

# NEURONALE NETZE

---

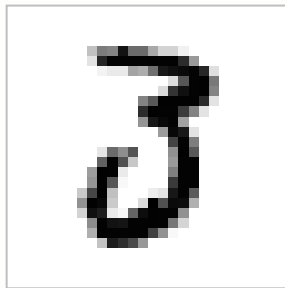
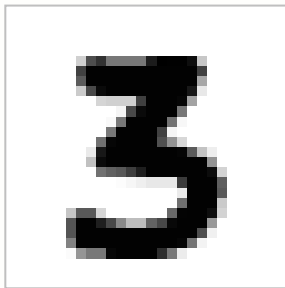
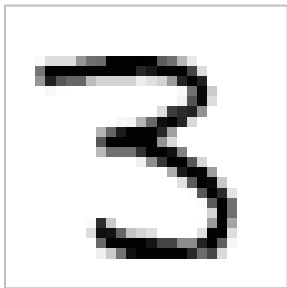
Handschriftliche Zahlen erkennen

Jasper Gude

27. November 2023  
Carl-Friedrich-Gauß-Gymnasium

## 2.1

# Modellierung des Problems



**Modellierung**



Künstliche Neuronen



Training

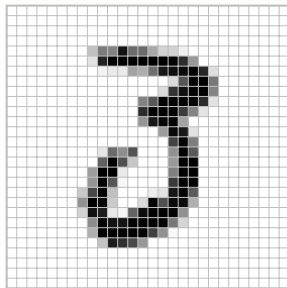
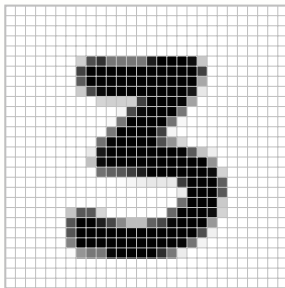
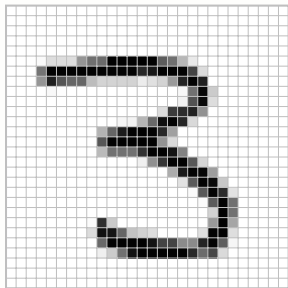


Topologie



## 2.2

# Modellierung des Problems



**Modellierung**



Künstliche Neuronen



Training

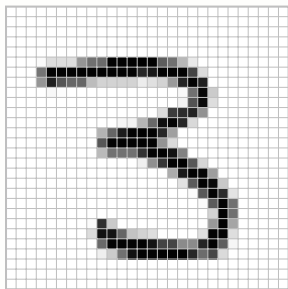


Topologie

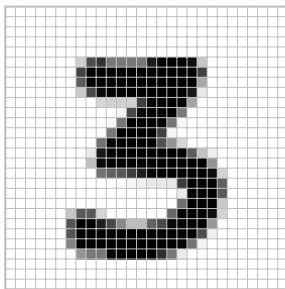


## 2.3

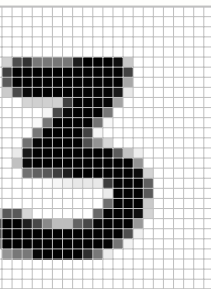
# Modellierung des Problems



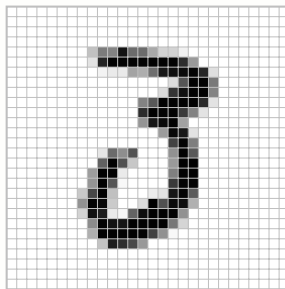
0,00



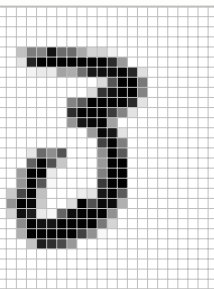
0,25



0,50



0,75



1,00



**Modellierung**



Künstliche Neuronen



Training

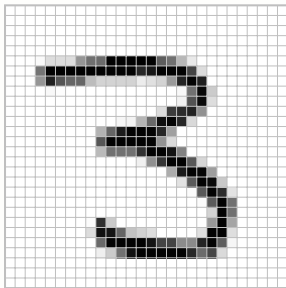


Topologie



### 3.1

## Überführung auf eine Netzstruktur



28px × 28px



**Modellierung**



Künstliche Neuronen



Training

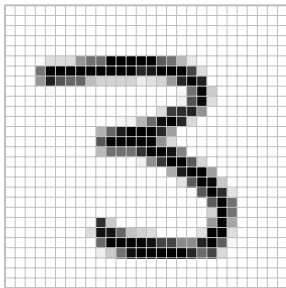


Topologie

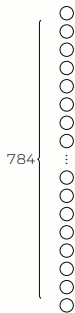


## 3.2

# Überführung auf eine Netzstruktur

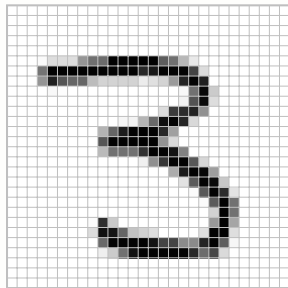


28px x 28px



### 3.3

## Überführung auf eine Netzstruktur



28px x 28px



- 9
- 8
- 7
- 6
- 5
- 4
- 3
- 2
- 1
- 0



Modellierung



Künstliche Neuronen



Training

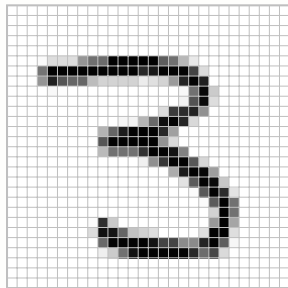


Topologie

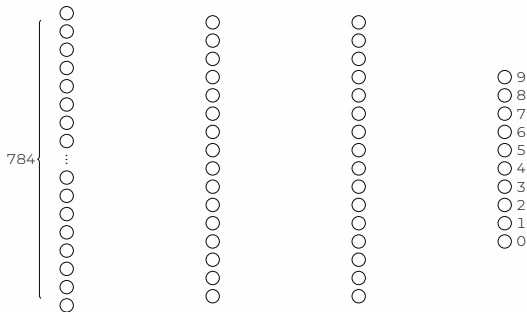


### 3.4

## Überführung auf eine Netzstruktur



28px x 28px



Modellierung



Künstliche Neuronen



Training



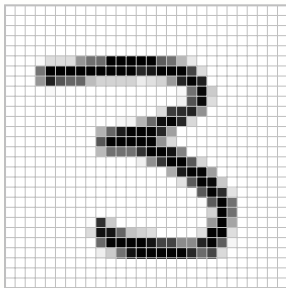
Topologie



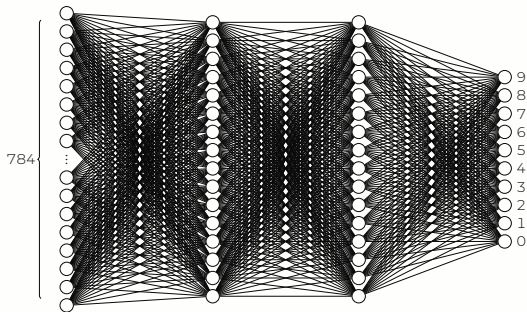


### 3.5

## Überführung auf eine Netzstruktur



28px x 28px



Modellierung



Künstliche Neuronen



Training

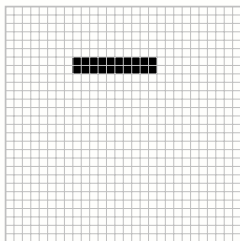


Topologie



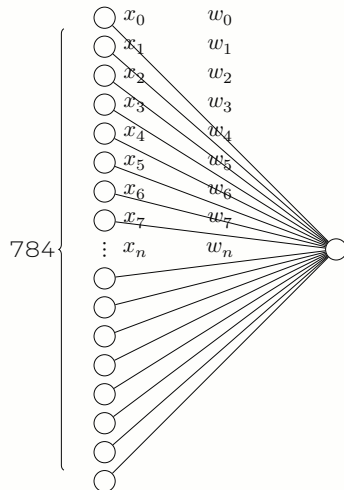
## 4.1

## Gewichtungen setzen



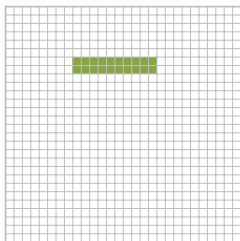
Linearkombination

$$x_0 w_0 + x_1 w_1 + x_2 w_2 + \dots + x_n w_n$$



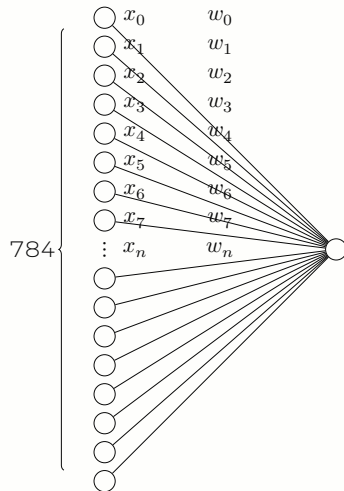
## 4.2

## Gewichtungen setzen



Linearkombination

$$x_0w_0 + x_1w_1 + x_2w_2 + \dots + x_nw_n$$



Modellierung



Künstliche Neuronen



Training

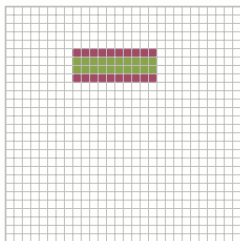


Topologie



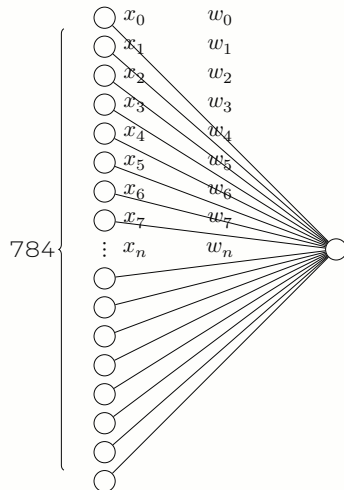
## 4

## Gewichtungen setzen



Linearkombination

$$x_0w_0 + x_1w_1 + x_2w_2 + \dots + x_nw_n$$



Modellierung



Künstliche Neuronen



Training

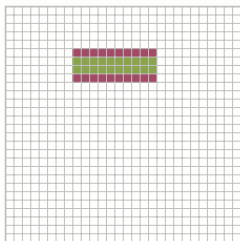


Topologie



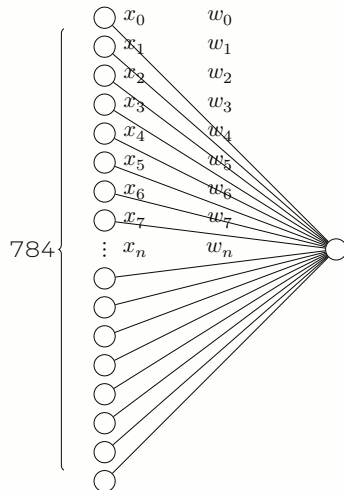
## 4

## Gewichtungen setzen



Linearkombination

$$x_0w_0 + x_1w_1 + x_2w_2 + \dots + x_nw_n - 10$$



Modellierung



Künstliche Neuronen



Training



Topologie

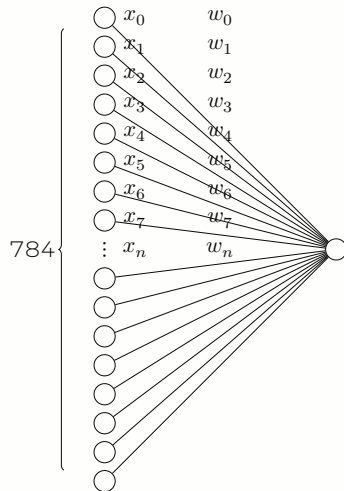
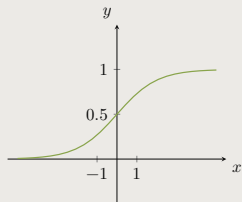


## 5

## Zahlenbereich begrenzen

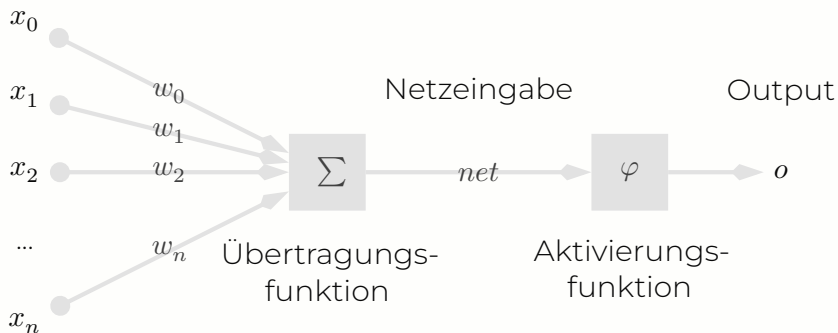
## Sigmoidfunktion

$$\varphi(x) = \frac{1}{1 + e^{-x}}$$



## 6

## Aufbau eines Perzeptrons

Inputvektor  $\vec{x}$ 

## 7

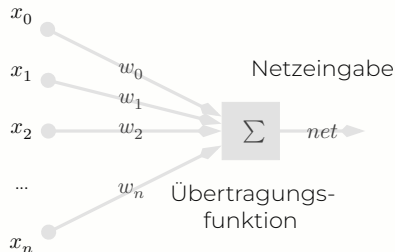
## Übertragungsfunktion

## Linearkombination

$$net = x_0w_0 + x_1w_1 + x_2w_2 + \dots + x_nw_n$$

oder

$$net = \sum_{i=0}^n x_iw_i$$

Inputvektor  $\vec{x}$ 



## 8

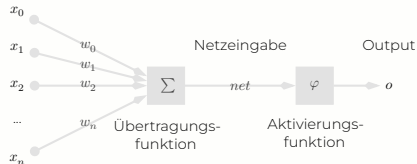
## Fehlerfunktion

## Dataset

$$X = \left\{ (\vec{x}_0, y_0); (\vec{x}_1, y_1); (\vec{x}_2, y_2); (\dots, \dots); (\vec{x}_n, y_n) \right\}$$

## Mean Squared Error

$$E = \frac{1}{2} \sum_{i=0}^n (y_i - o_i)^2$$

Inputvektor  $\vec{x}$ 

# 9

## Dataset

$$X = \left\{ (\vec{x}_0, y_0); (\vec{x}_1, y_1); (\vec{x}_2, y_2); (\dots, \dots); (\vec{x}_n, y_n) \right\}$$

## 10

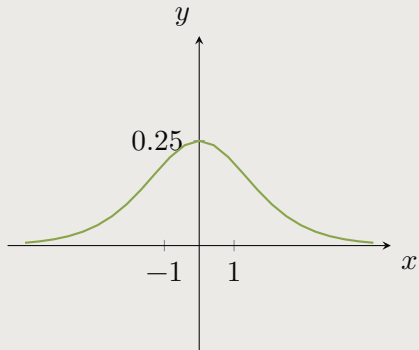
## Ableitung der Aktivierungsfunktion

## Ableitung der Sigmoidfunktion

$$\varphi'(x) = \frac{1}{1 + e^{-x}} \cdot \left(1 + \frac{1}{1 + e^{-x}}\right)$$

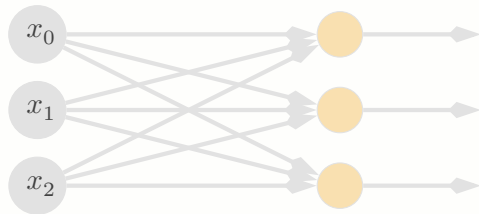
oder

$$\varphi'(x) = \varphi(x) \cdot (1 + \varphi(x))$$



# 11

## Einschichtiges feedforward-Netz



Ausgabeschicht



Modellierung



Künstliche Neuronen



Training



**Topologie**



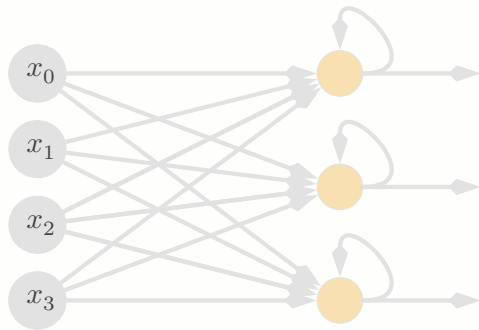
## 12

# Mehrschichtiges feedforward-Netz



# 13

## Rekurrentes Netz



Ausgabeschicht



Modellierung



Künstliche Neuronen



Training



**Topologie**



**Jasper Gude**

Hockenheim, 27. November  
2023