

# Künstliche Intelligenz – Wie geht das denn?

## Methoden und Komponenten am Beispiel der Bildverarbeitung

Stefan Helmert

Chaostreff Chemnitz e.V.

3./4. November 2018

## ① Was wollen wir?

Anwendungen

Autonomie

## ② Umsetzung

Da gibt es doch was auf Github

Überwachtes Lernen

Unüberwachtes Lernen

Backpropagation

Neuron

Faltungsnetz

Deeplearning

XOR-Problem

Aktivierungsfunktion

## ③ Praxis

Der Netzwerk-Graph

Aufwand

## ④ Fragen?

# Was wollen wir?

## Anwendungen

- Bildersuche
- Suche in Videos
- Klassifikation von Mediendaten
- Analyse der Wirkung und Stimmung von Medien
- Verbesserung
  - Monochrom → Farbe
  - SD → HD
- Erstellung neuer Inhalte

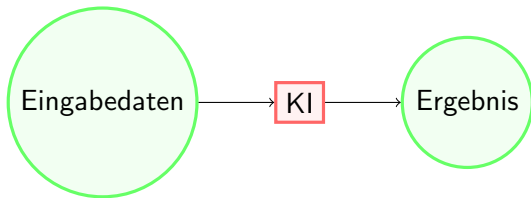
# Was wollen wir?

## Autonomie

- Lernen statt Programmieren
- Keine (einschränkenden) Vorgaben
- Funktion ohne Vorwissen
- Eigenständige Verbesserung
- Effizienz

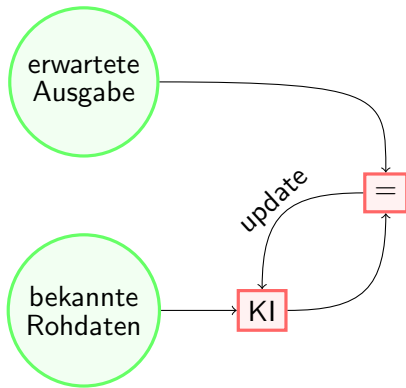
# Umsetzung

Da gibt es doch was auf Github



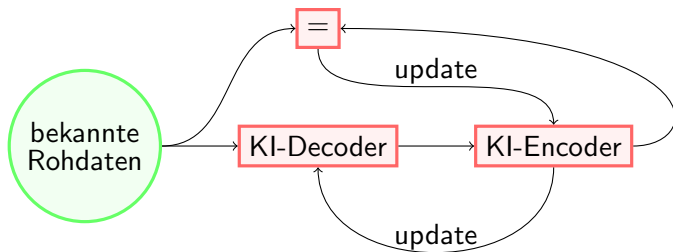
# Umsetzung

## Überwachtes Lernen



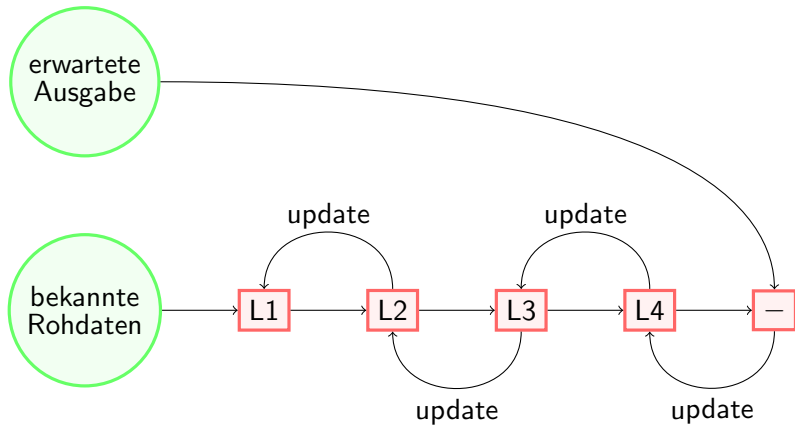
# Umsetzung

## Unüberwachtes Lernen



# Umsetzung

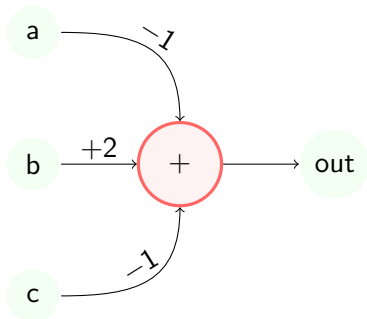
## Backpropagation





# Umsetzung

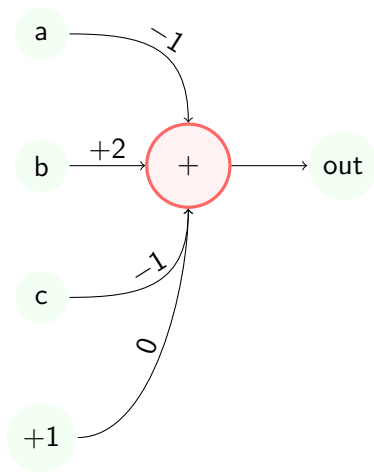
## Neuron



Was fehlt noch?

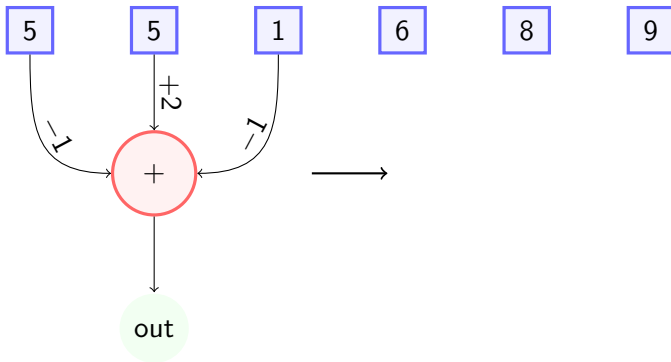
# Umsetzung

Neuron



# Umsetzung

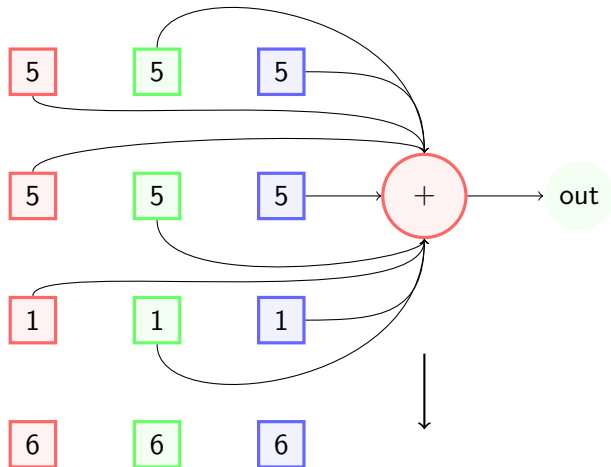
## Faltungsnetz



Und bei Farbbildern?

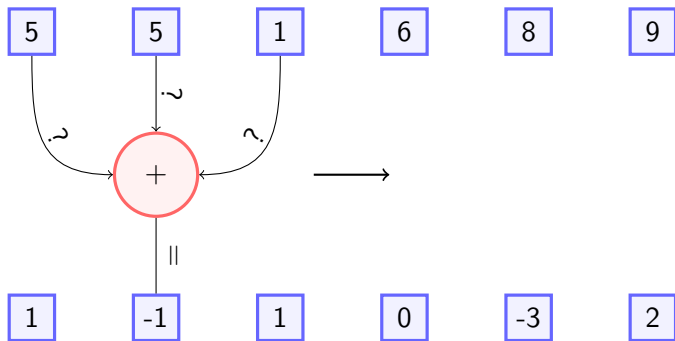
# Umsetzung

## Faltungsnetz



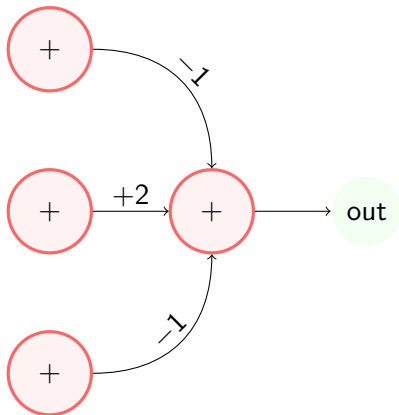
# Umsetzung

## Faltungsnetz



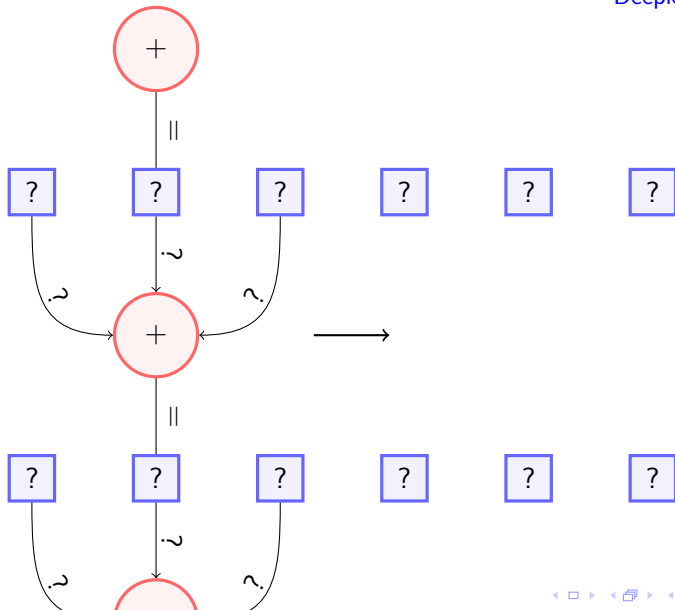
# Umsetzung

## Deeplearning



# Umsetzung

## Deeplearning



# Umsetzung

## XOR-Problem

$a$	$b$	$\text{not}(a)$	$\text{or}(a, b)$	$\text{and}(a, b)$	$\text{xor}(a, b)$
0	0	1	0	0	0
0	1	1	1	0	1
1	0	0	1	0	1
1	1	0	1	1	0



# Umsetzung

## XOR-Problem

$a$	$b$	$\text{not}(a)$	$\text{or}(a, b)$	$\text{and}(a, b)$	$\text{xor}(a, b)$
0	0	1	0	0	0
0	1	1	1	0	1
1	0	0	1	0	1
1	1	0	1	1	0

$$\text{xor}(a, b) = \text{and}(\text{or}(a, b), \text{not}(\text{and}(a, b)))$$

$$\text{xor}(a, b) = \text{or}(a, b) - \text{and}(a, b)$$

# Umsetzung

## XOR-Problem

$$\text{xor}(a, b) = \text{or}(a, b) - \text{and}(a, b)$$

$$\text{not}(a) = 1 - a; \text{or}(a, b) = a + b; \text{and}(a, b) = \frac{a + b}{2}$$

$a$	$b$	$\text{not}(a)$	$\text{or}(a, b)$	$\text{and}(a, b)$	$\text{xor}(a, b)$
0	0	1	0	0	0
0	1	1	1	0,5	0,5
1	0	0	1	0,5	0,5
1	1	0	2	1	1

# Umsetzung

## XOR-Problem

$$\text{and}(a, b) = a \cdot b$$

→ Nein, das ist nicht linear.

# Umsetzung

## XOR-Problem

$$\text{xor}(a, b) = \text{and}(\text{or}(a, b), \text{not}(\text{and}(a, b)))$$

$$\text{not}(a) = 1 - a$$

$$\text{or}(a, b) = a + b$$

$$\text{and}(a, b) = \frac{a + b}{2}$$

$$\text{xor}(a, b) = \frac{(a + b) + \left(-\frac{a+b}{2}\right)}{2} = \frac{a + b}{4}$$

# Umsetzung

## XOR-Problem

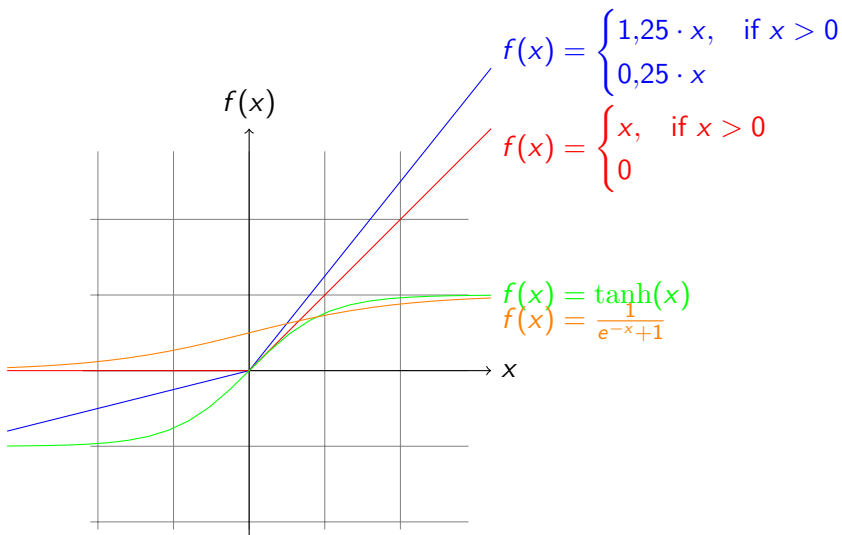
$$\text{xor}(a, b) = \frac{(a + b) + \left(-\frac{a+b}{2}\right)}{2} = \frac{a + b}{4}$$

$a$	$b$	$\text{not}(a)$	$\text{or}(a, b)$	$\text{and}(a, b)$	$\text{xor}(a, b)$
0	0	1	0	0	0
0	1	1	1	0,5	0,25
1	0	0	1	0,5	0,25
1	1	0	2	1	0,5

# Umsetzung

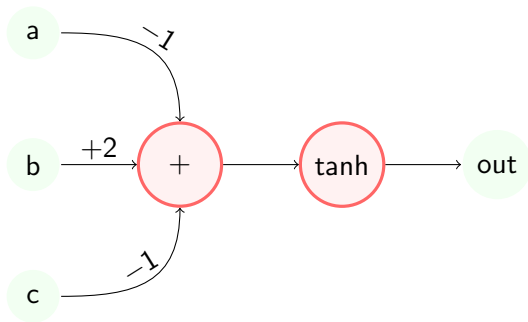
## Aktivierungsfunktion

### Lösung – Aktivierungsfunktion



# Umsetzung

## Aktivierungsfunktion



# Umsetzung

## Aktivierungsfunktion

$$\text{xor}(a, b) = \tanh(\text{or}(a, b)) - \text{and}(a, b)$$

$$\text{or}(a, b) = a + b$$

$$\text{and}(a, b) = \frac{a + b}{2}$$

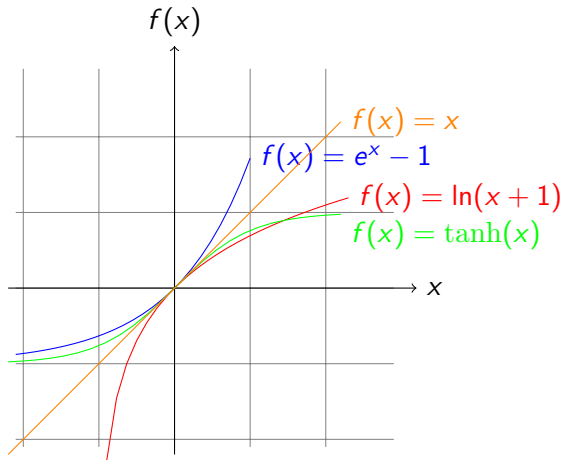
$a$	$b$	$\text{not}(a)$	$\tanh(\text{or}(a, b))$	$\text{and}(a, b)$	$\text{xor}(a, b)$	$4 \cdot \text{xor}(a, b)$
0	0	1	0	0	0	0
0	1	1	0,76	0,5	0,26	1,04
1	0	0	0,76	0,5	0,26	1,04
1	1	0	0,96	1	-0,04	-0,16



# Umsetzung

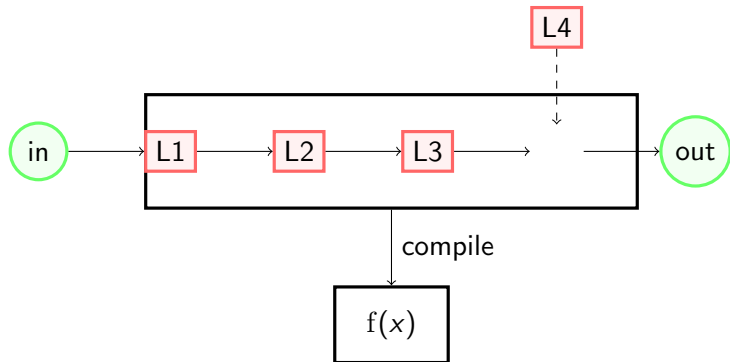
## Aktivierungsfunktion

Aktivierungsfunktion  $\tanh(x)$  approximiert andere Funktionen.



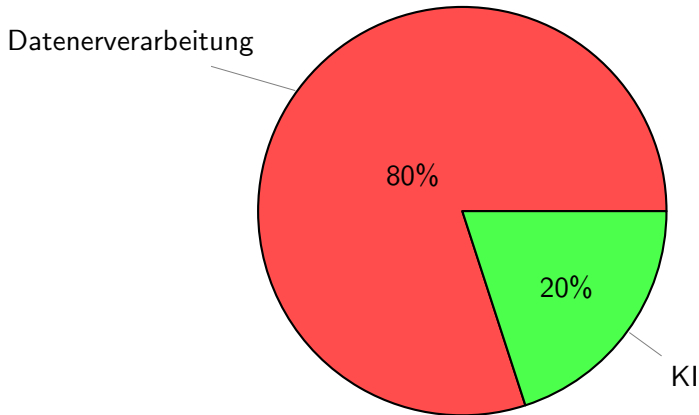
# Praxis

## Der Netzwerk-Graph



# Praxis

Aufwand



Fragen?

???