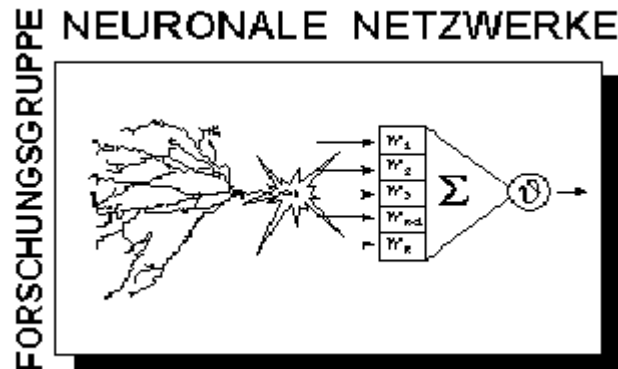


# Vorlesung Neuronale Netzwerke



Heinrich Werner  
Universität Kassel

## Inhaltsverzeichnis

- |   |   |
|---|---|
| 1. <a href="#">Titelseite</a>                         | 17. <a href="#">Funktion der Nervenzellen</a> |
| 2. <a href="#">Einleitung</a>                         | 18. <a href="#">Neuronenmodell</a>            |
| 3. <a href="#">Das biologische Vorbild</a>            | 19. -   |
| 4. <a href="#">40'er Jahre: McCulloch &amp; Pitts</a> | 20. -   |
| 5. <a href="#">50'er Jahre: Rosenblatt</a>            | 21. -   |
| 6. <a href="#">Perceptron Lernregel</a>               | 22. -   |
| 7. <a href="#">60'er Jahre: Minsky &amp; Papert</a>   | 23. -   |
| 8. <a href="#">80'er Jahre: Rumelhart</a>             | 24. -   |
| 9. <a href="#">80'er Jahre: Hopfield</a>              | 25. -   |
| 10. <a href="#">Rückkopplung und Hebb'sche Regel</a>  | 26. -   |
| 11. <a href="#">Optische Mustererkennung</a>          | 27. -   |
| 12. <a href="#">Das Spiralenproblem</a>               | 28. -   |
| 13. <a href="#">Lösung des Spiralenproblems</a>       | 29. -   |
| 14. <a href="#">Computervs. Gehirn</a>                | 30. -   |
| 15. <a href="#">Komponenten eines KNN</a>             | 31. -   |
| 16. <a href="#">Beispiele dazu</a>                    | 32. <a href="#">Inhaltsverzeichnis</a>        |

# Was sind künstliche neuronale Netzwerke?

**Neuronale Netzwerke** (NN, oft auch **künstliche Neuronale Netzwerke** **KNN** bezeichnet) sind Computerprogramme, die die Arbeitsweisen natürlicher Neuronen nachahmen.

Zweck dieser Programme können sein

- Modellierung tatsächlicher neuronaler Vorgänge in der Biologie
- Einsatz der Möglichkeiten neuronaler Systeme in Computerprogrammen



Die Arbeitsweise neuronaler Netzwerke ist bestimmt durch

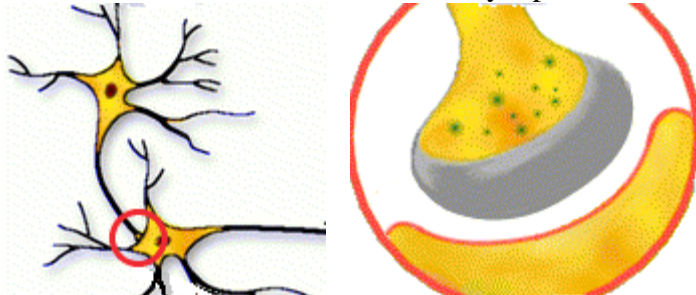
- massiv parallele Informationsverarbeitung in den **Neuronen**
- Propagierung der Informationen über die **Synapsen** (Neuronenverbindungen)
- Vorbereitung der Netze für ihre **Arbeitsphase** (**Propagierung**) durch die **Aufbauphase** (**Netztopologie**) und eine **Trainingsphase** (kein Programmieren!)

## Das biologische Vorbild

Biologisches Vorbild der künstlichen neuronalen Netze sind die Neuronengeflechte, wie sie in menschlichen und tierischen Nervenbahnen und Gehirnen vorkommen.



Ein menschliches Gehirn besteht aus ca. 100 Billionen Neuronen, die mit ihren Axonen und Dendriten über Synapsen zusammengeschaltet sind.

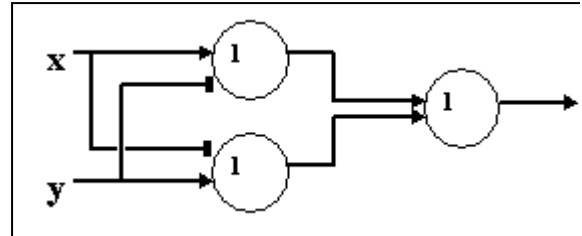


# historische Vorbemerkungen

## McCulloch-Pitts(1940)

Grundsätzliche Überlegungen über Fragen der Berechenbarkeit sowie über Berechnungsvorgänge im Gehirn, in Maschinen und in Algorithmen.

(W. McCulloch, W. Pitts: A logical calculus of the ideas immanent in nervous activity. Bull. of Math. Biophysics 5, 1943, 115-133)



- ↑ Diese einfache Modell sieht die Neuronen als Input-abhängige 0-1-Signalgeber, die interne
- ↓ i Schwellwerte haben. Wenn der Gesamtinput diesen Schwellwert erreicht oder überschreitet, gibt das Neuron 1 aus (es **feuert**) und sonst 0. Der Input kommt über Leitungen, die **excitatorisch** (anregend) oder **inhibitorisch** (hindernd) sein können, jedes Neuron ermittelt nun seinen Gesamtinput als die Summe aller Signale auf den excitatorischen Leitungen vermindert um die Summe aller Signale auf den inhibitorischen Leitungen zu dem Neuron.

Erstes Ergebnis dieser Untersuchungen war, daß die durch solche neuronalen Netze (am grünen Tisch durch Programme) realisierbaren Funktionen genau die im mathematischen Sinne berechenbaren Funktionen sind. Dieser Netztyp realisiert aber noch keine Anpassung (Training/Lernen) an ein gegebenes Problem.

# Das Perzeptron

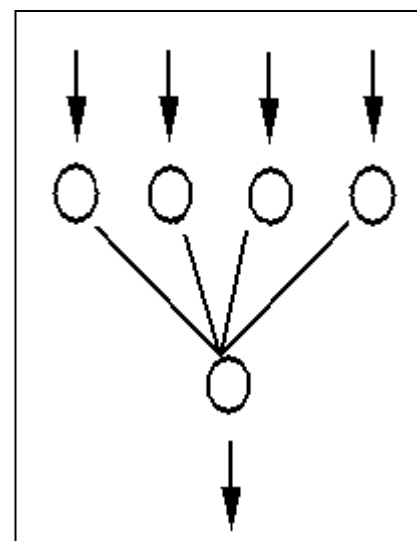
## Rosenblatt(1957)

Es mußte ein Konzept geschaffen werden, das die Anpassung eines bestehenden Netzwerks an ein bestimmtes Problem hinerlaubt.

Grundidee war hier, die Leitungen mit "Gewichten" zu versehen, mit denen sie die durch sie geschickten Signale modifizieren. Das

- ↑ Neuron summiert diese modifizierten Signale zum Gesamtinput auf.
- ↓ i Eine Veränderung eines gegebenen Netzwerks besteht nun in der Veränderung der Gewichte.
- ↓ Daserste Modell für einen solchen Netzwerktyp war das **Perzeptron**, das aus einer Schicht von Input-Neuronen besteht, die alle mit einem Output-Neuron (bzw. einer Schicht von Neuronen) verbunden sind.

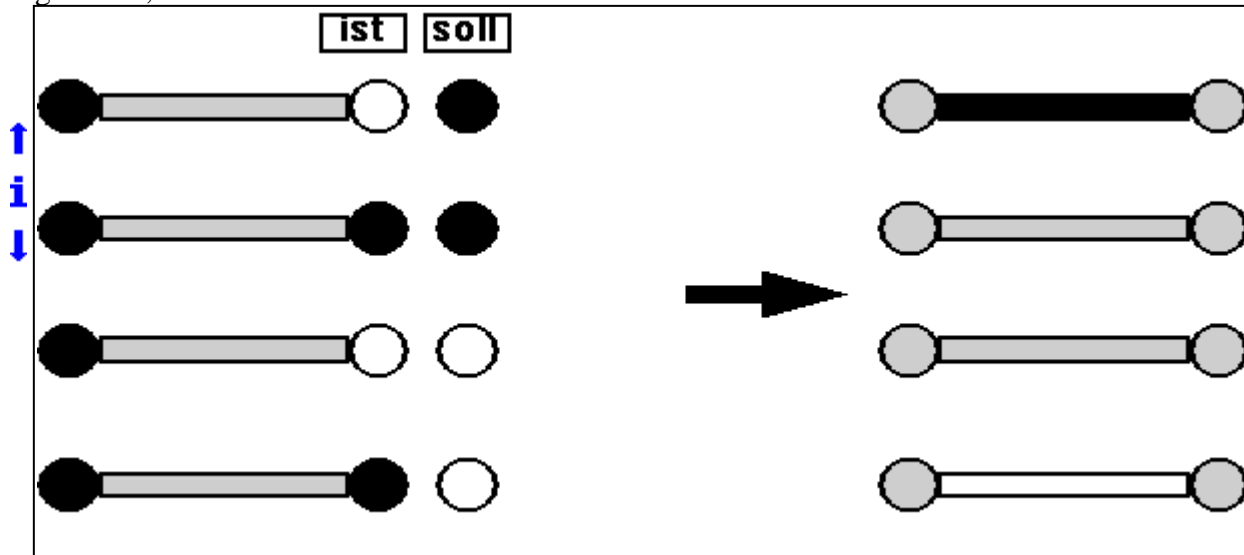
(F. Rosenblatt: The Perceptron: a perceiving and recognizing automaton. Cornell Aeronautical Laboratory, 1957)



# Die Perzeptron Lernregel

Einzugehöriger Algorithmus zur Anpassung der Gewichte (anmaßend: ein Lernalgorithmus) an ein gefordertes Input-Output-Verhalten konnte formuliert werden und stellt den Prototyp eines überwachten Lernvorgangs ("Lernen aus Fehlern"), dar.

Verhält sich ein Neuron nicht, wie es soll, so wird das Gewicht der Verbindung zum Neuron so geändert, daß dieser Fehler nicht mehr auftritt.



## Beschränkungen des Perzeptrons

### Minski & Papert (1969)

Minski und Papert führten eine rigorose Analyse des Perzeptrons durch und zeigten seine Möglichkeiten und Grenzen auf.

#### Delta-Regel

Synapsenverstärkung bei zu geringer Aktivität, bzw. Synapsenverringern bei zu hoher Aktivität.

#### Perzeptron-Konvergenz-Satz

Der Satz garantiert, daß ein Perzeptron eine Funktion mit diesem

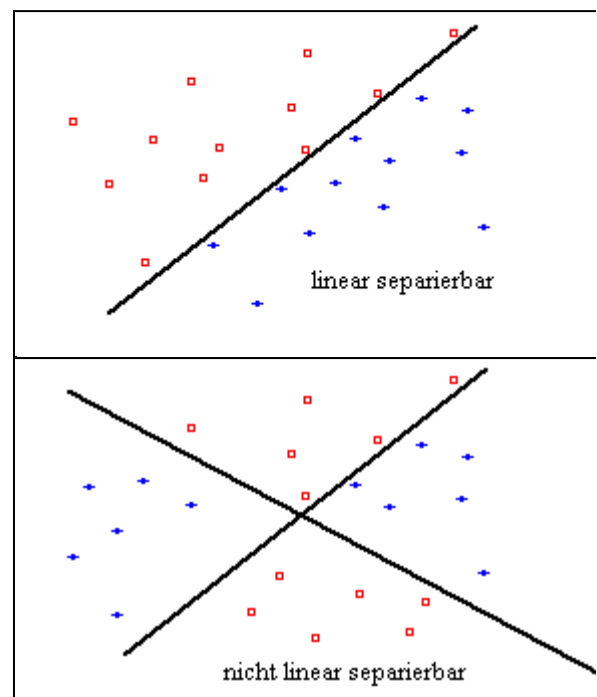
Verfahren in endlich vielen Schritten lernen kann, sofern es sie überhaupt lernen kann.

#### Perzeptron-erlernbare-Funktionen

Nur die linear separablen Funktionen sind mit einem Perzeptron erlernbar und auch die Anzahl der Schritte ist a priori nicht abschätzbar.

Dies war eine herbe Ernüchterung und ein gewaltiger Rückschlag, da es für diese Funktionen schon damals viel bessere und mathematisch einfachere Darstellungsgab.

Das führte zu einem fast 20-jährigen Dornröschenschlaf der NN (M. Minski, S. A. Papert: Perceptrons. Cambridge, Mass. MIT Press 1969)

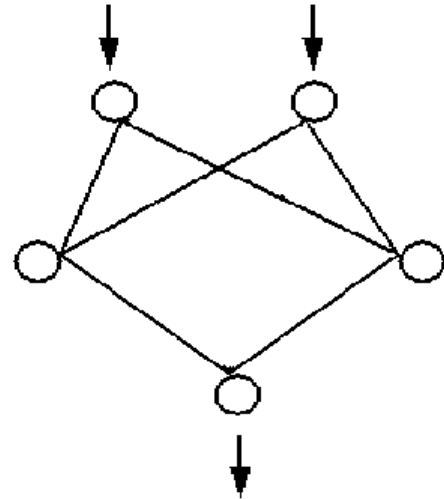


# Der Backpropagation Algorithmus

## Rumelhart, Hinton (1986)

Die alten Ideen um das Perceptron wurden zu mehrschichtigen Modellen weiterentwickelt und von Rumelhart und anderen dazu der Backpropagation Algorithmus als seine Erweiterung der Delta-Regel formuliert. Bei diesem Verfahren wird der Fehler an der Ausgangsschicht ermittelt, von Schicht zu Schicht zurück berechnet und danach die Änderung der Synapsengewichte vorgenommen.

Durch die Mehrschichtigkeit wird auch die Einschränkung der darstellbaren Funktionen auf linear separierbare Modelle aufgehoben. (D.E. Rumelhart, G. Hinton, R.J. Williams: Learning representations by backpropagating errors. Nature 323, 1986, 533-536)



# Netzwerk mit Rückkopplung

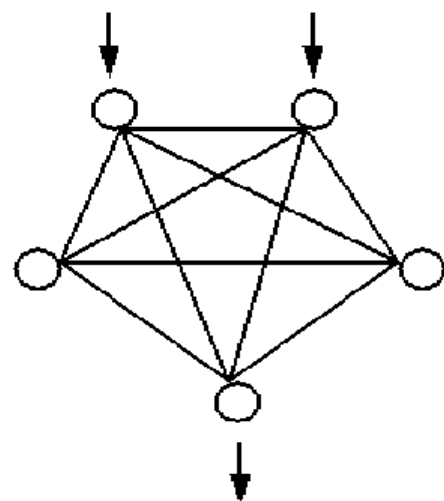
## Hopfield, Tank (1985)

In den 80'er Jahren gab es einen weiteren Durchbruch für Neuronale Netzwerke, bei dem man sich statt der alten Perceptron-Idee auf rückgekoppelte Netzwerke mit selbstorganisierten Lernalgorithmen, d.h. "Lernen durch Gewöhnung", konzentrierte.

Als wichtige Vertreter sind hier T. Kohonen und J. Hopfield zu nennen, deren Ideen zum Lernalgorithmus auf Erkenntnissen von Hebb basieren.

Das Problem, das durch die Rückkopplung entsteht, ist, daß das System sich selbst anregt und deshalb möglicherweise nicht zur Ruhe kommt. Deshalb müssen weitere Bedingungen dafür sorgen, daß das System in einen definierten Endzustand einschwingt (**relaxation**).

(J. Hopfield, D. Tank: Neural Computation of decision in optimization problems. Cybernetics 52, 1985, 141-152)

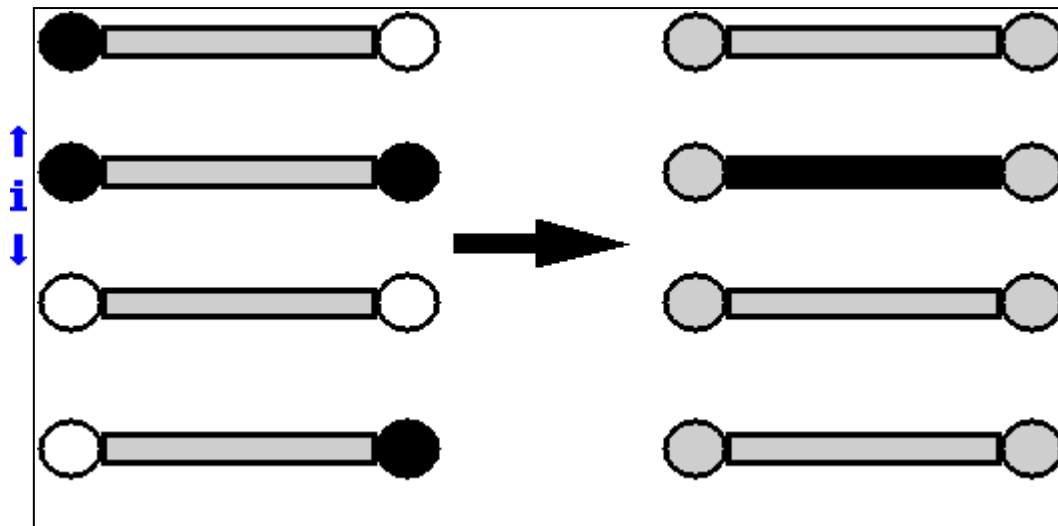


# Die Hebb'sche Lernregel

Hebb(1949)

## Hebb'sche Regel

Bei starker gleichzeitiger Aktivität der Neuronen wird deren Verbindungsgewicht erhöht.

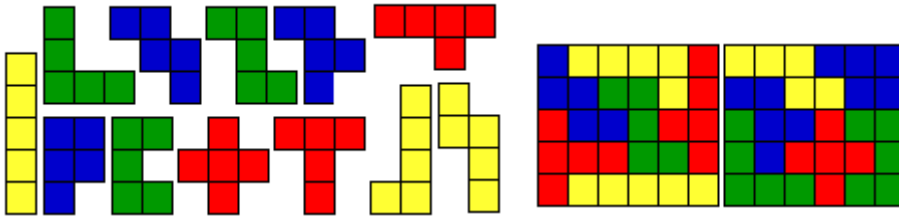


(D.Hebb: The organization of behavior. John Wiley, New York 1949 )

## optische Mustererkennung

Natürliche Neuronale Netzwerke zeigen erstaunliche Fähigkeiten im Bereich der Mustererkennung, ein Gebiet, das klassischen Rechnern erhebliche Probleme bereitet.

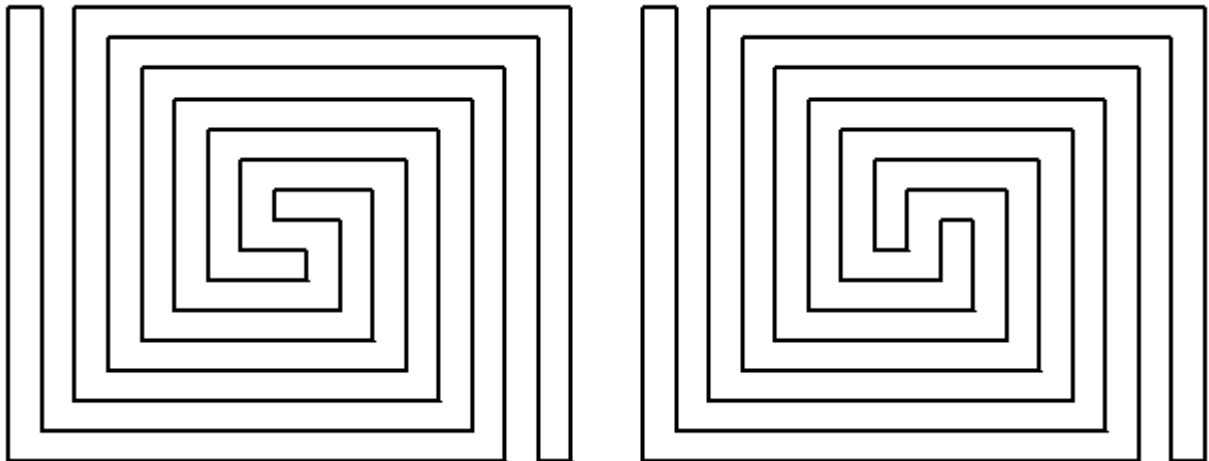
Betrachtet man z.B. das Parkettierungsproblem, mit den 12 Pentominos (zusammenhängende Spielsteine aus 5 Quadraten) eine rechteckige Fläche von  $5 \times 12$  bzw.  $6 \times 10$  Quadraten zu überdecken, so sieht man eine deutliche Überlegenheit des menschlichen Gehirns gegenüber einem herkömmlichen Computerprogramm



Als Übung suche eine Überdeckung eines  $3 \times 20$  Rechtecks.

## Das Problem der zwei Spiralen

Aber auch bei ganz eindeutigen Mustererkennungsproblemen können natürliche Neuronale Netze ihre Probleme haben, wie z.B. beim Problem der zwei Spiralen:

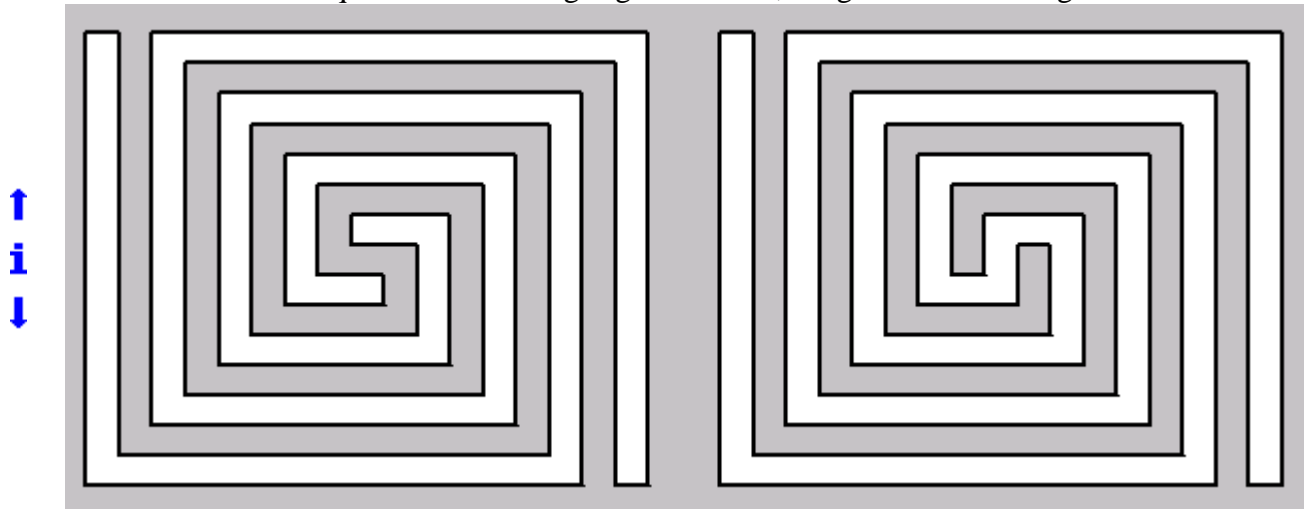


**Welches der beiden Bilder zeigt zwei Spiralen?**

Die Antwort bereitet oft einen sequentiellen Algorithmus (z.B. Einfärben) größte Schwierigkeiten und da auch ein gutes Auge kann keine sichere Antwort vermitteln.

## Lösung des Spiralenproblems

Wendet man einen sequentiellen Färbungsalgorithmus an, so ergibt sich die Lösung sehr leicht:



Lokal sind die beiden ungefärbten Bilder nicht zu unterscheiden

## Computer vs. Gehirn

	Computer	Gehirn
Prozessor	hochspezialisiert	sehr einfach
Geschwindigkeit	10 <sup>6</sup> Instruktionen/sec	10 <sup>4</sup> Instruktionen/sec
Prozessor-Zahl	< 20	> 10 <sup>9</sup>
Verbindungen	< 10 <sup>2</sup> pro Prozessor	> 10 <sup>4</sup> pro Prozessor
Programmierung	durch Anweisungen	durch Training
↑ Arbeitsmodus	seriell	parallel
i Selbstorganisation	schwierig	einfach
↓ Adaptionfähigkeit	mäßig	gut
Berechnungen	sehr genau/sehr schnell	ungenau/langsam
Speicherung:		
Objekte	einfach	komplex
Retrieval	exakt	ungenau
Robustheit bei:		
Änderungen	schlecht	gut
Zerstörung	sehr schlecht	gut

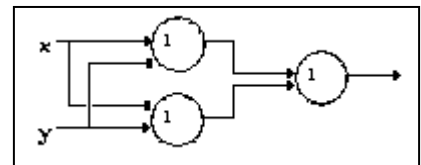
## Elemente eines Neuronalen Netzwerks



Zu einem Neuronalen Netzwerk gehören folgende Details:

1. Eine Menge von **Neuronen** (die elementaren Prozessoren)
2. Das Netz der **Verbindungen (Synapsen)** zwischen den Neuronen
3. Die **Gewichte** der Verbindungen
4. Die **Propagierungsregel** für das Weiterleiten eines Inputs
5. Die Aktivations-**Zustände** der Neuronen
6. Die **Output-Funktionen** und **Schwellwerte** der Neuronen
7. Die **Aktivierungsregel**, nach der alle Inputs eines Neurons mit dessen momentanen Zustand zu einem neuen Zustand des Neurons kombiniert werden
8. Die **Lernregel**, nach der die Gewichte der Verbindungen auf der Basis von Trainingseingaben verändert und angepaßt werden können
9. Die **Umgebung**, in der das lernende System arbeiten soll

## Beispiel (McCulloch-Pitts)



1. 3 Neuronen (durch die Kreise in der Zeichnung symbolisiert).
2. 7 Verbindungen, je zwei von jedem Input in die Neuronen der ersten Schicht, 2 Verbindungen von den Neuronen der ersten Schicht in das Output-Neuron und schließlich die Verbindung vom Output-Neuron nach außen.
3. Gewichte: 1 und -1 (in der Graphik durch --> und --| gekennzeichnet).
4. Propagierungsregel: Leitung des Output der Neuronen über die Verbindungen. In rückgekoppelten Netzen kommt dazu, wann das System in einen stabilen Zustand eingeschungen ist.
5. Die Aktivations-Zustände der Neuronen ergeben sich als Summe  $x_1 w_1 + \dots + x_n w_n$  aus den Produkten Neuronen-Inputs  $x_i$  mal Synapsengewichten  $w_i$ .
6. Output-Funktion ist die Sprungfunktion, die den Wert 0 ausgibt, wenn der Zustand  $z$  unterhalb des Schwellwerts  $s$  (d.h. die Zahl im Kreis) bleibt, und sonst 1 **feuert**:  $out = (z < s) ? 0 : 1$ .
7. Aktivierungsregel: Skalarprodukt-Regel, d.h. der Zustand  $z = x_1 w_1 + \dots + x_n w_n$ , der Aktivations-Zustand ist also das Skalarprodukt aus dem Input-Vektor mit dem Gewichts-Vektor des Neurons.
8. Das Lernen wird in diesem Beispiel noch nicht behandelt.
9. Umgebung der zweistelligen Booleschen Funktionen, d.h. zwei Booleschen Inputs  $x, y$  (0 oder 1) wird in einen Booleschen Wert als Output zugewiesen (hier  $x \text{ XOR } y$ ).

# Die Funktion der Nervenzellen

Eine biologische Nervenzelle besteht aus dem eigentlichen Zellkörper (**Soma**), den **Dendriten**, das sind kurze Leitungen (um 0.4mm), über die Reize in die Nervenzelle gelangen können, dem **Axon**, das ist die Leitung (mm-Bruchteil bis mehrere cm lang), über die Zellaktivität an andere Zellen weitergeleitet wird, und schließlich den **Synapsen**, das sind die Kontaktstellen zwischen Axonen und Dendriten bzw. Soma. Da Dendriten und Axon stark verzweigen, entstehen die hohen

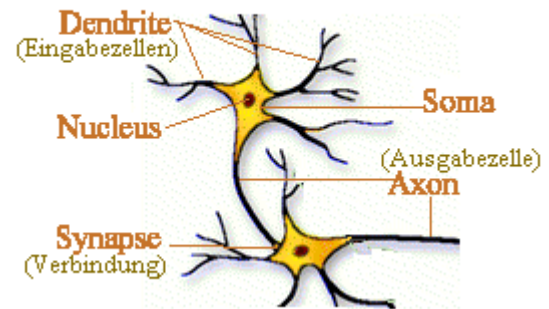


Verbindungszahlen biologischer Nervensysteme.

Wird das Soma über einen gewissen Schwellwert hinaus über die Dendriten angeregt, so erzeugt die Zelle einen kurzzeitigen elektrischen Impuls (**Spike**), der vom Axon an andere Zellen übertragen wird. Da die biologische Information nicht in der Amplitude der Spikes sondern in deren **Frequenz** codiert ist, können diese Spikes ohne weiteren Verstärkungsmechanismus auf alle Verzweigungen des Axons weitergeleitet werden.

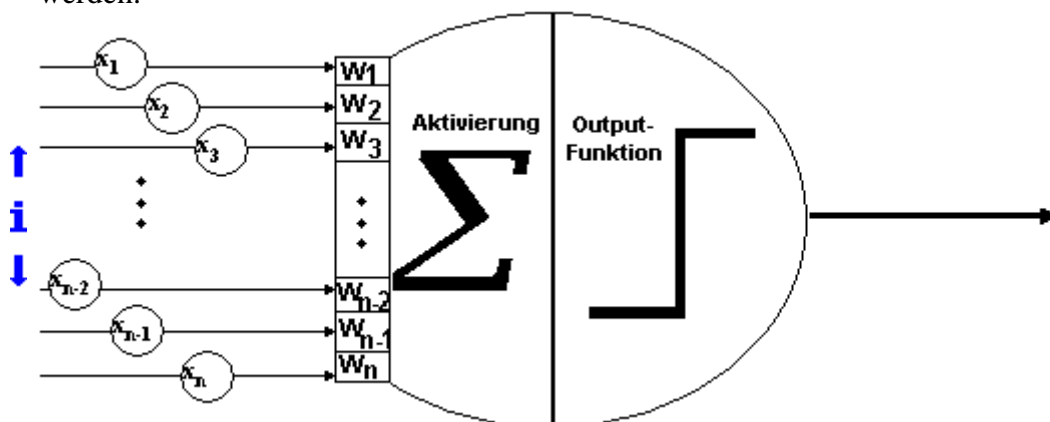
Die Synapsen sind chemische Kontakte, die auf einen Spike mit der Ausschüttung gewisser Ionen (**Neurotransmitter**) reagieren, die dann wiederum Potentialänderungen (und damit Spikes) in den Dendriten erzeugen. Die neue Spikefrequenz ist dann in Abhängigkeit von der Stärke des Kontaktes (**Synapsengewicht**) proportional zur ankommenden Spikefrequenz. Dabei können die Spikes zur Anregung des Somas beitragen (exzitatorische Synapsen) oder im Gegenteil hemmend wirken (inhibitorische Synapsen).

## Schematischer Aufbau des Neurons



## Das künstliche Neuronenmodell

Der Input ist gekennzeichnet durch die **Aktivierungs-Funktion**. Diese legt fest, wie die über die Verbindungen eingehenden Reize  $x_i$  mit den Gewichten  $w_i$  zu einem Gesamtreiz zusammengefasst werden.



Die **Output-Funktion** legt fest, wie der Output aus dem oben berechneten Gesamtreiz ermittelt wird. Der Output wird dann über die Verbindungen als Reiz an andere Neuronen geleitet.