

Neuronale Netze/ Soft Computing

Teil 1

**BiTS, Wintersemester 2004/2005
Dr. Stefan Kooths**

Gliederung

1. Einführung und Einordnung
2. Neuronale Netze 1: Grundlagen
3. Neuronale Netze 2: Konzeption und Anwendung
4. Neuro-Fuzzy-Systeme
5. Genetische Algorithmen
6. Zusammenfassung und Ausblick

Gliederung

1. Einführung und Einordnung
2. Neuronale Netze 1: Grundlagen
3. Neuronale Netze 2: Konzeption und Anwendung
4. Neuro-Fuzzy-Systeme
5. Genetische Algorithmen
6. Zusammenfassung und Ausblick

Einordnung: NN/SC ↔ EUS bzw. Business Intelligence

Diplom: EUS

- Entscheidungslehre/
Simulation von
Entscheidungen
- Wissensbasierte
Systeme/Experten-
systeme

- Künstliche Intelligenz
- *Neuronale Netze*

BA/MA: BI

- Management Support
Systems
- Knowledge-Based
Systems
- *Soft Computing*
- Case Studies

Prüfungsleistung: Projektarbeit

Einordnung in die BI-relevanten KI-Gebiete

	Wissensbasierte Systeme (Expertensysteme)	Soft Computing	Multiagentensysteme (verteilte Intelligenz)
Kennzeichnung	<p>Trennung von Wissensbasis und Problemlösungsstrategie</p> <p>Erklärung der gefundenen Lösung</p>	<p>Lernfähigkeit und Toleranz gegenüber unscharfem, ungewissem und unvollständigem Wissen</p>	<p>Emergente Problemlösung durch Kooperation vieler kontextbezogener, autonomer und flexibler Module</p>
Teilgebiete	<ul style="list-style-type: none"> - logikbasiert - regelbasiert - framebasiert - modellbasiert - fallbasiert - statistisch <p><i>jeweils für Zwecke der Diagnose, Simulation und Konstruktion</i></p>	<ul style="list-style-type: none"> - Fuzzy Logic - Neuronale Netze - Genetische Algorithmen <p>(- Probabilistisches Schließen)</p>	<ul style="list-style-type: none"> - Agentendesign - Kommunikation - Kooperation und Konfliktlösungen

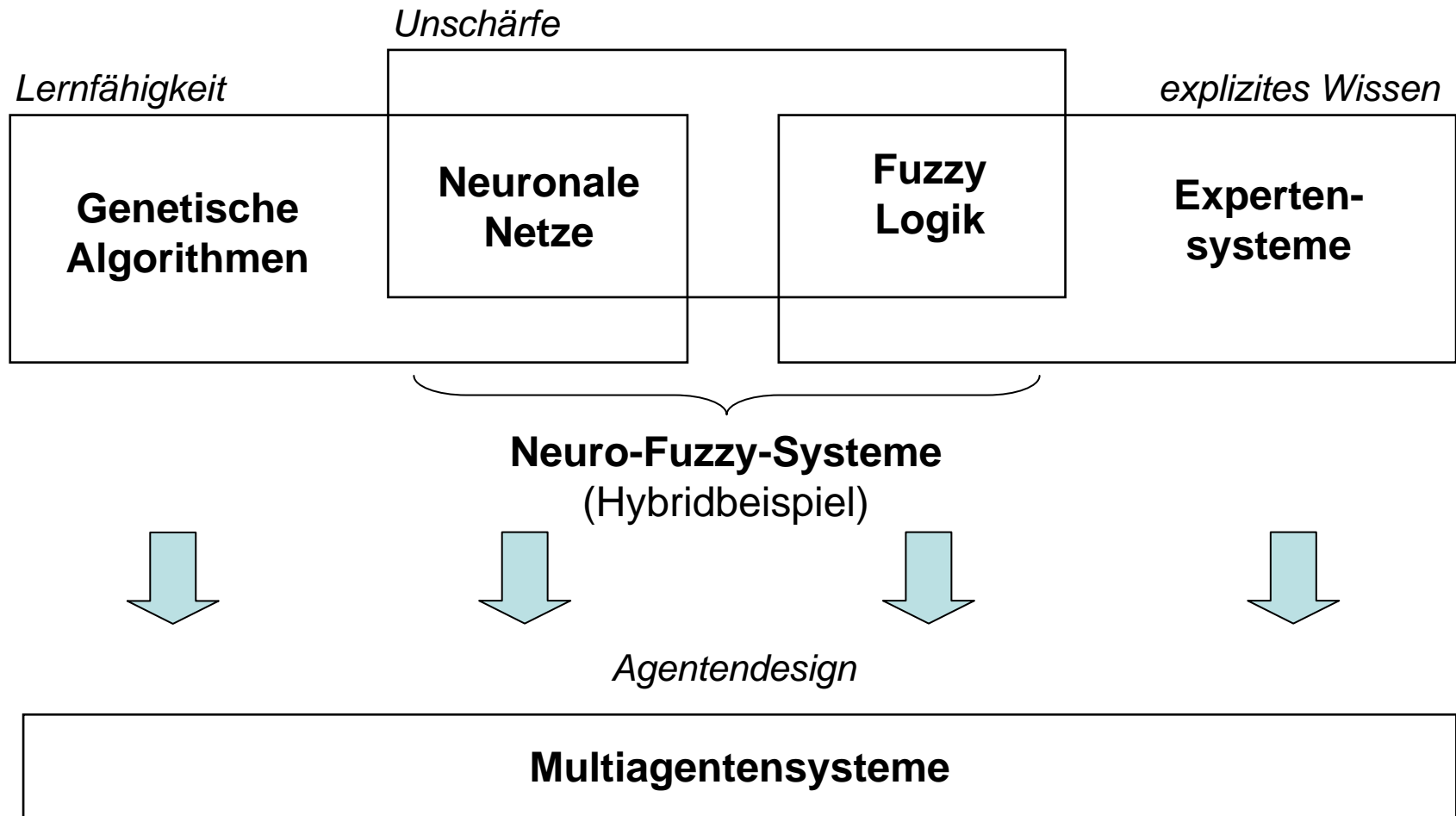
Soft Computing: Kennzeichnung

- Lernfähigkeit und Toleranz gegenüber Ungenauigkeit, Unschärfe, Unsicherheit und Datenfehlern
- Ziel:
 - handhabbare, robuste und kostengünstige Softwarelösungen, ...
 - ... die nicht notwendigerweise perfekt sind, ...
 - ... aber unter den gegebenen Bedingungen als befriedigend gelten können
- natürliche Vorbilder (aber: kein Nachbau, sondern Inspiration)
- 50 % Wissenschaft und 50 % Kunst (= Können = Erfahrung)!

Soft Computing: Teilgebiete (Basistechnologien)

- Neuronale Netze
 - Informationsverarbeitung orientiert am biologischen Vorbild Gehirn
 - Behandlung komplexer, nicht-linearer Systeme zur Mustererkennung
- Fuzzy Logik
 - multivalente (graduelle) Verallgemeinerung der Mengenlehre
 - Repräsentation von approximativem (Experten-) Wissen
 - Weiterentwicklung: Computing with Words
- Genetische Algorithmen
 - evolutionäre Problemlösungsheuristik
 - Problemdarstellung in Chromosomenform, Lösungssuche durch Vererbung, Mutation und Auslese

Soft Computing im KI-Kontext: Überschneidungen und Hybridansätze



Literatur

- Backhaus, K. u.a.: Multivariate Analysemethoden; 10. Aufl., Springer-Verlag, Berlin u.a.O. 2003. [Kap. 12]
- Badiru, A. B. / Cheung, J. Y.: Fuzzy Engineering Expert Systems with Neural Network Applications; John Wiley & Sons 2002. [Ch. 8,9]
- Callan, R.: Neuronale Netze im Klartext; Pearson Studium, München 2003.
- Lange, C.: Neuronale Netze in der wirtschaftswissenschaftlichen Prognose und Modellgenerierung; Physica-Verlag, Heidelberg 2004.
- Luger, G. F.: Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 5th Ed., Addison-Wesley, Upper Saddle River 2005. [Ch. 11,12]
- Kooths, S.: Erfahrungsregeln und Konjunkturdynamik – Makromodelle mit Neuro-Fuzzy-generierten Erwartungen; Peter Lang Europäischer Verlag der Wissenschaften, Frankfurt/M. u.a.O. 1998. [Kap. 2]
- McCord Nelson, M. / Illingworth, W. T.: A Practical Guide to Neural Nets; Addison-Wesley, New York u.a.O. 1994.
- Rehkugler, H. / Zimmermann, H. G. (Hrsg.): Neuronale Netze in der Ökonomie – Grundlagen und finanzwirtschaftliche Anwendungen; Verlag Vahlen, München 1994.
- Turban, E. / Aronson, J. E. / Liang, T.-P.: Decision Support Systems and Intelligent Systems, 7th Ed., Prentice Hall, Upper Saddle River 2005. [Ch. 12]

Weblinks

- Einführung in Neuronale Netze (Uni Münster)
<http://wwwmath.uni-muenster.de/SoftComputing/lehre/material/wwwnnscrip/startseite.html>
- Soft Computing Home Page
<http://www.soft-computing.de>
- Neurocomputing
<http://www.softcomputing.de>

Gliederung

1. Einführung und Einordnung
2. Neuronale Netze 1: Grundlagen
3. Neuronale Netze 2: Konzeption und Anwendung
4. Neuro-Fuzzy-Systeme
5. Genetische Algorithmen
6. Zusammenfassung und Ausblick

(Künstliche) Neuronale Netze

- Verwandte Begriffe
 - Konnektionismus
 - Parallel Distributed Processing
 - Subsymbolisches Paradigma
- Grundidee
 - Verbindung vieler, relativ einfach aufgebauter Einheiten (Neuronen)
 - Leistungsfähigkeit ergibt sich aus Zusammenschaltung und Parallelverarbeitung im Verbund
- Wissenserwerb und -repräsentation
 - Lernen anhand von Beispielen (Training)
 - Kodierung des Wissens liegt in impliziter Form vor (unterhalb der für Menschen verständlichen Symbolebene)

(Künstliche) Neuronale Netze

- Verwandte Begriffe
 - Konnektionismus
 - Parallel Distributed Processing
 - Subsymbolisches Paradigma
- Grundidee
 - Verbindung vieler, relativ einfach aufgebauter Einheiten (Neuronen)
 - Leistungsfähigkeit ergibt sich aus Zusammenschaltung und Parallelverarbeitung im Verbund
- Wissenserwerb und -repräsentation
 - Lernen anhand von Beispielen (Training)
 - Kodierung des Wissens liegt in impliziter Form vor (unterhalb der für Menschen verständlichen Symbolebene)

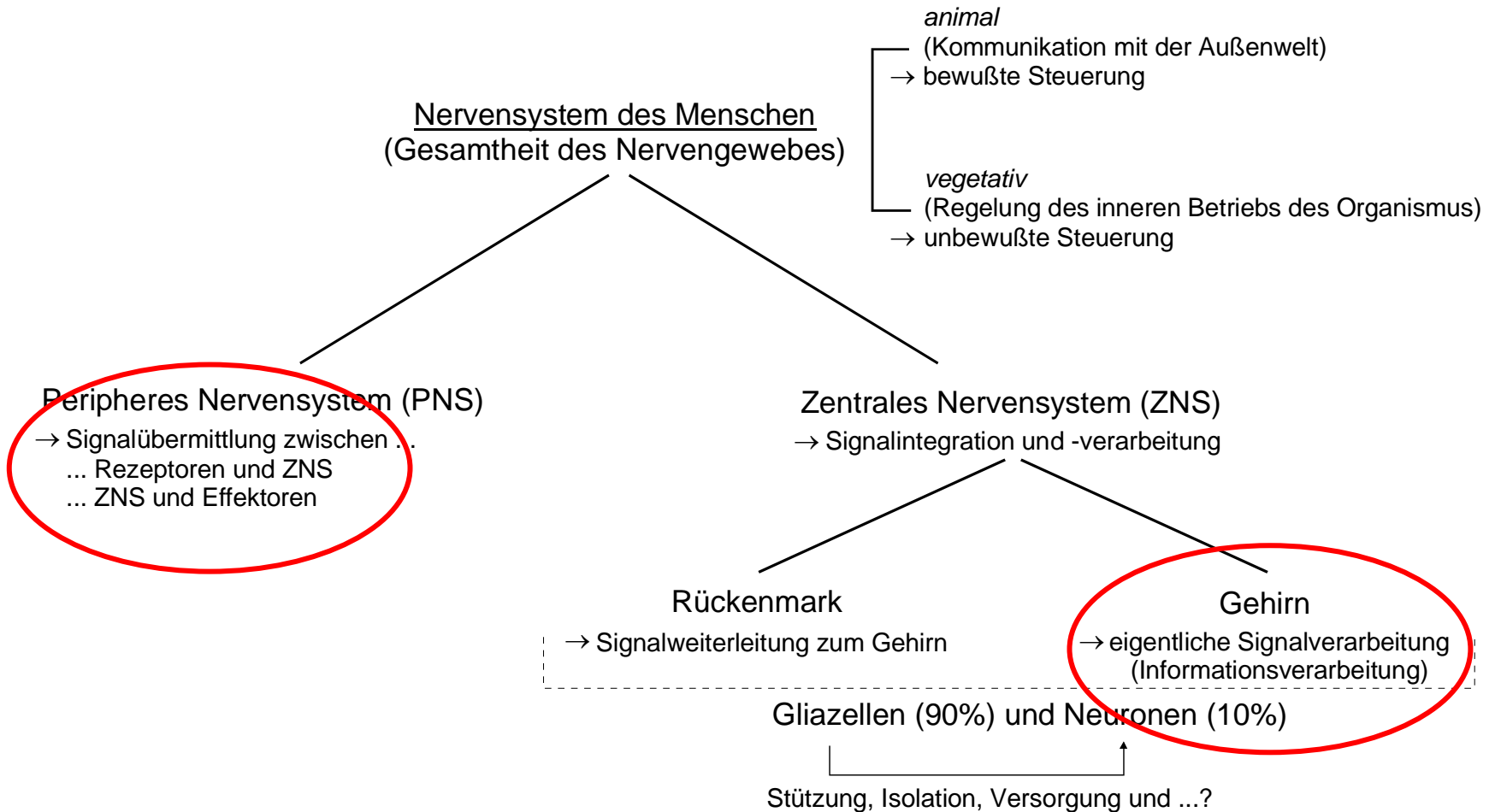
Haupteinsatzgebiete/-eigenschaften

- universelle, beliebig-dimensionale Funktionsapproximation
- Schätzverfahren (Alternative zu statistischen Regressionsanalysen)
- Mustererkennung
- Fehlertoleranz
 - gegenüber Daten
 - gegenüber eigener Struktur
 - ⇒ liegen Daten nur bruchstückhaft vor oder wird ein Teil des Netzes zerstört, so leidet nur die Güte des Ergebnisses, nicht die Fähigkeit des Netzes, überhaupt eine (Näherungs-) Lösung erarbeiten zu können
- Voraussetzung: Repräsentative Trainingsbeispiele für die abzubildenden Zusammenhänge

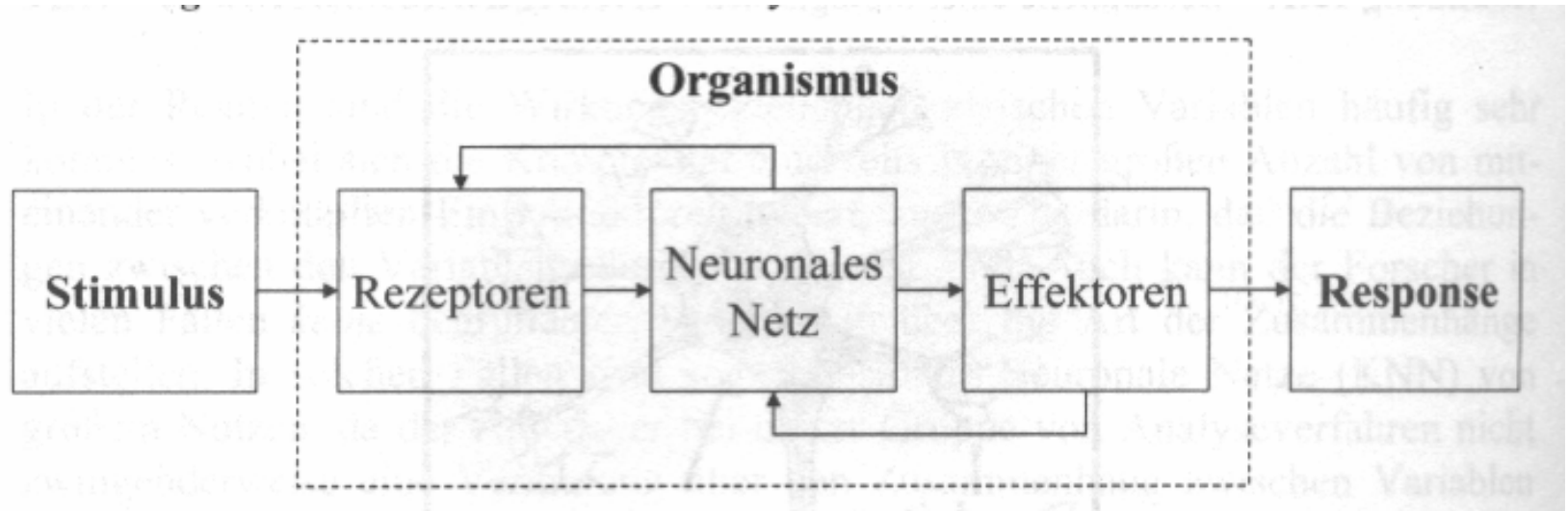
Natürliches Vorbild: Menschliches Gehirn

- Motivation
 - Identifizierung von Ähnlichkeiten (Mustererkennung, Unschärfe- und Fehlertoleranz)
 - Generalisierungsfähigkeit (Einschätzung bislang unbekannter Situationen und Ableitung geeigneter Verhaltensweisen)
 - Lernfähigkeit (Anpassung an neue Umweltzustände durch Erfahrungsbildung)
- Aufbau
 - 10 bis 100 Mrd. (relativ langsame) Neuronen
 - bis zu 10.000 Verbindungen von einem Neuron zu anderen Neuronen
 - Gewicht: 1375 g (Männer), 1245 g (Frauen)

Menschliches Nervensystem



SOR-Modell

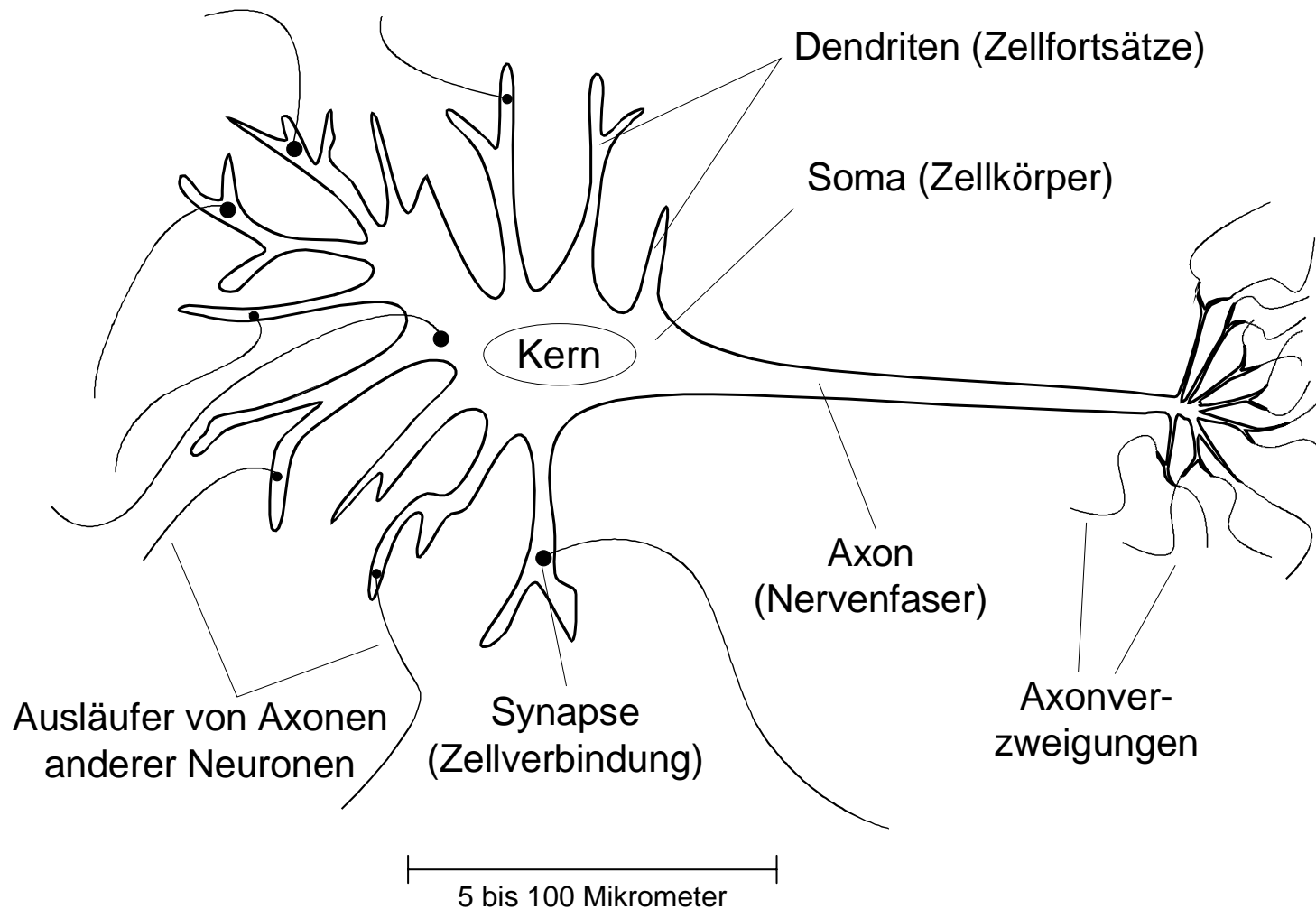


INPUT
(Problem)

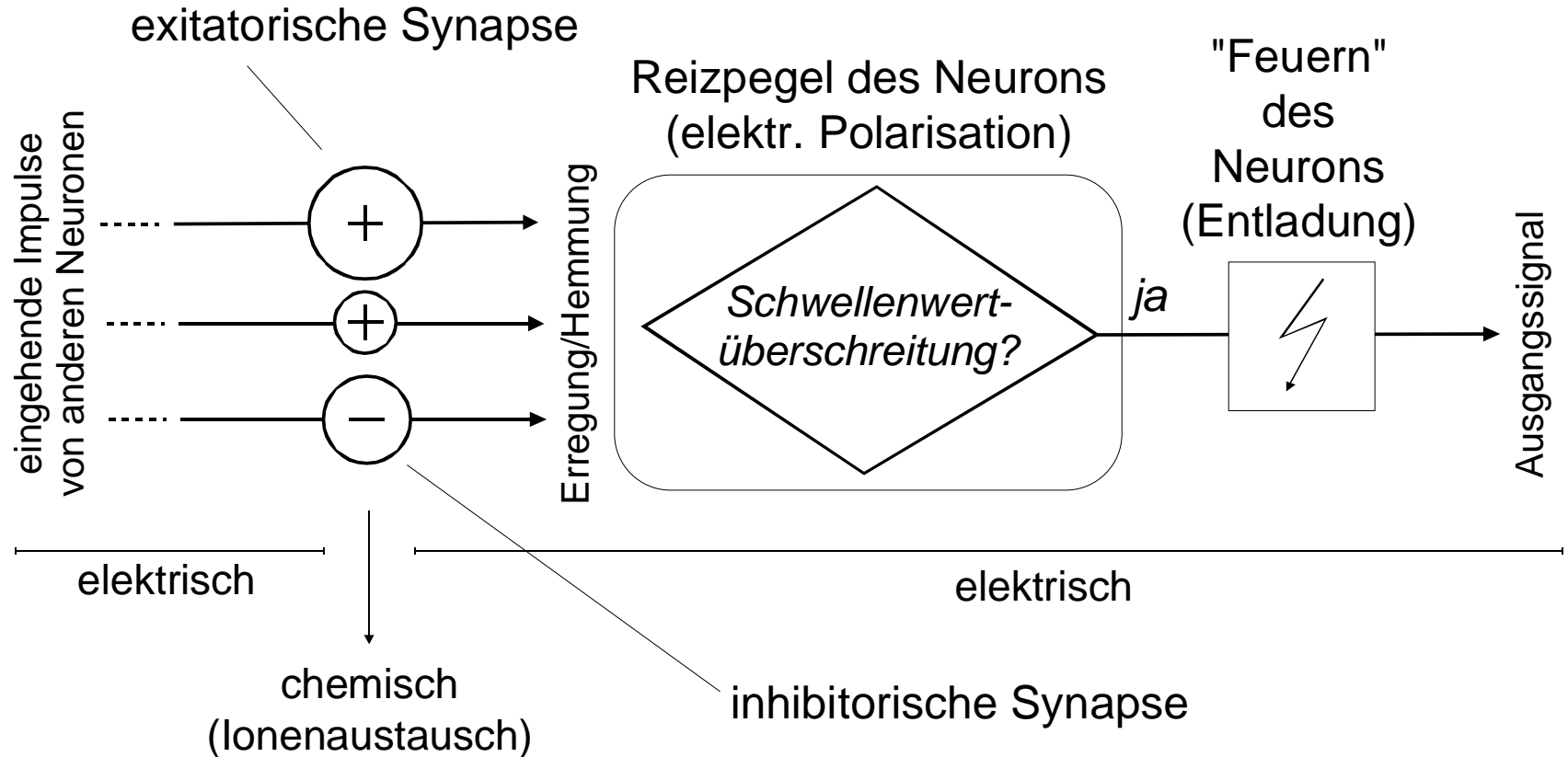
„Informations“-
Verarbeitung

OUTPUT
(Entscheidung)

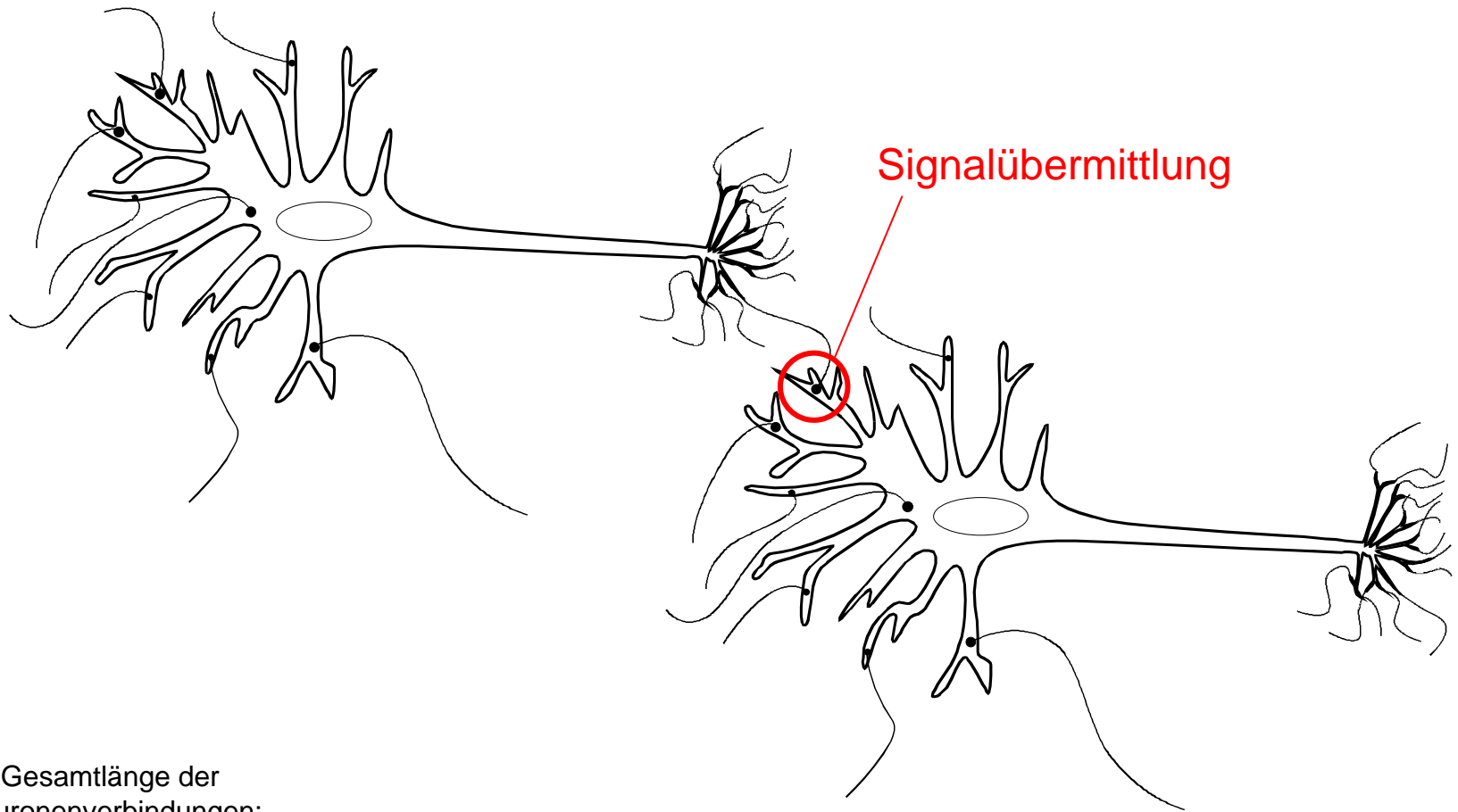
Biologische Nervenzelle



Signalfluss in einem natürlichen Neuron



Vernetzung und synaptische Plastizität



Gesamtlänge der
Neuronenverbindungen:

500 Millionen Meter

„Informationsverarbeitung“, Wissensrepräsentation und -akquisition

- Informationsverarbeitung
 - Signalübermittlung
 - Beeinflussung der neuronalen Aktivitätsniveaus
- Wissensrepräsentation
 - Signale bzw. Aktivitätsniveaus tragen für sich genommen keine sinnhafte Bedeutung
 - implizite Abbildung symbolischer Informationen durch das (eigenorganisierte) Aktivierungsmuster einer Gruppe von Neuronen
 - Wissen steckt im Netzaufbau (Anzahl und Verbindungen der Neuronen) sowie in den Verbindungsgewichten
- Wissensakquisition
 - Trainingsprozesse (Lernen aus Fehlern)
 - Anpassung der Gewichte, Ändern der Topologie

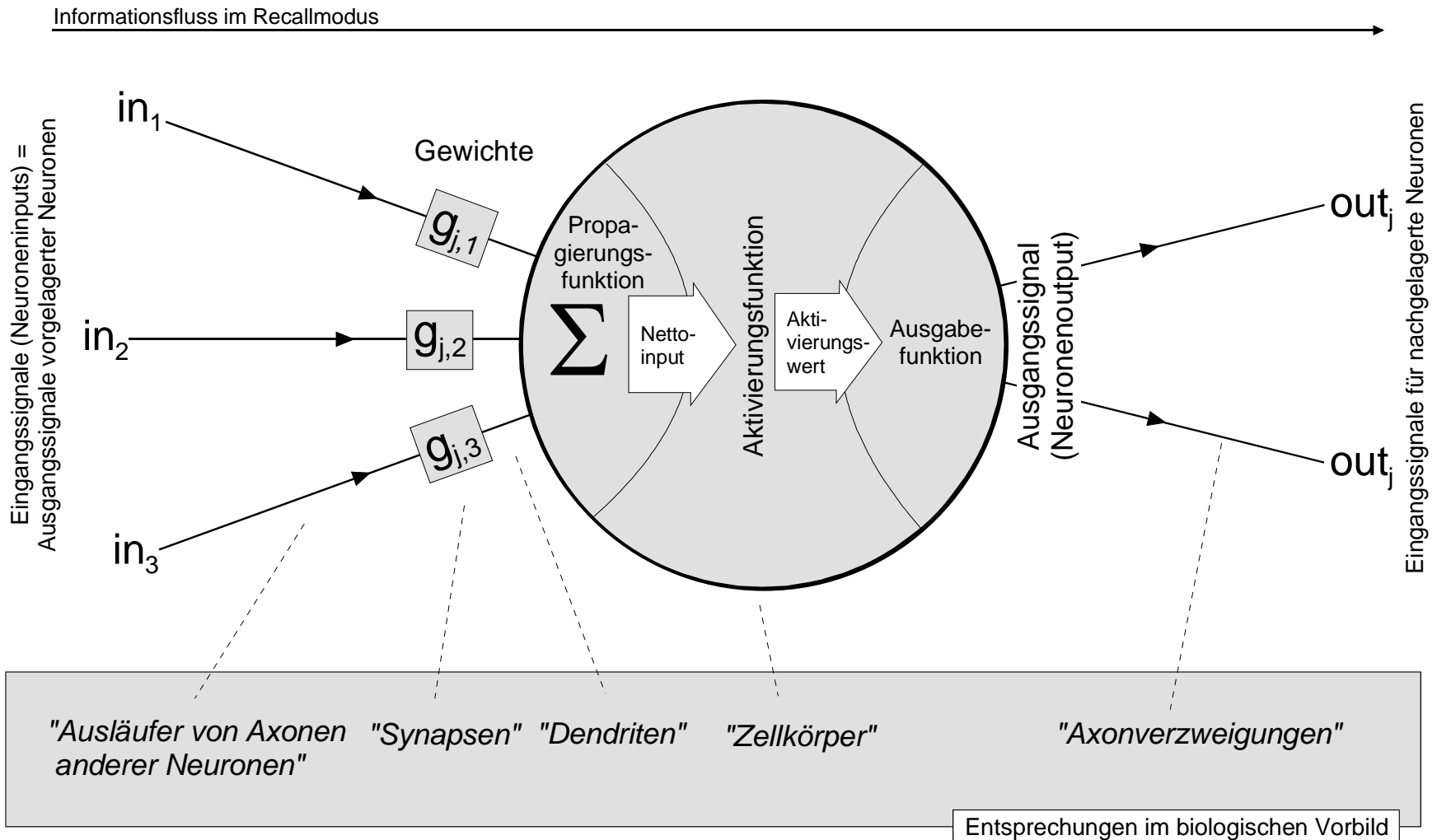
Einsatzmodi

- Recallmodus (Arbeitsmodus)
 - Einsatz des erworbenen Wissens
- Trainingsmodus
 - Wissenserwerb aus Trainingsbeispielen

Formale Neuronen (Processing Units)

- Eigenschaften
 - verfügen über Ein- und Ausgänge
 - empfangen Signale
 - aggregieren diese zu einem Nettosignal
 - generieren aus dem Nettosignal einen eigenen Signalwert
 - senden diesen Signalwert an nachgelagerte Neuronen
- Arten
 - Inputneuronen (Signalempfang von der Außenwelt)
→ vgl. natürliche Rezeptorzellen
 - innere/verdeckte Neuronen (Signalverarbeitung und -verteilung) → interne Repräsentation der Domäne
 - Outputneuronen (Signalweitergabe an die Außenwelt)
→ vgl. natürliche Effektorzellen

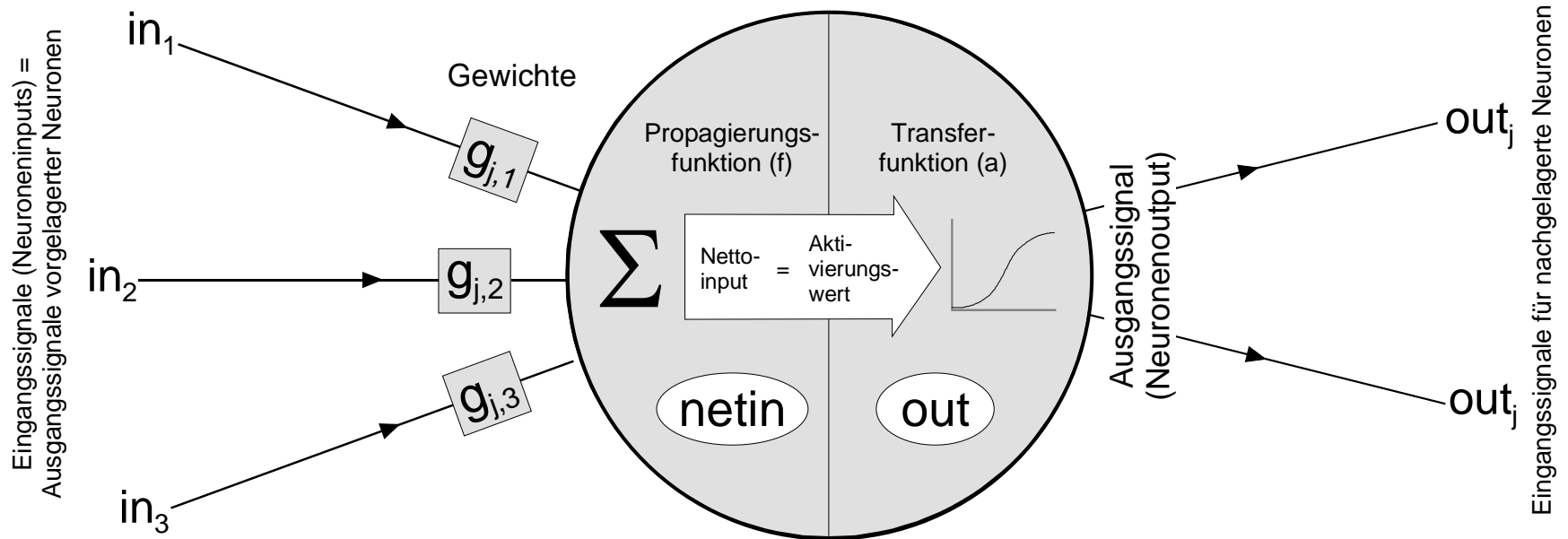
Schematischer Aufbau formaler Neuronen



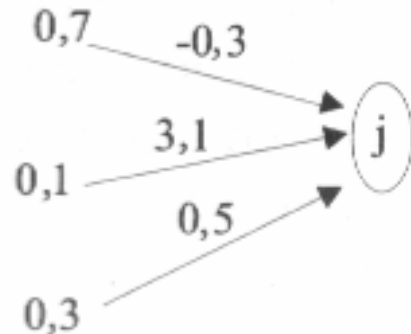
Interne Funktionen eines Neurons

- Propagierungsfunktion: $netin = f(g_1 \cdot in_1, \dots, g_n \cdot in_n)$
 - bestimmt Nettoinput $netin$ durch Summation der gewichteten Eingangssignale
- Aktivierungsfunktion: $Z = g(Z_{-1}, netin)$
 - bestimmt Aktivierungswert Z des Neurons aus (optional: bisheriger Aktivierung und) Nettoinput
- Ausgabefunktion: $out = a(Z)$
 - bestimmt Ausgabewert out aus Aktivierungswert
- Vereinfachung: $out = a(netin)$
 - Transferfunktion als Zusammenfassung von Aktivierungs- und Ausgabefunktion
 - Aktivierungsfunktion als Identität ($Z = netin$)
- beachte: uneinheitliche Terminologie
 - Propagierungsfunktion = Integrationsfunktion
 - Transferfunktion = Aktivierungsfunktion = Ausgabefunktion

Vereinfachtes formales Neuronenschema



Bestimmung des Nettoinputs



$$net_j = \sum_{i=1}^n x_i w_{ij}$$

$$net_j = (0,7 \times -0,3) + (0,1 \times 3,1) + (0,3 \times 0,5) = 0,25$$

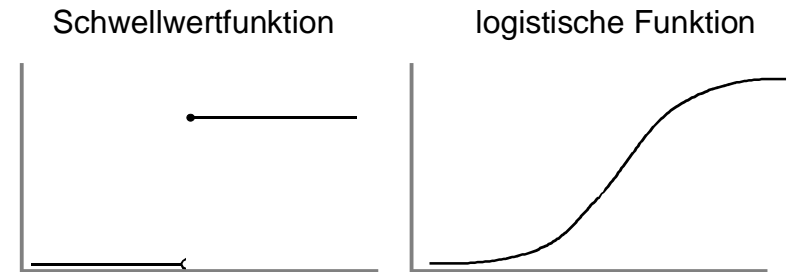
oder alternativ in Vektornotation

$$\begin{bmatrix} 0,7 & 0,1 & 0,3 \end{bmatrix} \begin{bmatrix} -0,3 \\ 3,1 \\ 0,5 \end{bmatrix} = 0,25$$

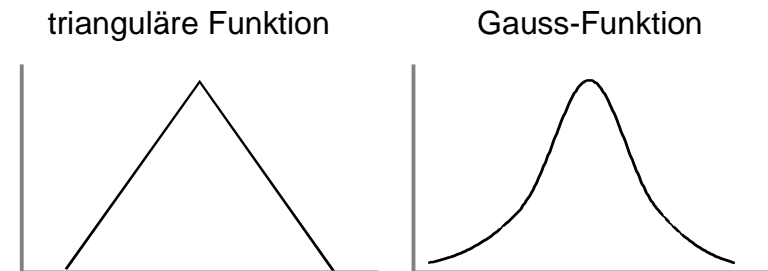
hier:
netin = net
in = x

Transferfunktionen

- Halbraum-aktiv
 - Schwellenwertfunktion
 - sigmoide Funktion



- lokal aktiv
 - trianguläre Funktion
 - Gauß-Funktion



- entscheidend: nicht-lineare Funktionstypen

Komponenten/Eigenschaften eines Neuronalen Netzes

- Neuronen
(Menge von einfachen Verarbeitungseinheiten)
 - Propagierungsfunktion
 - Aktivierungsfunktion
 - Ausgabefunktion } Transferfunktion
- Konnektivitätsmuster
(Struktur der Neuronenverbindungen)
- Festlegung des Signalflusses
(Regel zum Propagieren von Signalen durch das Netzwerk)
- Lernregel zum Anpassen der Gewichte

Systematisierung Neuronaler Netze (Netztypen) 1

- Neuronentyp
- Topologie
- Lernverfahren
- Funktion/Aufgabe (Lerngegenstand)

Systematisierung Neuronaler Netze (Netztypen) 2

- Neuronentyp
 - Bestimmung des Aktivierungszustands
 - $Z = g(Z_{-1}, \text{netin})$
 - $Z = g(\text{netin}) \mid Z = \text{netin}$
 - Funktionstyp der Ausgabefunktion
 - halbraum-aktiv
 - lokal aktiv (Radiale Basisfunktionen)
- Topologie
- Lernverfahren
- Funktion/Aufgabe (Lerngegenstand)

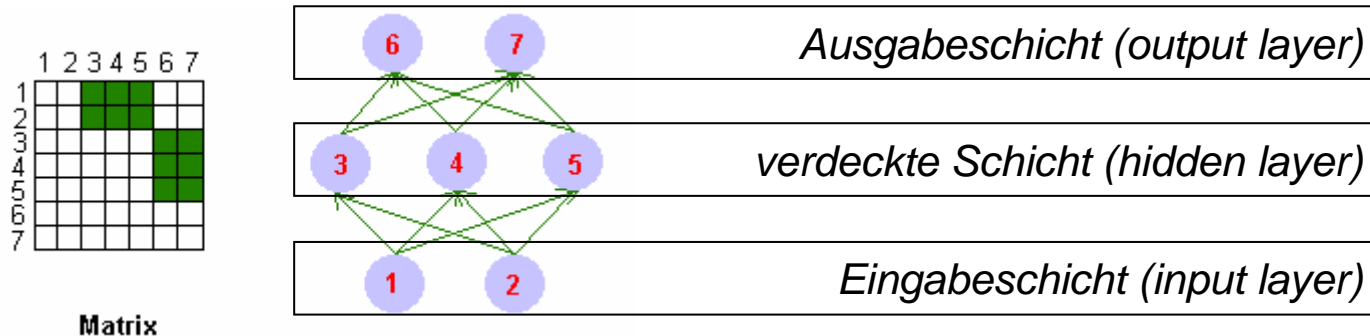
Systematisierung Neuronaler Netze (Netztypen) 3

- Neuronentyp
 - Topologie
 - Anzahl der Neuronen
 - Anordnung der Neuronen
 - geschichtet
 - ungeschichtet
 - Signalfluss
 - ohne Rückkopplung (feedforward)
 - mit Rückkopplung (feedback)
- } Konnektivitätsmuster
- Lernverfahren
 - Funktion/Aufgabe (Lerngegenstand)

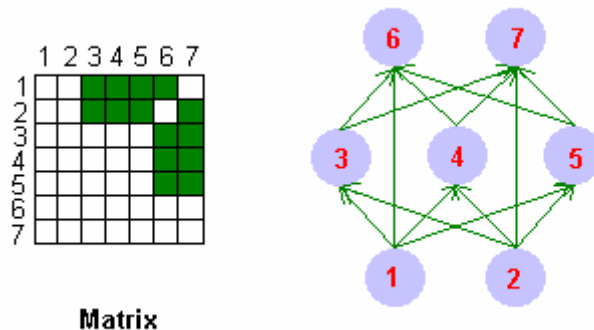
Netztopologien 1:

Feedforward-Netze (Netze ohne Rückkopplung)

- geschichtet (ebenenweise verbunden)

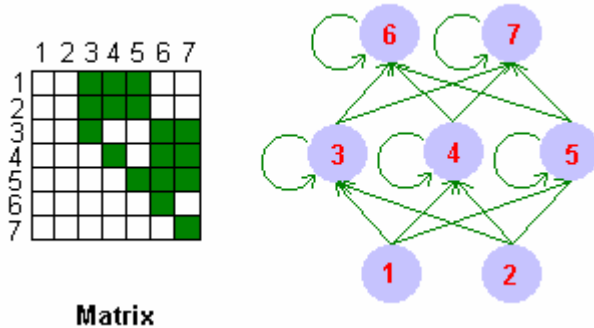


- allgemein (Existenz von Shortcuts)



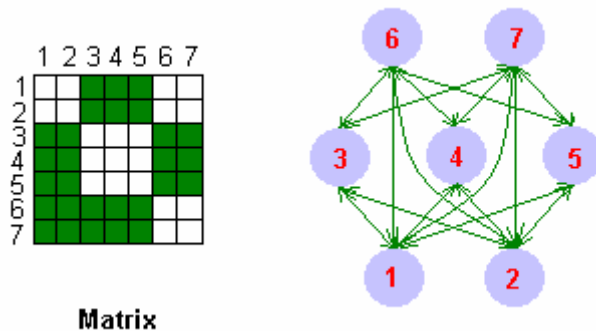
Netztopologien 2: Rekurrente Netze (Netze mit Rückkopplung)

- direkte Rückkopplung (direct feedback)



*Tendenz zu Grenzzuständen
bezüglich der Aktivierung
einzelner Neuronen*

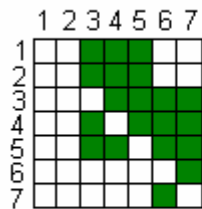
- indirekte Rückkopplung (indirect feedback)



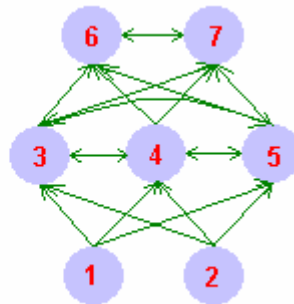
*Aufmerksamkeitssteuerung auf
bestimmte Eingabemerkmale
durch das Netz*

Netztopologien 3: Netze mit Rückkopplungen (Forts.)

- laterale Kopplung



Matrix

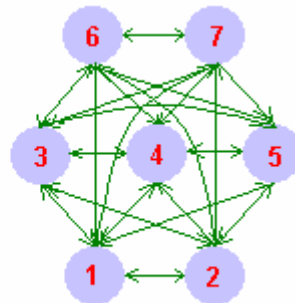


*oftmals als winner-take-all Netze genutzt
(laterale Hemmung)*

- vollständig verbundene Netze

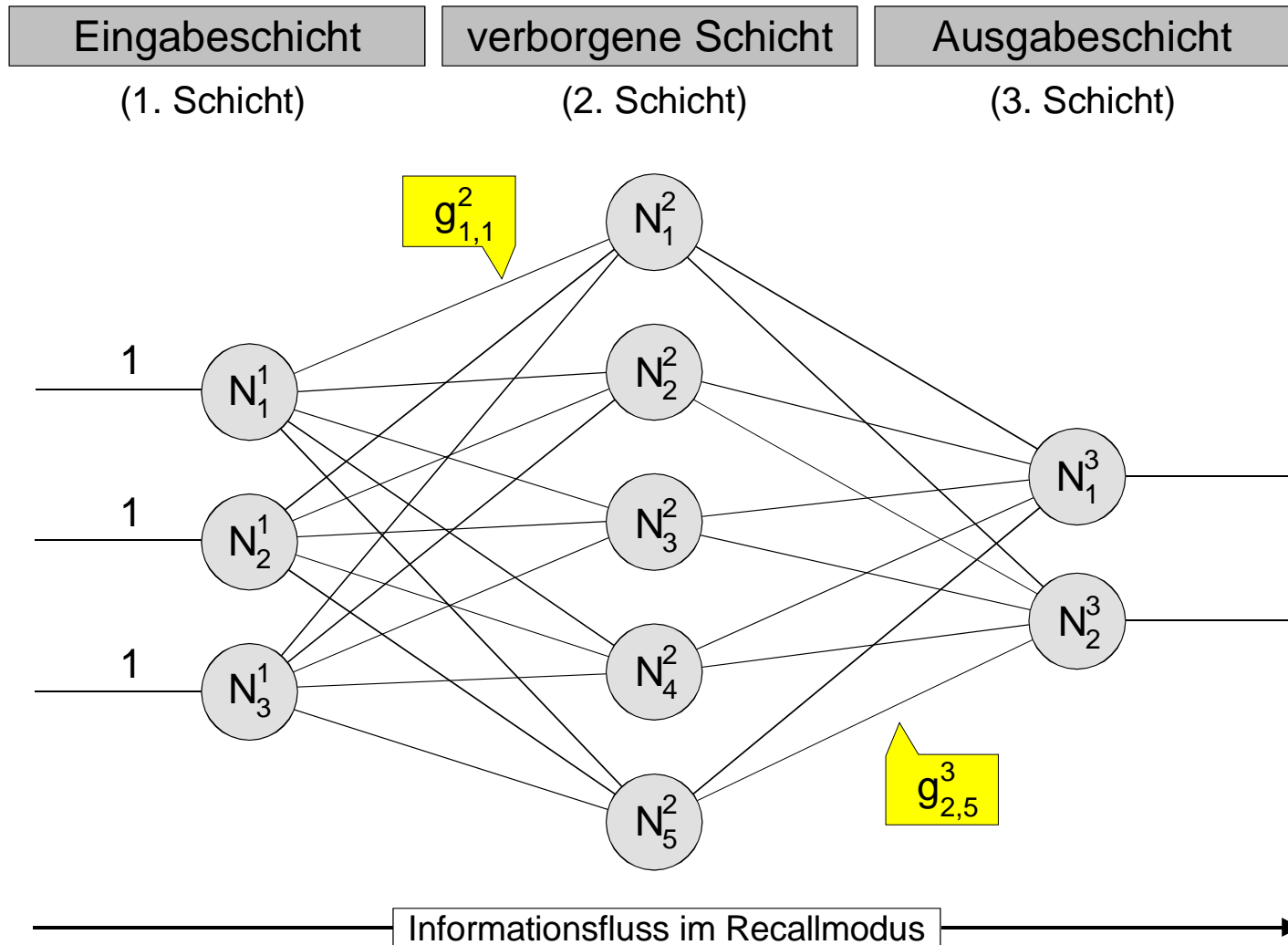


Matrix

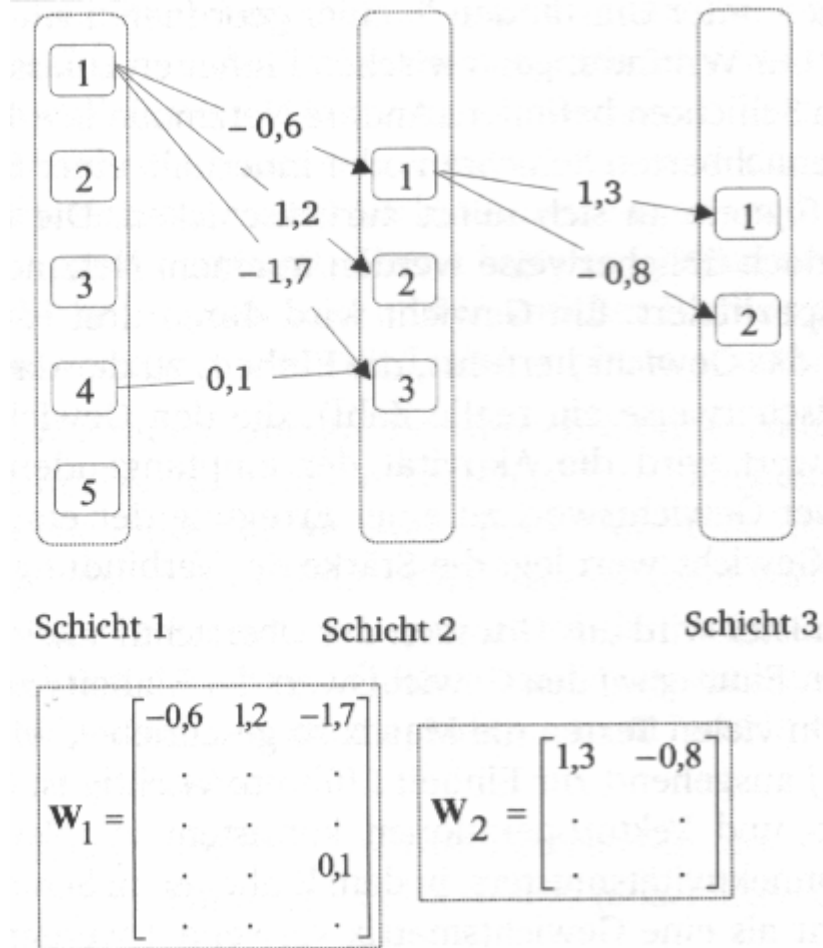


Hopfield-Netztyp

Zwei- bzw. dreischichtiges Feedforward-Netz



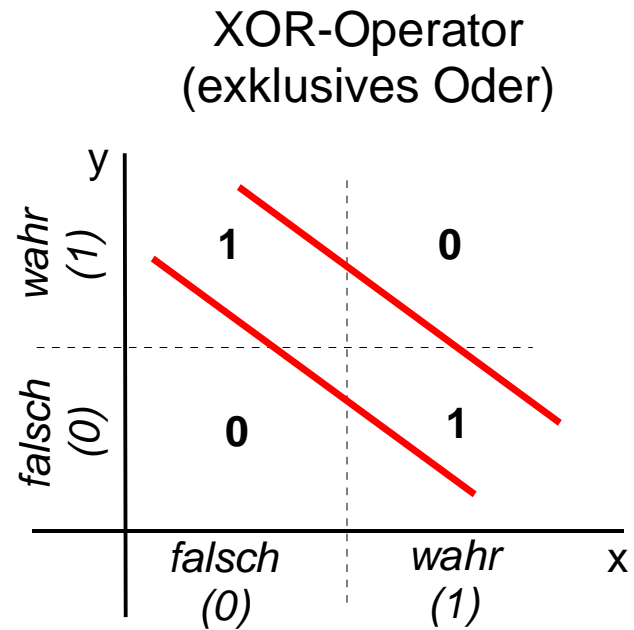
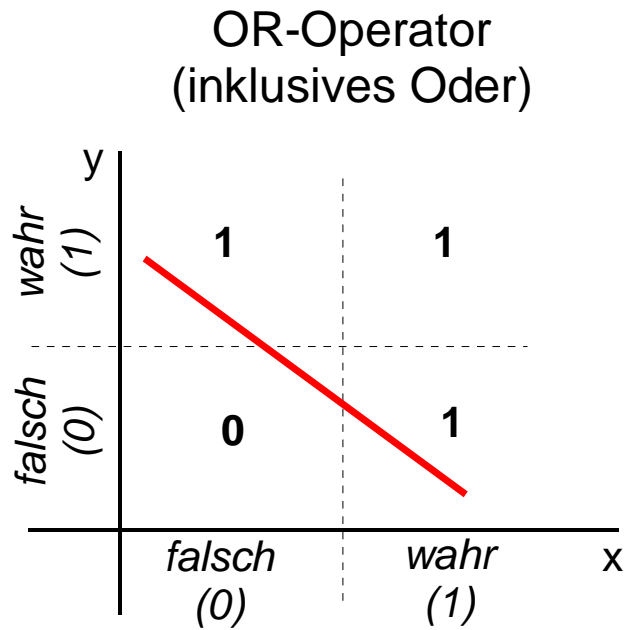
Konnektivitätsmuster in Matrixnotation



Demo 1

Problemkomplexität und Netzdesign

- Komplexe Problemstellung (nicht linear-trennbar)
 - Bedeutung der Nichtlinearität der Transferfunktion
 - Bedeutung der verdeckten Schicht
- OR- und XOR-Operator im Vergleich



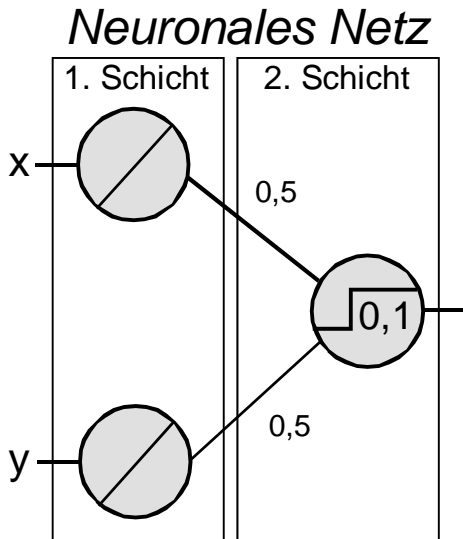
Demo 1

Lösungsbaustein: Perceptrons

- Erfinder: Frank Rosenblatt (Anfang der 1960er)
- binäre Transferfunktion mit Schwellwert θ
 - falls $net_{in} < \theta$, dann $out = 0$
 - sonst $out = 1$
- führt lineare Trennung des Eingaberaumes durch

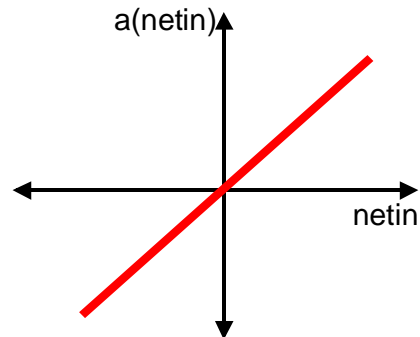
Demo 1

OR-Operator

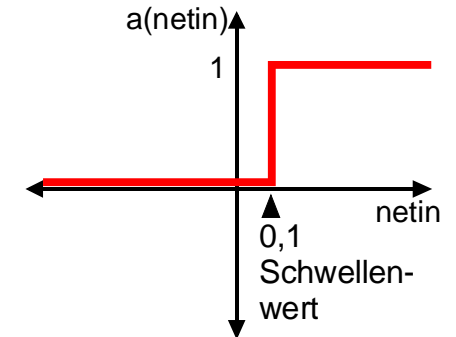


Transferfunktionen

1. Schicht

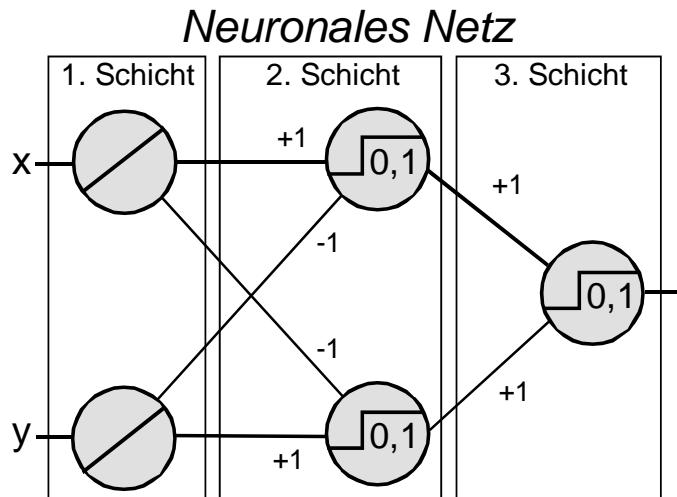


2. Schicht



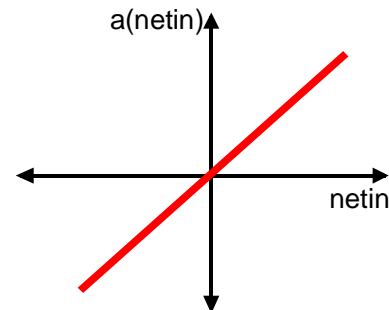
Demo 1

XOR-Operator

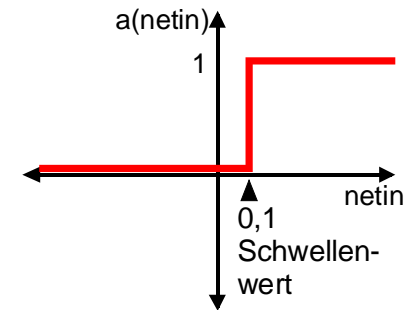


Transferfunktionen

1. Schicht



2. und 3. Schicht

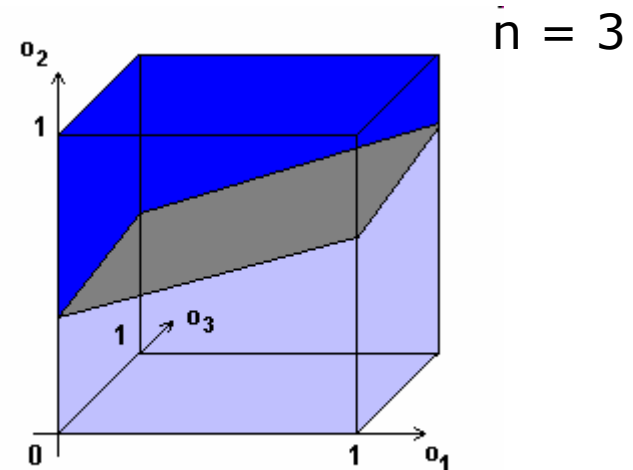
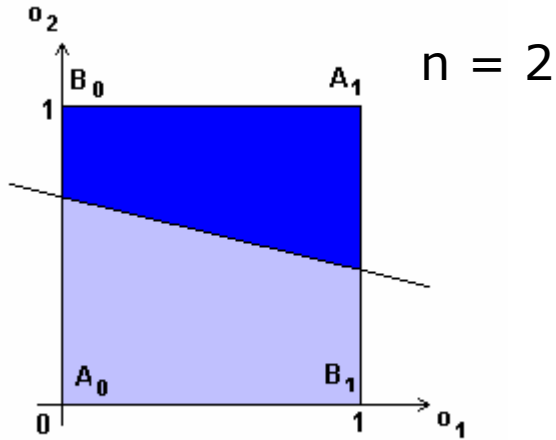


- ohne verdeckte Schicht geht es nicht
- Transferfunktion in der verdeckten Schicht muss nicht-linear sein (sonst äquivalent zu zweischichtigem Netz)

Demo 1

Grafische Interpretation von Perceptrons 1

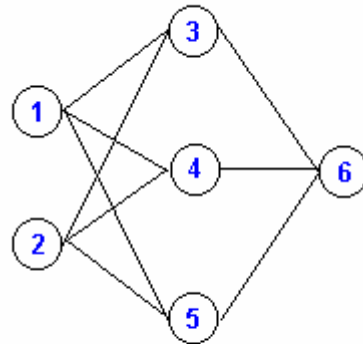
- einstufige (= zweischichtige) Perceptron-Netze



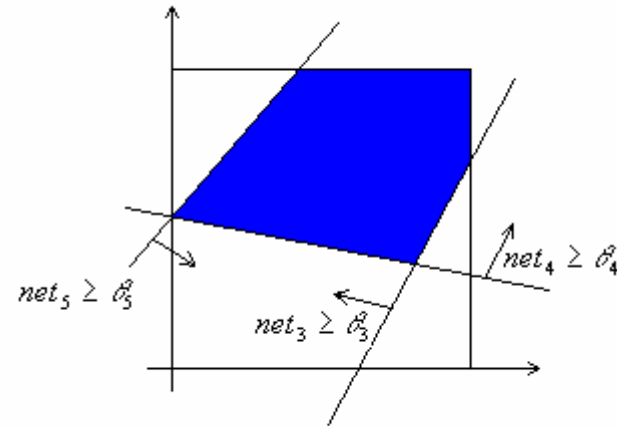
Demo 1

Grafische Interpretation von Perceptrons 2

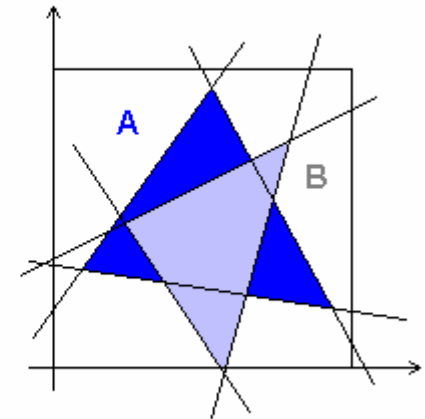
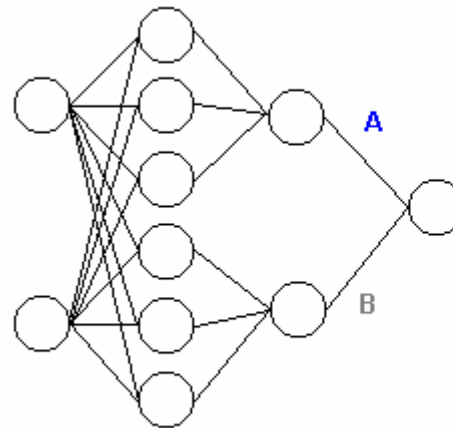
- zwei-stufige (dreischichtige) Perceptron-Netze



Blau: akzeptiertes Gebiet



- drei-stufige (vierschichtige) Perceptron-Netze



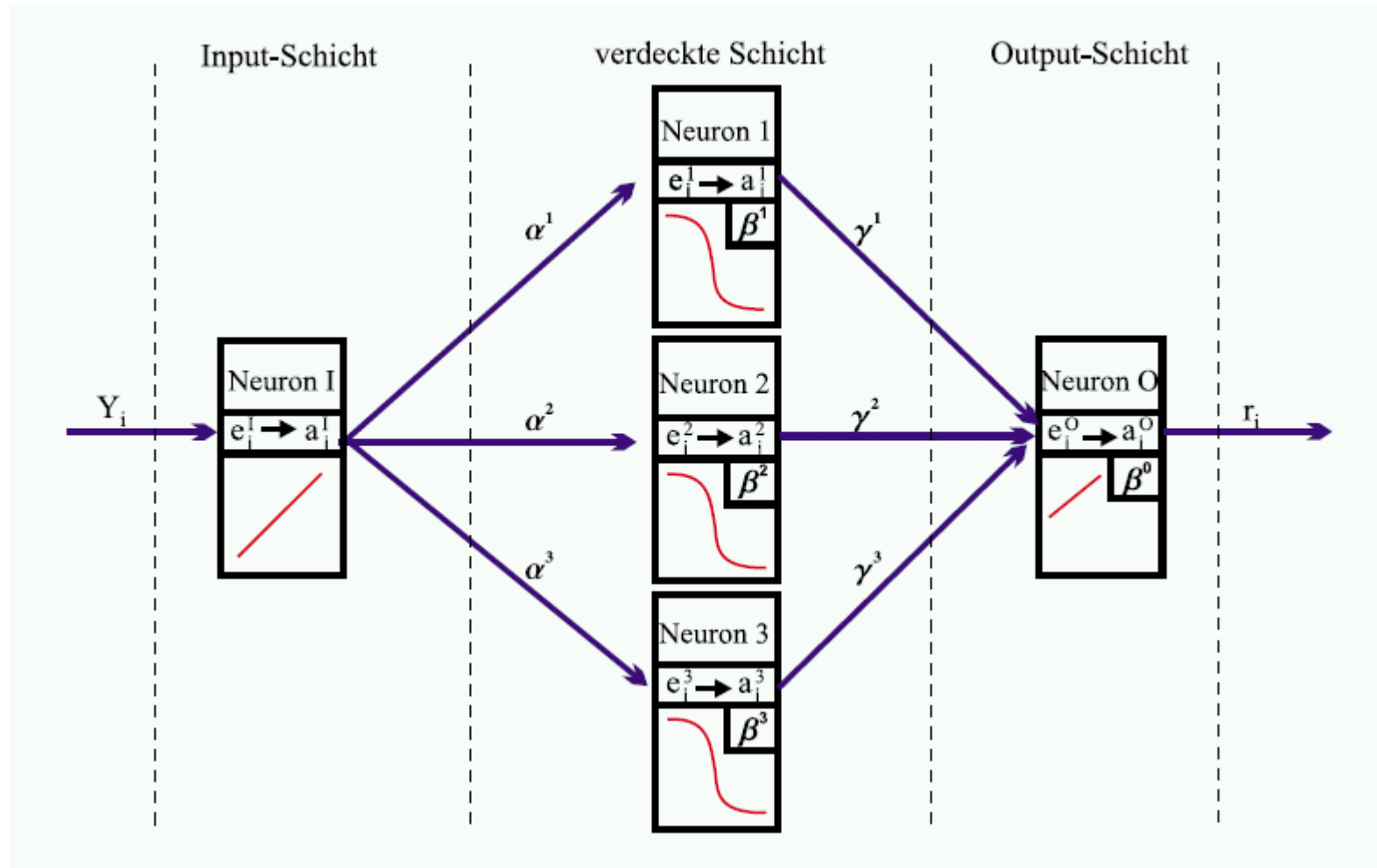
Demo 2

Funktionsapproximation

- Problemformulierung:
 - $f(x) = \sin(x+5)+1$
 - Approximation auf dem Intervall $[0;9]$
- Neuronale Netzlösung
 - Topologie: 1-3-1 Feedforward-Netz
 - Neuronentyp
 - hidden: logistisch
 - sonst: Identität
 - Technischer Trick: Bias-Neuron
 - Schwellwertkodierung (hidden und output layer)
 - Skalierungsfaktor (hidden layer)

Demo 2

Netzmodell (ohne Bias-Neuronen)



Systematisierung Neuronaler Netze (Netztypen) 4

- Neuronentyp
- Topologie
- Lernverfahren
 - überwacht (supervised)
 - korrigierend (mit „Lehrer“)
 - verstärkend (mit „Bewerter“, reinforcement learning)
 - unüberwacht (unsupervised)
 - kooperativ
 - wettbewerblich (winner-take-all)
- Funktion/Aufgabe (Lerngegenstand)

Allgemeine Aspekte neuronaler Lernverfahren

- Recall- vs. Trainingsphase
 - Recall: „gegeben x und gegeben G , finde y “
 - Training: „gegeben x und gegeben y , finde G “
mit:
 - x : Inputvektor
 - y : Outputvektor
 - G : Menge der Netzgewichte
- Generalisierungsfähigkeit vs. Overlearning
 - Auswahl der Trainingsmenge P
 - Rechtzeitiger Abbruch des Trainings
- Trainingsmusterpaar für überwachtetes Lernen
 - Eingabemuster: $x(p)$
 - (Soll-) Ausgabemuster: $y(p)$

} für Trainingsmuster p

Lernzyklus

- Schritt 1
 - Auswahl eines Trainingsmusters $p[x(p),y(p)]$
 - Anlegen des Eingabemusters an die Inputschicht
- Schritt 2
 - Verarbeitung des ausgewählten Trainingsmusters zum Netzoutput out
- Schritt 3
 - Soll-Ist-Vergleich: Netzoutput vs. Ausgabemuster des Trainingsbeispiels ($out(p)$ vs. $y(p)$)
 - Ermittlung des Fehlermaßes $E(p)$
- Schritt 4
 - Berechnung der durch das Fehlermaß des Trainingsbeispiels induzierten Änderung der Netzgewichte $\Delta g(p)$ (Anwendung der Lernregel)

Trainingsepochen und -modus

- Trainingsepoche
 - Vollständige Präsentation aller Trainingsbeispiele aus der Trainingsmenge P
- Trainingsmodus
 - Batch-Modus (epochales Lernen)
 - Kumulation der induzierten Gewichtsänderungen über die gesamte Epoche
 - Änderung der tatsächlichen Netzgewichte am Ende einer Epoche
 - Online-Modus (musterweises Lernen)
 - Änderung der tatsächlichen Netzgewichte nach jedem Trainingsbeispiel

Lernziel

- Minimierung des Netzfehlers

$$E = 0,5 \cdot \sum [\text{out}(p) - y(p)]^2$$

- unter Beibehaltung der Generalisierungsfähigkeit

Lernphase

- Initialisierung der Netzgewichte
- Epochendurchlauf bis ...
- ... Abbruchkriterium erfüllt
 - vorgegebene Epochenanzahl
 - Unterschreiten der Güteschwelle
 - Anstieg des Testmengenfehlers
- evtl. Pruning der Netzgewichte mit Validierungsdaten
- Performancetest des Netzes mit Generalisierungsdaten

Musterkategorien

- Modellierungsdaten
 - Trainingsdaten
 - Testmengen- bzw. Validierungsdaten
- Generalisierungsdaten

Hebbsche Lernregel: Einfache Form

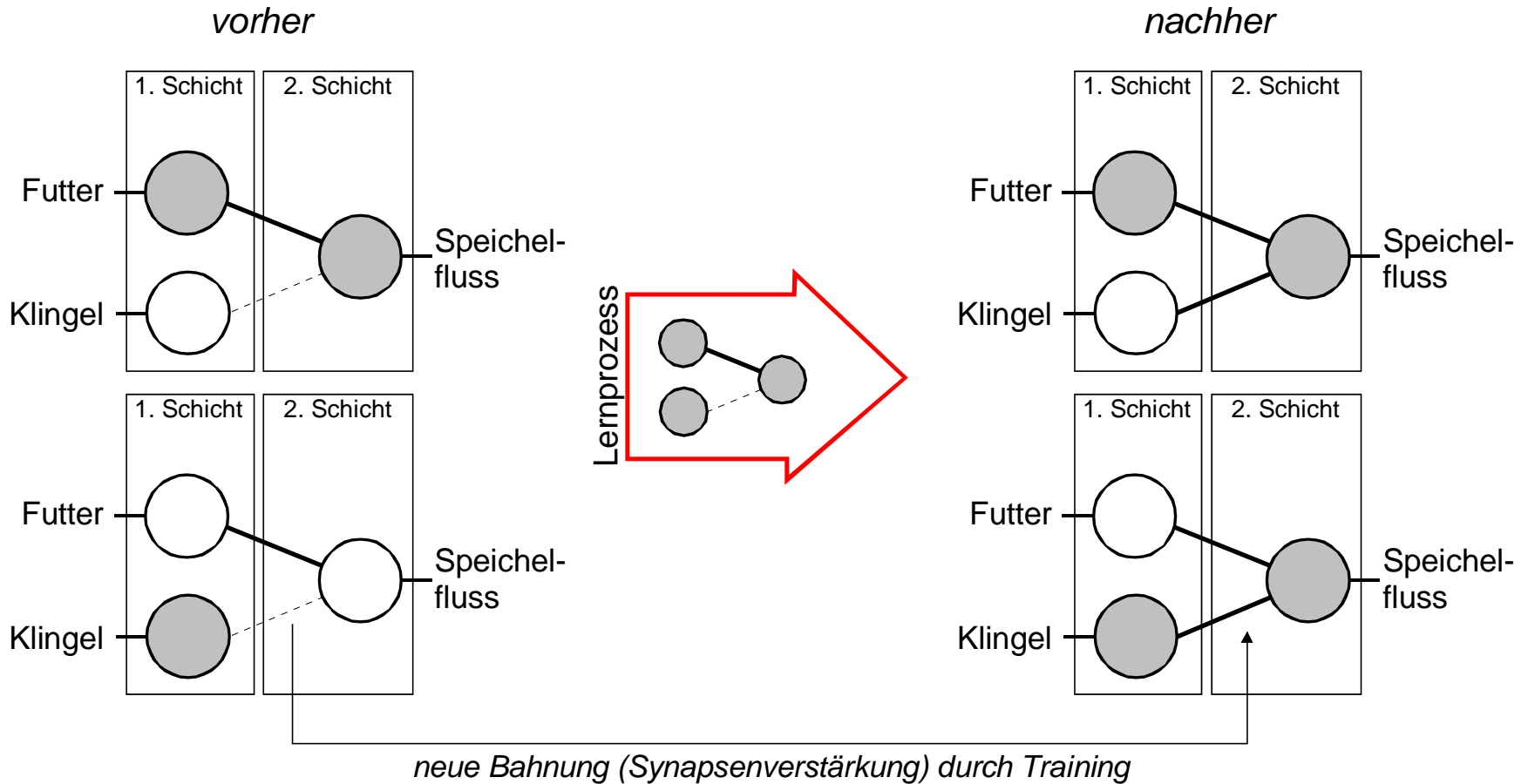
- D. O. Hebb (1949)
- Wenn zwei Neuronen gleichzeitig aktiviert sind, so verstärkt sich die zwischen ihnen geschaltete Verbindung, andernfalls baut sie sich ab.
- einfache Form

$$\Delta g_{j,i}^s = \eta \cdot \text{out}_j^s \cdot \text{out}_i^{s-1}$$

mit: μ = Lernrate

- Einfluss der Lernrate
 - niedrig → langsames Lernen
 - hoch → Gefahr des Vergessens

Hebbsche Lernregel und Klassische Konditionierung (Pawlowsche Hunde)



Hebbsche Lernregel: Allgemeine Form

- Problem der einfachen Hebb-Regel
 - Gewichte wachsen im laufenden Lernprozess immer weiter (keine Stabilisierung)
- Lösung (allgemeine Form der Hebbschen Lernregel)
 - Berücksichtigung eines Zielwertes (target) für das Neuron, zu dem das jeweils betrachtete Gewicht gehört
 - Berücksichtigung des bisherigen Niveaus des Gewichtes

$$\Delta g_{j,i}^s = \eta \cdot \varphi(\text{out}_j^s, \text{tar}_j^s) \cdot \psi(\text{out}_i^{s-1}, g_{j,i}^s)$$

- Spezielle Lernregeln
 - Spezifizierung der Funktionen φ und ψ (sprich: Phi und Psi)

Delta-Regel (Widrow-Hoff-Regel)

- Widrow/Hoff (1960)
- Abhängigkeit der Gewichtsänderung vom Fehlersignal des Neurons, zu dem das Gewicht gehört
- Spezifikationen für φ und ψ

$$\varphi(\text{out}_j^s, \text{tar}_j^s) = \text{tar}_j^s - \text{out}_j^s = \delta_j^s$$

$$\psi(\text{out}_i^{s-1}, g_{j,i}^s) = \text{out}_i^{s-1}$$

- Delta-Regel

$$\Delta g_{j,i}^s = \eta \cdot (\text{tar}_j^s - \text{out}_j^s) \cdot \text{out}_i^{s-1} = \eta \cdot \delta_j^s \cdot \text{out}_i^{s-1}$$

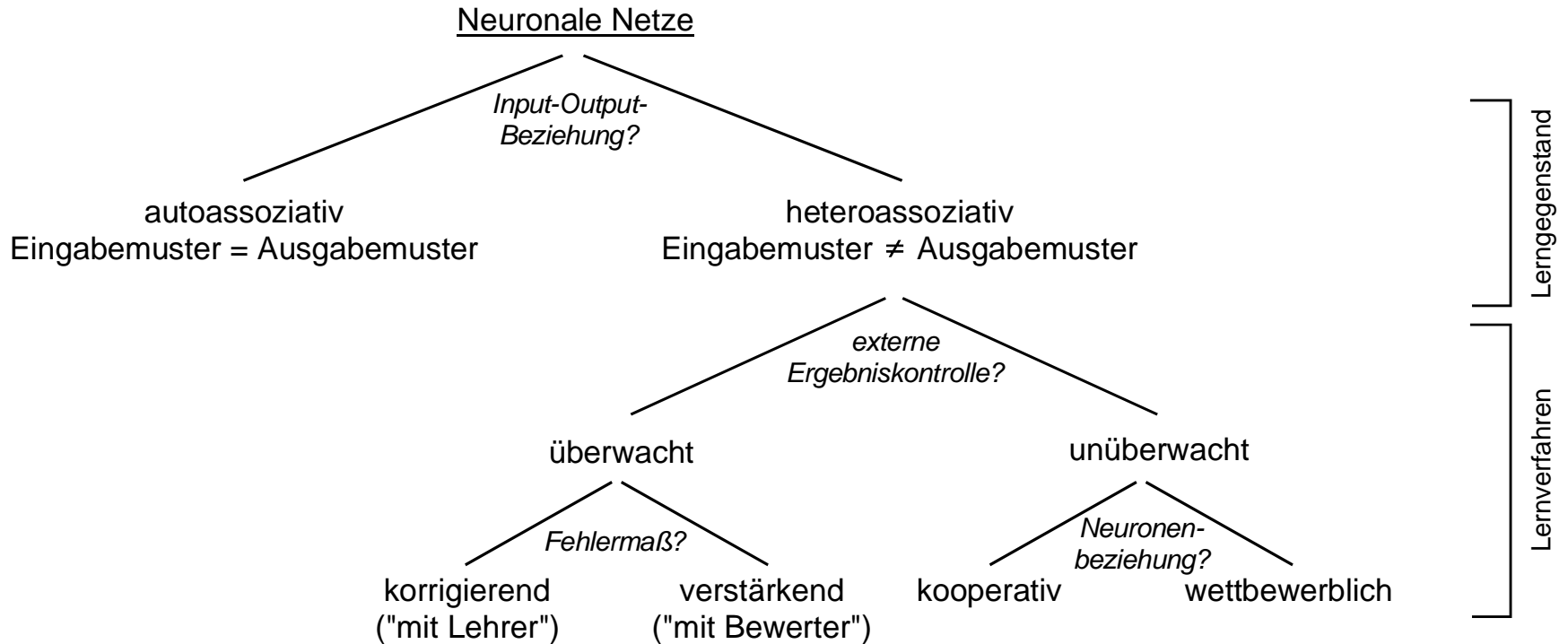
Generalisierte Delta-Regel (Error-Backpropagation-Algorithmus)

- Problem der Delta-Regel
 - Wie kommt man zu einem Fehlersignal für die verdeckten Neuronen?
- Lösung: Generalisierte Delta-Regel!
(s. nächster Gp.)

Systematisierung Neuronaler Netze (Netztypen) 5

- Neuronentyp
- Topologie
- Lernverfahren
- Funktion/Aufgabe (Lerngegenstand)
 - heteroassoziativ
 - autoassoziativ

Lerngegenstand und Lernverfahren



Vereinfachungen gegenüber dem biologischen Vorbild

- Wesentlich geringere Zahl von Neuronen
- Viel geringere Zahl von Verbindungen
- Nur ein Parameter für die Stärke der synaptischen Kopplung (Vernachlässigung zeitlicher Phänomene und des Einflusses verschiedener Neurotransmitter)
- Keine genauere Modellierung der zeitlichen Vorgänge der Nervenleitung
- Konzentration auf homogene Netzwerke
- Keine Berücksichtigung chemischer Einwirkungen benachbarter Neuronen
- Biologisch unplausible Lernregeln
- ...