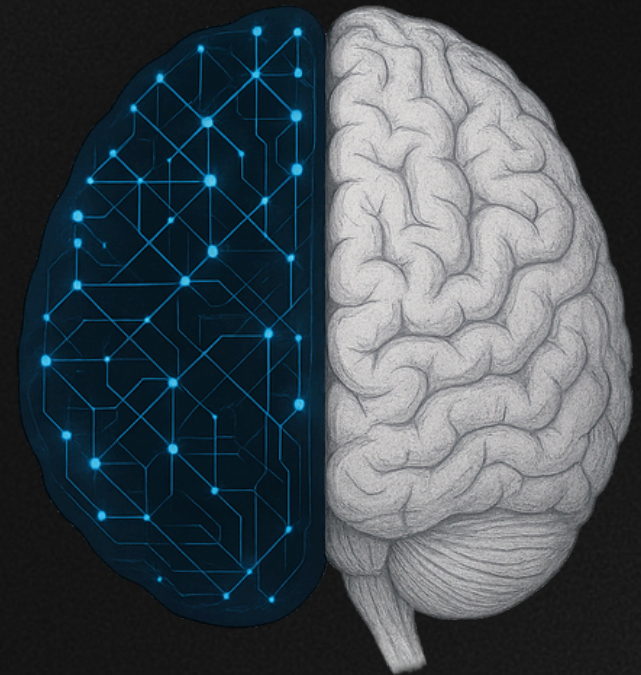


WAHRSCHEINLICHKEIT IN

KI

NEURONALEN NETZEN

VON KASEM RASHRASH (41398)

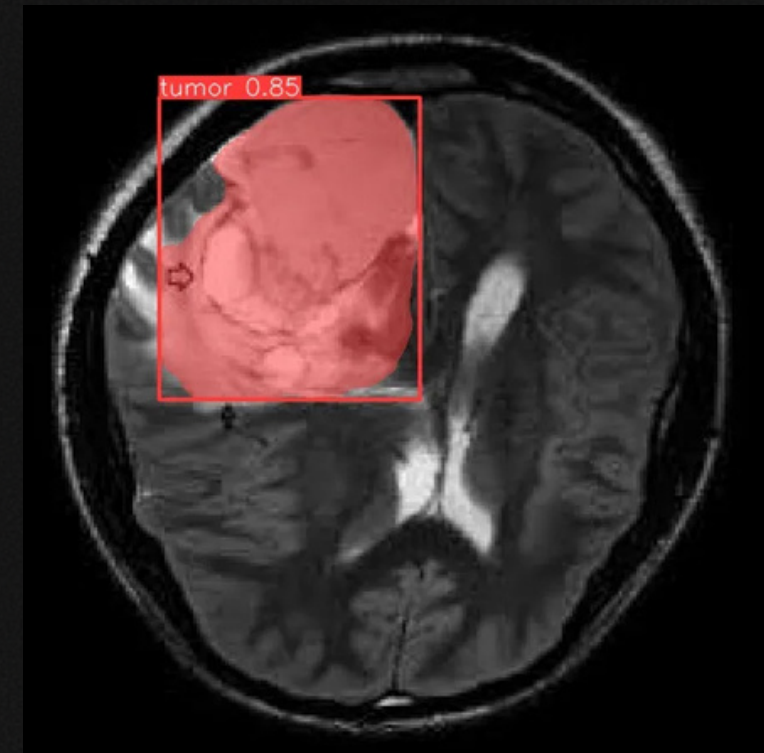
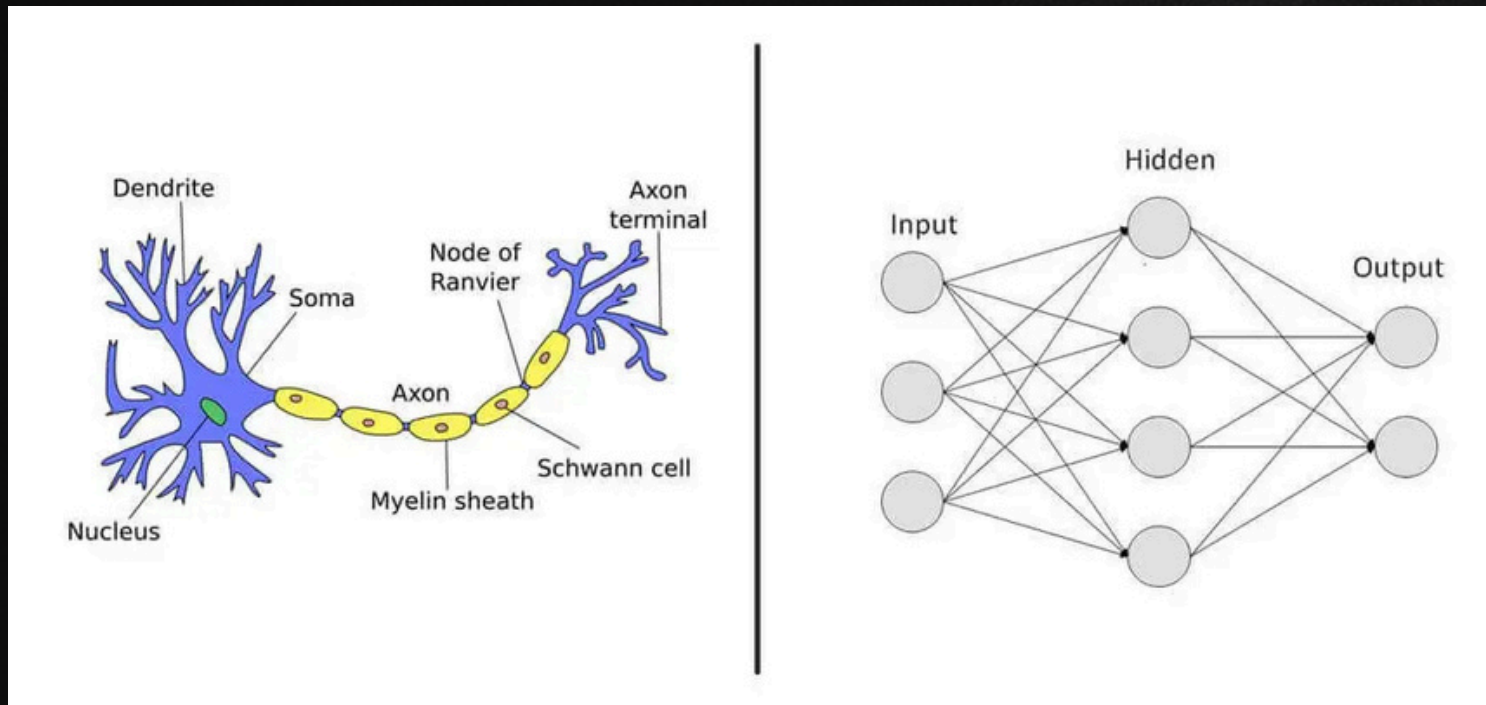


GLIEDERUNG

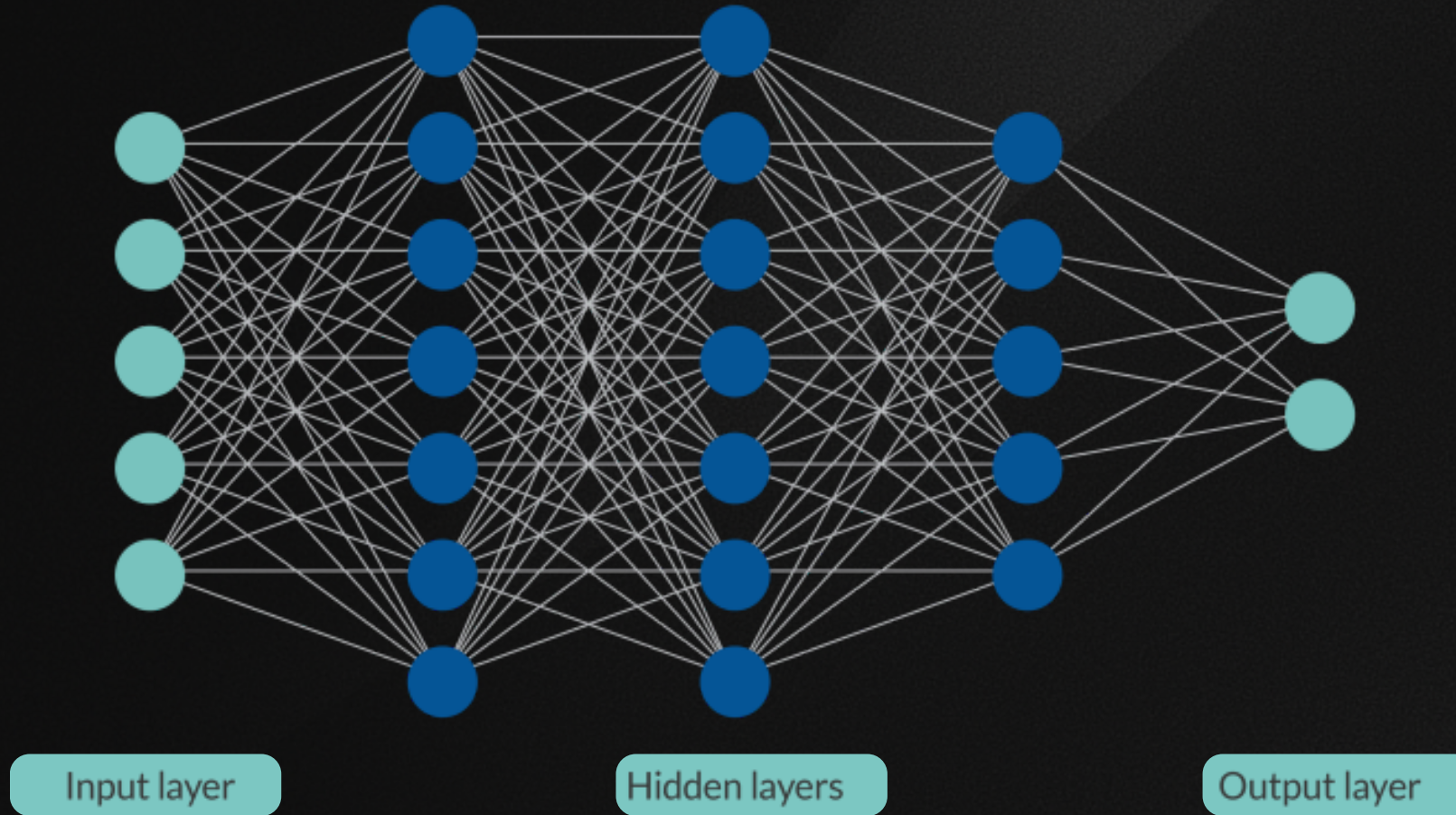
- Einführung
- Grundlagen neuronaler Netze
- Aufbau eines neuronalen Netzes
- Aktivierungsfunktionen
 - *Sigmoid-Funktion (für binäre Klassifikation)*
 - *Softmax-Funktion (für Mehrklassen-Klassifikation)*
- Wie entscheidet das Netz? (z. B. Schwellenwert, höchste Wahrscheinlichkeit)
- Unsicherheit und Fehlerwahrscheinlichkeit
- Beispiel: Tumorerkennung mit neuronalen Netzen
- Fazit

EINFÜHRUNG

- Bedeutung von Wahrscheinlichkeit in KI und neuronalen Netzen
- Beispiel: Tumorerkennung mit neuronale Netzen/KI
- Ziel: Verstehen, wie ein neuronales Netz Wahrscheinlichkeiten berechnet



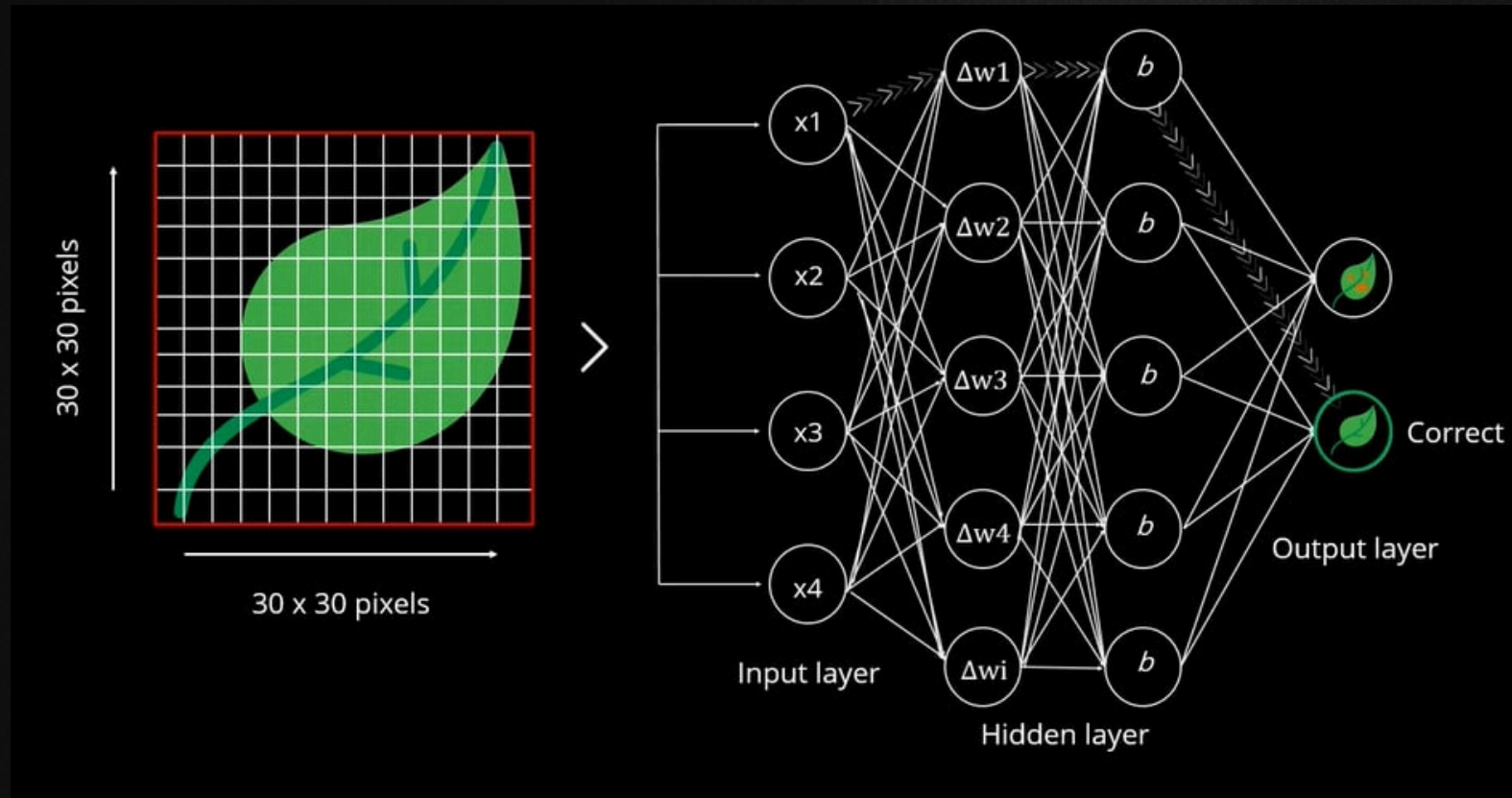
NEURONALES NETZ - AUFBAU



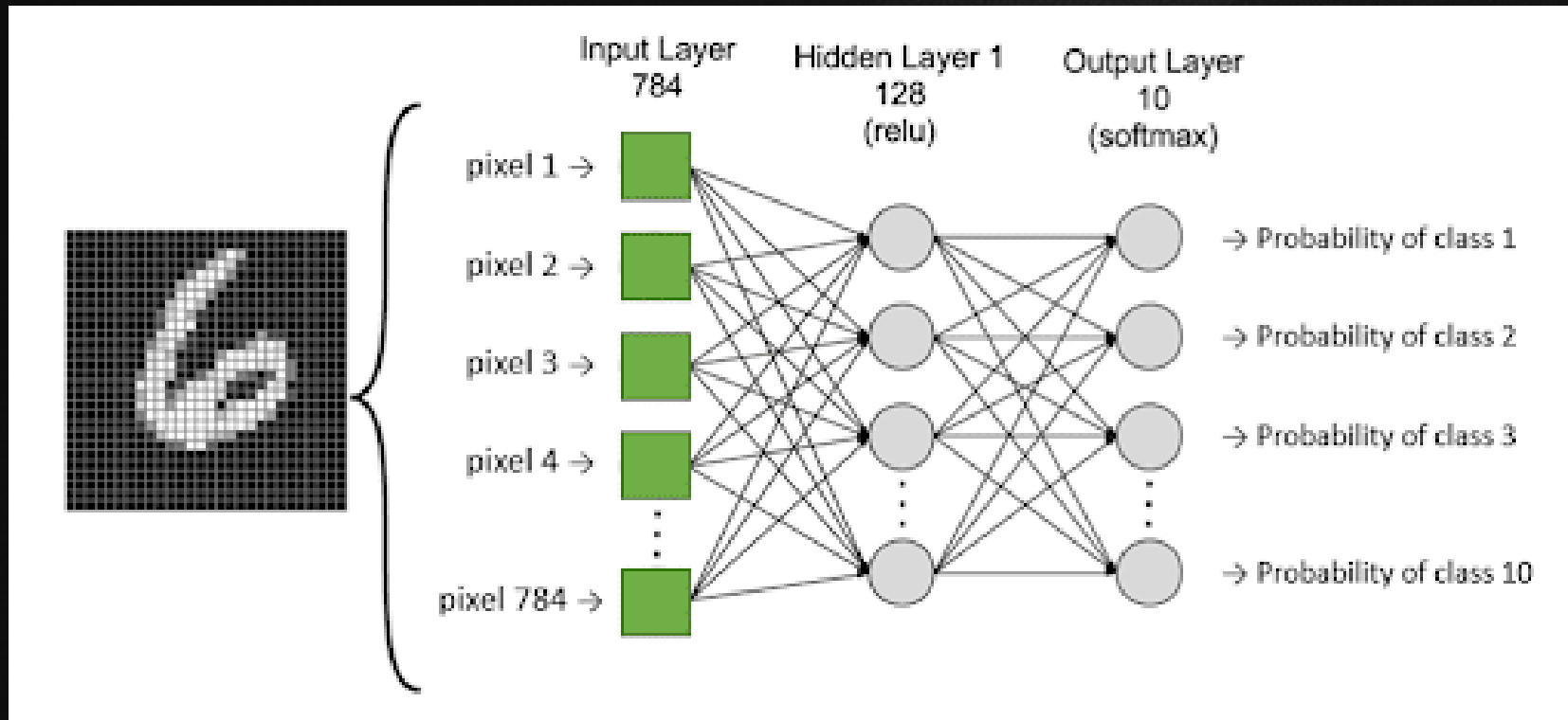
AUFBAU UND FUNKTION

- **Eingaben (Inputs):**
 - Das Neuron bekommt Werte von anderen Neuronen oder direkt von den Eingabedaten.
- **Gewichtung:**
 - Jede Eingabe wird mit einem bestimmten Gewicht multipliziert. Das Gewicht bestimmt, wie wichtig der jeweilige Input ist.
- **Aufsummieren:**
 - Alle gewichteten Eingaben werden aufsummiert. Dazu kommt meistens noch ein Bias (Schwellenwert).
- **Aktivierungsfunktion:**
 - Die Summe wird durch eine Aktivierungsfunktion (z. B. Sigmoid, ReLU, Softmax) geschickt.
Dadurch entsteht der Output des Neurons.

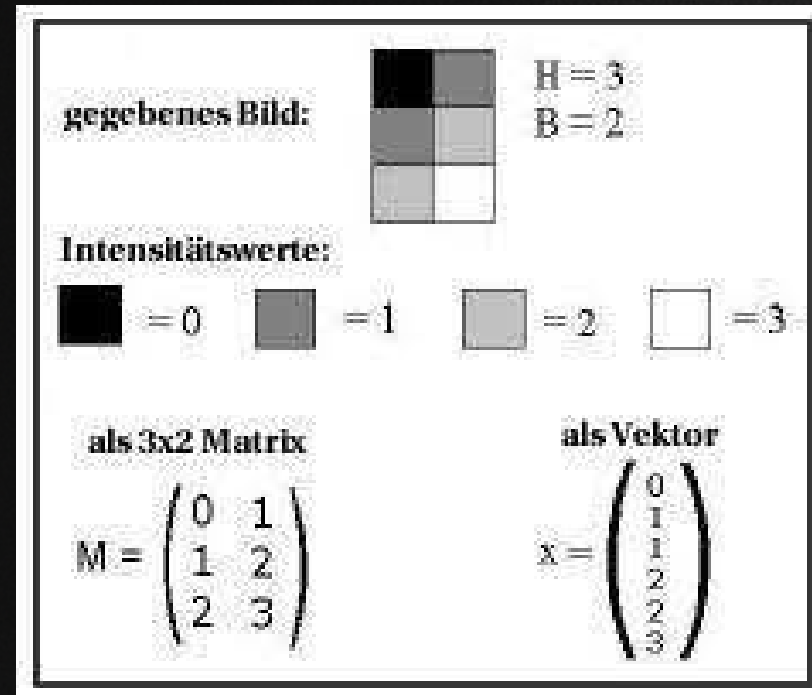
NEURONALES NETZ - BEISPIEL



NEURONALES NETZ - BEISPIEL



DAS EINGABEBILD ALS VEKTOR



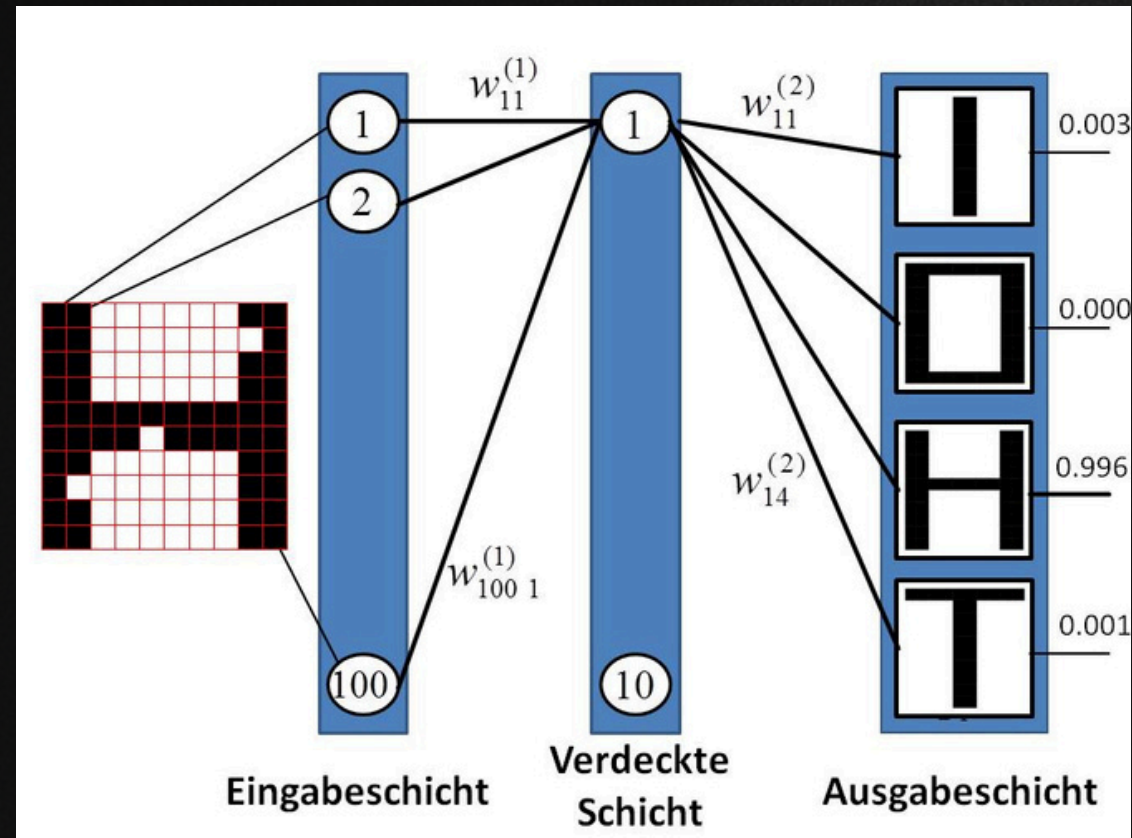
GEWICHTE

- Sie bestimmen, wie sehr ein Neuron die Eingabe beeinflusst

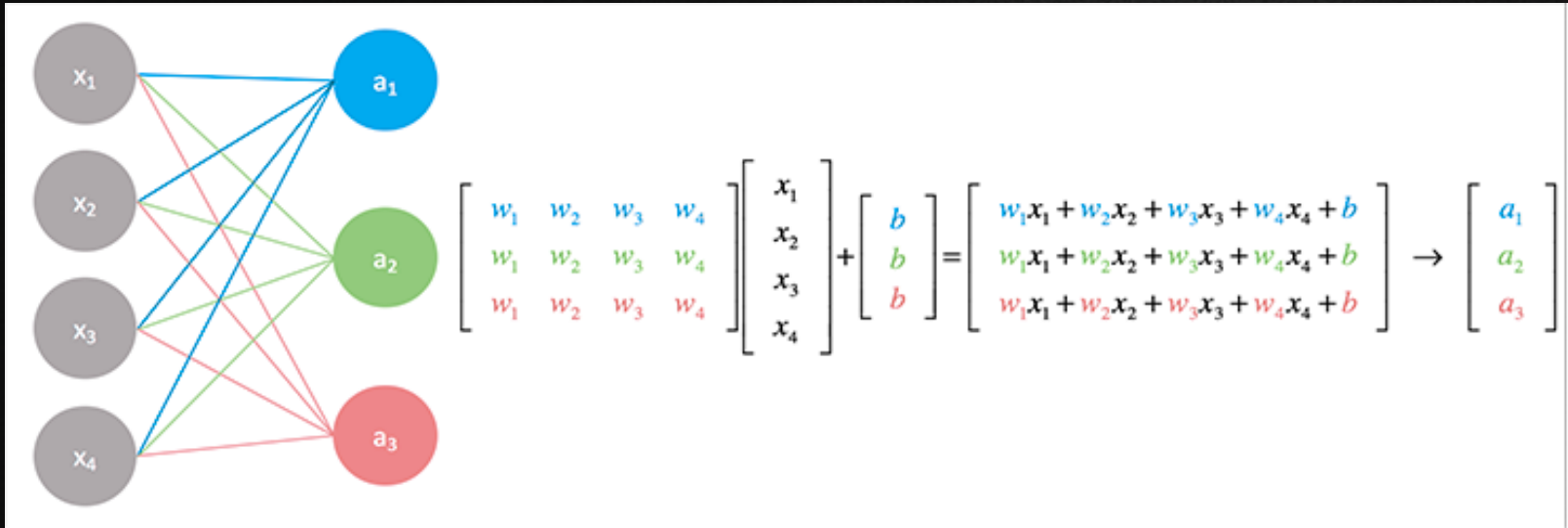
z.B. :

- Tumorgroße, Sehr wichtig → wird stark gewichtet (0.9)
- Alter der Person → Weniger Einfluss (0.2)
- Kontrastwert im MRT → Negativer Zusammenhang (-0.3)
- zufällig gewählt

NEURONALES NETZ - BEISPIEL



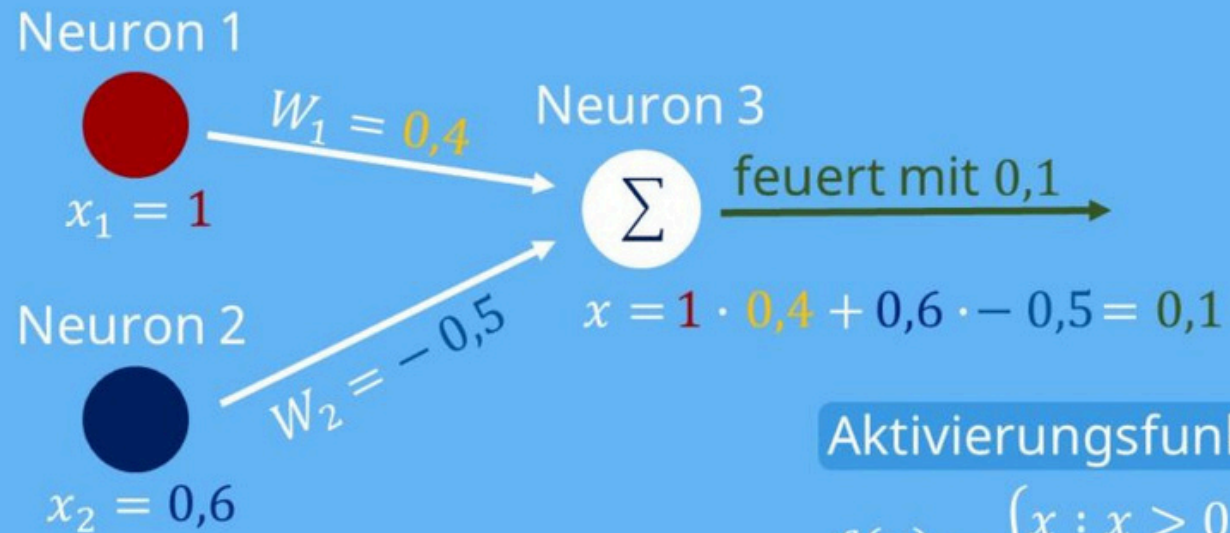
GEWICHTUNG & BIAS



- Berechnung: $z = \sum (w_i * x_i)$
- Eingesetzt werden hier: x_i = Pixelhelligkeiten, w_i = Gewichtungen, b = Bias
- Der Bias rettet die Aktivierung, wenn der Input allein nicht ausreicht und erlaubt dem Netzwerk, auch bei Null- oder schwachen Signalen zu lernen.

NEURONALES NETZ - BEISPIEL

Aktivierung neuronaler Netze



Aktivierungsfunktion

$$f(x) = \begin{cases} x & : x > 0 \\ 0 & : x \leq 0 \end{cases}$$

WAHRSCHEINLICHKEITSBERECHNUNG MIT SIGMOID

- Beispiel: $z = 1,54$
- Ergebnis: $\approx 0,823$
- Interpretation: 82,3% Wahrscheinlichkeit für Tumor
- Warum Sigmoid? → Gibt einen klaren Wert zwischen 0 und 1 → direkt als Wahrscheinlichkeit interpretierbar

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

SIGMOID - BEISPIEL 1

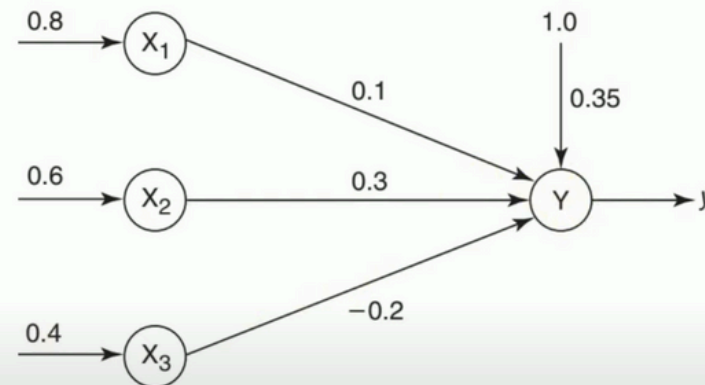
- The net input to the output neuron is

$$y_{in} = b + \sum_{i=1}^n x_i w_i$$

$$= b + x_1 w_1 + x_2 w_2 + x_3 w_3$$

$$= 0.35 + 0.8 \times 0.1 + 0.6 \times 0.3 + 0.4 \times (-0.2)$$

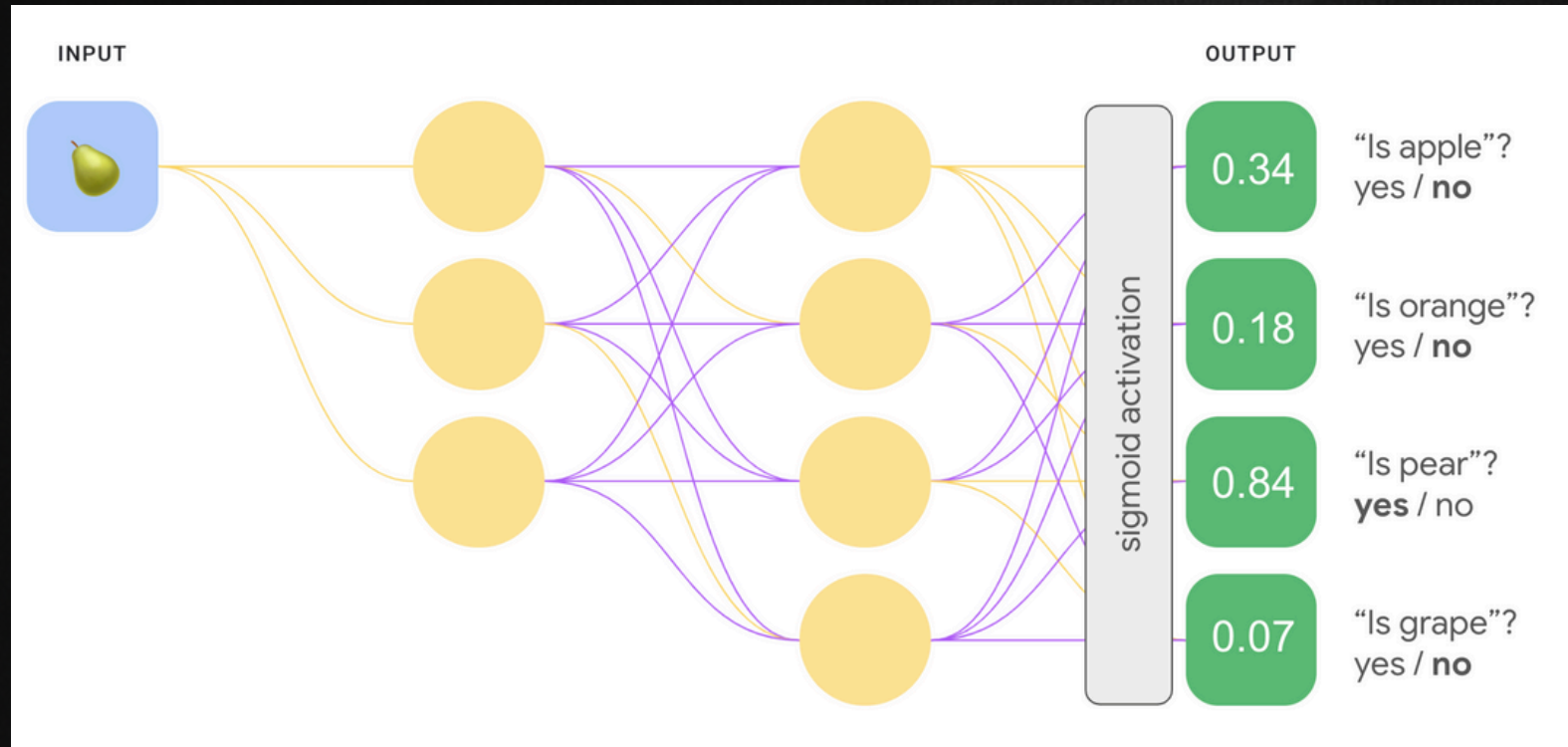
$$= 0.35 + 0.08 + 0.18 - 0.08 = 0.53$$



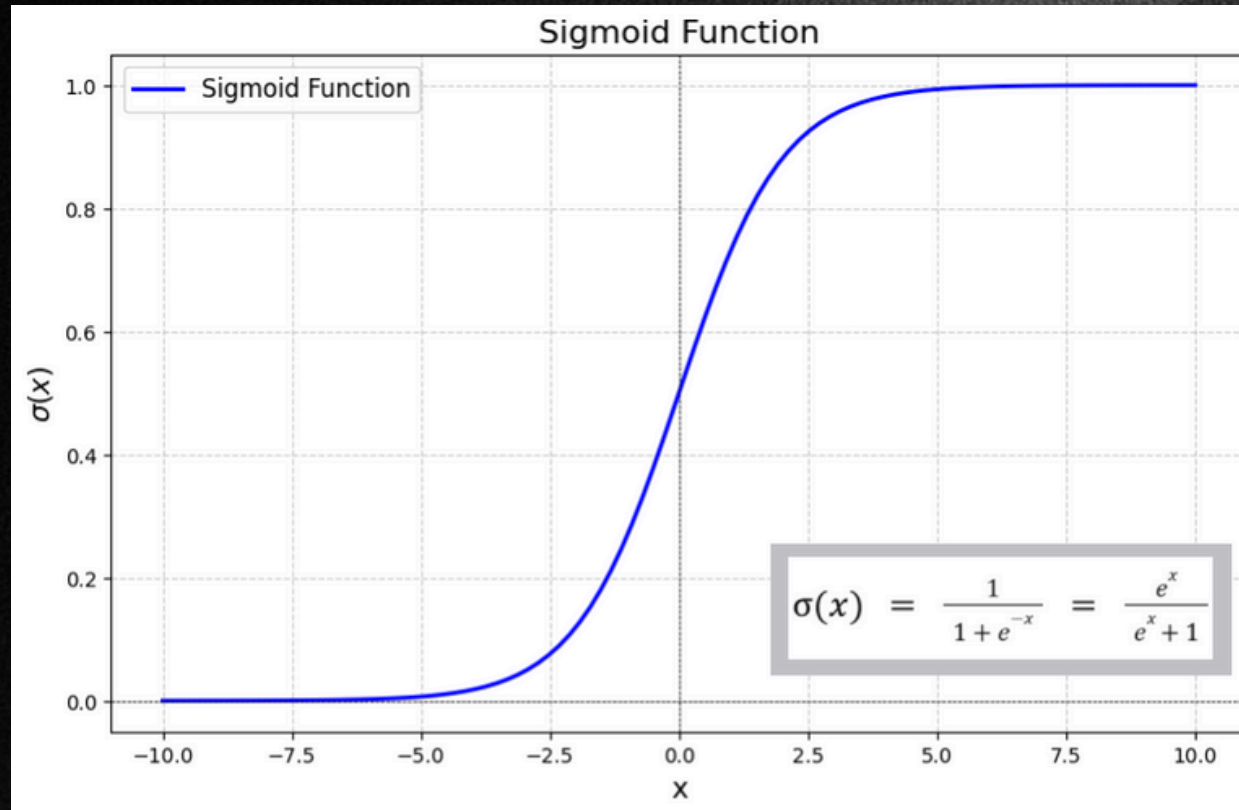
- Binary Sigmoid Activation Function

$$y = f(y_{in}) = \frac{1}{1 + e^{-y_{in}}} = \frac{1}{1 + e^{-0.53}} = 0.625$$

SIGMOID - BEISPIEL 2



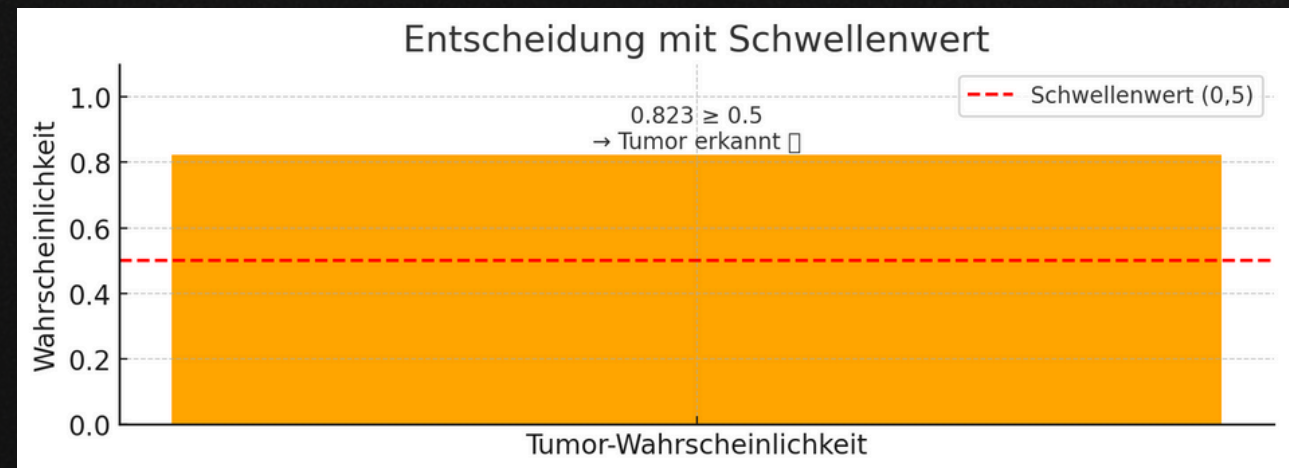
SIGMOID FUNKTION



WIE LERNT DAS NETZ? (TRAININGSPHASE)

- Netz gibt eine Vorhersage (z. B. $p = 0,85$)
- Vergleich mit Ground Truth (z. B. 1)
- Fehler wird berechnet mit: $\text{Loss} = -(1 \cdot \log(0.85) + (1-1) \cdot \log(1-0.85)) = -\log(0.85) \approx 0.1625$
- Beispiel: $\text{Loss} \approx 0,195$
- Bedeutung: Je kleiner der Loss, desto näher liegt das Netz an der Wahrheit

$$L = - \sum_{k=1}^K y_k \log(p_k)$$

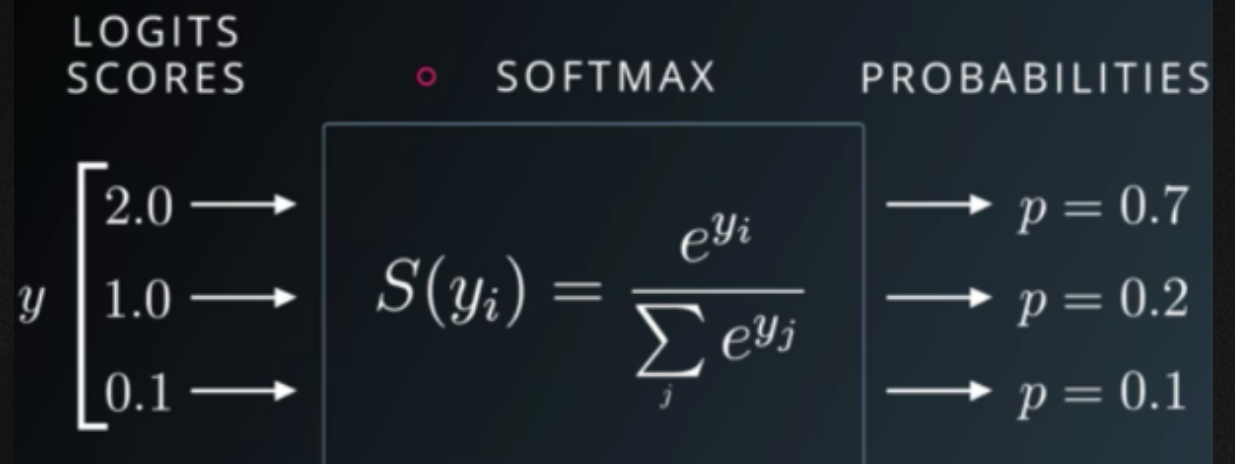


WIE LERNT DAS NETZ? (TRAININGSPHASE)

- Entscheidung mit Schwellenwert
- Standard-Schwelle: 0,5
- Wenn Wahrscheinlichkeit $\geq 0,5 \rightarrow$ Tumor erkannt
- Unser Fall: $0,823 \geq 0,5 \rightarrow$ Tumor erkannt 

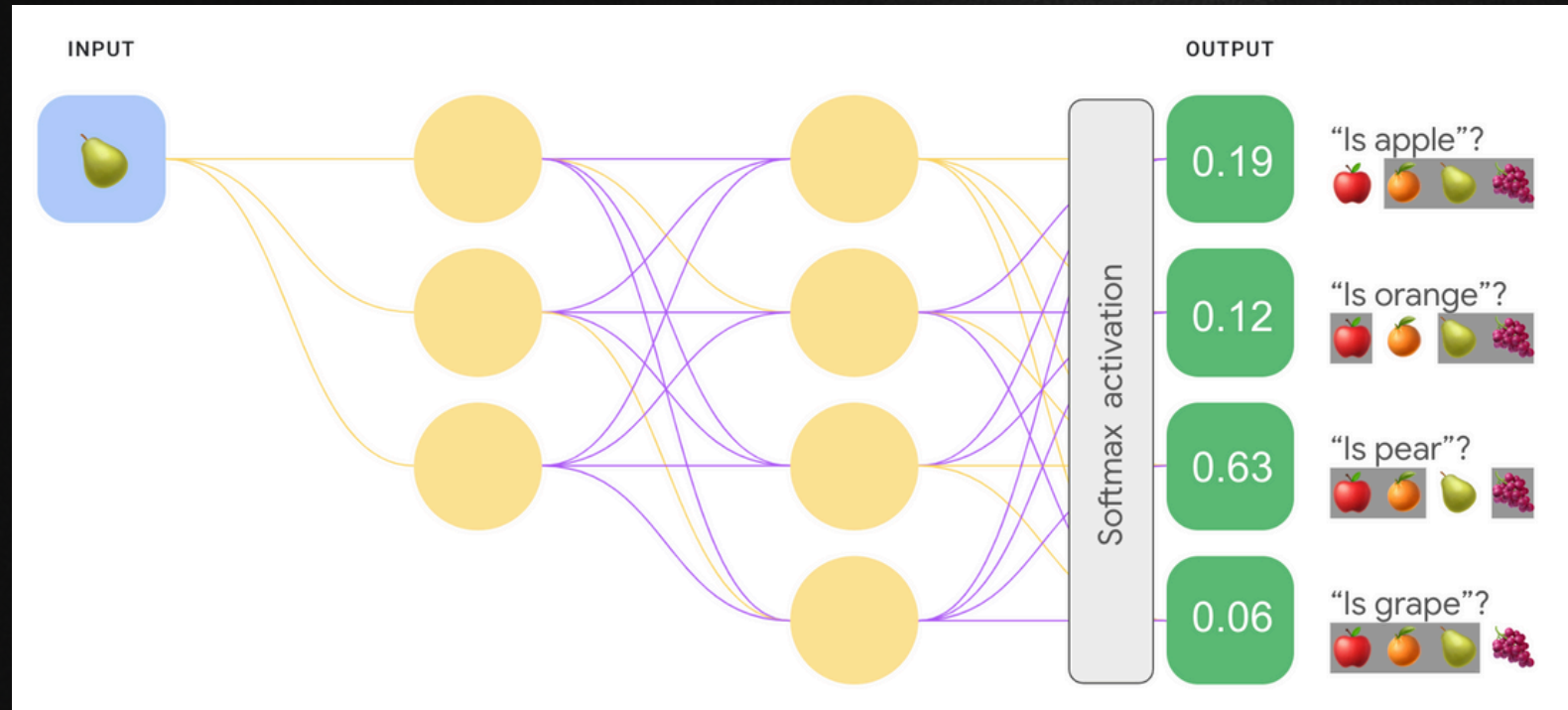
SOFTMAX-FUNKTION

- Konvertiert Output neuronaler Netze in Wahrscheinlichkeitsverteilung
- Nur nötig bei mehr als 2 Klassen (z. B. gutartig / bösartig / unklar)



```
1 import numpy as np
2 def softmax(z):
3     return np.exp(z) / np.sum(np.exp(z))
4 softmax([2.0, 1.0, 0.1])
5
```

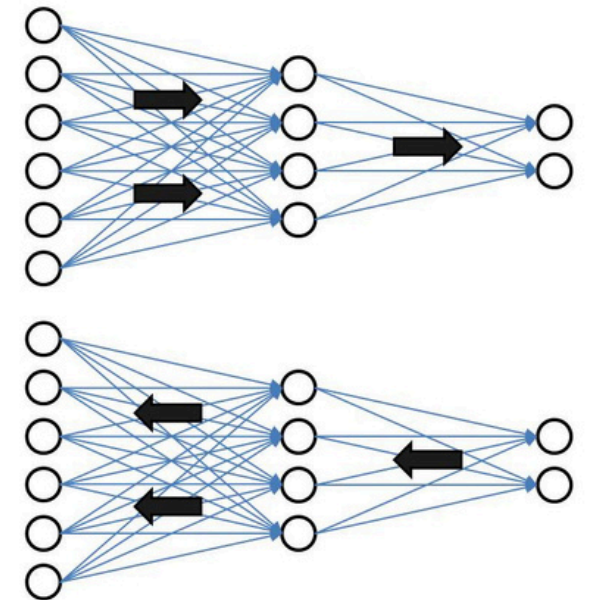

SOFTMAX-FUNKTION - BEISPIEL



GEWICHTE VERBESSERN DURCH LERNEN

- Fehler wird zurück durch das Netz geschickt (Backpropagation)
- Gewichte werden angepasst: $w = w - \eta * \partial \text{Loss} / \partial w$
- Das Gewicht, das angepasst werden soll - Die Lernrate
- Ziel: Fehler minimieren → bessere Vorhersagen

$$L = - \sum_{k=1}^K y_k \log(p_k)$$

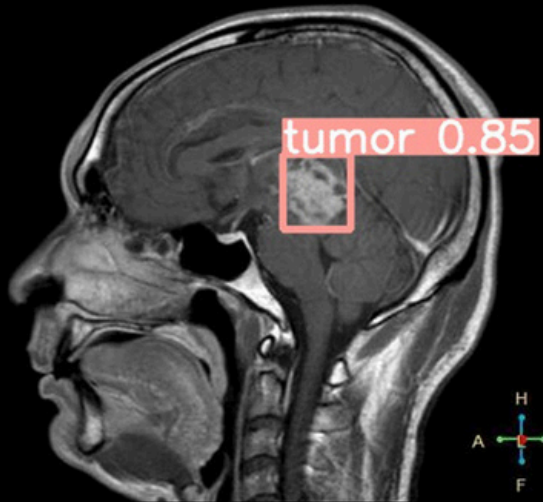


BEISPIEL: TUMORERKENNUNG MIT NEURONALEN NETZEN

- KI-Modell analysiert medizinische Bilddaten (z. B. MRT-Aufnahmen).
- Output: Wahrscheinlichkeit, ob ein Tumor vorhanden ist oder nicht.
- Hohe Wahrscheinlichkeit (z. B. 95 %): **Sehr sicher**, Tumor erkannt.
- Mittlere Wahrscheinlichkeit (z. B. 55 %): **Unsicher**, weitere Untersuchungen nötig.
- Niedrige Wahrscheinlichkeit (z. B. 5 %): **Sehr sicher**, kein Tumor.

BEISPIEL: TUMORERKENNUNG MIT NEURONALEN NETZEN

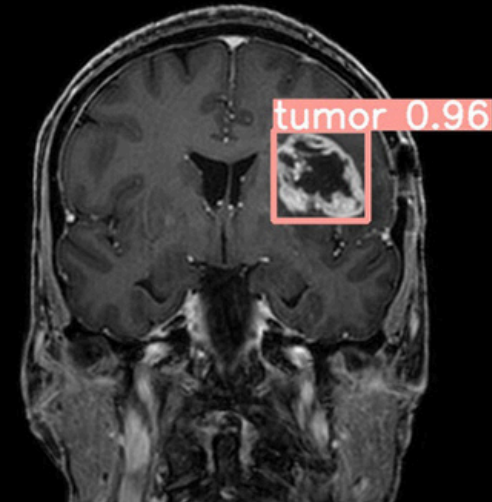
Patient 1



¹⁸ Patient 2



Patient 3



FAZIT

- **Wahrscheinlichkeitsrechnung ist entscheidend für KI**
- **Wichtigste Konzepte: Softmax-Funktion, Kreuzentropie, Satz von Bayes**
- **Mathematische Grundlagen sind essenziell für Verständnis und Weiterentwicklung von KI-Modellen**
- **Das Netz lernt durch Vergleichen von Vorhersage und Wahrheit**
- **und verbessert seine interne Logik (Gewichte) immer weiter.**

QUELLEN

- <https://sebastiandoern.de/neuronale-netze/>
- https://madoc.bib.uni-mannheim.de/1100/1/TR_2005_008_FaceRecognition.pdf
- <https://www.quantib.com/blog/how-does-deep-learning-work-in-radiology>
- Studyflix (2024), „Wahrscheinlichkeitsverteilungen“, [online] [Zugriff: 20.04.2025]
- Sofatutor (2024), „Bedingte Wahrscheinlichkeit“, [online][Zugriff: 20.04.2025]
- StudySmarter (2024), „Satz von Bayes“, [online] [Zugriff: 20.04.2025]
- Wikipedia (2024), „Kreuzentropie“, [online] [Zugriff: 20.04.2025]
- IBM Developer Blog (2024), „Was ist eine Verlustfunktion?“, [online] [Zugriff: 20.04.2025]
- Jay Kumaran “Brain Tumor Detection and Segmentation with Computer Vision YoloV8”, [online] [Zugriff 10.05.2025]