

# Künstliche Intelligenz

## Eine Einführung

# Künstliche Intelligenz – Eine Einführung



Was ist Künstliche Intelligenz?



Wie hat sich die Künstliche Intelligenz entwickelt?



Wie funktioniert Künstliche Intelligenz?



Wo findet Künstliche Intelligenz heute Anwendung?

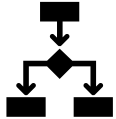


Welche Limitierungen hat Künstliche Intelligenz (aktuell)?



Wo geht die Reise hin und welche Chancen und Risiken gibt es?

# Was ist ein Algorithmus?



Eindeutige Abfolge von Handlungsschritten zur Lösung eines Problems.

```
def maximum(int a, int b):
```

**Wenn**  $a > b$ , dann:

return a

**Ansonsten:**

return b

```
def maxList(int[] liste):
```

tempMax=liste[0]

länge=liste.length()

**Für**  $i=0$  &  $i<\text{länge}$  &  $i++$ :

tempMax=maximum(tempMax, liste[i])

return tempMax

5

2

3

10

13

9

6

3

11

13

7

# Was ist Intelligenz?



Die Fähigkeit, aus Erfahrung zu lernen, Probleme zu lösen und sich an neue Situationen anzupassen. [01]



Mentale Fähigkeiten, die es einer Entität ermöglichen [02]:



(zu lernen) schwere Probleme zu lösen.



erfolgreich in bekannten, unbekannten und dynamischen Umgebungen zu operieren.



abstrakt, intuitiv und rational zu denken und dabei Inkonsistenzen und Widersprüche zu vermeiden,



das eigene Handeln zu reflektieren und entsprechend übergeordneter Ziele und Normen anzupassen.



mit anderen Entitäten sozial zu interagieren und die eigenen Werte und Normen an einem höheren Ziel auszurichten.

# Was ist Künstliche Intelligenz?



Die Wissenschaft Computer zu ermächtigen Probleme zu lösen, die aktuell besser von Menschen gelöst werden können. [03]

## Schwache Künstliche Intelligenz:

Computerprogramm, dass Menschen bei der Lösung **bestimmter Klassen von Problemen** überlegen ist.

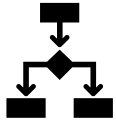
## Starke Künstliche Intelligenz:

Computerprogramm, dass Menschen bei der Lösung **aller denkbaren Probleme** ebenbürtig oder überlegen ist.

# Schwache Künstliche Intelligenz



# Symbolische



Logik und Entscheidungsregeln werden vorgegeben.



„Wissen“ wird explizit vorgeben und gespeichert.



*„Alles ist berechenbar.“*

# & Subsymbolische KI



Zielstellung und Erfolgskriterien werden vorgegeben.



Algorithmus „lernt“ mithilfe von Daten.



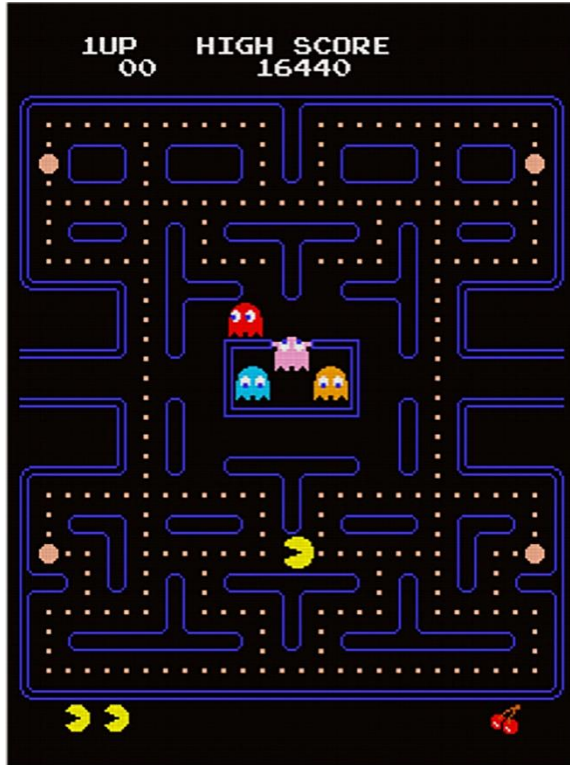
*„Alles ist klassifizierbar.“*

Alles ist berechenbar:

# **SYMBOLISCHE KÜNSTLICHE INTELLIGENZ**



# Spiele KI als Beispiel Symbolischer KI [04]



Geister wechseln zwischen 3 verschiedenen Zuständen: Patrouille, Jagen und Fliehen.



Jeder Geist hat eine eigene „Persönlichkeit“ bestimmt durch sein Jagdverhalten.



Die aktuelle Position Pacmans als Ziel.



4 Felder in Pacmans Blickrichtung als Ziel.



Versucht Pacman mithilfe von Rot einzukesseln.



Jagt Pacman nur, wenn er in der Nähe ist.

# Spiele KI als Beispiel Symbolischer KI [04]



Jemand hat sich vorher ausgedacht, wie genau sich die KI verhalten soll.



Das Verhalten ist zu jedem Zeitpunkt anhand der Logik nachvollziehbar.



Fehler im Verhalten sind einfach zu lokalisieren und zu beheben.

# Akinator als Expertensystem



Ist deine Figur real?

Ist deine Figur weiblich?

Kennt man deine Figur aus einem Film?

Ist deine Figur der Hauptcharakter?

Ist deine Figur älter als 50 Jahre?

Ist deine Figur ein Schuldirektor?

Ist deine Figur aus Harry Potter?

Ist deine Figur Albus Dumbledore?

# Frühe Beispiele: Expertensysteme [05]



Expertensysteme als große Hoffnungsträger der KI in den 1970er – 1980er Jahren.



Wissensdatenbanken (oft in Form von Wenn-Dann-Beziehungen) zur Unterstützung von Entscheidungen.

## Beispiel: CADUCEUS

- Diente zur Diagnose von Krankheiten
- Jedes Symptom war mit bestimmten Krankheiten assoziiert
- Nutzer gab Liste von Symptomen an
- CADUCEUS diagnostizierte öfter die richtige Krankheit als Experten.



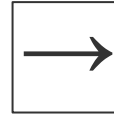
# Automatisches Beweisen



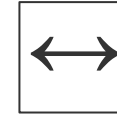
UND



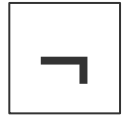
ODER



IMPLIZIERT



ÄQUIVALENZ

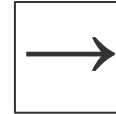


NICHT

1 Es regnet gerade

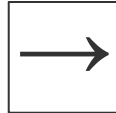


Die Temperatur ist  
zwischen -1°C und +1°C



Die Straße ist glatt

2 Die Straße ist  
glatt

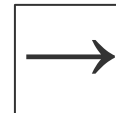


Fahre nicht schneller als  
30 km/h

! Es regnet gerade

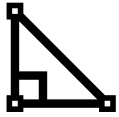


Die Temperatur ist  
zwischen -1°C und +1°C



Fahre nicht schneller als  
30 km/h

# Das Bi-Pythagoreische Triple Problem [06]



Pythagoreisches Triple:  $a^2 + b^2 = c^2$  |  $3^2 + 4^2 = 5^2$  |  $5^2 + 12^2 = 13^2$



Kann man die Zahlen von  $1 \dots N$  so in 2 Untergruppen aufteilen, dass in keiner Untergruppe ein Pythagoreisches Triple enthalten ist?



Bis 2016 ein ungelöstes mathematisches Problem.



Zahlen  $1 \dots 7825$  durchzuprobieren  $\rightarrow 2^{7825}$  Möglichkeiten = ca.  $3.63 \cdot 10^{2355}$   
(Im Universum gibt es  $10^{100}$  Teilchen).

# Die Lösung mittels automatischen Beweiser [06]



Führe für jede Zahl  $1 \dots N$  eine bool'sche Variable  $X_1, X_2, X_3 \dots X_N$  ein.  
Wahrheitswert gibt an, in welcher Gruppe die Zahl ist.

Bsp.: Zahlen  $1 \dots 10$ : Enthalten nur ein Pythagoreisches Tripel: 3;4;5  
D.h.  $X_3 = T, X_4 = T, X_5 = F$  ist eine gültige Belegung.



Für jedes Pythagoreisches Trippel muss gelten:  
 $(X_A \vee X_B \vee X_C) \wedge (\neg X_A \vee \neg X_B \vee \neg X_C)$



Automatischer Beweis: Generiere für alle Pythagoreischen Tripel in  $1 \dots N$   
die entsprechende boolsche Bedingung und errechne passende Belegung.

# Das Ergebnis [06]



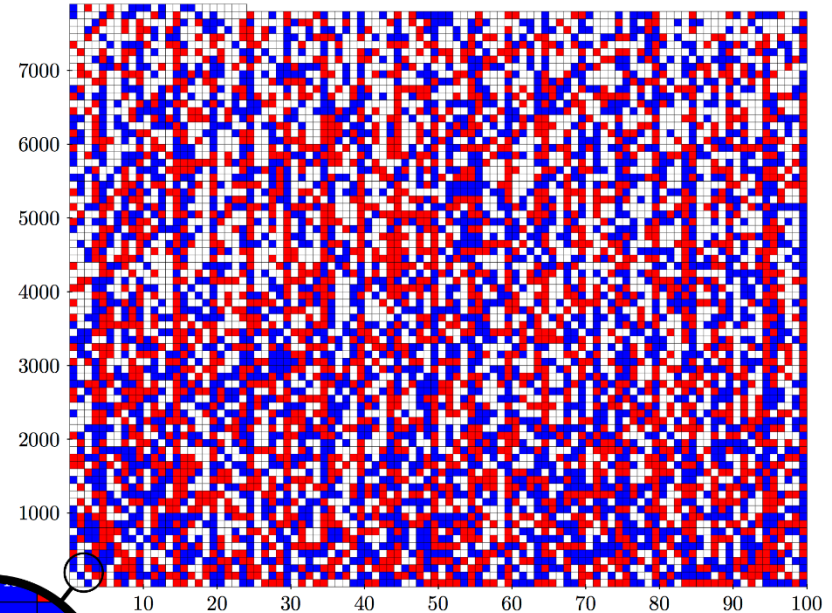
200 Terrabyte langer Beweis.



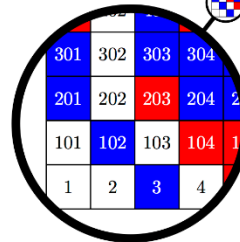
Für die Zahlen 1...7824 ist eine Bi-Partitionierung möglich.



Ab 1...7825 ist keine Bi-Partitionierung möglich.



Partitionierung für 1...7824





# Das Ergebnis [06]



200 Terrabyte langer Beweis.



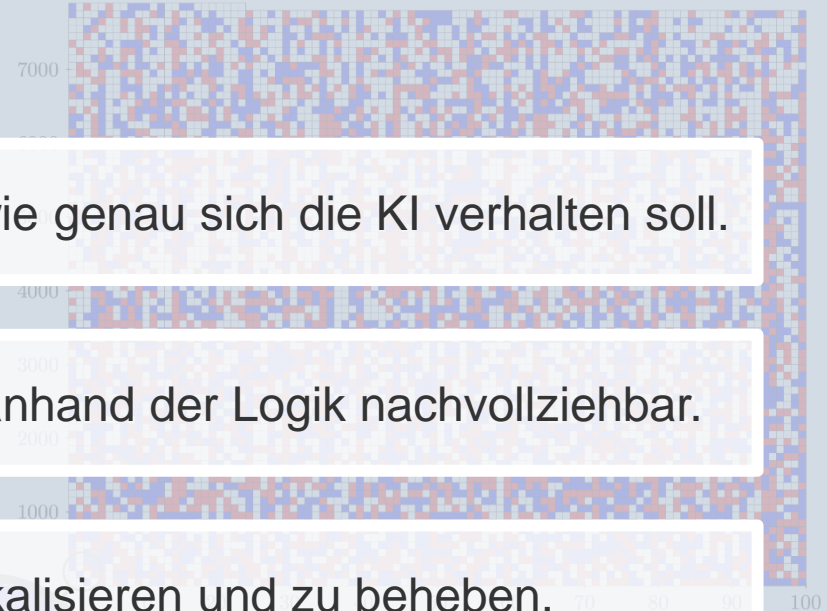
Jemand hat sich vorher ausgedacht, wie genau sich die KI verhalten soll.



Das Verhalten ist zu jedem Zeitpunkt anhand der Logik nachvollziehbar.



Fehler im Verhalten sind einfach zu lokalisieren und zu beheben.



Alles ist klassifizierbar:

# **SUBSYMBOLISCHE KÜNSTLICHE INTELLIGENZ**

# Chihuahua in Bildern erkennen:



4 Beine



Ziemlich klein



Spitze Ohren



Große schwarze  
Augen



Schwarze Stupsnase

# Unterscheidung: Chihuahua oder Muffin?



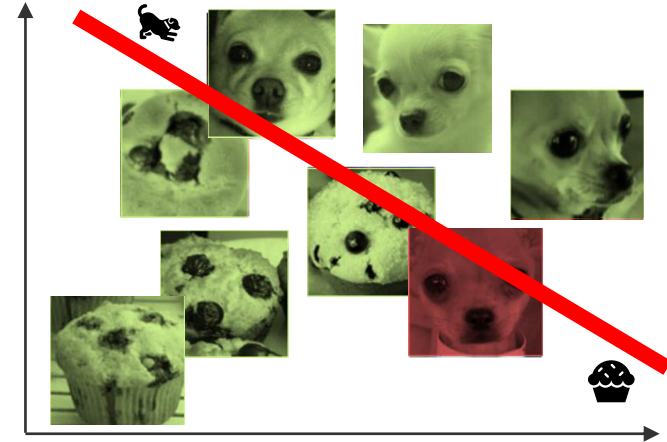
# Mustererkennung als Ursprung



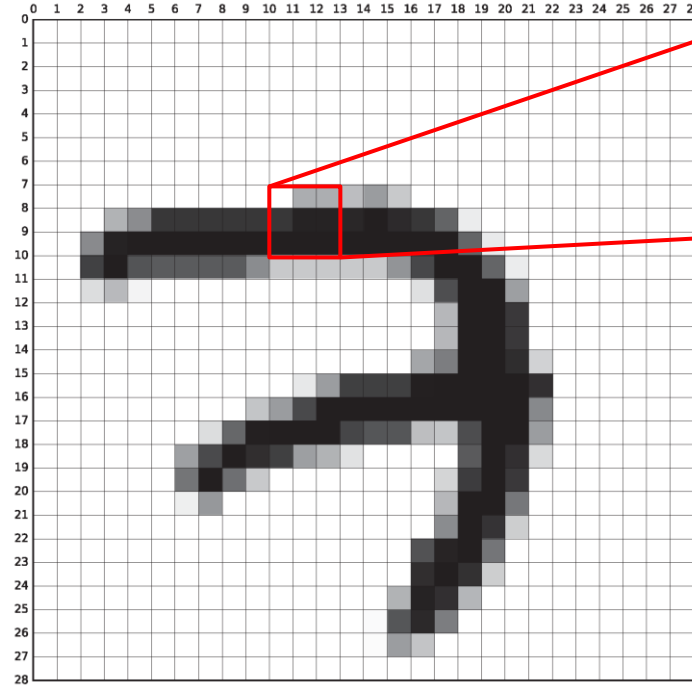
Ursprünge des Maschinellen Lernens liegen in der Mustererkennung:  
Anwendung von statistischen Methoden auf Daten.



Unterscheide Klassen anhand ihrer gewichteten Merkmale.



# Eigenschaften bei der Bildklassifizierung [07]



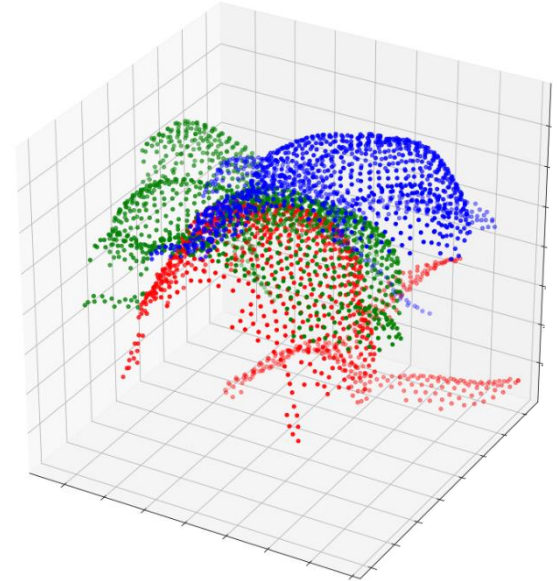
0,0	0,5	0,5
0,7	0,8	0,8
0,9	0,9	0,9

$$\begin{pmatrix} x_{1,1} & \cdots & x_{1,28} \\ \vdots & \ddots & \vdots \\ x_{28,1} & \cdots & x_{28,28} \end{pmatrix}$$

# Klassifizierung: 784-Dimensionalen Vektorraum

$$\left. \begin{array}{l} \boxed{2} \begin{pmatrix} x_{1,1} & \cdots & x_{1,28} \\ \vdots & \ddots & \vdots \\ x_{28,1} & \cdots & x_{28,28} \end{pmatrix} \\ \boxed{3} \begin{pmatrix} y_{1,1} & \cdots & y_{1,28} \\ \vdots & \ddots & \vdots \\ y_{28,1} & \cdots & y_{28,28} \end{pmatrix} \\ \boxed{4} \begin{pmatrix} z_{1,1} & \cdots & z_{1,28} \\ \vdots & \ddots & \vdots \\ z_{28,1} & \cdots & z_{28,28} \end{pmatrix} \end{array} \right\}$$

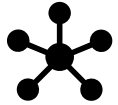
Jede Zahl wird hier durch  $28 \times 28 = 784$ -Dimensionen definiert.



Ziel jedes Mustererkennungsansatzes: Ziehe trennende Hyperebenen zwischen den Klassen ein, dass so viele Elemente wie möglich richtig klassifiziert werden.



# Wie funktioniert ein Neuronales Netz?



Ein Neuronales Netz besteht aus unterschiedlichen Arten von Neuronen:



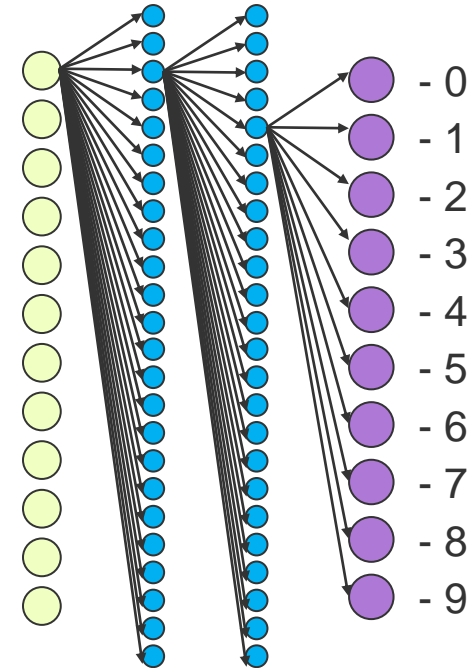
Neuronen des Input-Layers: Lesen die Eingabe und leiten diese weiter.



Neuronen des Hidden-Layers: Erkennen Muster innerhalb der Daten.

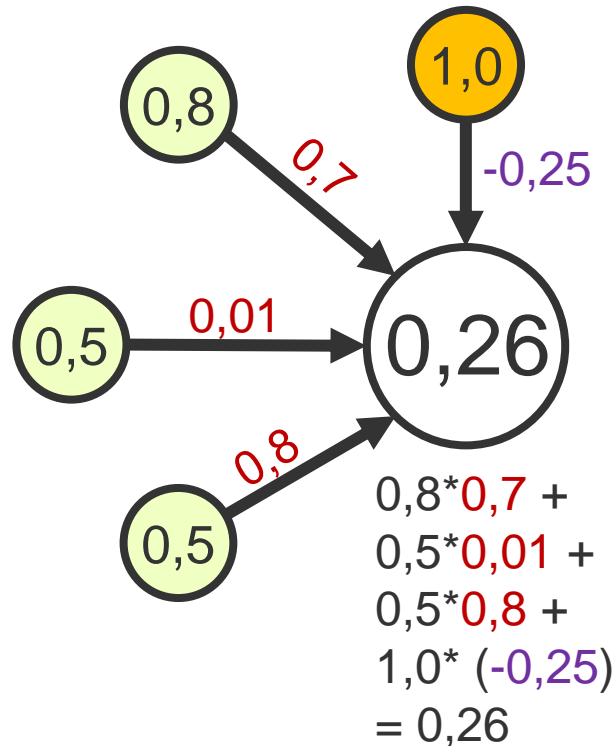


Neuronen des Output-Layers: Spiegeln das Ergebnis der Klassifizierung wieder.

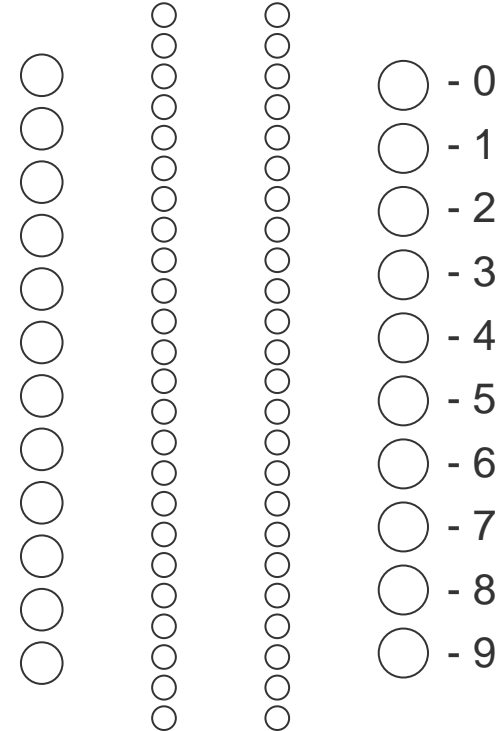
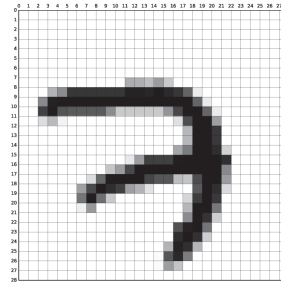




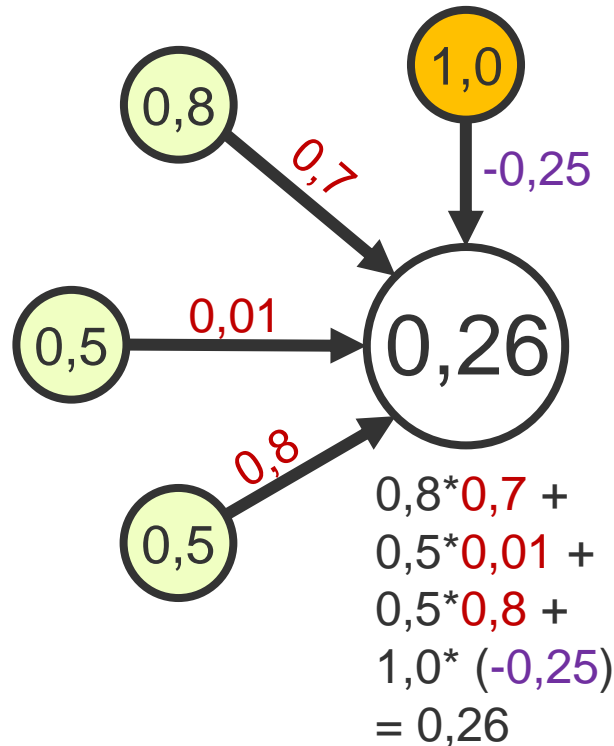
# Wie funktioniert ein Neuron? [08]



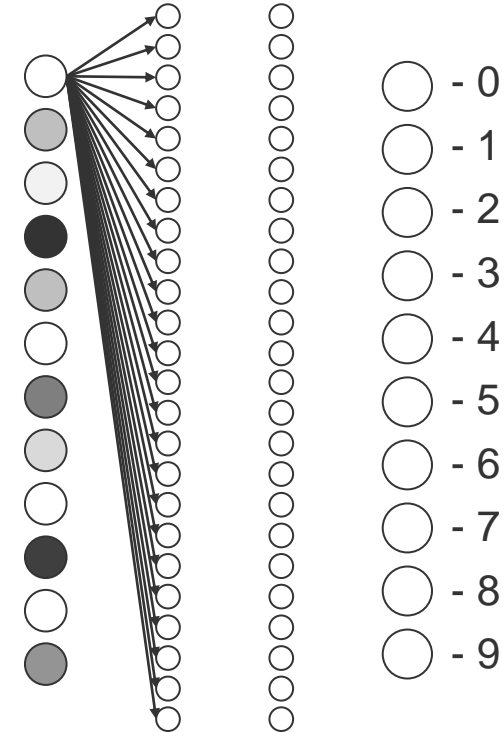
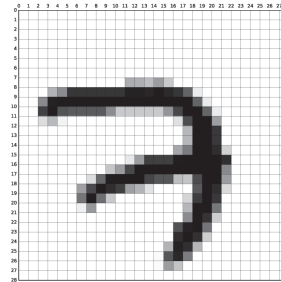
$$\begin{pmatrix} x_{1,1} & \cdots & x_{1,28} \\ \vdots & \ddots & \vdots \\ x_{28,1} & \cdots & x_{28,28} \end{pmatrix}$$



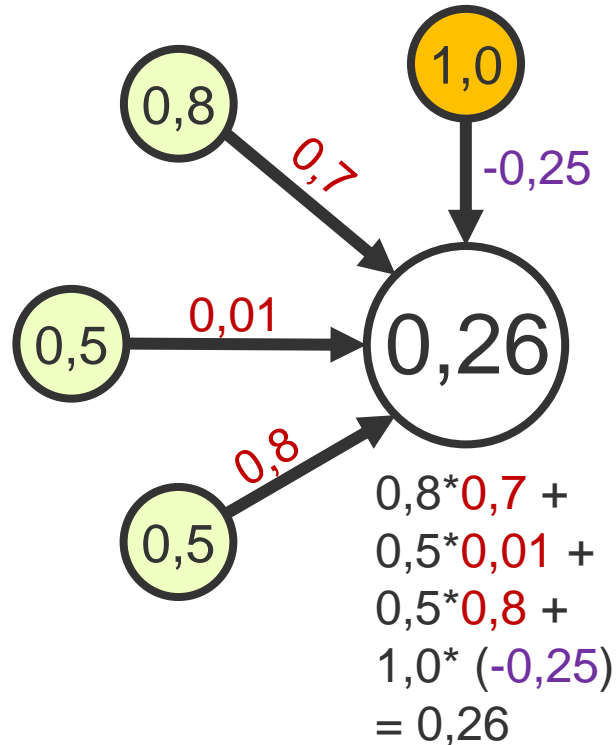
# Wie funktioniert ein Neuron? [08]



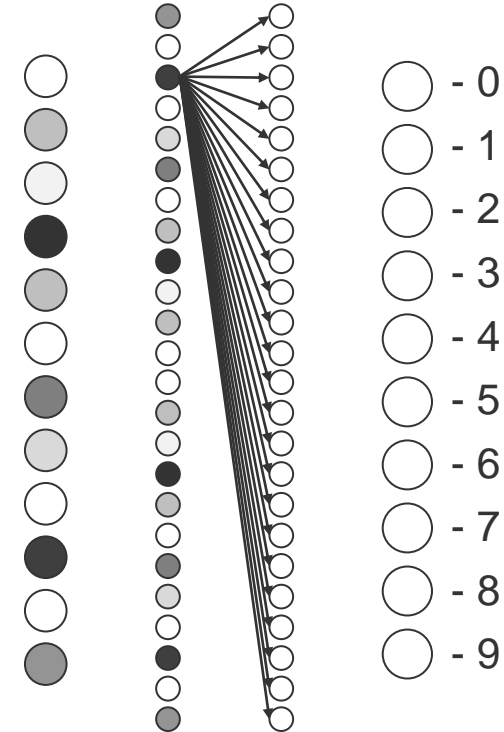
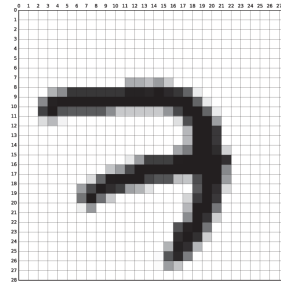
$$\begin{pmatrix} x_{1,1} & \cdots & x_{1,28} \\ \vdots & \ddots & \vdots \\ x_{28,1} & \cdots & x_{28,28} \end{pmatrix}$$



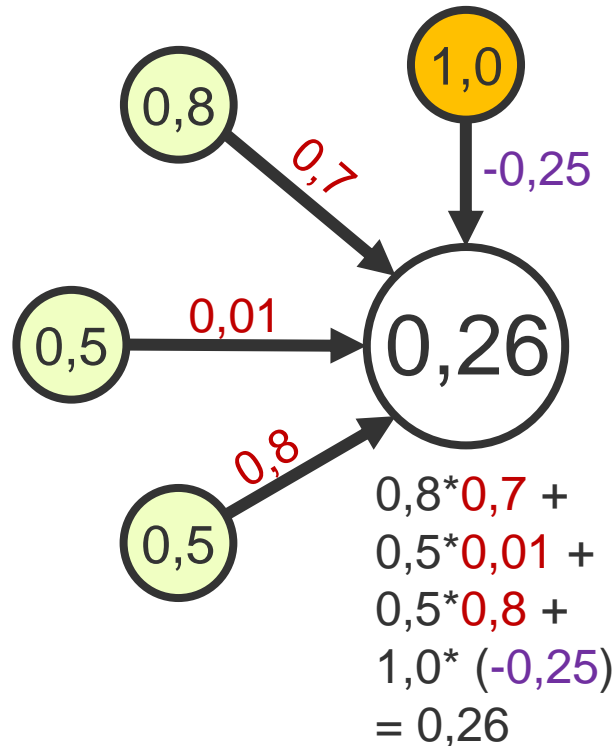
# Wie funktioniert ein Neuron? [08]



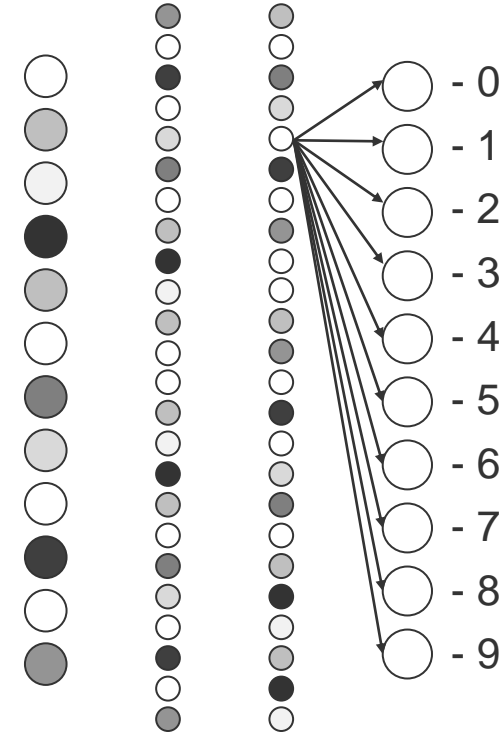
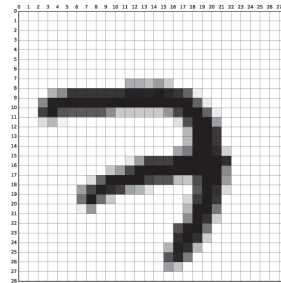
$$\begin{pmatrix} x_{1,1} & \cdots & x_{1,28} \\ \vdots & \ddots & \vdots \\ x_{28,1} & \cdots & x_{28,28} \end{pmatrix}$$



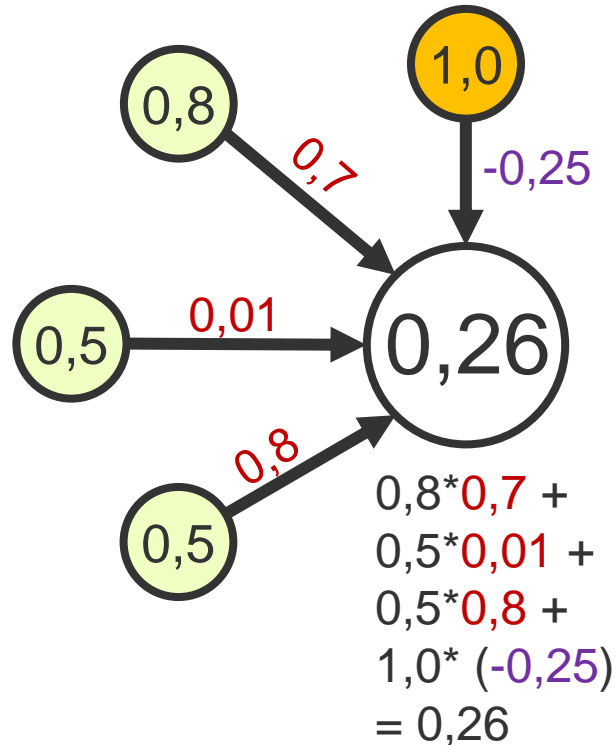
# Wie funktioniert ein Neuron? [08]



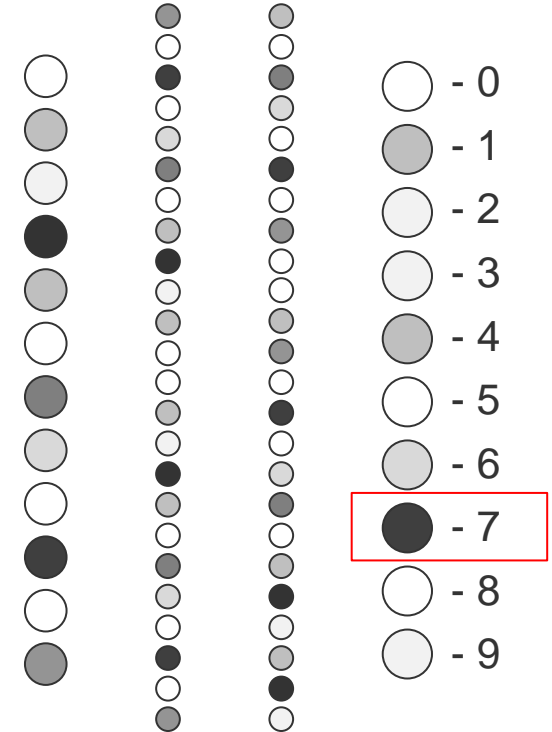
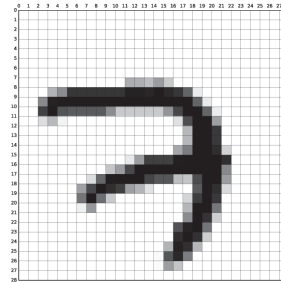
$$\begin{pmatrix} x_{1,1} & \cdots & x_{1,28} \\ \vdots & \ddots & \vdots \\ x_{28,1} & \cdots & x_{28,28} \end{pmatrix}$$



# Wie funktioniert ein Neuron? [08]



$$\begin{pmatrix} x_{1,1} & \cdots & x_{1,28} \\ \vdots & \ddots & \vdots \\ x_{28,1} & \cdots & x_{28,28} \end{pmatrix}$$



# Parameter eines Neuronalen Netzes [08]



Input-Layer: Bilder mit  $28 * 28 = 784$  Pixel  $\rightarrow$  **784 Neuronen**

Hidden-Layer: 2 Layer á 16 Neuronen = **32 Neuronen**

Output-Layer: **10 Neuronen**



Gewichte:  $784 * 16 + 16 * 16 + 16 * 10 =$  **12.960 Gewichte**

Biases:  $16 + 16 + 10 =$  **42 Biases**



Alternative Rechnung (mit einem Bias-Neuron pro Schicht):

$(784 + 1) * 16 + (16 + 1) * 16 + (16 + 1) * 10 =$  **13.002 Parameter**



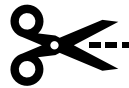
Wie werden diese Parameter bestimmt?

# Wie lernt ein Neuronales Netz?

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



Starten mit **gelabeltem** Datensatz und initialisieren alle Gewichte & Biases **zufällig**.



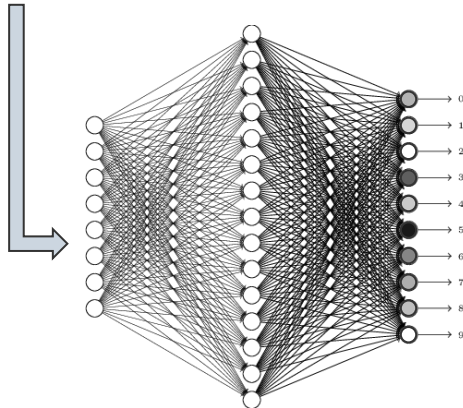
Unterteilung des Datensatzes in Trainings- und Testmenge.



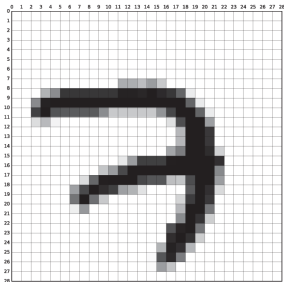
Erster Durchgang mit der Trainingsmenge: Zufällige Ergebnisse (das Netz rät).























Aufstellen einer Fehlerfunktion (Cost function)



# Die Fehlerfunktion



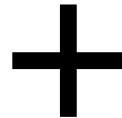
 - 0	 - 0
 - 1	 - 1
 - 2	 - 2
 - 3	 - 3
 - 4	 - 4
 - 5	 - 5
 - 6	 - 6
 - 7	 - 7
 - 8	 - 8
 - 9	 - 9



Die Fehlerfunktion gibt an wie weit wir vom richtigen Ergebnis entfernt sind.



Für jeden Output-Neuron:  
 $(\text{Ergebnis} - \text{Gewünschtes Ergebnis})^2$



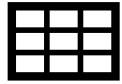
Aufaddierte quadrierte Distanz für alle  
 Output-Neuron & alle Trainingsdaten



Zielstellung: Minimierung der  
 Fehlerfunktion (finden des Tiefpunkts)



# Fehlerfunktion und Backpropagation



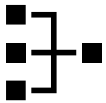
Unsere Fehlerfunktion besteht aus 13.002 Variablen (Gewichte des NN)



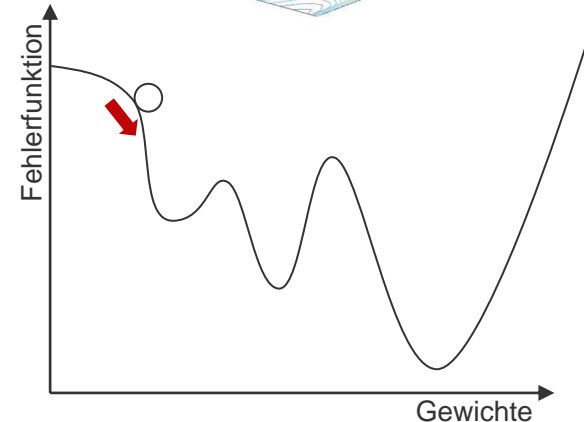
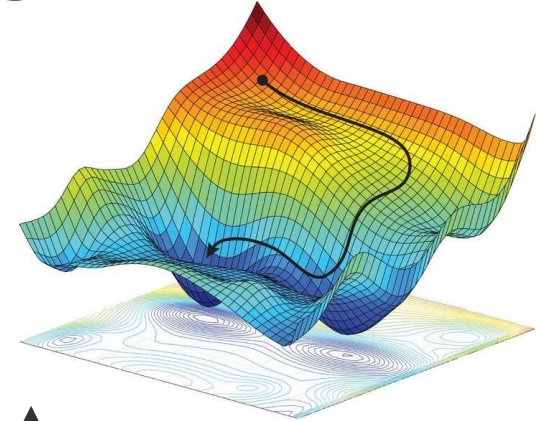
Direkte Berechnung des Tiefpunkts ist nicht möglich.



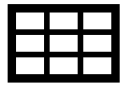
Iterative Backpropagation mittels gradient descent.



Zurückrechnen des Fehlers und Anpassen der Gewichte entsprechend der Lernrate.



# Fehlerfunktion und Backpropagation



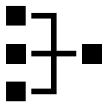
Unsere Fehlerfunktion besteht aus 13.002 Variablen (Gewichte des NN)



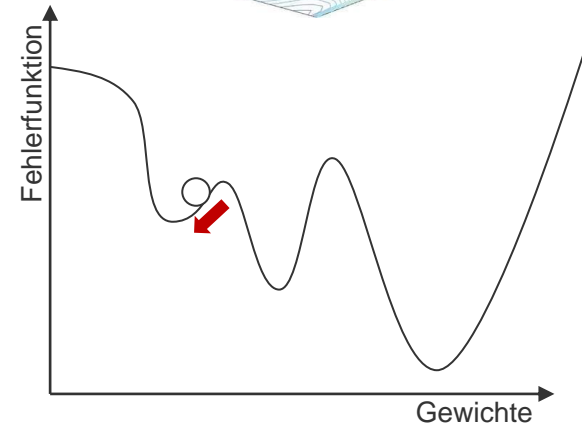
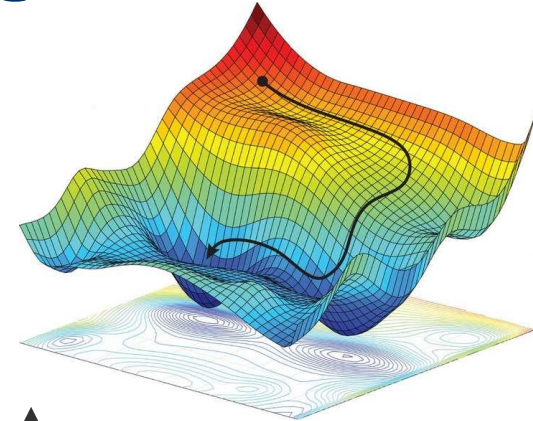
Direkte Berechnung des Tiefpunkts ist nicht möglich.



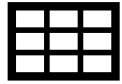
Iterative Backpropagation mittels gradient descent.



Zurückrechnen des Fehlers und Anpassen der Gewichte entsprechend der Lernrate.



# Fehlerfunktion und Backpropagation



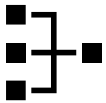
Unsere Fehlerfunktion besteht aus 13.002 Variablen (Gewichte des NN)



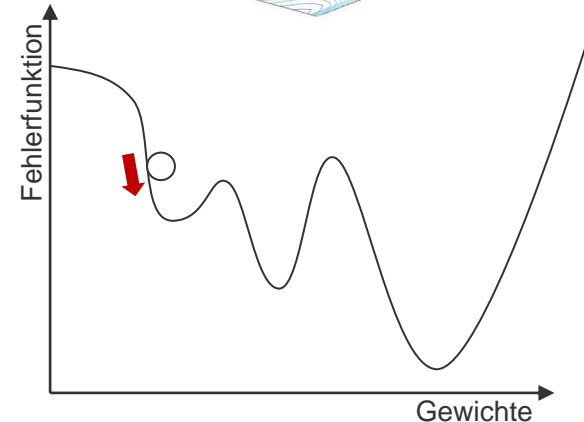
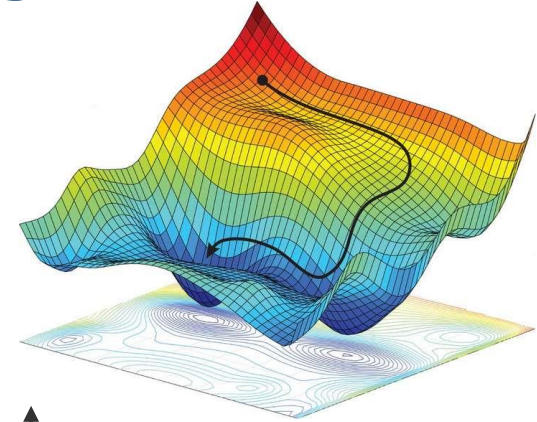
Direkte Berechnung des Tiefpunkts ist nicht möglich.



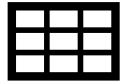
Iterative Backpropagation mittels gradient descent.



Zurückrechnen des Fehlers und Anpassen der Gewichte entsprechend der Lernrate.



# Fehlerfunktion und Backpropagation



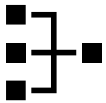
Unsere Fehlerfunktion besteht aus 13.002 Variablen (Gewichte des NN)



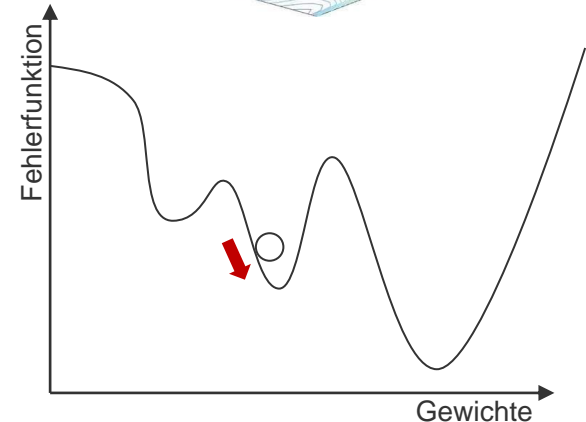
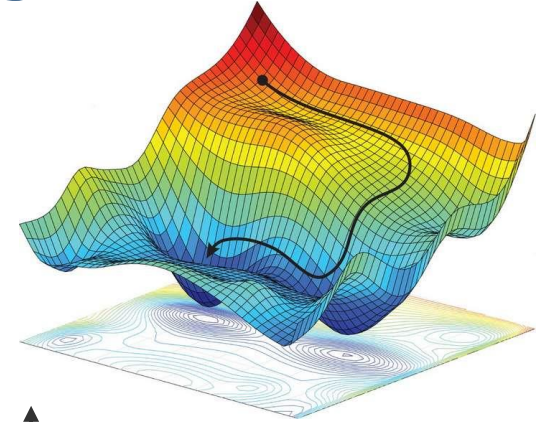
Direkte Berechnung des Tiefpunkts ist nicht möglich.



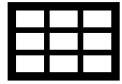
Iterative Backpropagation mittels gradient descent.



Zurückrechnen des Fehlers und Anpassen der Gewichte entsprechend der Lernrate.



# Fehlerfunktion und Backpropagation



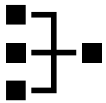
Unsere Fehlerfunktion besteht aus 13.002 Variablen (Gewichte des NN)



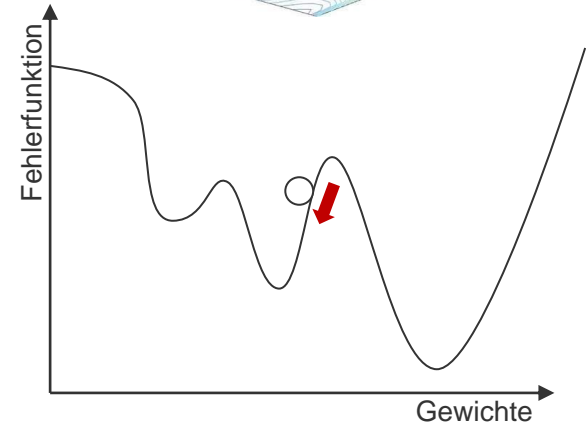
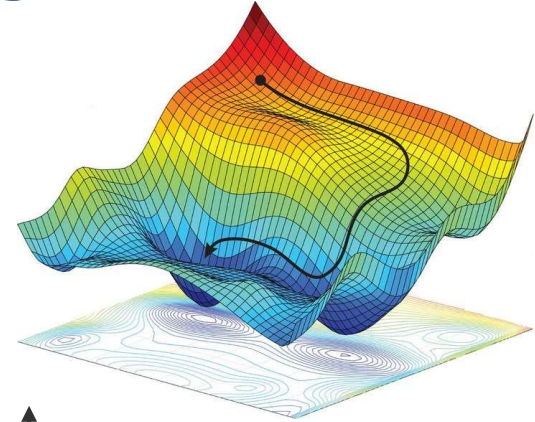
Direkte Berechnung des Tiefpunkts ist nicht möglich.



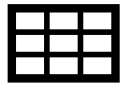
Iterative Backpropagation mittels gradient descent.



Zurückrechnen des Fehlers und Anpassen der Gewichte entsprechend der Lernrate.



# Fehlerfunktion und Backpropagation



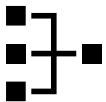
Unsere Fehlerfunktion besteht aus 13.002 Variablen (Gewichte des NN)



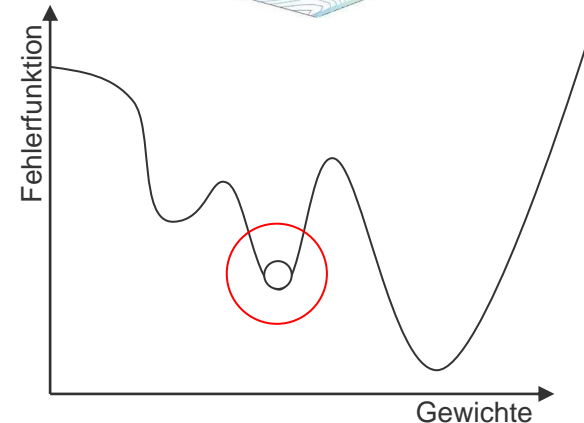
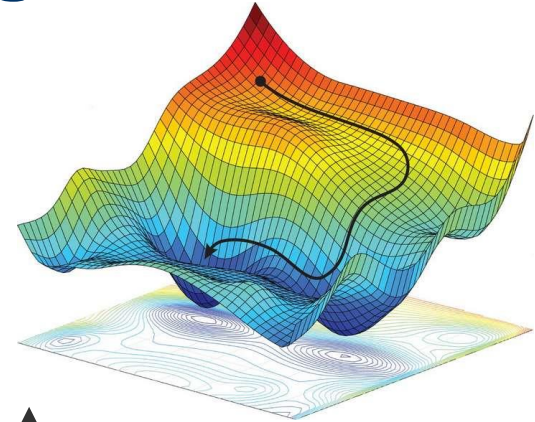
Direkte Berechnung des Tiefpunkts ist nicht möglich.



Iterative Backpropagation mittels gradient descent.



Zurückrechnen des Fehlers und Anpassen der Gewichte entsprechend der Lernrate.





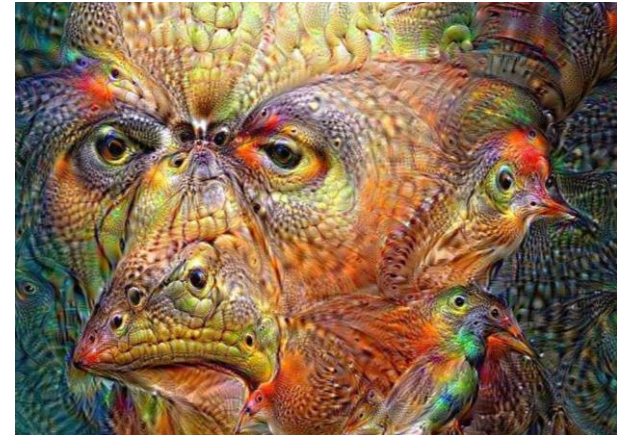
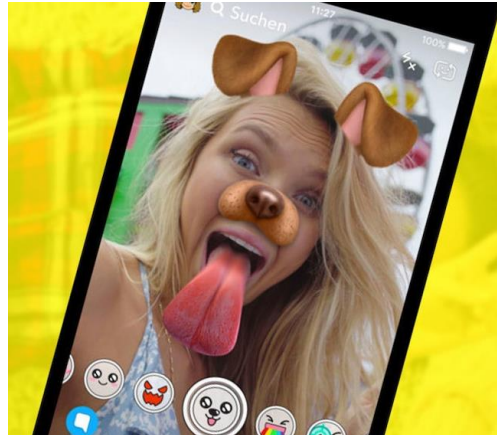
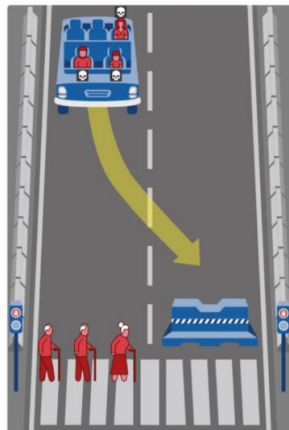
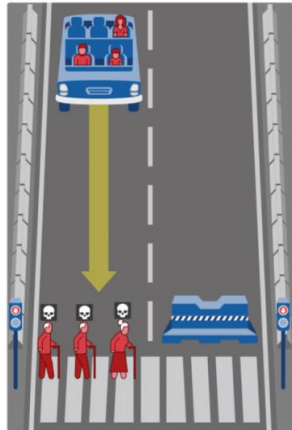
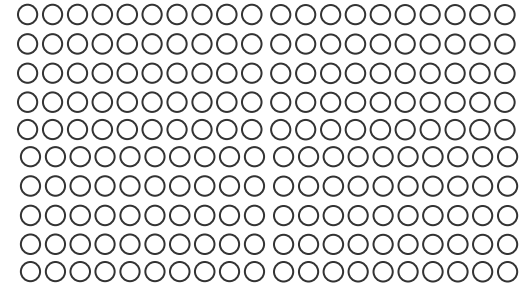
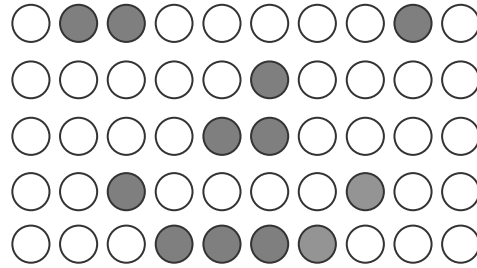
# Alles ist ein Klassifizierungsproblem



Geradeaus

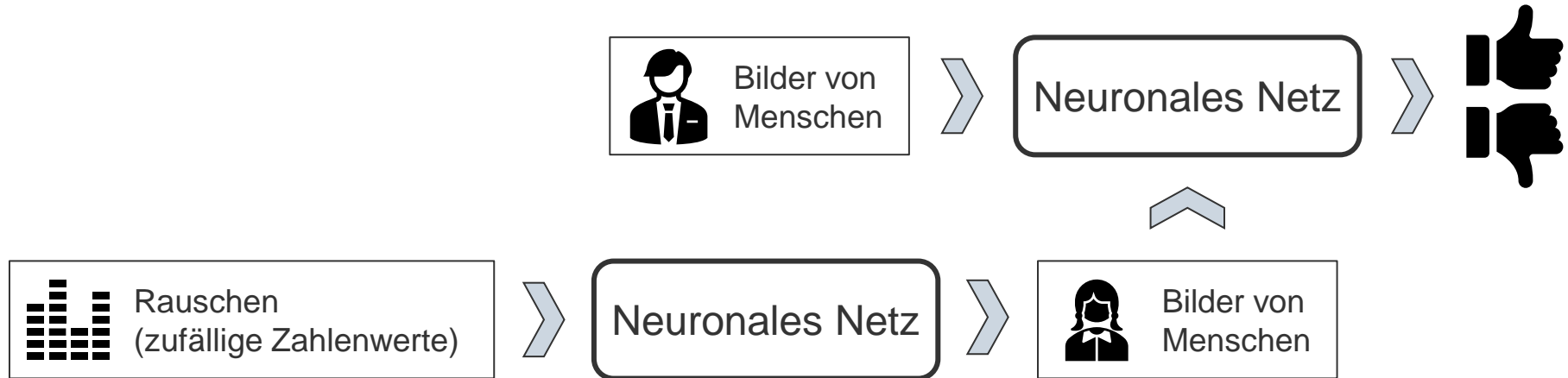


Ausweichen



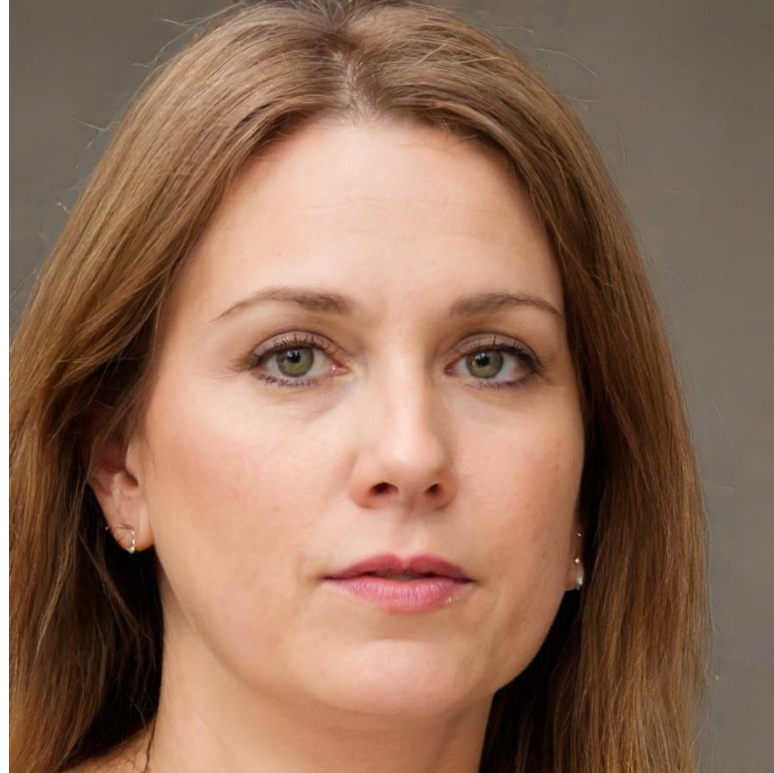
# Wie generieren Neuronale Netze Kunst?

? Wie können Neuronale Netze lernen, wenn es kein richtig und falsch gibt?





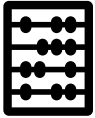
<https://thispersondoesnotexist.com/>



<https://thispersondoesnotexist.com/>



# Wie funktioniert ChatGPT? (im Ansatz) [09]



Neurales Netz mit **175 Mrd. Parametern.** (GPT3)



Eingabe (Fließtext) wird **unterteilt in Tokens.** Tokens = Silben / Wörter.

This sentence is divided into tokens.



**Trainingsdaten:** Ca. 680 Mrd. Tokens von Texten aus dem Internet.

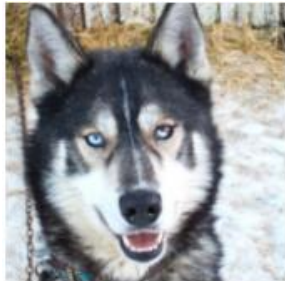
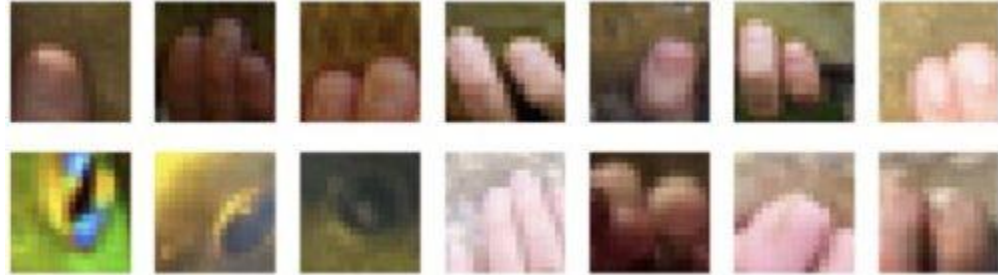


**Zielstellung:** Gegeben X Tokens → Sage das nächste Token X+1 vorher.  
Es wird auf Glaubwürdigkeit trainiert, nicht „Wahrheit“.

# Machine Learning als Black-Box [10, 11]



Im Gegensatz zu Symbolischer KI, werden bei Maschinellern Lernen keine (Klassifizierungs-) Regeln vorgegeben.





# Weitere Probleme von Maschinellen Lernen [12]



**Overfitting:** Trainingsdaten werden auswendig gelernt.



**Bias** in Daten führt zu Bias des Systems.



**Overconfidence:** Maschinelles Lernen kennt kein „Ich weiß es nicht“.



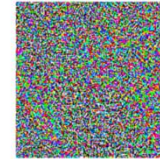
Maschinelles Lernen „ist“ Statistik und deshalb immer **nur so gut wie die Daten.**



“panda”

57.7% confidence

+ .007 ×



noise

=



“gibbon”

99.3% confidence



# Weitere Risiken der KI-Technologie



**Massenüberwachung:** Es braucht keine Menschen mehr um Chats, Internetaktivitäten, Kameras, Buchungen etc. zu überwachen.



**(Wahl-)Manipulation:** Perfekt auf die eigenen Weltanschauungen zugeschnittene Wahlwerbung.

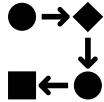


**Autonome Waffen:** Entscheidung zu töten wird automatisch getroffen.



**Monopolbildung:** Nur wenige Unternehmen (Google, Meta, Microsoft, etc.) besitzen die Daten für Entwicklung leistungsfähiger KI-Systeme.

# Künstliche Intelligenz und der Arbeitsmarkt



Routine & regelbasierte Aufgaben werden verstärkt von KI übernommen.



Der Mensch als Prüfer und nicht mehr Erzeuger von Inhalten.



Generieren kreativer Vorschläge durch maschinelles Lernen.



Was bleibt: soziale Arbeit, Entscheidungen treffen, investigative Recherche/Forschung, Handwerk, Hilfsarbeiten

# Zusammenfassung & Ausblick



Künstliche Intelligenz ist ein altes Konzept, weit verbreitet und mehr als Maschinelles Lernen.



Maschinelles Lernen ist „reine“ Statistik und lineare Algebra (Vektorrechnung).



Weit entfernt von einer starken künstlichen Intelligenz. Trotzdem braucht es kritische Betrachtung der Entwicklung.



„Warum lernen wir in der Schule nicht wie die Steuererklärung funktioniert?“



Vielen Dank!  
**FRAGEN?**



# Quellen

- [01] **Myers, David G.** *Psychologie*. Springer-Verlag, 2015.
- [02] **Benzmüller, Christoph.** "Symbolic AI and Gödel's Ontological Argument." (2022): 1-9.
- [03] **Rich, Elaine.** *Artificial intelligence*. McGraw-Hill, Inc., 1983.
- [04] [GameInternals - Understanding Pac-Man Ghost Behavior](#)
- [05] **Batson, Eric.** "Computer as consultant: Application of artificial intelligence in diagnosis." *Postgraduate Medicine* 75.2 (1984): 211-214.
- [06] **Heule, Marijn JH, Oliver Kullmann, and Victor W. Marek.** "Solving and verifying the boolean pythagorean triples problem via cube-and-conquer." *International Conference on Theory and Applications of Satisfiability Testing*. Springer, Cham, 2016.
- [07] **Baldominos, Alejandro, Yago Saez, and Pedro Isasi.** "A survey of handwritten character recognition with mnist and emnist." *Applied Sciences* 9.15 (2019): 3169.
- [08] [Aber was \\*ist\\* nun ein neuronales Netzwerk? | Teil 1, Deep Learning – YouTube](#)
- [09] [What are tokens and how to count them? | OpenAI Help Center](#)
- [10] [When data is messy - AI WeirdnessCommentShareCommentShare](#)
- [11] **Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin.** "" Why should i trust you?" Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016.
- [12] **Kolleck, Alma, and Carsten Orwat.** *Mögliche Diskriminierung durch algorithmische Entscheidungssysteme und maschinelles Lernen: ein Überblick*. Büro für Technikfolgen-Abschätzung beim Deutschen Bundestag (TAB), 2020.

# Bildquellen

[The King Competition Schachcomputer | Millennium Chess \(computerchess.com\)](#)

[Was ist Gesichtserkennung und wie funktioniert sie? \(dormakaba.com\)](#)

[Ghosts - SmashWiki, the Super Smash Bros. wiki \(ssbwiki.com\)](#)

[Expert system – Wikipedia](#)

[Akinator](#)

[karen zack auf Twitter: „chihuahua or muffin ? https://t.co/LzZ1lwoVrP“ / Twitter](#)

[Chihuahua \(Hunderasse\) - der kleinste Hund der Welt \(diehundezeitung.com\)](#)

[ICP for point cloud alignment — Pyoints 0.2.0 documentation \(laempy.github.io\)](#)

[Snapchat-Effekte: So nutzen Sie sie - COMPUTER BILD](#)

[Vincent Van Gogh in Google DeepDream Space – YouTube](#)

[Exploring Activation Functions for Neural Networks | by Dima Shulga | Towards Data Science](#)

[The Latest Trends in Deep Learning Optimization Methods - Do It Easy With ScienceProg](#)

[Schleie – Wikipedia](#)

[Breaking neural networks with adversarial attacks | by Anant Jain | Towards Data Science](#)

[Racial Bias and Gender Bias in AI systems | by big lex | The Comuzi Journal | Medium](#)

**Awad, Edmond, et al.** "Crowdsourcing moral machines." *Communications of the ACM* 63.3 (2020): 48-55.