

Perzeptronen

Katrin Dust Felix Oppermann

Universität Oldenburg, FK II - Department für Informatik

Vortrag im Rahmen des Proseminars 2004

2. Mai 2004

Zusammenfassung

Perzeptronen stellen eine einfache Form künstlicher neuronaler Netze dar. Sie wurden in den 50er Jahren von F. Rosenblatt eingeführt und basieren auf dem damaligen Wissen über biologische neuronale Netze. Da ein einfaches Perzeptron nur wenige Probleme zu lösen vermochte, wurde es zum Multilayer-Perzeptron weiterentwickelt. Ein Lernalgorithmus für diese komplexeren Modelle - der Backpropagation Algorithmus - wurde erst in den 80er Jahren entwickelt. Perzeptrone wurden bei der Erforschung der theoretischen Grundlagen neuronaler Netze verwendet und finden Anwendung in der Muster- und Spracherkennung.

1 Einleitung

Ursprünglich schuf Rosenblatt ¹ das Modell des Perzeptrons in den 50er Jahren, um die Vorgänge in biologischen neuronalen Netzen besser zu verstehen. Für ihn stellte das Gehirn nur einen Spezialfall von Verknüpfungen einfacherer Elemente dar und er hoffte durch das genaue Verständnis der Grundbausteine später auch komplexe Strukturen erklären zu können [Ros62]. Im Folgenden war dieses Modell auch bei der theoretischen Betrachtung paralleler Datenverarbeitung in technischen Systemen von Interesse. Hier dient es auch als Baustein für mächtigere Modelle. [Sim90]

Nach einem kurzen Überblick über die biologischen und mathematischen Vorbilder des Perzeptrons wird zunächst auf zwei bedeutende Definitionen eingegangen. Im Anschluss werden die Möglichkeiten und vor allem die Grenzen des einfachen Perzeptrons erläutert. Aufbauend darauf wird mit dem Multilayer-Perzeptron eine mächtigere Variante betrachtet, welche einige dieser Einschränkungen überwindet. Den Abschluss der Arbeit bilden zum Trainieren des Netzes gebräuchliche Lernalgorithmen für beide Varianten.

2 Vorbilder und Vorgänger der Perzeptrone

2.1 Vorbild Neuron

Bei höheren Lebewesen wird die Weiterleitung und Verarbeitung von Informationen aus der Umwelt von einem bestimmten Zelltyp übernommen. Diese so genannten Nervenzellen, auch Neurone genannt, bestehen aus dem Zellkörper, den Dendriten und dem Axon. Die Dendriten sind kurze und stark verästelte Fortsätze und

¹Rosenblatt, Frank (*1928 in New York): amerikanischer Psychologe. [Wik]

ermöglichen der Nervenzelle die Übernahme von Signalen anderer Zellen. Beim Axon handelt es sich um einen bis zu einem Meter langen Fortsatz, der eine Übertragung von Signalen an andere Zellen ermöglicht. (Neben diesem Grundschema gibt es auch abweichende Varianten.)

Die Dendriten übernehmen Signale von anderen Nervenzellen an besonderen Kontaktstellen, den Synapsen, und leiten diese als bioelektrisches Signal zum Zellkörper. Die einlaufenden Signale werden dort, wo sie aufeinander treffen, annähernd aufsummiert. Am Ursprung des Axons, dem Axonhügel, wird die Erregung dann mit einem Schwellenwert verglichen und entschieden, ob sie als Aktionspotential weitergeleitet wird. Das Aktionspotential folgt dem Alles-Oder-Nichts-Gesetz [SS00]. Wird der Schwellenwert nicht erreicht, so wird kein Signal weitergeleitet, ansonsten wird es in voller Stärke über das Axon und die sich am Ende befindenden Synapsen an andere Neurone weitergegeben. Hierbei kann das Signal bei der nachgeschalteten Zelle eine Signalerzeugung hervorrufen oder unterdrücken. [Gol02]

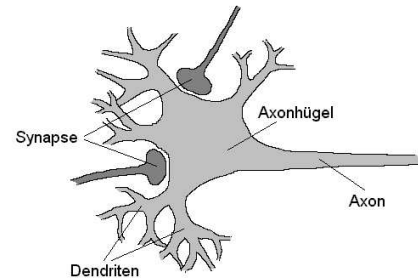


Abbildung 1: Neuron

2.2 Vorgänger McCulloch-Pitts-Zelle

Motiviert von diesem Wissen aus der Biologie entwickelten die Amerikaner W. S. McCulloch² und W. Pitts³ 1943 ein einfaches abstraktes Neuronenmodell: die McCulloch-Pitts-Netze. Sie bestehen aus Zellen, auch Neurone genannt, und Verbindungen zwischen diesen. Die einzelnen Zellen besitzen mehrere Eingänge und einen Ausgang. Die Eingangssignale können entweder aktivieren oder hemmen. Das Ausgabesignal der Zelle wird berechnet, indem die Eingangssignale aufsummiert und mit einem Schwellenwert verglichen werden. Wird dieser erreicht, so gibt die Zelle eine Eins aus, ansonsten eine Null. Sind keine Eingabesignale vorhanden oder liegt an einem der hemmenden Eingänge eine Eins an, so ist die Ausgabe ebenfalls Null. Das Modell verwendet ausschließlich binäre Signale. [Roj96]

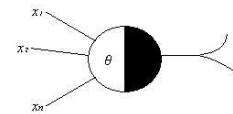


Abbildung 2: McCulloch-Pitts-Zelle

Mit McCulloch-Pitts-Zellen können Netze aufgebaut werden, die beliebige logische Funktionen und endliche Automaten simulieren. Lernen ist aber mit dieser Art von Netzelementen nur durch eine Veränderung der Schwellenwerte und der Netztopologie möglich und benötigt sehr komplexe Algorithmen [Ros62].

3 Das Perzeptron

Der Begriff Perzeptron umfasst verschiedene Verständnisse ähnlicher Modelle neuronaler Netze. In der Literatur wird der Begriff meist nur für jene Formen verwendet,

²McCulloch, Warren Sturgis (*1898 in Orange, NJ; † 1969): Amerikanischer Neurophysiologe und Cybernetiker. Studierte an der Universität in Yale Psychologie und Philosophie und an der Universität von Columbia Psychologie. [Wik]

³Pitts, Walter (*1924 in Detroit, MI; † 1969): Amerikanischer Logiker und Autor im Bereich der kognitiven Psychologie. Pitts hatte weder einen Schulabschluss, noch einen akademischen Grad, er begann jedoch bereits im Alter von zehn Jahren mit dem Selbststudium mathematischer und psychologischer Sachverhalte. [Wik]

welche M. Minsky⁴ und S. Papert⁵ im ersten Kapitel ihrer Abhandlung „Perceptrons“ [MP88] definieren.

Um die Zusammenhänge der verschiedenen Modelle deutlich zu machen, folgt eine chronologisch geordnete Erklärung der unterschiedlichen Theorien.

3.1 Rosenblatt und Minsky-Papert

Da das Lernen in McCulloch-Pitts-Netzen sehr aufwändig ist und F. Rosenblatt das Modell für biologische Systeme für weniger plausibel hielt, schlug er 1958 ein anderes neuronales Modell vor - das Perzeptron [Ros89]. Ähnlich wie die McCulloch-Pitts-Netze ist das Perzeptron aus drei Neuronenschichten und Verbindungen zwischen diesen aufgebaut. Punkte auf der Retina, der Netzhaut, liefern binäre Eingabewerte an die erste Neuronenschicht, welche Abbildungsregion genannt wird. Die Verbindungen von den Punkten zu den Abbildungsneuronen sind deterministisch und fest. Die Verbindungen von der Abbildungsregion zur zweiten Schicht, der Verknüpfungsregion sowie von der zweiten zur dritten Schicht, der Reaktionsschicht, sind stochastisch. Die Idee von Rosenblatt war, mit der Verknüpfungsregion bestimmte Muster aus den Retinapunkten zu erkennen und die entsprechende Reaktion zu steuern. Ein Lernalgorithmus muss die notwendigen Verknüpfungen finden, um Muster auf der Retina mit Reaktionen zu assoziieren. [Roj96]

Das Modell von Rosenblatt wurde von Minsky und Papert ausführlich untersucht und dabei auf das minimal Notwendige reduziert, um die Eigenschaften und Grenzen dieser Netze ausfindig zu machen. Hauptmerkmal dieser Perzeptrone sind Gewichte an den Verbindungen, mit deren Hilfe Lernen stattfindet. Es gibt bei diesem Modell wiederum Punkte auf der Retina, die direkt mit logischen Prädikaten verbunden sind. Die Werte der Prädikate werden über gewichtete Leitungen an eine Zelle weitergegeben, die dann die Ausgabe mit Hilfe eines Schwellenwertes berechnet. Die Abbildung 3 veranschaulicht ein Perzeptron nach Minsky und Papert. Die Prädikate x_1, x_2, x_3 und x_4 liefern Informationen über die Projektionsfläche und erzeugen binäre Ausgabewerte. Sie werden weitergegeben und mit dem zugehörigem Gewicht multipliziert. Die letzte Zelle summiert die Eingaben und vergleicht sie mit einem Schwellenwert θ . Allgemein ist das Perzeptron nach Minsky und Papert ein Netz, das für eine Reihe von n binären Prädikaten x_1, \dots, x_n dazugehörigen reelwertigen Gewichten w_1, \dots, w_n und einem reelwertigen Schwellenwert θ nur dann eine Eins ausgibt, wenn $\sum_{i=1}^n w_i \cdot x_i \geq \theta$ gilt. [Roj96]

Um die Theorie der Perzeptronen weiter zu untersuchen, überführen wir das Modell von Minsky und Papert in ein Schema eines einfachen Perzeptrons. Ein einfaches Perzeptron ist eine Art McCulloch-Pitts-Zelle, deren Eingabewerte gewichtet werden. Es besteht aus n Eingängen x_1, \dots, x_n , dazugehörige Gewichten w_1, \dots, w_n , einem

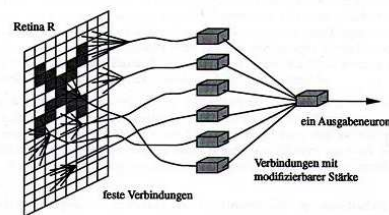


Abbildung 3: Minsky & Papert Perzeptron [Zel94]

⁴Minsky, Marvin (*1927 in New York): Mathematiker am MIT. Minsky baute 1951 eine der ersten auf neuronalen Netzen basierende Lernmaschine und ist Mitbegründer des „MIT's AI Laboratory“. 1970 wurde er mit den Turing Award ausgezeichnet. [Wik]

⁵Papert, Seymour (*1928 in Pretoria, Südafrika): Mathematiker am MIT. Erfinder der Programmiersprache LOGO. [Wik]

Schwellenwert θ und einer Ausgabe a , wobei gilt:

$$a = f\left(\sum_{i=1}^n w_i \cdot x_i - \theta\right), \text{ mit } f(x) = \begin{cases} 1, & \text{falls } x \geq 0 \\ 0, & \text{falls } x < 0 \end{cases}$$

Mit Hilfe der Perzeptrone kann man also eine Funktionskomposition $\Phi\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ berechnen, wobei $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ die Integrationsfunktion und $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ die Ausgabefunktion ist. Die Integrationsfunktion entspricht der Addition der gewichteten Eingaben und die Ausgabefunktion, oder auch Aktivierungsfunktion genannt, vergleicht diese Summe dann mit dem Schwellenwert bzw. Null. [Roj96]

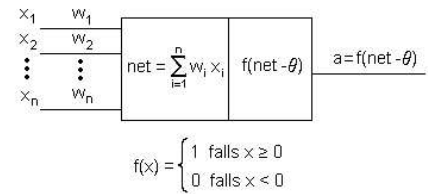


Abbildung 4: Einfaches Perzeptron

3.2 Beispiel

Zur Veranschaulichung führen wir hier ein Beispiel eines Perzeptrons für die logische AND-Funktion an. Es besitzt zwei Eingänge x_1 und x_2 und einen Ausgang a . Damit für die Ausgabe $a = x_1 \wedge x_2$ gilt, setzt man nun das Gewicht w_1 des Eingangssignals $x_1 = 1$, ebenso $w_2 = 1$ und den Schwellenwert $\theta = 1,5$. Durch die Tabelle 1 wird deutlich, dass das Perzeptron hierdurch ein logisches AND repräsentiert.

x_1	x_2	$x_1 \wedge x_2$	$w_1 x_1 + w_2 x_2 > \theta \Rightarrow a$
0	0	0	$0 + 0 < 1,5 \Rightarrow 0$
0	1	0	$0 + 1 < 1,5 \Rightarrow 0$
1	0	0	$1 + 0 < 1,5 \Rightarrow 0$
1	1	1	$1 + 1 > 1,5 \Rightarrow 1$

Tabelle 1: Tafel

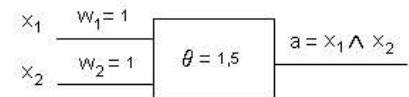


Abbildung 5: logische AND Funktion

3.3 Geometrische Interpretation

Anhand des vorangegangenen Beispiels erläutern wir nun die geometrische Interpretation von Perzeptronen. Da das Perzeptron der AND-Funktion zwei Eingänge besitzt, befinden wir uns im zweidimensionalen Raum. Die Werte der Eingänge werden auf der jeweiligen Achse eingetragen. Die jeweiligen Punkte bzw. Eingaben werden in zwei Klassen eingeteilt. In die erste Klasse gehören diejenigen Punkte, deren Ausgabe eine Eins, in die zweite Klasse diejenigen, deren Ergebnis eine Null ist. In diesem Beispiel gehören also die Punkte $(0, 0)$, $(0, 1)$ und $(1, 0)$ zu einer Klasse und der Punkt $(1, 1)$ zur anderen. Die Gerade $w_1 x_1 + w_2 x_2 = \theta$ (bzw. eingesetzt: $x_2 = -x_1 + 1,5$) trennt die beiden Klassen voneinander.

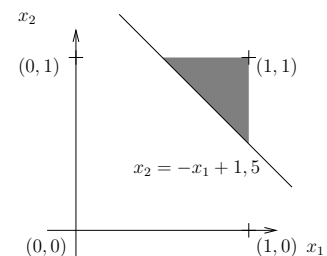


Abbildung 6: Geometrische Interpretation

Allgemein läßt sich sagen, dass ein Perzeptron mit n Eingängen einen n dimensionalen Raum in zwei Regionen mit Hilfe einer Hyperebene einteilt. In der einen Region befinden sich die Punkte, bzw. Eingaben, bei denen das Perzeptron eine Eins, und in der anderen die Punkte, in der es eine Null ausgibt. Man sagt, die Hyperebene trennt die Klassen der Punkte linear voneinander. [LC01]

3.4 Repräsentierbarkeit

Es kann nun untersucht werden, welche Funktionen mit einem einzigen Perzeptron dargestellt werden können. Unter Verwendung einer einzigen Zelle lassen sich welche logischen Funktionen berechnen? Wir untersuchen diese Frage anhand von booleschen Funktionen $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, von denen es 2^4 verschiedene gibt. Die AND-Funktion aus Abschnitt 3.2 ist z.B. eine solche Funktion.

Durch die geometrische Interpretation aus dem vorherigen Kapitel wurde ersichtlich, dass sich all diejenigen Funktionen im zweidimensionalen Raum berechnen lassen, bei denen man durch eine Gerade die zwei Mengen von Punkten linear voneinander trennen kann. Es wird deutlich, dass die beiden Funktionen, bei denen jeweils die gegenüberliegenden Punkte zu einer Klasse gehören, nicht mit Hilfe eines Perzeptrons dargestellt werden können. Es sind die XOR-Funktion und die Identität. [Roj96]

Im folgenden ist ein Beweis angegeben, der zeigt, dass die XOR-Funktion nicht berechnet werden kann.

Beweis: Damit ein Perzeptron die XOR Funktion berechnet, muss für die Eingänge x_1 und x_2 und den Ausgang a gelten:

x_1	x_2	a	$\sum_{i=1}^2 w_i \cdot x_i$	für θ folgt:
0	0	0	$w_1x_1 + w_2x_2 = 0$	$\theta \geq 0$
1	0	1	$w_1x_1 + w_2x_2 = w_1$	$\theta < w_1$
0	1	1	$w_1x_1 + w_2x_2 = w_2$	$\theta < w_2$
1	1	0	$w_1x_1 + w_2x_2 = w_1 + w_2$	$\theta \geq w_1 + w_2$

Die letzte Gleichung steht im Widerspruch zu den ersten dreien - die XOR-Funktion läßt sich also nicht durch ein Perzeptron darstellen.

Es gibt folglich Funktionen, die nicht durch ein einziges Perzeptron berechnet werden können. Die Eigenschaft einer Funktion, dass die Klassen der Punkte durch eine Hyperebene getrennt werden können, bezeichnet man als linear trennbar bzw. linear separierbar. [LC01]

Ein einzelnes Perzeptron kann also nur linear trennbare Funktionen berechnen. Interessant ist, wie viele solcher Funktionen existieren. Dies ist abhängig von der Anzahl der Variablen der Funktion. Bei zwei Variablen hat sich oben gezeigt, dass es 14 linear separierbare von 16 möglichen Funktionen gibt. Bei drei Variablen sind es 104 von 256 bei vier Variablen 1882 von 65536. Mit steigender Zahl an Eingangsvariablen nimmt der Prozentsatz linear trennbarer Funktionen exponentiell ab. [Roj96]

4 Multilayer-Perzeptron

Die Einschränkungen aus 3.4 des einschichtigen Perzeptrons wurden 1969 von M. Minsky und S. Papert nachgewiesen. Die Euphorie in der Erforschung neuronaler Netze ließ daraufhin in den folgenden Jahren nach. Da die McCulloch-Pitts-Zellen

ein Untermodell der Perzeptronen darstellen, war es zwar bekannt, dass durch das Verschalten von Perzeptronen komplexere Probleme, wie z.B. das XOR-Problem, gelöst werden können, aber man kannte zu dieser Zeit noch keinen effizienten Lernalgorithmus für ein mehrschichtiges Perzeptron (Multilayer-Perzeptron) [Roj96]. Als allerdings 1986 der Backpropagation-Algorithmus als Lernverfahren für mehrschichtige Netze entdeckt wurde, wurde das Multilayer-Perzeptron wieder interessant.

Ein Multilayer-Perzeptron besteht aus einzelnen einfachen Perzeptronen, die in Schichten angeordnet sind. In strengen Netzen sind die Neuronen der einen Schicht mit denen der nachfolgenden Schicht vollständig verknüpft, und es dürfen keine Schichten übersprungen werden. Die erste Schicht wird als Eingabe- und die letzte als Ausgabeschicht bezeichnet. Die mittleren Schichten werden versteckte Schichten (Hidden Layers) genannt. Da nur Verbindungen von der Eingabeschicht in Richtung der Ausgabeschicht bestehen, gehören diese Netze, wie die Perzeptrone, zu den vorwärtsgerichteten Netzen (Feedforward Net). Grundsätzlich reichen drei Schichten aus, um alles berechnen zu können, was auch eine Turingmaschine berechnen kann, doch häufig werden mehr Schichten verwendet, da diese Netze oftmals leichter zu trainieren sind. [Roj96]

Zur Veranschaulichung dient die Abbildung 7, ein 2-schichtiges Multilayer Perzeptron zur Lösung der XOR-Funktion.

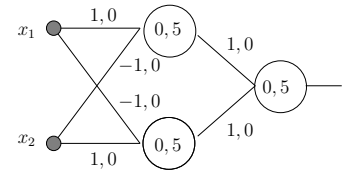


Abbildung 7: Lösung des XOR-Problems

5 Lernalgorithmen

5.1 Der Perzeptron-Lernalgorithmus

Lernen in neuronalen Netzen bedeutet meistens, die Gewichte zu trainieren. Man initialisiert das Perzeptron mit Zufallszahlen und wendet dann einen Lernalgorithmus an, der die Gewichte des Netzes verändert. Dem Netz werden hierzu wiederholt Muster präsentiert. Es werden also nicht explizit Regeln gelernt, sondern die Regeln werden implizit gelernt. [LC01]

Minsky und Papert schlugen folgenden Algorithmus vor:

Es existieren zwei endliche Punktmengen (die Klassen der Punkte) P und N in einem n -dimensionalen Raum. Es wird ein Gewichtevektor $w = w_1, \dots, w_n$ gesucht, der die beiden Mengen linear trennt.

Algorithmus:

Start: Der Gewichtsvektor w_0 wird zufällig generiert.
Setze $t := 0$.

Testen: Ein Punkt $x \in P \cup N$ wird zufällig gewählt.

Falls $x \in P$ und $w_t * x > 0$ gehe zu Testen

Falls $x \in P$ und $w_t * x \leq 0$ gehe zu Addieren

Falls $x \in N$ und $w_t * x < 0$ gehe zu Testen

Falls $x \in N$ und $w_t * x \geq 0$ gehe zu Subtrahieren

Addieren: Setze $w_{t+1} = w_t + x$.

Setze $t := t + 1$. Gehe zu Testen

Subtrahieren: Setze $w_{t+1} = w_t - x$.

Setze $t := t + 1$. Gehe zu Testen

Ein Gewicht wird immer dann korrigiert, wenn ein Punkt einer Klasse bzw. Menge nicht richtig zugeteilt wurde. Der Algorithmus kann also dann terminieren, wenn alle Punkte der richtigen Menge zugeordnet sind. Hierfür müßte man einen Test implementieren, der alle Punkte entsprechend testet.

Der Konvergenzsatz für den Perzeptronen-Lernalgorithmus garantiert, dass der Algorithmus mit einer endlichen Anzahl von Korrekturen der Gewichte erfolgreich terminiert, falls die Mengen P und N linear trennbar sind.

Konvergenzsatz: Nach endlich vielen Schritten t wird ein Gewichtsvektor w_t gefunden, der die Mengen P und N linear trennt, falls die Mengen P und N endlich und linear trennbar sind. [Roj96]

Der Perzeptronen-Algorithmus konvergiert also in endlich vielen Schritten, aber es kann unter Umständen sehr lange dauern. Die Delta-Regel beschleunigt den Lernalgorithmus. Es ist ein Verfahren von Widrow & Hoff von 1960, ursprünglich für lineare Berechnungselemente, das von Aleksander auf Perzeptronen übertragen wurde. Bei jedem Lernschritt wird der Fehler δ berechnet und $k(\delta + \epsilon)x$ zum Gewichtsvektor addiert. Die Konstante k ist dabei ein Lernfaktor, der den Grad der Änderung der Gewichte und somit die Konvergenzgeschwindigkeit des Algorithmus beeinflusst.[LC01]

5.2 Backpropagation Algorithmus

In den siebziger Jahren wurde der Backpropagation Algorithmus von unterschiedlichen Forschern zu unterschiedlichen Zwecken entwickelt, geriet dann jedoch in Vergessenheit, bis er als Lernalgorithmus für neuronale Netze mit mehreren Schichten von Rumelhart, Hinton und Williams 1986 wiederentdeckt wurde. Der Backpropagation Algorithmus ist somit auch ein, bzw. das Lernverfahren für Multilayer-Perzeptrene.

Das Verfahren erfolgt in zwei Schritten: Zuerst wird das zu lernende Muster präsentiert und Schicht für Schicht werden die Zwischenergebnisse berechnet bis zur letzten Schicht und damit zur Ausgabe. In der zweiten Phase wird der Fehler bestimmt und damit dann eine entsprechende Gewichtsänderung. Hierbei wird wiederum schrittweise, nun aber angefangen bei der letzten Schicht, vorgegangen. Die produzierte Ausgabe wird mit der gewünschten Ausgabe verglichen und ein Fehler-signal wird gebildet. Abhängig von diesem werden die Gewichte zwischen der letzten Schicht und der vorherigen geändert sowie ein neues Fehlersignal für die nächste Schicht berechnet. Ist der Fehler bis zur ersten Schicht vorgedrungen, kann wieder ein Muster angelegt werden. Wiederholt man diese Lernprozedur, so wird der Fehler Schritt für Schritt verringert. Der Lernvorgang kann nach einer bestimmten Anzahl von Lernvorgängen oder einer gewissen Fehlertoleranz beendet werden. [Köh90]

Durch eine Lernrate wird bestimmt, wie groß die Änderung der Gewichte ist. Wählt man die Lernrate zu groß, so kann ein Lernschritt zuvor Erlerntes zerstören. Ist die Lernrate zu klein, sind sehr viele Lernschritte nötig, um das Netz zu trainieren. Üblich sind Werte im Bereich zwischen 0.1 und 1.0.[Köh90]

Backpropagation ist ein Beispiel für ein Gradientenabstiegsverfahren, das zur Minimierung von Netzfehlern eingesetzt wird[LC01].

6 Zusammenfassung

Perzeptrone erlauben als relativ einfaches Modell eine vollständige mathematische Analyse und ermöglichen die Erkundung der Möglichkeiten paralleler Datenverarbeitung in neuronalen Netzen. Den Anspruch, Vorgänge in biologischen Systemen erklären zu können, konnten sie jedoch nicht erfüllen, da sie sich hierfür als zu einfach erwiesen. Sie dienten jedoch oft als Baustein oder Inspiration für höhere Modelle wie zum Beispiel Minskys „Society of Mind“ [MP88], einer Theorie über das menschliche Gehirn, welche das Perzeptron mit anderen Ansätzen aus der KI-Forschung verbindet. Neben seinen Beiträgen zur theoretischen Untersuchung neuronaler Netze, dient das Modell des Perzeptrons auch als Grundlage technischer Anwendungen, wenn auch bisher meist nur im wissenschaftlichen Umfeld. Perzeptrone eignen sich in der praktischen Anwendung vorrangig zur Klassifizierung von Eingabedaten und werden vor allem im Bereich der Sprach- und Bilderkennung verwendet [DAR90, Sim90].

Literatur

- [DAR90] DARPA, Herausgeber. *DARPA Neural Network Study*. AFCEA International Press, Fairfax, Virginia, 1990.
- [Gol02] E. Bruce Goldstein. *Wahrnehmungspsychologie*, S. 25–33. Spektrum, Heidelberg; Berlin, Zweite Ausgabe, 2002.
- [Köh90] Monika Köhle. *Neurale Netze*. Springer, Wien, 1990.
- [LC01] Uwe Lämmle und Jürgen Cleve. *Lehr- und Übungsbuch: Künstliche Intelligenz*. Fachbuchverlag Leipzig, Leipzig, 2001.
- [MP88] Marvin Minsky und Seymour Papert. *Perceptrons: An introduction to computational geometry*. The MIT Press, Dritte Ausgabe, 1988.
- [Roj96] Raul Rojas. *Theorie der neuronalen Netze*. Springer, Berlin, Vierte Ausgabe, 1996.
- [Ros62] Frank Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, DC, 1962.
- [Ros89] Frank Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain. In: James A. Anderson und Edward Rosenfeld, Herausgeber, *Neurocomputing: Foundations of Research*. The MIT Press, 1989.
- [Sim90] Patrick K. Simpson. *Artificial Neural Systems: foundations, paradigms, applications and implementations*, Kapitel 5.4.1, S. 100–105. Pergamon, 1990.
- [SS00] Robert F. Schmidt und Hans-Georg Schaible. *Neuro- und Sinnesphysiologie*, Kapitel 1.2, S. 4–7. Springer, Vierte Ausgabe, 2000.
- [Wik] Wikipedia: The Free Encyclopedia. <http://en.wikipedia.org/>.
- [Zel94] Andreas Zell. *Simulation Neuronaler Netze*, Kapitel 7, S. 97–104. Addison-Wesley, Bonn, 1994.