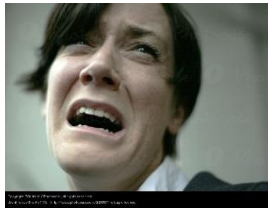
automotive engineering iav





CNNs für Ersties

- Wie geht‘s?
- Arthur Wilms, Berlin, Juli 2019



Convolutional Neuronal Network

Freude
90%

Wut
3%

Trauer
1%

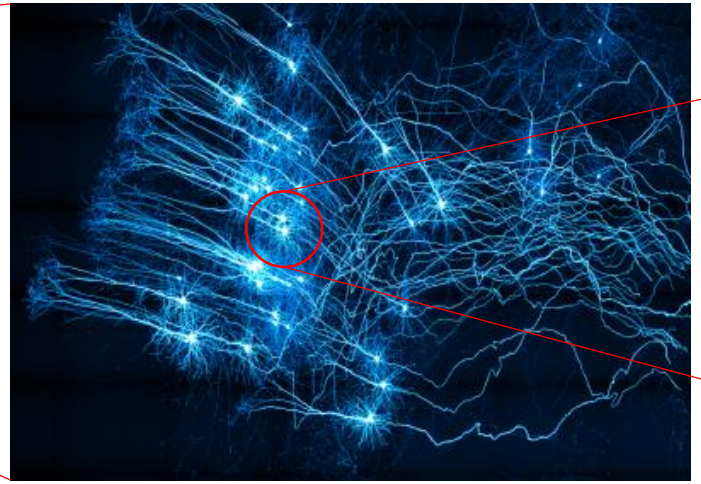
Ekel
4%

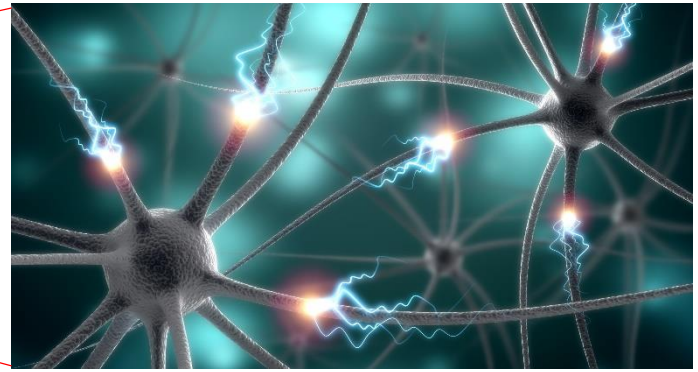
Angst
2%

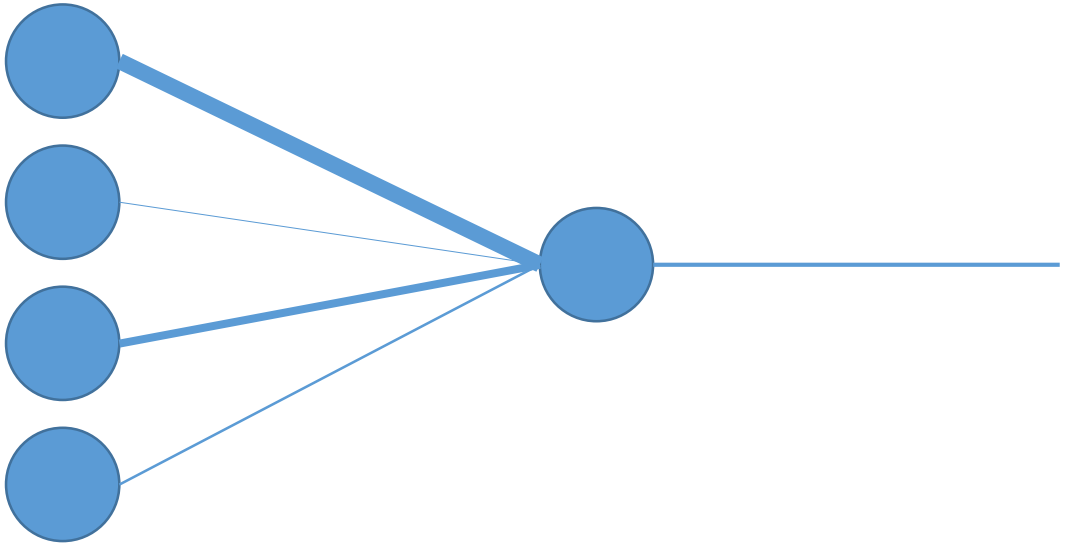
Wozu?

Aaahhhhh cooooool

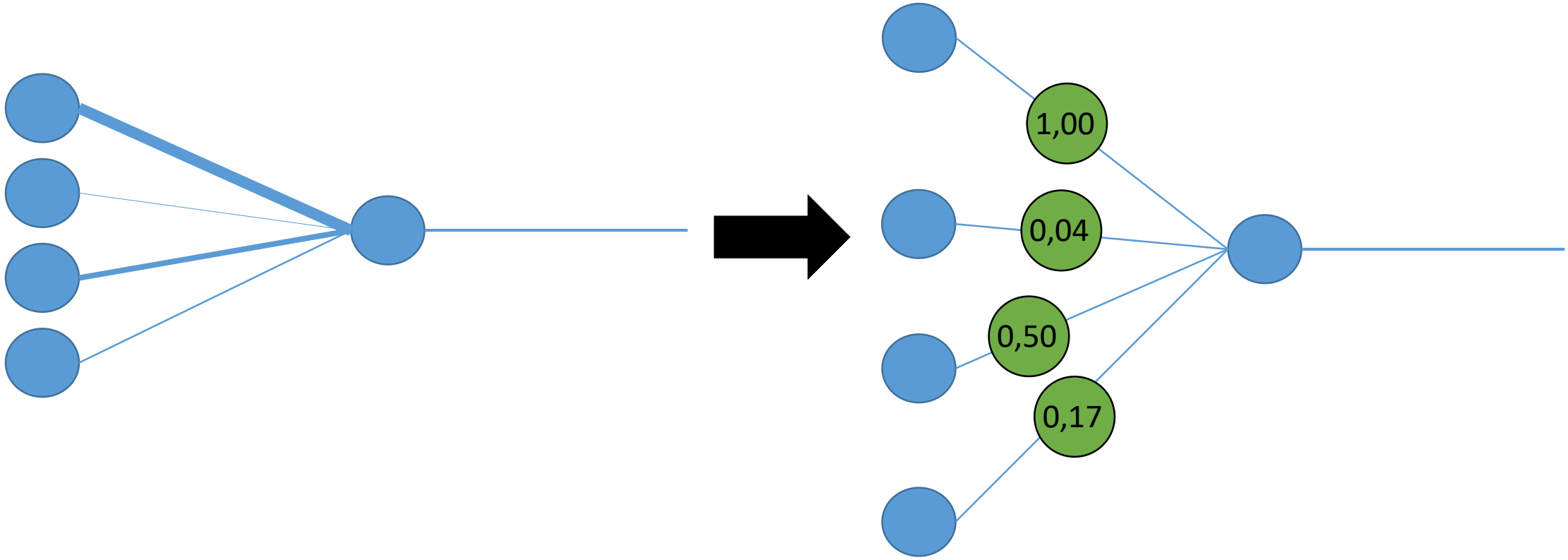
Inspiration



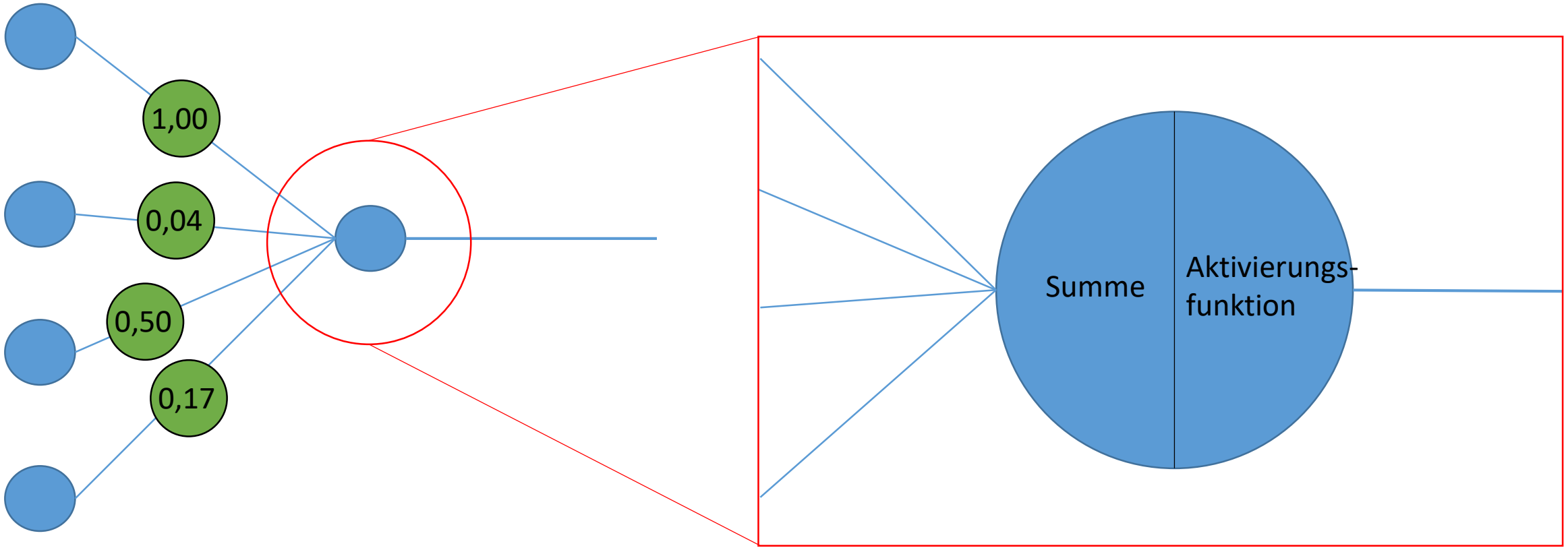




Vom Natürlichen zum Künstlichen



Aktivierungsfunktion

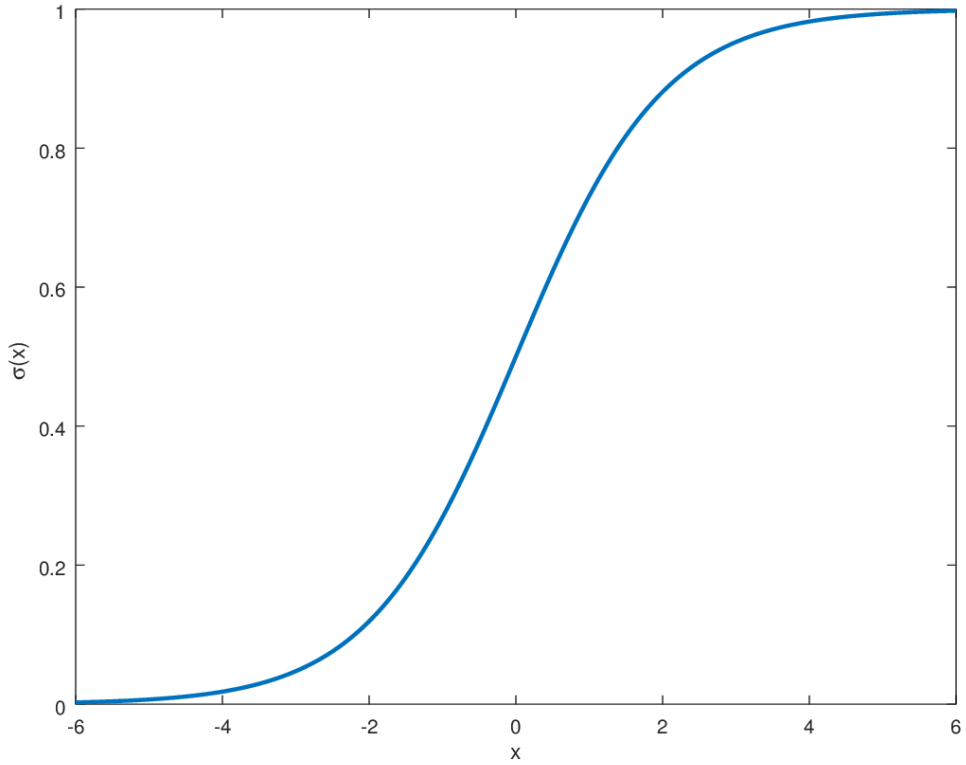


Aktivierungsfunktion: Inspiration

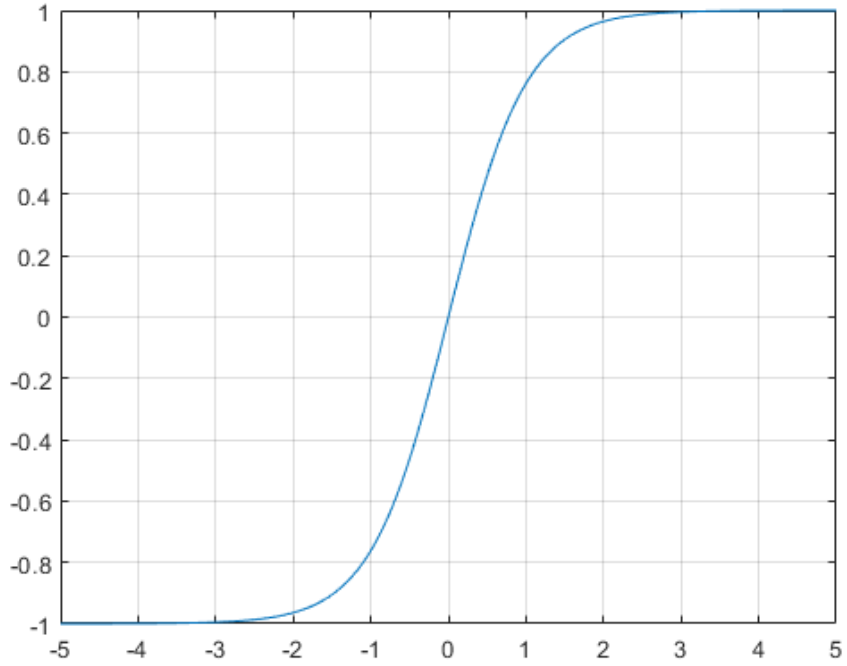
Summe

Wert der Aktivierungs-funktion

Beispiele für Aktivierungsfunktionen

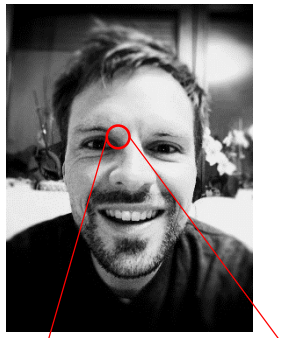


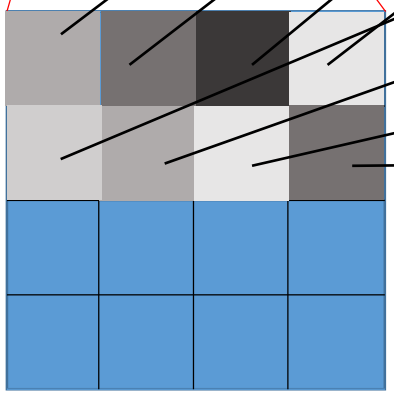
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



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

Netzwerk





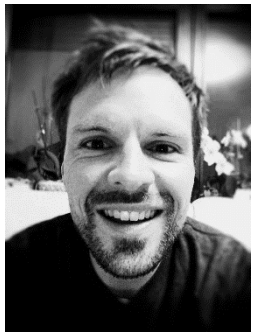
Eingabe-
schicht

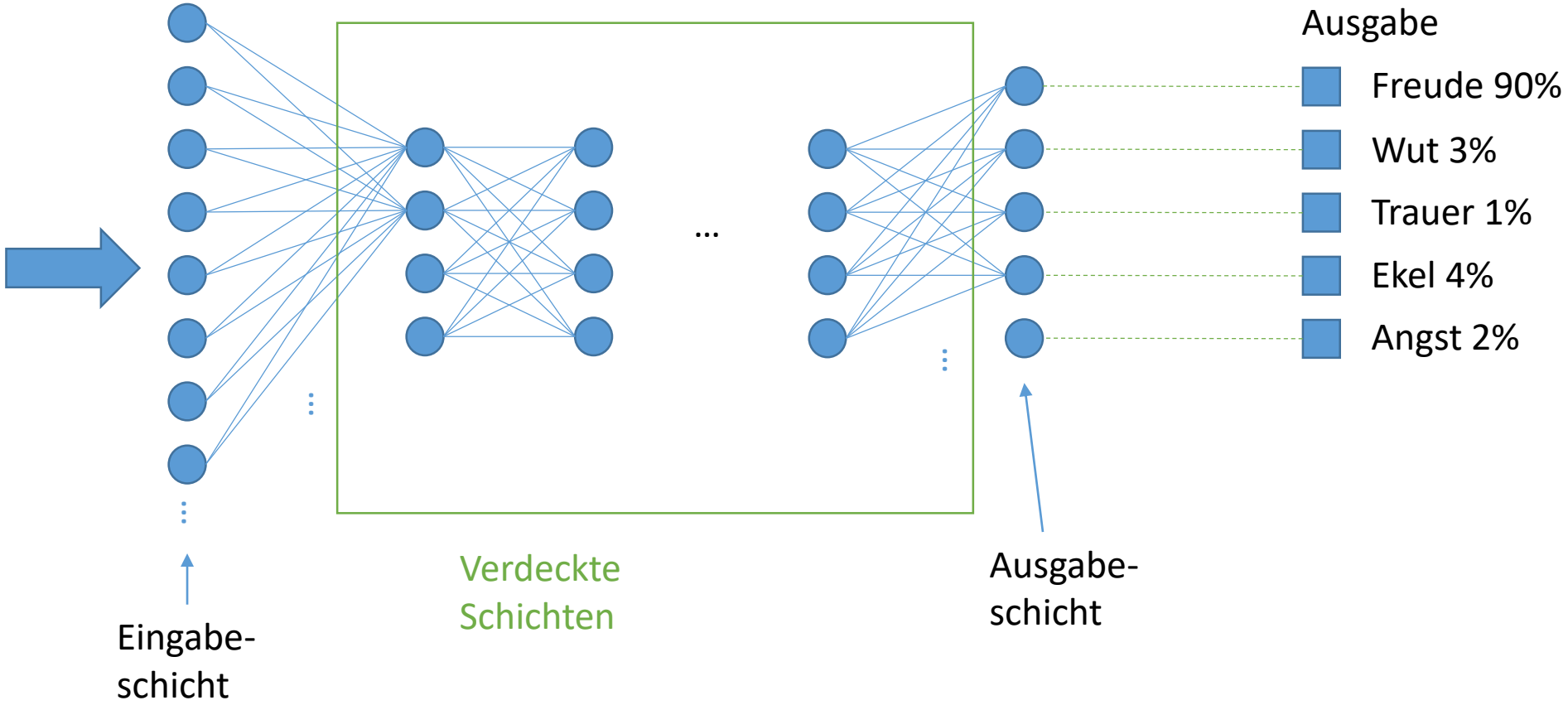
Convolutional
Neuronal
Network

Ausgabe

- Freude 90%
- Wut 3%
- Trauer 1%
- Ekel 4%
- Angst 2%

Netzwerk hinter den Kulissen

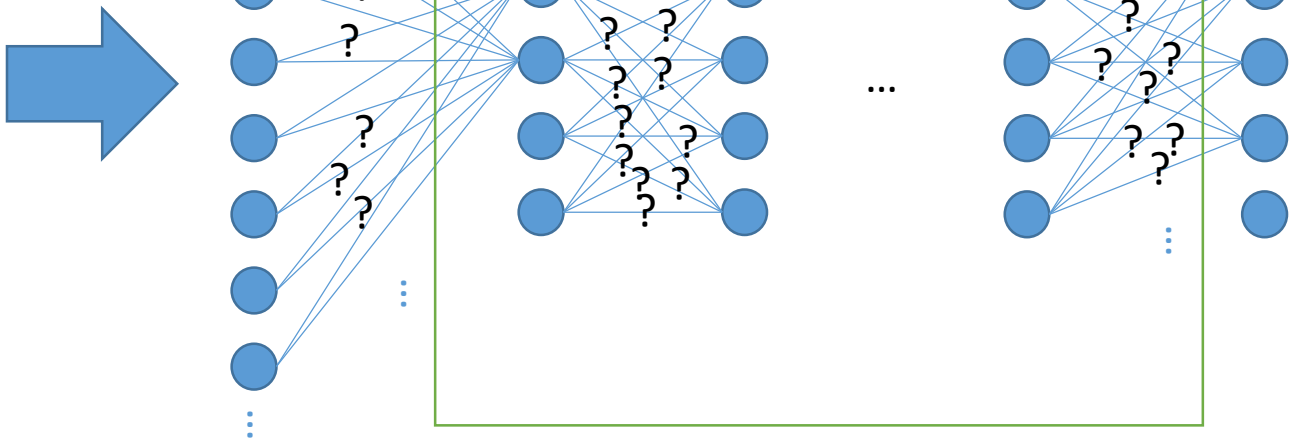




An jeder durchgezogenen Linie ein Gewichtsfaktor

→Welche Werte müssen sie haben?

Netzwerk dimensionieren



Freude
Wie wahrscheinlich?

Wut
Wie wahrscheinlich?

Trauer
Wie wahrscheinlich?

Ekel
Wie wahrscheinlich?

Angst
Wie wahrscheinlich?

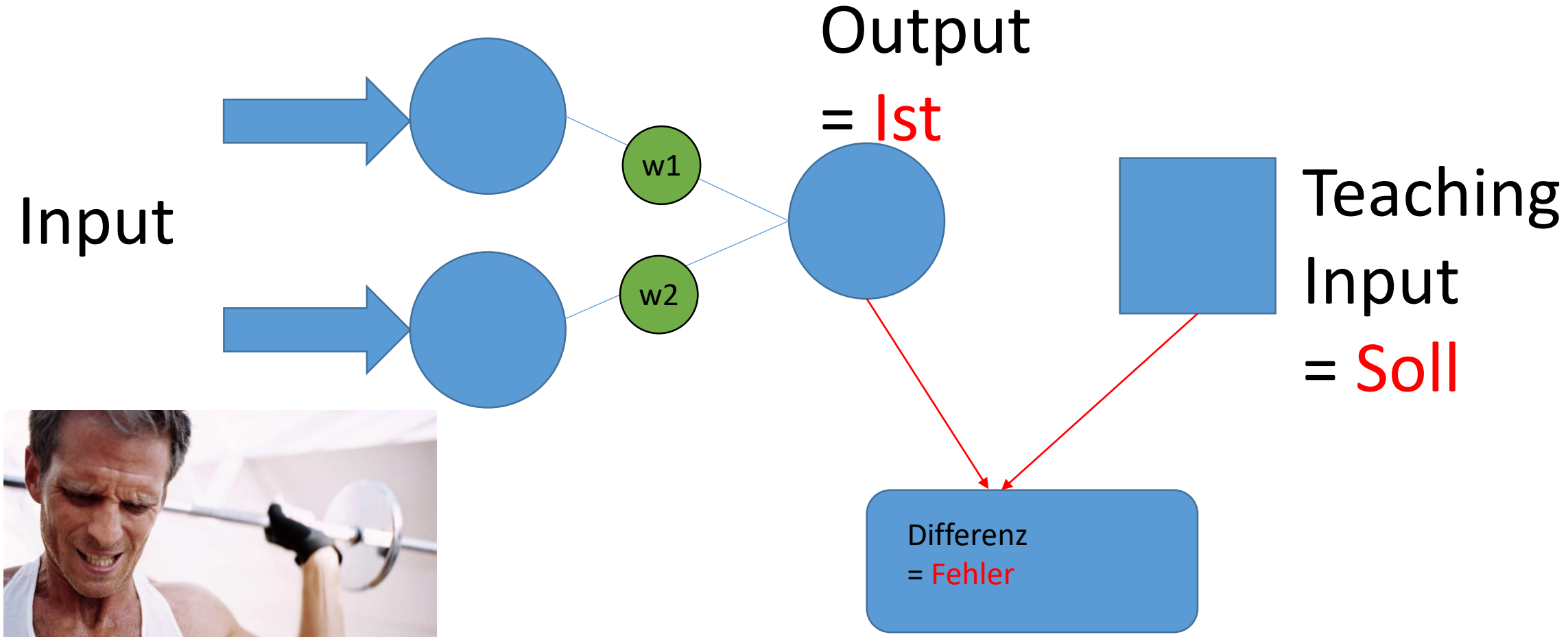
Aus entsprechendem Input muss richtiger Output folgen
→Welche Werte müssen dann die millionen „?“ haben?

Vereinfachung zur Veranschaulichung

1.000.000D → … →  4D →  3D →  2D

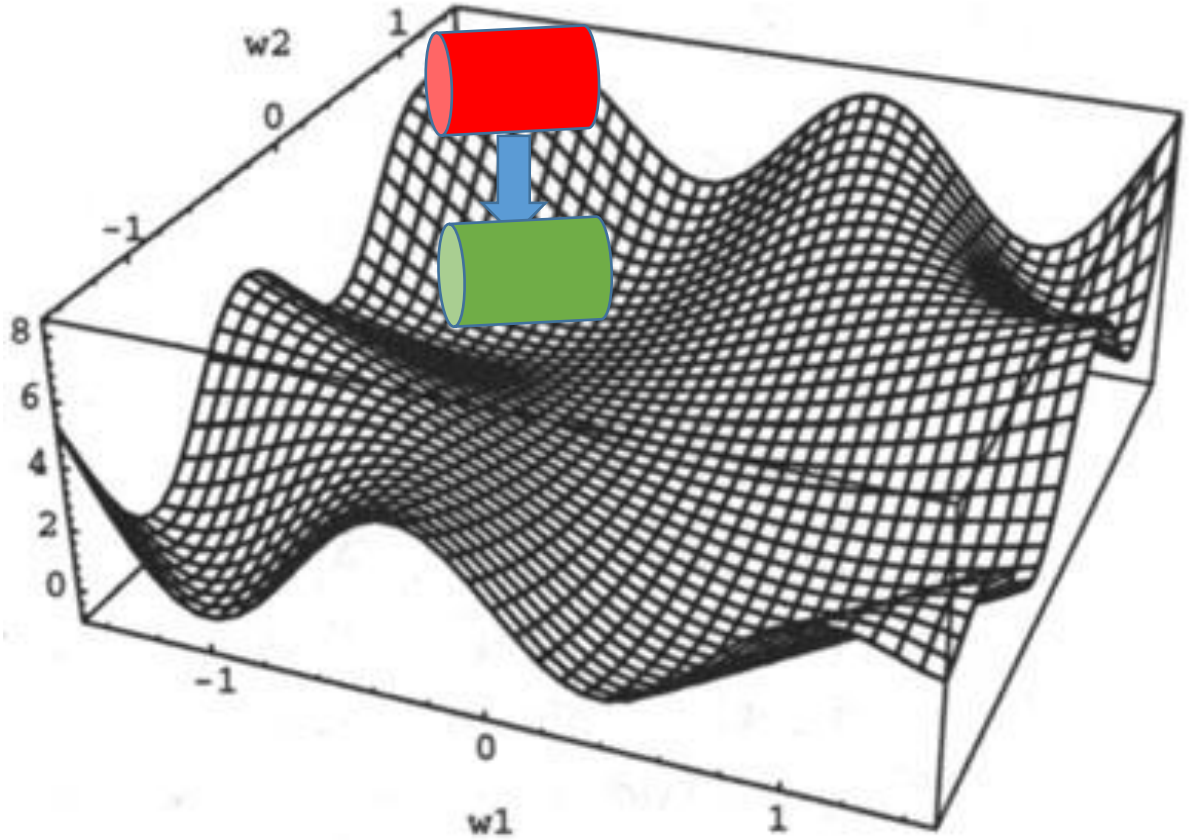
Betrachtung: 2 Gewichtsfaktoren

Training


Output
= Ist
Input
w1
w2
Teaching
Input
= Soll
Differenz
= Fehler

Training iterativ

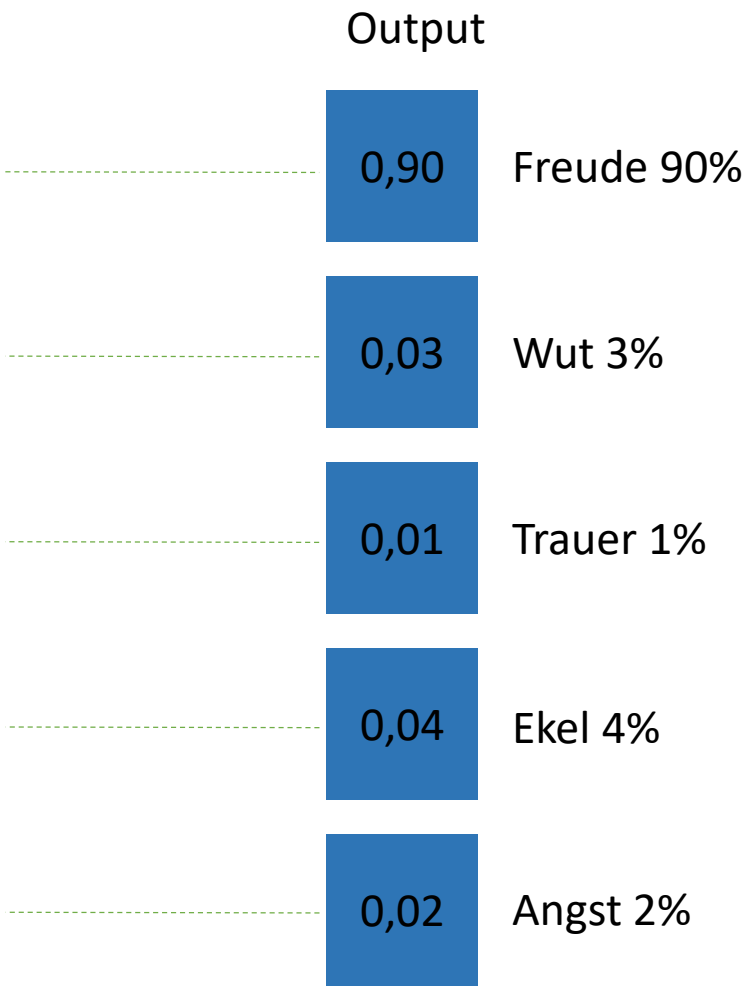
- Aktueller Vektor [w1;w2] sowie Fehler
- Bestimme durch partielle Ableitungen des Fehlers nach jeweils w1 und w2 Vektor [v1;v2], der in Richtung geringerer Fehler zeigt
- Gehe mit [w1;w2] in Richtung [v1;v2] und erhalte so den neuen Vektor [w1*;w2*]



Höhe
= Fehler

→ Für jeden Iterationsschritt ein Vektor als Trainings-Input (z.B. ein Bild) und ein Label (was ist auf dem Bild zu sehen?) benötigt

Output und Teaching Input als Vektoren



We know how you feel! ☺