

# Mathematikaufgabe 88

---

[Home](#) | [Startseite](#) | [Impressum](#) | [Kontakt](#) | [Gästebuch](#)

**Aufgabe:** Konzipieren Sie eine logische UND- bzw. ODER-Schaltung mit Hilfe eines neuronalen Netzwerks.

**Lösung:** Die Wahrheitstafel einer Ja-Nein-Maschine mit zwei Eingängen  $x_1$  und  $x_2$  und einem Ausgang  $y$  für die logische Konjunktion und Disjunktion lautet:

$x_1$	$x_2$	$x_1 \wedge x_2$	$x_1 \vee x_2$	$w_{i1}x_1 + w_{i2}x_2$
0	0	0	0	0
1	0	0	1	1
0	1	0	1	1
1	1	1	1	2

Durch die Vorschrift

$$y_1 = g(w_{10}x_0 + w_{11}x_1 + w_{12}x_2),$$
$$y_2 = g(w_{20}x_0 + w_{21}x_1 + w_{22}x_2)$$

wird jedem Mustervektor  $\mathbf{x}$  ein Ausgabevektor  $\mathbf{y} = (y_1, y_2)$  zugeordnet, der jeweils vier verschiedene Werte  $y_i = \mathbf{g}(\mathbf{w}_i \cdot \mathbf{x})$  annehmen kann, wobei das Skalarprodukt gegeben ist durch

$$\mathbf{w}_i \cdot \mathbf{x} = \sum_{j=0}^2 w_{ij}x_j.$$

Die Schwellen werden dabei durch die konstante Eingabe  $x_0 = 1$  und die Gewichte  $w_{i0}$  berücksichtigt. Wie man sich überzeugen kann, lösen die Gewichte

$$\text{UND: } (w_{10}, w_{11}, w_{12}) = (-1.5, 1.0, 1.0),$$
$$\text{ODER: } (w_{20}, w_{21}, w_{22}) = (-0.5, 1.0, 1.0)$$

das Problem vollständig. Nach unserer Vektorgleichung wird der Musterraum  $\mathbf{x} = (x_1, x_2)$  in zwei Klassen unterteilt, die der Bedingung

$$\mathbf{w}_i \cdot \mathbf{x} < 0 \quad \text{bzw.} \quad \mathbf{w}_i \cdot \mathbf{x} > 0$$

genügen. Die Gleichung  $\mathbf{w}_i \cdot \mathbf{x} = 0$  trennt diese beiden Klassen und entspricht einer Hyperebene im Musterraum, zu der der Vektor  $\mathbf{w}_i$  jeweils orthogonal ist. Für die beiden Verknüpfungen sind die Hyperebenen Geraden, die von den folgenden Gleichungen beschrieben werden:

$$\text{UND: } x_1 + x_2 = 1.5,$$
$$\text{ODER: } x_1 + x_2 = 0.5.$$

## Mathematikaufgabe 88

---

Verwendet man die Heavyside-Funktion, kann eine scharfe Trennung in die beiden Klassen  $y=1$  und  $y=0$  vorgenommen werden. Nimmt man wie im nachfolgenden MATLAB-Beispielprogramm eine Sigmoidfunktion, ist die Ausgangsaktivität  $y$  ein Maß für den noch nicht in Sättigung befindlichen Abstand von der Hyperebene.

### Anhang

```
% Programm undoderschaltung

% Die Transferfunktion out ist eine Sigmoidkurve
% Lernregel ist die Deltaregel
% Achtung: Das Programm funktioniert nicht für die XOR-Verknüpfung
clear all

% Wahrheitstabelle der zu verknüpfenden Größen
input = [0 0; 1 0; 0 1; 1 1];
numIn = 4;
% Erwünschter Output für die UND-Verknüpfung
% desired_out = [0;0;0;1];
% Erwünschter Output für die ODER-Verknüpfung
desired_out = [0;1;1;1];
% Konstante w0*x0 muß zwischen -1 und +1 liegen
bias = -1;
% Lernrate
coeff = 0.7;
rand('state',sum(100*clock));
weights = -1*2.*rand(3,1);

% Zahl der Iterationen
iterations = 1000;

for i = 1:iterations
    out = zeros(4,1);
    for j = 1:numIn
        y = bias*weights(1,1)+input(j,1)*weights(2,1)+input(j,2)*weights(3,1);
        out(j) = 1/(1+exp(-y));
        delta = desired_out(j)-out(j);
        weights(1,1) = weights(1,1)+coeff*bias*delta;
        weights(2,1) = weights(2,1)+coeff*input(j,1)*delta;
        weights(3,1) = weights(3,1)+coeff*input(j,2)*delta;
    end
end

out
weights

>> undoderschaltung
out =
    0.0072
    0.9971
    0.9971
    1.0000
weights =
    4.9241
   10.7758
   10.7718
```