Emmy -Noether-Gymnasium

Benjamin Jank

Im Fach Informatik und Biologie (Bezugsfach)

Betreut durch Herr Ostrzinski

Berlin, d

Inwieweit kann der Arbeitsalltag von Mediziner:innen mithilfe von faltenden neuronalen Netzen effektiver gestaltet werden – am Beispiel der Diagnose von Hirntumoren?

# Gliederung

* Motive für die Wahl der Leitfrage
  1. Motive für die Wahl des Themas der künstlichen neuronalen Netze
  2. Motive dafür, dass das Thema am Beispiel der Diagnose von Hirntumoren in der Medizin mithilfe von faltenden neuronalen Netzen abgehandelt wird
* Erläuterung des fachübergreifenden Aspekts
* Was ich in der Präsentation darstellen möchte
* Arbeitsprozess
  1. Verwendete Mittel und Methoden
  2. Reflexion des bisherigen Arbeitsweges und der bisherigen Ergebnisse
  3. Aufgetretene Probleme
  4. Was ich retrospektiv anders gemacht hätte
  5. Was ich bisher gelernt habe
* Verwendung von Quellen
  1. Top Drei und Erläuterung
* Anhang
  1. Perspektivische Zeitplanung (bereits mit Prüfungsantrag abgegeben)
  2. Tätigkeitsübersicht
  3. Übersicht über Beratungsgespräche
  4. Vollständiges Literatur- und Quellenverzeichnis
  5. Selbstständigkeitserklärung

# Motive für die Wahl der Leitfrage

1. **Motive für die Wahl des Themas der künstlichen neuronalen Netze**

Heute ist das Thema künstliche neuronale Netze aktueller denn je. Ein Großteil der Services und Dienstleistungen von großen IT-Konzernen wie Google oder Microsoft, z.B. personalisierte Feeds in sozialen Netzwerken, Speech-to-Text-Anwendungen oder Chatbots, basiert im Kern auf künstlichen neuronalen Netzen. Es lässt sich somit feststellen, dass neuronale Netze aus der Wirtschaft und letztendlich unserem Alltag kaum wegzudenken sind.  
   
Doch auch gesellschaftlich sind künstliche neuronale Netze relevant: Beispielsweise benötigen Unternehmen wie Google für das Optimieren von künstlichen neuronalen Netzen Unmengen an Daten, die durch ständige Überwachung des Nutzers generiert werden. Die Problematik dieser akuten Überwachung mit dem Ziel der Optimierung neuronaler Netze und letztlich auch der Konsum- und Gewinnmaximierung wird in vielen gesellschaftspolitischen Fachkreisen diskutiert. Ein anderes Beispiel wäre die Diskussion über die Gefahren von vorurteilsbehafteten neuronalen Netzen, die Minderheiten (z.B. PoC) aufgrund unausgewogener Trainingsdatensätze benachteiligen. Solche oder ähnliche Fragen machen künstliche neuronale Netze nicht nur wirtschaftlich, sondern auch gesellschaftlich aktuell zu einem sehr präsenten Thema.

Diese gesellschaftliche und wirtschaftliche Relevanz macht faltende neuronale Netze für mich zu einem sehr spannenden, aktuellen Thema und ist somit passend für eine Präsentation im Rahmen der 5. Prüfungskomponente.

1. **Motive dafür, dass das Thema am Beispiel der Diagnose von Hirntumoren in der Medizin mithilfe von faltenden neuronalen Netzen abgehandelt wird**

Ich war mir also sehr früh schon sicher, dass ich in meiner 5. PK. Neuronale Netze behandeln möchte. Doch musste ich mich in meinem Thema noch etwas einschränken. Zufällig bin ich dann auf eine [Publikation](https://www.hindawi.com/journals/cin/2021/2392395/) gestoßen, welche die Rolle von sogenannten faltenden neuronalen Netzen in der Medizin behandelt. Konkret ging es in dem Artikel darum, wie faltende neuronale Netze auf Basis von MRT-Scans vom Gehirn Aussagen darüber treffen konnten, mit welcher Wahrscheinlichkeit beim Patienten eine Krebserkrankung vorliegt.

Obgleich ich nur einen Bruchteil des Gelesenen verstand, klang das Ganze ziemlich interessant. Deshalb habe ich dann meinen Vater, der früher als Arzt an Krebszellen geforscht hat, nach seiner Meinung zu dem Thema gefragt. Er konnte mir sagen, dass die Thematik in der Medizin sehr aktuell ist, faltende neuronale Netze aufgrund von noch zu niedriger Zuverlässigkeit und ethischen Problematiken jedoch auch hier noch umstritten sind. Und genau diese Kontroversität hat mich dann bekräftigt, mich im Rahmen der 5. PK mit neuronalen Netzen als Hilfsmittel in der Medizin zu beschäftigen.

Ein großer Anwendungsbereich faltender neuronaler Netze in der Medizin ist die Unterstützung bei der Diagnose von Gehirntumoren. Hierbei wird auf Basis von Bildgebung des Gehirns (z.B. MRT) die Wahrscheinlichkeit einer Tumor-Erkrankung vorhergesagt. Es bot sich an, mein Vortragsthema beispielhaft anhand dieser Anwendung abzuhandeln.

# Erläuterung des fachübergreifenden Aspekts

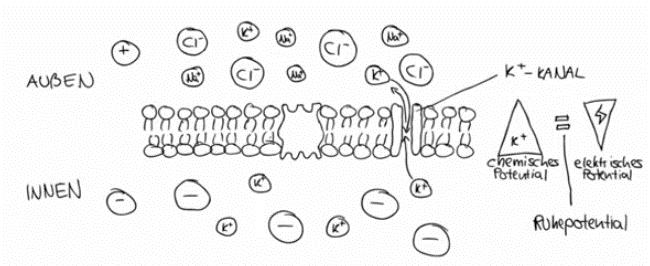
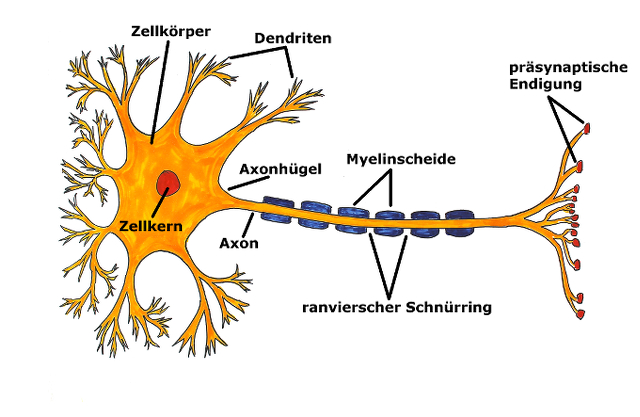


Abb. 1: Aufbau eines Neurons / Nervenzelle Abb. 2: Eine Axonmembran

Um zu verstehen, wie künstliche neuronale Netze mit tatsächlichen neuronalen Netzen verknüpft sind, empfiehlt es sich, zunächst einen Blick auf die Arbeitsweise von Neuronen zu werfen. Die Zellmembran trennt extrazellulären von intrazellulärem Raum. Im extrazellulären Raum liegen negative Chlorid-Ionen (Cl-) und positive Natrium-Ionen (Na+) vor, im intrazellulären Raum negative Protein-Ionen (-) und positive Kalium-Ionen (K+). Die Membran ist nur für Kalium-Ionen durchlässig. Um sich gleichmäßig im Raum zu verteilen, diffundieren die Kalium-Ionen nun zunehmend in den extrazellulären Raum. Mit zunehmender Menge diffundierter Kalium-Ionen baut sich eine elektrische Spannung zwischen intra- und extrazellulärem Raum (aufgrund der zunehmend ungleichen Ladungsverteilung) auf, welche die Kalium-Ionen wiederum zunehmend an der Diffusion hindert. Ab einem bestimmten Punkt steht der elektrischer Gradient mit dem chemischen Gradienten im Gleichgewicht, wodurch die Kalium-Ionen die Membran nicht mehr durchqueren. Die elektrische Spannung zwischen extrazellulärem und intrazellulärem Raum (ca. -70 mV) zu diesem Zeitpunkt nennt man Ruhepotenzial.

Entsteht unter bestimmten Umständen nun im Axonhügel ein neues Aktionspotenzial, so öffnen sich am Axonhügel spannungsempfindliche Natrium-Kanäle, die eine Diffusion der Natrium-Ionen nach innen ermöglichen. Dies hat zur Folge, dass die elektrische Spannung von -70 mV innerhalb ansteigt. Überschreitet diese einen Schwellenwert von -50 mV, öffnen sich neue weiter entlang des Axons liegende Natrium-Kanäle. Dadurch diffundieren mehr Natriumionen weiter längs des Axons in die Zelle, die Spannung innerhalb steigt und neue Natriumkanäle noch weiter entlang des Axons liegend werden geöffnet. So bewegt sich ein elektrisches Potenzial (Aktionspotenzial / AP) am Axon bis zu den präsynaptischen Endigungen (Synapsen).

Die Synapsen sind eng mit den Dendriten (Teil des Somas) nachgeschalteter Neuronen verbunden. Wenn ein Aktionspotenzial nun in einer Synapse angelangt, sorgt es dort durch an dieser Stelle irrelevante Vorgänge dafür, dass ein Neurotransmitter in den extrazellulären Raum ausgeschüttet wird. Die Neurotransmitter öffnen dann temporär Ionenkanäle an den Dendriten, die dafür sorgen, dass negative oder positive Ionen in das nachfolgende Soma diffundieren. Synapsen sind neurotransmitterspezifisch. Die Synapsen, die bei der Ankunft eines APs die Neurotransmitter zum Öffnen positiver Ionen ausschütten, nennt man exzitatorische Synapsen. Den Effekt des Anstiegs der Spannung im Neuron durch Einstrom positiver Ionen nennt man exzitatorisches postsynaptisches Potenzial (EPSP). Synapsen, die die Ausschüttung von Neurotransmittern zur Öffnung negativer Ionen bewirken, nennt man inhibitorische Synapsen. Wegen des Einstroms negativer Ionen sinkt die Spannung im Soma, exzitatorische Synapsen bewirken somit ein inhibitorisches postsynaptisches Potenzial. Die Menge des ausgeschütteten Neurotransmitters ist außerdem entscheidend für die Anzahl geöffneter Ionenkanäle und somit auch für die Stärke des postsynaptischen Potenzials (EPSP oder IPSP). Die Anzahl an Neurotransmitter-Molekülen hängt hierbei von der Größe der Synapse ab.

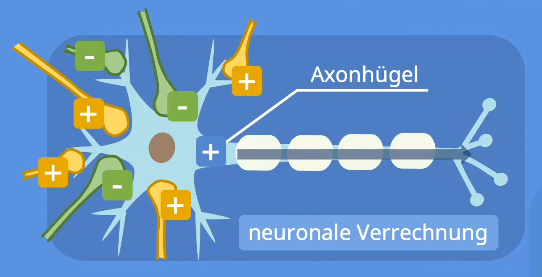
Gerade im zentralen Nervensystem (Gehirn, Rückenmark) sind die Neuronen, im Gegensatz zum peripheren Nervensystem (v.a. periphere Nervenbahnen), jedoch nicht in einer einfachen Reihe geschaltet. Vielmehr bilden die Neuronen ein komplexes Netz: Viele Synapsen unterschiedlicher Neuronen münden hierbei in ein einziges Soma. Im Soma ist nun die Bilanz aus IPSPs (grünes -) und EPSPs (gelbes +) aller Synapsen ausschlaggebend für die Gesamtspannung im Soma (blaues +). Diese Verrechnung nennt man synaptische Integration. Überschreitet die Gesamtspannung einen bestimmten Schwellenwert, entsteht am Ende des Somas (Axonhügel) ein neues Aktionspotenzial, das nun an nachfolgende Neuronen weitergeleitet wird (s. zweiter Absatz). Nur durch ein solch komplexes Netz können Eingangssignale zu Ausgangssignalen verarbeitet werden.

Abb. 3: Verschaltung von inhibitorischen und erregenden Synapsen

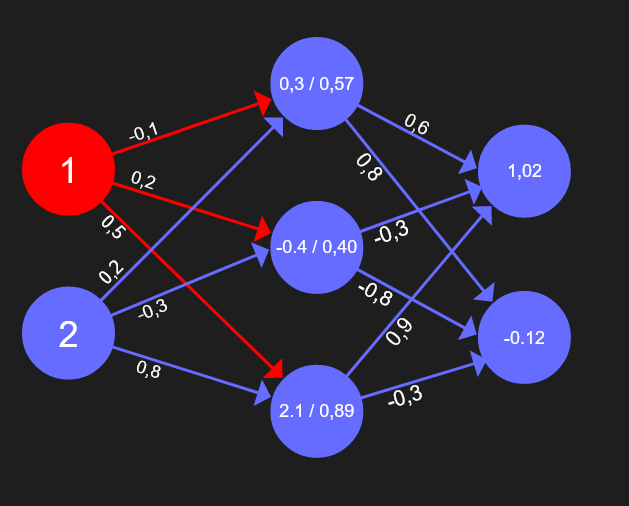
Und genau dieses Prinzip der Reizverarbeitung nehmen sich künstliche neuronale Netze zum Vorbild. In der nebenstehenden Abbildung sieht man eine beispielhafte Verschaltung künstlicher Neuronen. Grundsätzlich sind die Neuronen hierbei in Schichten angeordnet. Ein Neuron entspricht in der Abbildung dem rot markierten Bereich. Das Soma (auch node genannt) (roter Kreis) ist hierbei über Synapsen (auch weights genannt) (rote Pfeile) mit allen Somata nachfolgender Neuronen verschaltet. Die elektrische Spannung in einem Soma wird hierbei durch einen Float-Wert dargestellt. Die Wirkung einer Synapse wird ebenfalls durch einen Float-Wert repräsentiert. Negative Zahlen entsprechen hierbei einem inhibitorischen, positive Zahlen hingegen einem exzitatorischen postsynaptischen Potenzial. Die Größe des Werts repräsentiert die Größe der Synapse, also die Stärke des postsynaptischen Potenzials.

Abb. 4: Reizweiterleitung in einem künstlichen neuronalen Netz

Bei der Erregungsweiterleitung wird nun in jedes Soma der Eingabeschicht ein Float-Wert geladen. Zusammen codieren diese Zahlen eine Information (z.B. deutsches Wort). Die Spannung in einem Soma ergibt sich nun aus der Summe aller Produkte vom Float-Wert aller verbundenen, vorgeschalteten Synapsen und der Spannung ihres Somas. Dieser Prozess ist vergleichbar mit der synaptischen Integration. Nach diesem Prinzip berechnet sich die Spannung des ersten Somas in der zweiten Schicht im abgebildeten künstlichen neuronalen Netz beispielsweise wie folgt:

Aus Gründen, die ich in meiner Präsentation darstelle, wird die Spannung eines Somas nun noch durch eine Aktivierungsfunktion geführt. Im Beispiel ist dies eine Sigmoid-Funktion, welche für x=0,3 0,57 zurückgibt. Die Berechnung der Spannung der Ausgabe-Somata funktioniert analog. Auf Basis dieses Prinzips werden also Eingangssignale gemäß einer festgelegten Verschaltung simulierter Neuronen zu einem oder mehreren Ausgangssignalen verrechnet.

Programmintern werden die Spannungen der Somata durch n-dimensionale Vektoren und die Synapsen durch Matrizen repräsentiert. Die Reizweiterleitung wird mithilfe von Operationen der linearen Algebra realisiert. Die in der Abbildung dargestellte Reizweiterleitung funktioniert nach diesem Prinzip intern wie folgt:

Somata Synapsen neue Somata

Somata Synapsen neue Somata

Außerdem ist die Struktur des Gehirns, anders als bei einem Computer-Schaltkreis beispielsweise, dynamisch. Diesen Effekt nennt man neuronale Plastizität. Die Funktionsweise der Neuronen (funktionale Plastizität) und Anzahl und Verschaltung (strukturelle neuronale Plastizität) der Neuronen verändert sich je nach eingehenden Signalen, was wiederum die Weiterleitung neuer Signale verändert. Vereinfacht betrachtet versucht das zentrale Nervensystem durch diese strukturelle Veränderung, sich den gewünschten Ausgangssignalen für bestimmte Eingangssignale anzunähern. Dies ist für biologische Lernprozesse unabdinglich.

Und auch künstliche neuronale Netze lernen auf Basis einer solchen strukturellen Dynamik. Um dem künstlichen neuronalen Netz beizubringen, welche Ausgangssignale zu bestimmten Eingangssignalen passen, werden die künstlichen Neuronen in ihrer Wirkungsweise gezielt angepasst. In mehreren Trainingsdurchgängen (*epochs*) werden dem neuronalen Netz gewünschte Ausgangssignale (*truth values)* passend zu bestimmten Eingangssignalen (*inputs*) präsentiert. Das neuronale Netz versucht nun, die Struktur einiger Neuronen so zu verändern, dass sich das tatsächliche Ausgangssignal, das auf den *inputs* beruht, den *truth values* annähert. Der Prozess ist also vergleichbar mit funktionaler neuronaler Plastizität. Wie diese gezielte Anpassung im Computer realisiert wird, werde ich in meiner Präsentation darlegen.

Die strukturelle neuronale Plastizität in biologischen neuronalen Netzen ist hingegen noch zu unerforscht, als dass diese durch Programme nachgeahmt werden könnten.

Ein faltendes neuronales Netz (engl. Convolutional Neural Network) zeichnet sich nun dadurch aus, dass vor ein herkömmliches Netz ein oder mehrere sogenannte faltende Schichten (Convolutional Layers) geschaltet werden, die sehr große Matrizen (z.B. HD-Bilder) ohne Informationsverlust zu kleineren Matrizen komprimieren. So kann die Anzahl der Neuronen der Eingangsschicht minimiert und somit Speicherplatz und Rechenlast während des Trainings reduziert werden.

Es lässt sich zusammenfassend feststellen, dass sich faltende neuronale Netze in ihrer Funktionsweise stark an der Funktionsweise tatsächlicher neuronaler Netze orientieren. Diese Verknüpfung macht deutlich, was neuronale Netze als Werkzeug der Informatik mit Biologischem verbindet.

# Was ich in der Präsentation darstellen möchte

Zunächst möchte ich in meiner Präsentation zu Beginn kurz nochmal auf den Zusammenhang zwischen biologischen neuronalen Netzen und künstlichen neuronalen Netzen darstellen. Darauffolgend werde ich die grundlegende Funktionsweise von faltenden neuronalen Netze darstellen (v.a. Algorithmus für das Training) und meine Implementierung eines neuronalen Netzes erläutern. Danach soll die Leitfrage insbesondere mit Blick auf die Grenzen faltender neuronaler Netze und ethischen Fragestellungen bearbeitet werden.

Ohne meiner Präsentation zu viel vorwegzunehmen, möchte ich im Folgenden zudem kurz meine Position zur Leitfrage umreißen.  
Während meines Arbeitsweges gelangte ich zunehmend zu der Ansicht, dass faltende neuronale Netze ausschließlich als medizinisches Hilfsmittel bei der Diagnose von Krankheiten auf Basis von Bildgebung eingesetzt werden. Die kontrollierende und entscheidende Instanz sollte nach wie vor der Arzt sein. Würde man sich bei Diagnosen auf Basis von Bildgebung jedoch gänzlich auf faltende neuronale Netze verlassen, so würde das grundlegende ethische Fragestellungen aufwerfen.   
Darüber hinaus arbeiten faltende neuronale Netze bisher auch noch nicht zuverlässig genug, als dass sie Krankheiten mit hundertprozentiger Sicherheit erkennen können.

# Arbeitsprozess

1. **Verwendete Mittel und Methoden**

Es war von Anfang an klar, dass ich im Rahmen meiner 5. PK auch ein faltendes neuronales Netz programmieren möchte. Somit musste ich mich zuallererst für eine passende Programmiersprache entscheiden. Meine Wahl viel schlussendlich auf Python. Diese hat den Vorteil, dass sie im Gegensatz zu anderen Programmiersprachen eine einfache und kompakte Syntax aufweist. Das macht Python-Quellcodes übersichtlich. Zudem übernimmt Python komplexe Aufgaben (z.B. Speicherverwaltung), um die sich der Programmierer bei anderen Programmiersprachen (z.B. C oder Assembler) selbst kümmern muss. Außerdem habe ich mich im Rahmen des Informatikunterrichts und meiner Freizeit bereits mit Python beschäftigt, wodurch ich einige Vorkenntnisse darin habe. So kann ich mich auf mein eigentliches Thema, was für mich bereits ausreichend kompliziert ist, konzentrieren, ohne mich mit unwesentlichen Belangen beschäftigen zu müssen. Ein Nachteil von Python ist jedoch, dass es eine interpretierte anstelle einer kompilierten Sprache ist. Deshalb laufen Python-Programme langsamer, was sich letztendlich auf die Trainingszeit meines künstlichen neuronalen Netzes auswirken wird. Im Vergleich zu den zahlreichen Vorteilen ist dieser Nachteil jedoch unwesentlich.

Programmiert habe ich nahezu alles in der Jupyter-Notebook-Umgebung. In dieser kann Programmcode zellenweise ausgeführt werden, ohne dass das gesamte Programm ausgeführt werden muss. Dies macht das Konzipieren und Testen bestimmter Programmteile und Funktionen wesentlich leichter.

1. **Reflexion des bisherigen Arbeitsweges und der bisherigen Ergebnisse**

Wie aus der perspektivischen Zeitplanung hervorgeht, war mein Plan, zunächst beispielhaft ein faltendes neuronales Netz zu programmieren. Da ich zunächst davon ausging, dass sich meine gesamte Präsentation und mein methodisches Arbeiten um das Programmierte drehen würde, wollte ich zunächst das Programm gänzlich beenden, um dann erst mit der Arbeit an meiner Präsentation zu beginnen. Aus diesem Grund programmierte ich also einfach los (ca. Mitte November) und wollte Arbeitsergebnisse schaffen, um schnellstmöglich mit der Arbeit an der Präsentation beginnen zu können. Dies hatte zur Folge, dass ich mich vor dem Programmieren nicht ausreichend tief in die Thematik einlas, was zu einer Art Halbwissen führte. Erst während des Arbeitsprozesses las ich mich immer mehr in die Thematik ein. Dieses Halbwissen machte das Programmieren zu Beginn zu einem äußerst langwierigen und unproduktiven Prozess, in dem viele Probleme überwunden werden mussten. Und da ich stets an dem Kredo festhielt, zunächst das Programm fertigzustellen und dann erst mit der methodischen Arbeit (Exposé, Präsentation) zu beginnen, verschob sich die Arbeit an Präsentation und Exposé immer weiter nach hinten. Das brachte mich allmählich in Zeitnot. Vor rund einem Monat (ca. Anfang Februar) wurde ich mir dann aber zunehmend meines planerischen Versagens bewusst und begann parallel zum Programmieren die Arbeit an Präsentation und Exposé.

Trotz des langwierigen Prozesses kann ich sagen, dass ich beim Programmieren des faltenden neuronalen Netzes meinen Erwartungen an mich selbst gerecht werden konnte. Aktuell bewertet mein faltendes neuronales Netz ca. 85% aller ihm vorgeführten Bilder richtig. Professionelle faltende neuronale Netze kommen i.d.R. auf 95% bis 99%. Das macht mich, offen gesagt, ein wenig stolz. Nichtsdestotrotz bin ich mir darüber bewusst, dass 85% noch kein optimales Ergebnis ist und es noch einiges an Optimierung bedarf, um das Ergebnis zu verbessern.

1. **Aufgetretene Probleme**

Ein Problem, das während des gesamten Programmierungsprozesses sehr präsent war, war das Problem der Laufzeitoptimierung. Diese Problematik ist insbesondere bei der Entwicklung faltenden neuronaler Netze sehr präsent. Grund dafür ist, dass die verschiedenen Operationen der linearen Algebra (Matrixoperationen) sehr leistungshungrig sind, insbesondere wenn es darum geht, sehr große Matrizen mit vielen Dimensionen zu verarbeiten. Da es in meinem Projekt im Kern darum geht, durch große Matrizen repräsentierte MRT-Scans mit Matrixoperationen zu verarbeiten, wurde auch ich mit dem Problem der Laufzeitoptimierung konfrontiert:   
Mein ursprünglicher Ansatz bestand darin, alle Matrixoperationen selbst zu implementieren. Dies realisierte ich zunächst auf Basis von serieller Bearbeitung mithilfe von verschachtelten For-Schleifen. Beim Ausführen des Programmes stellte ich eine sehr hohe Laufzeit der einzelnen Trainingsiterationen (epochs) fest, was auf die exponentiell skalierende Laufzeit von verschachtelten For-Schleifen zurückzuführen ist. Angesichts dieses Problems recherchierte ich nach Optimierungsmöglichkeiten. Ich stieß auf einen Ansatz, der sich Shared Libraries nennt. Hierbei werden leistungshungrige Funktionen in die deutlich schnellere, kompilierte Programmiersprache C ausgelagert. Diese ausgelagerten Funktionen können dann mithilfe von diversen Modulen von Python aus aufgerufen werden. So implementierte ich den leistungshungrigsten Teil der Matrixoperationen auf dem Shared-Libraries-Ansatz beruhend in C. Es sollte sich jedoch herausstellen, dass dieser Ansatz nach wie vor nicht schnell genug ist. Die einzige Lösung bestand jetzt nur noch darin, die Matrixoperationen zu parallelisieren. Die Implementierung von parallelisierten Matrixoperationen stand jedoch nicht zur Option, da dies ein großes, grundlegendes Verständnis von Problemen der technischen Informatik erfordert. Da mir die Kenntnisse darüber fehlen, war ich gezwungen, zumindest teilweise auf parallelisierte Implementierungen aus dem Numpy- und Computervision-Modul zurückzugreifen. Ich hätte von Anfang an auch ausschließlich parallelisierte Implementierung von Matrixoperationen aus vorgefertigten Python-Libraries (z.B. Numpy) benutzen könne. Dadurch hätte ich den Großteil der für die Entwicklung von seriellen Matrixoperationen aufgewandten Zeit einsparen können. Die Verwendung externer Module wollte ich jedoch stets möglichst klein halten, da alles andere nicht gerade dem Prinzip einer „Eigenleistung im Rahmen der 5. PK“ entspricht.

Ein weiteres Problem stellte für mich grundsätzlich die Mathematik dar, auf der faltende neuronale Netze beruhen. So habe ich mich in meiner bisherigen Schullaufbahn bspw. an keiner Stelle tiefgreifend mit Matrizen beschäftigt. Auch Konzepte wie partielle Ableitungen waren grundlegend neu für mich. Deshalb musste ich mich in einige mir fremde mathematische Themenkomplexe von Grund auf einarbeiten, um die Funktionsweise von faltenden neuronalen Netzen verstehen zu können.

Neben diesen zwei grundlegenden Problemen sah ich mich während meines Arbeitsweges nur mit unwesentlichen, kleinen Problemen (kleinere Fehlermeldung während des Programmierens oder Rechenfehler) konfrontiert.

1. **Was ich retrospektiv anders gemacht hätte**

Zum einen hätte ich mich, wie aus der vorangegangenen Darstellung hervorgeht, vor Beginn des Programmierens tiefgründiger in die Thematik einlesen sollen, um Halbwissen zu vermeiden. Mit entsprechendem Ausganswissen hätte ich mich nicht an Dingen aufgehalten, die letztendlich sowieso keinen Sinn ergeben. Bestes Beispiel hierfür ist der Versuch, ein faltendes neuronales Netz mit seriellen Prozessen zu implementieren. Mit Vorkenntnissen hätte ich gewusst, dass serielle Prozesse bei faltenden neuronalen Netzen nicht zielführend sind. So hätte ich mir die langwierige Arbeit an einer seriellen Implementierung in Python C und so eine Menge Zeit gespart. Mit Vorinformation wäre der Programmierungsprozess also deutlich effizienter gewesen.

Im Nachhinein kann ich auch sagen, dass das Kredo, zunächst alle Programmieraufgaben abzuschließen, bevor ich mit Präsentation und Exposé beginne, mir nicht geholfen hat. Letztendlich stellt sich auch heraus, dass Exposé bzw. Präsentation gar nicht so stark an meine Implementierung gebunden sind, weshalb des Kredo keinen Vorteil bringt. Vielmehr brachte mich das Aufschieben des Exposés und der Präsentation angesichts der Abgabetermine sogar in Zeitnot. Retrospektiv kann ich also sagen, dass eine gleichzeitige Arbeit an Implementierung und Präsentation bzw. Exposé zeittechnisch besser gewesen wäre, zumal das Programm ohnehin vier Wochen später als das Exposé, nämlich erst zur tatsächlichen Präsentation, fertig sein muss.

1. **Was ich bisher gelernt habe**

In erster Linie habe ich während meiner Arbeit wohl methodisch dazu gelernt. Ich weiß nun beispielsweise, wie wichtig es ist, sich zu einem Thema ausreichend zu belesen, bevor an dem Thema praktisch gearbeitet wird. Dies spart enorm Zeit. Auch weiß ich nun, wie wichtig es ist, sich selbst und den eigenen Arbeitsprozess ständig zu reflektieren, um die Effektivität zu steigern. Hätte ich bspw. meinen Arbeitsprozess frühzeitig hinterfragt, hätte ich die Sinnlosigkeit des beschrieben Kredos frühzeitig realisiert. Diese methodischen Erkenntnisse erachte ich insbesondere mit Blick auf ein mögliches Studium als sehr wertvoll.

Aber auch rein fachlich gesehen habe ich ziemlich viel dazu gelernt. Insbesondere im Bereich Mathematik und C-Programmierung habe ich einiges dazugelernt. Vielleicht helfen diese Kenntnisse mir sogar mal, mich auf dem Arbeitsmarkt oder im Studium zu behaupten.

# Verwendete Quellen

1. **Top Drei und Erläuterung**
2. Neuronale Netze selbst programmieren – ein verständlicher Einstieg (Tariq Rashid)

Dieses Buch war für mich während des Programmierens unabdinglich, denn die mathematischen Grundkonzepte von neuronalen Netzen werden in diesem Buch in einer außerordentlich verständlichen Weise dargelegt. Faltende neuronale Netze erwähnt das Buch jedoch nicht.

Alles, was ich danach programmierte, basiert auf dem mathematischen Wissen, das ich aus diesem Buch erlangt habe. Während des Programmierens stellte das Buch jedoch, anders als der Titel verspricht, nur eingeschränkt eine Hilfe dar. Denn der Autor stellt zwar die Programmierung eines künstlichen neuronalen Netzes, jedoch nicht das eines faltenden neuronalen Netzes dar.

1. An Intuitive Explanation of Convolutional Neural Networks (Ujjwal Karn)

Der Autor veranschaulicht in diesem Artikel das Konzept faltender neuronaler Netze, während mathematische Basiskenntnisse über die grundlegende Funktionsweise neuronaler Netze an dieser Stelle nicht angesprochen werden. Der Artikel von ujjwalkarn (Internet-Synonym) bietet also die perfekte Ergänzung zum Buch von Tariq Rashid.

1. The Future Computed (Microsoft Corporation)

Dieses Buch von Microsoft widmet sich weniger der technischen, sondern vielmehr den ethischen Aspekten von neuronalen Netzen und künstlicher Intelligenz allgemein. Für die Problematik, inwieweit faltende neuronale Netze den Arbeitsalltag von Medizinern erleichtern können und welche ethische und gesellschaftliche Relevanz der Einsatz von faltenden neuronalen Netzen in der Medizin hat, lieferte mir dieses Buch aufgrund dessen bisher eine Vielzahl an Denkanstößen und Ideen.

An dieser Stelle ist etwas kritisch zu betrachten, dass das Buch von Microsoft herausgegeben wurde. Ein wesentlicher Teil seines Umsatzes verdient Microsoft mit neuronalen Netzwerken. Insofern kann man annehmen, dass neuronale Netze in einem überaus positiven Licht dargestellt und wenig kritisch betrachtet werden. Es stellt sich jedoch heraus, dass das Buch einen überraschend kritischen und differenzierten Blick auf neuronale Netze und künstliche Intelligenz im Allgemeinen liefert.

# Anhang

1. **Perspektivische Zeitplanung (bereits mit Prüfungsantrag abgegeben)**

|  |  |
| --- | --- |
| **Zeitleiste** | **geplante** **Arbeitsschritte** |
| Bis 21. 9. 2021 | Themenvorauswahl |
| Bis 26. 10. 2021 | Grobrecherche zu zwei bisher erwägten Themen |
| Bis 2. 11. 2021 | Auswahl eines der zwei Themen, Finalisierung Leitfrag, Abgabe Prüfungsantrag |
| Bis 2. 1. 2022 | Recherche, beispielhaftes Programm/Modell in Python fertigstellen |
| Bis 16. 1. 2022 | Rohpräsentation fertigstellen, Problemfrage Beantwortung fertig |
| Bis 31. 1. 2022 | Wortlaut für Vortrag fertigstellen, Powerpoint fertigstellen, mehrmals durchsprechen, ggf. Hinweise von Zuschauern einarbeiten |
| Bis 28. 2. 2022 | Finalisierung der Dokumentation |
| Bis spätestens 9. 3. 2022 | abschließende Kontrolle, ggf. Korrektur des Erarbeiteten, Abgabe der Dokumentation über die PP an den Referenz-Prüfer |
| 9. 3. bis 6. 4. 2022 | Üben der Präsentation, Portfolio |
| 6. 4. bis 8. 4. 2022 | Prüfungstage für die 5. PK, Abgabe des Portfolios |

1. **Tätigkeitsübersicht**

|  |  |
| --- | --- |
| **Zeitleiste** | **Arbeitsschritte** |
| 25. 10. 2021 bis ca. 31. 12. 2021 | Materialsichtung, Materialauswahl, Recherche zum Thema 🡪 kleine Recherchen noch nicht gänzlich abgeschlossen |
| 15. 11. 2021 bis 1. 3. 2022 | Programmierarbeit, kleinere Recherchen auf Stackoverflow und Github(v.A. zu C und Numpy), Sichtung von Datensets für Training des neuronalen Netzes |
| 1. 12. 2021 bis 1. 3. 2022 | Durcharbeiten von Literaturquellen, Ideensammlung zu Leitfrage |
| ab 1. 2. 2022 | Arbeit an Exposé |
| Ab 1. 3. 2022 | Arbeit an tatsächlicher Präsentation und Wortlaut |

1. **Übersicht über Beratungsgespräche**
2. **Vollständiges Literatur- und Quellenverzeichnis**

|  |  |
| --- | --- |
| **Buchquellen** | * **Monographie**: Microsoft Corporation: The Future Computed – Die gesellschaftliche Bedeutung von Künstlicher Intelligenz (KI). Redmond, USA 20181. * **Monographie:** Rashid, Tariq: Neuronale Netze selbst programmieren. Übersetzt aus dem Englischen. Heidelberg 20171 |
| **Internetquellen** | * <https://www.spektrum.de/lexikon/neurowissenschaft/plastizitaet-im-nervensystem/9979> * <https://flexikon.doccheck.com/de/Neuronale_Plastizit%C3%A4t#:~:text=Die%20synaptische%20Plastizit%C3%A4t%20betrifft%20die,Anzahl%20und%20Organisation%20der%20Synapsen>. * <https://www.desy.de/~guenterg/prosem/Einf_hrung_und_Geschichte_neuronaler_Netze.html> * <https://www.maibornwolff.de/blog/geschichte-von-neural-networks-und-deep-learning> * <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/> * <https://medium.com/analytics-vidhya/implementing-convolution-without-for-loops-in-numpy-ce111322a7cd> * <https://towardsdatascience.com/how-are-convolutions-actually-performed-under-the-hood-226523ce7fbf> * <https://stackoverflow.com/questions/48097941/strided-convolution-of-2d-in-numpy> * <https://stackoverflow.com/questions/51740214/find-the-max-from-each-row-in-python> * <https://medium.com/secure-and-private-ai-math-blogging-competition/https-medium-com-fadymorris-understanding-vectorized-implementation-of-neural-networks-dae4115ca185> * <https://www.youtube.com/watch?v=VfHy5eyJ6hU> * <https://datascience.stackexchange.com/questions/40462/how-to-prepare-the-varied-size-input-in-cnn-prediction> * <https://stackoverflow.com/questions/48121916/numpy-resize-rescale-image> * <https://www.youtube.com/watch?v=pkVwUVEHmfI> * <https://medium.com/analytics-vidhya/when-and-why-are-batches-used-in-machine-learning-acda4eb00763> * <https://www.kaggle.com/ahmedhamada0/brain-tumor-detection> * <https://datascience.stackexchange.com/questions/78132/back-propagation-through-a-simple-convolutional-neural-network> * <https://datascience.stackexchange.com/questions/5987/how-do-i-calculate-the-delta-term-of-a-convolutional-layer-given-the-delta-term?rq=1> * <https://www.youtube.com/watch?v=pUCCd2-17vI> * <https://pavisj.medium.com/convolutions-and-backpropagations-46026a8f5d2c> |
| **Bildquellen und Grafiken** | Abb. 1: [https://www.abiblick.de/nervenzelle / 22. 2. 2022 18:56](https://www.studienkreis.de/fileadmin/lernen/assets/courses/media/nerv-ca.png%20/%2022.%202.%202022%2018:56)  Abb. 2: https://www.youtube.com/watch?v=VOARV8YeTRI  [/ 22. 2. 2022 18:56](https://www.studienkreis.de/fileadmin/lernen/assets/courses/media/nerv-ca.png%20/%2022.%202.%202022%2018:56)  Abb. 3: https://studyflix.de/biologie/epsp-ipsp-2803  [/ 22. 2. 2022 18:56](https://www.studienkreis.de/fileadmin/lernen/assets/courses/media/nerv-ca.png%20/%2022.%202.%202022%2018:56)  Abb. 4: selbst erstellt |

1. **Selbstständigkeitserklärung**