

Mathematische Modelle der biologischen Nervenzelle als
Bausteine künstlicher neuronaler Netze und deren
Anwendung im Gesundheitswesen

Thorsten Suckow-Homberg

Zulassungsarbeit zum Master-Fernstudium

Aachen, 21.11.2023

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Zulassungsarbeit selbstständig und nur unter Verwendung der von mir angegebenen Quellen und Hilfsmittel verfasst zu haben. Dies umfasst auch die Verwendung von KI-Anwendungen, die nach bestimmten Vorgaben automatisiert Inhalte erstellen können. Sowohl inhaltlich als auch wörtlich entnommene sowie von KI-Anwendungen generierte Inhalte wurden als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Thorsten Suckow-Homberg
Aachen, 21.11.2023

Kurzfassung

Neuronale Netze sind in den vergangenen Jahren ein wesentlicher Bestandteil von Forschung und Anwendung in der Medizin geworden.

Die vorliegende Arbeit stellt einige solcher Netze sowie deren theoretischen Unterbauten vor und trägt hierfür Ergebnisse aus Wissenschaft und Forschung zusammen.

Der erste Teil konzentriert sich zunächst auf die Fachlichkeit, die Grundlage künstlicher neuronaler Netze ist: Das biologische Neuron mit seinen komplexen biochemischen Vorgängen, die Signalverarbeitung und -weiterleitung in einem Netz aus Nervenzellen ermöglichen.

Darauf aufbauend bietet der zweite Teil eine Einführung in das mathematische Gerüst des McCulloch-Pitts-Neurons und Rosenblatt-Perzeptrons, zwei frühe Modelle künstlicher Neuronen, die wesentlich zur Entwicklung künstlicher neuronaler Netze beigetragen haben. Hierfür werden die Forschungsarbeiten der Autoren in Auszügen vorgestellt, und Anwendungsbeispiele beleuchten die unterschiedlichen Eigenschaften hinsichtlich Statik des McCulloch-Pitts-Neurons und Anpassungsfähigkeit des Rosenblatt-Perzeptrons.

Im dritten Teil werden einige wegweisende Architekturen und Algorithmen künstlicher neuronaler Netze vorgestellt, darunter Backpropagation sowie Faltungsoperationen, die in den tiefen Netzen Anwendung finden, die heutzutage u.a. in der Medizin grosse Erfolge vorweisen können.

In diesem Zusammenhang bezeugen im vierten und abschliessenden Teil ausgewählte Forschungsarbeiten und -Ergebnisse aus den Bereichen Gesundheitswirtschaft, der Pharmaforschung sowie der Diagnostik und Therapie die Leistungsfähigkeit künstlicher neuronaler Netze.

Inhaltsverzeichnis

1	Einführung	1
2	Signalweiterleitung im biologischen Neuron	2
2.1	Ionenkonzentrationen und Membranspannungen	2
2.2	Das Aktionspotenzial	4
2.2.1	Auslösung eines Aktionspotenzials	5
2.2.2	Signalweiterleitung über das Axon	5
3	Künstliche Neuronen	7
3.1	Das McCulloch-Pitts-Neuron	7
3.1.1	Der Kalkül	7
3.1.2	Anwendung des Modells	8
3.1.3	Aktivierungs- und Eingabefunktion	9
3.1.4	Modellierung der booleschen XOR-Funktion	10
3.2	Das Rosenblatt-Perzeptron	12
3.2.1	Das Perzeptron - ein linearer Klassifizierer	12
3.2.2	Das Modell	12
3.2.3	Lineare Trennbarkeit	13
3.2.4	Die Lernregel	16
3.2.5	Die XOR-Funktion	17
4	Künstliche Neuronale Netze	20
4.1	Mehrschichtige neuronale Netze	20
4.1.1	Backpropagation	20
4.1.2	Neocognitron	22
4.1.3	Convolutional Neural Networks	23
5	Neuronale Netze im Gesundheitswesen	26
5.1	Automatisierung in der Gesundheitswirtschaft	26
5.2	Pharmaforschung	27
5.3	Diagnostik	28
5.4	Therapie und Prognose	29

6	Zusammenfassung und Bemerkung	31
----------	--------------------------------------	-----------

Einführung

Der Einsatz künstlicher neuronaler Netze als Teilgebiet der Künstlichen Intelligenz konnte in der Medizin in den letzten Jahren beachtliche Erfolge vorweisen. Diese Arbeit stellt einige dieser Erfolge vor. Sie möchte darüber hinaus einen ersten Überblick über Themen und Aspekte aus den Bereichen Neurobiologie, Mathematik und Informatik bieten, um ein Verständnis für Modelle und Verfahren zu schaffen, die zu diesen Erfolgen beigetragen haben.

Wir beschäftigen uns deshalb zunächst mit dem Neuron als “strukturelle und funktionelle Einheit des Nervensystems“ [SD07, 42]. Im Hinblick auf neuronale Netze ist für uns das natürliche Vorbild der Signalweiterleitung in Nervenzellen von Interesse, weshalb wir uns mit der Zellmembran und dem *transmembranalen Transport* von Ionen beschäftigen: Es wird sich zeigen, dass Änderungen der Nervenzellmembran-Eigenschaften hinsichtlich ihres Potenzials grundlegend sind für die Ausschüttung von hemmenden oder erregenden Neurotransmittern, die von den Rezeptoren nachgelagerter Nervenzellen weiterverarbeitet und dort gemäß eines Alles-oder-Nichts-Prinzips erneut Signale auslösen können.

Dies bildet die Grundlage für die Neuronenmodelle von McCulloch und Pitts sowie Rosenblatt im zweiten Teil der Arbeit, in dem mathematische Modelle von Eingabe- und Ausgabefunktion sowie deren Anwendung betrachtet werden. Einige diese Vorgänge werden in diesem Abschnitt in Formeln und Gleichungen der Aussagenlogik bzw. analytischen Geometrie überführt, damit sie in der Informatik Anwendung finden können.

Im dritten und vierten Teil werden darauf aufbauend Architektur und Algorithmen einiger künstlicher neuronaler Netze sowie deren überaus erfolgreiche Anwendung im Gesundheitswesen vorgestellt.

Signalweiterleitung im biologischen Neuron

Eingehende Signale empfängt eine Nervenzelle über seine **Dendriten**: Baumförmige Fortsätze, die um den Zellkörper - das **Soma** - herum gelagert sind. Sie fungieren *postsynaptisch* und empfangen über **Rezeptoren** afferente Signale in Form von **Neurotransmittern** (siehe Abbildung 2.1).

Dendriten leiten die Signale weiter an das Soma, in dem sich, durch die **Neuronenmembran** von der Umgebung getrennt, **Zytosol** befindet: Eine salzige, wässrige Flüssigkeit mit einem hohen Anteil von Kalium.

Ob das Neuron Signale weiterleitet, entscheidet sich in der Nähe des **Axonhügels**: Hier entspringt das **Axon**, das in einer "salzigen extrazellulären Flüssigkeit mit hoher Leitfähigkeit" liegt [BCP18, 61]. Eine Depolarisation der Membran an dieser Stelle kann ein **Aktionspotenzial** auslösen (vgl. [BCP18, 142 f.]).

2.1 Ionenkonzentrationen und Membranspannungen

Ein Neuron weist *in Ruhe* eine ungleiche Ionenverteilung zwischen der durch die Zellmembran getrennten **intrazellulären Flüssigkeit** (das Zytosol, im folgenden auch IZF) und der **extrazellulären Flüssigkeit** (im folgenden EZF) auf. In der IZF befinden sich mehr positiv geladene Natrium-Ionen (Na^+), und in der EZF mehr positiv geladene Kalium- und Calcium-Ionen (K^+ und Ca^{2+}) sowie mehr negativ geladene Chlorid-Ionen (Cl^-). Die Membran selber ist mit **Ionenkanälen** durchsetzt, von denen viele **selektiv permeabel** sind.

Wenn kein **postsynaptisches Potenzial** (PSP) wirkt und das Neuron selber keinen Impuls abgibt, liegt das Ruhepotenzial V_r der Zelle zwischen $-70mV$ und $-90mV$: Das Zytosol weist entlang der Membranoberfläche in der IZF eine negative Ladung auf. Diese **Membranspannung** V_m wird durch eine ungleiche Ionenverteilung bewirkt, verursacht durch die Ladung der Teilchen in der IZF und EZF in Membrannähe (siehe Abbildung 2.2).

In Ruhe ist die Leitfähigkeit der Membran für Na^+ gering, für K^+ hingegen hoch (vgl. [SD07, 44]). K^+ -Ionen folgen ihrem **Konzentrationsgradienten** und gelangen über die Ionenkanäle in die EZF, bis die **Potenzialdifferenz** entlang der

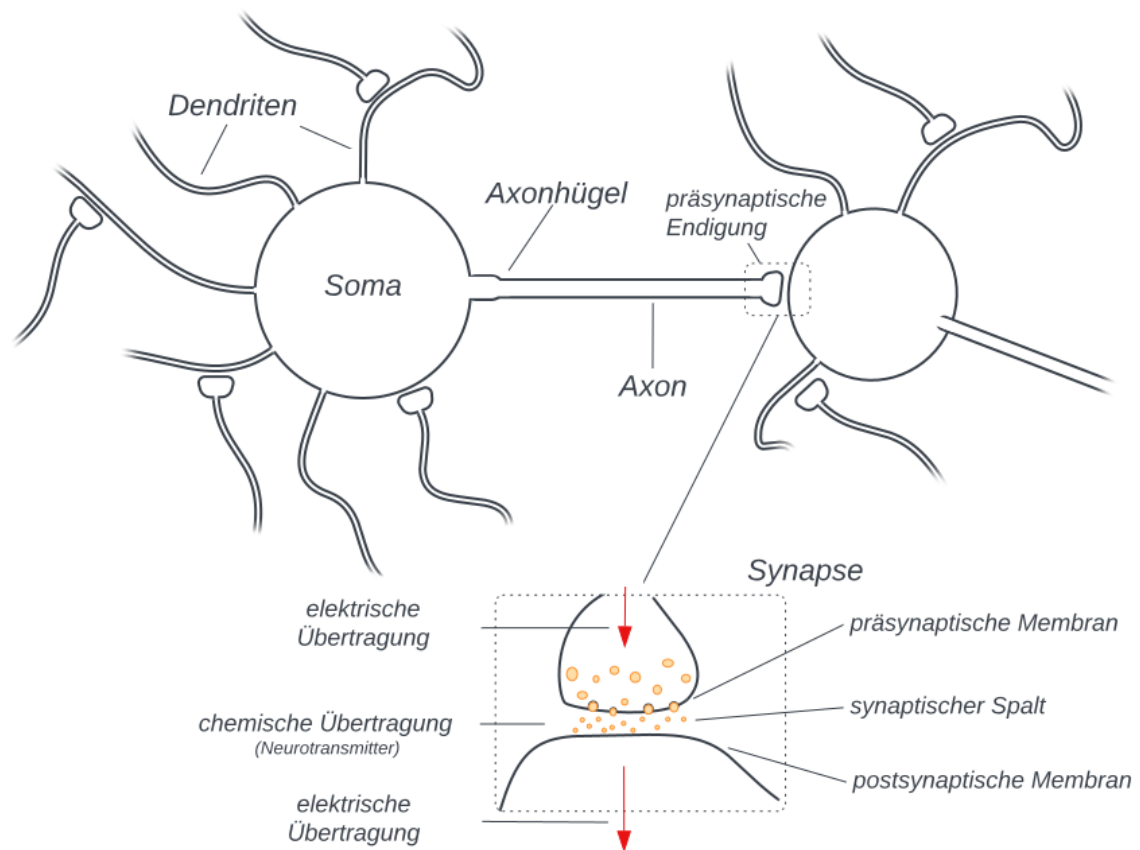


Abb. 2.1. Schematische Darstellung einer Nervenzelle. (Quelle: in Anlehnung an [SD07, 43, Tafel 2.1])

Die Dendriten leiten afferente Signale zum Soma, dem Zellkörper. Das Axon leitet ein efferentes („efferre“ (lat.): hinaustragen, mitnehmen) Nervensignal über präsynaptische Endigungen (auch „Axonterminale“) an (häufig weit entfernte) Effektoren wie Muskeln, Drüsen oder nachgeschaltete Neuronen weiter.

Ion	Konzentration EZF (mmol/l)	Konzentration IZF (mmol/l)	Verhältnis
K^+	5	100	1 : 20
Na^+	150	15	10:1
Ca^{2+}	2	0,0002	10000 : 1
Cl^-	150	13	11,5 : 1

Tabelle 2.1. Ionenkonzentration eines Neurons in Ruhe (Quelle: [BCP18, 75, Abb. 3.15])

Neuronenmembran ausströmende K^+ -Ionen zurückhält: Wenn diese Differenz den

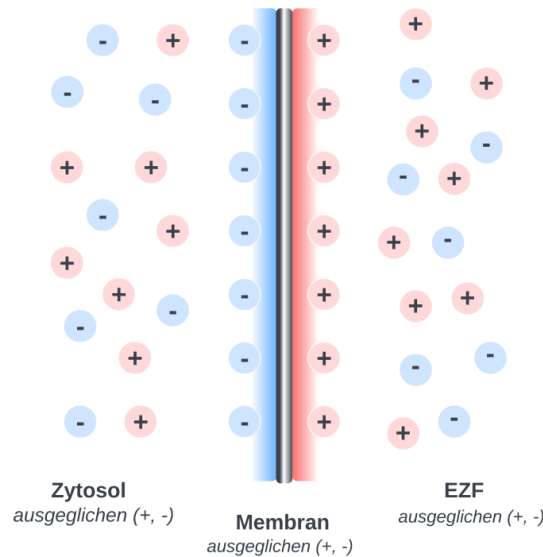


Abb. 2.2. Ionenverteilung im Zytosol und der EZF. (Quelle: in Anlehnung an [BCP18, 72, Abb. 3.13])

Aufgrund der elektrostatischen Anziehungskraft ziehen sich Anionen (neg. geladen) und Kationen (pos. geladen) in der Nähe der Membran gegenseitig an, es kommt zu einer negativen Spannung in Membrannähe (zwischen -70mV und -90mV in Ruhe). Dadurch ist die negative Ladung nicht gleichmässig im Zytosol verteilt. *Bear et al.* weisen darauf hin, dass trotzdem der größte Teil des Zytosols und der EZF elektrisch neutral ist (vgl. [BCP18, 72, Abb. 3.13])

Konzentrationsgradienten für K^+ kompensiert, ist das **Gleichgewichtspotenzial** erreicht, und es findet keine Nettoionenbewegung statt¹.

2.2 Das Aktionspotenzial

In Ruhe ist die Ladung in Membrannähe in der IZF negativ, in der EZF positiv. Durch eine schnelle Umkehrung dieser Verhältnisse ist eine Nervenzelle dazu in der Lage, ein Signal auszulösen (vgl. [BCP18, 86]). Hierzu bedarf es eines **Aktionspotenzials**, das durch Ionenbewegungen entsteht. Die Ionenbewegungen selber sind auf die Änderung des Membranpotenzials zurückzuführen, wodurch ein Öffnen oder Schliessen von ionenspezifischen Kanälen verursacht wird (vgl. [BCP18, 96]).

Kandel et al. führen 4 wichtige Eigenschaften des Aktionspotenzials auf (vgl. [Kan+13, 148 f.]):

1. Es gibt einen **Schwellenwert** für die Auslösung des Potenzials.
2. Das Aktionspotenzial ist ein **Alles-oder-Nichts** Ereignis.
3. Das Aktionspotenzial wird ohne Verlust weitergeleitet.

¹ vgl. [BCP18, 72]. *Kandel et al.* schreiben hierzu: “the equilibrium potential of any ion that is present on both sides of a membrane permeable to that ion“ [Kan+13, 130]

4. Nach dem Auslösen des Aktionspotenzial kommt es zu einer **Refraktärzeit**, in der zunächst kein weiteres Aktionspotenzial ausgelöst werden kann (**absolute Refraktärzeit**). Bevor V_r wieder erreicht ist, wird für eine kurze Dauer ein stärkeres Signal benötigt, um das Aktionspotenzial erneut auszulösen (**relative Refraktärzeit**).

2.2.1 Auslösung eines Aktionspotenzials

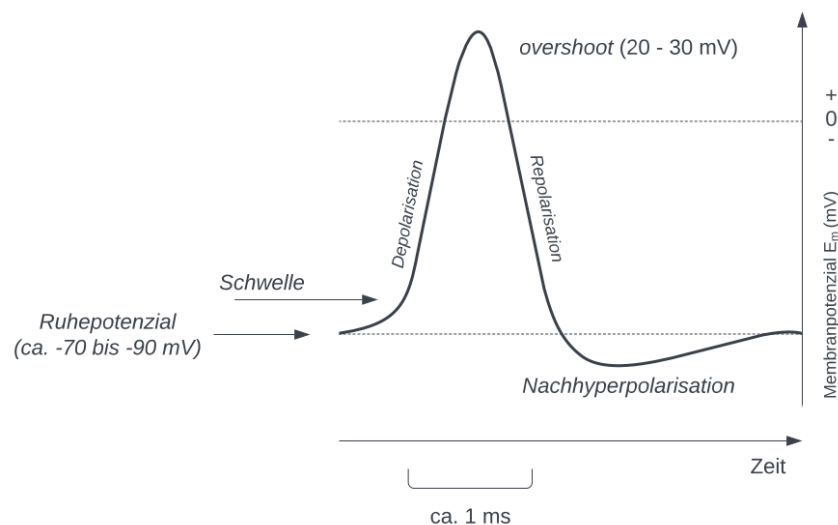


Abb. 2.3. Aktionspotenzial. (Quelle: in Anlehnung an [SD07, 47, Tafel 2.3])

Die **Initiationszone** des Aktionspotenzials ist der Axonhügel. Hier findet sich eine besonders dichte Anhäufung von spannungsabhängigen Na^+ -Kanälen, deren Aktivierungskurve um ca. $10mV$ zu den negativen Membranpotenzialen verschoben ist, was die Initiierung des Signals begünstigt (vgl. [Jon19, 77]).

Damit ein Aktionspotenzial ausgelöst werden kann, muss die Membran nahe der Initiationszone über ihren **Schwellenwert**² V_t ($> V_r$) depolarisiert werden (vgl. [BCP18, 111]). Erst wenn dieser Schwellenwert übertroffen wird, „feuert“ das Neuron: Aktionspotenziale funktionieren deshalb nach dem **Alles-oder-Nichts-Prinzip** (vgl. [Kan+13, 157]).

2.2.2 Signalweiterleitung über das Axon

Kurz nach der Aktionspotenzialbildung befinden sich daran beteiligte Membrane in der Refraktärphase, ihre Na^+ -Kanäle sind inaktiviert. Somit pflanzt sich das Aktionspotenzial nur in eine Richtung fort³ (vgl. [BCP18, 106]). Die Fortleitung

² „das kritische Niveau der Depolarisation, das überschritten werden muss, um ein Aktionspotenzial auszulösen“ [BCP18, 88] oder auch „Erregungsschwelle“ [FE19, 69].

³ *orthodrome* Fortleitung. *Bear et al.* verweisen auf *antidrome* Fortleitung (entgegengesetzte Verlaufsrichtung), die in Experimenten ausgelöst werden kann [BCP18, 106]

geschieht in Richtung der Axonterminal, wo sich die präsynaptischen Endigungen befinden und das elektrische Signal in chemische Signale umgewandelt werden (siehe Abbildung 2.1). Die durch das Aktionspotenzial ausgelöste Depolarisation der Membran an den Axonterminalen bewirkt eine Öffnung spannungsgeladener Calcium-Kanäle (vgl. [Kan+13, 184]). Calcium-Ionen strömen in das Innere der Zelle und lösen die Exozytose von **synaptischen Vesikeln** aus, kleine, mit einer Membran von der IZF getrennte Strukturen von etwa $50nm$ Durchmesser, die mit Neurotransmittern gefüllt sind (vgl. [BCP18, 1000]).

Rezeptoren an den postsynaptischen Endigungen der Folgezelle wandeln diese Botenstoffe in hemmende oder erregende Signale um, die dann nach dem **Alles-oder-Nichts-Prinzip** erneut integriert werden.

Künstliche Neuronen

3.1 Das McCulloch-Pitts-Neuron

Im Jahr 1943 veröffentlichen *Warren McCulloch* und *Walter Pitts* ihre Arbeit [MP43]. Sie stellen darin ein auf **Aussagenlogik** basierendes mathematisches Modell zur Erklärung von Signalverarbeitung und -weiterleitung im Nervensystem vor, und liefern damit grundlegende Ideen für die Kybernetik und die Von-Neumann-Rechnerarchitektur. Ihre Arbeit leistet insgesamt einen wichtigen Beitrag für die Entwicklung *Künstlicher Intelligenz* (vgl. [Arb00, 1]).

3.1.1 Der Kalkül

In der **Principia Mathematica** stellen *Russel* und *Bertrand* die Mathematik auf ein Gerüst logischer Grundprinzipien (vgl. [She26, 225]). Inspiriert dadurch beschäftigt sich McCulloch mit der Frage, ob es nicht auch möglich sei, die Vorgänge im Gehirn durch Anwendung logischer Prinzipien erklärbar zu machen (vgl. [Arb00, 4]). Die Zweiwertigkeit des Alles-oder-Nichts-Prinzips (siehe Abschnitt 2.2) führt zu dem Kalkül:

The ‘all-or-none law’ of nervous activity is sufficient to insure that the activity of any neuron may be represented as a proposition. ([MP43, 100])

McCulloch und Pitts gehen für ihren Kalkül von folgenden physischen Eigenschaften einer Nervenzelle aus (vgl. [MP43, 101]):

1. Ein Neuron arbeitet auf Basis des Alles-oder-Nichts-Prinzip.
2. Eine gewisse Anzahl von Synapsen muss innerhalb einer bestimmten Zeit angeregt werden, um ein Neuron zu erregen.
3. Die einzige signifikante Verzögerung bei der Signalübertragung innerhalb des Nervensystems ist bedingt durch synaptische Verzögerungen.
4. Eine einzige hemmende Synapse verhindert die Erregung des betreffenden Neurons.
5. Die Struktur des neuronalen Netzes ändert sich nicht.

3.1.2 Anwendung des Modells

Das in diesem Abschnitt vorgestellte Modell soll - in Anlehnung an die Arbeit von McCulloch & Pitts - folgende Anforderungen erfüllen¹:

1. Das Verhalten des künstlichen Neurons ist zweiwertig, also *binär*. Das Neuron feuert, oder es feuert nicht. Im Sinne der Beziehung zwischen Eingabe und Ausgabe verstehen wir unter "feuern" den Wert 1. Wenn ein Neuron nicht feuert, wird dies durch den Wert 0 repräsentiert, der ebenfalls als Ausgabe vorliegt. In diesem Sinne produziert unser Modell also immer ein "Signal".
2. An einem Neuron liegen (beliebig viele) Eingaben an.
3. Jedes Neuron hat einen individuellen Schwellenwert. Der Schwellenwert muss getroffen oder übertroffen werden, damit ein Neuron eine "1" feuert.
4. Hemmung ist "absolut": Liegt ein hemmendes Signal zusammen mit nicht-hemmenden Signalen an, feuert das Neuron nicht.

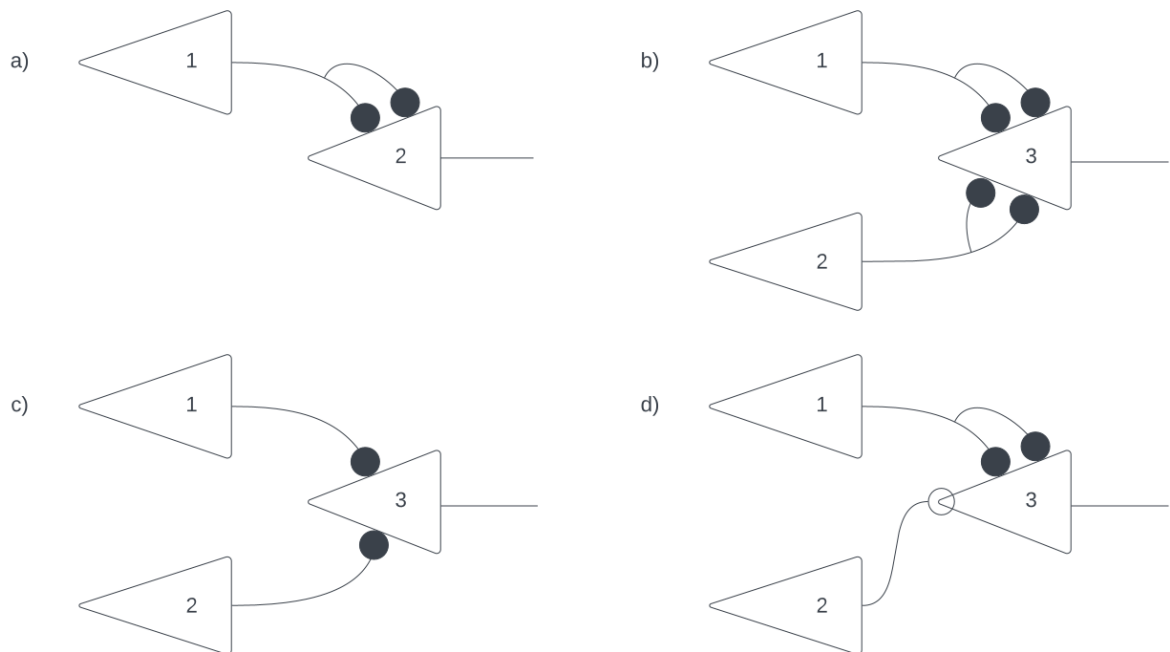


Abb. 3.1. Schematische Darstellung von MCP-Zellen (Quelle: in Anlehnung an [MP43, 105, Figure 1])

Schwarze Kreise sind erregende Verbindungen, offene Kreise hemmende. Im ursprünglichen Modell benötigt ein Neuron zwei erregende Eingaben, um aktiviert zu werden.

In a) ist damit ein Netz dargestellt von zwei Neuronen, bei denen N_2 feuert, wenn N_1 feuert. Unter Berücksichtigung der Zeiteinheit folgt die formale Darstellung $N_2(t) \equiv N_1(t-1)$.

Gleicherweise ergibt sich für b), dass N_3 nur feuert, wenn N_1 **oder** N_2 feuern: $N_3(t) \equiv N_1(t-1) \vee N_2(t-1)$. Analog folgt für c) $N_3(t) \equiv N_1(t-1) \wedge N_2(t-1)$.

Für d) ergibt sich somit $N_3(t) \equiv N_1(t-1) \wedge \neg N_2(t-1)$

¹ vgl. [Fau94, 26 f.]. Für unsere einführende Betrachtung folgen wir [Roj93, 33 f.] und ignorieren zunächst die Zeitwerte aus der Original-Arbeit: Dort wird für die Signalübertragung ein diskreter Zeitwert $t \in \mathbb{Z}$ vereinbart: Die Übertragung eines Signals dauert eine Zeiteinheit, also $t+1$. Vgl. hierzu Abbildung 3.1.

3.1.3 Aktivierungs- und Eingabefunktion

Mit $n \in \mathbb{N}_0$ und $m \in \mathbb{N}_0$ wird festgelegt, dass ein Neuron $n + m$ Eingaben haben soll, mit $n \geq 1 \vee m \geq 1$, wobei n die Anzahl der erregenden und m die der hemmenden Eingaben ist.

Für die Schwellenwertfunktionen eines Neurons N ergeben sich folgende Anforderungen: Die Summe der erregenden Eingabesignale x_1, x_2, \dots, x_n und der hemmenden Eingabesignale y_1, y_2, \dots, y_m muss den für das Neuron festgelegten Schwellenwert $\Theta \in \mathbb{N}_0$ treffen oder überschreiten, damit als Ausgabe eine 1 erzeugt wird, ansonsten liefert die Funktion eine 0 zurück.

Eingabefunktion

Zur Integration der eingehenden Signale wird eine **Eingabefunktion** benötigt. Der Wert dieser Eingabefunktion wird dann auf eine Funktion angewendet, die entscheidet, ob das Neuron feuert oder nicht - also eine 1 oder 0 ausgibt.

Für die Eingabesignale X

$$X \in \{1, 0\}^{n+m} := (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m) \quad (3.1)$$

definieren wir die *Gewichte* $w_+ \in \{2, 1\}$ mit $w_+^1 := 1, w_+^2 := 2$ für erregende Signale, $w_- := -1$ für hemmende Signale (vgl. [Fau94, 27-28]).

Die **Eingabefunktion** g

$$g : \{1, 0\}^{n+m} \rightarrow \mathbb{Z}, X \mapsto \sum_{j=1}^n x_j w_+ + \sum_{k=1}^m y_k w_- \quad (3.2)$$

liefert die Summe der hemmenden und erregenden Eingabesignale zurück.

Aktivierungsfunktion

Die Schwellenwertfunktion wird im Kontext von künstlichen neuronalen Netzen auch **Aktivierungsfunktion** genannt (vgl. [RN12, 847]), da sie entscheidet, ob einzelne Neuronen aktiviert werden oder nicht. In diesem Fall wird sie als **Treppenfunktion** realisiert, die 1 zurückliefert, falls $g(X) \geq \Theta$, und 0 sonst :

$$f : \mathbb{Z} \rightarrow \{0, 1\}, f(g(X)) = f(u) = \begin{cases} 1 & \text{falls } u \geq \Theta \\ 0 & \text{falls } u < \Theta \end{cases} \quad (3.3)$$

Da die Erregung absolut ist, muss Θ die Ungleichung

$$\Theta > \left(\sum_{j=1}^{k^{w_+^2}} 2 \right) + \left(\sum_{j=1}^{k^{w_+^1}} 1 \right) + w_- \quad (3.4)$$

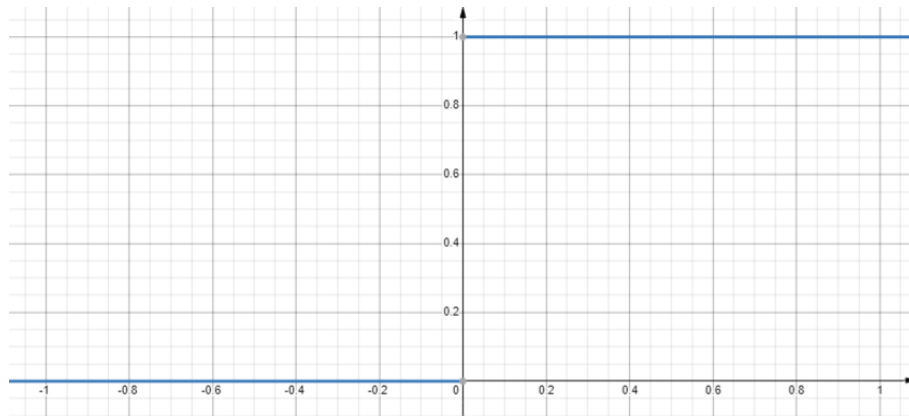


Abb. 3.2. Plot einer Treppenfunktion, hier als Heaviside-Funktion mit $H(x) = \{x < 0 : 0, x \geq 0 : 1\}$. (Quelle: Eigene Darstellung)

erfüllen, was abgekürzt werden kann zu

$$\Theta > 2k^{w_+^2} + k^{w_+^1} - 1 \quad (3.5)$$

3.1.4 Modellierung der booleschen XOR-Funktion

A	B	$A \oplus B$
1	1	0
1	0	1
0	1	1
0	0	0

Tabelle 3.1. Die Wahrheitstabelle für **XOR**

Für die Konstruktion eines Netzes zur Modellierung von **XOR** hilft zunächst das in Abbildung 3.3 dargestellte Modell der **OR**-Funktion. Der Schwellenwert ($\Theta = 2$) wird erreicht, wenn entweder x_1 oder x_2 ein erregendes Signal weiterleitet. Allerdings wird der Schwellenwert auch erreicht, wenn jeweils x_1 und x_2 gleichzeitig aktiv sind, denn dann ist $f_{xor}(g(X)) = 4$. Also muss eine Hemmung zwischengeschaltet werden (s. Abbildung 3.4): Ein aktives x_1 hemmt N_2 . N_2 leitet in dem Fall 0 an Y weiter. Die Zellen x_2 und N_1 werden analog verbunden.

Die **disjunktive Normalform** (Gleichung 3.6) und die **konjunktive Normalform** (Gleichung 3.7) von **XOR** liefern die Formel für das Netz. Insbesondere die disjunktive Normalform ist in Abbildung 3.4 erkennbar.

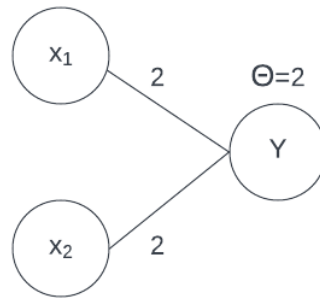


Abb. 3.3. Ein MCP-Neuron zur Modellierung von **OR** (Quelle: Eigene Darstellung)

$$A \oplus B \equiv (\neg A \wedge B) \vee (A \wedge \neg B) \quad (3.6)$$

$$A \oplus B \equiv (A \vee B) \wedge (\neg A \vee \neg B) \quad (3.7)$$

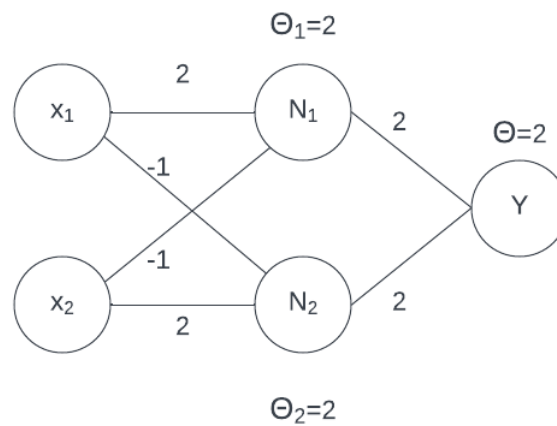


Abb. 3.4. Entwurf für ein MCP-Netz zur Modellierung von **XOR** (Quelle: Eigene Darstellung)

3.2 Das Rosenblatt-Perzeptron

MCP-Netze sind durch vorhergehende Analyse der Anforderungen und entsprechender Anpassung der Topologie zur Lösung verschiedener Aufgaben imstande. *Donald Hebb's* Theorien über *synaptische Plastizität* führen einige Jahre nach dem MCP-Neuron zu einem Modell, das in der Lage ist, sich selbst anzupassen.

3.2.1 Das Perzeptron - ein linearer Klassifizierer

Bereits 1954 wurden Versuche unternommen, lernfähige neuronale Netze zu modellieren (vgl. [Ros62, 24]). 1958 schafft es ein Modell, für Aufsehen zu sorgen: Das **Perceptron** (im folgenden Perzeptron) (vgl. [AR88, 89]). 1957 beschreibt es sein Schöpfer *Frank Rosenblatt* (1928 - 1971) in [Ros57] als Teil eines internen Forschungsprojektes des *Cornell Aeronautical Labors*.

3.2.2 Das Modell

Rosenblatt definiert das Perzeptron wie folgt²:

A perceptron is a network of S, A, and R units with a variable interaction matrix V which depends on the sequence of past activity states of the network. ([Ros62, 83 "DEFINITION 17"; Hervorhebung i.O.]

Dabei ähneln die dort erwähnten A-Units als verrechnende Einheiten nicht bloß zufällig der in Abschnitt 3.1.3 vorgestellten Aktivierungsfunktion: Rosenblatt selber weist darauf hin, dass er sein Modell direkt von dem von McCulloch und Pitts eingeführten Modell ableitet³.

Rojas schreibt, dass das klassische Rosenblatt-Perzeptron in einem Netz von Eingabe- und Ausgabeknoten gewichtete Verbindungen nutzt - die Knoten selber sind Schwellenwertelemente, Verbindungen werden stochastisch ermittelt (siehe [Roj93, 51]). Er weist ebenda darauf hin, dass das Modell nach Rosenblatts Veröffentlichung analysiert und verfeinert wurde, u. a. von *Minsky und Papert* in [MP88], die wie folgt definieren:

A perceptron is a device capable of computing all predicates which are linear in some given set Φ of partial predicates. ([MP88, 12; Hervorhebung i.O.]

Prädikate sind hier Verbindungen zu den Eingabesignalen, die einen Wahrheitswert 0 oder 1 basierend auf der Eingabe X berechnen. Die Ausgaben der Prädikate werden individuell gewichtet und an die Zelle weitergeleitet, die die Aktivierungsfunktion implementiert (vgl. [Roj93, 52 f.] und [MP88, 8-12]).

Diese Eingabefunktion setzt sich wie Gleichung 3.2 aus der Summe der Produkte der Prädikate $P_i \in \Phi$ (wobei $P_i(X) \in \{0, 1\}$) und den Gewichten $w_i \in \mathbb{R}$

² Ausführliche Definitionen aller Zustände, Signale und Funktionen in [Ros62, 79 - 94]

³ vgl. auch "Ein einfaches Perzeptron ist eine McCulloch-Pitts-Zelle, die ihre Eingabe gewichtet berechnet." in [Roj93, 57, "Definition 3.1"; Hervorhebung i.O.]

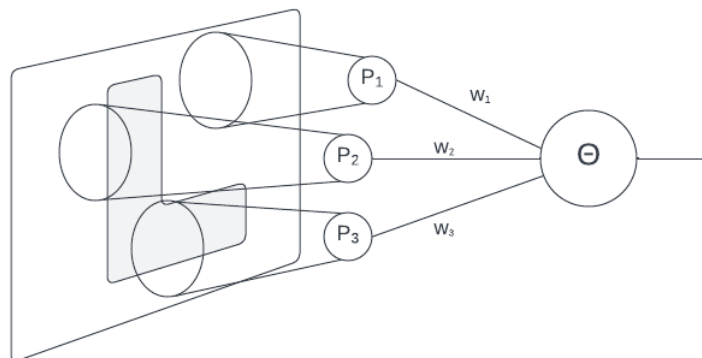


Abb. 3.5. Rosenblatt-Perzeptron. (Quelle: in Anlehnung an [Roj93, 53, Abb. 3.2])

Schematische Darstellung eines Perzeptron-„Automats“ (siehe [Ros57, 7]): Eingaben werden an die Prädikate P_n weitergeleitet, deren binäre Ausgaben mit den Gewichten w_n multipliziert werden. Die Summe der Produkte wird mit dem Schwellenwert Θ verglichen (vgl. [Roj93, 53]).

der Verbindungen zusammen (s. Gleichung 3.8), und die Aktivierungsfunktion (s. Gleichung 3.9) ist wieder eine Treppenfunktion mit dem reellen Schwellenwert Θ :

$$g := g(X) = \sum_{i=1}^n P_i(X)w_i \quad (3.8)$$

$$f := f(g(X)) = f(x) = \begin{cases} 1 & \text{falls } x \geq \Theta \\ 0 & \text{falls } x < \Theta \end{cases} \quad (3.9)$$

3.2.3 Lineare Trennbarkeit

Mit Gleichung 3.9 folgt für eine Eingabe, dass sie entweder eine 0 oder 1 als Ausgabe erzeugt. Es liegt ein binärer Wertebereich vor, der als zwei unterschiedliche Klassen aufgefasst werden kann: Eingabedaten können entsprechend ihrer Ausgabe einer der beiden Klassen zugeordnet werden (vgl. [RN12, 812]). Im Folgenden sollen die Zusammenhänge geometrisch dargestellt werden. Der Einfachheit halber beschränken wir uns hierzu auf den zweidimensionalen Raum \mathbb{R}^2 und betrachten dort die 1. *Winkelhalbierende* im I. und III. Quadranten des kartesischen Koordinatensystems. Die zugehörige *Gerade* L ist

$$L = \{(x_1, x_2) \in \mathbb{R}^2 : x_1 = x_2\} \quad (3.10)$$

Für beliebige Punkte $(x_1, x_2) \in \mathbb{R}^2$ gilt damit offensichtlich

$$x_1 - x_2 \begin{cases} > 0 & \text{falls } x_1 > x_2 \\ = 0 & \text{falls } x_1 = x_2 \\ < 0 & \text{falls } x_1 < x_2 \end{cases} \quad (3.11)$$

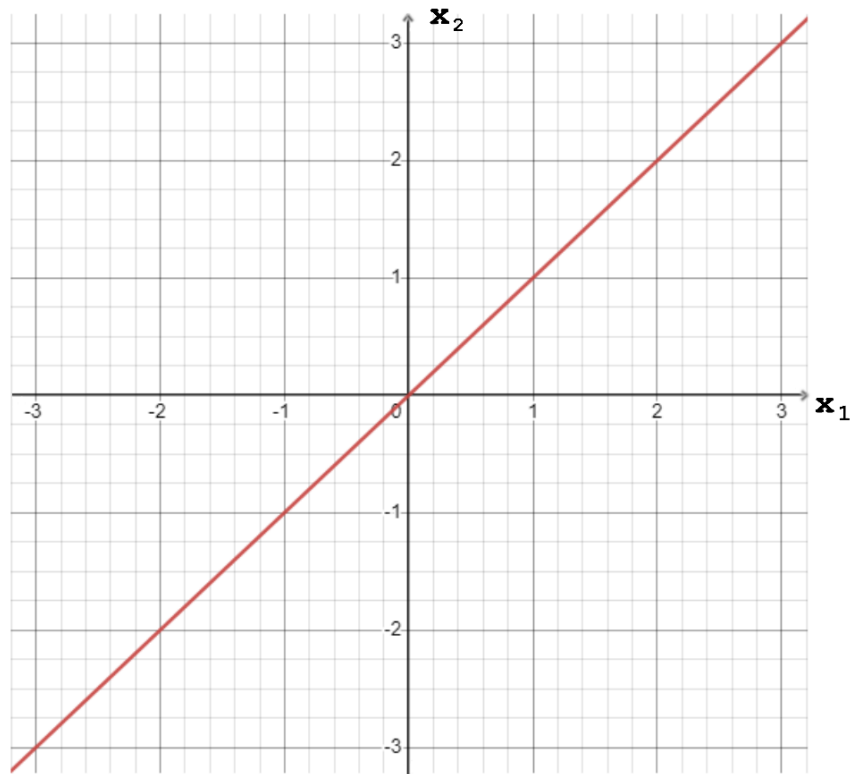


Abb. 3.6. 1. Winkelhalbierende im kartesischen Koordinatensystem (Quelle: Eigene Darstellung)

Die Gerade L repräsentiert eine *Hyperebene*⁴ im \mathbb{R}^2 . Punkte, die nicht zu dieser Hyperebene gehören, liegen in zwei unterschiedlichen *Halbräumen* (siehe geometrische Darstellung in Abbildung 3.7).

Wir können feststellen, dass

- Punkte, die $x_1 - x_2 > 0$ erfüllen (im folgenden M_-) in dem Halbraum *unter* der durch L beschriebenen Gerade liegen
- Punkte, für die $x_1 - x_2 < 0$ gilt (im folgenden M_+) *über* der durch L beschriebenen Gerade liegen
- Punkte mit $x_1 - x_2 = 0$ *auf* der Geraden liegen (im folgenden als Teilmenge von M_-)

M_+ und M_- sind damit *linear separierbar*. Nach *Rojas* lautet die Definition für **Lineare Trennbarkeit**:

Zwei Mengen A und B von Punkten in einem n -dimensionalen Raum sind *linear trennbar*, falls $n+1$ reelle Zahlen w_1, \dots, w_{n+1} existieren, so dass für jeden Punkt $x_1, \dots, x_n \in A$ gilt

⁴ siehe [Bur+12, 81, Definition 2.3]

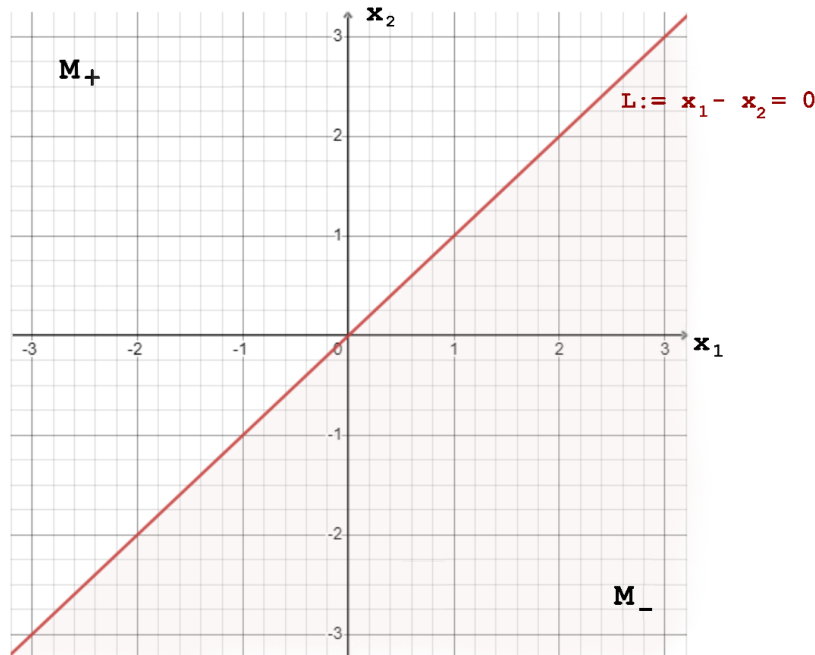


Abb. 3.7. Halbräume im \mathbb{R}^2 (Quelle: Eigene Darstellung)

Skizzierung der durch die 1. Winkelhalbierende entstandenen Halbräume. Die Mengen M_- und M_+ sind linear separierbar, die Gleichung für die *Trenngerade* hierzu lautet $x_1 - x_2 = 0$

$$\sum_{i=1}^n w_i x_i \geq w_{n+1} \quad (3.12)$$

und für jeden Punkt $x_1, \dots, x_n \in B$

$$\sum_{i=1}^n w_i x_i < w_{n+1} \quad (3.13)$$

([Roj93, 60 f., “Definition 3.2“; Hervorhebung i.O.; Nummerierung eigene])

Insgesamt kann ein Perzeptron als Funktion verstanden werden. *Ertel* definiert hierzu :

Sei $w = (w_1, \dots, w_n) \in \mathbb{R}^n$ ein Gewichtsvektor und $x \in \mathbb{R}^n$ ein Eingabevektor. Ein **Perzeptron** stellt eine Funktion $P : \mathbb{R}^n \rightarrow \{0, 1\}$ mit

$$P(x) = \begin{cases} 1 & \text{falls } wx = \sum_{i=1}^n w_i x_i > 0 \\ 0 & \text{sonst} \end{cases}$$

dar. ([Ert21b, 212, “Definition 8.3“; Hervorhebung i.O.])

Das Bias-Gewicht

Die sogenannte **bias unit** ermöglicht eine Trennbarkeit von Punktmengen, die nicht durch eine Ursprungsgerade separierbar sind (siehe Abbildung 3.8): Das Bias-Gewicht⁵ ist ein Wert, der zu der Gleichung 3.8 hinzuaddiert wird. In der geometrischen Darstellung sorgt dieser Wert für eine Verschiebung der Ursprungsgeraden entlang der x_2 -Achse⁶.

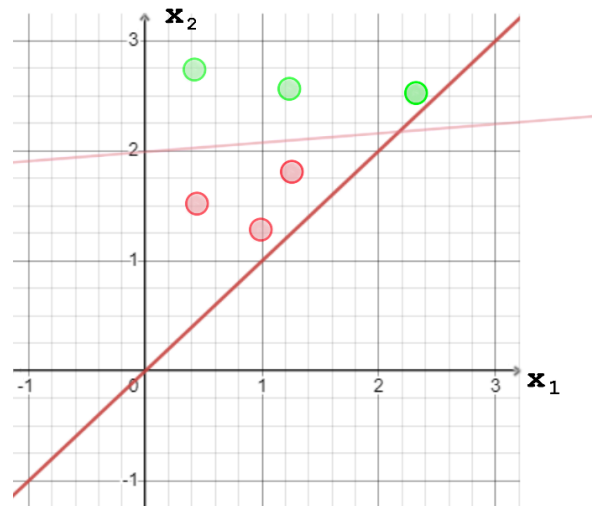


Abb. 3.8. Punkte im \mathbb{R}^2 , die nicht durch eine Ursprungsgerade separierbar sind. Angedeutet eine mögliche Trenngerade, die durch $(0, 2)$ geht. (Quelle: Eigene Darstellung)

3.2.4 Die Lernregel

Die Eigenschaft linearer Trennbarkeit von Daten ist eine wesentliche Voraussetzung dafür, dass ein Perzeptron **konvergiert**: Die **Lernregel** des Perzeptrons passt während der Laufzeit die Gewichte $w_1 \dots w_n$ solange an, bis sie - eingesetzt in eine lineare Gleichung (vgl. [Ert21c, 311]) - die n -dimensionalen Daten entsprechend Gleichung 3.9 *klassifizieren* kann. Aus diesem Grund wird das Perzeptron auch **linearer Klassifizierer** genannt (vgl. [Ert21b, 210-216]).

Das Perzeptron **lernt** diese Gewichte zunächst durch *Trainingsdaten*. Jeder Eintrag dieser Trainingsdaten ist einer erwarteten Ausgabe zugeordnet. Der Algorithmus besteht aus folgenden Schritten (vgl. [RM87, 65] sowie [RN12, 842]):

1. Wähle einen Datensatz und berechne die Ausgabe.
2. Wenn die Ausgabe 1 ist, obwohl sie 0 sein sollte (Fehler⁷ = -1), verringere die Gewichte.

⁵ vgl. [RN12, 839]

⁶ im \mathbb{R}^n durch eine Hyperebene im Ursprung (vgl. [Ert21b, 215]).

⁷ Der **Fehler** ist die Differenz von erwarteter Ausgabe und tatsächlicher Ausgabe.

3. Wenn die Ausgabe 0 ist, obwohl sie 1 sein sollte (Fehler = 1), erhöhe die Gewichte.
4. Wenn die Ausgabe korrekt ist, passe die Gewichte nicht an.

Die Schritte werden so lange durchlaufen, bis für alle Trainingsdaten die Ausgabe korrekt ist, oder eine maximale Anzahl von Trainingsläufen erreicht wurde. Einen Trainingslauf nennt man dabei **Epoche**⁸. Sind die Trainingsdaten linear separierbar, konvergiert⁹ das Perzeptron nach einer endlichen Zahl von Epochen (vgl. [MP88, 164])¹⁰, und ist danach in der Lage weitere Daten zu **generalisieren** (vgl. [Ert21b, 202]).

3.2.5 Die XOR-Funktion

Wenn ein Perzeptron nicht konvergiert, kann es ausreichen, die Anzahl der Epochen zu erhöhen, damit ein passender Gewichtsvektor gefunden wird (vgl. [Arb03, 20]). Allerdings kann es bereits bei wenigen Daten und beliebig großer Epochenzahl vorkommen, dass ein Perzeptron nicht konvergiert, nämlich wenn die Daten nicht linear separierbar sind (vgl. [Arb03, 20]).

Als Beispiel betrachten wir die boolesche Funktion **XOR** (vgl. Tabelle 3.1). In Abbildung 3.10 ist die geometrische Repräsentation der möglichen Interpretationen für $A \oplus B$ dargestellt. Zwar lassen sich die Elemente separieren, aber nicht linear. Es müsste sonst ein w_1, w_2 existieren, das folgende Ungleichungen erfüllt:

$$w_1 0 + w_2 0 < \Theta$$

$$w_1 1 + w_2 0 \geq \Theta \implies w_1 \geq \Theta$$

$$w_1 0 + w_2 1 \geq \Theta \implies w_2 \geq \Theta$$

$$w_1 1 + w_2 1 < \Theta$$

Offensichtlich kann $w_1 + w_2 < \Theta$ nicht erfüllt werden, wenn gleichzeitig $w_1 \geq \Theta$ und $w_2 \geq \Theta$ gilt (s. Abbildung 3.10).

⁸ siehe [Fau94, 436]

⁹ “iterative training processes converge if the weight updates reach equilibrium (stop changing)” [Fau94, 425 “Convergence”].

¹⁰ Das Konvergenz-Theorem besagt: “if a linear separation exists, the perceptron error-correction scheme will find it.” [Arb03, 20]. Beweise führen [Ros62, 111 ff.], [MP88, 168 ff.] sowie [Nov62].

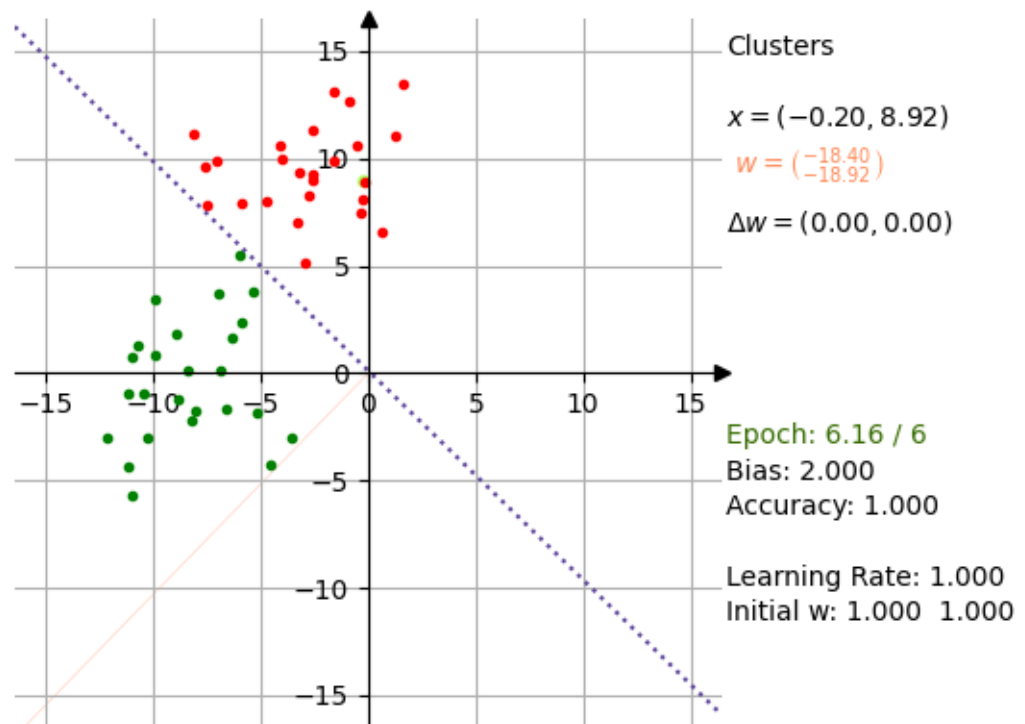


Abb. 3.9. Perzeptron-Training für grosse Datenmengen. (Quelle: Eigene Darstellung)

Ein Perzeptron wird mit einer großen Datenmengen (50 Einträge) trainiert. Nach knapp 300 Trainings-schritten (in der 6ten Epoche) wird die Trenngerade (gestrichelt) gefunden.

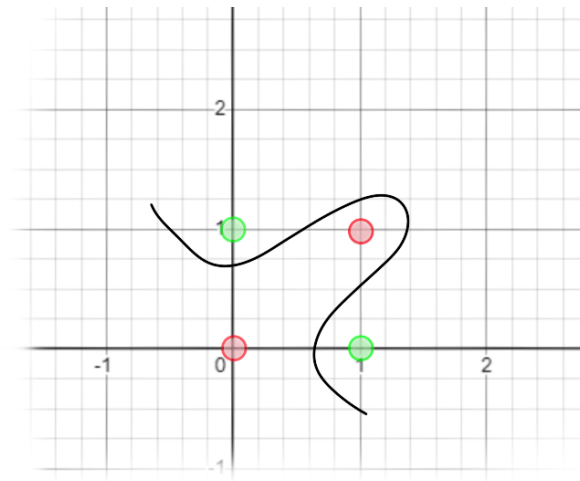


Abb. 3.10. XOR im \mathbb{R}^2 . (Quelle: Eigene Darstellung)

Interpretationen der XOR-Funktion im kartesischen Koordinatensystem. Hier existiert keine Trenngerade für die Punkte $\{(0,0), (1,1)\}$ und $\{(0,1), (1,0)\}$.

Künstliche Neuronale Netze

4.1 Mehrschichtige neuronale Netze

Bei dem Rosenblatt-Perzeptron, das alleine nicht in der Lage ist, XOR zu erlernen, handelt es sich um ein **einschichtiges neuronales Feedforward-Netz**. Das bedeutet, dass es nur Eingabe und Ausgabe gibt und die Informationen ausschließlich in Richtung Ausgang fließen (vgl. [RN12, 848]).

Ein **mehrschichtiges Perzeptron** (MLP, *multi-layer perceptron*) [GBC18, 6] ist in der Lage, nicht-linear-separierbare Daten zu klassifizieren. Ein MLP repräsentiert ein tiefes Feedforward-Netz, in dem die Eingabeschicht (**Input Layer**) und die Ausgabeschicht (**Output Layer**) über weitere Schichten von Neuronen (**hidden layer**) verbunden ist; die Neuronen in diesen Schichten implementieren Eingabe- und Aktivierungsfunktion und leiten ihre Berechnungen an die nächsten Zellen bis zu der Ausgabeschicht weiter.

4.1.1 Backpropagation

Murtagh zeigt in [Mur91, 185 f.] ein MLP, das mittels **Backpropagation**¹ Daten klassifiziert, die keinen linearen Zusammenhang besitzen. Der Algorithmus geht auf *Rumelhart, Hinton und Williams* zurück, die in [RM87, 318 ff.] eine Methode² vorstellen, um berechnete Werte *rückwärts* in das Netz einzuspeisen. Hierbei wird für die Netzausgabe (*forward pass*) ein Approximationsfehler berechnet, der als Basis für die Gewichtsänderungen beim rückwärtigen Lauf (*backward pass*) bis zur ersten verborgenen Schicht genutzt wird. Der Vorgang (forward pass, backward pass, forward pass...) wird so lange für alle Trainingsbeispiele wiederholt, bis die Gewichte sich nicht mehr ändern, oder eine andere Schranke (wie bspw. Epoche oder Zeit) erreicht ist (vgl. [Ert21c, 315])³.

Die mathematische Basis für Backpropagation ist das **Gradientenabstiegsverfahren**, das dabei hilft, in einem neuronalen Netz Parameter für möglichst optimale (d.h. kleine) Verlust-Werte (also geringe Fehler-Werte) zu finden (vgl. [RN12,

¹ “das meist genutzte neuronale Modell“ [Ert21c, 313]

² “The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure.“ [RHW86, 533]

³ ausführlicher Algorithmus in [RN12, 853, Abbildung 18.24].

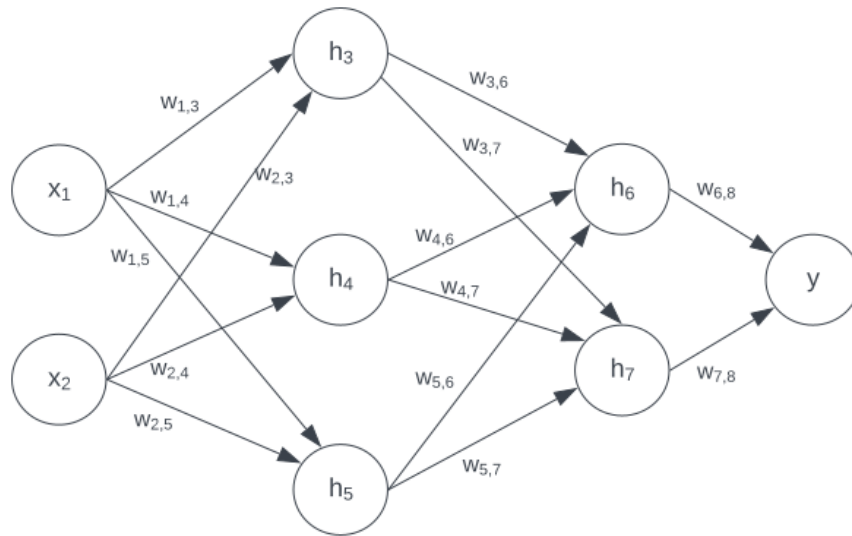


Abb. 4.1. Exemplarische Darstellung der Architektur eines einfachen Feed-Forward-Netzes. (Quelle: Eigene Darstellung)

Ein Feed-Forward-Netz mit zwei Eingabeneinheiten x_1, x_2 , zwei versteckten Schichten und einer Ausgabeneinheit y . Das dargestellte Netz ist vergleichsweise klein. Grosse Modelle wie Microsoft's Turing-NLG besitzen bis zu 105 Schichten und 530.000.000.000 Parameter (<https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/>, abgerufen 27.09.2023).

837]; s. a. Abbildung 4.3). Als Aktivierungsfunktion unterstützt eine **Sigmoid-Funktion** (Gleichung 4.1 sowie Abbildung 4.2) aufgrund ihres nicht-linearen Charakters eine größere Klasse darstellbarer Funktionen und damit Lösungen für Probleme, die ein klassisches Perzeptron nicht lösen kann (vgl. [Ert21c, 316]).

Sigmoid:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (4.1)$$

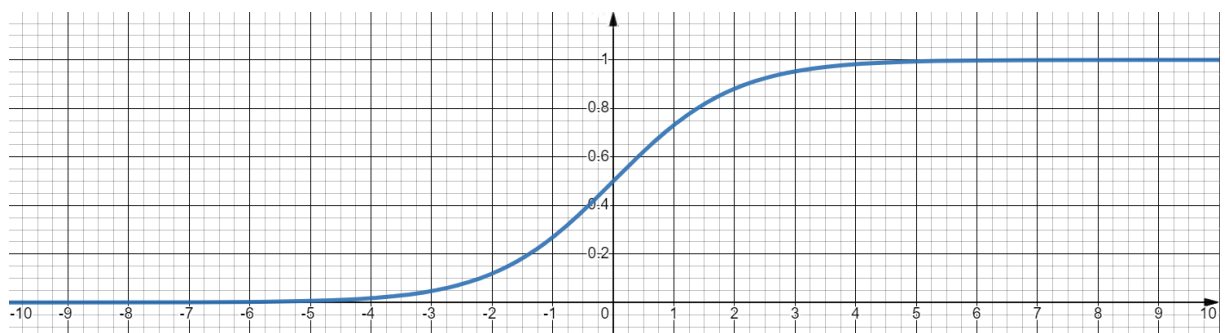


Abb. 4.2. Plot einer Sigmoid-Funktion. (Quelle: Eigene Darstellung)

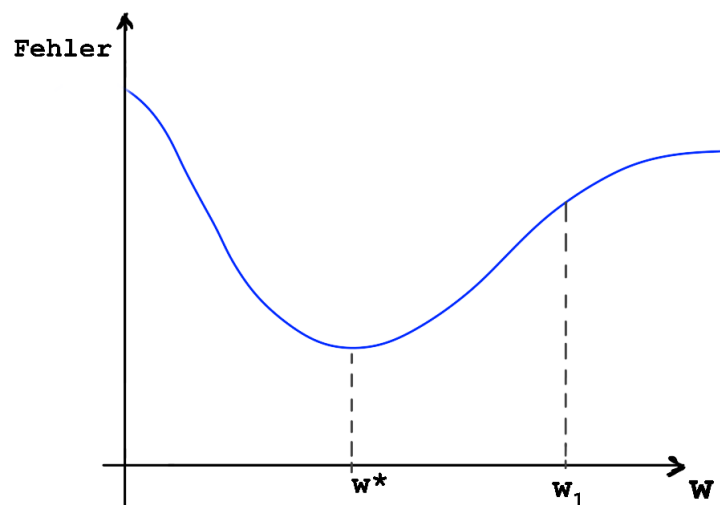


Abb. 4.3. Skizzierung des Zusammenhangs Fehlerfunktion und lokale Minima (Quelle: in Anlehnung an [Son22, 52, Abb. 2.15])

Die y -Achse repräsentiert den Fehlerwert, die x -Achse den berechneten Gewichtsvektor w_1 in einem neuronalen Netz. Offensichtlich existiert in dem Netz w^* , für den der Fehler geringer ist als für w_1 . Das Gradientenabstiegsverfahren wird genutzt, um diese Minima zu finden (vgl. [Son22, 52]).

4.1.2 Neocognitron

Unter dem Namen *Cognitron* 1975 beschreibt *Fukushima* in [Fuk75] ein mehrschichtiges Netz mit selbst-organisierenden Eigenschaften zur Mustererkennung, in dem Zellen selektiv auf häufig präsentierte Merkmale reagieren. 1983 veröffentlichten *Fukushima et al.* eine Modifikation dieser Architektur in [FMI83]⁴ unter dem Namen **Neocognitron**⁵; sein biologisches Vorbild ist das durch *Hubel und Wiesel* in [HW62] beschriebene hierarchische Modell des Wahrnehmungssystems (vgl. [FMI83, 827]). In dem künstlichen neuronalen Netz haben Zellen in tiefer liegenden Schichten die Eigenschaft, selektiv komplexere Merkmale der Stimuli zu extrahieren, womit sie weniger anfällig gegenüber Rauschen in den Eingabedaten sind. In [Fuk80] war der Trainingsprozess durch wiederholte Einspeisung von Mustern ohne weitere Information gegeben (vgl. [Fuk80, 197])⁶: Die Erweiterung des Modells beinhaltet nun die Verstärkung der modifizierten Synapsen durch überwachtes Lernen, um bessere Resultate bei handgeschriebenen Zeichen zu erzielen (vgl. [FMI83, 829]). *Anderson und Rosenfeld* attestieren dem Netz von *Fukushima et al.* Aspekte, die bei der Entwicklung neuronaler Netze eine wesentliche Rolle spielen werden (siehe [AR88, 524]): Tatsächlich inspiriert das Neocognitron die

⁴ mit Bezug auf [Fuk80] ("Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position").

⁵ Video mit Demonstration des Netzes unter <https://www.youtube.com/watch?v=Qil4kmvm2Sw>, abgerufen 04.09.2023

⁶ "learning without a teacher", also unüberwachtes Lernen (unsupervised learning).

Convolutional Neural Networks (CNN) (vgl. [LBH15, 439]), deren erster Entwurf 1989 von *LeCun* in [LeC89] vorgestellt wird.

4.1.3 Convolutional Neural Networks

Yann LeCun veröffentlicht 1989 seine Arbeit [LeC89], in der er verschiedene Netzwerkarchitekturen auf Generalisierungsfähigkeit (s. Abschnitt 3.2.4) und Performance untersucht. Er kommt zu dem Schluss, dass eine Reduzierung der *freien Parameter*⁷ in einem mehrschichtigen Netz zu einer hohen Generalisierungsfähigkeit führt: Bessere Ergebnisse im Vergleich zu ein- bzw. zweischichtigen Netzen können erzielt werden, indem sog. *feature maps* genutzt werden, die in den Schichten für die Merkmalsextraktion der Eingabedaten (hier: zweidimensionale Bilder) verantwortlich sind. Zusätzliche Information wie die Lage der Merkmale in den Eingabedaten werden näherungsweise gespeichert, was zu einer Reduzierung der Größe der *feature maps* im Vergleich zu der Größe der Eingabedaten führt, und damit auch zu einer Reduzierung der Gewichte. Darüber hinaus können mehrere *feature maps* die gleichen Merkmale an unterschiedlichen Orten (*shift invariance*) in den Eingabedaten extrahieren, wodurch die Gewichte unter diesen *feature maps* geteilt werden können (vgl. [LeC89, 151 f.]). In [LeC+89] stellen *LeCun et al.* diese Architektur als **Convolutional Network LeNet-1** vor (vgl. [LeC+98, 13]), das äußerst erfolgreich mittels Backpropagation handgeschriebene Postleitzahlen erkennt. Das Netz performt mit 30 Klassifizierungen pro Sekunde, lediglich die Normalisierung der Eingabedaten stellt einen Flaschenhals bei dem Prozess dar. Wird dieser berücksichtigt, werden 10-12 Ergebnisse pro Sekunde erzielt (vgl. [LeC+89, 549]).

Eine mathematische Basis von CNNs ist die **Faltungsoperation**: Hierbei wird auf Eingabedaten ein **Kernel** (die *Faltungsmatrix*)⁸ angewendet, wobei das Ergebnis der Faltungsoperation die **feature map** (*Merkmalskarte*) ist (vgl. [GBC18, 370]). *Goodfellow et al.* erläutern in [GBC18, 374 ff.] die Optimierungen, die durch den Einsatz von Faltung hervorgehen: Durch den Einsatz von Kernen, die nur aus einem Bruchteil der Größe der Eingabedaten bestehen⁹, können “aussagekräftige Merkmale” aufgespürt werden, was dazu führt, dass weniger Parameter gespeichert werden müssen und gleichzeitig die “statistische Effizienz” des Netzes erhöht wird (**sparse interaction** bzw. **sparse weights**). Durch **parameter sharing** (auch: **tied weights**) kann die Effizienz des Netzes erhöht und Speicherplatzverbrauch verringert werden, und Schichten weisen eine **Äquivarianz** gegenüber Verschiebung auf.

⁷ *freie Parameter* sind die Parameter des Netzes, die durch den Lernvorgang festgelegt werden, also bspw. die Gewichte, Schwellenwerte oder auch die Lernrate, wenn diese durch Verwendung bestimmter Optimierungsalgorithmen adaptiv ist, wie bspw. **Adam** (vgl. [GBC18, 346] mit Verweis auf [KB17])

⁸ “das biologische Analogon zu den Filtern [Kernen] sind die bei den Sinnesorganen vorkommenden rezeptiven Felder.“ [Ert21c, 326]. Vgl. hierzu auch [LBH15, 439].

⁹ *Goodfellow et al.* beziffern für Eingaben von 1.000.000 Pixeln Kernel mit einer Größe von einigen hundert Pixeln als ausreichend (siehe [GBC18, 374]).

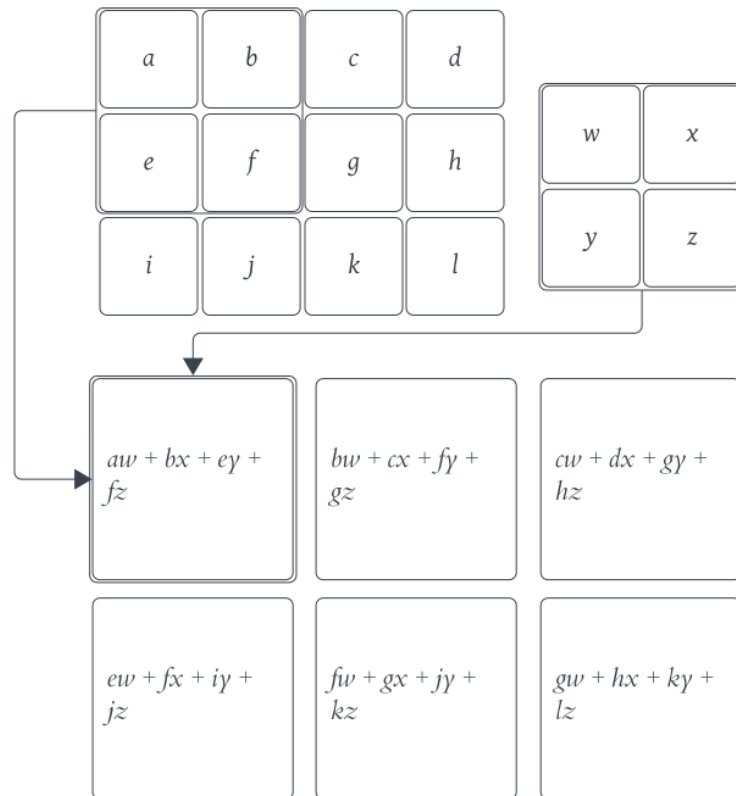


Abb. 4.4. Beispiel einer Faltungsoperation. (Quelle: in Anlehnung an [GBC18, 372, Abbildung 9.1])

In dem Beispiel wird zur Faltung ein 2×2 -Kernel mit einer 3×4 -Eingabe verwendet. Das Ergebnis ist eine 2×3 -feature map.

CNNs nutzen neben den Convolution Schichten auch Pooling Schichten (vgl. [Ert21c, 325]), die die Ausgaben des Netzes durch eine “statistische Größe der nahegelegenen Ausgaben“ ersetzt [GBC18, 379], was die Zahl der Pixel auf den feature maps reduziert¹⁰, außerdem wird als Aktivierungsfunktion meistens die nichtlineare ReLU (Rectified Linear Unit, s. Gleichung 4.2 und Abbildung 4.5) verwendet, die die Konvergenz der Netze verbessert (vgl. [Ert21c, 327]).

ReLU:

$$f(x) = \max(0, x) \quad (4.2)$$

¹⁰ *LeCun et al.* weisen darauf hin, dass die Pooling Schicht auch die Aufgabe hat, semantisch ähnliche Merkmale zusammenzufassen: “the role of the pooling layer is to merge semantically similar features into one.” [LBH15, 439]

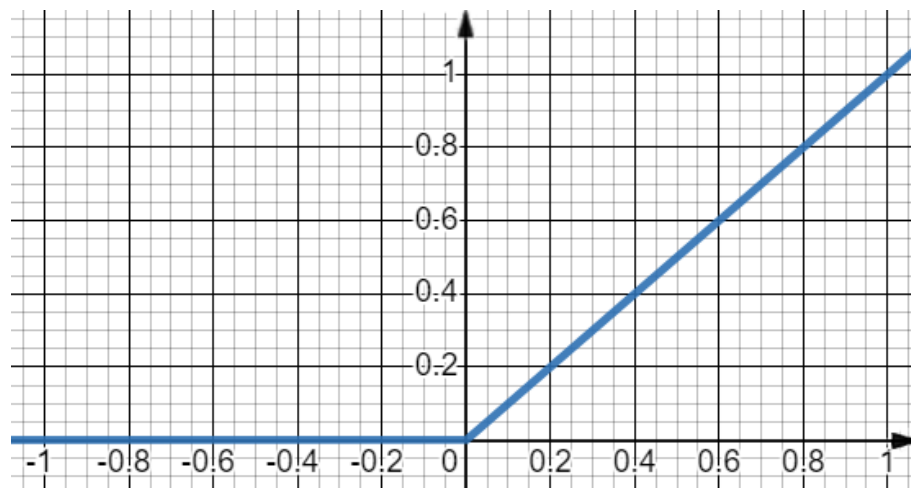


Abb. 4.5. Plot der ReLU. (Quelle: Eigene Darstellung)

Neuronale Netze im Gesundheitswesen

Pfannstiel stellt in [Pfa22, 35, Abb. 1.14] die Rolle der Künstlichen Intelligenz im Gesundheitswesen als Schichtenmodell dar: Künstliche Intelligenz wird umgeben von vier Schichten, von denen jede eine Menge repräsentativer Leistungen enthält. Diese lassen sich grob einteilen in

1. Versorgung des Patienten
2. Herstellung / Zulieferung durch Pharmazie bzw. Medizintechnologie
3. Organisation des Gesundheitswesens
4. Selbstversorgung des Patienten

In jeder dieser Schicht übernimmt die KI wichtige Rollen, z.B. in der Diagnostik (1.), der Wirkstoffentwicklung (2.), der Abrechnung von Leistungen (3.) oder als Software in embedded Systems wie Wearables (4.). Zum Beispiel werden die in Abschnitt 4.1.3 vorgestellten CNNs aufgrund ihrer Erfolge bei der Objektklassifikation¹ in der Diagnostik eingesetzt²: Je nach Verfahren zeigt sich, dass die Netze mindestens genauso gute Diagnosen erstellen wie erfahrene Ärzte, und bessere als wenig erfahrene Ärzte (vgl. [She+19, 7]). Doch auch in anderen Bereichen des Gesundheitswesens finden neuronale Netze praktische Anwendungen. Im Folgenden werden einige vorgestellt.

5.1 Automatisierung in der Gesundheitswirtschaft

Um für die in Deutschland knapp 73 Millionen gesetzlich Krankenversicherten³ Leistungen im Gesundheitswesen abzurechnen, werden durch die Leistungserbringer Finanzdienstleister beauftragt. So leiten bspw. Apotheken eingelöste Kassenre-

¹ vgl. [Ert21c, 330]. *Goodfellow et al.* stellen etwas allgemeiner fest, dass CNNs “gewaltige Erfolge in der Praxis gebracht“ haben [GBC18, 369]. Als Beispiel sei PSPNet genannt, ein CNN zur semantischen Segmentierung von Bilddaten (“*scene parsing*“), das *Zhao et al.* in [Zha+17] vorstellen. Heutzutage betreiben Unternehmen wie Google, Facebook, Microsoft und IBM Forschung und Entwicklung im Bereich der Bildverarbeitung mit CNNs, und im Bereich Hardware entwickeln bspw. NVIDIA und Samsung Chips zur Echtzeiterkennung der Umgebung [LBH15, 440].

² bspw. stellen *Brinker et al.* in [Bri+18] 13 Arbeiten zur Klassifikation von Hautläsionen mit Hilfe von CNNs vor.

³ https://www.gkv-spitzenverband.de/krankenversicherung/kv_grundprinzipien/alle_gesetzlichen_krankenkassen/alle_gesetzlichen_krankenkassen.jsp, abgerufen 12.09.2023

zepte an Apothekenrechenzentren weiter, die die Abrechnungen mit den Krankenkassen durchführen. Obwohl Rezepte heutzutage überwiegend maschinell erstellt werden, kommt es bei den Rechenzentren nach der Digitalisierung der Rezepte (bspw. durch Scannen) häufig zu manueller Nachkorrektur, wenn abrechnungsrelevante Daten nicht vollständig erfasst werden können, wie Höfer *et al.* in [HWN22, 698] feststellen. Um den zusätzlichen Arbeitsaufwand durch Nachkorrekturen zu reduzieren (aufgrund von Störfaktoren wie ausgedünnte Zeichen oder Textüberlagerungen), zeigen Höfer *et al.* ebenda ein neuronales Netz, das in Auftrag eines führenden Abrechnungs- und Finanzdienstleisters erstellt wurde. Ziel des Netzes ist die Rekonstruktion⁴ von Rezepten, die bei Plausibilitätskontrollen auffällig sind.

Ihr neuronales Netz zur Rekonstruktion von Rezepten ist ein CNN mit 7 Faltungsschichten, denen jeweils eine Poolingschicht folgt. Insgesamt werden 40.000 Datensätze für das Training genutzt, sowie 10.000 Validierungsdatensätze. Die Rezepte werden auf den für die Abrechnung relevanten Teil zugeschnitten, sodass die Eingabedaten aus 720×460 Pixeln bestehen. Sie zeigen, dass Ihr Netz bei der Rekonstruktion von Korrekturzeichen eine relative Verbesserung von 27,85% erreicht⁵ und aufgrund dieses Erfolges in den Workflow der zu dem Auftraggeber gehörenden Abrechnungszentren integriert wurde.

5.2 Pharmaforschung

Im Jahr 2016 gelingt es der von *Google DeepMind* entwickelten Software *AlphaGo*, den Südkoreaner Lee Sedol als einen der weltbesten *Go*-Spieler in diesem Spiel zu schlagen⁶. *Go* verfügt über $2,08 \times 10^{170}$ gültige Spielpositionen und gehört damit zu den komplexesten Brettspielen⁷. Kurz nach diesem beachtlichen Erfolg entscheidet *Google DeepMind*, dass das in der Software verwendete 13-lagige CNN-Netz mit seiner 19×19 Eingabe- und Ausgabematrix sowie trainiert mit über 30 Millionen Spielzügen (vgl. [Ert21a, 371]) reif sei für den Einsatz in der Wissenschaft. Im selben Jahr beginnt das Unternehmen mit der Forschung an *AlphaFold*⁸: Die Software soll helfen, das Problem der **Proteinfaltung** zu vereinfachen. Darunter versteht man die Vorhersage der Struktur eines Proteins auf Basis seiner Aminosäuresequenz (vgl. [Dil+08]); mit 10^{300} verschiedenen Konstellationen⁹ ist die Struktur schwer zu berechnen. Der Pharmaindustrie erlaubt das Wissen über die Struktur von Proteinen die Herstellung von Medikamenten, deren Wirkstoffe bspw. Proteinfunktionen aktivieren oder hemmen.

⁴ tatsächlich liegt die Vermutung nahe, das Netz sollte eingesetzt werden, um den Hochleistungs-scanner zu ersetzen. Ziel war es aber, in einem Vorverarbeitungsschritt die Bildqualität zu erhöhen (vgl. [HWN22, 698])

⁵ 5% waren ursprünglich anvisiert (vgl. [HWN22, 711]).

⁶ <https://www.spiegel.de/netzwelt/gadgets/go-duell-software-alphago-siegt-gegen-lee-sedol-a-1081975.html>, abgerufen 08.09.2023

⁷ <https://trompt.github.io/go/legal.html>, abgerufen 08.09.2023

⁸ [Jum+21], s. a. <https://www.deepmind.com/research/highlighted-research/alphafold/timeline-of-a-breakthrough>, abgerufen 08.09.2023

⁹ <https://web.archive.org/web/20110523080407/http://www-miller.ch.cam.ac.uk/levinthal/levinthal.html>, abgerufen 08.09.2023

Großen Erfolg hat die Software 2020 bei dem CASP14-protein-folding-contest. Mit 170.000 Proteinstrukturen trainiert und von ~ 200 GPUs unterstützt erreicht AlphaFold2 Bestwerte, was als Durchbruch für die Medizin bewertet wird (vgl. [Cal20, 204]).

5.3 Diagnostik

Szolovits et al. argumentieren 1988 in [Szo88], dass die Abfrage von Symptomen zur Feststellung von Krankheiten¹⁰ auch von Expertensystemen übernommen werden kann. Es zeigt sich dort ebenfalls, dass Ärzte kritisch gegenüber computergestützten Assistenzsystemen stehen. *Teach und Shortliffe* stellen bereits 1981 fest, dass es medizinischem Fachpersonal wichtig ist, dass das System den Entscheidungsweg zur Diagnose erklären kann; dass der Computer stets die korrekten Diagnosen stellt, wird als weniger wichtig bewertet (vgl. [TS81, 551, “Table V“: “D.1“ sowie “D14“]. In dieser Studie werden auch ethische Bedenken seitens der Mediziner aufgeführt. Dass computergestützte medizinische Assistenzsysteme dann kaum Anwendung im klinischen Umfeld fanden, führen *Lucieri et al.* in [Luc+22] auch auf solche Bedenken zurück: Erst steigende Rechenleistung und die fortschreitende Leistungsfähigkeit tiefer neuronaler Netze weckte das Interesse an KI unter den Medizinern, dank der Erfolge bildbasierter CNNs auch in der Diagnostik (vgl. [Luc+22, 728]). Dem oben erwähnten Wunsch einer **erklärbaren KI (xAI)**¹¹ wird seitdem nachgegangen, was sich aufgrund des *Blackbox-Charakters*¹² insbesondere bei Architekturen mit Millionen von Parametern¹³ in neuronalen Netzen aber als schwierig erweist¹⁴.

Trotz allem ist nicht von der Hand zu weisen, dass neuronale Netze in der Qualität der Diagnosen gleichauf sind mit denen von medizinischem Fachpersonal (vgl. [She+19, 1]). *Amato et al.* listen in [Ama+13] unter anderem Studienergebnisse bzgl. des Einsatzes neuronaler Netze bei der Diagnostik von kardiovaskulären Krankheiten, Tumor-Erkrankungen sowie Diabetes auf, und kommen bei der Auswertung zu dem Ergebnis, dass die neuronalen Netze für eine Vielzahl verschiedener Symptome korrekte Diagnosen erstellen: Zum Beispiel werden Audioaufnahmen der Pumpfähigkeit des Herzens zur Klassifizierung von Herzklappenfehler genutzt, bei dem das eingesetzte Netz - ein MLP mit 3-Schichten unter Verwendung von Backpropagation - eine Erfolgsrate von knapp 95% vorweisen kann.

In [Est+17] trainieren *Estava et al.* ein CNN basierend auf *GoogLeNet Inception v3* mit über 1.000.000 Bildern aus 1000 verschiedenen Kategorien, um danach

¹⁰ bei gleichzeitigem Ausschluss anderer Krankheiten

¹¹ Abkürzung für (engl.) *explainable AI*

¹² <https://spectrum.ieee.org/ai-failures>, abgerufen 09.09.2023

¹³ **AlexNet**: 60.000.000 Parameter [KSH12, 1]

¹⁴ Einige erfolgreiche Ansätze jüngster Zeit fassen *Lucieri et al.* in [Luc+22, 733 ff] zusammen. *Steinwender* geht in [Ste22, 765] auf die Zertifizierungspflicht von Medizinprodukten in der EU ein, und das mit der damit verbundenen Datenschutzgrundverordnung auch eine Interpretierbarkeit bzw. Erklärbarkeit für die KI-Systeme einhergeht

über *transfer learning*¹⁵ mit 129.450 gelabelten Bilddateien aus 2032 verschiedenen Krankheiten ein Netz zur Diagnose von Hautkrebs zu erstellen. Sie zeigen, dass das Netz bei der Korrektheit der gestellten Diagnosen genauere Diagnosen erstellt als ein Mediziner¹⁶.

Irving et al. stellen in [Irv+19] “CheXpert”¹⁷ vor, einen öffentlichen Datensatz mit 224.316 Aufnahmen des Thorax von 65.240 verschiedenen Patienten zur Klassifizierung von 14 Befunden (u.a. Lungenentzündung, Fraktur, Pleuraerguss). Für das Training nutzen sie *DenseNet121*, ein CNN, in dem jede Schicht mit jeder anderen verbunden ist (vgl. [HLW16]). Die Inputdaten bestehen aus 320×320 Pixeln. Bei den Tests zeigt sich, dass ihr Netz bei der Diagnose von Kardiomegalie, Ödemen sowie Pleuraergüssen besser abschneidet als die Radiologen, die für die Studie zum Vergleich eingesetzt wurden.

5.4 Therapie und Prognose

Die HL7-Organisation¹⁸ wurde 1987 mit dem Ziel gegründet, einen Kommunikationsstandard für den elektronischen Datenaustausch im Gesundheitswesen zu etablieren. Teil des Standards ist die *FHIR*-Spezifikation¹⁹, die von *Rajkomar et al.* in [Raj+18] für ein neuronales Netz genutzt wird, das für einen mit diesen Daten verknüpften Patienten bei der Hospitalisierung Vorhersagen über folgende Punkte erstellt:

- Mortalität während des Krankenhausaufenthaltes (“inpatient mortality”)
- ungeplante stationäre Wiederaufnahme (“unexpected readmissions within 30 days”)
- Verlängerung des Aufenthaltes (“long length of stay”)
- Befund bei Entlassung (“discharge diagnoses”)

Validiert wurde das Netz über insg. 46.864.534.945 Datenpunkte: Die von verschiedenen Krankenhäusern zur Verfügung gestellten 216.221 Patientenakten wurden hierzu aufgeteilt in 194.470 Trainingsdaten und 21.751 Testdaten, allesamt im FHIR-Format. Die Autoren des Papiers konnten zeigen, dass ihr Netz die traditionellen klinischen Modelle in den ersten 3 genannten Punkten übertrifft.

Eine ähnliche Arbeit stellen *Choi et al.* in [Cho+16] vor, in der sie untersuchen, ob elektronische Patientendaten genutzt werden können, um mit Hilfe rekurrenter neuronaler Netze Vorhersagen zu Diagnose, Medikation und Wiedervorstellung beim Arzt zu treffen. Sie zeigen, dass ihr Netz als medizinisches Assistenzsystem in den aufgeführten Punkten geeignet ist, und darüber hinaus mittels *transfer*

¹⁵ beim *transfer learning* werden semantisch gleiche Ausgaben für verschiedenklassige Eingaben genutzt (vgl. [GBC18, 602 f.]).

¹⁶ “The deep learning CNN outperforms the average of the dermatologists at skin cancer classification using photographic and dermoscopic images” [Est+17, 3, Figure 2]

¹⁷ <https://stanfordmlgroup.github.io/competitions/chexpert/>, abgerufen 10.09.2023

¹⁸ “Health Level Seven” <https://hl7.org>, abgerufen 11.09.2023

¹⁹ “Fast Healthcare Interoperability Resources”, <https://hl7.org/fhir/>, abgerufen 11.09.2023

learning in Kliniken eingesetzt werden kann, bei denen die Datenmenge nicht für ein Training des Netzes ausreicht.

Prognosemodelle werden auch auf anderer Ebene genutzt, bspw. bei der Evaluierung der korrekten Behandlungsmethode. In [Li+21] stellen *Li et al.* *G-Net* vor, ein Deep Learning Framework für “counterfactual prediction“ als Assistenzsystem für Ärzte. Es soll dabei helfen, unter Berücksichtigung von zur Verfügung stehenden Daten wie Therapie- und Krankheitsverlauf der Patienten die richtige Behandlungsstrategie auszuwählen. Unter “counterfactual prediction“ sind hier alternative Szenarien gemeint, also das Ergebnis einer zunächst in der Theorie eingeschlagenen Therapie, für die von der Software das Behandlungsergebnis ermittelt wird²⁰.

Als Beispiel führen *Li et al.* die Flüssigkeitszufuhr zur Stabilisierung von Intensivpatienten an, die einen septischen Schock erlitten haben. Anwendungszeitpunkt, -art und -menge bestimmen den Genesungsverlauf und können unerwünschte Nebenwirkungen verhindern, allerdings auch solche begünstigen, die bis hin zum Tode führen²¹. Die in *G-Net* verwendeten Netze sollen dabei helfen, in Verbindung von Parametern wie Laborwerte des Patienten sowie Zeitpunkt und Menge der Verabreichung eine Aussage über den Behandlungseffekt zu treffen²², um so die bestmögliche Behandlungsmethode auszuwählen.

²⁰ “This is particularly important in clinical settings, where physicians may have to choose between multiple treatment strategies for their patients but are unable to test all of them before making a decision“ [Li+21, 282]

²¹ vgl. [Sha+20, 5 f.]

²² unter <https://news.mit.edu/2022/deep-learning-technique-predicts-clinical-treatment-outcomes-0224> (abgerufen 10.09.2023) ist eine ausführlichere Pressemitteilung mit Verweis auf das zitierte Papier zu finden

Zusammenfassung und Bemerkung

Ein biologisches neuronales Netz ist ein Kommunikationsverbund, in dem hochkomplexe biochemische Prozesse ineinandergreifen. Diese Prozesse ermöglichen eine Informationsverarbeitung: Eine einzelne Nervenzelle ist in der Lage, Signale zu empfangen, zu verrechnen und ein Signal auszugeben. Löst ein Aktionspotenzial aus, diffundieren Neurotransmitter in den synaptischen Spalt, die Folgezelle integriert die daraus entstehenden elektrischen Signale und so setzt sich die Kette fort, oft bis tausende von Neuronen ein Signal an das Ende der Kommunikationskette weitergeleitet haben.

Die Modellierung künstlicher neuronaler Netze folgt in Teilen dem biologischen Vorbild, und es mag zunächst überraschen, dass einige solcher Verschaltungsmuster - wie die **Divergenz** oder die **rekurrente** oder **laterale Hemmung** - zunächst an Logikgatter erinnern (vgl. [Eil19, 58 f.]). Es liegt nahe, dass die Wissenschaft versucht ist, solche naturgegebenen verrechnenden Einheiten und logischen Verschaltungen zu verstehen und in mathematische Formeln zu gießen, um ein Modell zu erstellen, das menschliche Intelligenz erklärt und nachstellt, und dann als Abstraktionen auf Maschinen übertragen wird: “if you really understand something, you can usually make a machine do it“ [AR88, xiii].

Ein früher Versuch davon wurde in Abschnitt 3.1 mit dem MCP-Modell vorgestellt, ein **empirisches Modell**, das auf Analyse und einfacher Schwellenwertlogik basiert (vgl. [AR88, 16]). Dabei wurde ein neurobiologischer Prozess, der heutzutage bei maschinellen Lernverfahren Standard ist, nicht in dem Modell berücksichtigt: Die dynamische Anpassung des Netzes. Das Rosenblatt-Perzeptron dagegen realisierte eine Art von “Lernen“.

Der frühe Enthusiasmus wird jedoch in den 1970er Jahren durch die Einsicht gebremst, dass insb. das Rosenblatt-Perzeptron nicht auf beliebig komplexe Probleme angewendet werden kann¹. Der Lighthill Report² wird 1973 die britischen Regie-

¹ *Russell und Norvig* fassen zusammen, dass *Minsky und Papert* in [MP88] bewiesen haben, dass ein Perzeptron alles lernen kann, was es auch darstellen kann, aber es könnte halt nur sehr wenig darstellen (vgl. [RN12, 45])

² “Workers entered the field around 1950, and even around 1960, with high hopes that are very far from having been realised in 1972. In no part of the field have the discoveries made so far produced the major impact that was then promised.“ in https://www.chilton-computing.org.uk/inf/literature/reports/lighthill_report/p001.htm, “3 Past disappointments“, abgerufen 28.08.2023

rung dazu bewegen, das Budget für die Forschung an KI zu kürzen (vgl. [RN12, 45]): Neuronale Netze werden als Grundlage künstlicher Intelligenz zunächst verworfen, und Forschungsarbeiten an ihnen geht bis zum Anfang der 1980er Jahre zurück. Diese Periode ist gemeinhin als “KI-Winter“ bekannt.

Die in Kapitel 4 vorgestellten Architekturen und Algorithmen sorgten dann zusammen mit dem technologischen Fortschritt für ein erneutes Aufleben der Forschung an neuronalen Netzen und Künstlicher Intelligenz in den 1980er Jahren. Einige der Ergebnisse dieser Anstrengungen wurden in Kapitel 5 in Auszügen vorgestellt. Dass dabei die Algorithmen nicht immer der Funktionsweise des natürlichen Vorbildes entsprechen, haben wir bei dem Backpropagation-Verfahren gesehen (siehe Abschnitt 4.1.1).

Bei allem Fortschritt, der durch Forschung und Wissenschaft in den vergangenen Jahren erreicht wurde, ist bei den hier vorgestellten Ergebnissen aber vor allem eins deutlich: Die hohe Leistungsfähigkeit einiger dieser Netze kann nur durch Lernen erreicht werden (vgl. [AHR19, 40]), wozu umfangreiche, qualitativ hochwertige Daten benötigt werden.

Nguyen und Patrick stellen in [NP16] fest, dass klinische Daten aufgrund verschiedener Faktoren wie Heterogenität (maschinell / handschriftlich erstellte Daten) und weiterem Rauschen, wie unbekannten Abkürzungen oder Rechtschreibfehlern, als Trainingsdaten für maschinelles Lernen eine besondere Herausforderung darstellen. Normierte Daten können deshalb helfen, die Auswertung, Anwendung aber auch den Austausch derselben zu erleichtern, wie das Beispiel aus Abschnitt 5.4 zeigt (vgl. auch [AHR19, 42]).

Wir dürfen schließen, dass Künstliche Intelligenz in Form neuronaler Netze nicht nur durch komplexe mathematische Modelle und hohe Rechenleistung ermöglicht wird: Sie benötigt zu der erfolgreichen Bewältigung ihrer Aufgaben auch eine gesunde Datenbasis für ihre Lernverfahren³.

³ *Dash et al* fassen in [Das+19] einige wesentliche Punkte zum Thema Big Data im Gesundheitswesen zusammen, ausserdem *Rüping und Sander* in [RS19], die dort feststellen: “Bei der datengetriebenen Entwicklung von Systemen ist zu beachten, dass die Qualität des Ergebnisses zentral von der Qualität der Eingabedaten abhängt.“ [RS19, 19]

Literaturverzeichnis

- [AHR19] Christoph Auer, Nora Hollenstein und Matthias Reumann. „Künstliche Intelligenz im Gesundheitswesen“. In: *Gesundheit digital: Perspektiven zur Digitalisierung im Gesundheitswesen*. Hrsg. von Robin Haring. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, S. 33–46. ISBN: 978-3-662-57611-3. DOI: 10.1007/978-3-662-57611-3_3.
- [Ama+13] Filippo Amato u. a. „Artificial Neural Networks in Medical Diagnosis“. In: *Journal of Applied Biomedicine* 11.2 (Juli 2013), S. 47–58. ISSN: 1214021X, 12140287. DOI: 10.2478/v10136-012-0031-x.
- [AR88] James A. Anderson und Edward Rosenfeld, Hrsg. *Neurocomputing: Foundations of Research*. Cambridge, MA, USA: MIT Press, 1988. ISBN: 0-262-01097-6.
- [Arb00] Michael A. Arbib. „Warren McCulloch’s Search for the Logic of the Nervous System“. In: *Perspectives in Biology and Medicine* 43.2 (2000), S. 193–216. ISSN: 1529-8795. DOI: 10.1353/pbm.2000.0001.
- [Arb03] Michael A. Arbib, Hrsg. *The Handbook of Brain Theory and Neural Networks*. 2nd ed. Cambridge, Mass: MIT Press, 2003. ISBN: 978-0-262-01197-6.
- [BCP18] Mark F. Bear, Barry W. Connors und Michael A. Paradiso. *Neurowissenschaften: Ein Grundlegendes Lehrbuch Für Biologie, Medizin Und Psychologie*. Hrsg. von Andreas K. Engel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. ISBN: 978-3-662-57263-4. DOI: 10.1007/978-3-662-57263-4.
- [Bri+18] Titus Josef Brinker u. a. „Skin Cancer Classification Using Convolutional Neural Networks: Systematic Review“. In: *J Med Internet Res* 20.10 (Okt. 2018), e11936. ISSN: 1438-8871. DOI: 10.2196/11936. PMID: 30333097.
- [Bur+12] Klemens Burg u. a. „Vektorräume Beliebiger Dimensionen“. In: *Höhere Mathematik Für Ingenieure Band II: Lineare Algebra*. Wiesbaden: Vieweg+Teubner Verlag, 2012, S. 75–146. ISBN: 978-3-8348-2267-3. DOI: 10.1007/978-3-8348-2267-3_2.
- [Cal20] Ewen Callaway. „‘It Will Change Everything’: DeepMind’s AI Makes Gigantic Leap in Solving Protein Structures“. In: *Nature* 588 (Dez. 2020), S. 203–204. DOI: 10.1038/d41586-020-03348-4.
- [Cho+16] Edward Choi u. a. „Doctor AI: Predicting Clinical Events via Recurrent Neural Networks“. In: *Proceedings of the 1st Machine Learning for Healthcare Conference*. Hrsg. von Finale Doshi-Velez u. a. Bd. 56. Proceedings of Machine Learning Research. Northeastern University, Boston, MA, USA: PMLR, Aug. 2016, S. 301–318. DOI: 10.48550/arXiv.1511.05942.
- [Das+19] Sabyasachi Dash u. a. „Big Data in Healthcare: Management, Analysis and Future Prospects“. In: *Journal of Big Data* 6.1 (Juni 2019), S. 54. ISSN: 2196-1115. DOI: 10.1186/s40537-019-0217-0.

- [Dil+08] Ken A. Dill u.a. „The Protein Folding Problem“. In: *Annual Review of Biophysics* 37.1 (2008), S. 289–316. DOI: 10.1146/annurev.biophys.37.092707.153558.
- [Eil19] J. Eilers. „Nervenzellen“. In: *Physiologie Des Menschen: Mit Pathophysiologie*. Hrsg. von Ralf Brandes, Florian Lang und Robert F. Schmidt. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, S. 57–64. ISBN: 978-3-662-56468-4. DOI: 10.1007/978-3-662-56468-4_5.
- [Ert21a] Wolfgang Ertel. „Lernen Durch Verstärkung (Reinforcement Learning)“. In: *Grundkurs Künstliche Intelligenz: Eine Praxisorientierte Einführung*. Wiesbaden: Springer Fachmedien Wiesbaden, 2021, S. 351–377. ISBN: 978-3-658-32075-1. DOI: 10.1007/978-3-658-32075-1_10.
- [Ert21b] Wolfgang Ertel. „Maschinelles Lernen Und Data Mining“. In: *Grundkurs Künstliche Intelligenz: Eine Praxisorientierte Einführung*. Wiesbaden: Springer Fachmedien Wiesbaden, 2021, S. 201–283. ISBN: 978-3-658-32075-1. DOI: 10.1007/978-3-658-32075-1_8.
- [Ert21c] Wolfgang Ertel. „Neuronale Netze“. In: *Grundkurs Künstliche Intelligenz: Eine Praxisorientierte Einführung*. Wiesbaden: Springer Fachmedien Wiesbaden, 2021, S. 285–349. ISBN: 978-3-658-32075-1. DOI: 10.1007/978-3-658-32075-1_9.
- [Est+17] Andre Esteva u. a. „Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks“. In: *Nature* 542.7639 (Feb. 2017), S. 115–118. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature21056.
- [Fau94] Laurene Fausett. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. USA: Prentice-Hall, Inc., 1994. ISBN: 0-13-334186-0.
- [FE19] B. Fakler und J. Eilers. „Ruhemembranpotenzial Und Aktionspotenzial“. In: *Physiologie Des Menschen: Mit Pathophysiologie*. Hrsg. von Ralf Brandes, Florian Lang und Robert F. Schmidt. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, S. 65–71. ISBN: 978-3-662-56468-4. DOI: 10.1007/978-3-662-56468-4_6.
- [FMI83] Kunihiko Fukushima, Sei Miyake und Takayuki Ito. „Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition“. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13.5 (1983), S. 826–834. DOI: 10.1109/TSMC.1983.6313076.
- [Fuk75] Kunihiko Fukushima. „Cognitron: A Self-Organizing Multilayered Neural Network“. In: *Biological Cybernetics* 20.3 (Sep. 1975), S. 121–136. ISSN: 1432-0770. DOI: 10.1007/BF00342633.
- [Fuk80] Kunihiko Fukushima. „Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position“. In: *Biological Cybernetics* 36.4 (Apr. 1980), S. 193–202. ISSN: 1432-0770. DOI: 10.1007/BF00344251.
- [GBC18] Ian Goodfellow, Yoshua Bengio und Aaron Courville. *Deep Learning. Das umfassende Handbuch*. 2018. Aufl. mitp Professional. Frechen, Germany: MITP, Okt. 2018. ISBN: 978-3-95845-700-3.

- [HLW16] Gao Huang, Zhuang Liu und Kilian Q. Weinberger. „Densely Connected Convolutional Networks“. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), S. 2261–2269.
- [HW62] D. H. Hubel und T. N. Wiesel. „Receptive Fields, Binocular Interaction and Functional Architecture in the Cat’s Visual Cortex“. In: *The Journal of Physiology* 160.1 (Jan. 1962), S. 106–154. ISSN: 00223751. DOI: 10.1113/jphysiol.1962.sp006837.
- [HWN22] Tobias Höfer, Frederik Weishäupl und Alfred Nischwitz. „Neuronale Netze Zur Effizienzsteigerung Der Texterkennung in Der Rezeptabrechnung“. In: *Künstliche Intelligenz Im Gesundheitswesen: Entwicklungen, Beispiele Und Perspektiven*. Hrsg. von Mario A. Pfannstiel. Wiesbaden: Springer Fachmedien Wiesbaden, 2022, S. 697–714. ISBN: 978-3-658-33597-7. DOI: 10.1007/978-3-658-33597-7_33.
- [Irv+19] Jeremy Irvin u. a. *CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison*. Jan. 2019. arXiv: 1901.07031 [cs, eess].
- [Jon19] P. Jonas. „Aktionspotenzial: Fortleitung Im Axon“. In: *Physiologie Des Menschen: Mit Pathophysiologie*. Hrsg. von Ralf Brandes, Florian Lang und Robert F. Schmidt. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, S. 72–82. ISBN: 978-3-662-56468-4. DOI: 10.1007/978-3-662-56468-4_7.
- [Jum+21] John Jumper u. a. „Highly Accurate Protein Structure Prediction with AlphaFold“. In: *Nature* 596.7873 (Aug. 2021), S. 583–589. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2.
- [Kan+13] E.R. Kandel u. a. *Principles of Neural Science, 5th Edition*. McGraw-Hill’s AccessMedicine. McGraw-Hill Education, 2013. ISBN: 978-0-07-139011-8.
- [KB17] Diederik P. Kingma und Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Jan. 2017. arXiv: 1412.6980 [cs].
- [KSH12] Alex Krizhevsky, Ilya Sutskever und Geoffrey E Hinton. „ImageNet Classification with Deep Convolutional Neural Networks“. In: *Advances in Neural Information Processing Systems*. Hrsg. von F. Pereira u. a. Bd. 25. Curran Associates, Inc., 2012.
- [LBH15] Yann LeCun, Yoshua Bengio und Geoffrey Hinton. „Deep Learning“. In: *Nature* 521.7553 (Mai 2015), S. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539.
- [LeC+89] Y. LeCun u. a. „Backpropagation Applied to Handwritten Zip Code Recognition“. In: *Neural Computation* 1.4 (Dez. 1989), S. 541–551. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.4.541.
- [LeC+98] Y. LeCun u. a. „Gradient-Based Learning Applied to Document Recognition“. In: *Proceedings of the IEEE* 86.11 (1998), S. 2278–2324. DOI: 10.1109/5.726791.
- [LeC89] Y. LeCun. „Generalization and Network Design Strategies“. In: *Connectionism in Perspective*. Hrsg. von R. Pfeifer u. a. Zurich, Switzerland: Elsevier, 1989.

- [Li+21] Rui Li u.a. „G-Net: A Recurrent Network Approach to G-Computation for Counterfactual Prediction Under a Dynamic Treatment Regime“. In: *Proceedings of Machine Learning for Health*. Hrsg. von Subhrajit Roy u.a. Bd. 158. Proceedings of Machine Learning Research. PMLR, Dez. 2021, S. 282–299.
- [Luc+22] Adriano Lucieri u.a. „Erklärbare KI in Der Medizinischen Diagnose – Erfolge Und Herausforderungen“. In: *Künstliche Intelligenz Im Gesundheitswesen: Entwicklungen, Beispiele Und Perspektiven*. Hrsg. von Mario A. Pfannstiel. Wiesbaden: Springer Fachmedien Wiesbaden, 2022, S. 727–754. ISBN: 978-3-658-33597-7. DOI: 10.1007/978-3-658-33597-7_35.
- [MP43] Warren S. McCulloch und Walter Pitts. „A Logical Calculus of the Ideas Immanent in Nervous Activity“. In: *The bulletin of mathematical biophysics* 5.4 (Dez. 1943), S. 115–133. ISSN: 1522-9602. DOI: 10.1007/BF02478259.
- [MP88] Marvin Minsky und Seymour A Papert. *Perceptrons: An Introduction to Computational Geometry*. London, England: MIT Press, 1988.
- [Mur91] Fionn Murtagh. „Multilayer Perceptrons for Classification and Regression“. In: *Neurocomputing* 2.5 (1991), S. 183–197. ISSN: 0925-2312. DOI: 10.1016/0925-2312(91)90023-5.
- [Nov62] A. B. Novikoff. „On Convergence Proofs on Perceptrons“. In: *Proceedings of the Symposium on the Mathematical Theory of Automata*. Bd. 12. New York, NY, USA: Polytechnic Institute of Brooklyn, 1962, S. 615–622.
- [NP16] Hoang Nguyen und Jon Patrick. „Text Mining in Clinical Domain: Dealing with Noise“. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, S. 549–558. ISBN: 9781450342322. DOI: 10.1145/2939672.2939720.
- [Pfa22] Mario A. Pfannstiel. „Einleitung „Künstliche Intelligenz Im Gesundheitswesen““. In: *Künstliche Intelligenz Im Gesundheitswesen: Entwicklungen, Beispiele Und Perspektiven*. Hrsg. von Mario A. Pfannstiel. Wiesbaden: Springer Fachmedien Wiesbaden, 2022, S. 1–47. ISBN: 978-3-658-33597-7. DOI: 10.1007/978-3-658-33597-7_1.
- [Raj+18] Alvin Rajkomar u.a. „Scalable and Accurate Deep Learning with Electronic Health Records“. In: *npj Digital Medicine* 1.1 (Mai 2018), S. 18. ISSN: 2398-6352. DOI: 10.1038/s41746-018-0029-1.
- [RHW86] David E. Rumelhart, Geoffrey E. Hinton und Ronald J. Williams. „Learning Representations by Back-Propagating Errors“. In: *Nature* 323.6088 (Okt. 1986), S. 533–536. ISSN: 1476-4687. DOI: 10.1038/323533a0.
- [RM87] David E. Rumelhart und James L. McClelland. „PDP Models and General Issues in Cognitive Science“. In: *Parallel Distributed Processing*:

- Explorations in the Microstructure of Cognition: Foundations*. 1987, S. 110–146.
- [RN12] Stuart Russell und Peter Norvig. *Künstliche Intelligenz*. 3. Aufl. Pearson Studium - IT. Munich, Germany: Pearson Studium ein Imprint von Pearson Deutschland, Juni 2012. ISBN: 978-3-86894-098-5.
- [Roj93] Raúl Rojas. *Theorie der neuronalen Netze*. Springer Berlin Heidelberg, 1993. DOI: 10.1007/978-3-642-61231-2.
- [Ros57] F. Rosenblatt. *The Perceptron - A Perceiving and Recognizing Automaton*. Techn. Ber. 85-460-1. Ithaca, New York: Cornell Aeronautical Laboratory, Jan. 1957.
- [Ros62] Frank Rosenblatt. *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Bd. 55. Spartan books Washington, DC, 1962.
- [RS19] Stefan Rüping und Jil Sander. „Big Data in Gesundheitswesen und Medizin“. In: *Gesundheit digital: Perspektiven zur Digitalisierung im Gesundheitswesen*. Hrsg. von Robin Haring. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, S. 15–31. ISBN: 978-3-662-57611-3. DOI: 10.1007/978-3-662-57611-3_2.
- [SD07] Stefan Silbernagl und Agamemnon Despopoulos, Hrsg. *Taschenatlas Physiologie*. Georg Thieme Verlag, 2007. DOI: 10.1055/b-002-89576.
- [Sha+20] Zach Shahn u.a. „Fluid-Limiting Treatment Strategies among Sepsis Patients in the ICU: A Retrospective Causal Analysis“. In: *Critical Care* 24.1 (Feb. 2020), S. 62. ISSN: 1364-8535. DOI: 10.1186/s13054-020-2767-0.
- [She+19] Jiayi Shen u.a. „Artificial Intelligence Versus Clinicians in Disease Diagnosis: Systematic Review“. In: *JMIR Med Inform* 7.3 (Aug. 2019), e10010. ISSN: 2291-9694. DOI: 10.2196/10010. PMID: 31420959.
- [She26] Henry M. Sheffer. „Principia Mathematica. Whitehead, Alfred North, Russell, Bertrand“. In: *Isis; an international review devoted to the history of science and its cultural influences* 8.1 (1926), S. 226–231. DOI: 10.1086/358383.
- [Son22] Daniel Sonnet. „Neuronale Netze“. In: *Neuronale Netze Kompakt: Vom Perceptron Zum Deep Learning*. Wiesbaden: Springer Fachmedien Wiesbaden, 2022, S. 17–70. ISBN: 978-3-658-29081-8. DOI: 10.1007/978-3-658-29081-8_2.
- [Ste22] Joachim Steinwendner. „Explainable AI Im Gesundheitswesen“. In: *Künstliche Intelligenz Im Gesundheitswesen: Entwicklungen, Beispiele Und Perspektiven*. Hrsg. von Mario A. Pfannstiel. Wiesbaden: Springer Fachmedien Wiesbaden, 2022, S. 755–767. ISBN: 978-3-658-33597-7. DOI: 10.1007/978-3-658-33597-7_36.
- [Szo88] Peter Szolovits. „Artificial Intelligence in Medical Diagnosis“. In: *Artificial Intelligence* 108.1 (1988). DOI: 10.7326/0003-4819-108-1-80.
- [TS81] Randy L. Teach und Edward H. Shortliffe. „An Analysis of Physician Attitudes Regarding Computer-Based Clinical Consultation Systems“.

- In: *Computers and Biomedical Research* 14.6 (1981), S. 542–558. ISSN: 0010-4809. DOI: 10.1016/0010-4809(81)90012-4.
- [Zha+17] H. Zhao u. a. „Pyramid Scene Parsing Network“. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Juli 2017, S. 6230–6239. DOI: 10.1109/CVPR.2017.660.

Abbildungsverzeichnis

2.1	Schematische Darstellung einer Nervenzelle. (Quelle: in Anlehnung an [SD07, 43, Tafel 2.1])	3
2.2	Ionenverteilung im Zytosol und der EZF. (Quelle: in Anlehnung an [BCP18, 72, Abb. 3.13])	4
2.3	Aktionspotenzial. (Quelle: in Anlehnung an [SD07, 47, Tafel 2.3])	5
3.1	Schematische Darstellung von MCP-Zellen (Quelle: in Anlehnung an [MP43, 105, Figure 1])	8
3.2	Plot einer Treppenfunktion, hier als Heaviside-Funktion mit $H(x) = \{x < 0 : 0, x \geq 0 : 1\}$. (Quelle: Eigene Darstellung)	10
3.3	Ein MCP-Neuron zur Modellierung von OR (Quelle: Eigene Darstellung)	11
3.4	Entwurf für ein MCP-Netz zur Modellierung von XOR (Quelle: Eigene Darstellung)	11
3.5	Rosenblatt-Perzeptron. (Quelle: in Anlehnung an [Roj93, 53, Abb. 3.2])	13
3.6	1. Winkelhalbierende im kartesischen Koordinatensystem (Quelle: Eigene Darstellung)	14
3.7	Halbräume im \mathbb{R}^2 (Quelle: Eigene Darstellung)	15
3.8	Punkte im \mathbb{R}^2 , die nicht durch eine Ursprungsgerade separierbar sind. Angedeutet eine mögliche Trenngerade, die durch $(0, 2)$ geht. (Quelle: Eigene Darstellung)	16
3.9	Perzeptron-Training für grosse Datenmengen. (Quelle: Eigene Darstellung)	18
3.10	XOR im \mathbb{R}^2 . (Quelle: Eigene Darstellung)	19
4.1	Exemplarische Darstellung der Architektur eines einfachen Feed-Forward-Netzes. (Quelle: Eigene Darstellung)	21
4.2	Plot einer Sigmoid-Funktion. (Quelle: Eigene Darstellung)	21
4.3	Skizzierung des Zusammenhangs Fehlerfunktion und lokale Minima (Quelle: in Anlehnung an [Son22, 52, Abb. 2.15])	22
4.4	Beispiel einer Faltungsoperation. (Quelle: in Anlehnung an [GBC18, 372, Abbildung 9.1])	24

4.5	Plot der ReLU. (Quelle: Eigene Darstellung)	25
-----	---	----