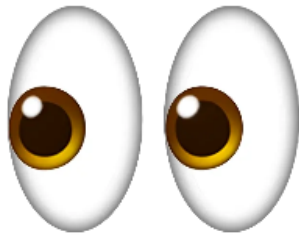


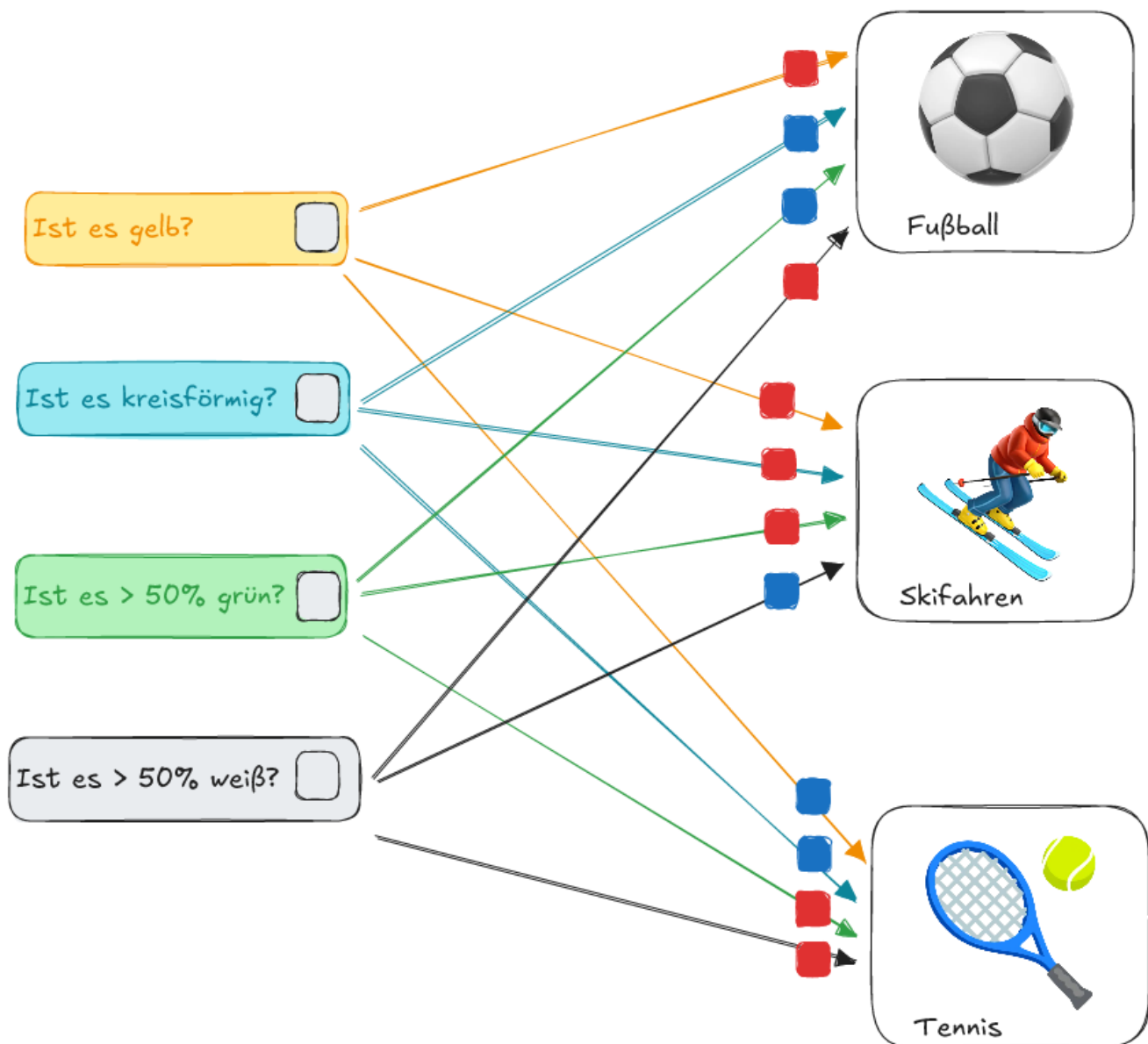
"Wolffaden"-Perzeptron



Was siehst du?



Kann nichts sehen,
aber Auswerten ob "rot"
oder "blau" überwiegt.



Funktionsweise:

Es gibt zwei Gruppen:

- Die Gehirn Gruppe kann nichts sehen und erhält Informationen von der Augen-Gruppe
- Die Augen Gruppe hat für jedes Merkmal nur zwei Möglichkeiten der Meldung an die Gehirn-Gruppe: Ja/Nein, bzw. 0 oder 1 - je nachdem, ob das Merkmal erfüllt ist, oder nicht.

1. Hinter der Gehirn-Gruppe wird ein Trainings-Bild hochgehalten.
2. Die Augen-Gruppe stellt für jedes sichtbare Merkmal das passende Stellschild auf
3. Die Gehirn-Gruppe schaut, was durch die Verbindungen bei ihnen ankommt: Die Ausprägung jedes Merkmals (0/1) multiplizieren sie mit der Anzahl der Wäscheklammern auf der Schnur und legen Chips in der passenden Anzahl und Farbe unter die Klammern.
4. Die Gehirn-Gruppe zeigt Ergebnis: Wer mehr blaue als rote Chips hat, hält die blaue 0 hoch. Bei mehr roten als blauen Chips oder Gleichstand, wird die rote 1 gezeigt.

Man kann mit den dargestellten "Wäscheklammerverteilungen" beginnen - warum?

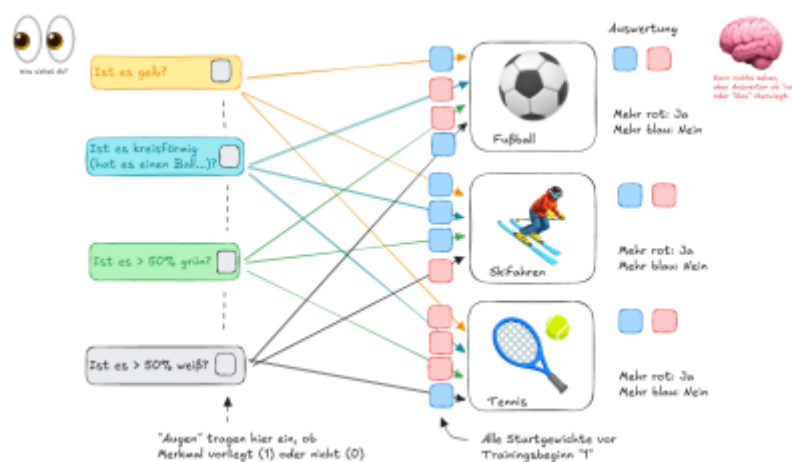
Ausprobieren

Du kannst das Wollfadenperzeptron auch ohne Wollfäden ausprobieren, um zu verstehen, wie es "lernt":

- Du kannst die folierten Ausdrucke verwenden und die Gewichte als Zahlen in die blauen/roten Felder schreiben.
- Oder du kannst das Bild/die PDF-Datei in deinem Tablet laden, um dort die Gewichte und die Ergebnisse für jedes Trainingsbild zu notieren.

Material

- Perzeptron-PDF
- Trainingsbilder



- Lade dir die [Trainingsbilder](#) herunter und entpacke sie. Wähle jeweils 2 beliebige Bilder jeder Sportart aus und verschiebe sie in einen anderen Ordner, den du "test" nennst.
- Nimm von den verbliebenen Bildern eines und "trainiere" das Perzeptron mit diesem Bild:
 - Die "Augen" analysieren das Bild und tragen auf der Eingabeseite die erkannten Merkmale mit 0/1 ein.

- Das "Gehirn" betrachtet die Eingänge und berechnet jeweils den Wert $\text{Merkmal} \cdot \text{Gewicht}$. Beim ersten Trainingsbild sind alle Gewichte 1.
- Die "roten" Summen und die "blauen" Summen werden gebildet.
- Wenn die blaue Summe größer ist als die rote, ist das Ergebnis "Blau" (falsch). Wenn die rote Summe größer ist oder bei Gleichstand ist das Ergebnis "Rot" (richtig).

Aus Fehlern lernen: Bei diesem Vorgang werden falsche Aussagen entstehen: Das Ergebnis für "Skifahren" ist "richtig", obwohl auf dem Bild ein Fußballer zu sehen war o.ä. Um zu "lernen", passen wir jetzt die Gewichte an, in der Hoffnung, dass die Erkennung sich dadurch verbessert, und das geht so:

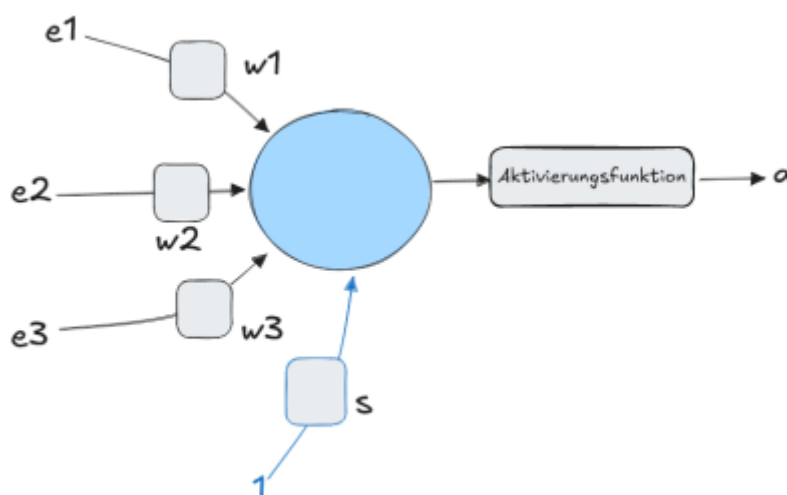
- War das Ergebnis "Rot" statt "Blau", dann werden alle Gewichte, durch die eine 1 angekommen ist, "blauer". D.h. die Gewichte der blaue Eingänge werden um eins vergrößert, bei roten Eingängen wird das Gewicht um eins verkleinert. (Dabei können auch negative Werte auftreten)
- War das Ergebnis "Blau" statt "Rot", dann werden alle Gewichte, durch die eine 1 angekommen ist, "roter". D.h. die Gewichte der blaue Eingänge werden um eins verkleinert, bei roten Eingängen wird das Gewicht um eins vergrößert.
- Bei den korrekten Ergebnissen werden die Gewichte nicht verändert.

Führe jetzt die Gewichts Anpassungen durch, notiere die neuen Gewichte an den Eingängen und trainiere das Netz mit dem nächsten Bild.

Wenn du alle Trainingsbilder verwendet hast, kannst du die so gefundenen Gewichte mit den Testbildern testen.

Etwas allgemeiner und formaler

Ein Perzeptron ist ein simuliertes Neuron:

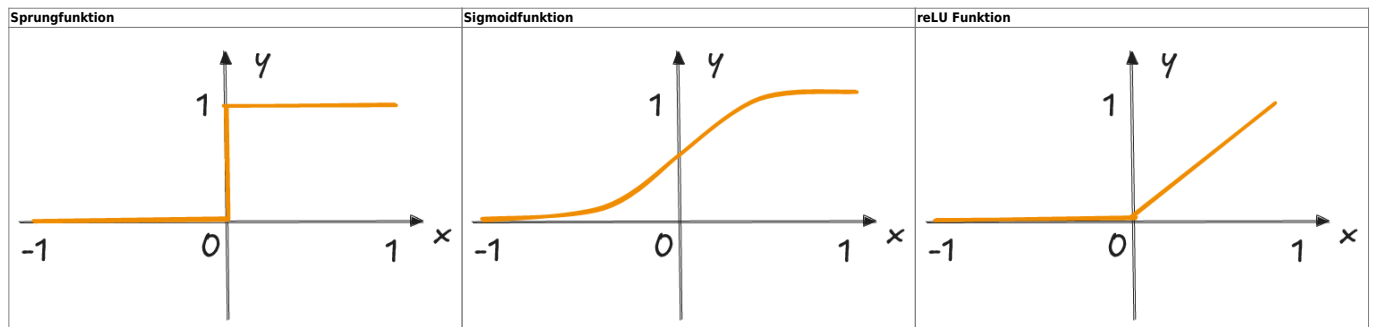


Eingänge:

- Mehrere durch die gewichte w_i gewichtete Einflußfaktoren e_i
- Ein Schwellwert s ("Bias")

Erregungspotential $p = s + e_1 \cdot w_1 + e_2 \cdot w_2 + e_2 \cdot w_2 \dots$

Das Erregungspotential geht in eine **Aktivierungsfunktion**, die festlegt, was am Ausgang a des künstlichen Neurons ausgegeben wird. Es gibt unterschiedliche Aktivierungsfunktionen:



Tierklassifikator

From:
<https://info-bw.de/> -

Permanent link:
https://info-bw.de/faecher:informatik:oberstufe:machine_learning:mljs:perzeptron:start?rev=1740078635

Last update: 20.02.2025 19:10

