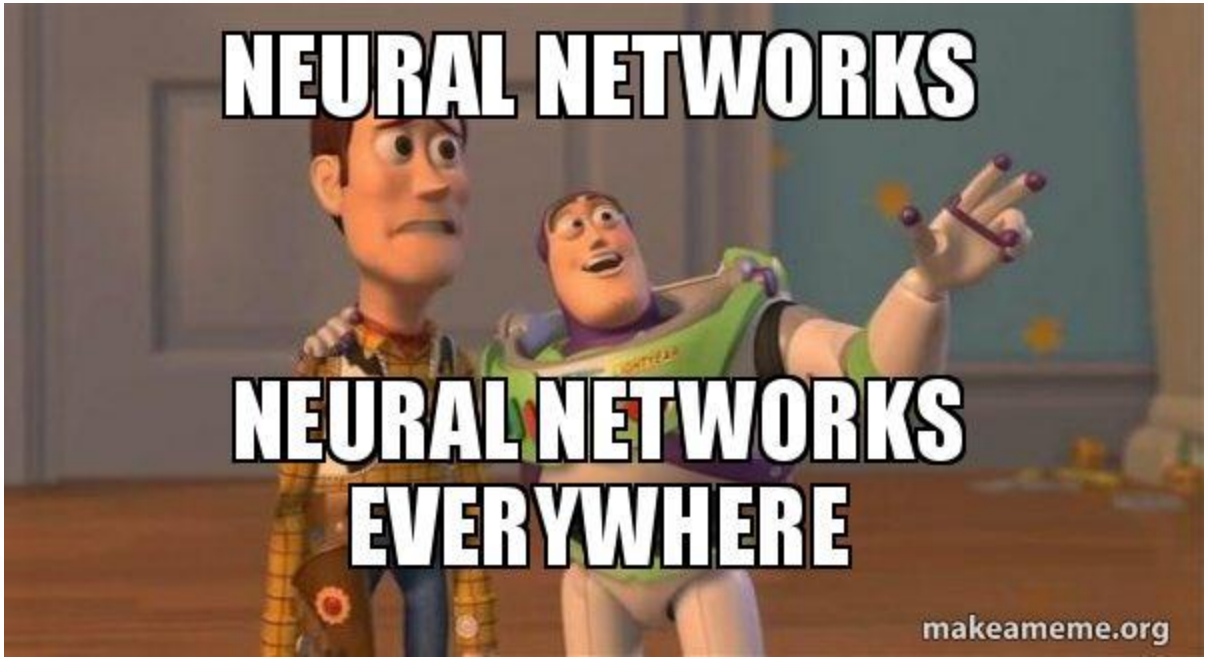
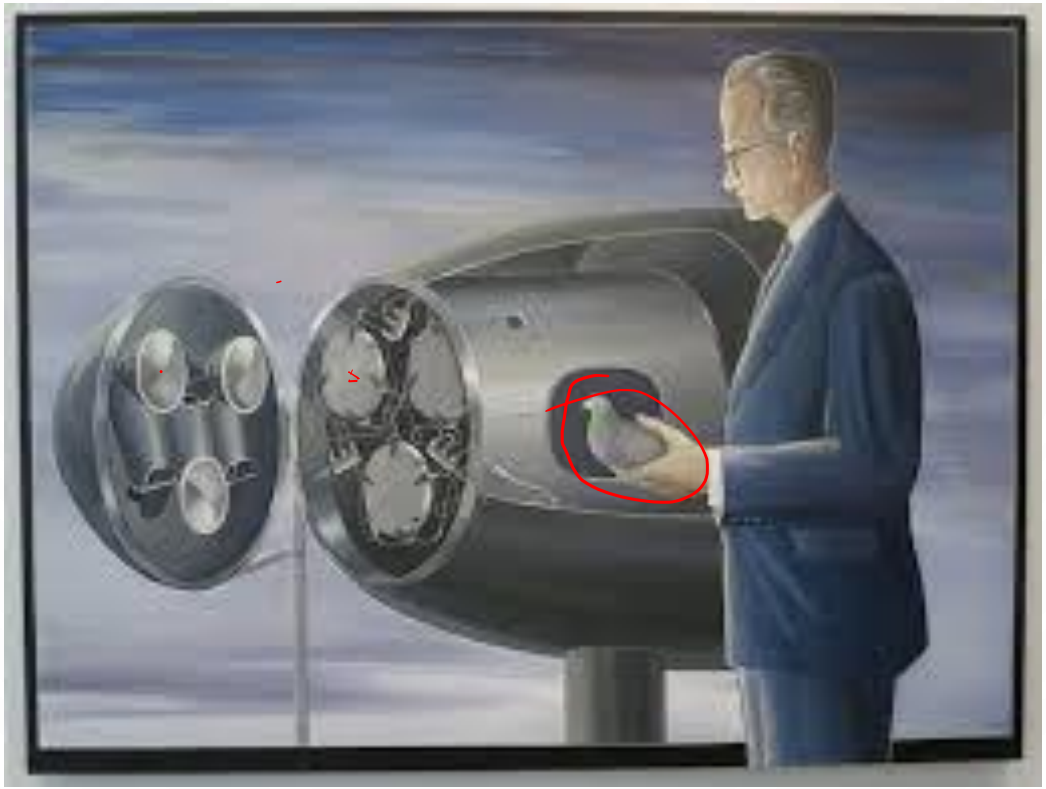
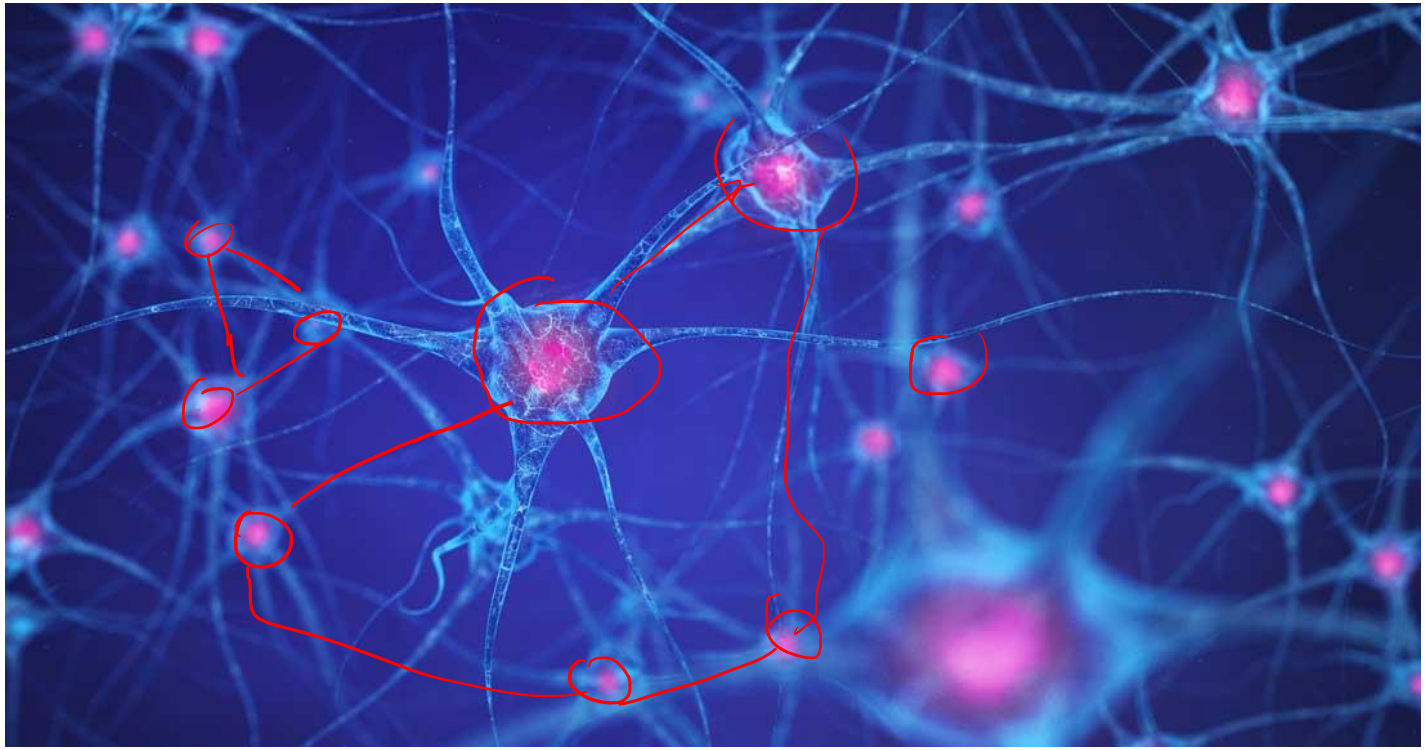
NEURAL NETWORK

JENS BAETENS

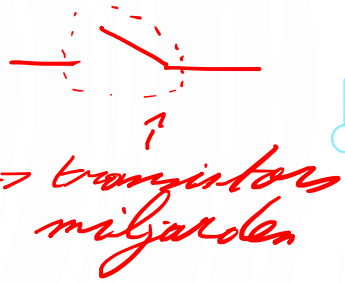

NEURAL NETWORKS
NEURAL NETWORKS
EVERYWHERE
makeameme.org

Biologische computer / Neuraal Netwerk

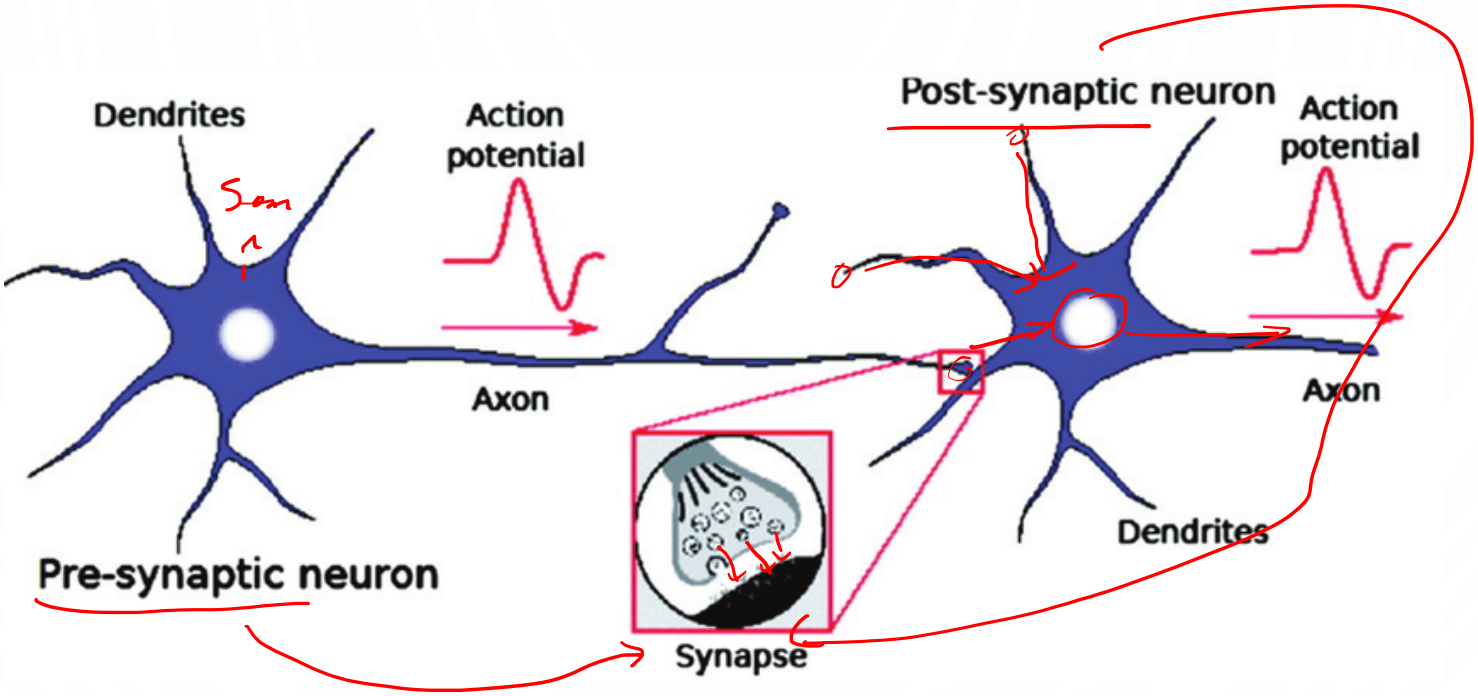




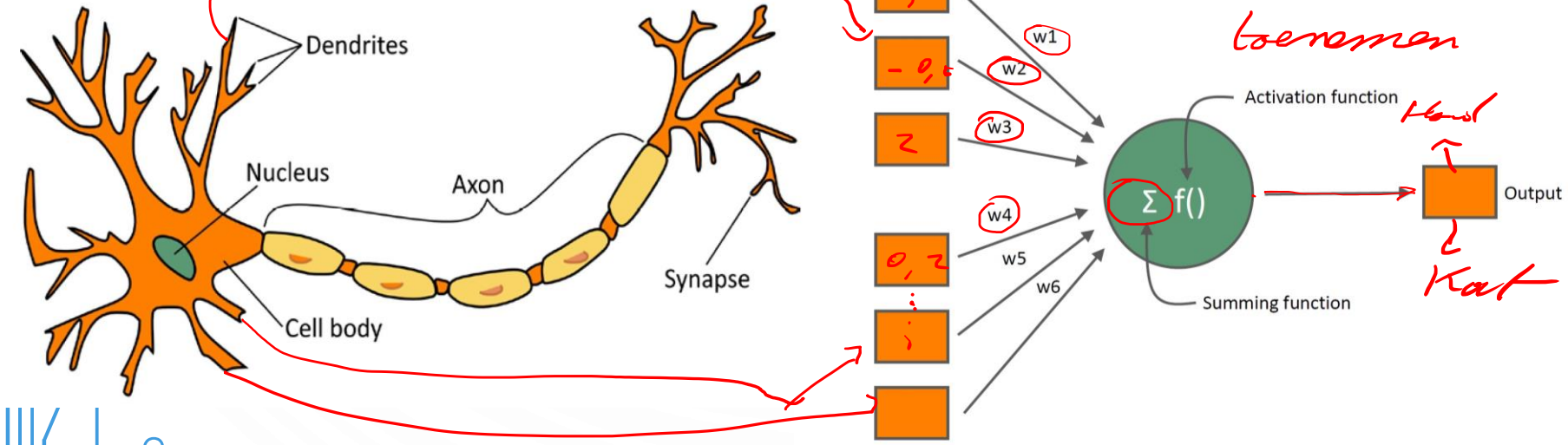
	Brain	Computer
number of processors	≈ 10 billion *neurons* (massively parallel)	1 *CPU* (intrinsically serial)
processor complexity	simple inaccurate	complex accurate
processor speed	slow (millisec)	fast (nanosec)
inter-processor communications	fast (μsec)	slow (millisec)
learning mode	learn from experience	manual programming
failure robustness	many neurons die without drastic effect	single fault often leads to system failure
memory organization	content addressable (CAM)	location addressable (LAM)



MENSELIJK NEURON



ARTIFICIEEL NEURON/PERCEPTRON



ARTIFICIEEL NEURON/PERCEPTRON

Summing function

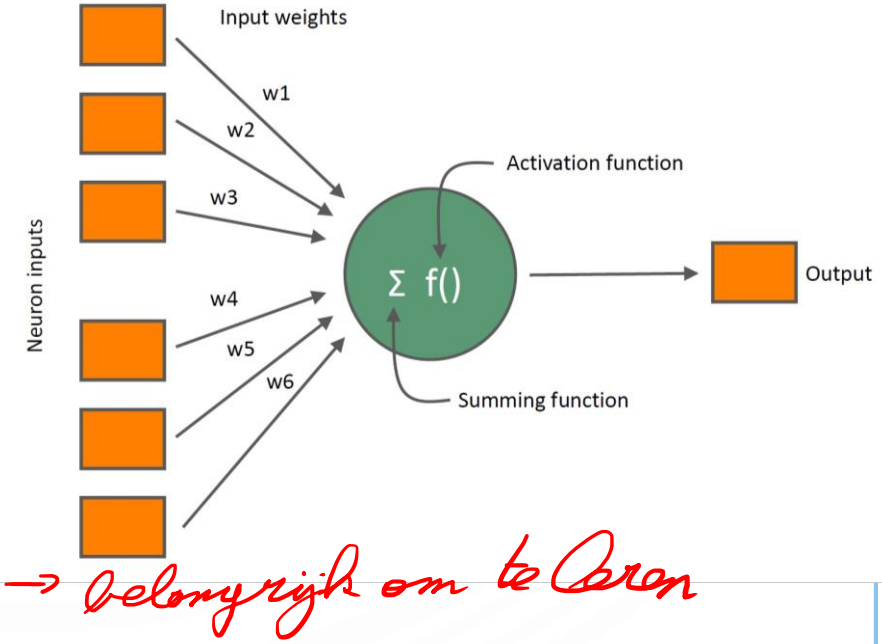
$$\mu = w_1 * x_1 + w_2 * x_2 + \cdots + w_6 * x_6$$

Activation function (bvb sigmoid)

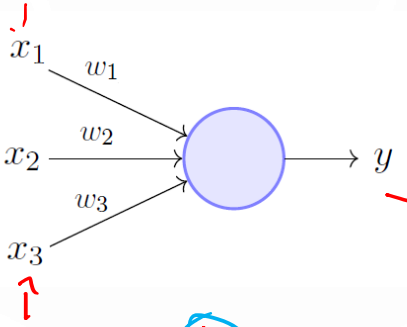
$$y = f(\mu) = \frac{1}{1-e^{-\mu}}$$

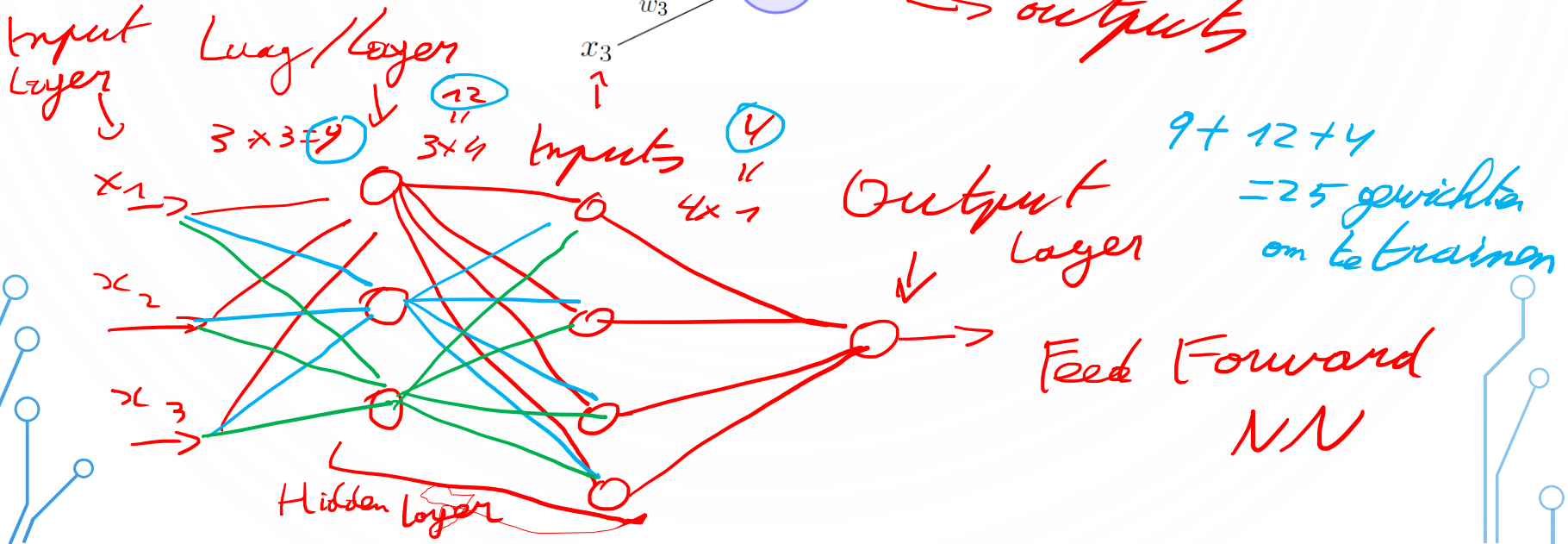
$$f(x) = \frac{1}{1-e^{-x}}$$

↳ afgeleidbaar te bepalen → belangrijk om te leren



ARTIFICEEL NEURON/PERCEPTRON





NETWERK OF NEURONS

Te kiezen (hyperparameters)

- Architectuur van het network
- Hoe leert het network
- Activatiefunctie

FF, RNN, ...

#lagen

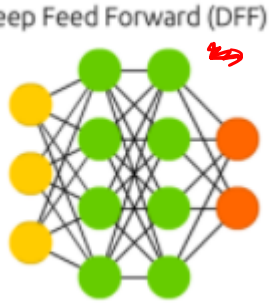
#neuronen per laag

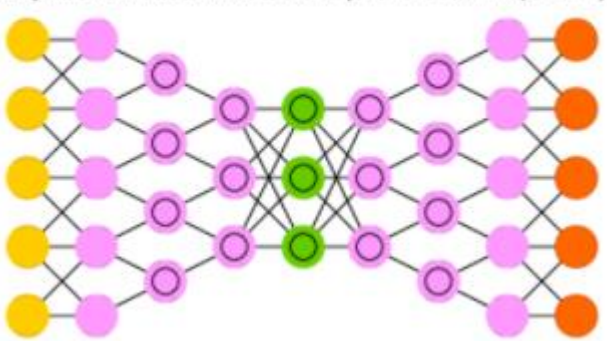
→ optimizers → Stanford is OK.

$f(\mu) = \frac{1}{1-e^{-\mu}}$ (sigmoïd)

NETWERK ARCHITECTUUR

https://www.asimovinstitute.org/neural-network-zoo/





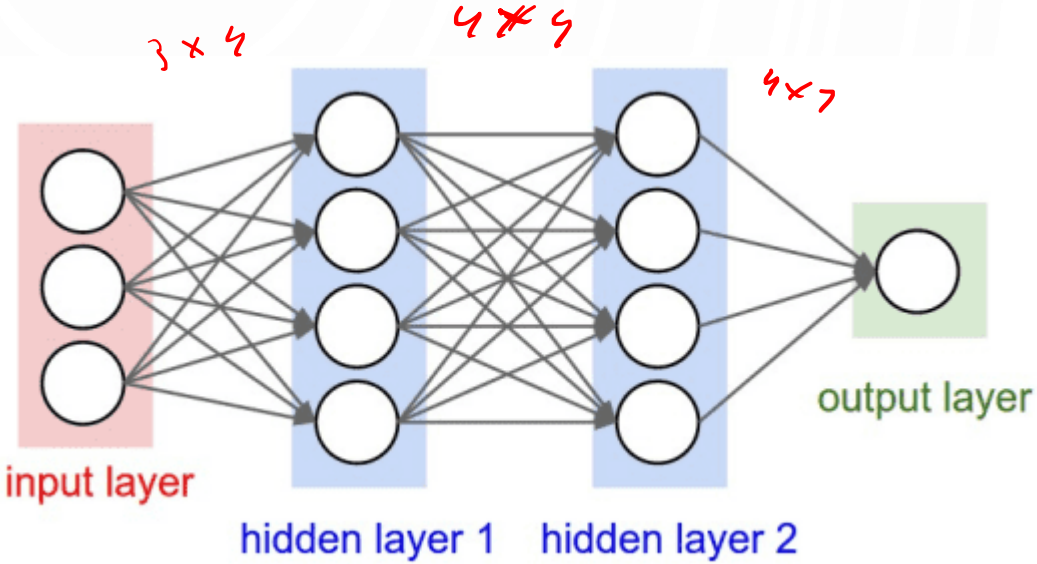
FEEDFORWARD NEURAL NETWORK

Hoeveel gewichten?

3 2

Informatie gaat van links naar rechts

- FeedForward

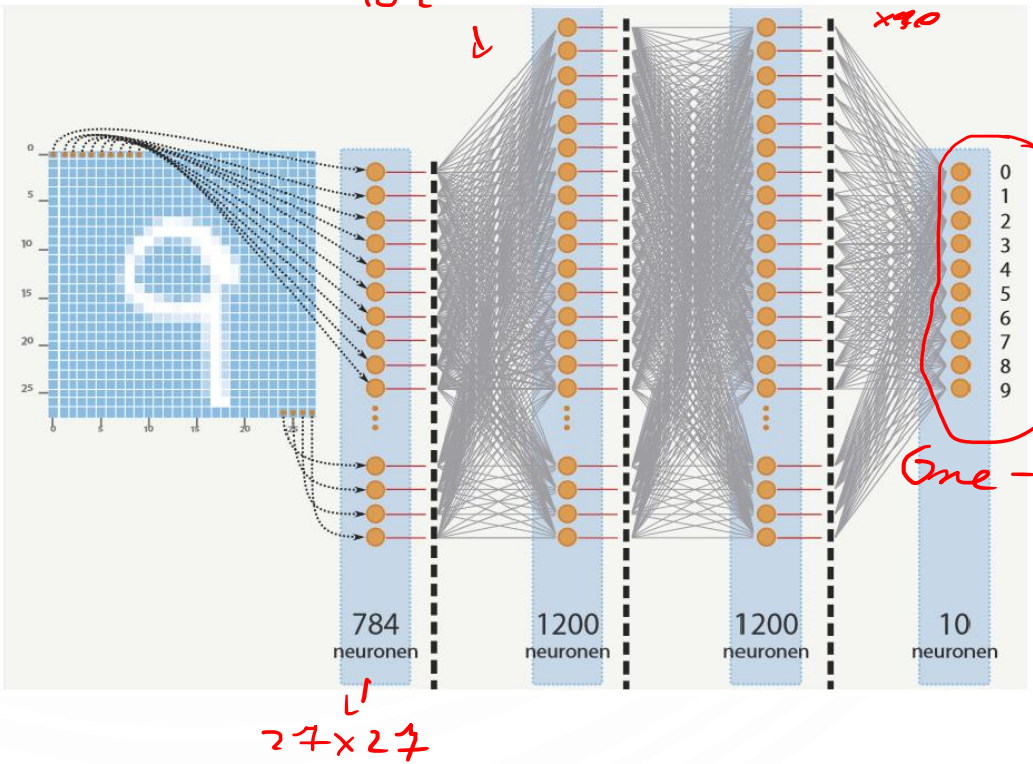


FEEDFORWARD NEURAL NETWORK

n_inputs = n_features

One-hot encoding

Prob of each class



BACKPROPAGATION

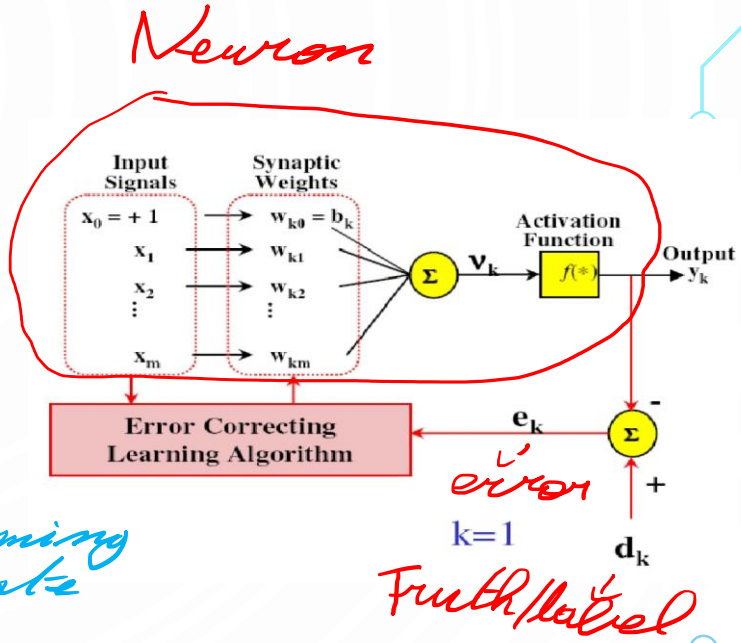
Omgekeerde beweging van FeedForward

Gradient Descent Algoritme gebaseerd op de fout

Wiskunde:

https://medium.com/@14prakash/back-propagation-is-very-simple-who-made-it-complicated-97b794c97e5c

https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

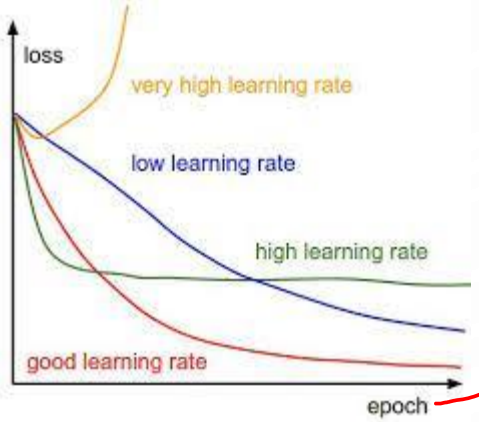


BACKPROPAGATION

Error functie

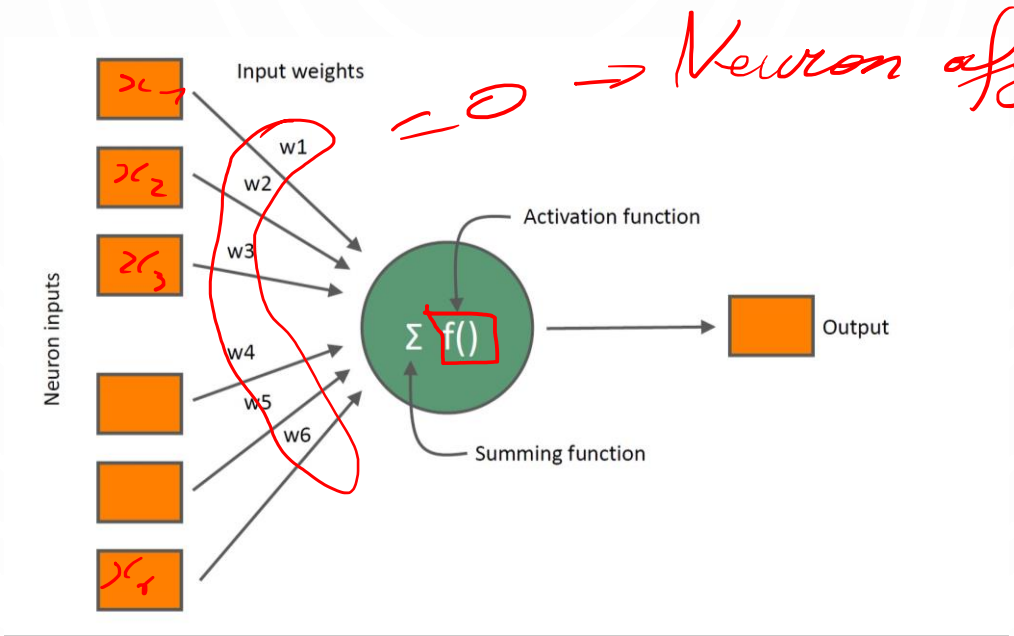
- Afhankelijk van de learning rate (automatisch = ADAM)
- Epochs = aantal keer volledige trainingsdata gezien

Data augmentation
-> spiegelen
-> roteren
-> inzoomen
-> verplaatsen

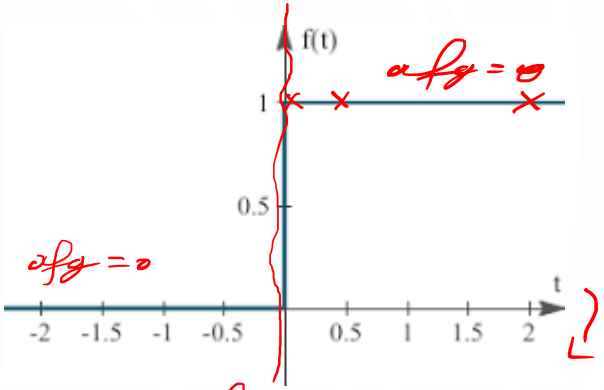


keer VOLLEDIGE dataset door het systeem

ACTIVATION FUNCTION



STEP FUNCTION



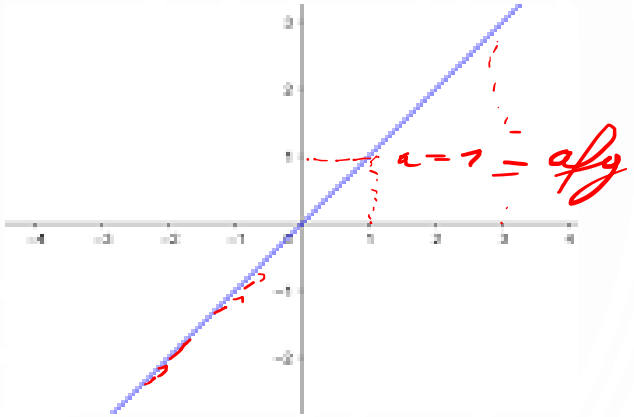
X > 0 => output = 1

X < 0 => output = 0

Nadelen:

- Enkel ja of nee
- Geen indicatie hoe verkeerd of correct het neuron is
- Backpropagation werkt niet (afgeleide is 0)
- Meerdere klasses die op 1 staan, welke is het dan?

LINEAIRE FUNCTION



Output = X*a

Nadelen:

- Enkele lineaire scheidingen mogelijk
- Afgeleide is constant en geen relatie meer met de ingang

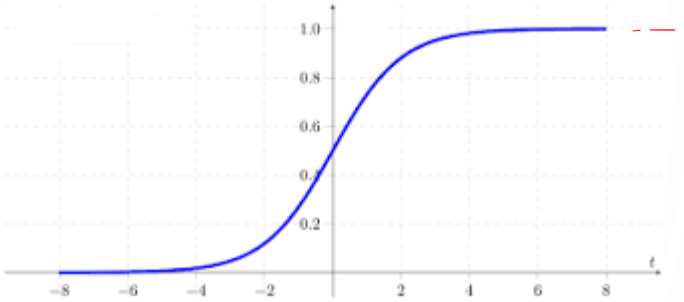
Gebruikt voor:

- Input layer (a = 1)
- Output layer voor regressie uit te voeren →
  ↳ kwadratische verbanden door hidden layer(s)

SIGMOID FUNCTION

↳ benadering stap-functie

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



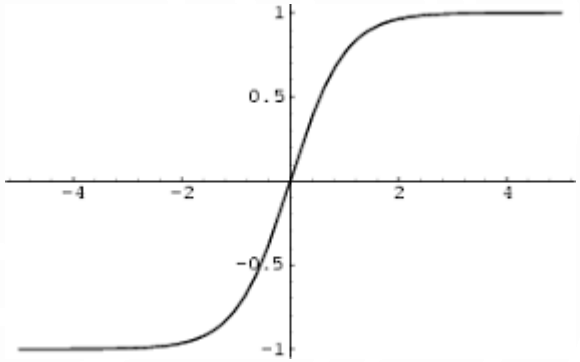
Nadelen:

- Vanishing gradient problem (Afgeleidde gaat naar 0 bij grote |x|)
  - Vooral problem bij veel hidden layers
- Rekenintensief

Gebruikt voor:

- Soms in output layer voor classificatie

HYPERBOLIC TANGENT



$$f(x) = \tanh(x)$$

Nadelen:

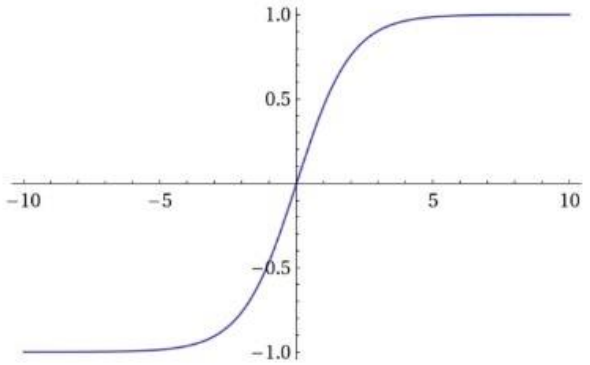
- Vanishing gradient problem
- Rekenintensief

Gebruikt voor:

- Zelden maar kan voor output in classificatie

SOFTMAX FUNCTION

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

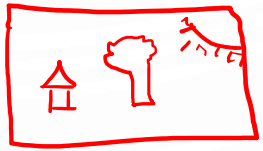


Generalisatie van sigmoid

Gebruikt voor:

- In output layer voor classificatie
- Kan meerdere klassen tegelijkertijd aangeven

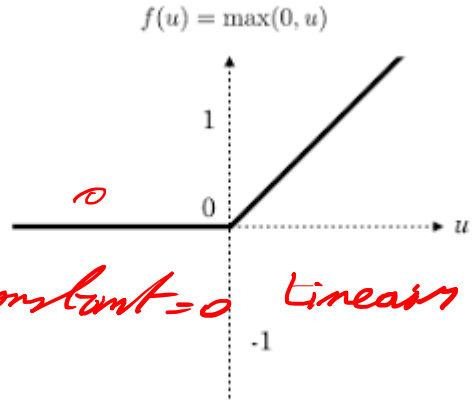
Hond of Kat



Huis
Boom
Zon

Multi-label classificatie

RECTIFIED LINEAR UNIT (RELU)



$$f(x) = \max(0, x)$$

Voordelen:

- Kan elke functie benaderen
- Heel rekenefficient
- Sparse activation

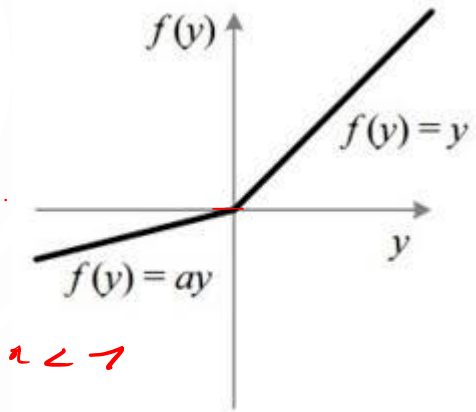
Nadelen:

- Dode Neurons blijven dood

Gebruikt voor:

- Hidden layers

LEAKY RELU



Voordelen:

- Neurons gaan niet dood

Nadelen:

- Meer parameters om te trainen

Gebruikt voor:

- Hidden layers

ACTIVATION FUNCTIONS - SAMENVATTING

Hidden layers

- Eerste relu, indien problemen leaky relu
- Geen sigmoid of tanh

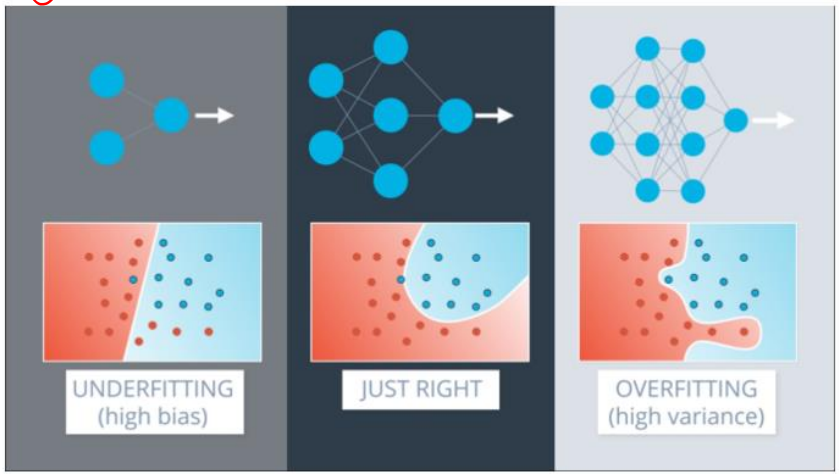
Output layer

- Regressie: Lineaire activation function
- Classification:
  - Slechts 1 klasse tegelijkertijd: Sigmoid
  - Meerdere klassen tegelijkertijd: Softmax

UNDERFITTING EN OVERFITTING

Te groot netwerk/ niet voldoende data -> overfitting -> Meer data

Te klein netwerk -> underfitting -> Complexer netwerk



WEIGHT REGULARISATION

Straf gebruik van hoge gewichten af zodat er gefocust wordt op de belangrijke

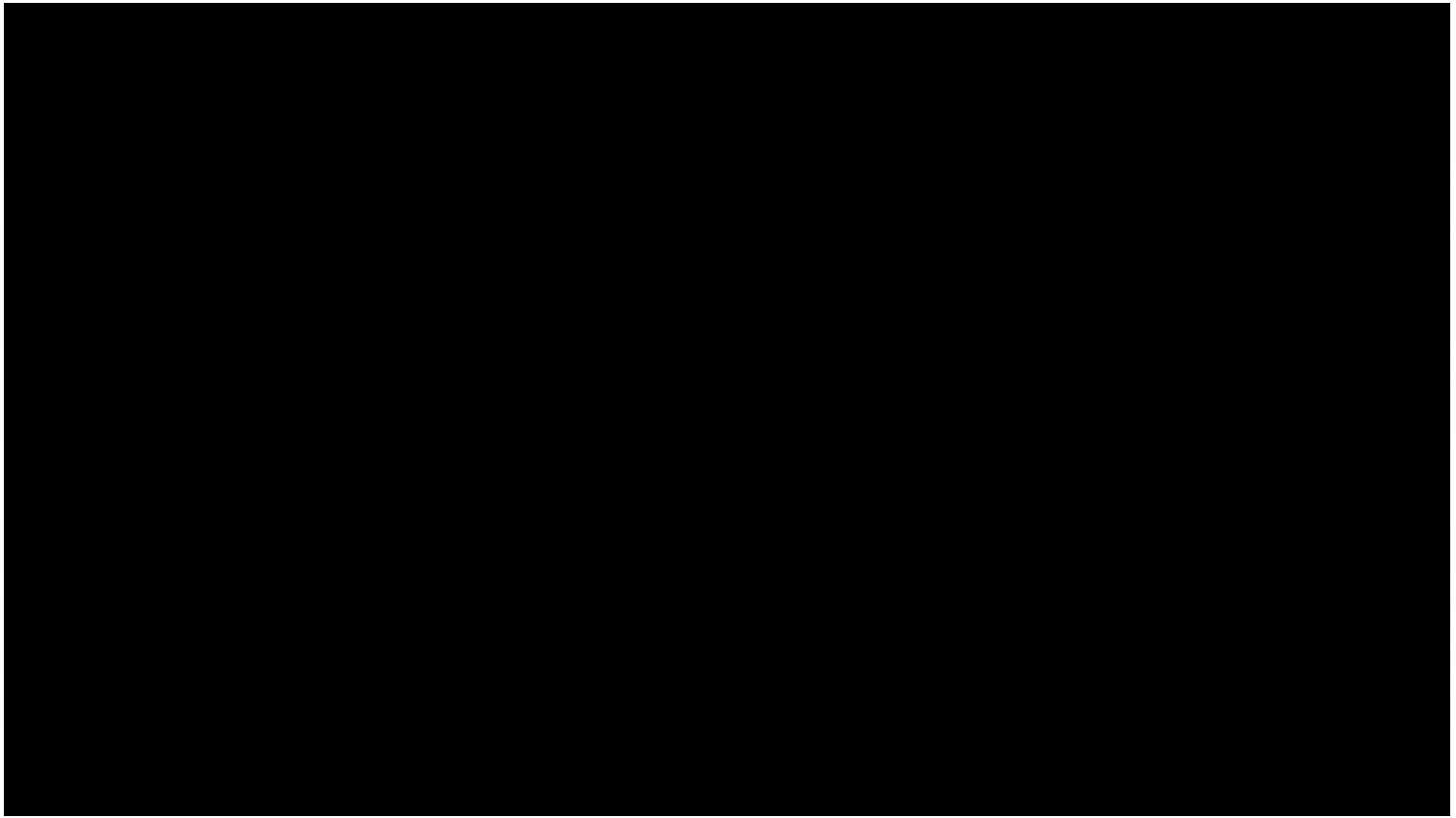
Extra kost op basis van L2-norm of L1-norm.

Neuron : $|w_1| + |w_2| + |w_3| + |w_4| + |w_5| + |w_6| = V$ (L_1-norm)

$w_1^2 + w_2^2 + w_3^2 + w_4^2 + \dots = V$ (L_2-norm)

UNDERFITTING EN OVERFITTING - DROPOUT

bij training



https://www.youtube.com/watch?v=NhZVe50QwPM&t

UNDERFITTING EN OVERFITTING - DROPOUT

Techniek om overfitting te voorkomen

Willekeurig uitschakelen van neurons bij training

- Andere neurons moeten inspringen om een correct resultaat te geven

- Vermijd dat andere neurons afsterven

Bootst een ensemble van netwerken na wat het robuster en accurater maakt

HOE NEURAAL NETWERK IMPLEMENTEREN