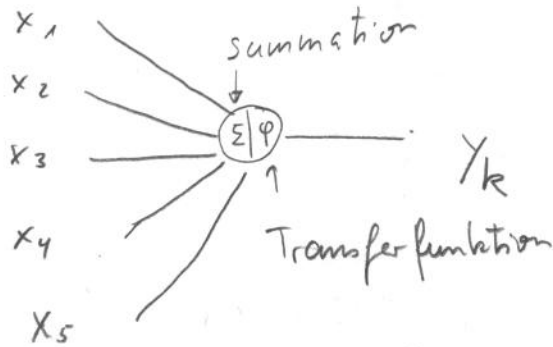


Warren McCulloch und Walter Pitts 1943



$$y_k = \varphi \left( \sum_{j=0}^m w_{kj} x_j \right)$$

$\varphi(\cdot)$  oft die Heaviside Stufenfunktion  $\Theta(\cdot)$

(binäre Signale McCulloch-Pitts-Neuron)

Mit Lernregel erhält man ein Perceptron

$$w_{ij}(t+1) = w_{ij}(t) + \alpha (t_j - y_j) \cdot x_i$$

↑  
Lernrate

Target (Zielausgabe)

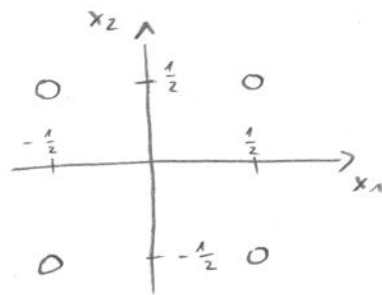
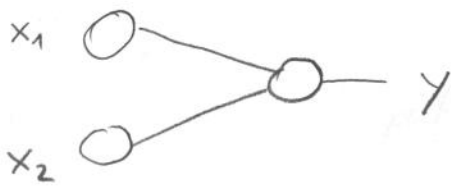
Oh, was wenn  $x_i = 0$   
Homogene Koo.

- Trainingsdaten  $D = \{(x_n, t_n)\}$
- Trainingsmethoden

Wahl des  $\alpha$   
Randomisiert etc

Beispiel 2 Input, ein Output, binär ( $\pm \frac{1}{2}$ )

$l=2$   $n=1$



$x_1$	$x_2$	$y$
0	0	0
1	0	0
0	1	0
1	1	0
0	0	1
1	0	1
0	1	1
1	1	1

$D_1$   $D_2$   $D_3$   
 $D_1$   
 $D_1$   
 $D_3$   
 $D_2$   $D_3$   
 $D_2$   $D_3$   
 $D_1$   $D_2$

immer nur 4 ( $2^l$ ) sich nicht widersprechende Trainingsdaten

$D_1 \equiv \text{AND}$   
 $D_2 \equiv \text{OR}$   
 $D_3 \equiv \text{XOR}$

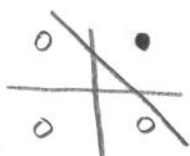
Wieviele Möglichkeiten?

$$2^3 = 2^{l+n}$$

Frage können alle logischen Operatoren durch geeignete Wji realisiert werden?

XOR - Gatter nicht

AND



OR



XOR

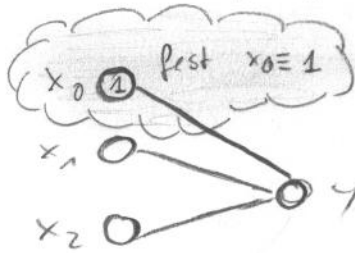
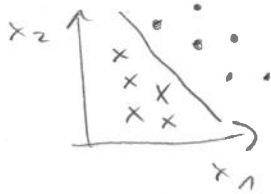


Nicht linear separierbar!

# Die Schwelle

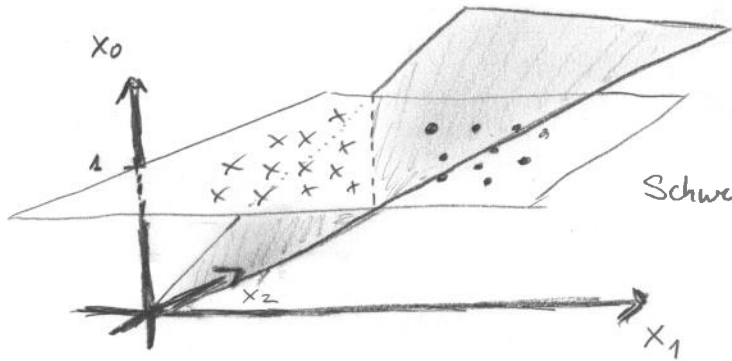
$$y_k = \Theta \left( \sum_{j=0}^m w_{kj} x_j \right)$$

$\equiv 0$  definiert eine Hyperfläche Codim. 1 durch Ursprung



- Namen
- on-Neuron
  - Bias
  - Schwelle

Was es wirklich ist



Schwelle hebt Daten auf Ebene

- kommt aus der Projektiven Geometrie

## homogene Koordinaten

affine Transformationen werden durch Matrizen dargestellt. (Translation ist ausserdem eine Vektoraddition.)

- XOR Problem bleibt bestehen!

# Backpropagation - Fehlerrückführende Netze

Am Beispiel von Nature 331,679 (1988)  
Zipser u. Andersen.

Zunächst: Stabile Raumwahrnehmung trotz Eigenbewegung

Ähnlich dem XOR Problem

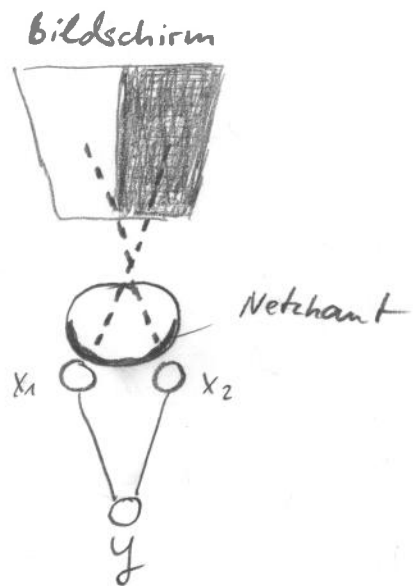
Frage:

Kann  $y$  entscheiden, ob

ein Kontrast-Muster

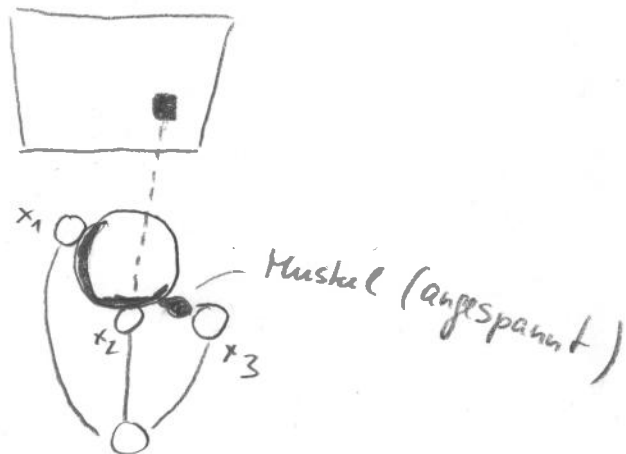
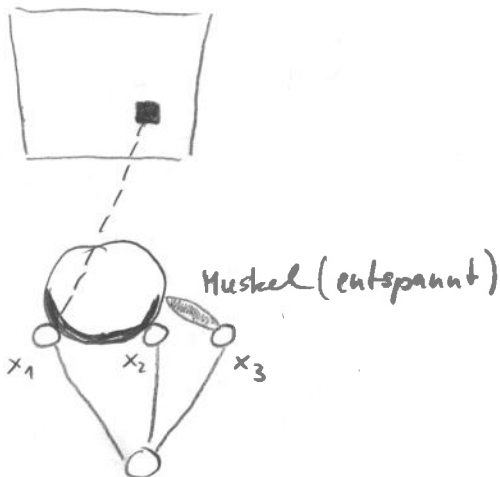
(links hell, rechts dunkel oder vice versa) auf dem Bildschirm ist?

Nein. □



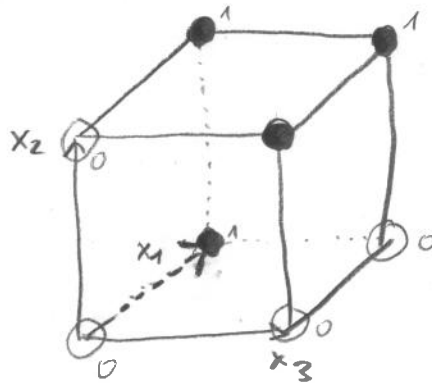
## Drei Zellen

Frage: ist rechts unten ein Quadrat?



Wahrheitstabelle

$x_1$	$x_2$	$x_3$	$y$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

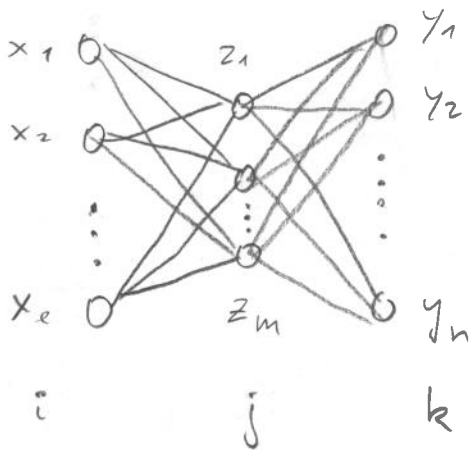


nicht linear separierbar

Je größer die Dimension desto wahrscheinlicher, dass sich Daten in Ebenen verteilen.

Lösung: Mehrschichtige Netzwerke unterliegen nicht der Beschränkung linearer Trennbarkeit.

Datenfluss →



← Fehlerfluss

allerdings

Credit Assignment Problem:

Wie lernen die Neurone der "versteckten" Zwischenschicht?

Lösung: Fehlerrückführung (Backpropagation)

Fehlerfunktion

$$E_p = \frac{1}{2} \left( \sum_{k=1}^n (y_k - t_k)^2 \right)$$

Lernregel soll Fehlergradienten hinabstoßen!

$$\nabla E = \left( \frac{\partial E}{\partial w_{11}^{vs}}, \dots, \frac{\partial E}{\partial w_{ml}^{vs}}, \frac{\partial E}{\partial w_{11}^{os}}, \dots, \frac{\partial E}{\partial w_{nm}^{os}} \right)$$

$\uparrow$   
 $m \times l$  Ableitungen in der versteckten Schicht (vs)

$n \times m$  ... Ausgangsschicht (os)

Mit

$$z_j = \varphi(a_j) \quad \text{und} \quad y_k = \varphi(a_k)$$

$$a_j = \sum_{i=1}^l w_{ji} x_i$$

$$a_k = \sum_{j=0}^m w_{kj} z_j$$

Wenn  $\varphi(\cdot)$  differenzierbare Funktion (nicht Heaviside!)

dann ist

$E_p(y_k)$  differenzierbare Funktion der  
Ausgangsaktivitäten  $y_k$

Schichtung führt zur Kettenregel

$$(f \circ g)' = f'(g(\cdot)) \cdot g'(\cdot)$$

Steifung der Fehlerfunktion  
Ausgangsschicht

$$\frac{\partial E_p}{\partial w_{kj}} = \frac{\partial E_p}{\partial a_k} \cdot \frac{\partial a_k}{\partial w_{kj}} = \delta_k z_j$$

mit Fehler  $\delta_k := \frac{\partial E_p}{\partial a_k} = \varphi'(a_k) \frac{\partial E_p}{\partial y_k}$

Analogy in der versteckten Schicht

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} = \delta_j x_i \quad \text{mit}$$

Fehler  $\delta_j := \frac{\partial E_p}{\partial a_j}$  ← kennt man nicht

nach der Kettenregel

$$\delta_j = \sum_{k=1}^n \frac{\partial E_p}{\partial a_k} \frac{\partial a_k}{\partial a_j}$$

$$= \sum_{k=1}^n \delta_k \frac{\partial (\sum_{r=1}^m w_{kr} \varphi(a_r))}{\partial a_j}$$

"j=r"-Termin  
wird  
herausgegriffen

$$\delta_j = \varphi'(a_j) \sum_{k=1}^n \delta_k w_{kj}$$

Backpropagation Regel

Interpretation:

Wäre  $\varphi(a_j) = a_j$  (Identität, statt Sigmoid)

$\Rightarrow \varphi'(a_j) = 1$ , dann:

Fehler ("Credit") des Neurons  $j$  berechnet sich aus  $w_{kj}$ -gewichteter Summe aller Fehler der Ausgangsschicht.

Beispiel

$\varphi(a_k) \equiv a_k$  lineare Transferfunktion in Ausgangsschicht

$$\varphi(a_j) \equiv \frac{1}{1 + e^{-a_j}}$$

$$\varphi'(a_j) = g(a_j) (1 - g(a_j))$$

super, haben wir ja schon berechnet! siehe Datenfluss.

$$\delta_k = y_k - t_k$$



$$\delta_j = z_j(1-z_j) \sum_{k=1}^n w_{kj} \delta_k$$

$$\frac{\partial E_p}{\partial w_{ji}} = \delta_j x_i \quad \text{und} \quad \frac{\partial E_p}{\partial w_{kj}} = \delta_k z_j$$

Lernregel

$$\Delta w_{ji} = -\alpha \delta_j x_i$$

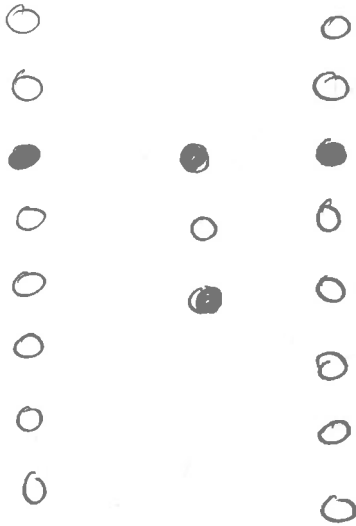
$$\Delta w_{kj} = -\alpha \delta_k z_j$$

Wie beim Perzeptron,  $\alpha$  Lernrate

Ziel bei Fehlerrückführenden Netzen :

# Interne Datenrepräsentation

Muster #3 von 8 Mustern



Kann dies durch eine 3-zellige versteckte Schicht gelöst werden?

"Flaschenhals"

... die v.S. entdeckt den

Binärcode  $2^3 = 8$  ✓

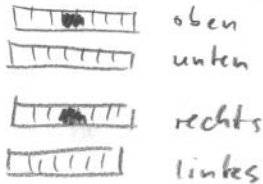
Analog Refferenzprinzip und Backpropagation nach

Zipser & Andersen Nature 1988

Rehinotopie Karte V1, V2



Efferenzkopie A8, A19, A22



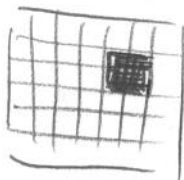
MT, MST, VIP



guter Kandidat, da

entsprechende

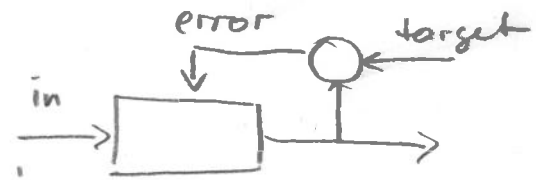
Kodierung



Motorische Zentren

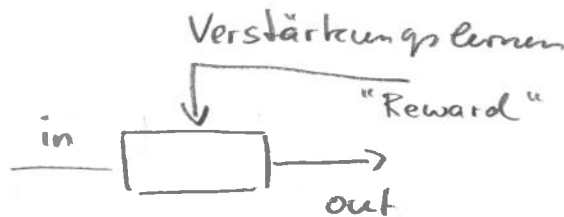
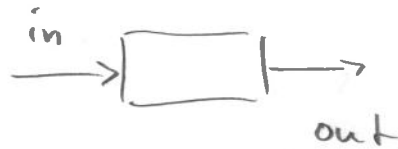
# Probleme und Plausibilität

- Überwachtes Lernverfahren
- Informationsfluss vom post- zum präsynaptischen Neuron
- Retrosynaptisches Signal lokal an der Synapse



## Alternativen

unüberwacht



Mathematisch / physikalisch

opt. Gradientenabstieg setzt Metrik voraus  
 führt zu "Information Geometry"

# Hopfield Netz

Rekurrente Netze haben Rückkopplungsschleifen

Hopfield sind Eingabe- und Ausgabeschicht gleich dem gesamten Netz.

Zustände  $y_i \in \{-1, 1\}$  Spinvariable

Symmetrische Kopplungsgewichte (synaptische Gewichte)

$w_{ij} = w_{ji}$  aber keine Selbstkopplung  $w_{ii} = 0$

Jedes Neuron hat eine Schwelle  $\Theta_i$

Zur Zeit  $t$  wirkt auf Neuron  $i$  lokales Feld

$$v_i(t) = \sum_{j=1}^N w_{ij} y_j(t) - \Theta_i$$

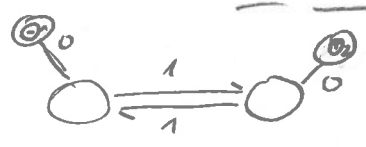
Transferfunktion

$$y_{t+1} = H(v) = \begin{cases} +1 & \text{für } v \geq 0 \\ -1 & \text{für } v < 0 \end{cases} \quad \text{oder}$$

$$y_{t+1} = \begin{cases} +1 & v > 0 \\ -1 & \text{für } v < 0 \\ y_t & v = 0 \end{cases}$$

Beispiel

Synchrone Schaltdynamik



$S(0) = (+1, -1)$

$S(1) = (-1, +1)$

$S(2) = (+1, -1)$

...

periodische Lösung

Stabiler Zustand eines rekurrenten Netzes wenn alle Neuronen sich nicht ändern bei  $t \rightarrow t+1$

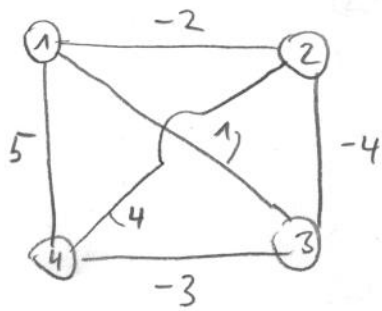
Bei asynchroner Schaltdynamik

endet man immer in einem stabilen Zustand !!

- 1. Wähle jedes Neuron mit Wahrscheinlichkeit  $\frac{1}{N}$
- 2. dann Update
- 3. goto 1.

Beispiel

Endet im stabilen Zustand?



alle Schwellen  $\Theta_i = 0$

Beweis über Energiefunktion

$$E(y) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i w_{ij} y_j + \sum_{i=1}^N y_i \Theta_i$$

$$E(y) = -\frac{1}{2} y^T w y + y^T \Theta \quad (\text{Vektorschreibweise})$$

$$(-1, -1, -1, -1) \mapsto -1 \quad \begin{matrix} \text{aktiv} \\ \downarrow \downarrow \\ (-4, 2, 6, -6) \end{matrix}$$

$$(-1, -1, -1, +1) \mapsto 11 \quad \begin{matrix} \downarrow \downarrow \downarrow \\ (6, 10, 0, -6) \end{matrix}$$

$$(-1, -1, +1, -1) \mapsto -13 \quad (-2, -6, +6, -12)$$

$$(-1, +1, -1, -1) \mapsto -5 \quad \begin{matrix} \uparrow \text{kein aktives Neuron!} \\ (-8, 2, -2, 2) \\ \uparrow \end{matrix}$$

$$(-1, +1, -1, +1) \mapsto -9 \quad ($$

Energieniveau  
↓

11  $(-1, -1, -1, 1)$  und drei andere

-1  $(-1, -1, -1, -1)$

$(-1, +1, -1, -1)$

-13  $(-1, -1, +1, -1)$

