1. Einführung

1.1 Biologische Neuronen und Berechenbarkeitsaspekte
1.2 Rolle von künstlichen neuronalen Netzen
1.3 Historischer Überblick
1.4 Charakterisierung von neuronalen Netzen

1.1 Biologische Neuronen und Berechenbarkeitsaspekte

Überblick:

- Nervenzellen im Gehirn
- Strukturelle Komponenten der Nervenzellen
- Prinzipien neuronaler Informationsverarbeitung
- Berechenbarkeitsaspekte

Nervenzellen im Gehirn

- Große Zahl: $\mathbf{10^5}$ pro $\boldsymbol{mm^3}$, Hirn gesamt: $\mathbf{10^{11}}$.
- Relativ homogene Grundstruktur.
- Hohe Spezialisierung.
- Bilden Gruppen.

Strukturelle Komponenten der Nervenzellen

Zellkörper: Verarbeitung von Informationen.

Dendriten: Aufnehmen von Informationen anderer Zellen.

Axone: Weiterleiten von Informationen.

Synapsen: Speichern von Informationen:

Anregende oder Hemmende Synapsen.

Bezug zu künstlichen Neuronen:

Hohe Reizschwelle $\equiv$ Niedriges Gewicht.

Prinzipien neuronaler Informationsverarbeitung

- Langsam im Vergleich zu elektrischen Schaltnetzen: biologisch $\mathbf{1}\boldsymbol{\mu s} - \mathbf{1}\boldsymbol{ms}$, elektrisch $\mathbf{1}\boldsymbol{ns}$.
- Prozessoren (Zellkörper) und Instruktionssatz sehr einfach.
- Massive Parallelität.
- Vernetzungsstrukturen (pro Neuron $\mathbf{10^4}$ Synapsen).

Prinzipien neuronaler Informationsverarbeitung

- Informationen über viele Freiheitsgrade verteilt (Reizschwellen der Synapsen, Vernetzungsstruktur der Neuronen, Kodierung der Signale, etc.).
- Nicht programmierbar, aber lernfähig.
- Enger Bezug zwischen Gehirnregion und Funktion.
- Selbstorganisierendes, adaptives dynamisches System.

Berechenbarkeitsaspekte

Künstliche Neuronale Netze:

Paralleler Ablauf von vielen verbundenen Berechnungselementen.

Algorithmus wird nicht manuell programmiert, sondern Netzstruktur und Netzparameter werden automatisch ermittelt.

→ Robustheit, d.h.
unempfindlich gegen sporadisch fehlerbehaften Eingaben.

→ Adaptivität, d.h.
anpassbar an geänderte Bedingungen.

1.2 Rolle von künstlichen neuronalen Netzen

Klassifikation (z.B. Mustererkennung)

Funktionsapproximation (z.B. Sensor-Motor-Abbildungen)

1.2 Rolle von künstlichen neuronalen Netzen

Bereiche	**Anwendungen**
Industrie	Qualitätskontrolle, Robotersteuerung
Finanzen	Wertpapier-/Aktienbewertung
Telekommunikation	Optimierung des Signalverkehrs
Medizin	Diagnose
Marketing	Werbewirksamkeit bestimmen
Öffentlicher Dienst	Verarbeitung Formulare, Handschrifterkennung
Auto/Verkehr	Motorsteuerung, Hinderniserkennung

1.3 Historischer Überblick

Zeitphase	**Forscher**	**Forschungsbereich**
I. 1940-69:	McCulloch-Pitts	Logikelemente
Frühe Enthusiasmusphase;	Hebb	Synaptische Lernregel
Entwicklung grundlegender Konzepte.	Rochester	Erste Computer-Simulation
	Rosenblatt	Perzeptron
	Widrow	Adaline

1.3 Historischer Überblick

Zeitphase	**Forscher**	**Forschungsbereich**
II. 1969-82:	Grossberg	ART 1
Frustationsphase;	Kohonen	Assoziativspeicher
Kaum Arbeiten.		

1.3 Historischer Überblick

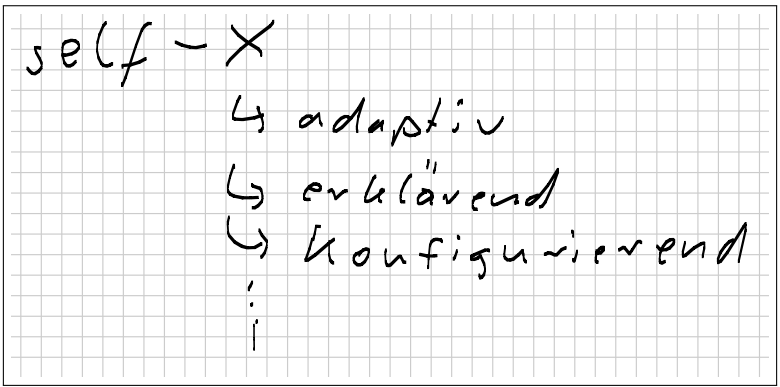
Zeitphase	Forscher	Forschungsbereich
III. 1982-2000: Renaissancephase;	Rumelhart	Backpropagation Multi-Layer-Perzeptron
Lösung nichtlinearer Probleme;	Poggio	Radiale-Basisfunktionen-Netze
Erste praktische Anwendungen.	Kohonen	Selbstorganisierende, topologische Karten
	Martinetz, Fritzke	Neural Gas, Dynamische Netze
	Vapnik	Support-Vektor Maschinen

1.3 Historischer Überblick

Zeitphase	**Forscher**	**Forschungsbereich**
IV. 2001-heute:	Müller-Schloer	Organic Computing
Für Menschen;	Breiman	Random Forest
Zuverlässigkeit;	Bengio, LeCun	Deep Learning
Vielfältige praktische Relevanz.		

1.3 Historischer Überblick

Einschub: Organic Computing

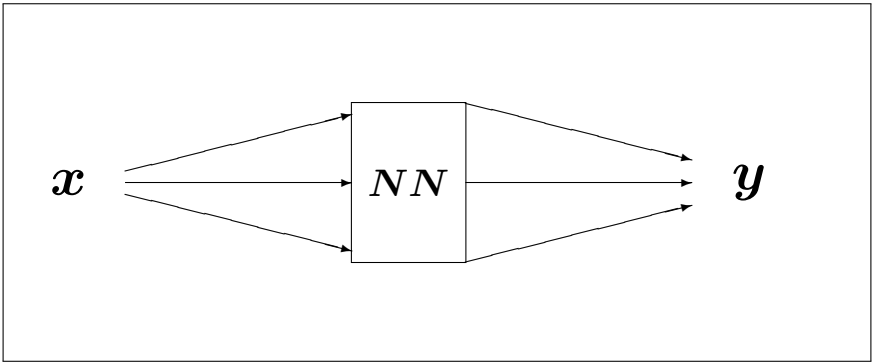


1.4 Charakterisierung von neuronalen Netzen

Überblick:

- Netzwerktopologie
- Knotenfunktionen
- Datenfluß
- Neuronales Lernen

Netzwerktopologie



Netzwerktopologie

$\boldsymbol{x} := (\boldsymbol{x_1}, \ldots, \boldsymbol{x_I})^T$: Input-Vektor bildet Eingabeschicht.

$\boldsymbol{y} := (\boldsymbol{y_1}, \ldots, \boldsymbol{y_O})^T$: Output-Vektor bildet Ausgabeschicht.

$\boldsymbol{NN}$: Verbindungsstruktur von verdeckten Knoten.

Knoten: Verarbeitungselement
Kanten: Datenfluß

} gerichteter bewerteter Graph

Beispiel für eine Verbindungsstruktur: Geschichtete Anordnung von Knoten und *vollständige Verbindung* zwischen benachbarten Schichten.

Netzwerktopologie

Formal: Quintupel $\mathcal{A} := (\mathcal{K}, \mathcal{V}, \mathcal{I}, \mathcal{O}, \mathcal{H})$

$\mathcal{K}$: Knotenmenge

$\mathcal{V}$: Kantenmenge $\mathcal{V} \subset \mathcal{K} \times \mathcal{K}$

$\mathcal{I}$: Eingabeknoten $\mathcal{I} \subset \mathcal{K}$

$\mathcal{O}$: Ausgabeknoten $\mathcal{O} \subset \mathcal{K}$

$\mathcal{H}$: Verdeckte Knoten: $\mathcal{H} := \mathcal{K} \backslash (\mathcal{I} \cup \mathcal{O})$

Netzwerktopologie

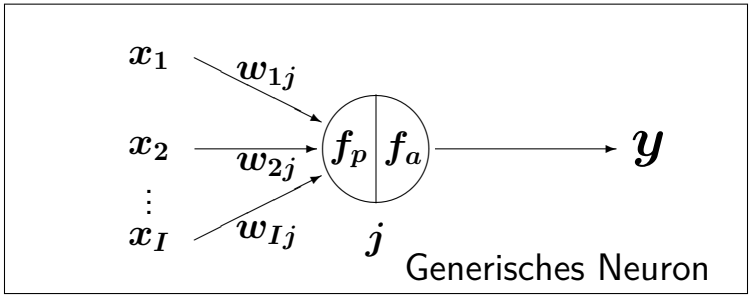
Kantengewicht w_{ij}:

i-tes Neuron aus vorhergehender Schicht.

j-tes Neuron aus nächster Schicht.

Gewichtsvektor $w := (w_{1j}, \ldots, w_{Ij})^T$

Kantengewichte und Knotenfunktionen



$$y := f_a(f_p(x^T, w^T))$$

f_p, f_a werden oft nicht angegeben, wenn sie einheitlich für alle Knoten definiert wurden, und diese aus dem Zusammenhang klar sind.

Knotenfunktionen

f_p: Propagierungs- oder Inputfunktion. Berechnung der aktuellen Eingabe an internen Knoten anhand der Ausgabe vorgeschalteter Knoten und der Gewichte der Verbindungen.

f_a: Aktivierungs- oder Übertragungsfunktion. Bestimmung des neuen Aktivierungszustandes des internen Knotens in Abhängigkeit vom propagierten Wert und eventuell vom vorherigen Aktivierungszustand.

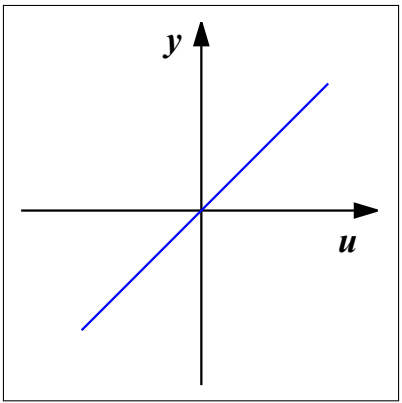
Knotenfunktionen

Beispiele für Propagierungsfunktionen:

- $u_j := \sum_i w_{ij} x_i$ (linearer Assoziator)
- $u_j := \prod_i w_{ij} x_i$ (nicht-linearer Assoziator)
- $u_j := \max_i \{w_{ij} x_i\}$ (Maximum gewichtete Eingaben)
- $u_j := \sum_i s_i$, mit $s_i := \begin{cases} +1 & : \text{ falls } w_{ij} x_i > 0 \\ -1 & : \text{ sonst} \end{cases}$

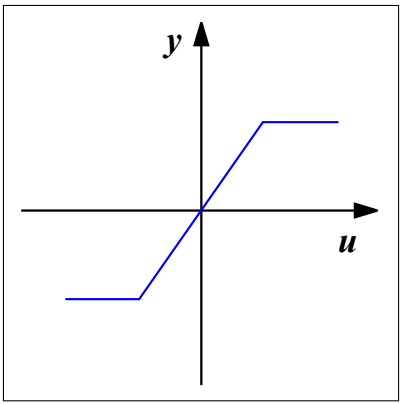
Knotenfunktionen

Beispiel für Aktivierungsfunktion: Identität



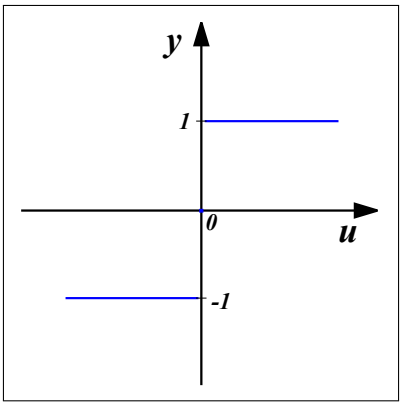
Knotenfunktionen

Beispiel für Aktivierungsfunktion: Rampe



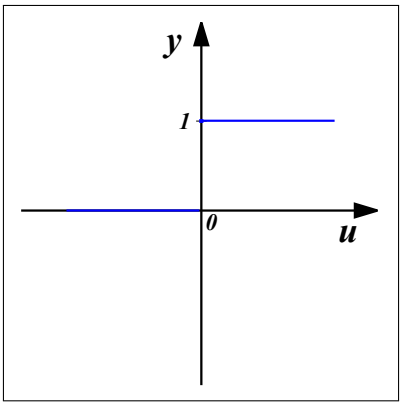
Knotenfunktionen

Beispiel für Aktivierungsfunktion: Signum



Knotenfunktionen

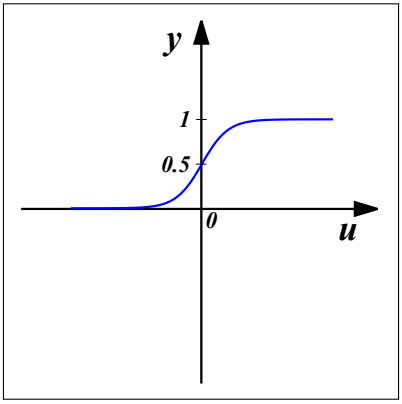
Beispiel für Aktivierungsfunktion: Stufe



Knotenfunktionen

Beispiele für Aktivierungsfunktion: Sigmoid

$y(u) := \frac{1}{1+e^{-\beta u}}$



Parameter β beeinflußt die Steigung des Rampenabschnitts.

Knotenfunktionen

Erweiterung des Neuronenmodells:

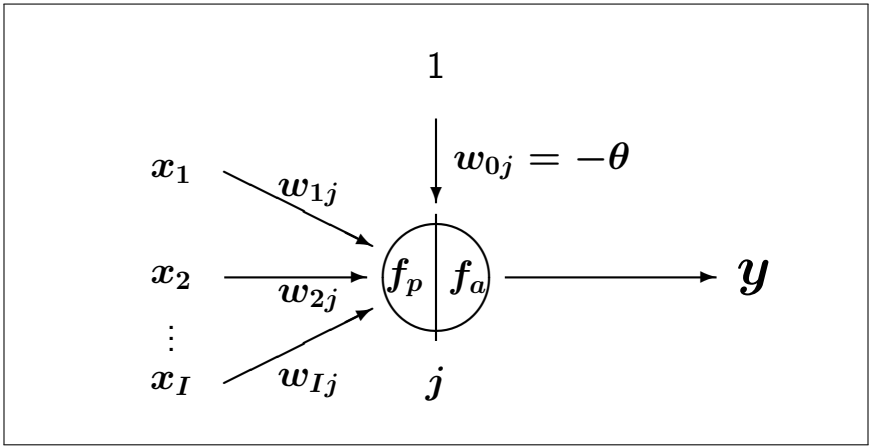
Einführung eines Schwellenwertes $\boldsymbol{\theta}$; oft muß propagierter Wert erst diese Schwelle überschreiten, um sich am Ausgang des Knotens auszuwirken.

$$\boldsymbol{x^e := (1, x_1, \ldots, x_I)^T}$$

$$\boldsymbol{w^e := (\underbrace{-\theta}_{=:w_0}, w_1, \ldots, w_I)^T}$$

$$\boldsymbol{u := w^{eT} x^e = w_0 + \sum_{i=1}^{I} w_i x_i}$$

Knotenfunktionen



Knotenfunktionen

Wechselwirkungen höherer Ordnung (Higher Order Networks)**:**

$$u := w_0 + \sum_i w_i x_i + \sum_{j,l} w_{jl} x_j x_l + \dots$$

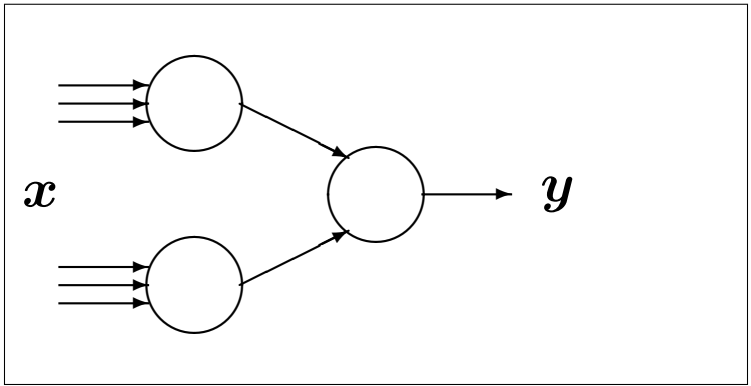
Datenfluß

Gemeint ist hier der Datenfluß während der Anwendungsphase eines gelernten Netzes.

Der sog. Aktualisierungmodus bestimmt, wann ein einzelnes Neuron den Aktivierungszustand aktualisiert.

Datenfluß

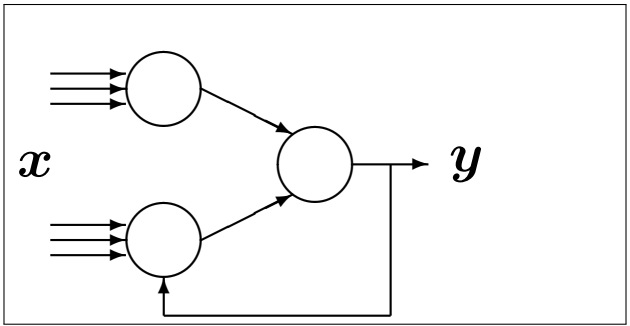
Vorwärts gerichtetes Netz: Rein vorwärtsgerichteter Datenfluß.



Zeitpunkt der Berechnung einzelner Knoten belanglos für das Endgebnis.

Datenfluß

Rückgekoppelte (rekursive, rekurrente) Netze: Informationsfluß mit Rückkopplung.



Berechnungsprozeß nicht mehr allein durch die Vernetzung festgelegt. Zeitliche Abfolge der einzelnen Berechnungen entscheidend.

Neuronales Lernen

Modifikation

- Gewichte der Verbindungen
- Struktur des Netzes

Lernregel bestimmt, wie und wann die Gewichte der Verbindungen und/oder die Struktur des Netzes zu ändern sind.

Ziel ist hierbei, eine bestimmte (Mindest-) Qualität für die Klassifikation bzw. Funktionsapproximation zu erhalten.

Neuronales Lernen

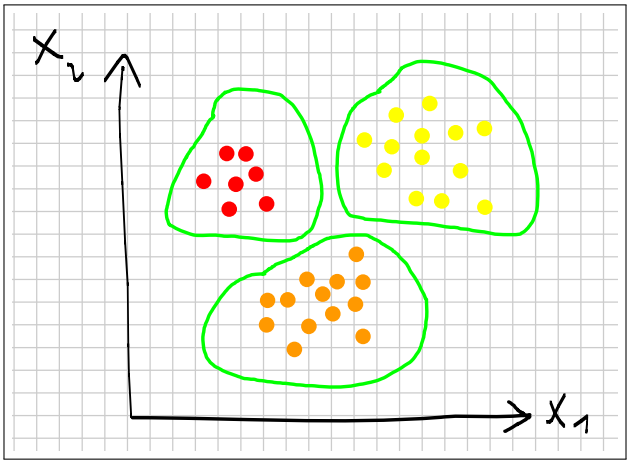
Lernkonzept Überwachtes Lernen (Supervised Learning):

Erfordert Lehrer, der Menge von Stichprobenelementen $\boldsymbol{x}^m$ zusammen mit den dafür korrekten Sollantworten $\boldsymbol{y}^m$ vorgibt, $\boldsymbol{m = 1, 2, \ldots}$

Lernen erfolgt offline.

Neuronales Lernen

Einschub: Beispiel für überwachtes Lernen zur Klassifikation



Neuronales Lernen

Lernkonzept Unüberwachtes Lernen (Unsupervised Learning):

Erfordert nur Stichprobenelemente $\boldsymbol{x}^m$.

Ziel ist das Finden von Gruppen/Clustern/Strukturen/ Zusammenhängen in Stichprobenmenge.

Lernen erfolgt offline oder online.

Neuronales Lernen

Lernkonzept Belohnungs-/Bestrafungslernen (Reinforcement L.):

Erfordert Stichprobenelemente $\boldsymbol{x}^m$ und Rückmeldung einer (lokalen) Bewertung $\boldsymbol{r}^m$.

Die lokale Bewertung erhält man durch Sensoren, z.B. "Kollision passiert (negativ)", oder "Fahrtziel erreicht (positiv)".

Es gibt keinen Lehrer, d.h. es gibt keine Beispiele für optimales Systemverhalten (d.h. keine Beispiele von optimalen Roboterfahrten).

Aufgabe ist das Lernen einer Entscheidungsfunktion für ein (global) optimales Systemverhalten.

Lernen erfolgt online, z.B. Roboter während der Fahrt.