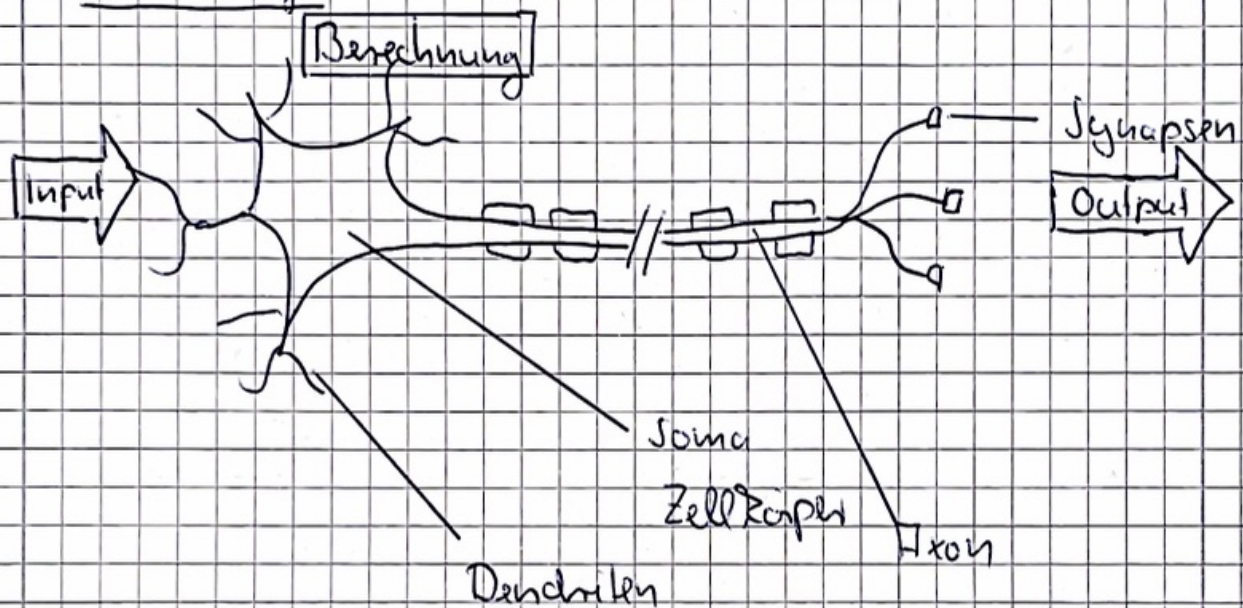


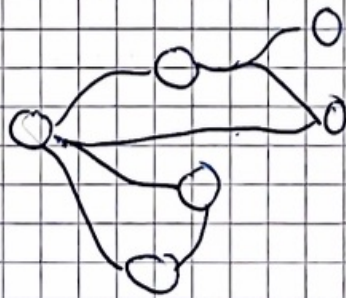
# Das Perceptron

- 1957 in den USA von Frank Rosenblatt
- 1969 bekannt durch Buchpublikation  
der KI-Pioniere Marvin MINSKY und  
Seymour PAPER

Grundlage: menschliches Neuron



Neuronale Verschaltung



=> Netzwerk

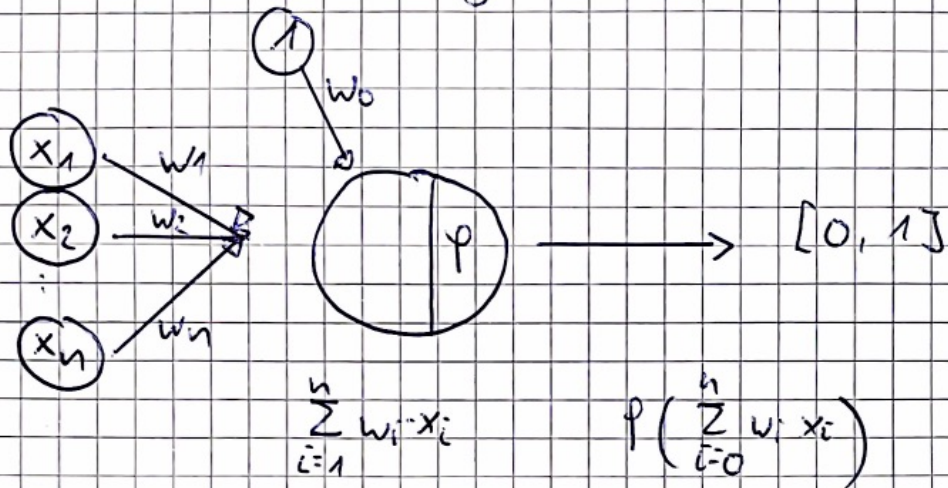


# Künstliches Neuron

Input

Berechnung

Output



$w_1, \dots, w_n$  Gewichte

mit Bias

$$\sum_{i=0}^n w_i \cdot x_i$$

$\phi$  Aktivierungsfunktion

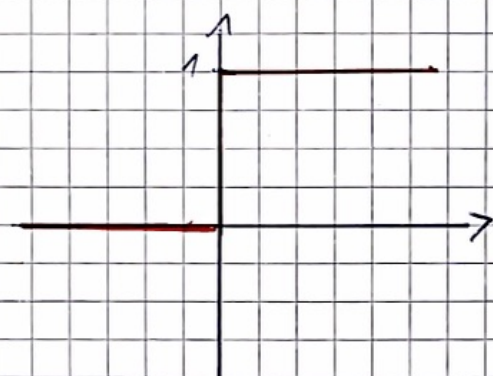
$\sum_{i=0}^n w_i \cdot x_i$  Übertragungsfunktion  
= Rohinput  
= Zwischenberechnung



## Aktivierungsfunktion beim Perzeptron

Heaviside - Funktion

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$



## Lernalgorithmus

1) Initialisiere die Gewichte

z.B. alle  $w_i = 0$

oder kleine zufällige Werte

2) Vorhanden Trainingsset von Datenvektoren

$$\vec{x}_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1n} \end{pmatrix}, \dots, \vec{x}_m = \begin{pmatrix} x_{m1} \\ \vdots \\ x_{mn} \end{pmatrix}$$

mit jeweils zugehörigem (erwartetem)

Ergebnis  $d_i$

3) Perzeptron bestimmt auf Input alle  $w_i$



3) Führe für alle  $i = 1, \dots, m$  aus

- Perceptron bestimmt als Input  $\vec{x}_i$  und berechnet Output  $y_i$

$\Rightarrow d_i$  ist das, was wir gerne gehabt hätten und  $y_i$  ist das, was wir berechnet haben.

- Die Gewichte werden aktualisiert

$$w_j \leftarrow w_j + \alpha \cdot x_{i,j} \cdot \underbrace{(d_i - y_i)}$$

$$d_i = y_i \Rightarrow 0$$

$$d_i \neq y_i \Rightarrow 1$$

Lernfaktor  $\alpha \in (0, 1]$

$\Rightarrow \alpha$  kann selbst festgelegt werden

$\Rightarrow$  Wenn eine Ausgabe richtig war wird nichts geändert.

$\Rightarrow \alpha$  klein  $\rightarrow$  kleine Lernänderung  
 $\rightarrow$  längerer Lernvorgang

Vorteil: man springt nicht über den optimalen Punkt hinweg.  
genaueres Lernen



#### 4) Lern-Loop:

- Gewichte werden nach jedem Trainingswert angepasst.
- anders wäre es auch gegangen (globalen Fehler berechnen)

wieder zu Schritt 3

Berechnung eines Perceptrons später

Ziel eines Perceptrons

a) Klassifizierung

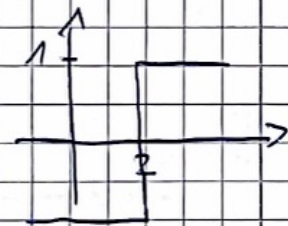
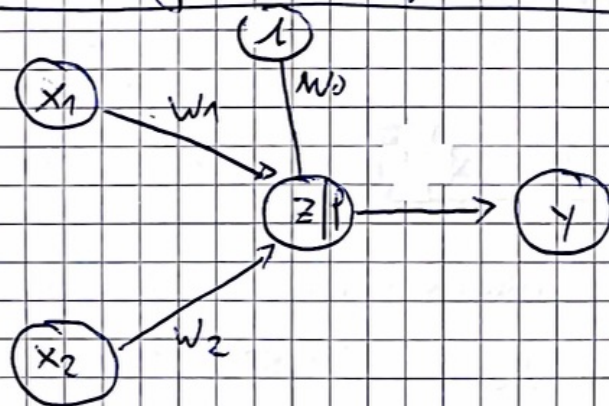
1, 0  $\rightarrow$  Hund und Katze

b) numerische Abhängigkeiten

0, ..., 1  $\rightarrow$  welche lineare Abhängigkeit haben zwei Werte

Bsp:

Modellierung eines logischen UND's



$x_1$	$x_2$	$y = x_1 \text{ \& } x_2$
0	0	0
0	1	0
1	0	0
1	1	1

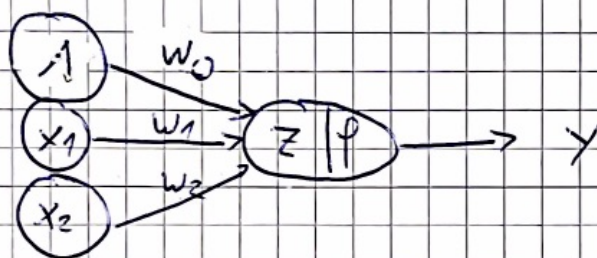


setze  $w_1 = 1$

$w_2 = 1$

$w_0 = -2$

Modellierung eines logischen Oders:



$x_1$	$x_2$	$y = x_1 \vee x_2$
0	0	0
0	1	1
1	0	1
1	1	1

setze  $w_1 = 1$

$w_2 = 1$

Bias  $w_0 = -1$

Modellieren eines XOR

=> geht nicht: Marvin Minsky

Jerry S. Forrester

=> es wird eine weitere Schicht benötigt.