TECHNISCHE UNIVERSITÄT CHEMNITZ

SAXSIM
Topologieoptimierung mittels Deep Learning

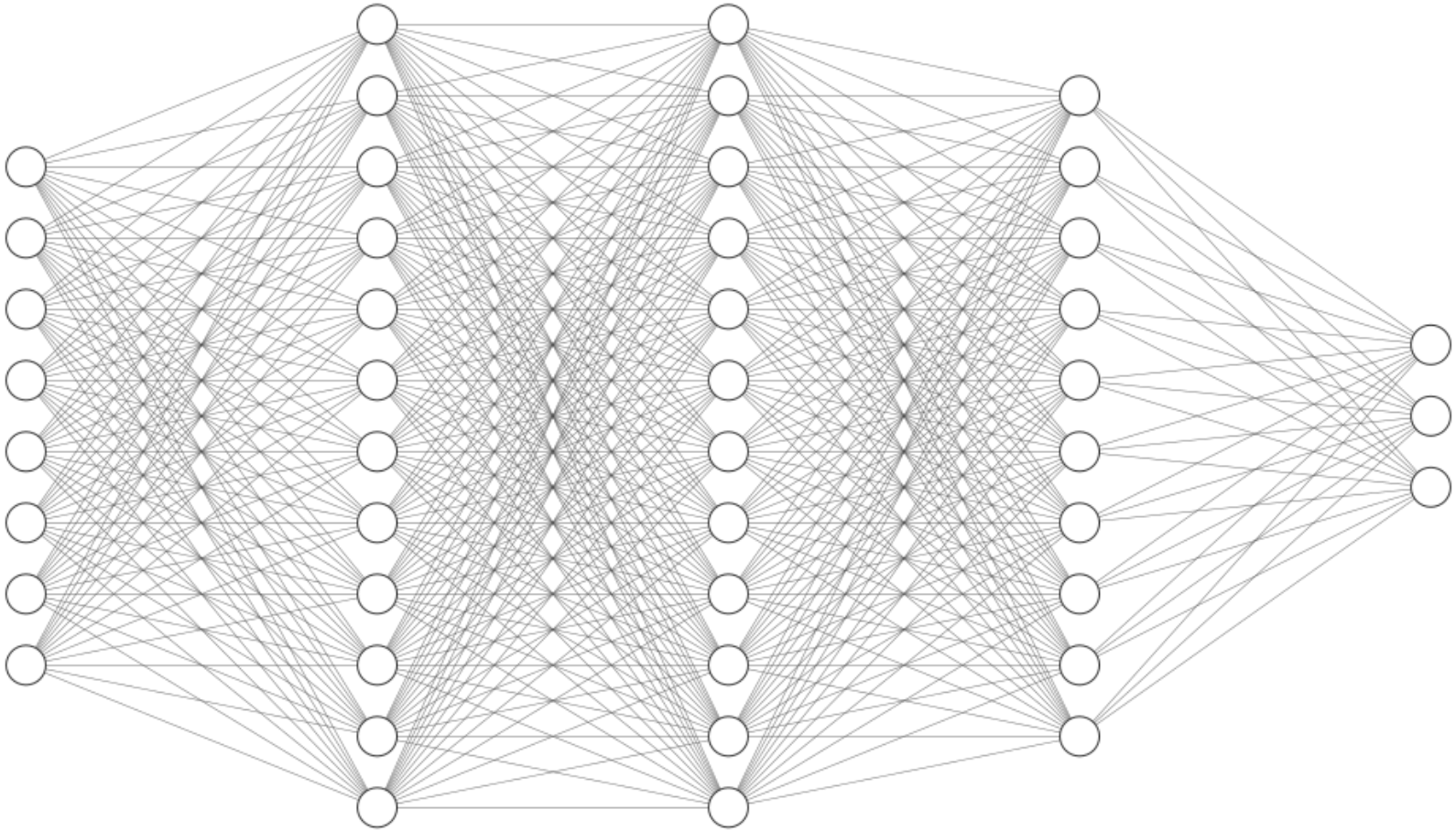
Professur Maschinenelemente und Produktentwicklung (MP)

Institut für Konstruktions- und Antriebstechnik

M. Sc. Alex Halle
Prof. Dr. sc. ETH A. Hasse

TU Chemnitz
Reichenhainer Straße 70
09126 Chemnitz

Tel.: +49 371 531 30225
Fax.: +49 371 531 830225
E-Mail: alex.halle@
mb.tu-chemnitz.de
www.tu-chemnitz.de/mb/mp

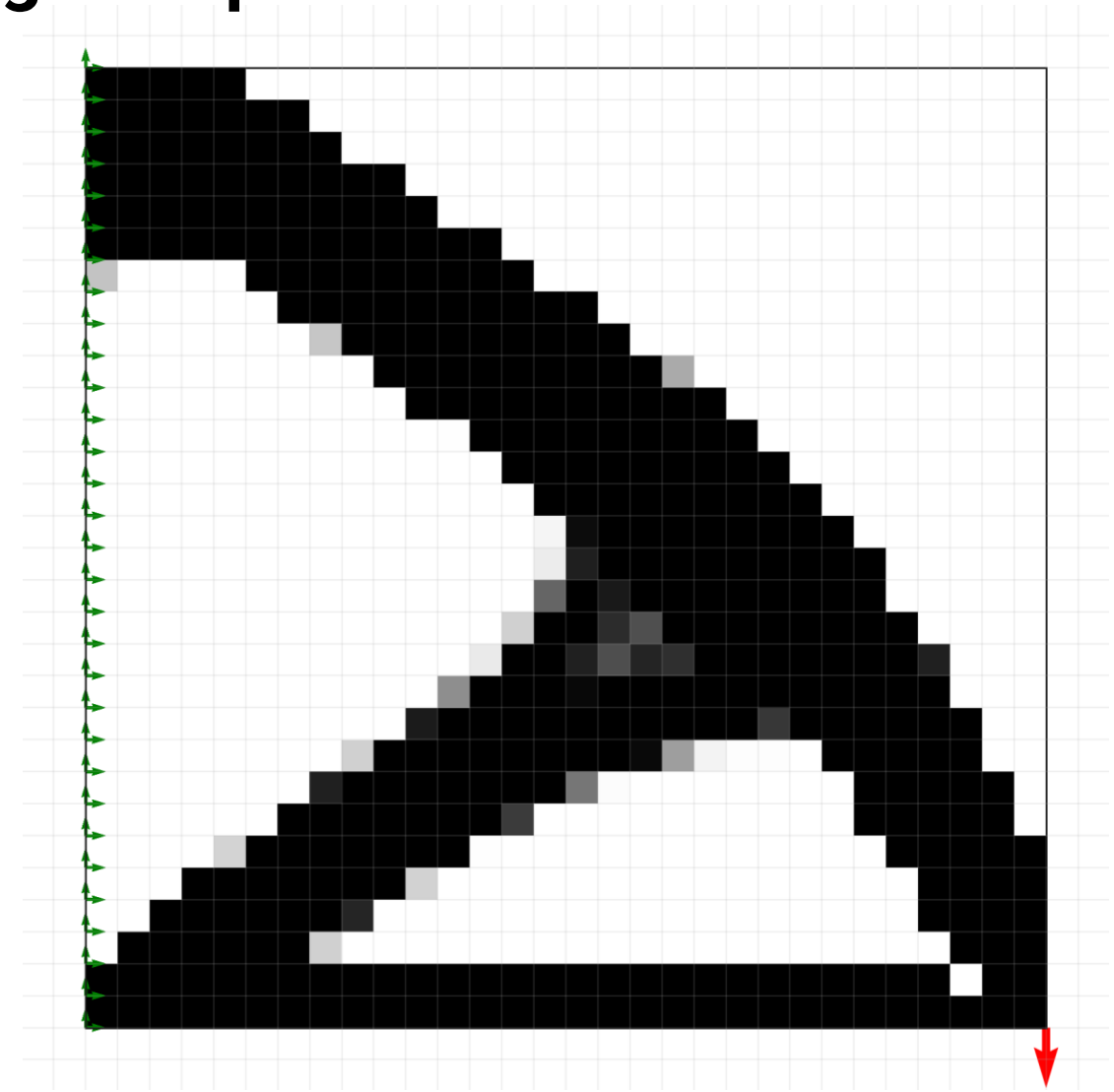


Einleitung: Topologieoptimierung

… Berechnung der optimalen Gestalt (Topologie)

- Unter gewissen Randbedingungen
  - Kräfte
  - Lager
  - Bauraum
  - Füllgrad
- Nach gewissen Kriterien:
  - Steifigkeit
  - Spannungsverteilung
  - Eigenfrequenzverhalten

Topologieoptimierung: Beispiel



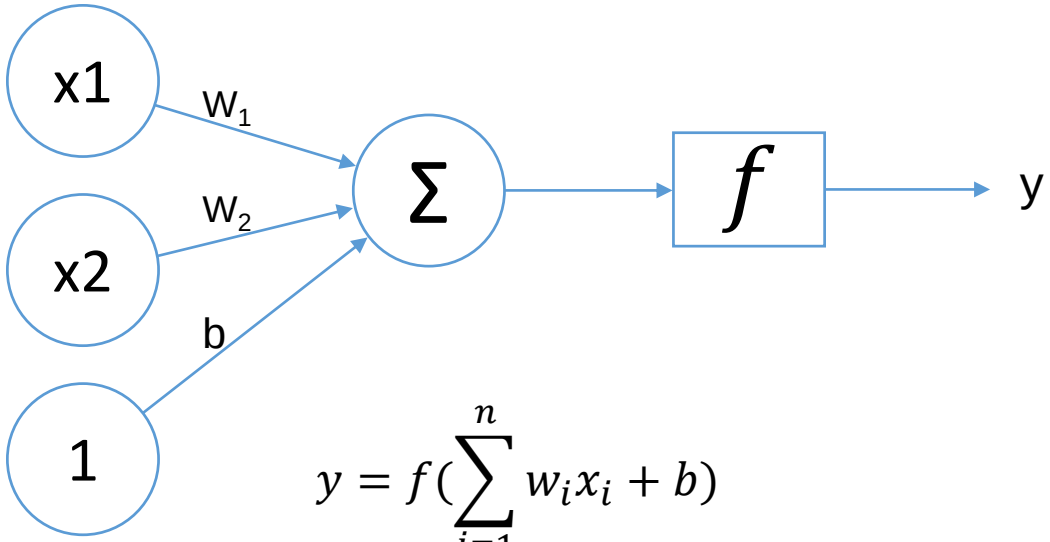
Beispiel nach:
Bendsøe, M. P. (2004),
Topology optimization:
Theory, methods, and
applications

Einleitung: Deep Learning

KNN - Künstliches Neuronales Netzwerk

- 1943 – erstes technisches Neuronenmodell
  - Warren S. McCulloch, Walter Pitts
- Angelehnt an menschliche Neuronen bzw. Nerven
  - mathematischer Ansatz
- Findet Anwendung:
  - Mustererkennung (Zeichen-, Sprach-, Signalerkennung)
  - Autonomes Fahren

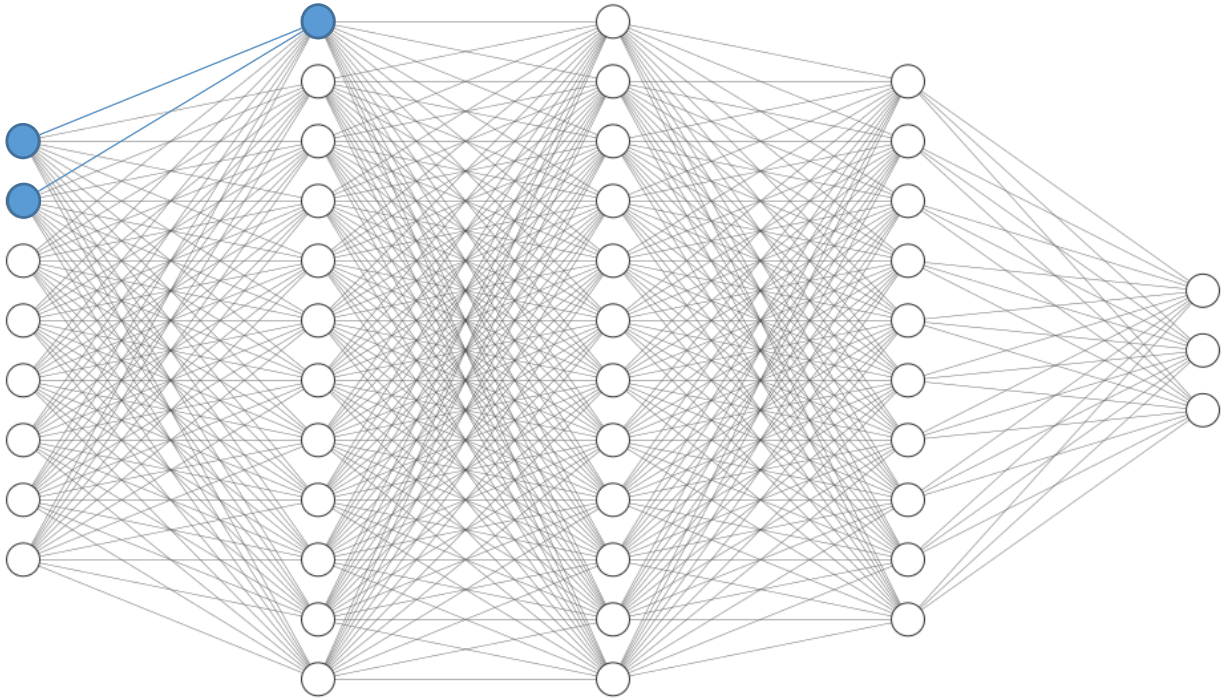
Das Neuron



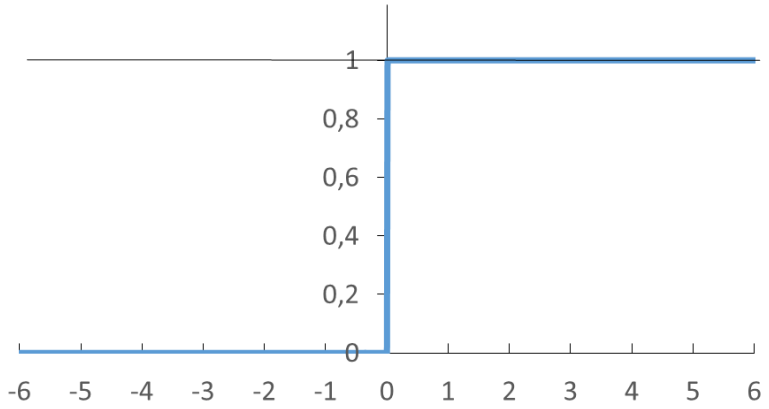
$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

f	Aktivierungsfunktion
w	Gewichtung
b	Schwellwert

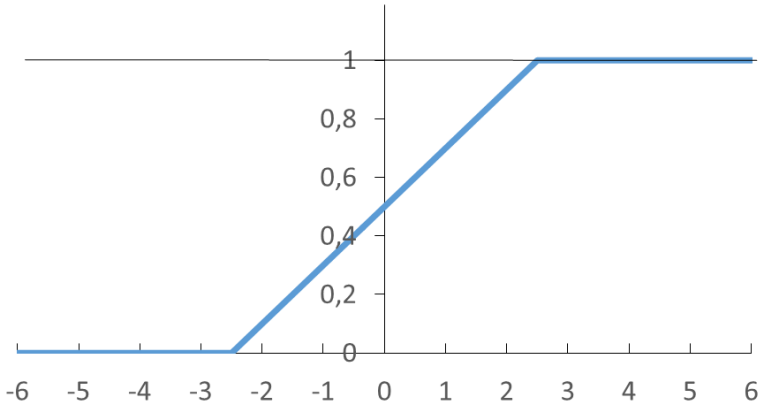
Das Neuron als Teil eines großen Netzes



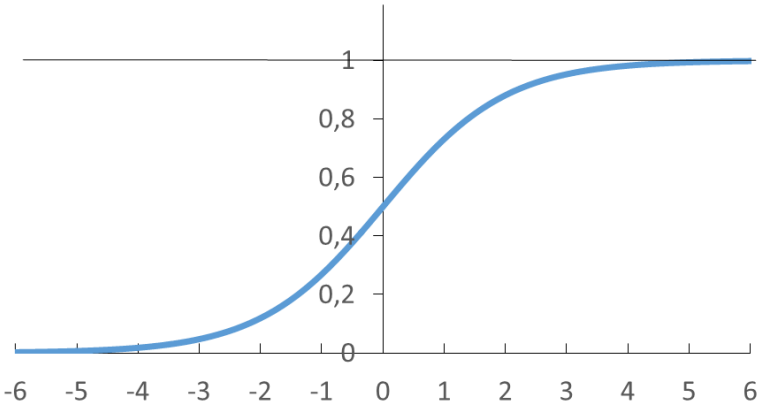
Aktivierungsfunktionen - Auswahl



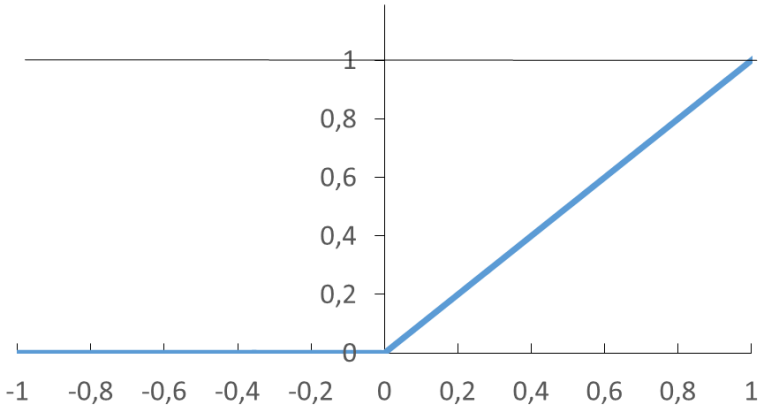
Heaviside



Hardsigmoid



Sigmoid

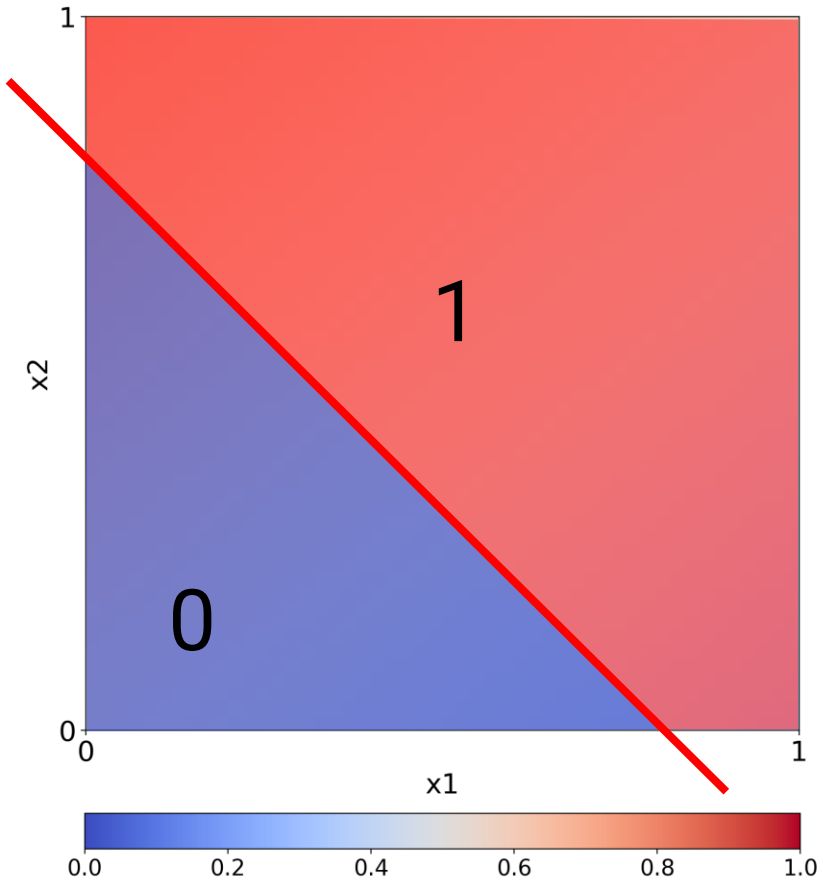


ReLU

Einfaches KNN – OR-Verknüpfung

- Lineare Separierbarkeit möglich

→ einschichtiges KNN reicht

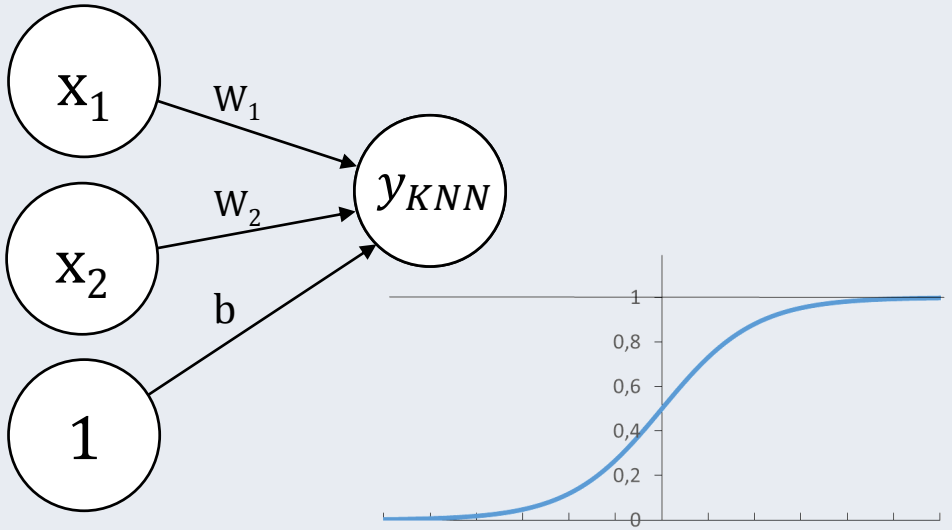


OR-Verknüpfung: KNN

Eingabedateien

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

Künstliches Neuronales Netzwerk



$$y_{KNN} = f(\sum_{i=1}^{n} w_i x_i + b)$$

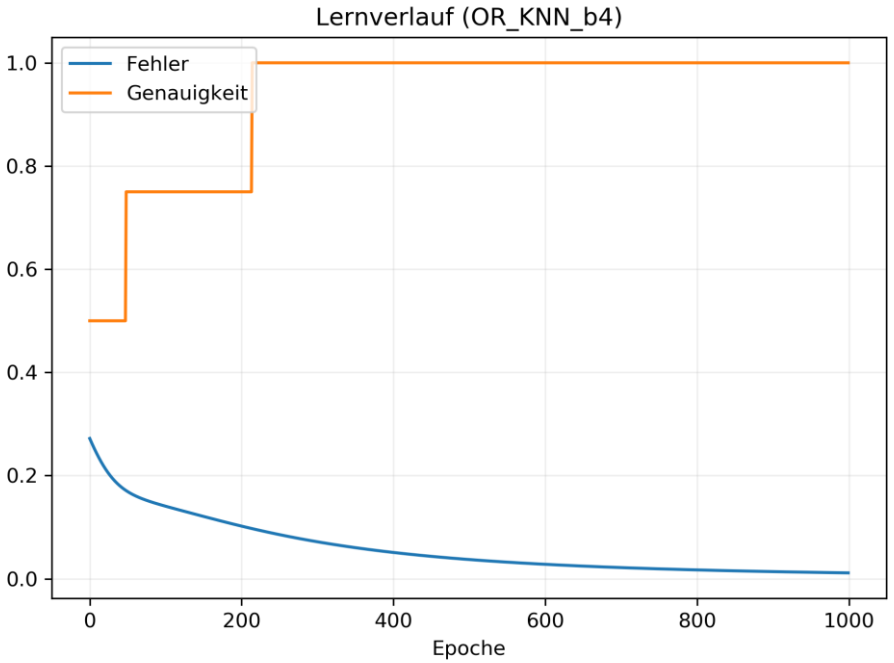
Berechnung

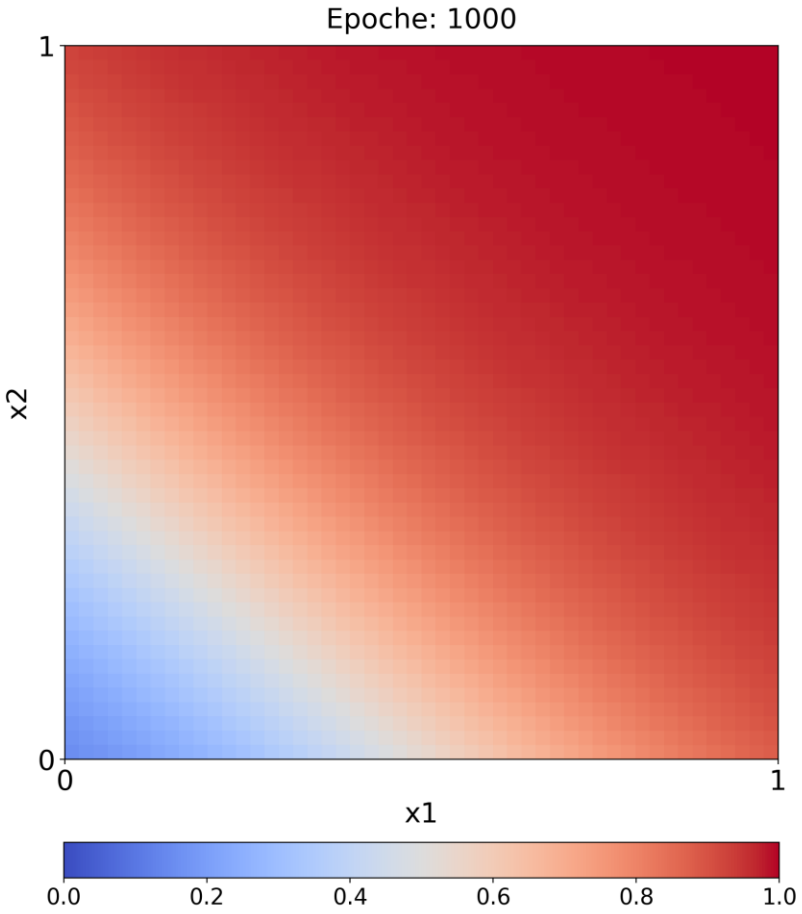
$$y_{KNN} = f(x_1 \cdot w_1 + x_2 \cdot w_2 + b)$$
$$0{,}0474 = f(0 \cdot 5 + 0 \cdot 5 - 3)$$
$$0{,}8808 = f(0 \cdot 5 + 1 \cdot 5 - 3)$$
$$0{,}8808 = f(1 \cdot 5 + 0 \cdot 5 - 3)$$
$$0{,}9991 = f(1 \cdot 5 + 1 \cdot 5 - 3)$$
$$\delta = \frac{\sum_{i=1}^{n}(y_{KNN,j} - y_j)^2}{k} = 0{,}008$$

- δ → Fehler (Mittlerer quadratischer Fehler)
- Fehler soll, durch anpassen der Parameter (Gewichtungen und Schwellwert) minimiert werden:

$$w_i(t+1) = w_i(t) + \eta \cdot x_i \cdot \delta$$
$$b(t+1) = b(t) - \eta \cdot \delta$$

OR-Verknüpfung: Lernen

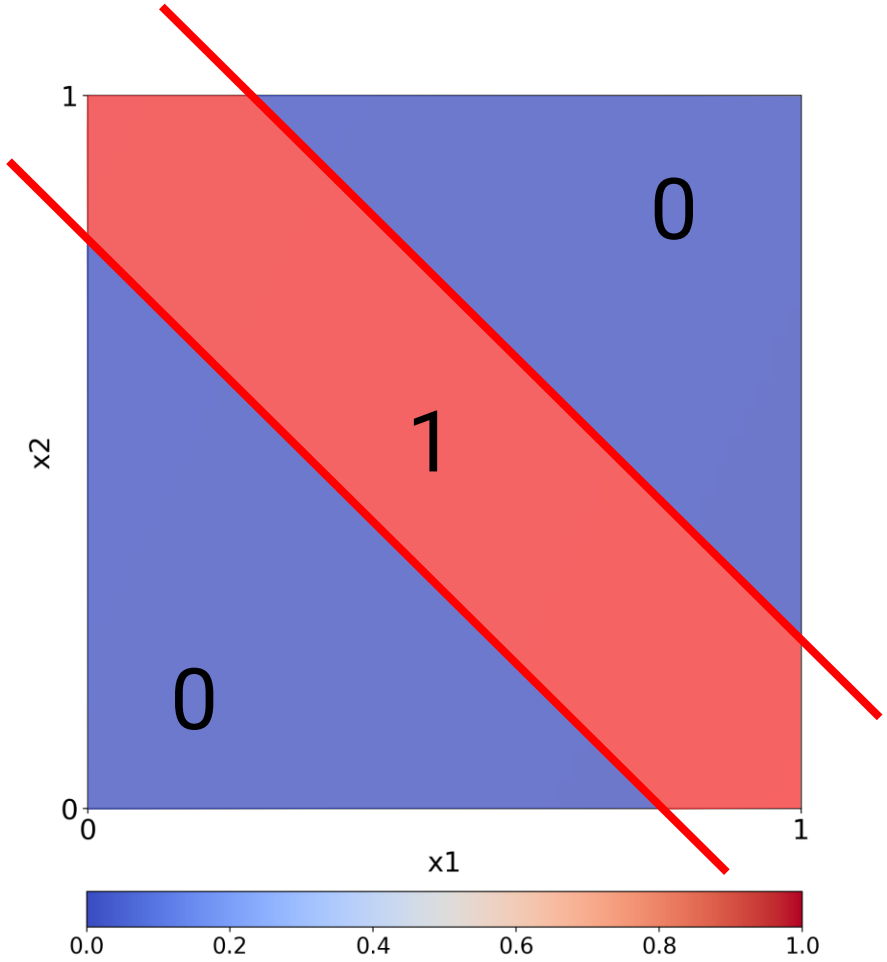




Einfaches KNN – XOR-Verknüpfung

- Lineare Separierbarkeit **nicht** möglich

→ einschichtiges KNN reicht nicht

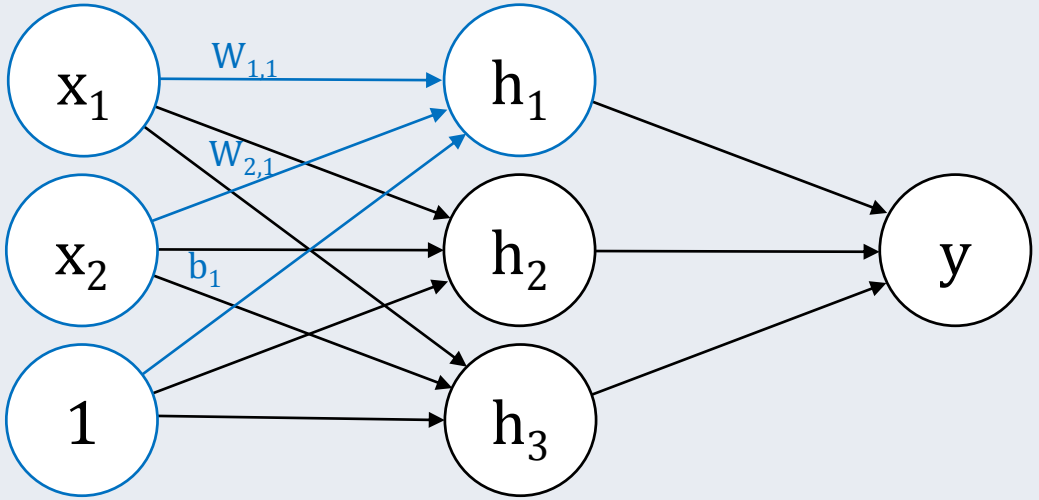


XOR-Verknüpfung: KNN

Eingabedateien

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

Künstliches Neuronales Netzwerk

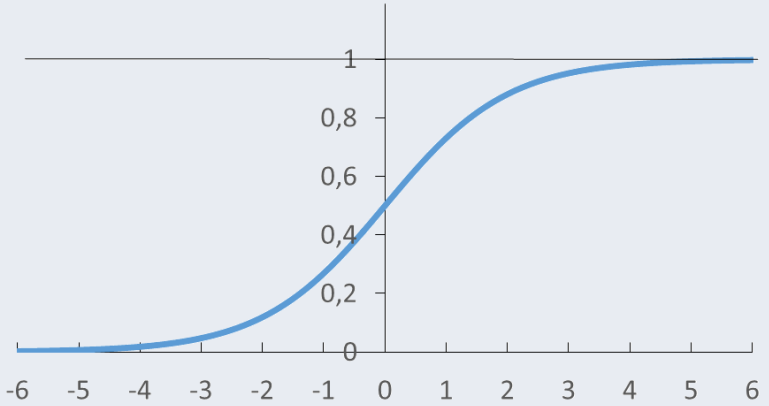


$$h_1 = f(\sum_{i=1}^{2} w_{i,1} x_i + b_1)$$

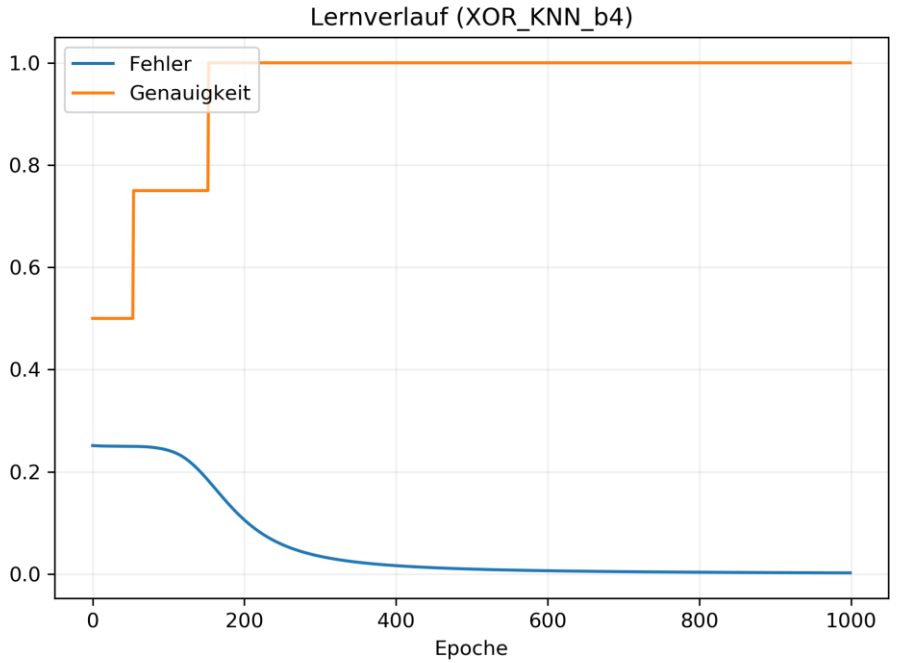
$$y = f(\sum_{j=1}^{3} h_j) = f(\sum_{j=1}^{3} f(\sum_{i=1}^{2} w_{i,j} x_i + b_j))$$

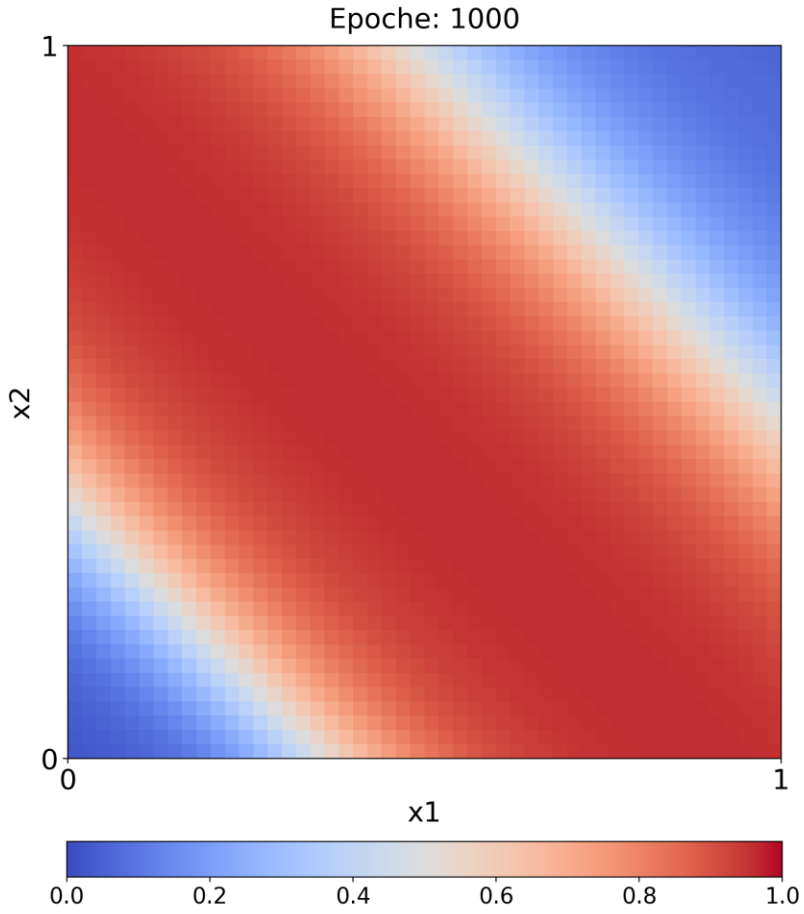
- δ → Fehler (Mittlerer quadratischer Fehler)
- Fehler soll, durch anpassen der Parameter (Gewichtungen und Schwellwert) minimiert werden

→Backpropagation



Einfaches KNN – XOR-Verknüpfung





Was ist Deep Learning?

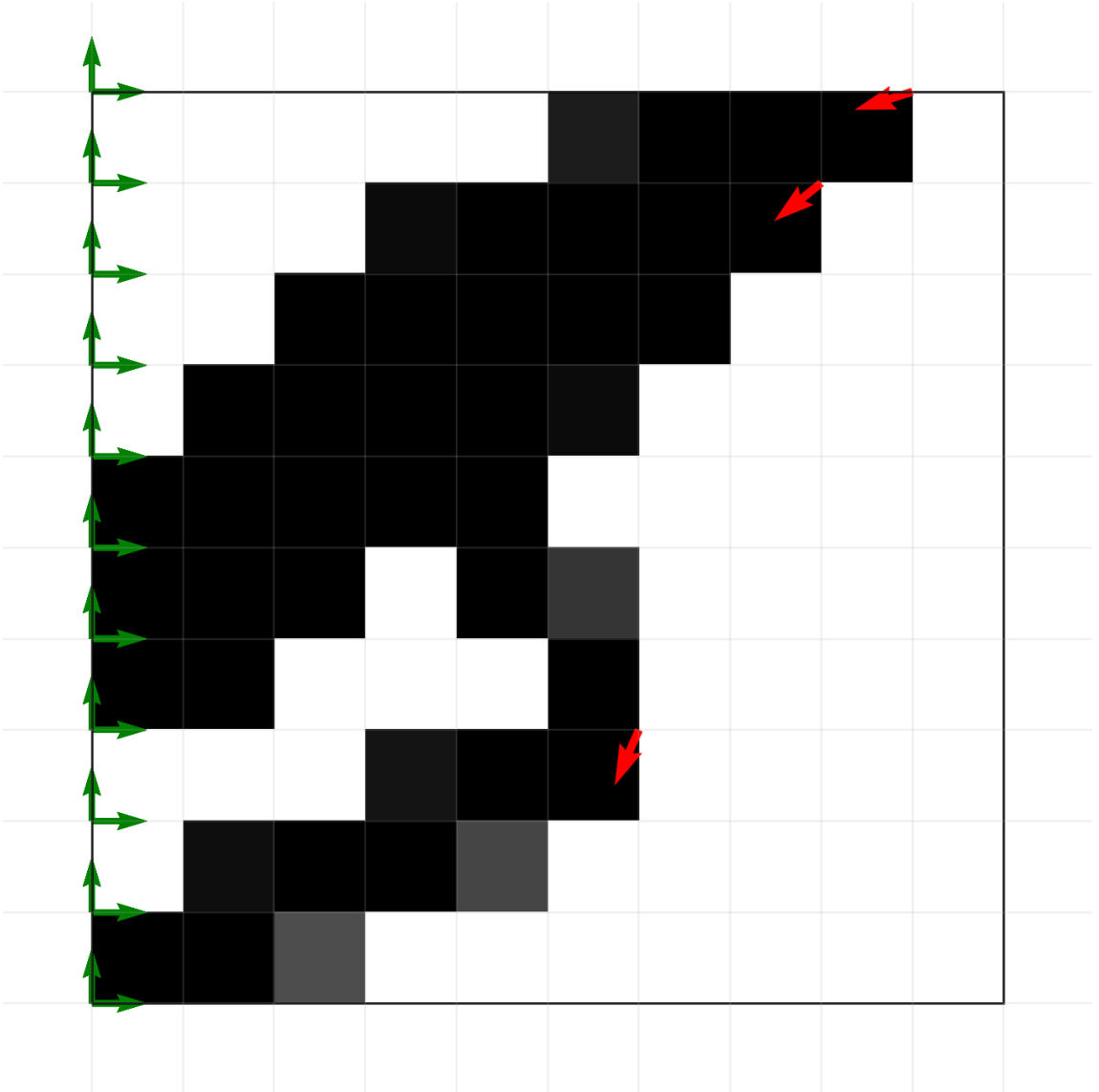
- Bei KNN mit mehr als 2 versteckten Schichten spricht man von Deep Learning

Ziel

- Topologieoptimierung mittels Deep Learning

Beschränkungen

- Größe der Geometrie: 10x10
- Nur Stahlwerkstoff
- Alle Kräfte 100 N
- Lager entlang der linken Seite



Motivation

- Deutlich schnellere „Optimierung“ als herkömmliche Methoden
  - mögliche Designs vorschlagen während des Konstruierens
  - Einsatz auch auf Langsamen Geräten oder
  - Online möglich

Ansatz

- Zwei miteinander Verknüpfte KNN
  1. Qualitätsbestimmung der Geometrie (Diskriminator)
  2. Geometrieerstellung (Generator)
  - Angelehnt an „Generative Adversarial Networks“ (GAN)

Genutz wird:

- Python
- Keras API
- Tensorflow Backend

Quelle: https://www.tensorflow.org/

Diskriminator-KNN 1/4

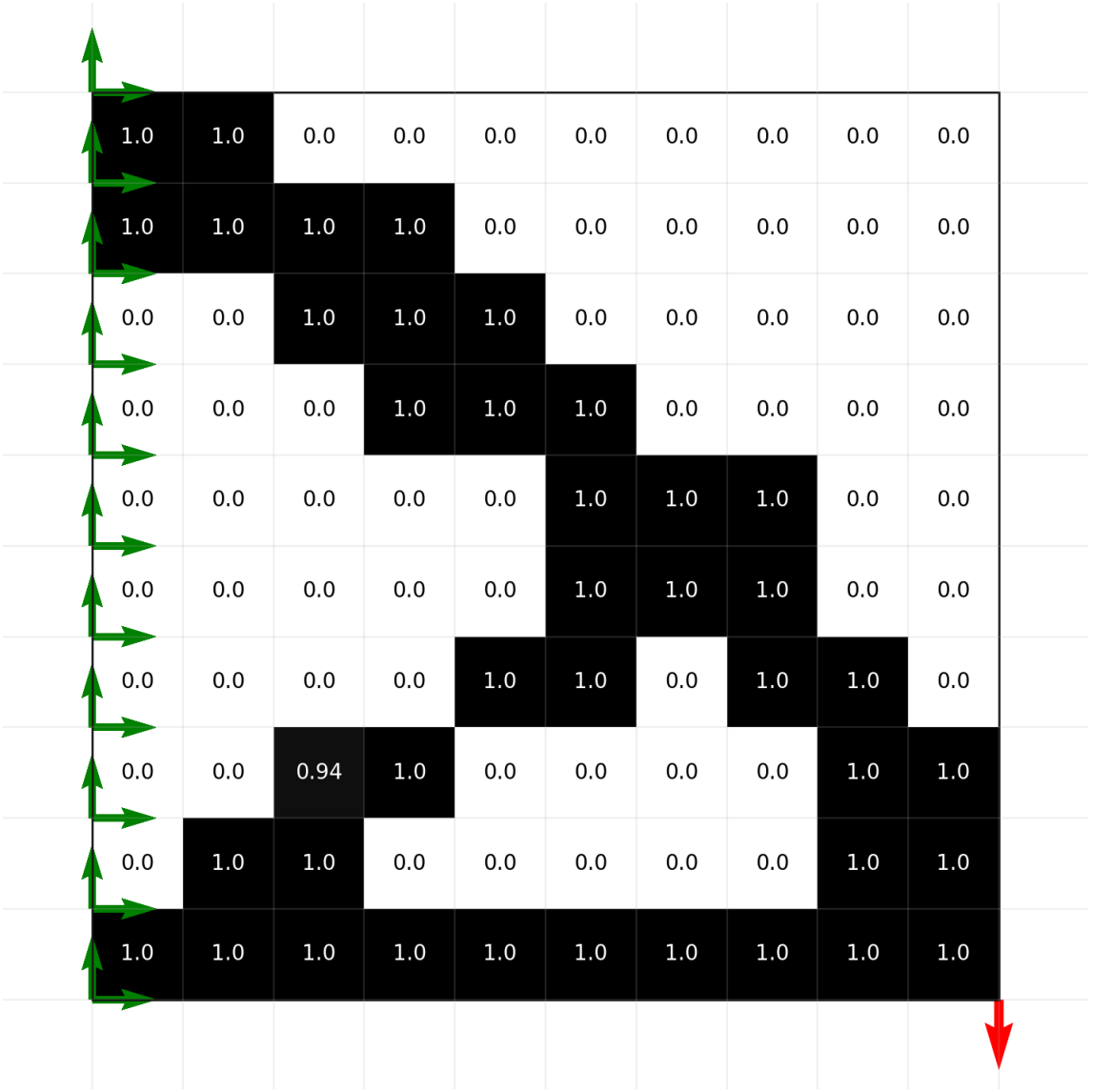
- Alle Gewichte sind fest initialisiert
- Alle Schichten mathematisch vordefiniert

} Bedarf keinen Lernprozess

- Qualitätsbestimmung der Geometrie
  - Ermittelt die Nachgiebigkeit

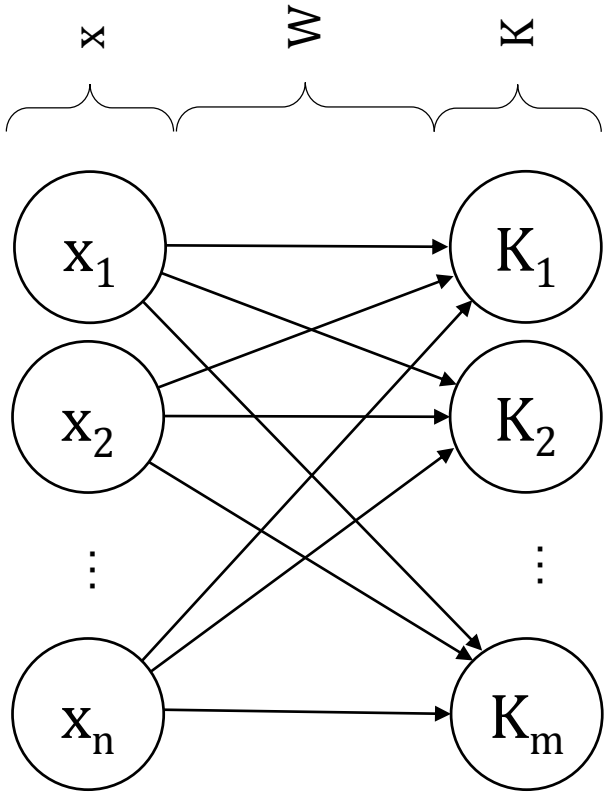
Diskriminator-KNN 2/4

- Grundgleichung der FEM
  - $F = K \cdot U \rightarrow U = K^{-1} \cdot F$
- Aufstellen der Steifigkeitsmatrix K
  - $K = W \cdot x$
  - x : Designvariable (Material, siehe Bild)



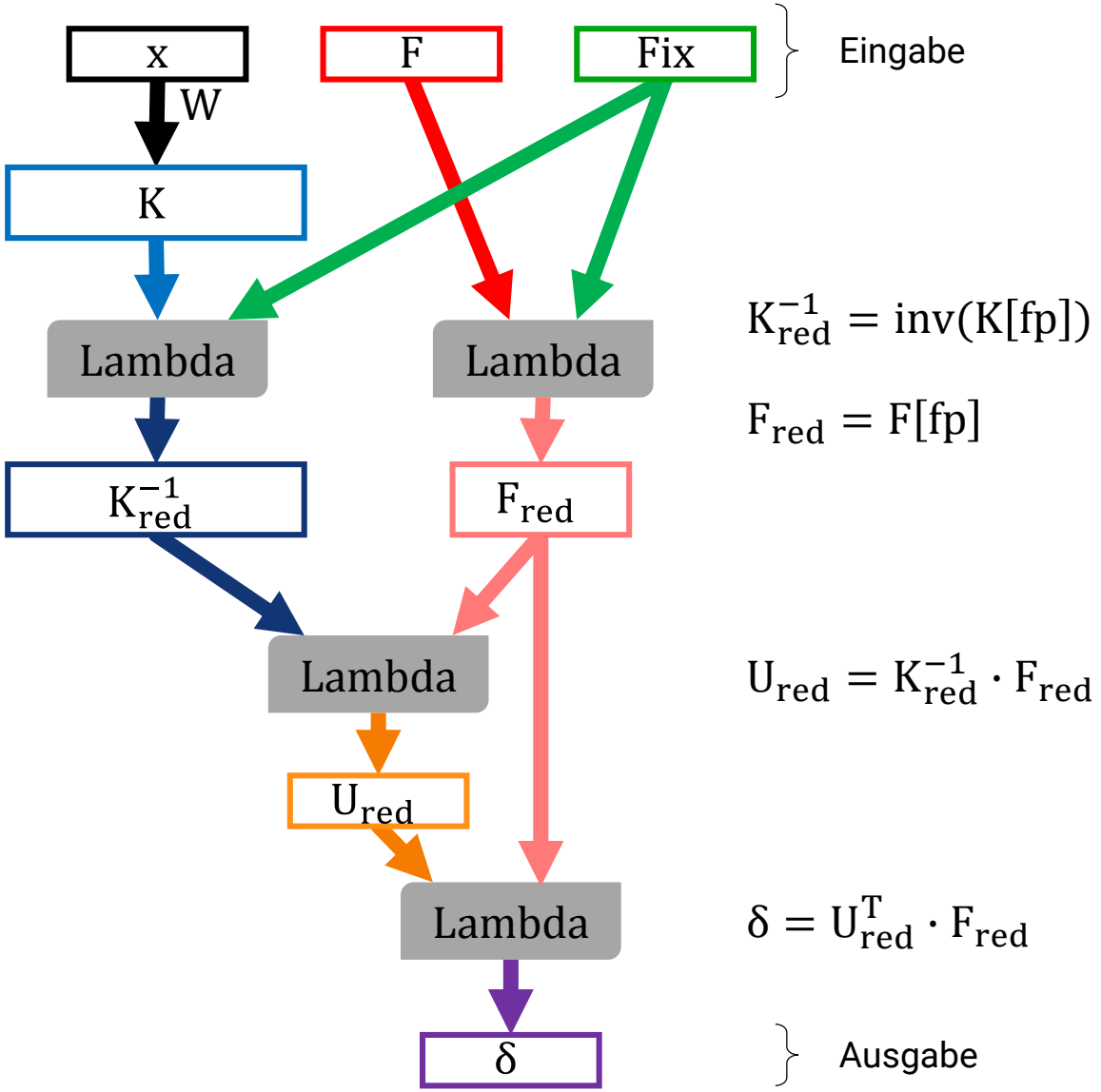
Diskriminator-KNN 3/4

- $K_m = W_{m,n} \cdot x_n$
- $n = d^2 = 10^2 = 100$
- $m = (2(d+1)^2)^2 = 58564$
- $K_1 = \sum_{n=1}^{100} W_{n,1} \cdot x_n$



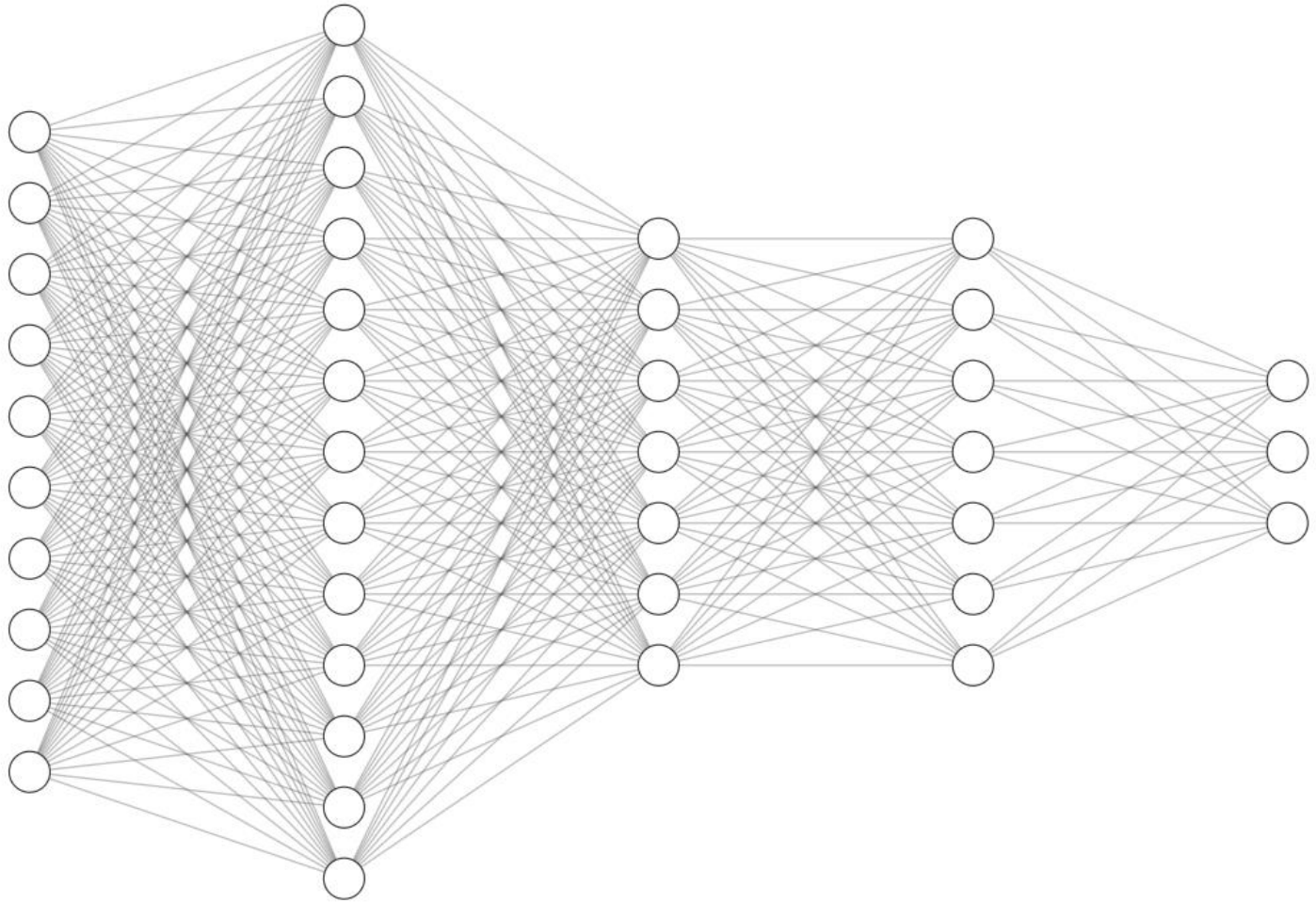
Diskriminator-KNN 4/4

F	Kraftrandbedingung
Fix	Lagerrandbedingung
fp	Freie Punkte
K	Steifigkeit
Lambda	math. def. KNN-Schicht
U	Verschiebungen
δ	Nachgiebigkeit

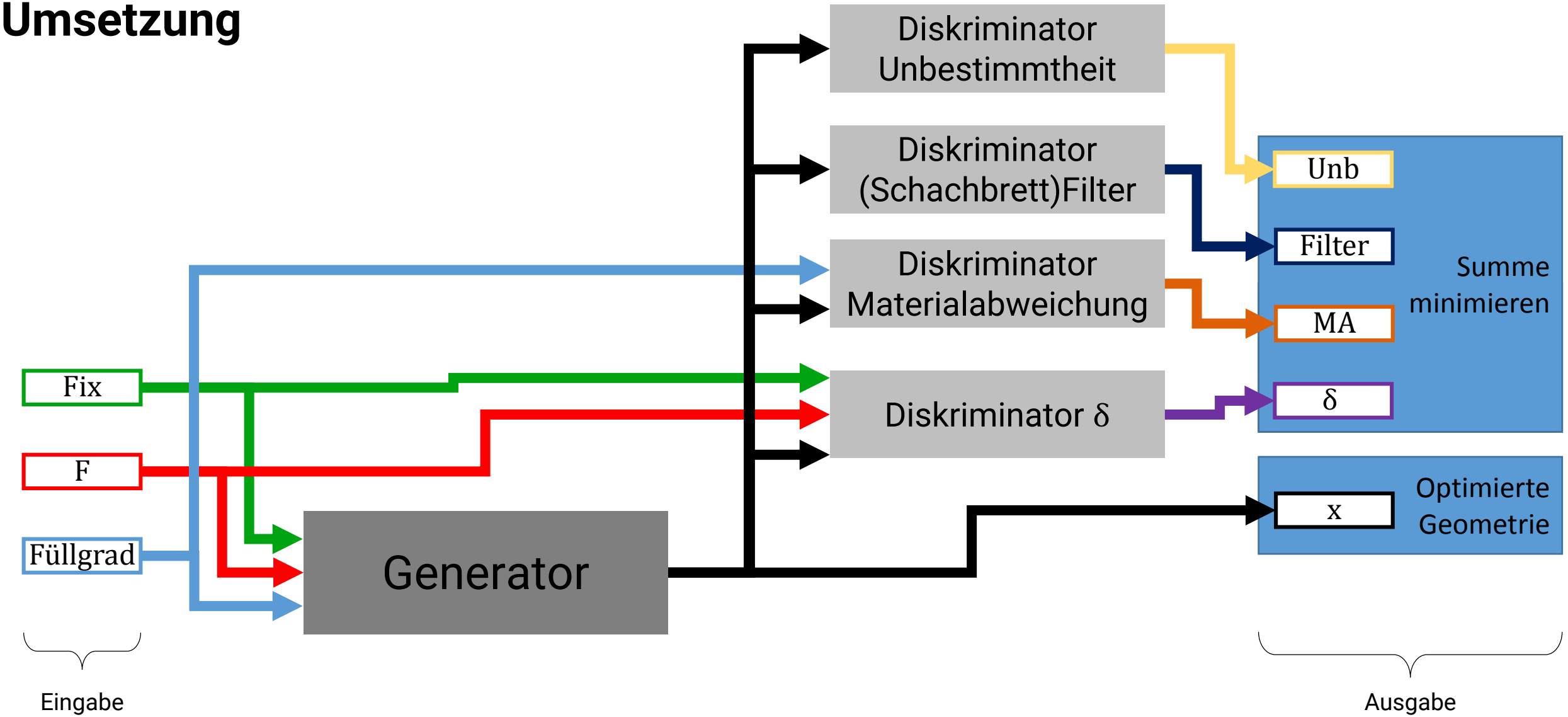


Generator-KNN

- „Simples“ KNN
  - 485 Eingaben
  - Versteckte Schichten:
    - 512, PReLU
    - 256, PReLU
    - 256, PReLU
  - 100 Ausgaben, Sigmoid

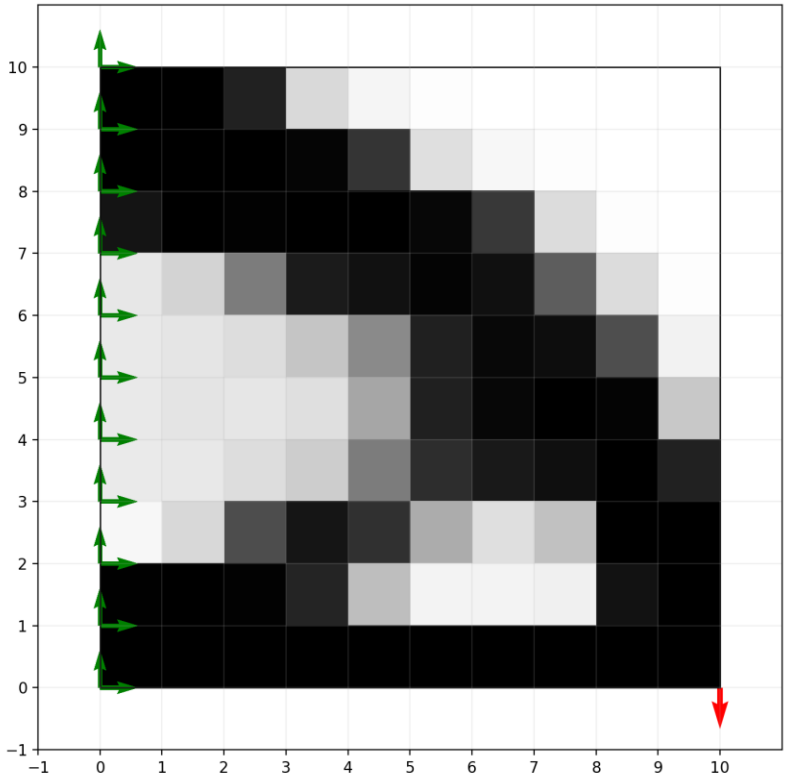


Umsetzung


Diskriminator Unbestimmtheit
Diskriminator (Schachbrett)Filter
Diskriminator Materialabweichung
Diskriminator δ
Unb
Filter
MA
δ
Summe minimieren
x
Optimierte Geometrie
Fix
F
Füllgrad
Generator
Eingabe
Ausgabe

Beispiel: [Nachgiebigkeit + Materialabweichung + Filter + Unbestimmtheit]

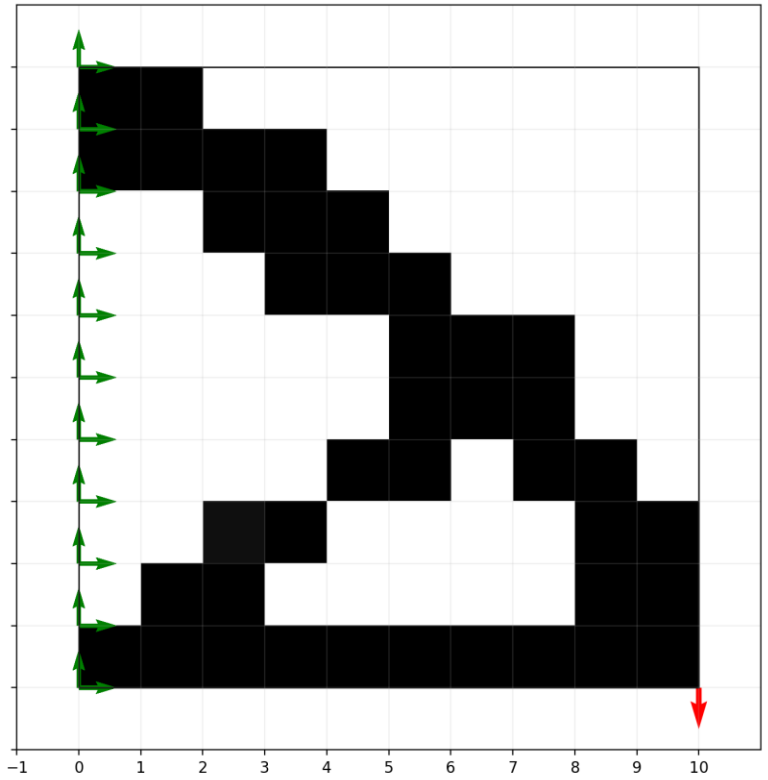
Deep Learning topology optimization



Bewertung	1.096
Nachgiebigkeit	0.794
Materialabweichung	0.164
Unbestimmtheit	0.084
Filter	0.054
Dauer	0.010 *s*

mse	0.130
mae	0.200
$Genauigkeit_{0.1}$	58.0
$Genauigkeit_{0.05}$	45.0
$Genauigkeit_{0.01}$	32.0
$Matanteil_{soll}$	0.400

99 line topology optimization



Bewertung	1.302
Nachgiebigkeit	1.123
Materialabweichung	0.000
Unbestimmtheit	0.000
Filter	0.178
Dauer	1.858 *s*

Zusammenfassung

- Topologieoptimierung mittels Deep Learning kann nahezu optimale Geometrien erzeugen
- Benötigt nur einen Bruchteil Rechenzeit
  - 50x – 350x schneller
- Eröffnet einen neuen Anwendungsbereich im Maschinenbau

Vielen Dank für Ihre Aufmerksamkeit!