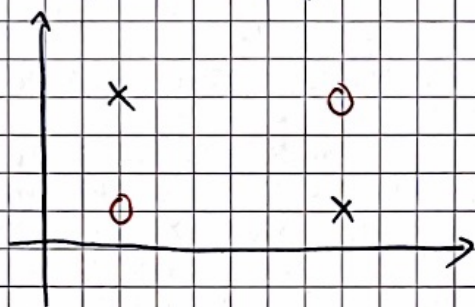


Mehrlagiges Perceptron

- = multilayer perceptron (MLP)
- besteht eigentlich aus mehreren Neuronen
- Perceptron hat eigentlich die Heaviside-Funktion als Aktivierungsfunktion
- MLP nimmt andere Aktivierungsfunktionen
- benötigt: XOR kann nicht mit einem Perceptron dargestellt werden



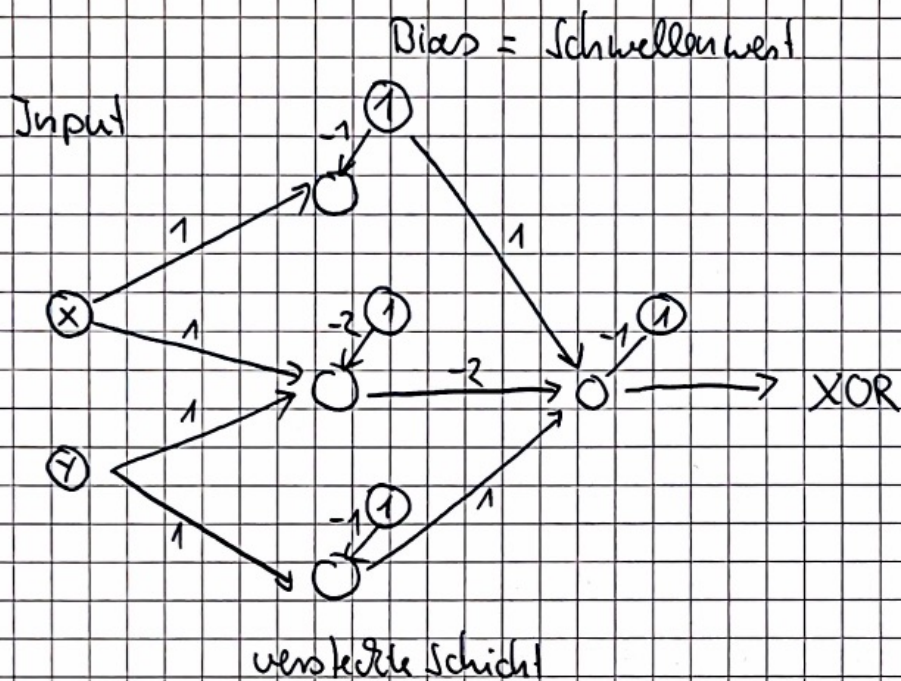
Perceptron kann hier keine Gerade legen um x und o zu trennen.

⇒ 1969 Minsky und Papert:

⇒ 20 Jahre keine Weiterentwicklung obwohl Netze schon in Diskussion waren

Perceptron kann nur Daten trennen, die linear separierbar sind

- Idee: Output von Neuronen wird Input von anderen Neuronen
 - versteckte Schichten
 - Insgesamt funktioniert es wie ein einzelnes Neuron



Verbindungen mit Gewicht 0 werden nicht gezeichnet.

Perceptron in Python:

```
p = Perceptron (0, 1, -1)
```

```
p.activate (2, 1) // Perceptron mit Daten versehen
```

```
p.activate (1, 2)
```

```
p1 = p3 = Perceptron (-1, 1) // oben und unten
```

```
p2 = Perceptron (2, 1, -1) // in der Mitte
```

```
q = Perceptron (-1, 1, -2, 1)
```

```
def xor (x, y):
```

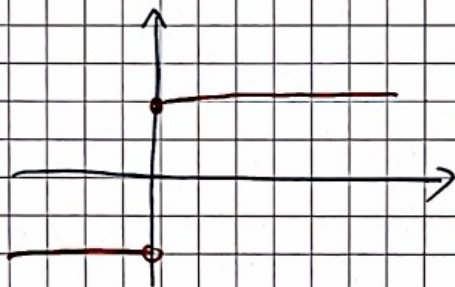
```
    return q.activate (p1.activate (x), p2.activate (x, y),  
                      p3.activate (y))
```

```
xor (0, 0), xor (0, 1), xor (1, 0), xor (1, 1)
```


Universal approximation theorem

NN mit nur 1 versteckter Schicht (beliebig viele)
dann kann dieses NN jede stetige Funktion
nachbilden, wenn Aktivierungsfunktionen
diffbar sind und Werte von 0 bis 1 haben für $x \rightarrow \pm \infty$

Betrachte Heaviside-Funktion:



\Rightarrow nicht stetig

\Rightarrow nicht diffbar

\Rightarrow nicht verwendbar

\Rightarrow logistische Funktion

> $x = \text{np.linspace}(-5, 5, 100)$

$$y = \frac{1}{1 + e^{-x}}$$

> $y = 1 / (1 + \text{np.exp}(-x))$

> $\text{plt.plot}(x, y)$

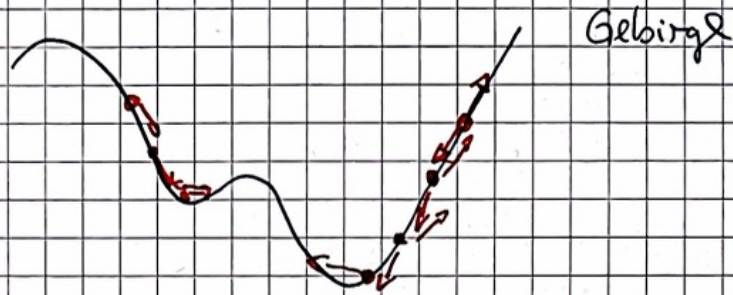
Mit Parameter lässt sich das Aussehen
verändern:

a) sehr steil \rightarrow Klassifizierungswert
da fast nur 0 und 1
rauskommt.

b) Regressionsnetzwerk, wenn sehr flach.

\Rightarrow Eine Aktivierungsfunktion für beide Netzwerke möglich einsetzbar.

Lernverfahren: Gradientenabstiegsverfahren



1. Suche, wo gehts am steilsten bergab
2. Stückweite ist Parameter; zu lange Stücke man schießt über das Ziel hinaus.
3. Gehe bergab

Problem



Problem: Lokale Minimum

\Rightarrow Startwerte der Gewichte müssen verändert werden

Backpropagation