

## Neurale netwerken

### Neuron - Brein



### Mogelijkheden

Herkenning en classificatie

### Supervised learning

Het netwerk leert door training.

1. Zet het artificiële neurale netwerk (ANN) op.

2. Herhaal:

Voer data in waarvan de uitvoer bekend is

Vergelijk de bekomen uitput met de gewenste uitput

Pas het ANN aan of laat het ANN zichzelf aanpassen

Doe dit tot je tevreden bent met de output

3. Voer echte data in

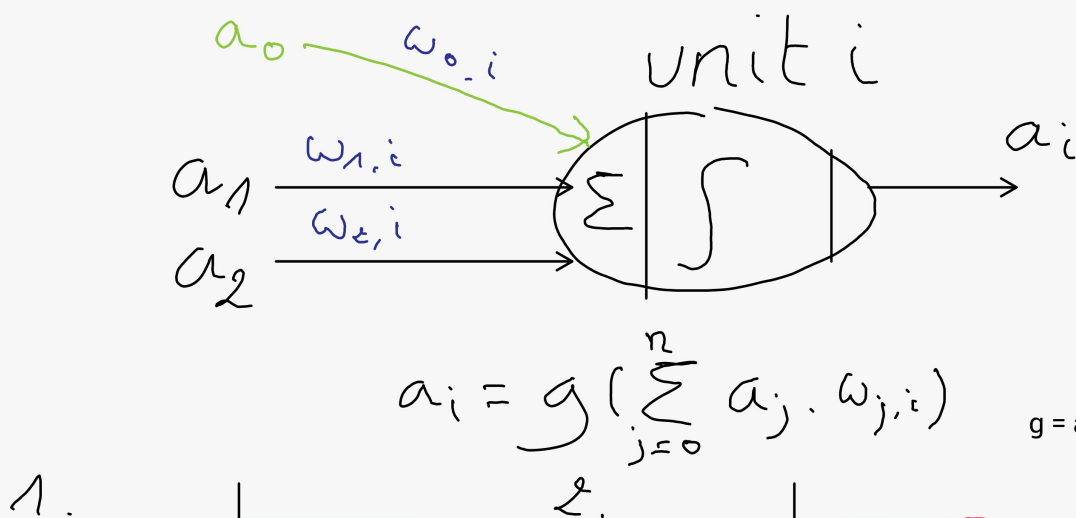
Grote test dataset nodig

Veel herhaling

Een ANN is een black box. De output kan niet verklaart worden.

In sommige gevallen is dit een probleem: toekennen van een lening.

### Neuron



bayeslijn

De bayeslijn komt niet altijd voor.



$g$  = activatiefunctie

- 1. threshold
- 2. sigmoid

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

## Netwerk structuren

- Feed forward netwerken
  - Single layer
  - Multilayer
- Recurrente netwerken

## Perceptron update rule

Uitvoer = functie van de invoer als de gewichten vast zijn.

$h_{\omega}(\underline{x}) = g(\sum_{i=0}^n x_i \omega_i)$       onderstreept wil zeggen dat het uit meerder componenten bestaat.

Stel  $\underline{x} = (x_0, x_1, \dots, x_n)$  gegeven met gewenste uitvoer  $y$ .

1. Als  $y = h_{\omega}(\underline{x})$

Niets doen

2. Als  $y = 1$ , maar  $h_{\omega}(\underline{x}) = 0$

$\sum x_i \omega_i$  is te klein

1. Als  $x_i > 0$ , dan  $\omega_i$  groter maken

2. als  $x_i < 0$ , dan  $\omega_i$  kleiner maken

$\omega_i \leftarrow \omega_i + \alpha (y - h_{\omega}(\underline{x}))$       Perceptron learning/update rule

$\alpha$  zorgt ervoor dat we niet te bruuske bewegen en is typische zeer klein ca. 0.1

3. Als  $y = 0$  maar  $h_{\omega}(\underline{x}) = 1$

## Backpropagation algoritme

1. Bereken netwerk van links naar rechts

2. Verdeel de fouten van rechts naar links

Als alle fouten gekend zijn pas je de gewichten aan