

Suche

Erweiterte Suche...

Navigation

- Informations-, Daten- und Wissensmanagement
- Entwicklung und Management von Informationssystemen
- Inner- und überbetriebliche Informationssysteme
- Technologische und methodische Grundlagen
 - Hardware
 - Informatik, Grundlagen
 - KI und Softcomputing
 - Künstliche Intelligenz
 - Agententechnologie
 - Multiagentensystem
 - Expertensystem
 - Reinforcement Learning
 - Neuronales Netz
 - Case Based Reasoning
 - Soft Computing
 - Fuzzy-Systeme
- Methoden zur Unterstützung des strategischen Managements
 - Operations Research
 - Rechnernetze und Kommunikationssysteme
 - Software
 - Sprache
 - Statistik
- Übergreifender Teil

Neuronales Netz

Michael H. Breitner

Zur (künstlichen) Intelligenz zählen Abstraktionsfähigkeit, Kombinationsfähigkeit, intellektuelle Beweglichkeit, schlussfolgerndes Denken, Auffassungsgenauigkeit, Gedächtnis, Sprachbeherrschung, räumliches Vorstellungsvermögen, Rechnen sowie Phantasie und Kreativität. Eine Möglichkeit zur Realisierung der Künstlichen Intelligenz wird darin gesehen, ein biologisches neuronales Netz durch ein Künstliches Neuronales Netz (KNN) in einem Rechner zu simulieren.

Die menschliche Intelligenz beruht auf einem biologischen neuronalen Netz mit ca. 100.000.000.000 Neuronen im Gehirn, die teils mit nur wenigen, teils aber auch mit vielen Tausenden von anderen Neuronen hochgradig parallel verbunden sind. In ca. einer Millisekunde kann ein aktiviertes Neuron andere Neuronen aktivieren, mit denen es über Axone und Synapsen verbunden ist.

Eine Möglichkeit Künstliche Intelligenz zu realisieren wird darin gesehen, das biologische neuronale Netz durch ein Künstliches Neuronales Netz (kurz: KNN, engl. ANN = Artificial Neural Network) im Rechner zu simulieren. Die biologischen Vorgänge des menschlichen Denkens und Lernens (Aktivierung von Neuronen, chemische Veränderung von Synapsen usw.) werden, so gut wie möglich, mathematisch beschrieben und in Software oder Hardware modelliert. Der erste erfolgreiche Neurorechner Mark I Perceptron wird 1957/58 von Frank Rosenblatt, Charles Wightman und Mitarbeitern am MIT entwickelt (vgl. Abb. 1, Quelle: Habilitationsschrift des Autors).



Abb. 1: Frank Rosenblatt mit seinem Mark I Perceptron 1958

Spezialisiert auf Mustererkennungsprobleme kann dieser Analogrechner mit einem 20 mal 20 Pixel großen Bildsensor einfache Ziffern erkennen. Der Mark I Perceptron funktioniert mit Hilfe von 512 motorgetriebenen Potentiometern, d. h. ein Potentiometer für jedes der variablen, trainierbaren Gewichte im künstlichen neuronalen Netz, das eine sogenannte Perzeptron-Topologie hat (Abb. 2, Quelle: Habilitationsschrift des Autors). Bis heute ist es noch nicht annähernd möglich ein menschliches Gehirn in einem Rechner durch künstliche neuronale Netze zu simulieren, da zum einen die Anzahl der künstlichen Neuronen um mehrere Zehnerpotenzen geringer ist und da zum anderen die hochgradig parallele Verknüpfung vieler künstlicher Neuronen noch nicht möglich ist. Die Verknüpfungen zwischen den künstlichen Neuronen werden als *KNN-Gewichte* und die Art der Aktivierung und Vernetzung als *KNN-Topologie* bezeichnet.

Für das *Training* eines künstlichen neuronalen Netzes sind zwei wesentliche Verfahrensklassen zu unterscheiden: Das *überwachte Lernen* und das *unüberwachte Lernen*. Beim überwachten Lernen werden einem KNN Input-/Outputzusammenhänge, die Muster, präsentiert. Es wird ein Fehlervektor berechnet, dessen Komponenten die Differenzen zwischen den aktuellen KNN-Outputs und den Soll-Outputs für die Inputs sind. Im Idealfall wird dem künstlichen neuronalen Netz durch geeignete *Gewichte- und Topologieoptimierung* sukzessive die Fähigkeit antrainiert, selbständig und ausreichend genau den zu einem Input gehörenden Soll-Output zu liefern, vgl. z. B. sogenannte 3- und 4-lagige Perzeptrons und Radial-Basis-Netze sowie Abb. 2. „E“ steht dabei für Eingabe, „A“ für Aktivierung, „+“ für die Summe der eingehenden Aktivierungen multipliziert mit den Gewichten: Die Auswertung der künstlichen neuronalen Netze erfolgt dabei von links nach rechts den Pfeilen folgend. „Bias“ steht für ein Neuron mit konstanter Aktivierung, das einen Schwellenwert für die Aktivierung nachfolgender Neuronen liefert.

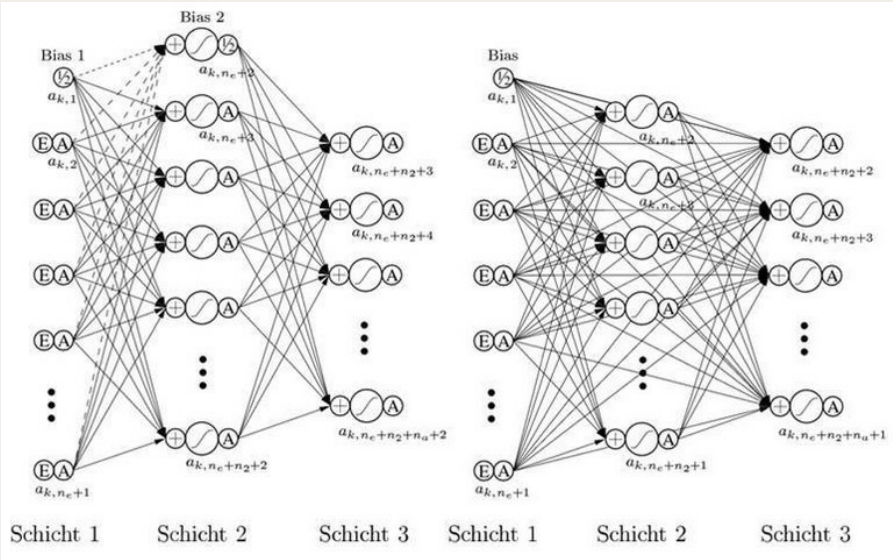


Abb. 2: Vollständig verbundene, dreilagige Perzeptrons ohne (links) und mit (rechts) Direktverbindungen

Ein gut trainiertes künstliches neuronales Netz soll möglichst oft auch für ähnliche, verzerrte, verrauschte oder unvollständige Inputs den richtigen Soll-Output liefern (*Generalisierungseigenschaft*). Dies entspricht den Anforderungen betriebswirtschaftlicher Entscheidungsprozesse, die oft nur unvollständige, ungenaue oder unsichere Informationen enthalten. Beim unüberwachten Lernen werden einem künstlichen neuronalen Netz nur Inputs präsentiert. Das KNN soll sich dann selbständig adaptieren, indem die Gewichtsvektoren den Eingabevektoren ähnlicher gemacht werden. Dieser Vorgang wird auch als Selbstorganisation bezeichnet. Die Neuronen sind in diesem Fall Repräsentanten typischer Eingabedaten und bilden sogenannte Cluster, vgl. z. B. sogenannte selbstorganisierende Karten, Assoziativspeicher und Kohonen-Netze.

Für die meisten Praxisprobleme in der Wirtschaftsinformatik ist das überwachte Lernen mit mehrlagigen Perzeptrons, von besonderer Bedeutung. Wichtige Anwendungsbereiche sind z. B.

- Zeitreihenanalysen und -prognosen,
- Bild- und Schrifterkennung, insbes. OCR, sowie
- Rating-, Kontroll- und Warnsysteme mit Mustererkennung.

Moderne Neurosimulatoren, z. B. der Hannoveraner Neurosimulator FAUN (Fast approximation with universal neural networks), nutzen

- moderne Hochleistungsoptimierungsverfahren, z. B. SQP-Verfahren, um akzeptable Rechenzeiten zu erreichen,
- Ansätze der globalen Optimierung, um bei der Minimierung des Fehlers das Problem der extrem vielen schlechten lokalen Minima zu vermeiden,
- lineare und nichtlineare Beschränkungen für die Gewichte, u. a. zur Konvergenzoptimierung und zur Einhaltung von Krümmungs- oder Konvexitätsrestriktionen,
- graphische Benutzeroberflächen mit diversen online- und offline-Visualisierungen des Trainings und der KNN-Gewichte,
- automatische Aufteilung der Muster in Trainings-, Crossvalidierungs- und Generalisierungsdaten.

Einige nützliche Links:

- <http://www.snn.ru.nl/enns> (European Neural Network Society)
- http://www.ieee-cis.org/about_cis (IEEE Computational Intelligence Society)
- <http://www.gi-ev.de/gliederungen/fachbereiche/kuenstliche-intelligenz-ki> (Gesellschaft für Informatik, Fachbereich KI)
- <http://www.nips.cc> (Neural Information Processing Systems)

Autor



Prof. Dr. Michael H. Breitner, Leibniz Universität Hannover, Institut für Wirtschaftsinformatik, Königsworther Platz 1, 30167 Hannover

[Autoreninfo](#)

Zuletzt bearbeitet: 26.09.2014 13:23

Letzter Abruf: 09.08.2016 16:53

[Kommentare](#)



[Add Comment](#)