschliessend wiedergeben kann. Um dies zu erreichen, verwenden wir das Modell eines Neurons. Wir erläutern Struktur und Funktion des biologischen und des informatischen Modells. Schliesslich fällt unsere, ausführlich begründete Entscheidung auf das spiking neural Network. Das die Version des informatischen Modells, welches dem biologischen am nächsten kommt. Danach nähern wir uns mittels mathematischen Formeln, basierend auf dem spiking neural Network, den Funktionen an, welche die Schaltung erfüllen

muss. Daraufgehend werden die Teilfunktionen erklärt und es wird erläutert, wie unsere Schaltung sie bewältigen soll. Die Messungen des Prototypen zeigen, dass die Vorwärtsrechnung, also das wiedergeben des gelernten, funktioniert. Das Problem ist, dass bei unserem Endprodukt nichts funktioniert, da die Leiterbahnen auf dieser Version sehr kurzschlussanfällig sind und wir keine Zeit mehr hatten, alle Kurzschlüsse zu beheben.

B. English

In this BMA we, that's Malik and Tendai, ask ourselves, whether or not it's possible, to design an electrical circuit, which can learn certain, elementary logic and replicate it subsequently. In order to reach this, we use the model of a neuron. We explain structure and function of the biological and informatical model. In the end, our decision falls onto the spiking neural network. That's the informatical model, which comes closest to the biological one. Then, we draw up the functions, that our circuit has to implement, through mathematical formulas. After that, the purposes of the smaller subcircuits, we designed, is explained. In the measurements of the Prototype it is shown, that the replication of logic works. Unfortunately the learning itself doesn't. Besides of that, our final product doesn't work at all, because there are shortcircuits all over the print and we didn't have time, to repair them all.

IV. Einleitung

Das Ziel unserer BMA ist es, den Vorgang des Lernens mit greifbaren, elektronischen Bauteilen zu simulieren. Dieser Aspekt allein hat schon eine gewisse Faszination an sich. Unter Machine-Learning versteht man etwas 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; 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 Funktion 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 LPKF ProtoMat S63, welcher uns die ersten Prototypen in Höchstgeschwindigkeit zur Verfügung stellen konnte. Da wir zum Schluss unserer BMA noch eine letzte Version aus dem Ausland bestellten, welche sich verspätete, mussten wir leider auch diesen noch selbst ausfräsen lassen.

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

Was ist eine Möglichkeit, ein Neuron (Nervenzelle) mittels Elektronischen teilen zu realisieren und kann damit eine selbstlernende, elektrische Schaltung ("Neuron") erstellt werden?

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

Zu Beginn des Hauptteils wird das Konzept zur Lösung des Problems in all ihren Details und mathematischen Formeln erklärt. Danach wird die Schaltung erklärt, die wir erstellt haben und nach den erläuterten Funktionen arbeitet. Im Anhang finden sie das Design und das Schema des Prints auf EAGLE¹ sowie die Auflistung der Bauteile, welche auf den Print gehören.

V. Modelle eines Neurons

In diesem Kapitel Zeigen wir zwei unterschiedliche Modelle eines Neurons in verschiedenen Fachbereichen. Das Kapitel macht verständlich, welche Anforderungen wir an unser Produkt haben und wie wir überprüfen ob diese Anforderungen erfüllt wurden. Dieses Kapitel enthält auch unsere Hypothese, wie diese Anforderungen erfüllt werden können, inklusive den im Voraus ersichtlichen Problemen.

A. Biologisches Model

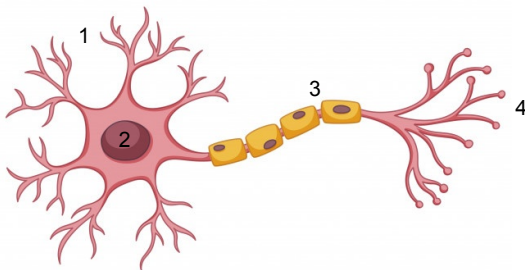


Abb 1: Biologisches Neuron (Wikipedia, 2020)

Das Neuron ist eine Zelle, die auf Signalübertragung spezialisiert ist. Sie kann Reize aufnehmen und ein Signal abgeben. In grosser Zahl und miteinander verbunden, bilden sie das Nervensystem nahezu sämtlicher Säugetiere. Und ihre wohl wichtigste Funktion für uns Menschen ist, dass unser Zentrales Nervensystem (Gehirn) aus ihnen gebaut ist. Nach einer Schätzung etwa neunzig Milliarden². Die Signalübertragung geschieht über elektrische

¹ EAGLE ist ein Layouteditor, der es ermöglicht, Leiterplatten zu designen

² Diese Schätzung wurde aus (Wikipedia, 2020) entnommen

Signale, genau genommen über das weiterleiten von Ionensättigungen, und nicht wie bei einer elektrischen Schaltung über einen Elektronenfluss.

In der Abb 1 ist ein solches Neuron Modellgetreu abgebildet. Die Dendriten (Abb 1.1), auch Postsynapse genannt, nehmen die Reize von einem Verhergehenden Neuron oder einer Zelle auf. Der Zellkern/Zellkörper (Abb 1.2) ist das produktions und Speicher Zentrum der Zelle. Dort wird die DNA aufbewahrt, der Stoffwechsel mit der Umgebung durchgeführt und Proteine produziert. Die für uns relevante Funktion des Zellkerns und des Körpers ist, dass dort die Signale von den Dendriten integrierend³ verarbeitet werden. Über das Axon (Abb 1.3) wird das integrierte Signal nach dem überschreiten eines Thresholds als ein Impuls übertragen und an die Endknöpfchen (Abb 1.4), auch präsynapse genannt, übergeben. Die Endknöpfchen leiten das Signal mittels Neurotransmitter⁴ an ein anderes Neuron weiter. Dieser ganze Durchlauf dauert einige Millisekunden.

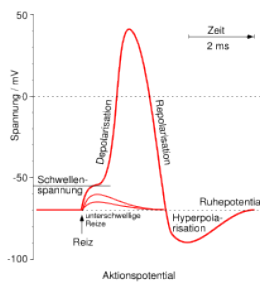


Abb 2: Aktionspotenzial im Axon (Wikipedia, 2020)

Der Impuls, der das Neuron am Axon empfängt, hat in etwa eine Form wie in Abb 1. In der Abbildung wird auch ersichtlich, wie das Potenzial zuerst in einer e-Kurve vom Ruhepotential ansteigt und ab der Schwellenspannung exponentiell bis zu einer Maximalspannung ansteigt, bevor die Spannung wieder zusammenbricht und das Neuron bis zum Wiedererlangen des Ruhepotentials Nichts mehr tun kann.

B. Informatisches Model

Das künstliche Neuron ist eine Struktur, welche dazu in der Lage ist, eine Funktion zu approximieren. Im Grunde hat diese Struktur im Minimum drei Abschnitte, die jeweils zu einem Abschnitt im Biologischen Neuron korrespondieren. Bisher wurden so viele Variationen der einzelnen Abschnitte entwickelt, die immer auf eine Anwendung optimiert sind, dass man sie hier nicht alle auflisten kann. Im Gegensatz zum biologischen Neuron hat das Künstliche Neuron eine Laufzeit von meist unter einer μs , was mehr als tausend Mal schneller ist.

³ Die Integration ist eine berechnung, die die Fläche unter einer Funktion bzw. Signalverlauf beschreibt

⁴ Botenstoffe die das Signal über die Synapse übertragen

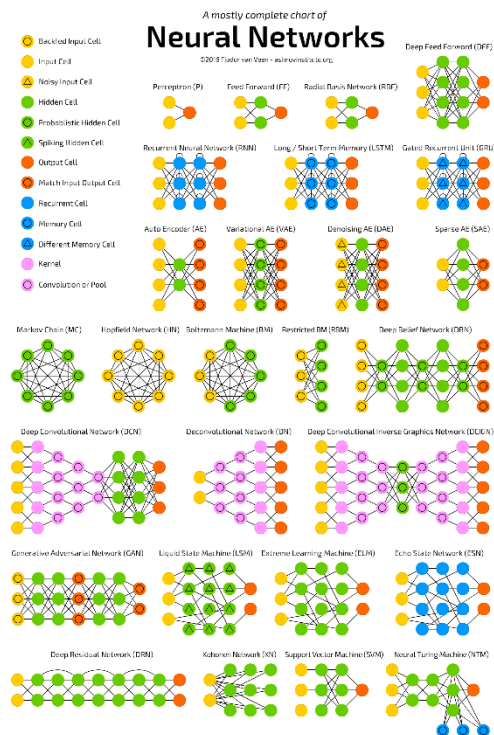


Abb 3: Chart of Neural Networks (Tch, 2017)
oder Spracherkennung, übernehmen können.

Einfach gesagt, werden die Eingangssignale unterschiedlich gewichtet. Das bedeutet, sie werden mit einem Wert, der zuvor gelernt wurde, multipliziert, dann summiert und zum Schluss durch eine Aktivierungsfunktion ausgegeben. Diese Aktivierungsfunktion ist meistens nicht linear oder eine Schrittfunktion.

Wie beim biologischen Neuron, ist in der Informatik ein einzelnes Neuron ziemlich nutzlos, da sie ausser simple Aufgaben, wie die Funktion verschiedener logischer Operatoren, nichts können. Darum begann man, diese künstlichen Neuronen miteinander zu verknüpfen, wodurch künstliche neuronale Netzwerke entstanden sind. Mit der Hilfe dieser Netzwerke werden bis heute noch Künstliche Intelligenzen konstruiert, die Aufgaben, wie Bild-

In Abb 3 werden alle bisher standardisierten neuronalen Netzwerke aufgelistet. Wobei die Punkte einem Neuron, und die Linien einer Verbindung zwischen zwei oder mehr Neuronen entsprechen.

An diesem Punkt kommt, eine für unsere BMA, wichtige Eigenschaft von künstlichen Neuronen ins Spiel.

Neuronale Netze werden auf einem Computer simuliert. Das bedeutet, dass der Signalverlauf durch das neuronale Netzwerk sequenziell und in der Taktung des Computers verläuft. Ein Computer kann immer nur eine Aufgabe zu einem Zeitpunkt erledigen und ist durch das, trotz der schnelleren Laufzeit, pro Neuron dem biologischen Neuron, welches alle Neuronen und Signale Simultan und in Echtzeit «Berechnen» kann, weit unterlegen.

C. Anforderungen an unser Modell

Es soll eine Schaltung erstellt werden, die Mathematischen Berechnungen eines Neurons imitiert. Dadurch kann eine lernende Schaltung entwickelt werden, die dynamische bzw. lernende Schaltkreise ermöglichen.

Als «lernen» ist zu verstehen, dass die Schaltung berechnet, wie sie welche Eingänge verarbeiten soll. Das passiert, indem der Nutzer am Eingang Daten anlegt und für eine

gewisse Trainingsphase, gleichzeitig einen Sollwert für den Ausgang definiert. Mit der Zeit lernt also die Schaltung, wie sie die Eingangsdaten verarbeiten soll und wie sie darauf reagieren soll. Nach der Trainingsphase soll die Schaltung selbständig neue Eingangsdaten interpretieren und einordnen.

Im Idealfall soll unser Neuron so schnell sein, wie das Neuron aus der Informatik, funktionieren und dennoch in Echtzeit und parallel die Berechnungen durchführen. Um die Schaltung einfach zu skalieren⁵ und eine Abhängigkeit von Herstellern zu vermeiden, soll es nur aus einfachen elektronischen Komponenten, wie Transistoren, Widerstände, Dioden und Kondensatoren bestehen. Das bedeutet, dass keine integrierte Schaltungen, auch ICs genannt, verwendet werden.

D. Überprüfung des Modells

Damit das Folgende auch gut verstanden werden kann, eine kurze Erklärung zu den Fachbegriffen. In der Elektronik bzw. Digitaltechnik wird mit dem binären Zahlensystem gearbeitet. Diese Zahlen werden aus Einsen und Nullen gebildet.

Jede 0 und 1 hat einen Zahlenwert, den sie repräsentiert. Im Binärsystem wird die Wertung von rechts nach links pro Stelle verdoppelt. D.h. die erste Ziffer von rechts steht für eins, die zweite für zwei, die dritte für 4 usw.. Steht eine Eins an der Stelle wird die entsprechende Wertung dazuaddiert, steht da eine Null, wird sie weggelassen.

Bsp.:
0001 entspricht im Dezimalsystem der Zahl 1.
0010 entspricht der Zahl 2.
0011 entspricht der Zahl 3.
0100 entspricht der Zahl 4.
Usw.

In der Praxis entspricht die 1 einer Spannung von 5 Volt (auch «High» genannt) und die 0 einer Spannung von 0V (auch «Low» genannt).

Um diese Informationen in der elementarsten Weise zu verarbeiten, werden drei Grundgatter verwendet:

AND:



Abb 4: UND Gatter (Stefan506, Wikipedia, 2005)

⁵ Um auch eine Verwendung als Integrierte Schaltung zu ermöglichen

Tabelle 1: Logiktablelle von einem UND Gatter

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Das AND-Gate verknüpft die Eingänge (A & B) so, dass der Ausgang (Y) nur einem «High» entspricht, wenn A und B zur gleichen Zeit auch «High» sind. Jede andere Kombination führt zu einem «Low» am Ausgang.

OR:

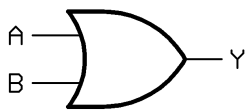


Abb 5: ODER Gatter (Stefan506, Wikipedia, 2005)

Tabelle 2: Logiktablelle von einem ODER Gatter

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Das OR-Gate verknüpft die Eingänge (A & B) so, dass der Ausgang (Y) einem «High» entspricht, wenn mindestens einer der beiden Eingänge «High» ist.

NOT:

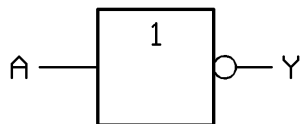


Abb 6: NICHT Gatter (Stefan506, Wikipedia, 2005)

Tabelle 3: Logiktablelle eines NICHT Gatter

A	Y
0	1
1	0

Das NOT-Gate besitzt nur einen Eingang (A). Das Gatter macht nichts anderes, ausser dem Gegenteil vom Eingang auszugeben.

1. Testmöglichkeiten

Um zu überprüfen ob das Neuron auch lernt, soll es die Aufgabe von logischen Gattern (AND, OR) übernehmen bzw. lernen. Um zu überprüfen, ob das Neuron in der Lage ist, zu merken, wenn ein Eingang nicht benötigt wird, wird neben den notwendigen Eingängen auch noch ein überflüssiger Eingang benutzt, der Ignoriert werden soll. Wenn dies zuverlässig funktioniert, kann man davon ausgehen, dass das Neuron in der Lage ist bedingt komplexe⁶ Zusammenhänge zu approximieren und wichtige von nicht wichtiger Information zu unterscheiden.

2. Unmöglichkeiten

Da das Neuron auf keinen Fall in der Lage ist, Invertierungen (NOT-Gatter) zu imitieren, sind Kombinationen auf der Basis «Wenn ein Eingang **nicht** eingeschaltet ist, dann...» von Anfang an auszuschliessen und unmöglich für das Neuron zu erlernen.

Um es in technischer Sprache zu sagen, das Neuron ist nicht in der Lage XOR, XNOR, NAND, NOR oder NOT Gatter zu erlernen, da nach dem Aufbau des Eingangs (Inputs⁷) immer ein Stromzufluss herrschen muss.

E. Unser Modell

Aus den Anforderungen wird klar, dass wir kein Rechenzentrum in getakteter form haben können, sondern eine Schaltung benötigen die fliegend den Ausgang berechnen kann.

Bei dem Paper «Introduction to spiking neural networks» von (Ponulak & Kasinski, 2011) ist uns aufgefallen, dass in der Informatik eine Art von neuronalem Netzwerk existiert, das darauf ausgelegt ist, ein biologisches Neuron so gut wie möglich zu imitieren. In der Informatik konnte sich diese Art von neuronalem Netzwerk bisher noch nicht gut durchsetzen, weil es mehr Rechenleistung als ein Netzwerk einer anderen Art benötigt und schwer zu trainieren ist.

Die Funktion dieser Art Neuron ist identisch zu der Grundstruktur vom Informatischen Model, jedoch ist der Ausgabewert nicht analog, sondern in einer digitalen Form, nach dem «Alles oder Nichts»- oder dem Binärprinzip. Von diesem Prinzip hat diese Art Netzwerk ihren Namen «Spiking Neural Network», was auf Deutsch «Gepulstes Neuronales Netzwerk» bedeutet.

⁶ Darunter werden Zusammenhänge, die ein mehrschichtiges Neuronales Netz erfordern, verstanden.

⁷ Die Schaltung wird im Kapitel VI.B Input auf der Seite 14 genauer erklärt.

Dieses gepulste Neuronale Netzwerk benötigt vor allem mehr Rechenleistung, weil es die elektronischen Vorgänge in einem Neuron simuliert. Das ist bei einem getaktetem Computer durchaus ein Problem, in der Elektronik jedoch nicht, da dort die Physik das Berechnen der Schaltung übernimmt.

Der Nachteil ist, es ist zeitabhängig. Was bedeutet, dass zeitlich alles gut aufeinander abgestimmt sein muss, was das Trainieren unheimlich erschwert.

A. Mathematischer Ansatz

In diesem Abschnitt wird die Berechnung der Schaltung vorgestellt und erklärt. Diese Formeln ermöglichen uns, zu erraten welche Schaltungsteile wir benötigen und wie wir sie ungefähr verbinden müssen. Sie zeigen auch auf, welche Funktionen erfüllt werden müssen. Es ist zu beachten, dass mit idealen Bauteilen gerechnet wird und keine Toleranzen von Bauteilen miteinbezogen sind, da diese nicht vorhersehbar sind.

1. Spannung am Kernkondensator

Aus diesem Paper «Introduction to spiking neural networks» von (Ponulak & Kasinski, 2011) konnten wir auch die Formel

$$C \frac{du}{dt}(t) = -\frac{1}{R}u(t) + \left(i_o(t) + \sum w_j i_j(t)\right), \quad (1)$$

Abb 7: «The Dynamics of the LIF unit» (Ponulak & Kasinski, 2011)

entnehmen, welche die Basis für die mathematische Beschreibung von unserem Modell ist. Aus der Formel von Abb 7: «The Dynamics of the LIF unit» (Ponulak & Kasinski, 2011) kann man herauslesen, dass die Eingänge i mit einem Werte w multipliziert (Gewichtet) und summiert werden. Hinzu kommt der sogenannte Bias i_0 , welcher einen Grundladestrom der konstant ist liefert. Von diesem Zuflussstrom wird ein Abflussstrom abgezogen, welcher von der Spannung u zum Zeitpunkt t und dem Widerstand R abhängig ist. Das C ist ein Kondensator der als Integrierendes Element benutzt wird.

Damit wir diese Formel aus Abb 7 in einer Schaltung darstellen können, müssen wir jedoch einige Änderungen vornehmen. Das grösste Problem ist die Gewichtung w , welche mit dem Eingangsstrom multipliziert wird. Es ist in der Elektronik nur sehr schwer möglich, Spannungen mit einer Variabel zu Multiplizieren, zudem wird dies durch die Anforderung, dass nur Einfache Bauelemente benutzt werden, nahezu unmöglich für uns.

Unsere Lösung ist, dass wir die Eingangsspannung durch einen variablen Widerstand teilen, was zu einem variablen Strom führt. Ausserdem ist in der Formel aus Abb 7 keine Funktion für das Zurücksetzen der Kondensatorladung miteinbezogen. Da dieses Zurücksetzen für uns eine essentielle Komponente ist, müssen wir sie in unserer Formel mit einberechnen.

$$C \frac{dU_c(t)}{dt} = -\frac{U_c(t)}{R_0} + \left(\sum_i^n \frac{U_{x_i}(t)}{R_{T_i}(U_{c_{T_i}}(t))} \right) + I_{bias}(t) - \frac{U_c(t)}{R_{disc}(U_a(t))}$$

Formel 1: Berechnung der Kernkondensatorspannung

U_c : Die Spannung am Kernkondensator

C : Kapazität des Kernkondensators

R_0 : Entladungswiderstand am Kernkondensator

n : Anzahl Eingänge

U_{x_i} : Spannung am Eingang i

R_{T_i} : Funktion des Transistor – Widerstands in Abhängigkeit der Spannung $U_{c_{T_i}}$

$U_{c_{T_i}}$: Spannung des Kondensators am Gate des Transistors

I_{bias} : Strom des statischen 5V Eingangs

R_{disc} : Widerstand des Entladungstransistors

Wenn der Kern-Kondensator eine gewisse Spannung (U_{ref}) erreicht, wird ein «Spike» erzeugt, was am Ausgang ausgegeben wird. Dieser «Spike»⁸ wird durch eine Aktivierungsfunktion erzeugt. Beim Generieren dieses «Spikes» wird der discharge-Transistor durchgesteuert und der Widerstand R_{disc} geht gegen 0Ω , was zu einer plötzlichen Entladung des Kernkondensators führt.

2. Aktivierung

$$Y(t) = \frac{d}{dt} \left(\begin{cases} 0, U_c(t) < U_{ref} \\ 1, U_c(t) > U_{ref} * V_{cc} \end{cases} \right)$$

Formel 2: Funktion der Ausgabe

U_c : Kernkondensator Spannung

U_{ref} : Statische Referenz Spannung

V_{cc} : Speise Spannung des Prints

⁸ Eine kurze Spannungsspitze

Y : Ausgabe der Schaltung

Sobald der Spike erzeugt wird, wird U_c über den Discharge-Transistor 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.

3. Fehlerberechnung

Damit das Neuron überhaupt etwas lernt, muss eine Art Feedback vom Nutzer oder von der Umgebung kommen. Dafür wird ein Rückgabewert (S) verwendet. Mit diesem Rückgabewert kann die Gewichtung $\frac{1}{R_{T_i}(U_{c_{T_i}}(t))}$ angepasst werden, indem $U_{c_{T_i}}$ verändert wird.

$$\frac{dU_{c_{T_i}}(t)}{dt} = \frac{1}{V_{cc}^2 * R_{LR}} * U_{X_i}(t) * |S(t) - Y(t)| * (|S(t) - Y(t)| - U_{c_{T_i}})$$

Formel 3: Berechnung des Fehlers

$U_{c_{T_i}}$: Spannung des Kondensators am Gate des Transistors

C_{T_i} : Kondensator am Gate des Transistors

U_{X_i} : Spannung am Eingang i

R_{LR} : Widerstand, der die Ent-/Lade Geschwindigkeit des Kondensators C_{T_i} angibt (Lern-Schrittgeschwindigkeit)

V_{cc} :Speise Spannung des Prints

S : Ziel Spike-Train, dass das Neuron erreichen soll

Da S und Y zeitbegrenzte Pulse sind, wird der Kondensator nur ein klein wenig entladen/geladen was den Transistor am jeweiligen Eingang etwas mehr sperren oder leiten lässt. Dadurch können alle Inputgewichte über die Zeit so eingestellt werden, dass $S(t)$ dem Ausgang $Y(t)$ entspricht.

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

B. Problematik von Theorie und Praxis

Die grösste Problematik ist, dass das Speichern von analogen Spannungen nur sehr schlecht geht und sehr störungsanfällig ist. Man kann die Schaltung etwas stabilisieren,

indem man den Print so gut wie möglich isoliert, um jeglichen Energieverlust über benachbarte Leitungen zu vermeiden.

C. Bedingungen/Einschränkungen

Es gibt gewisse Probleme bei der Übersetzung vom Modell 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. Hinzu kommt, dass die gewünschte Formel auch nicht in reiner Form vorkommen wird.

Um die Schaltung kleiner, bzw. in einer sinnvollen Grösse, zu behalten, verzichten wir auf negative Eingangsströme, was die Diversität an lernbaren Funktionen einschränkt, aber keinen Einfluss auf die Funktion der Schaltung hat.

D. Blockschaltbild

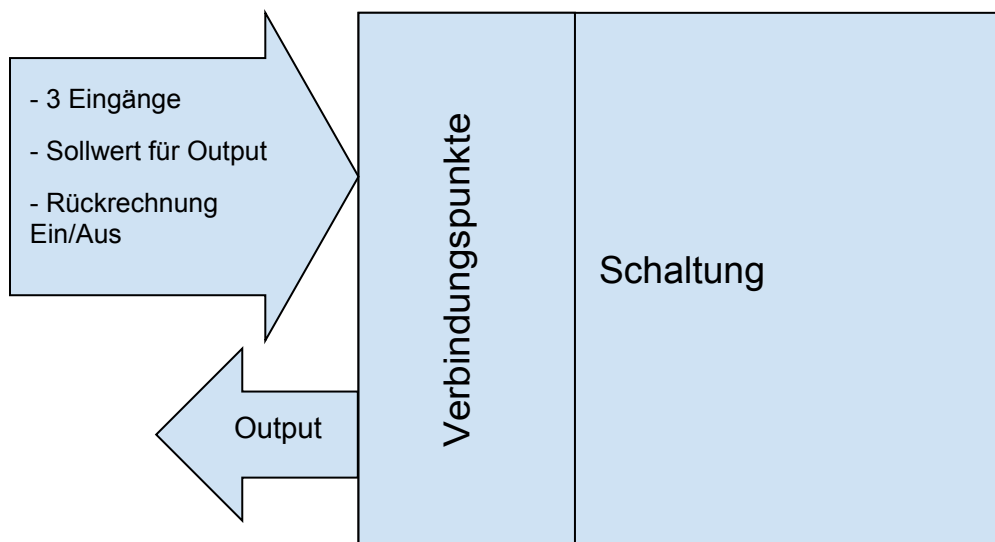


Abb 8: Blockartige Übersicht der Schaltung

In der obigen Abbildung (Abb 8) wird die minimalste Grundstruktur des gesamten Systems dargestellt. Die gesamte Peripherie der Schaltung besteht aus sechs Verbindungspunkten. Drei Eingänge, welche die Signale in die Schaltung führen. Einen Eingang für den Sollwert, welcher die Schaltung in der Trainingsphase benötigt, um zu lernen. Der Eingang für die Rückrechnung Ein/Aus ist dafür da, um der Schaltung zu sagen ob sie sich in der Trainingsphase oder im Normalbetrieb befindet. Der Output ist die Ausgabe der Schaltung für einen gegebenen Eingangszustand an den drei Eingängen. Die Ausgabe der Schaltung ist immer vorhanden, auch wenn sie sich in der Trainingsphase befindet.

VI. Praxis

A. Ganzes Neuron

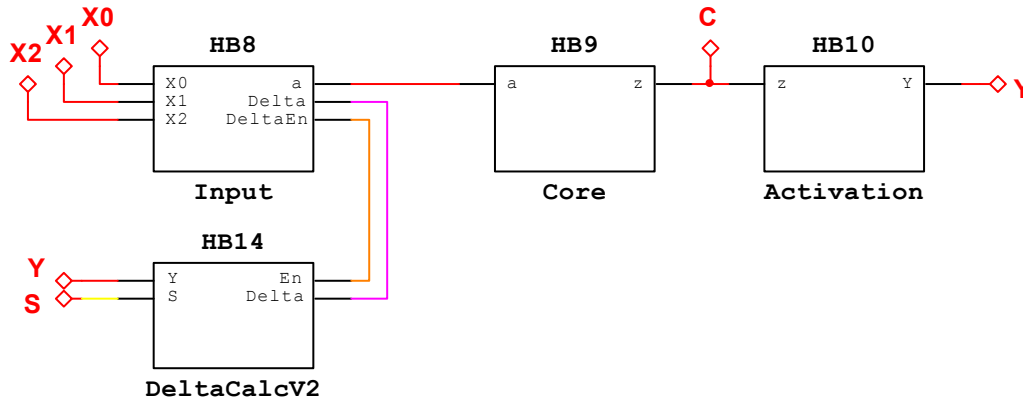


Abb 9: Detaillierte Blockartige Übersicht der Schaltung

Der Eingang (Input) verarbeitet die Spannung an den Eingängen X_0 bis X_2 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 S bei der Fehlerberechnung (DeltaCalcV2) entspricht dem Sollwert zu Trainingszeiten. Der Ausgang Y bei der Aktivierung wird zu jedem Zeitpunkt beim Training mit dem Sollwert mittels der Fehlerberechnung verarbeitet und dem 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.

B. Input

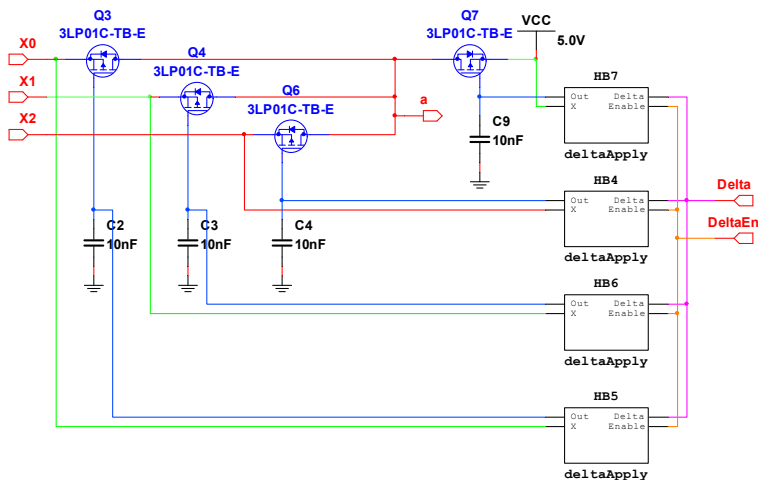


Abb 10: Schaltung für die Eingänge

Die Eingangsspannungen X_0 bis X_2 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.

Im mathematischen Sinne entspricht diese Teilschaltung dem Abschnitt

$$\left(\sum_i^n \frac{U_{X_i}(t)}{R_{T_i}(U_{C_{T_i}}(t))} \right) + I_{bias}(t)$$

Aus der Formel für die Vorwärtsrechnung.

Der $I_{bias}(t)$ kommt von dem Transistor Q_7 , der direkt an V_{cc} angeschlossen ist.

1. DeltaApply

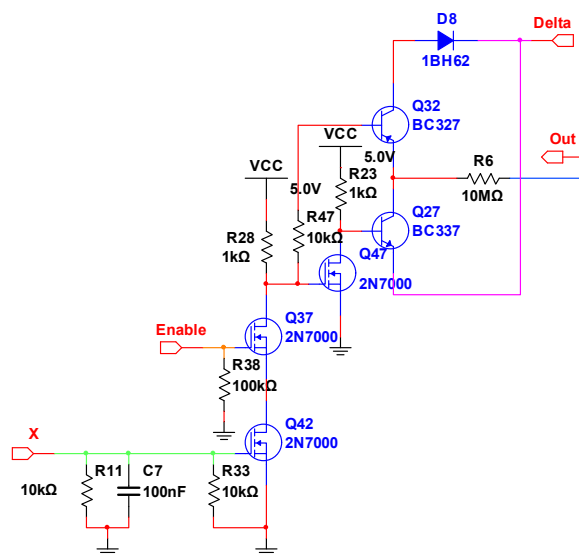


Abb 11: Schaltung für das Anpassen der Kondensatorspannung

Diese Teilschaltung ist für das Anpassen der Spannungen am Kondensator zuständig. Jeder Eingang besitzt eine solche Teilschaltung. Der Eingang Delta bestimmt, ob der Core-Kondensator geladen oder entladen wird. Enable schaltet mit der Spannung am Transistor den Rückrechnungsvorgang ein oder aus. Der Eingang X sagt der Schaltung, ob der Eingang der Gesamtschaltung überhaupt aktiv war.

Diese Teilschaltung ist die Umsetzung des Formelabschnittes $\frac{1}{V_{cc}^2 * R_{LR}} * U_{X_i}(t) * E * (D - U_{C_{T_i}})$, wobei E dem Enable Eingang, U_{X_i} dem Eingang X und D dem Delta Eingang entspricht.

C. Core

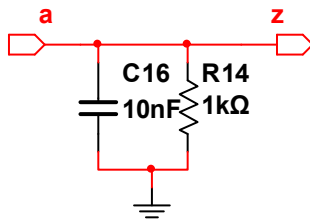


Abb 12: Schaltung des Kerns

Der Core besteht hauptsächlich aus einem Kondensator C , welcher über den Eingang a aufgeladen wird. Für den Fall, dass keine Signale über die Eingänge X_0 bis X_2 kommen, sorgt der Widerstand R dafür, dass sich der Kondensator langsam entlädt und nach einer gewissen Zeit der Kondensator wieder zurückgesetzt ist.

Die mathematische Darstellung der Teilschaltung ist $C \frac{dU_c(t)}{dt} = -\frac{U_c(t)}{R_0} + a$. Dabei entspricht a dem Ausgang der Input-Teilschaltung.

D. Activation

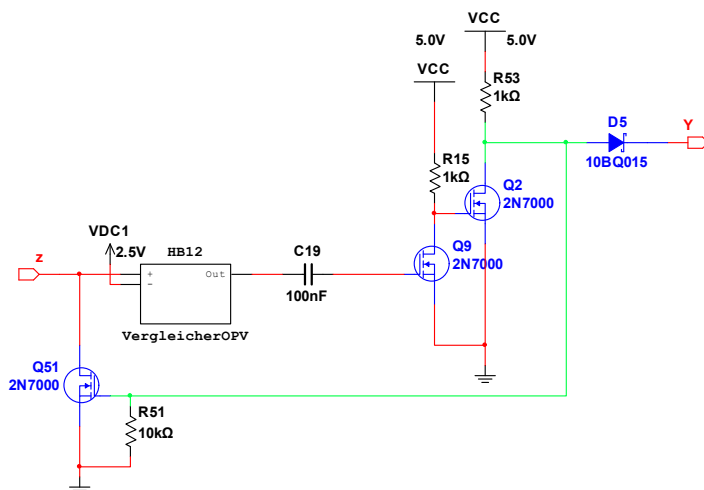


Abb 13: Aktivierungsschaltug

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 erreicht und auch mehr Strom liefern kann.

Dieses Ausgangssignal wird über die grüne Leiterbahn zurückgeführt und entlädt das Core (Kondensator) sehr schnell, wodurch dieser wieder 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, dass wenn der Ausgang des Vergleichers auf 0V ist, der Ausgang der Ganzen Schaltung einen «Tri-state»⁹ annimmt (Das bedeutet, dass kein Strom rückwärts in die Schaltung fließen kann).

1. **Vergleicher OPV**

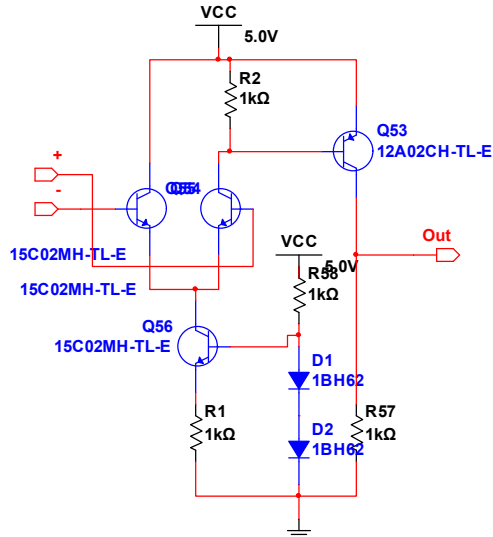


Abb 14: Primitiver Operations Verstärker

Dies ist eine primitive Schaltung eines Komparators bzw. Vergleichers, welcher in der Elektronik eine essentielle 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.

⁹ Tri-state ist eine Schaltung derer Ausgang so hochohmig ist, dass sie als nicht vorhanden zählt

E. Delta Calc

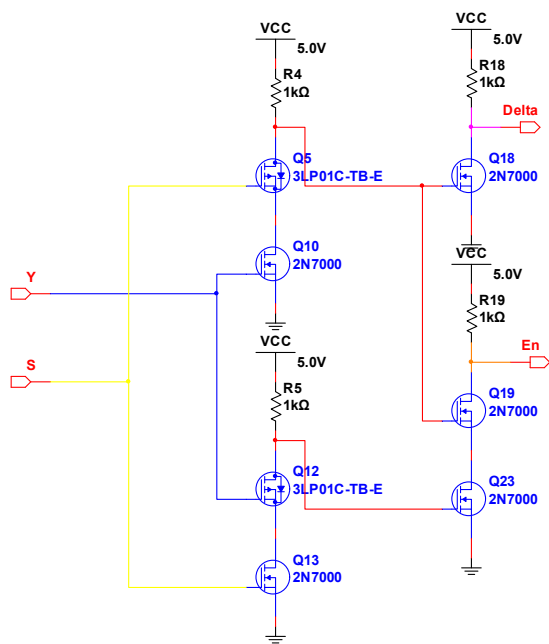


Abb 15: Schaltung für die Fehlerberechnung

Die Delta-Calculatation (Delta Calc) 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 X_0 bis X_2 geladen, oder entladen werden sollen. Der Ausgang «En» (Kurz für Enable) von diesem Abschnitt sagt, ob dieses Laden oder Entladen der Kondensatoren überhaupt durchgeführt werden soll oder nicht.

In Digitaltechnischer Sprache:

Der Ausgang «Delta» entspricht einem Und Gatter mit den Eingängen Y und \bar{S} . Der Ausgang «En» ist eine XNOR Verknüpfung von S und Y . Sprich wenn $S = Y$ dann sollen keine Änderungen an den Kondensatoren vorgenommen werden.

VII. Messungen

A. Messung des Eingangs

Um zu überprüfen ob der eingangs Stroms mit den Transistoren kontrolliert werden kann, haben wir eine Eingangs-Teilschaltung (Input) mit einem Kernstück mit dem Fräsbohrplotter erstellt. Es wurden wären der Messung zwei separate Spannungsquellen verwendet. Um die Spannungspotenziale aufeinander abzustimmen, wurde von allen Speisungen den Nullpunkt¹⁰ zusammengeschaltet. Eine Speisung wurde als die Eingangsspannung verwendet mit einem konstanten 5V Pegel. Die Zweite, wurde für die direkte Kontrolle über die Spannung des Kondensators¹¹ U_{C_T} , an den Kondensator direkt angelegt. Nach unserem Model sollte somit der Strom durch den Transistor mit der Spannung an dem Kondensator beeinflussbar sein.

U_{C_T} Eingang (in V)	U_C Kern (in V)	I Eingang (in mA)
4,0	0,000	0,01
3,9	0,000	0,01
3,8	0,000	0,01
3,7	0,031	0,01
3,6	0,154	0,03
3,5	0,584	0,08
3,4	1,019	0,30
3,3	1,948	0,93
3,2	3,654	2,67
3,1	3,967	2,97
3,0	4,911	3,82
2,9	4,939	3,84

Tabelle 4: Messergebnisse über den Eingangsstrom

Aus der Tabelle 4 lässt sich herauslesen, dass dies uns Erfolgreich gelungen ist. Es wird auch gut ersichtlich, dass der Strom Exponentiell ansteigt mit sinkender Spannung an U_{C_T} . Die Spannung an U_C zeigt die Spannung im Equilibrium, das ist der Punkt an dem I_{ein}

B. Messung der Aktivierung

Bei der Aktivierung (Activation) handelt es sich um eine Schaltung die, wenn der Referenzpunkt U_{ref} überschritten wird, einen Spike ausgibt. Wenn der Eingangsstrom konstant ist, fängt die Schaltung an, zu schwingen. Dies bedeutet, wenn eine Konstante DC Spannung angelegt wird, wird bei der Ausgabe ein Spiketrain¹² entstehen.

¹⁰ Die Spannung die 0 Volt entspricht

¹¹ Kondensator der sich unten an dem Transistor bei den Eingängen befindet.

¹² Eine Abfolge von einzelnen Spikes

Um dies zu messen, haben wir eine Teilschaltung dieser Aktivierung mit dem Fräsbohrplotter erstellt. Mit einem Eingang Z für die DC-Eingangsspannung und einem Ausgang Y als Messpunkt für das Ausgangssignal.



Abb 16: Messung der Activation Digital Oszi. (DC)

In der nebenstehenden Abbildung (Abb 16) ist das Ergebnis der Messung ersichtlich. Es ist sichtbar, dass die Eingangsspannung (der blaue Graph) immer einbricht, wenn die Schaltung einen Spike erzeugt. Diese Spikes (der gelbe Graph) haben jedoch einen etwas zu tiefen Höhepunkt. Daraus lässt sich schliessen, dass die Ausgangsspannung zu wenig verstärkt wird.

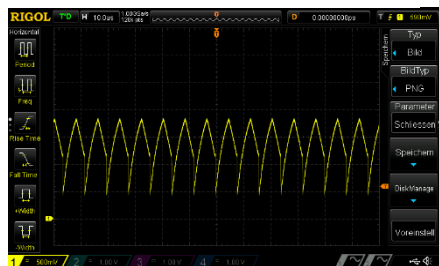


Abb 17: Messung der Spikes

In der Messung der Spikes (Abb 17), wo nur die Spikes an sich gemessen und ins Verhältnis gesetzt wurden, sieht man jedoch, dass schöne Spikes entstehen, die mit einer Frequenz von ca. 100kHz auftauchen, was nach Anforderungen von unserem Modell in einem akzeptablen

Bereich liegt. Dies könnte jedoch verbessert werden in dem man die einzelnen Bauteile etwas anpasst.

C. Messung der Delta Calculation

Wir wollten diese Teilschaltung auch verifizieren. Jedoch konnten wir diese Messung nicht durchführen, da ein Fehler beim Fräsbohrplotter aufgetaucht ist, welche zu einem Kurzschluss auf der Platine führte. Wir konnten den Fehler auch nach einigen Stunden nicht finden und mussten aufgrund des Zeitdrucks diese Messung überspringen. Wir konnten uns dies Erlauben, da diese Teilschaltung eine logische Schaltung ist und nach Vorgaben der Digitaltechnik funktionieren muss und in unserer Schaltung nicht als kritisches Element gilt.

D. Messung von DeltaApply

Diese Teilschaltung war für uns eine heiklere Schaltung, weil sie nur schlecht gemessen werden kann, ohne die Schaltung zu beeinflussen. Auch von dieser Schaltung wurde eine Teilschaltung mit dem Fräsbohrplotter erstellt und ausgemessen.

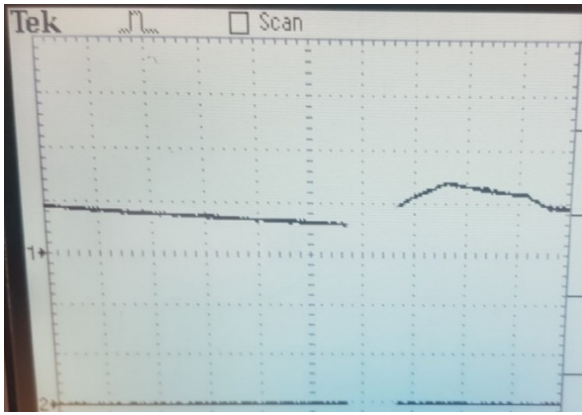


Abb 18: Messung von DeltaApply

Die obige Abbildung (Abb 18) zeigt den Verlauf der Kondensatorspannung, der mittels der DeltaApply Schaltung geladen und Entladen wurde. Die erste Strecke zeigt, wie der Kondensator kontrolliert entladen werden konnte. Der von links nach rechts Steigende Abschnitt zeigt ein Ladevorgang. Am obigen Wendepunkt der Spannungskurve, wurde der Kondensator weder geladen, noch Entladen. Es ist jedoch trotzdem eine kleine Entladung zu sehen, weil die Spannung abfällt. Dies passiert zu einem Anteil, weil der Kondensator sich über das Messgerät entladen kann, was das Ergebnis etwas verfälscht. Zum anderen ist der Kondensator nicht perfekt isoliert und kann sich über die Umgebungsluft Richtung Erde entladen. Dies hatten wir jedoch erwartet, zumal diese Teilschaltungsplatine von schlechter Qualität war und viele Möglichkeiten zum entladen anbot.

VIII. Fazit

Im Grunde sind die Messungen der Teilschaltungen positiv ausgefallen. Man kann aber wegen der nicht funktionierenden Gesamtschaltung keine eindeutigen Schlüsse ziehen.

Wir schafften es schlussendlich nicht, eine funktionierende Endversion unseres Prints zu produzieren. Das hat mehrere Gründe. Zum einen hätten wir uns grundsätzlich mehr Zeit einrechnen müssen für die Phase des Aufbaus und der Inbetriebnahme der Schaltung. Zum anderen ist unsere Printbestellung nicht wie geplant eingetroffen, was für uns massiv mehr Arbeit bedeutete, da wir die Endversion schlussendlich selbst ausfräsen mussten und unsere Fräsmaschine einige Teile der Schaltung sehr störungsanfällig konstruierte. Obwohl wir das gesamte Wochenende vor der BMA-Abgabe an dem Print gearbeitet haben, reichte die Zeit am Ende doch nicht. Mit einer Woche mehr Zeit, wäre all dies zu vermeiden gewesen. Dann hätten wir nämlich den «qualitativ hochwertigen» Print aus dem Ausland in Betrieb nehmen können.

Dies werden wir aus Interesse trotzdem noch tun. Ausserdem, sind wir davon überzeugt, dass man unsere Fragestellung, ob die Realisierung eines Neurons, in Form einer elektrischen Schaltung, möglich ist mit Ja beantworten kann. Grund dafür ist, dass bis auf die

Teilschaltung «delta Calc» alle bereits funktioniert haben, wie sie im Kapitel Messungen gesehen haben. Das «delta Calc» ist aber im Grunde nur eine Logikschaltung und man kann davon ausgehen, dass diese auch funktionieren würde.

Um unsere Schaltung vor dem Aufbau auszutesten, benutzten wir ein Simulationstool, namens Multisim. Mit diesem konnten wir unsere Theorien und Berechnungen überprüfen. In dieser Simulation funktionierte unsere Schaltung. Ein weiteres Indiz dafür, dass unsere Schaltung theoretisch hätte funktionieren können.

Wie in der Einleitung erwähnt, wollten wir unsere Fragestellung mit einem funktionierenden Endprodukt beantworten. Wegen der fehlenden Zeit, sehen wir aber das nichtfunktionieren der Schaltung nicht als repräsentativ an. Wir beantworten stattdessen die Fragestellung, aufgrund der beiden oben genannten Argumente, mit Ja.

Wir haben gelernt, dass es wichtig ist, bei einer Arbeit solchen Ausmasses früher anzufangen Prototypen zu realisieren. Die digitale Version im Simulationstool reicht natürlich nicht. Wir haben leider erst 3 Wochen vor der Abgabe angefangen, am ersten physischen Prototypen zu arbeiten.

Weiterführend könnte man sich die Frage stellen, ob mit mehreren solchen Neuronen ein selbstlernendes Netzwerk erstellt werden kann, indem man Ein und Ausgänge zusammenhängt. Unser Hirn ist ja im Prinzip auch nur ein Haufen zusammengeschalteter Neuronen. Diese Fragen finden ich und Tendai so interessant, dass wir uns vermutlich weiter damit befassen werden, nach der BMA.

IX. Formelverzeichnis

Formel 1: Berechnung der Kernkondensatorspannung.....	13
Formel 2: Funktion der Ausgabe	13
Formel 3: Berechnung des Fehlers	14

X. Abkürzungsverzeichnis

usw. = und so weiter

bzw. = beziehungsweise

DC = direct current / Gleichspannung

KO = Oszilloskop

d.h. = das heisst

XI. Abbildungsverzeichnis

Abb 1: Biologisches Neuron (Wikipedia, 2020)	6
Abb 2: Aktionspotenzial im Axon (Wikipedia, 2020)	7
Abb 3: Chart of Neural Networks (Tch, 2017).....	8
Abb 4: UND Gatter (Stefan506, Wikipedia, 2005)	9
Abb 5: ODER Gatter (Stefan506, Wikipedia, 2005).....	10
Abb 6: NICHT Gatter (Stefan506, Wikipedia, 2005).....	10
Abb 7: «The Dynamics of the LIF unit» (Ponulak & Kasinski, 2011)	12
Abb 8: Blockartige Übersicht der Schaltung	15
Abb 9: Detaillierte Blockartige Übersicht der Schaltung	16
Abb 10: Schaltung für die Eingänge	16
Abb 11: Schaltung für das Anpassen der Kondensatorspannung	17
Abb 12: Schaltung des Kerns	18
Abb 13: Aktivierungsschaltug	18
Abb 14: Primitiver Operations Verstärker	19
Abb 15: Schaltung für die Fehlerberechnung	20
Abb 16: Messung der Activation Digital Oszi. (DC)	22
Abb 17: Messung der Spikes.....	22
Abb 18: Messung von DeltaApply.....	23

XII. Literaturverzeichnis

Ponulak, F., & Kasinski, A. (1. Januar 2011). Von

<https://www.ane.pl/cookie?page=linkout.php?pii=7146> abgerufen

Stefan506. (14. Dezember 2005). *Wikipedia*. Von <https://de.m.wikipedia.org/wiki/Datei:Logic-gate-or-us.png> abgerufen

Stefan506. (14. Dezember 2005). *Wikipedia*. Von <https://de.wikipedia.org/wiki/Datei:Logic-gate-inv-iec.png> abgerufen

Stefan506. (14. Dezember 2005). *Wikipedia*. Von <https://de.wikipedia.org/wiki/Datei:Logic-gate-and-us.png> abgerufen

Tch, A. (4. August 2017). *towardsdatascience.com*. Von <https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464> abgerufen

Wikipedia. (3. Januar 2020). *Wikipedia/Neuron*. Von

<https://de.wikipedia.org/w/index.php?title=Nervenzelle&oldid=195474238> abgerufen

XIII. Anhang

- A. *Layout/Bestückungsplan von EAGLE***
- B. *Schema von EAGLE***
- C. *Stückliste***

XIV. Deklaration

Hiermit erklären wir, dass wir die vorliegende interdisziplinäre Arbeit selbstständig und in eigener Sprache verfasst und keine anderen als die angegebenen Hilfsmittel verwendet haben. Wörtliche und sinngemässe Übernahmen aus anderen Quellen wie Büchern, Zeitschriften, Internet, Interviews usw. sind als solche kenntlich gemacht.

Ort:.....

Datum:.....

Unterschrift:.....

Unterschrift:.....