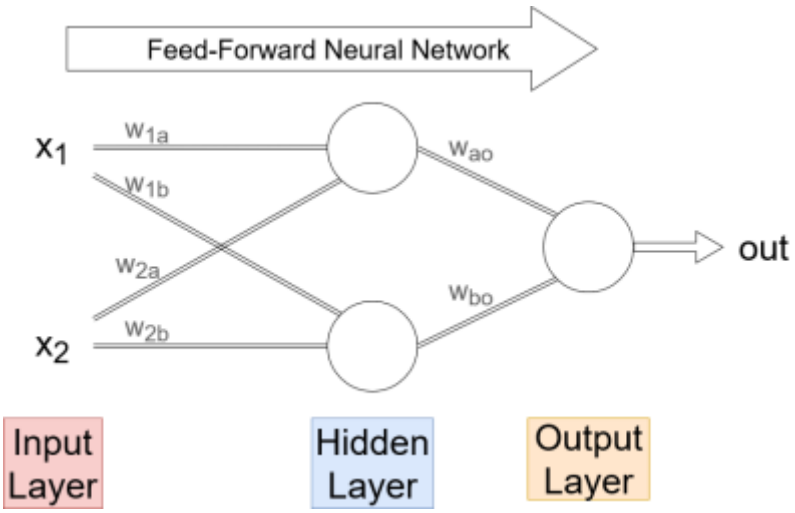
[1)]

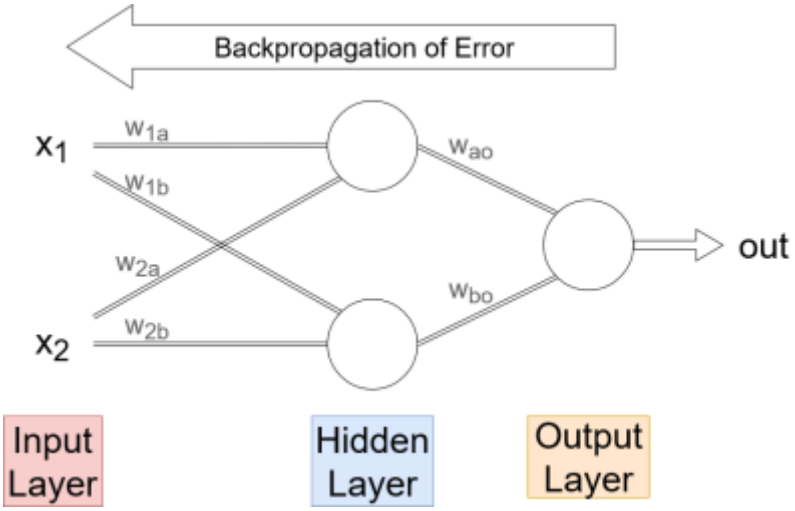
# Wie lernt ein neuronales Netz?

1. Nachdem ein neuronales Netz aufgebaut bzw. erstellt wurde und alle Verknüpfungen neu existieren passiert als Erstes etwas, was zunächst völlig irrational erscheint: Die Gewichte und Schwellenwerte (sogenannte Parameter) des neuronalen Netzes werden mit **zufälligen Werten** initialisiert. *Hintergrund dazu: Das Netz ist anfangs völlig "dumm", da die Parameter erst noch angepasst werden müssen. Indem man bewusst alle Werte auf einen zufälligen Wert setzt, stellt man sicher, dass das Netz später tatsächlich in der Lage ist, mathematische Lernfunktionen auszuführen. Würden alle Parameter den gleichen Wert speichern, so würde das Lernverfahren höchstwahrscheinlich fehlschlagen, da es (vereinfacht ausgedrückt) mit Unterschieden zwischen benachbarten Parametern arbeitet. Außerdem könnten durch von Menschen vorgegebene Parameter fälschlicherweise erste Falschinformationen im Netz "gespeichert" sein.*
2. Anschließend gibt man dem Netzwerk sogenannte Trainingsdaten. Das sind teils riesige Datensätze (häufig mehrere GB, bei Bilderkennung auch schnell mal TB), bei denen man den Input **und** den gewollten Output bereits kennt.
3. Das Netzwerk nimmt nun der Reihe nach alle Trainingsdaten und berechnet jeweils (anhand der anfangs zufälligen und völlig falschen Parameter) den Output. Dieser Output wird zu Beginn noch kolossal vom gewollten Output abweichen.
4. Dementsprechend vergleicht das Netz pro Ergebnis den **berechneten Output** mit dem **erwarteten Output** und passt die variablen Werte im Netz (insb. die Gewichte) entsprechend an. Diese Anpassung läuft nun **innerhalb des Netzes rückwärts zurück zu den Eingabeneuronen**. Daher spricht man auch vom **Backpropagation of Error**: der Fehler bzw. die Abweichung wird zurückgereicht.

Während wir bisher hauptsächlich bei einem fertig trainierten Netz links Input reingesteckt haben und rechts ein Ergebnis rausbekamen …




… läuft es beim Lernvorgang auch andersherum und der Fehler bzw. die Abweichung wird im Netz zurückgereicht.




# Berechnung für die Anpassung der Gewichte

Folgende Rechnung wird pro Ergebnis und pro "links-benachbartem" Gewicht berechnet, um jedes einzelne Gewicht stückchenweise anzupassen. *Hinweis: diese Gleichung ist vergleichsweise simpel und funktioniert daher nur in einfachen Netzwerken **ohne** Hidden-Layer. Für größere Netzwerke sind die Gleichungen ungleich komplizierter und würden den hiesigen Rahmen sprengen.*

$$w_{neu} = w_{alt} + l \cdot (e - y) \cdot x_i$$

Mit:

$w_{neu}$ und $w_{alt}$ → Neu berechnetes bzw. bisheriges Gewicht

$l$ → Lernrate (gibt an wie schnell das Netz lernen soll)

$e$ und $y$ → Das erwartete Ergebnis und das berechnete Ergebnis y zum jeweiligen Input

$x_i$ → Input

**Arbeitsblatt 3)**

Bearbeite jetzt das folgende Aufgabenblatt 3 und wende darin diese Gleichung an. Dir liegen zwei kleine Netzwerke vor, die mit zufälligen Werten initialisiert wurden. Nach wenigen Trainingsdurchläufen soll das erwartete Ergebnis aus der Tabelle mithilfe des Netzwerks berechnet werden können.

ab_3_-_maschinelles_lernen.pdf

Lösung:

a) $w_1 = -0,3$ und $w_2 = 0,3$
b) $w_1 = -0,4$ und $w_2 = 0,7$

| | | |
|---|---|---|
| ab_3_-_maschinelles_lernen.pdf | 153.1 KiB | 11.09.2023 18:17 |
| ab_3_-_maschinelles_lernen_-_loesung.pdf | 154.1 KiB | 11.09.2023 18:17 |

[1)]

Quelle siehe: Einführung in Maschinelles Lernen mithilfe von MemBrain

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

Permanent link:
**https://info-bw.de/faecher:informatik:oberstufe:machine_learning:einfuehrung_maschinelles_lernen:start**

Last update: **12.09.2023 18:44**