Günter Daniel Rey
Karl F. Wender

Neuronale Netze

Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung

3., überarbeitete Auflage

psychlehrbuchplus
hogrefe.de

hogrefe

Neuronale Netze

Neuronale Netze

Günter Daniel Rey, Karl F. Wender

Günter Daniel Rey
Karl F. Wender

Neuronale Netze

Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung

3., überarbeitete Auflage

hogrefe

Prof. Dr. Günter Daniel Rey
Technische Universität Chemnitz
Straße der Nationen 12, Raum 213
09111 Chemnitz
Deutschland
guenter-daniel.rey@phil.tu-chemnitz.de

Prof. Dr. Karl F. Wender
FB 1, Psychologie
Universität Trier
54286 Trier
Deutschland
wender@uni-trier.de

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://www.dnb.de abrufbar.

Anregungen und Zuschriften bitte an:
Hogrefe AG
Lektorat Psychologie
Länggass-Strasse 76
3000 Bern 9
Schweiz
Tel: +41 31 300 45 00
E-Mail: verlag@hogrefe.ch
Internet: http://www.hogrefe.ch

Lektorat: Dr. Susanne Lauri
Herstellung: Daniel Berger
Umschlagabbildung: ©Fabian Beck, Trier
Umschlag: Claude Borer, Riehen
Gestaltung und Druckvorstufe: Günter Daniel Rey
Druck und buchbinderische Verarbeitung: Finidr s.r.o., Český Těšín
Printed in Czech Republik

3., überarbeitete Auflage 2018

(E-Book-ISBN_PDF 978-3-456-95796-8)
ISBN 978-3-456-85796-1
http://doi.org/10.1024/85796-000

Inhaltsverzeichnis

Vorwort zur dritten Auflage

Sieben Jahre nach der zweiten Auflage erscheint zu unserer großen Freude eine neue Auflage unseres einführenden Lehrbuches zu neuronalen Netzen im Hogrefe Verlag. Künstliche neuronale Netze erlangten in den letzten Jahren für zahlreiche Forschungs- und Anwendungsbereiche eine zunehmende Bedeutung. Wir hoffen sehr, mit unserem Lehrbuch interessierten Leserinnen und Lesern den Einstieg in diesen Themenbereich erleichtern zu können.

Für die dritte Auflage haben wir das Lehrbuch abermals überarbeitet und aktualisiert. Dies betrifft vor allem das Kapitel zur Datenauswertung. Hier wurden Text und Abbildungen zu den Programmen Visual-XSel, MemBrain und SPSS auf den neuesten Stand gebracht. Zudem haben wir die Fehlerhinweise von Leserinnen und Lesern zur zweiten Auflage berücksichtigt. Besonders möchten wir uns an dieser Stelle bei Stefanie Dlouhy bedanken, die das gesamte Lehrbuch noch einmal gründlich auf Rechtschreib- und Zeichensetzungsfehler überprüft hat.

Wir wünschen Ihnen mit der dritten Auflage dieses Lehrbuches viel Spaß und Erfolg beim Lesen.

Aschaffenburg und Trier, Herbst 2017 Günter Daniel Rey und Karl F. Wender

Vorwort zur zweiten Auflage

Wir freuen uns sehr, dass aufgrund des Erfolges der ersten Auflage aus dem Jahr 2008 nach nur zweieinhalb Jahren die zweite Auflage des einführenden Lehrbuches zu neuronalen Netzen erscheinen kann. Dabei nutzten wir die Gelegenheit für eine Überarbeitung, Aktualisierung und Erweiterung des Textes. Die Überarbeitung diente in erster Linie der Verbesserung der Lesbarkeit, aber auch die eine oder andere Ungenauigkeit aus der ersten Auflage wurde beseitigt. Dazu berücksichtigten wir – nach gewissenhafter Prüfung – die hilfreichen Anmerkungen von Lesern zur ersten Auflage. Auf Wunsch mehrerer Leser wurden im gesamten Text Randbemerkungen eingefügt und das Seitenformat angepasst. Inhaltsverzeichnis und Überschriften wurden geringfügig umformatiert. Bei diversen Abbildungen haben wir die Unterscheidbarkeit der einzelnen Objekte noch weiter verbessert sowie fehlende Achsenbeschriftungen ergänzt. Darüber hinaus wurde für einen besseren Lesefluss die Anzahl an Querverweisen aus der ersten Auflage reduziert und zusammengehörige Sätze besser zu Absätzen gebündelt. Aktualisiert haben wir vor allem das Kapitel zur Datenauswertung. Dabei wurden Text und Abbildungen zu den Programmen Visual-XSel und MemBrain auf den neuesten Stand gebracht. Erweitert wurde das Lehrbuch im Kapitel 2.6 bei der Darstellung der Backpropagation-Lernregel, im Kapitel 3.5 zur Adaptiven Resonanz Theorie sowie in den Kapiteln 6.2 und 6.5. Letztgenanntes Kapitel beschreibt die Datenauswertung mittels neuronaler Netze in SPSS – einem Statistikprogramm, welches vor allem in der Psychologie sehr häufig zum Einsatz gelangt. Zudem wurden in der zweiten Auflage Exkurse zur Kaskadenkorrelation, zu Pfadanalysen und Strukturgleichungsmodellen sowie zum nichtlinearen Autoenkoder hinzugefügt.

Für die umfangreichen Verbesserungsvorschläge für die zweite Auflage dieses Lehrbuches möchten wir uns in alphabetischer Reihenfolge besonders bei Thorsten Aichele, Fabian Beck, Katrin Bertram, Christina Dippon, Veronika Distl, Andreas Fischer, Isabel Flamme, Thomas Jetter, Verena Jobst, Jan Lenhart, Anne Münzing, Michael Pfleger, Johannes Rodrigues, Johanna Rohr, Ursula Weiland und Simone Wiedenhöft ganz herzlich bedanken. Auch allen anderen Studierenden, die in unseren Seminaren mit ihren Fragen und Diskussionsbeiträgen geholfen haben, die Vermittlung des komplexen Themengebietes zu optimieren, sei vielmals gedankt.

Wir wünschen Ihnen viel Spaß und Erfolg beim Lesen dieses Lehrbuches.

Würzburg und Trier, im Herbst 2010 Günter Daniel Rey und Karl F. Wender

Vorwort zur ersten Auflage

Viele Menschen erliegen der Faszinationskraft neuronaler Netze. Dies mag zahlreiche Ursachen haben. Ein wichtiger Grund, warum mich neuronale Netze begeistern, sind die umfassenden Anwendungsmöglichkeiten inner- und außerhalb der Psychologie. Innerhalb der Psychologie lassen sich neuronale Netze in nahezu allen Teilbereichen Gewinn bringend einsetzen, um menschliches Verhalten und Erleben erklären und besser verstehen zu können. Das vorliegende Lehrbuch kann aus Platzgründen nur einen kleinen Ausschnitt aus diesem vielfältigen Anwendungsspektrum bereitstellen. Neben dem menschlichen Verhalten und Erleben besitzen statistische Datenauswertungen eine sehr große Bedeutung innerhalb der Psychologie. Auch hierfür können neuronale Netze neue Impulse liefern, wobei dieses Lehrbuch eine anwendungsorientierte Einführung in die Datenauswertung mittels neuronaler Netze zur Verfügung stellt.

Die Inhalte des Lehrbuches basieren auf einem Computerprogramm, welches im Rahmen meines Forschungspraktikums an der Universität Trier bei Karl F. Wender entstanden ist. Das dort entwickelte multimediale Lernprogramm diente als Ergänzung zu unseren zahlreichen Lehrveranstaltungen über neuronale Netze, wurde per CD an die Seminarteilnehmer verteilt und von diesen gründlich evaluiert. Die darauf aufbauende, deutlich umfangreichere Online-Version wurde gemeinsam mit Fabian Beck erstellt, der sich primär für die technische Umsetzung und Gestaltung der Homepage verantwortlich gezeigt hat. Die Internetseite kann nach wie vor unter www.neuronalesnetz.de abgerufen werden. Das Ihnen nun vorliegende Lehrbuch wurde auf Basis des – mit zahlreichen Illustrationen versehenen – Lerntextes der Internetversion verfasst. Hierbei erfolgte nicht nur eine sorgfältige Modifikation und Aktualisierung, sondern auch eine sehr starke inhaltliche Erweiterung, sodass der Umfang des Lehrbuches im Vergleich zu den vorangegangenen Manuskripten deutlich zugenommen hat.

Das Lehrbuch soll primär Studierende der Psychologie im Hauptstudium ansprechen, die sich entweder für kognitionswissenschaftliche Fragestellungen interessieren und/oder quantitative Datenauswertungen mittels neuronaler Netze vornehmen möchten. Darüber hinaus ist das Buch auch für zahlreiche Studierende anderer Fachbereiche geeignet. Hier sind vornehmlich Kognitions- und Neurowissenschaften, Soziologie, Biologie und Psychobiologie, Geowissenschaften sowie Informatik, Mathematik und Statistik zu nennen. Auch Dozenten, die bisher noch nicht oder nur wenig vertraut mit dem Themengebiet sind, soll der Einstieg in die Grundlagen und Anwendungen neuronaler Netze sowie der damit verbundenen Datenauswertung erleichtert werden. Insbesondere in den Sozialwissenschaften werden Datenanalysen zumeist auf Basis des Allgemeinen Linearen Modells (ALM) durchgeführt, während zahlreiche Forscher die innovative Auswertung mittels neuronaler Netze bislang eher selten einsetzen.

Aus Gründen der leichteren Lesbarkeit wird in diesem Lehrbuch durchgängig die männliche Form verwendet. Es sind jedoch selbstverständlich stets Frauen und Männer gemeint!

Für die umfangreichen Fehlerkorrekturen an diesem Lehrbuch möchte ich mich in alphabetischer Reihenfolge insbesondere bei Katrin Arens, Hans Bauer, Fabian Beck, Ricarda Bergmann, Florian Buchwald, Sabrina Ehses sowie Patricia Feith ganz herzlich bedanken. Auch allen Studierenden, die in meinen Seminaren zu neuronalen Netzen mit ihren Fragen und Diskussionsbemerkungen dazu beigetragen haben, die Vermittlung des komplexen Themengebietes zu optimieren, sei vielmals gedankt. Ebenfalls bedanken darf ich mich bei Fabian Beck für die Erstellung des schönen Umschlagbildes. Nicht zuletzt sei meiner Freundin Denise Reimnitz, meinem Bruder und meinen Eltern sowie allen Freunden und Bekannten gedankt, die mich immer unterstützt und dazu beigetragen haben, dass dieses Projekt realisiert werden konnte. Ich freue mich sehr darüber, dass ich dieses Lehrbuch schreiben konnte und durfte.

Trotz intensiver Korrekturarbeiten ist es wahrscheinlich, dass das Lehrbuch nicht frei von Fehlern ist. Hinweise zu diesen sowie sonstige Verbesserungsvorschläge nehme ich sehr dankend per E-Mail (GuenterDanielRey@web.de) entgegen.

Ich wünsche Ihnen nun viel Spaß und Erfolg beim Lesen dieses Lehrbuches.

Trier, im Frühjahr 2008 Günter Daniel Rey

Dieses Buch gibt eine Einführung in die Theorie und die Anwendung künstlicher neuronaler Netze. Dabei wird mehr Wert auf ein intuitives, anschauliches Verständnis als auf die Herleitung mathematischer Zusammenhänge gelegt. Künstliche neuronale Netze – im Folgenden neuronale Netze – haben in den vergangenen Jahren eine enorme, interfakultative Verbreitung gefunden. Ursprünglich waren sie entwickelt worden, um die neuronale Aktivität des menschlichen Nervensystems nachzubauen (McCulloch & Pitts, 1943). Dies gehört zwar heute auch noch zu den Zielsetzungen, aber längst nicht mehr in allen Fällen. In vielen Anwendungen werden durchaus psychische Prozesse simuliert, die Funktion der einzelnen Elemente ist dabei aber oftmals nicht den natürlichen Neuronen nachempfunden. Bei zahlreichen Anwendungen kommt es jedoch nicht auf die Ähnlichkeit zum natürlichen Nervensystem an, sondern um den Einsatz besonders effizienter Methoden zur Datenauswertung. Hierzu gehören Anwendungen in den Wirtschafts- und Ingenieurswissenschaften ebenso wie in der Geologie und Geographie sowie in der Biologie, Medizin und der Physik.

Der große Vorteil der neuronalen Netze liegt in ihrer Flexibilität und Anpassungsfähigkeit. Dies kann natürlich bei einer speziellen statistischen Anwendung zum Nachteil werden. Die große Anpassungsfähigkeit legt allerdings die Vermutung nahe, dass sie aus der strukturellen Ähnlichkeit mit dem menschlichen Nervensystem resultiert. Denn tatsächlich sind Menschen per se ja auch keine guten Statistiker. Neuronale Netze sind ursprünglich häufig in psychologischen Zusammenhängen entwickelt worden. Das vorliegende Buch richtet sich hauptsächlich an Psychologen. Es werden dementsprechend Modelle dargestellt, durch die psychische Prozesse simuliert werden sollen. Darüber hinaus wird aber auch auf die Datenauswertung eingegangen, bei der neuronale Netze als Alternative zu statistischen Verfahren (z. B. Regressions- bzw. Varianzanalysen) eingesetzt werden. Dies scheinen Anwendungsbereiche zu sein, die in naher Zukunft noch mehr Verbreitung finden könnten.

Trier, im Frühjahr 2008 Karl F. Wender

1 Grundlagen

1.1 Übersicht und Lernziele

Das erste Kapitel liefert einen kurzen Überblick über die Anwendungsbereiche neuronaler Netze und stellt dar, aus welchen Elementen diese Netze bestehen und wie sie miteinander verknüpft sind. Daneben wird die Funktionsweise von Neuronen beschrieben und es werden verschiedene Phasen und Darstellungsarten neuronaler Netze aufgeführt.

Folgende Lernziele sind Bestandteil dieses Kapitels:

- Welche Gemeinsamkeiten besitzen verschiedene neuronale Netze?
- Aus welchen Elementen bestehen neuronale Netze?
- Wie werden eintreffende Informationen in Neuronen weiterverarbeitet?
- Welche Phasen sind bei neuronalen Netzen zu unterscheiden?

1.2 Einleitung

Keine allgemein anerkannte Definition

Eine allgemein anerkannte Definition zu neuronalen Netzen existiert in der Literatur unseres Wissens *nicht*. Dies liegt wohl u. a. daran, dass es nicht *das* neuronale Netz gibt, sondern neuronale Netze vielmehr als Oberbegriff zu verstehen sind (z. B. Poddig & Sidorovitch, 2001). Die spezifischen, z. T. sehr heterogenen Netztypen, die in Kapitel 3 detailliert erörtert werden, lassen sich nur schwer in einer bündigen Definition zusammenfassen.

Gemeinsamkeiten neuronaler Netze

Allen neuronalen Netzen gemeinsam ist aber, dass sie Informationen erhalten, verarbeiten und ausgeben, wobei sich das Netz während der Verarbeitung umstrukturieren kann:

- **Informationsaufnahme:** Zunächst werden dem Netz (wiederholt) Informationen in Form von Zahlen als Eingabe zur Verfügung gestellt. Diese Zahlen sollen zumeist einen „Realitätsausschnitt" abbilden, beispielsweise Gesichter von Personen.
- **Informationsverarbeitung und Netzmodifikation:** Mithilfe dieser „Zahlenbündel" – man spricht in diesem Zusammenhang von Vektoren und Matrizen – wird das Netz verändert. Auch das neuronale Netz selbst kann als „Zahlenbündel" bzw. in Form von Matrizen repräsentiert werden. Die Modifikation dieses Zahlengebildes erfolgt in Kombination mit den aufgenommenen Informationen und einer Umformungsregel, der sogenannten Lernregel (Kapitel 2). Die Veränderung des Netzes findet typischerweise nicht in einem einzigen, sondern in einer Vielzahl von Schritten statt, wobei die dazu

notwendigen, oftmals sehr umfangreichen (Matrizen-)Berechnungen an Computern vorgenommen werden. Während und nach den Berechnungen zur Umformung des Netzes durchlaufen Informationen das neuronale Netz. Diese Zahlen werden durch das Netz modifiziert und verlassen dieses abschließend wieder – ebenfalls in Form einer Zahlenausgabe.

- **Informationsausgabe:** Die Informationsausgabe stellt die „Antwort" des Netzes auf die vorangegangene Eingabe dar. Beispielsweise kann ein Gesicht einer Person einem bestimmten Namen zugeordnet werden oder aber das Netz „erkennt", ob es sich um ein fröhliches oder trauriges Gesicht handelt.

Künstliche vs. biologische neuronale Netze

Im Gegensatz zu anderen mathematischen Verfahren (z. B. der multiplen Regression), die ebenfalls auf Matrizenberechnungen zurückgreifen, unterscheiden sich neuronale Netze u. a. darin, dass das menschliche Gehirn ursprünglich als „Vorbild" bei der Erstellung diente. Dieser Aspekt steht bei heutigen Arbeiten zu neuronalen Netzen jedoch häufig nicht mehr im Vordergrund (Poddig & Sidorovitch, 2001). Zu beachten ist, dass im vorliegenden Buch mit neuronalen Netzen *kein* biologisches Netz gemeint ist, wie beispielsweise im menschlichen Gehirn, sondern ein *künstliches* neuronales Netz am Computer simuliert wird.

Historischer Abriss

Unter den Ersten, die sich mit dem Themenbereich neuronale Netze beschäftigten, waren Warren McCulloch und Walter Pitts (1943) mit ihrem Formalmodell des Neurons. Die Arbeiten mit und zu neuronalen Netzen haben seit ca. 1986 stark zugenommen. Es liegen inzwischen zahlreiche wissenschaftliche Zeitschriften vor, die sich primär mit diesem Thema auseinandersetzen, wie die „IEEE Transactions on Neural Networks" oder das Journal „Neural Networks".

Anwendungsgebiete

Mittlerweile existieren unzählige Anwendungsfelder zu neuronalen Netzen. Dabei kann eine Klassifikation in zwei große Themenbereiche vorgenommen werden:

- **Modellierung menschlichen Verhaltens und Erlebens:** Neuronale Netze, die modelliert werden, um menschliches Verhalten und Erleben bzw. die diesen zugrundeliegenden Gehirnprozesse zu simulieren und dadurch besser zu verstehen.
- **Lösung konkreter Anwendungsprobleme:** Neuronale Netze, die dazu dienen, konkrete Anwendungsprobleme aus Bereichen wie z. B. Statistik, Wirtschafts- und Ingenieurwissenschaften, Informatik und vielen anderen Gebieten zu lösen.

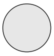
1.3 Units und ihre Verbindungen

Units

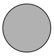
Neuronale Netze bestehen aus mehreren Neuronen. Diese Neuronen werden auch als Units, Einheiten oder Knoten bezeichnet. Sie dienen in der Regel dazu, Informationen aus der Umwelt oder von anderen Neuronen als Zahlenwerte aufzunehmen und an andere Units oder die Umwelt in modifizierter Form weiterzuleiten.

Arten von Units

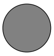
Man unterscheidet zwischen drei verschiedenen Arten von Neuronen:



- **Input-Units:** Units, die von der Außenwelt Signale (Reize, Muster) in Form von Zahlen erhalten.



- **Hidden-Units:** Units, die sich zwischen Input- und Output-Units befinden.



- **Output-Units:** Units, die Signale als Zahlenwerte an die Außenwelt ausgeben.

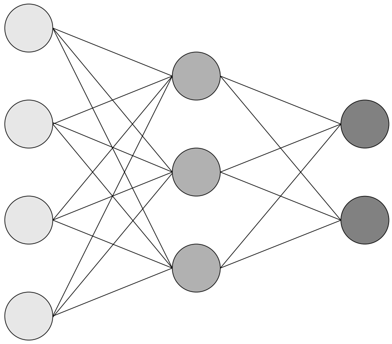


Abb. 1: Schematische Darstellung eines neuronalen Netzes.

Informationsausbreitung in Leserichtung

In Abb. 1 sowie in allen weiteren im Buch dargestellten Visualisierungen zu neuronalen Netzen verläuft die Informationsausbreitung – wenn nicht anders angegeben – in Leserichtung, d. h. von links (Input-Units) nach rechts (Output-Units).

Schichten bzw. Layer

„Übereinander" angeordnete Knoten (z. B. die beiden rechts befindlichen Output-Units in Abb. 1) fasst man als Schicht bzw. Layer zusammen. Während in neuronalen Netzen in aller Regel nur jeweils *eine* Input- und Output-Schicht vorhanden ist, kann ein Netz gar keine, eine oder mehrere Hidden-Schicht(en) enthalten. Hornik, Stinchcombe und White (1989) konnten zeigen, dass sämtliche Probleme, die mit mehreren Hidden-Schichten (auch als versteckte Schichten bezeichnet) lösbar sind, auch durch Netze mit nur einem einzigen Hidden-Layer bewältigt werden können. Dies gilt jedoch nur, wenn diese Schicht eine hinreichend große Anzahl an Neuronen aufweist. Dass die Lösung bestimmter Probleme zumeist den Einsatz mindestens einer Hidden-Schicht verlangt, wird im zweiten Kapitel erörtert.

Verbindungen

Units sind miteinander durch Kanten (auch Links genannt) verbunden. Die Stärke einer solchen Verbindung wird durch ein Gewicht ausgedrückt. Je größer der Absolutbetrag des Gewichts ist, desto größer ist der Einfluss einer Einheit auf eine andere.

Arten von Gewichten

- Ein **positives Gewicht** bringt zum Ausdruck, dass ein Neuron auf ein anderes einen exzitatorischen, d. h. erregenden Einfluss ausübt.
- Ein **negatives Gewicht** bedeutet, dass die Beeinflussung inhibitorisch, also hemmender Natur ist.
- Ein **Gewicht von Null** besagt, dass eine Unit auf eine andere aktuell keine Wirkung ausübt.

Wissen und Lernen

Das „Wissen" eines neuronalen Netzes ist typischerweise in seinen Gewichten gespeichert, wobei Lernen hier zumeist als Gewichtsveränderung zwischen den Einheiten definiert wird (vgl. Kapitel 2). Damit sind die Verbindungen und deren Gewichte zwischen den einzelnen Units bei neuronalen Netzen von zentraler Bedeutung. Wie das Lernen genau erfolgt, ist abhängig von der verwendeten Lernregel (Kapitel 2).

Eingabe-informationen

Ob der Lernprozess in neuronalen Netzen erfolgreich ist, hängt entscheidend davon ab, welche und wie die Informationen dem Netz dargeboten werden. Die Ermittlung der „richtigen“ Präsentation erfolgt oftmals durch simples Ausprobieren und mithilfe der Erfahrung, die der „Architekt“ des neuronalen Netzes besitzt. Die früher teilweise vertretene Auffassung, dass man den Eingabeinformationen keinerlei Bedeutung schenken müsse, da das Netz die korrekte „Lösung“ selbstständig findet, trifft *nicht* zu. Man spricht in diesem Zusammenhang auch von der Regel „garbage in, garbage out“ (Poddig & Sidorovitch, 2001).

1.4 Funktionsweise von Units

Überblick

Innerhalb einer einzelnen Unit werden Informationen von anderen Neuronen wie folgt verarbeitet, um anschließend einen Output an andere Units oder die Umwelt weiterzuleiten:

1. Berechnung der einzelnen Inputwerte
2. Bildung des Netzinputs mithilfe der einzelnen Inputwerte
3. Zuordnung des Netzinputs zu einem Aktivitätslevel
4. Erzeugung des Outputs bzw. der Ausgabe aus dem Aktivitätslevel

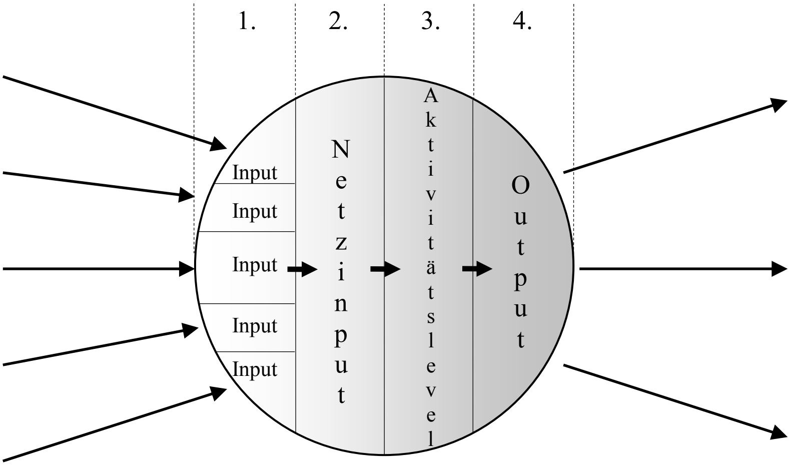


Abb. 2: Schematische Darstellung der Funktionsweise einer Unit.

Wichtige Hinweise

Der Input und Output einer einzelnen Unit ist dabei *nicht* mit den Input- und Output-Units zu verwechseln. Zudem gelten die in Abb. 2 dargestellten Schritte nur für Hidden- und Output-Units. Bei den Input-Units und den noch vorzustellenden Bias-Units werden Informationen aus der Umwelt direkt und ohne weitere Verarbeitung an die nachfolgende Schicht gesendet. Daher betrachten manche Bücher und Computerprogramme wie beispielsweise MATLAB die Input-Schicht *nicht* als eigenständigen Layer.

1.4.1 Input und Netzinput

Der Input (bzw. die Eingabe), den ein Neuron von einer anderen Unit empfängt, hängt von zwei Werten ab, die zumeist multiplikativ miteinander verknüpft sind: Input

- dem **Output** (bzw. der Ausgabe) der sendenden Einheit und
- dem **Gewicht** zwischen den beiden Neuronen

Je stärker der Outputbetrag der sendenden Einheit und je höher der Betrag des Gewichts zwischen den beiden Units ist, desto größer ist der Einfluss (Input) auf die empfangende Einheit. Ist einer der beiden Terme gleich Null, so ist kein Einfluss vorhanden. Auswirkungen

Der Input, den ein Neuron i von einem anderen Neuron j erhält, lässt sich dabei auch als Formel darstellen: Formel

$$input_{ij} = a_j \cdot w_{ij}$$

Dabei gilt:

- i = empfangende Unit
- j = sendende Unit
- a_j = Output bzw. Ausgabe der sendenden Unit j
- w_{ij} = Gewicht zwischen der sendenden Unit j und der empfangenden Unit i

Bei Input und Gewichten hat sich die kontraintuitive Konvention eingebürgert, dass die erste Indexstelle die empfangende Einheit repräsentiert, während der zweite Index sich auf die sendende Unit bezieht. Kontraintuitive Konvention

Angenommen, die Ausgabe einer sendenden Einheit beträgt –3, während das Gewicht zwischen sendender und empfangender Unit bei –4 liegt. In diesem Fall führt die Multiplikation der beiden Werte ((–3) · (–4)), zu einem Input von +12. Beispiel

Der gesamte Input einer Unit wird Netzinput, Netzeingabe, Netinput oder häufig auch Netto-Input genannt. Die sogenannte Propagierungsfunktion bestimmt dabei, wie sich der Netzinput aus den einzelnen Inputwerten ermittelt. Netzinput und Propagierungsfunktion

Sehr häufig wird als Propagierungsfunktion die Linearkombination, d. h. die gewichtete Summe herangezogen. Hierbei werden zunächst die einzelnen Inputs durch Multiplikation der Ausgaben der sendenden Einheiten mit den entsprechenden Gewichten zwischen den sendenden Neuronen und der empfangenden Unit berechnet. Anschließend erfolgt die Addition sämtlicher Inputs, welche die Einheit von anderen Neuronen erhalten hat. Linearkombination als Propagierungsfunktion

Eine solche Propagierungsfunktion kann auch als Formel dargestellt werden: Formel

$$netinput_i = \sum_j input_{ij} = \sum_j a_j \cdot w_{ij}$$

Dabei gilt:

- i = empfangende Unit
- j = sendende Unit
- a_j = Output bzw. Ausgabe der sendenden Unit j

- w_{ij} = Gewicht zwischen der sendenden Unit j und der empfangenden Unit i

Beispiel

Abb. 3 dient als Beispiel zur Berechnung des Netzinputs. Vier sendende Units mit den Ausgaben +3, +5, –3 und 0 sind durch vier unterschiedliche Gewichte (+5, +1, +6 und –4) mit der empfangenden Unit verbunden. Zur Berechnung des Netzinputs erfolgt in einem ersten Schritt die Multiplikation zwischen den Ausgaben der sendenden Einheiten und den entsprechenden Gewichten zwischen den sendenden und der empfangenden Unit. Im zweiten Schritt werden diese Werte aufsummiert.

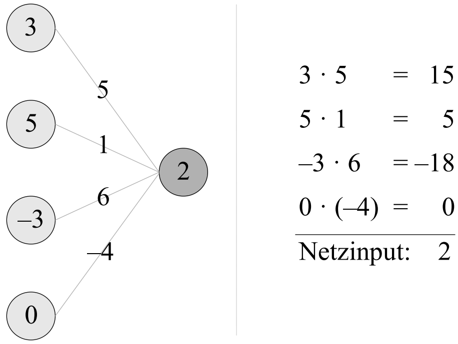


Abb. 3: Beispiel zur Berechnung des Netzinputs.

1.4.2 Aktivitätsfunktion

Aktivitätsfunktion

Nach Bildung des Netzinputs wird dieser mithilfe der Aktivitätsfunktion einem Aktivitätslevel zugeordnet. Die Aktivitätsfunktion – auch Transfer- oder Aktivierungsfunktion genannt – kann in einem zweidimensionalen Diagramm visualisiert werden, wobei auf der x-Achse der Netzinput der Einheit und auf der y-Achse der[1] entsprechende Aktivitätslevel abgetragen wird (Abb. 4).

Arten von Aktivitätsfunktionen

In der Literatur zu neuronalen Netzen finden sich zahlreiche Aktivitätsfunktionen. Typischerweise wird in den meisten neuronalen Netzen einheitlich *eine* bestimmte Aktivitätsfunktion für sämtliche Neuronen einer Schicht, manchmal sogar des gesamten Netzes verwendet. Nachfolgend sollen beispielhaft einige ausgewählte Funktionen näher beschrieben werden:

- **Lineare Aktivitätsfunktion:** Bei der linearen Aktivitätsfunktion ist der Zusammenhang zwischen Netzinput und Aktivitätslevel linear (Abb. 4 links oben). Der Wertebereich für den Aktivitätslevel ist folglich sowohl nach oben als auch nach unten hin unbeschränkt. Einen Spezialfall der linearen Aktivitätsfunktion stellt die Identitätsfunktion dar, bei welcher der Netzinput dem Aktivitätslevel entspricht.
- **Lineare Aktivitätsfunktion mit Schwelle:** Eine Sonderform der linearen Aktivitätsfunktion ist die lineare Aktivitätsfunktion mit Schwelle. Bevor der Zusammenhang zwischen den beiden Größen linear wird, muss eine zuvor festgelegte Schwelle überschritten werden. Schwellen können nicht nur bei der linearen Aktivitätsfunktion, sondern auch bei allen anderen Aktivitäts-

[1] *Der* Aktivitätslevel ist die korrekte Bezeichnung, wenngleich die Mehrheit der Studierenden sich in zahlreichen Seminarumfragen für *das* Aktivitätslevel ausgesprochen hat.

funktionen verwendet werden. Der Wertebereich des Aktivitätslevels ist im vorliegenden Fall nur nach unten begrenzt (Abb. 4 rechts oben).

- **Binäre Aktivitätsfunktion:** Bei der binären Schwellenwertfunktion – auch Heaviside-Funktion oder Schrittfunktion genannt – existieren lediglich zwei Zustände des Aktivitätslevels, 0 (bzw. manchmal auch –1) oder +1 (Abb. 4 links im mittleren Bereich).
- **Sigmoide Aktivitätsfunktion:** Diese Aktivitätsfunktion wird in den meisten Anwendungsfällen verwendet. Es kann hier zwischen der logistischen und der Tangens Hyperbolicus Aktivitätsfunktion unterschieden werden (z. B. Macho, 2002):
    - **Logistische Aktivitätsfunktion:** Bei der logistischen Aktivitätsfunktion, die man auch Fermifunktion nennt, ist der Wertebereich auf 0 bis +1 begrenzt. Ist der Netzinput, bezogen auf den Betrag, groß und negativ (z. B. –100), dann ist der Aktivitätslevel nahe 0. Dieser steigt mit zunehmendem Netzinput zunächst langsam an (eine Art Schwelle), wobei der Anstieg immer steiler wird und zwischenzeitlich einer linearen Funktion gleicht. Bei einem hohen Netzinput nähert sich der Wert dann asymptotisch der +1 an (Abb. 4 rechts im mittleren Bereich).
    - **Tangens Hyperbolicus Aktivitätsfunktion:** Diese Funktion nimmt einen ähnlichen Verlauf wie die logistische Aktivitätsfunktion an. Der Wertebereich liegt hier jedoch zwischen –1 und +1 (Abb. 4 links unten).
- **Normalverteilte Aktivitätsfunktion:** Eine normalverteilte Aktivitätsfunktion (Abb. 4 rechts unten) wird nur in seltenen Fällen verwendet (Poddig & Sidorovitch, 2001). Beim Statistikprogramm Visual-XSel (Kapitel 6.3) kann diese Aktivitätsfunktion neben drei anderen Funktionen ausgewählt werden.

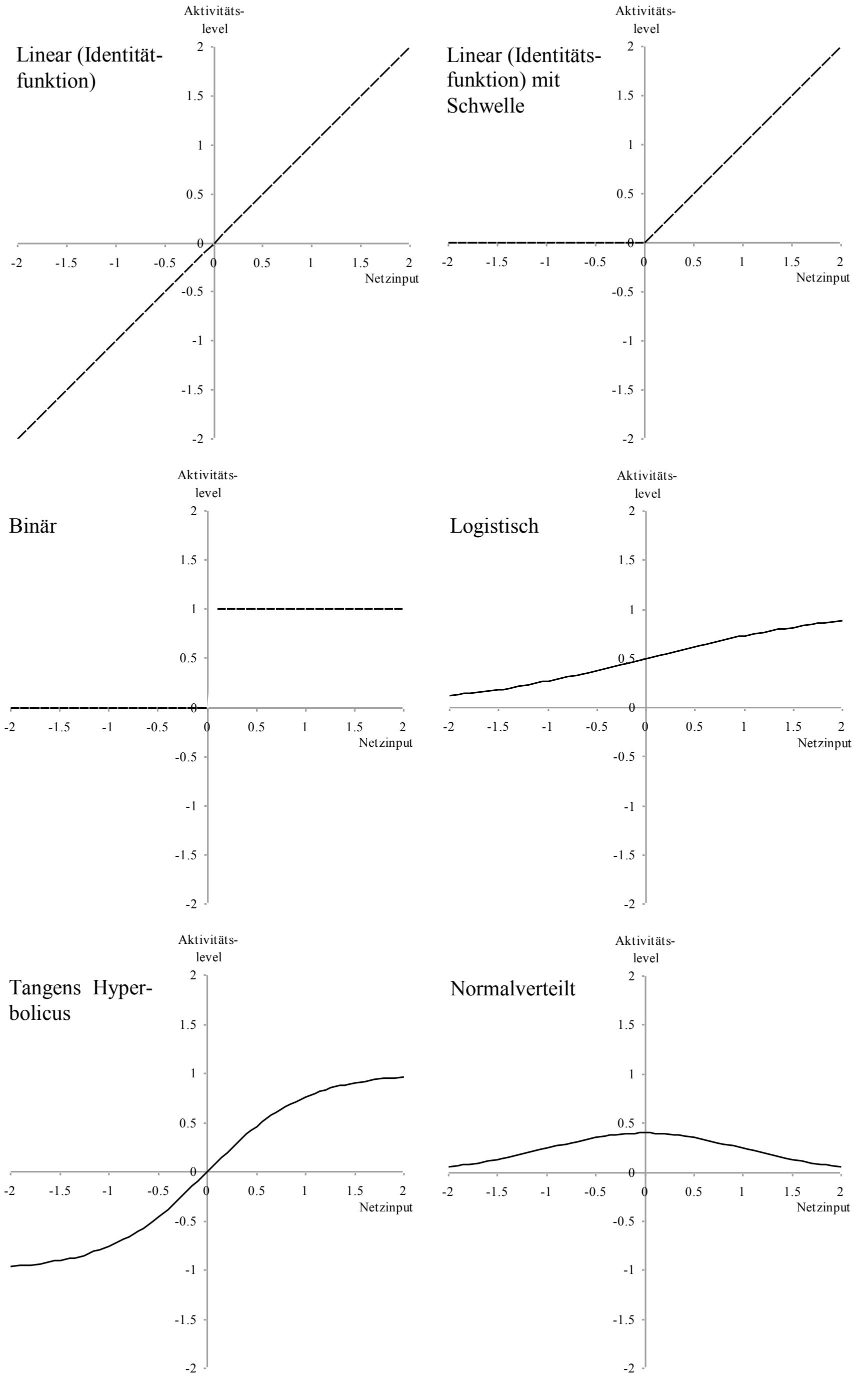


Abb. 4: Schematische Darstellung sechs verschiedener Aktivitätsfunktionen.

Der nachfolgende Exkurs über die Formeln zu den einzelnen Aktivitätsfunktionen soll als Hilfestellung bei der Erstellung eines eigenen neuronalen Netzes dienen. Sofern die Ableitungen der Aktivitätsfunktionen benötigt werden, können diese beispielsweise auf den Webseiten www.mathetools.de/differenzieren/ oder www.wolframalpha.com ermittelt werden.

Exkurs: Formeln zu den einzelnen Aktivitätsfunktionen

Identitätsfunktion (Spezialfall): $a_i = netinput_i$

Lineare Aktivitätsfunktion mit Schwelle 0:

$$a_i = \begin{cases} m \cdot netinput_i + b, \text{ falls } netinput_i \geq 0 \\ 0, \text{ falls } netinput_i < 0 \end{cases}$$

Binäre Aktivitätsfunktion mit Schwelle 0:

$$a_i = \begin{cases} 1, \text{ falls } netinput_i \geq 0 \\ 0, \text{ falls } netinput_i < 0 \end{cases}$$

Logistische Aktivitätsfunktion: $a_i = \dfrac{1}{1 + e^{-netinput_i}}$

Tangens Hyperbolicus Aktivitätsfunktion:

$$a_i = \tanh(netinput_i) = \frac{e^{netinput_i} - e^{-netinput_i}}{e^{netinput_i} + e^{-netinput_i}}$$

Normalverteilte Aktivitätsfunktion: $a_i = \dfrac{1}{\sqrt{2 \cdot \pi}} \cdot e^{\frac{-netinput_i{}^2}{2}}$

Dabei gilt:

a_i	= Aktivitätslevel der Unit i
$netinput_i$	= Netzinput der Unit i
m	= „Steigung“ der Geraden
b	= „Achsenschnittpunkt“ der Geraden
e	≈ 2.718 (Euler'sche Zahl)
π	≈ 3.142 (Pi)

Die unterschiedlichen Aktivitätsfunktionen besitzen verschiedene Eigenschaften:

Eigenschaften der Aktivitätsfunktionen

- **Begrenzung des Aktivitätslevels:** Im Gegensatz zu den linearen Aktivitätsfunktionen ist der Aktivitätslevel bei der binären, sigmoiden und normalverteilten Aktivitätsfunktion sowohl nach oben als auch nach unten begrenzt. Dies ist biologisch plausibel und verhindert, dass die Aktivität im Netz durch rekurrente Verbindungen (Kapitel 3.4) ungewollt „überläuft“ und dadurch nur noch Fehlerwerte produziert werden. Die binäre Aktivitätsfunktion im Speziellen leitet zudem sehr geringe Netzinput-Werte, die man als „Rauschen“ betrachten könnte, nicht als „Signal“ weiter (vgl. mit der Signalentdeckungstheorie). Stattdessen wird dieses Rauschen unterdrückt. Dieser potentielle Vorteil kann allerdings durch geeignete Parameterwahl auch mit anderen begrenzten Aktivitätsfunktionen erreicht werden (siehe Exkurs: Formeln zu den einzelnen Aktivitätsfunktionen). Die Begrenzung des Aktivitätslevels besitzt andererseits den Nachteil, dass nur ein sehr eng umfasster Wertebereich (z. B. 0 bis 1) als Ausgabe produziert werden kann. Vorausgesetzt, man passt die oben aufgeführte Formel nicht entsprechend an. Aufgrund des eingeschränkten Wertebereichs wird in neuronalen Netzen häufig in den Hidden-Schichten auf sigmoide Aktivitätsfunktionen zurückgegriffen, während die Output-Units mit linearen Aktivitätsfunktionen ausgestattet sind. Ein derartig zusammengestelltes Netz kann jede denkbare Funktion beliebig gut approximieren, d. h. sich ihr annähern. Alternativ können die Aktivitätslevel auch mithilfe der Ausgabefunktion (Kapitel 1.4.3) abschließend transformiert werden. Ebenfalls denkbar ist die nachträgliche Umwandlung der Ausgabewerte mittels einer vorher festgelegten Umformungsregel.

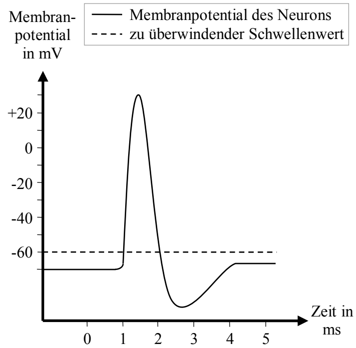


Abb. 5: Schematische Darstellung des zeitlichen Verlaufs des Membranpotentials einer Nervenzelle bei *einem* ausgelösten Aktionspotential durch eintreffende Reize.

- **Biologische Plausibilität:** Die Aktivitätslevel der binären, sigmoiden und normalverteilten Aktivitätsfunktion sind sowohl nach unten als auch nach oben hin begrenzt. Dies ist biologisch plausibel, da auch im menschlichen Gehirn bei der Signalübertragung nur ein begrenztes Aktionspotential von ca. –40 bis –80 mV auf maximal etwas unter +55 mV entsteht (z. B. Kandel, Schwartz & Jessell, 1995). Die normalverteilte Funktion ist dabei biologisch weniger naheliegend, da ein größer werdendes Membranpotential ab einem bestimmten Punkt nicht mit einem geringer werdenden Aktionspotential einhergeht. Aufgrund des *Alles-oder-Nichts-Prinzips* bei der Signalweiterleitung in menschlichen Nervenzellen könnte die binäre im Vergleich zu den sigmoiden Aktivitätsfunktionen als biologisch plausibler betrachtet werden. Abb. 5 visualisiert das Membranpotential bei einem ausgelösten Aktionspotential über die Zeit hinweg. Das Aktionspotential erfolgt immer in gleicher Stärke von maximal etwas unter +55 mV – unabhängig davon, wie stark der Schwellenwert, der ca. 10 mV über dem Ruhepotential liegt, zuvor überschritten wurde (z. B. Kandel et al., 1995). Dies kommt der binären Aktivitätsfunktion am nächsten, da auch hier der Aktivitätslevel nach Überschreiten einer Schwelle (d. h. eines spezifischen Netzinputs) immer gleich hoch ist (z. B. wird ein Aktivitätslevel von +1 ausgegeben). Andererseits wird eine ansteigende Reizstärke in künstlichen neuronalen Netzen zumeist *nicht* durch höherfrequente Aktionspotentialfolgen codiert, wie dies bei biologischen Neuronen der Fall ist. Insofern wären die sigmoiden Aktivitätsfunktionen biologisch plausibler, sofern der dort kontinuierliche Wert des Aktivitätslevels die Aktionspotentialfrequenz repräsentieren soll.

- **Differenzierbarkeit:** Im Gegensatz zu der binären Schwellenfunktion sind die lineare, sigmoide und normalverteilte Aktivitätsfunktion an allen Stellen differenzierbar, d. h., die Ableitung kann an allen Stellen gebildet werden. Dies ist beispielsweise eine notwendige Voraussetzung für das Gradientenabstiegsverfahren (Kapitel 2).

Exkurs: Differenzierbarkeit vs. Stetigkeit von Funktionen

Differenzierbarkeit und Stetigkeit sind nicht miteinander zu verwechseln. Unter Stetigkeit versteht man, dass in der Funktion marginale Änderungen der Argumente nur zu marginalen Änderungen des Funktionswertes führen. Dies bedeutet, dass die Funktion keine „Sprünge“ aufweist. Differenzierbarkeit besagt zudem, dass eine Funktion sich lokal um einen Punkt in eindeutiger Weise annähern lässt, d. h. die Ableitung gebildet werden kann. Eine differenzierbare Funktion enthält weder „Ecken“ noch „Sprünge“.

Differenzierbarkeit impliziert Stetigkeit, aber Stetigkeit schließt Differenzierbarkeit nicht zwangsläufig mit ein. Als Beispiel für eine stetige, aber nicht an allen Stellen differenzierbare Funktion wird häufig die Betragsfunktion $f(x) = |x|$ aufgeführt (siehe die zweite Funktion in Abb. 6). An der Stelle $x = 0$ liegt zwar kein „Sprung“ der Funktion vor, wohl aber eine „Ecke“, sodass dort die Steigung der Funktion nicht ermittelt werden kann. Unmittelbar vor $x = 0$ beträgt die Steigung noch –1, während sie für $x > 0$ einen Wert von +1 annimmt. Abb. 6 verdeutlicht den Unterschied zwischen Differenzierbarkeit und Stetigkeit.

Bestimmte mathematische Funktionen sind sogar an sämtlichen Stellen stetig, aber an keiner Stelle differenzierbar (eine Art „Sägezahnfunktion“). Die ersten Beispiele hierzu stammen vom Mathematiker Karl Weierstraß (1815–1897).

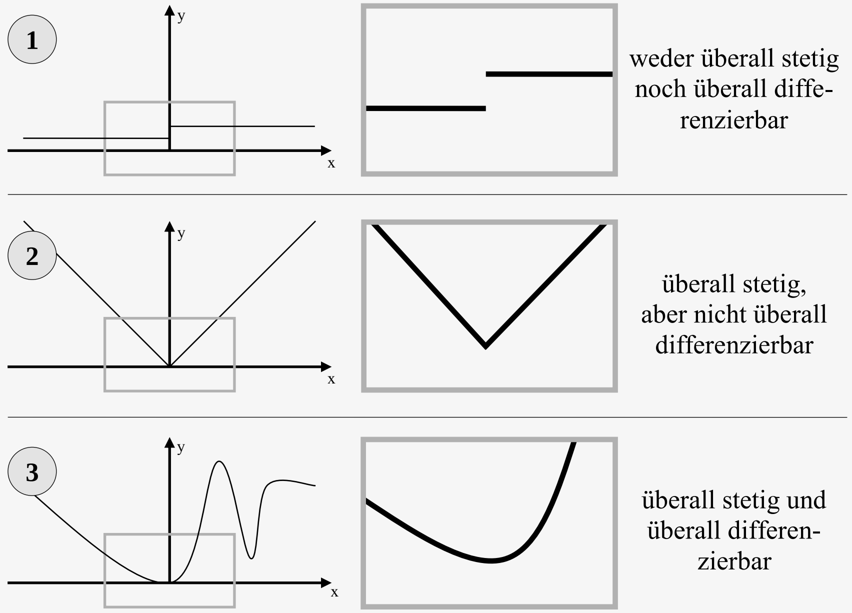


Abb. 6: Schematischer Vergleich zwischen einer nicht an allen Stellen stetigen und nicht überall differenzierbaren Funktion (1), einer stetigen, aber nicht differenzierbaren Funktion (2) sowie einer an allen Stellen stetigen und differenzierbaren Funktion (3). Erkennbar ist, dass die Funktion 1 einen „Sprung" aufweist, während Funktion 2 eine „Ecke" enthält. Lediglich Funktion 3 weist weder „Sprünge" noch „Ecken" auf.

1.4.3 Output

Output

Aus dem Aktivitätslevel einer Einheit kann der Output bzw. die Ausgabe der Unit mithilfe einer Ausgabefunktion (auch Ausgangsfunktion genannt) bestimmt werden. Gelegentlich kommt hierbei eine Schwellenwertfunktion zum Einsatz (Pospeschill, 2004). Die einfachste Zuordnung stellt jedoch die Identitätsfunktion dar, bei der der Aktivitätslevel dem Output der Unit entspricht. Als Formel ausgedrückt:

Formel

$$o_i = a_i$$

Dabei gilt:

- o_i = Output der Unit i
- a_i = Aktivitätslevel der Unit i

Aktivitätslevel und Output

Zahlreiche Bücher und Computerprogramme zu neuronalen Netzen verwenden diese Identitätsfunktion bzw. nehmen überhaupt keine Unterscheidung zwischen dem Output einer Unit und seinem Aktivitätslevel vor. Stehen hingegen in einem Simulator für neuronale Netze verschiedene Ausgabefunktionen zur Verfügung (wie beispielsweise im Computerprogramm „Neuro Visual"), so wird häufig auf eine Auswahlmöglichkeit unterschiedlicher Aktivitätsfunktionen verzichtet. Stattdessen wird dort die Identitätsfunktion angenommen, was zu denselben

Resultaten führt. Im vorliegenden Lehrbuch wird nachfolgend der Aktivitätslevel mit der Ausgabe einer Unit gleichgesetzt und mit a bezeichnet.

Keine Aufteilung des Outputs

Die Ausgabe einer Unit wird *nicht* an Einheiten der nachfolgenden Schicht aufgeteilt, die mit dieser Einheit verbunden sind. Stattdessen wird die Ausgabe jeweils in voller Stärke an sämtliche verknüpfte, nachfolgende Neuronen weitergegeben. Beträgt die Ausgabe einer Einheit beispielsweise +2, so erhalten alle nachfolgenden Units, die mit dieser Einheit verbunden sind, einen Input von ihr. Dieser setzt sich aus der Ausgabe +2 und dem dazugehörigen Gewicht zusammen (Kapitel 1.4.1).

1.5 Bias-Units

Die Bias-Unit – auch On-Neuron genannt – erhält selbst keinen Input, ihre Ausgabe bzw. ihr Aktivitätslevel beträgt immer +1. Das Gewicht von der Bias-Unit zu einer anderen Unit kann hingegen positiv oder negativ sein. Eine Bias-Unit kann auch als Input-Unit betrachtet werden, die fortlaufend den Wert +1 weiterleitet.

Positives Gewicht der Bias-Unit

Bei positivem Gewicht stellt die Bias-Unit sicher, dass die empfangende Einheit auch dann oft aktiv bleibt, wenn *kein* starker, positiver Input von anderen Einheiten vorliegt. Dies liegt daran, dass der Netzinput der empfangenden Einheit durch eine derartige Bias-Unit erhöht wird. Der positive Bias der Unit kann folglich als Voraktivierung betrachtet werden.

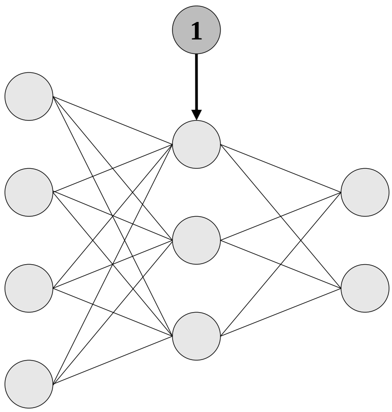


Abb. 7: Darstellung einer Bias-Unit.

Negatives Gewicht der Bias-Unit

Bei negativem Gewicht sorgt die Bias-Einheit hingegen dafür, dass Units in ihrem negativen bzw. inaktiven Zustand verharren. Dies kann in Kombination mit einer Aktivitätsfunktion, die eine Schwelle oder einen „schwellenwertähnlichen" Kurvenverlauf (z. B. die sigmoiden Aktivitätsfunktionen) beinhaltet, nützlich sein, um eine variable Schwelle zu simulieren, die andere sendende Units erst überschreiten müssen. Die so konzipierte Schwelle ist aufgrund des variablen Gewichts zwischen Bias-Unit und der mit ihr verbundenen Einheit während des Lernvorganges veränderlich. Im Gegensatz dazu besitzen Aktivitätsfunktionen mit Schwellen, aber ohne Bias-Units, eine vorab fixierte Schwelle.

Bias-Units in verschiedenen Schichten

Bias-Units können – wie in Abb. 7 dargestellt – ihr Aktivitätslevel auf bestimmte Units der Hidden-Schicht senden. Ebenso sind Bias-Neurone in der Output-Schicht möglich. Auf der Ebene der Input-Schicht sind Bias-Units nur dann sinnvoll, wenn die Informationen aus der Umwelt innerhalb der Input-Units weiterverarbeitet werden und *nicht* direkt an die nachfolgende Schicht gesendet werden (Kapitel 1.4).

1.6 Trainings- und Testphase

Bei neuronalen Netzen unterscheidet man typischerweise zwischen Trainingsphase und Testphase:

Trainingsphase

- **Trainingsphase:** In dieser Phase soll das Netz das vorgegebene Lernmaterial einüben. Das Lernmaterial wird dem neuronalen Netz als „Zahlenbündel", d. h. als Vektoren bzw. Matrizen zur Verfügung gestellt. Diese repräsentieren zumeist einen bestimmten Ausschnitt der „Realität", beispielsweise verschiedene Bilder, Texte, Sprachen, Musik oder bei der Datenauswertung erhobene Messwerte wie Intelligenz oder Motivation verschiedener Personen. In der Trainingsphase werden mithilfe dieser Zahlenwerte, die dem Netz in aller Regel wiederholt dargeboten werden, die Gewichte zwischen den einzelnen Neuronen modifiziert. Lernregeln geben dabei die genaue Art und Weise an, wie das neuronale Netz diese Veränderungen vornimmt (Kapitel 2). Grundsätzlich kann man folgende Klassifikation vornehmen, *wie* die Gewichte verändert werden:

  Art der Gewichtsmodifikation

  - **Supervised learning** (überwachtes bzw. beaufsichtigtes Lernen): Der korrekte Output – auch als Zielmuster oder teaching vector bezeichnet – wird dem Netz vorgegeben und an diesem werden die Gewichte optimiert.
  - **Reinforcement learning** (bestärkendes bzw. verstärkendes Lernen): Im Gegensatz zum supervised learning wird dem Netz *nicht* der exakte Output mitgeteilt. Stattdessen erhält das neuronale Netz lediglich eine Rückmeldung, ob die produzierte Ausgabe richtig oder falsch war.
  - **Unsupervised learning** (nicht überwachtes bzw. unbeaufsichtigtes Lernen, auch als self-organized learning bezeichnet): Es wird *kein* Output vorgegeben. Die Gewichtsveränderungen erfolgen in Abhängigkeit der Ähnlichkeit der Inputreize.
  - **Direct design methods** (direkte Designmethoden, auch hardwired systems genannt): Hier werden die Gewichte *nicht* verändert, sondern die Verschaltung wird vorab festgelegt. Lernen im Sinne einer Gewichtsmodifikation liegt hier folglich *nicht* vor.

  Des Weiteren kann danach unterschieden werden, *wann* die Gewichte verändert werden:

  Zeitpunkt der Gewichtsmodifikation

  - **Incremental training** (inkrementelles Training bzw. auch als musterbasierte Modifikation, online learning oder adaptives Training bezeichnet): Gewichtsveränderungen erfolgen nach Darbietung jedes einzelnen Inputreizes.
  - **Batch training** (stapelweises Training, auch blockbasierte bzw. kumulierte Modifikation, epochales Lernen oder offline learning genannt): Die Gewichte werden nach Präsentation sämtlicher Inputreize modifiziert. Die Darbietung aller Inputreize wird auch als Epoche bezeichnet.

- **Testphase:** In der Test- oder Ausbreitungsphase werden hingegen *keine* Gewichte verändert. Stattdessen wird hier auf Grundlage der bereits modifizierten Gewichte aus der Trainingsphase untersucht, ob und was das Netz gelernt hat. Dazu präsentiert man den Inputneuronen Reize und prüft, welchen Output das neuronale Netz berechnet. Zwei verschiedene Arten von Reizen bzw. Informationen können unterschieden werden:

Testphase

  - **Ausgangsreize:** Durch erneute Präsentation der in der Trainingsphase zu lernenden Reize bzw. Informationen wird geprüft, ob das neuronale Netz das Material selbst erfasst hat.

Reizarten in der Testphase

  - **Neue Reize:** Durch Präsentation neuer Reize bzw. Informationen kann man feststellen, ob das Netz über die zu lernenden Reize hinaus in der Lage ist, Aufgaben zu lösen. Damit soll die Frage beantwortet werden, ob das neuronale Netz auf diese neuen Reize generalisiert. Hierzu überprüft man beispielsweise, wie gut die Vorhersage des Netzes bei Darbietung der neuen Reize ausfällt. In diesem Fall muss die korrekte Ausgabe – ähnlich wie beim supervised learning – bekannt sein. Diese Werte beeinflussen aber *nicht* mehr die bereits abgeschlossene Gewichtsmodifikation.

- **Anwendungsphase:** Die Trainings- und Testphase wird gelegentlich auch noch von einer Anwendungsphase abgegrenzt. In dieser Phase gelangt ein bereits trainiertes Netz zur Lösung eines konkreten Anwendungsproblems zum Einsatz. Beispielsweise können neuronale Netze als Gesichtserkennungssoftware auf einem Flughafen eingesetzt werden. In einer solchen Phase werden die Gewichte entweder wie in der Trainingsphase weiter modifiziert oder aber der Lernvorgang wird wie in der Testphase als abgeschlossen betrachtet.

Anwendungsphase

1.7 Matrizendarstellung

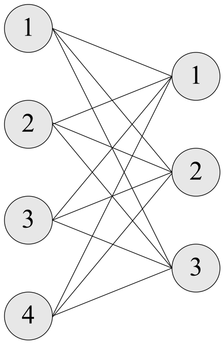
Neuronale Netze lassen sich auch als Matrizen darstellen. Dies hat den Vorteil, dass die im Kapitel 1.4 erläuterten Berechnungen mathematisch relativ einfach und zusammenfassend vorgenommen werden können.

Matrix

Eine Matrix ist eine mathematische Einheit, genau wie eine einzelne Zahl. Solch eine Matrix **W** besteht aus einer Menge von Elementen w_{ij}. Der erste Index i gibt dabei die Zeile der Matrix an, der zweite Index j hingegen die Spalte, in der das Element steht. Da das Lernen in neuronalen Netzen typischerweise in den Gewichten stattfindet und diese das gelernte Wissen des Netzes speichern, werden die Netzgewichte als Zahlenwerte in der Matrix repräsentiert. Die Input- und Output-Werte, die das Netz erhält bzw. berechnet, können hingegen zumeist als Vektoren abgebildet werden. Dabei stellt ein Vektor eine spezielle Matrix dar, die jeweils nur aus einer Zeile oder einer Spalte besteht (nähere Angaben zum Rechnen mit Matrizen finden sich beispielsweise bei Moosbrugger, 2002).

Anzahl benötigter Gewichtsmatrizen

Ein neuronales Netz kann man durch *eine* Gewichtsmatrix darstellen (Abb. 8), sofern *keine* Hidden-Schicht existiert. Bei einer Hidden-Schicht würde man zwei Gewichtsmatrizen benötigen, bei zwei Hidden-Schichten drei Matrizen usw.



$$= \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \end{bmatrix} = W$$

Abb. 8: Darstellung der Äquivalenz zwischen schematischer Illustration und Matrizenschreibweise bei neuronalen Netzen. Der jeweils erste Index der Gewichte bezieht sich auf die Output-Unit (1, 2 oder 3), der zweite Index gibt die entsprechende Input-Unit (1 bis 4) des neuronalen Netzes an.

Repräsentation der Gewichte in einer einzelnen Matrix

Ein neuronales Netz kann nicht nur mithilfe einer oder mehrerer Gewichtsmatrizen dargestellt werden, wie beispielsweise im Computerprogramm MATLAB, sondern sämtliche Gewichte des Netzes können auch durch nur *eine* einzige Matrix symbolisiert werden. Abb. 9 stellt eine solche Gewichtsmatrix dar. Weiß unterlegte Felder repräsentieren dabei nicht existierende Verbindungen zwischen zwei Neuronen. Im Rahmen der Matrizenberechnungen kann hier eine *nicht modifizierbare* Null eingesetzt werden, während eine modifizierbare Null eine Verbindung mit einem Gewicht von Null darstellt.

Informationsausbreitung in Leserichtung

Die Informationsausbreitung im Netz erfolgt wie gewohnt in Leserichtung von links nach rechts. Bei rekurrenten Verbindungen, die sich durch Rückkopplungen von Neuronen einer Schicht zu anderen Neuronen derselben oder einer vorangegangenen Schicht auszeichnen (Kapitel 3.4), wären zusätzliche Felder der Matrix in Abb. 9 grau zu hinterlegen.

Ähnlichkeiten verschiedener Matrizendarstellungen

Vergleicht man Abb. 8 mit Abb. 9, so ist erkennbar, dass die Matrizendarstellungen deutliche Ähnlichkeiten aufweisen. Die in der Abb. 8 visualisierte Matrix vom Typ 3 x 4 ist auch in der zweiten Darstellung vorhanden und zwar als grau markierte Fläche mit ebenfalls zwölf Zahlenwerten, welche die Gewichte zwischen den Input- und Hidden-Units repräsentieren. Die zweite grau markierte Fläche (Abb. 9 rechts unten) besteht aus sechs Feldern und stellt die Gewichte der insgesamt sechs Verbindungen dar, die sich zwischen Hidden- und Output-Schicht befinden. Diesen zweiten „Block" könnte man auch als weitere eigenständige Matrix ähnlich wie in Abb. 8 darstellen.