Introduction to Neural Networks

Prof. Dr. Ronny Hartanto

HOCHSCHULE RHEIN-WAAL
Rhine-Waal University of Applied Sciences

Supervised Learning: Recap

Regression Example:

Suppose you receive a table of physical measurements:

Number	Temperature (°C)	Size (cm)
1	*0*	*0*
2	*1*	*1*
3	*3*	*2*

Use Linear Regression to predict the size at 2 °C!

Neural Networks

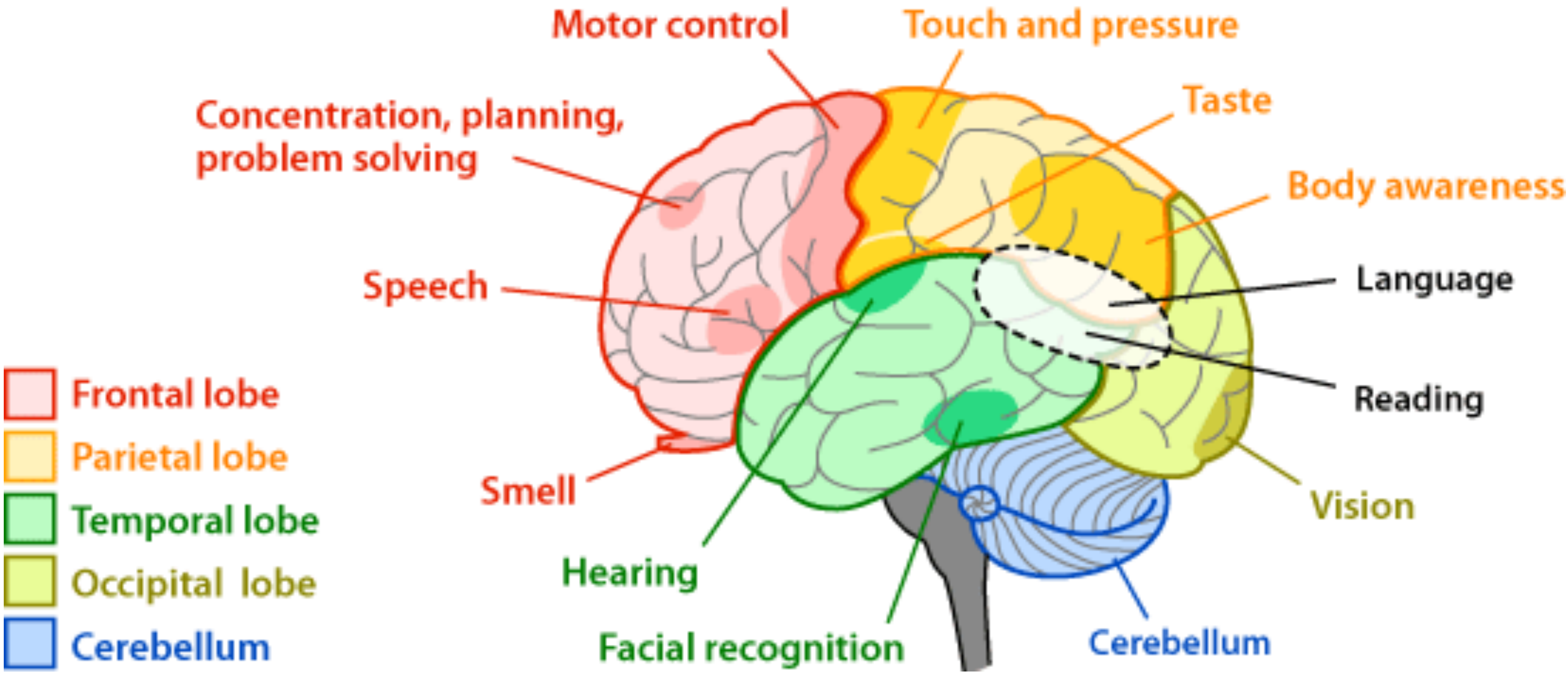
Bionics approach to learning:

1. Study the biological foundations of learning
2. Generate a model from these findings

What body parts are responsible for learning (in animals and humans)?

Are there differences in learning of visual/auditory/sensory information?

Neural Networks

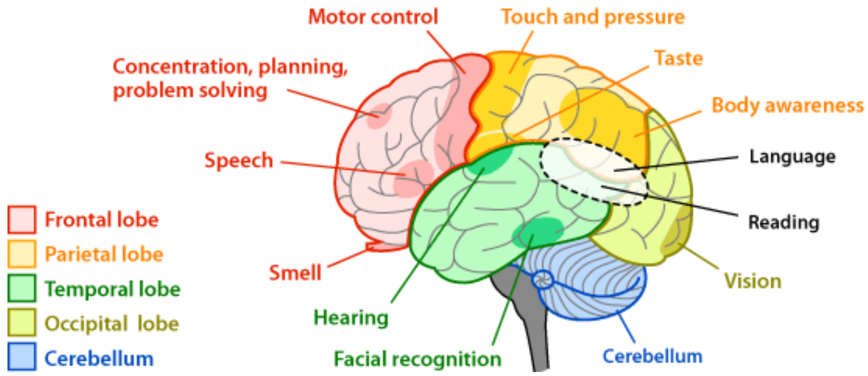


Source: Arizona State University, https://askabiologist.asu.edu/what-your-brain-doing

Neural Networks

"Re-wiring" experiment 1:

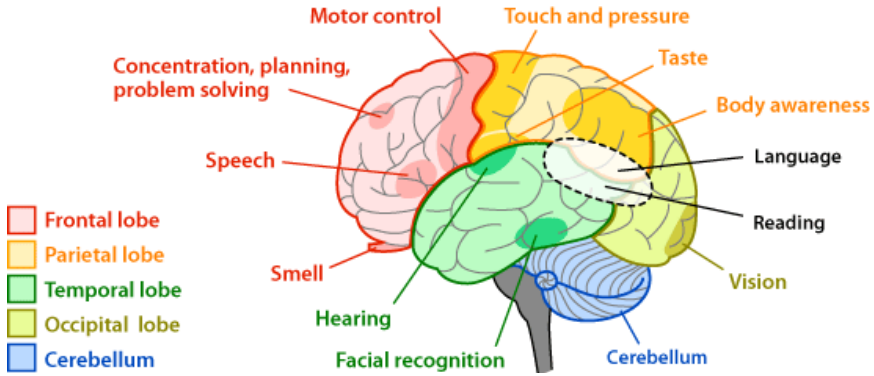
- Disrupt the connection from visual system to visual cortex
- When directing the visual input to the auditory cortex, the "re-wired" animal still learns to see
- [Roe et al 1992]



Neural Networks

"Re-wiring" experiment 2:

- Disrupt the connection from visual system to visual cortex
- When directing the visual input to the somatosensory cortex, the "re-wired" animal still learns to see
- [Metin/Frost1989]



Neural Networks

These findings suggest:

- Some parts of the brain are able to mimick other parts' functions.
- This is called *plasticity* of the brain.
- The learning algorithm in all parts of the brain appears to be the same.

Neural Networks

BrainPort V100

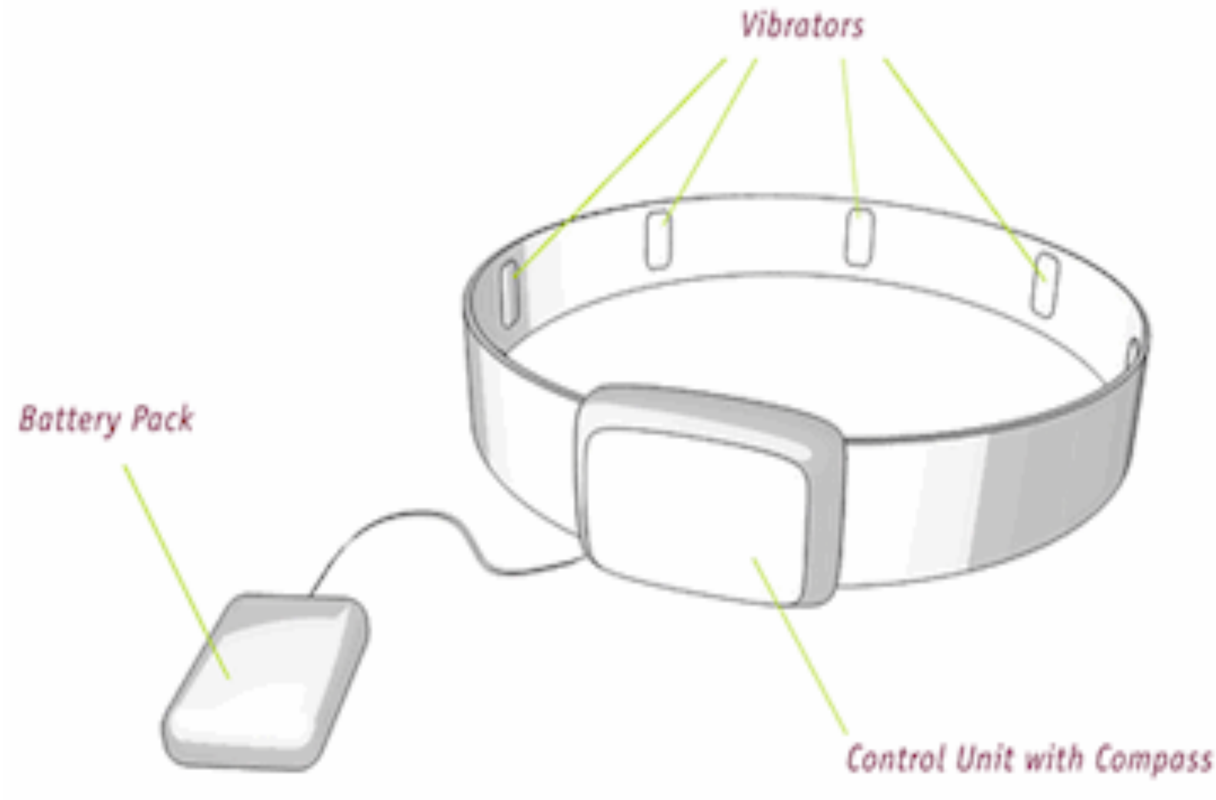
- Non-surgical assistive device for orientation, mobility, object identification
- Video Camera + electrical tongue stimulator
- Generates spatial patterns for shape, size, location and motion of objects

Source: BrainPort Technologies, http://www.wicab.com/en_eu/index.html

Neural Networks

feelspace magnetic perception:

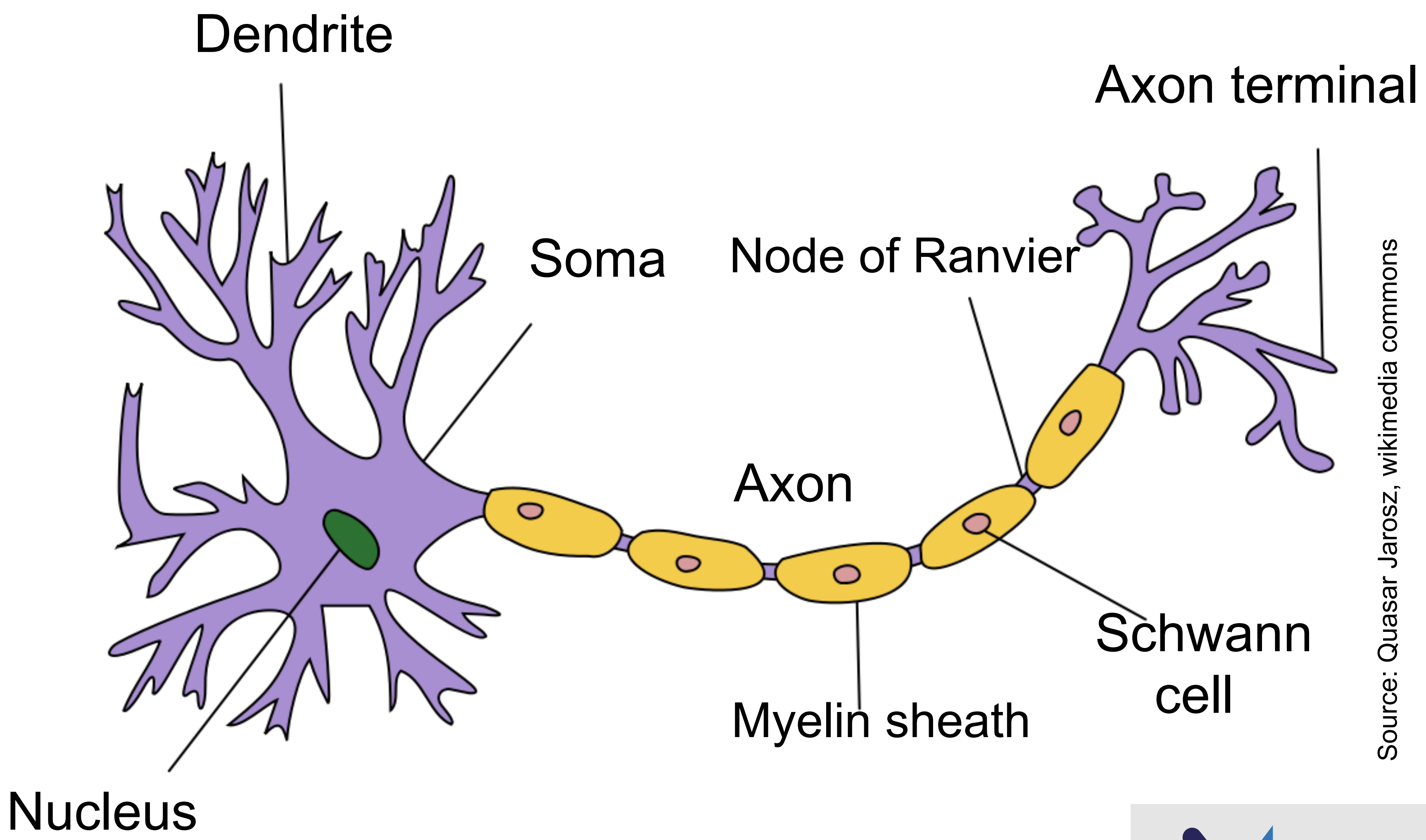
- Assistive device to augment perception with a magnetic sense
- vibrator near the magnetic north is activated
- users integrate this perception for motion planning and orientation



Source: http://feelspace.cogsci.uni-osnabrueck.de/

Neural Networks

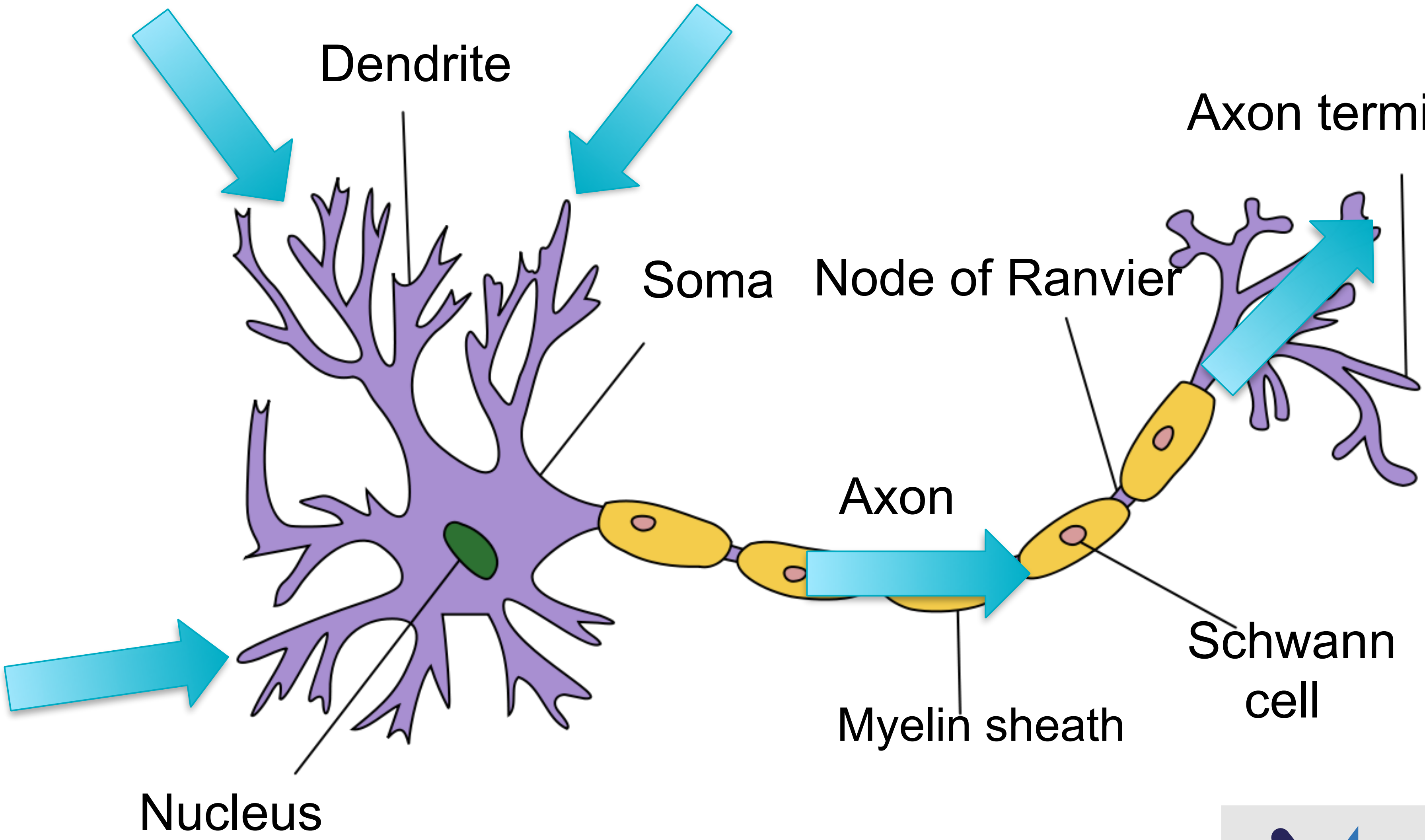
In all cases: *neurons* are the basic ingredient for learning



Source: Quasar Jarosz, wikimedia commons

Neural Networks

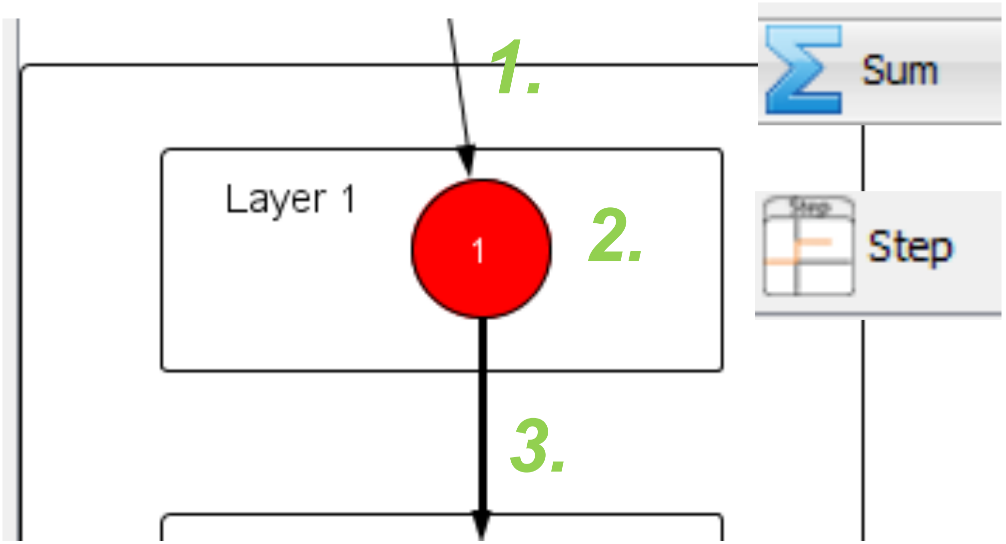
Signal transmission in neurons



Source: Quasar Jarosz, wikimedia commons

Neural networks

A simple model of the neuronal communication:

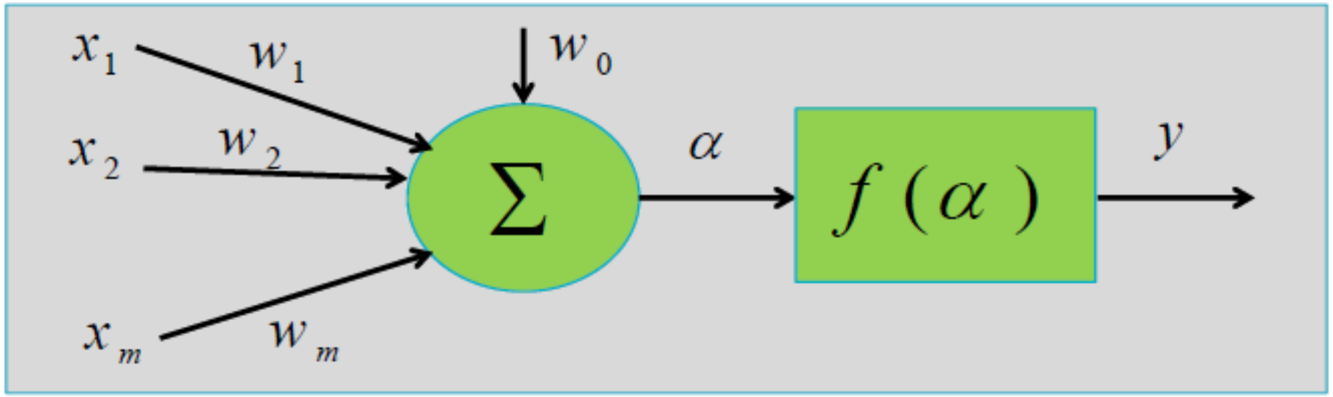


A neuron has

1. zero or more inputs (e.g. from other neurons), combined by an *input function*
2. a *transfer function*
3. an output (e.g. to other neurons)

Neural networks

A simple model of the neuronal communication:

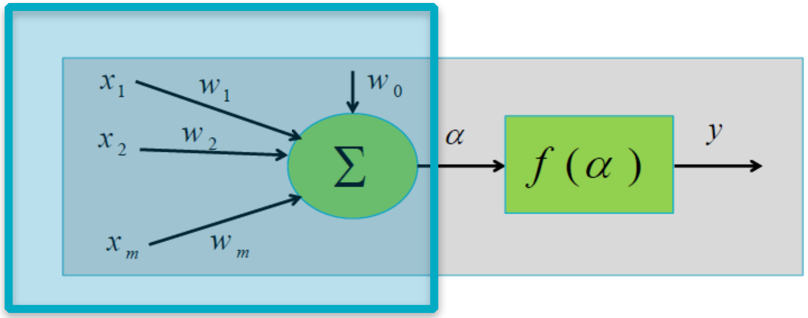


Source: P. Zhang

A neuron has

1. zero or more inputs (e.g. from other neurons), combined by an *input function*
2. a *transfer function*
3. an output (e.g. to other neurons)

Neural networks



Typical input functions:

Let $x = (x_1, \ldots, x_n)$ be the input values for the neuron.

- Sum

$$s(x) = \sum_{i=1}^{n} x_i$$

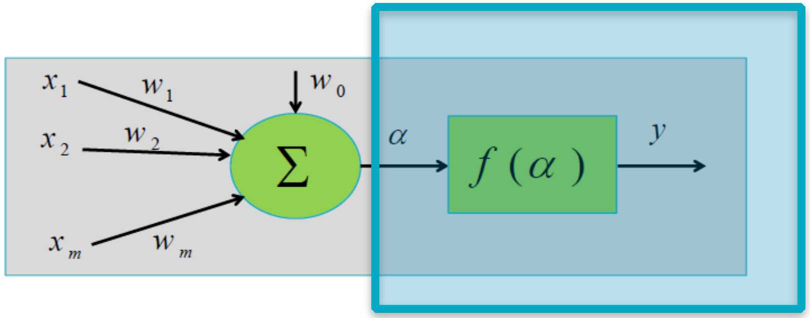
- Weighted sum (with fixed weights $\alpha_1, \ldots, \alpha_n$)

$$ws(x) = \sum_{i=1}^{n} \alpha_i x_i$$

- Squared sum

$$ss(x) = \sum_{i=1}^{n} {x_i}^2$$

Neural networks

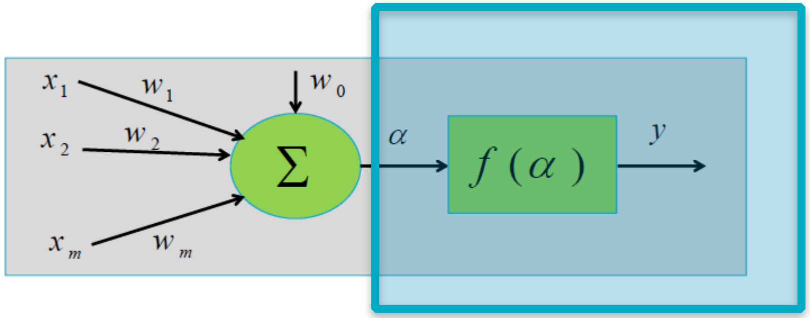


Typical transfer functions:

Let $\alpha = f(x)$ be the intermediate result from the input function.

- step function
- signum function
- ramp function
- linear function
- radial basis function
- sigmoid function

Neural networks

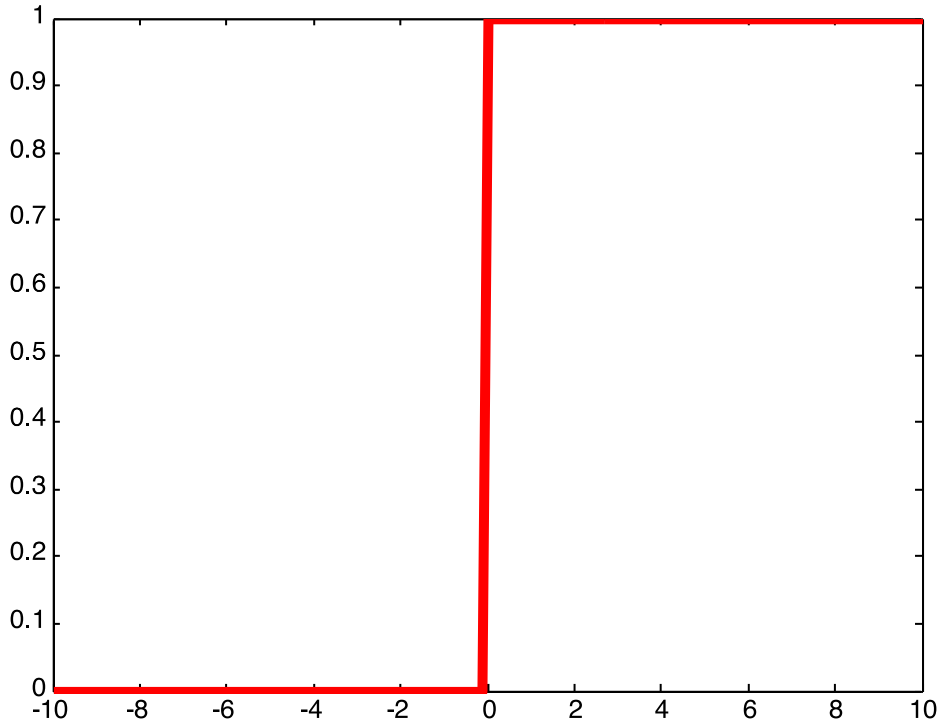


Typical transfer functions:

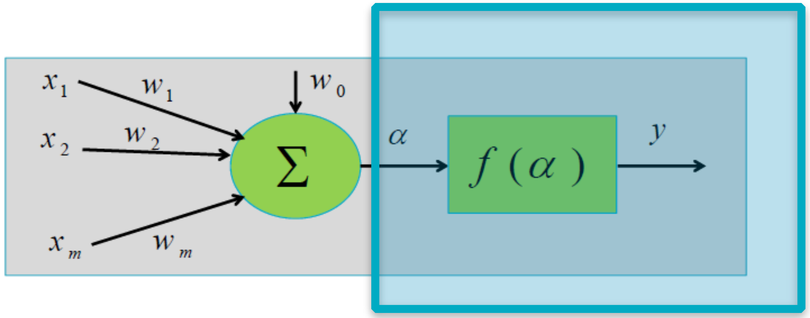
Let $\alpha = f(x)$ be the intermediate result from the input function.

- step function

$$y = \begin{cases} 0, & \alpha < 0 \\ 1, & otherwise \end{cases}$$



Neural networks

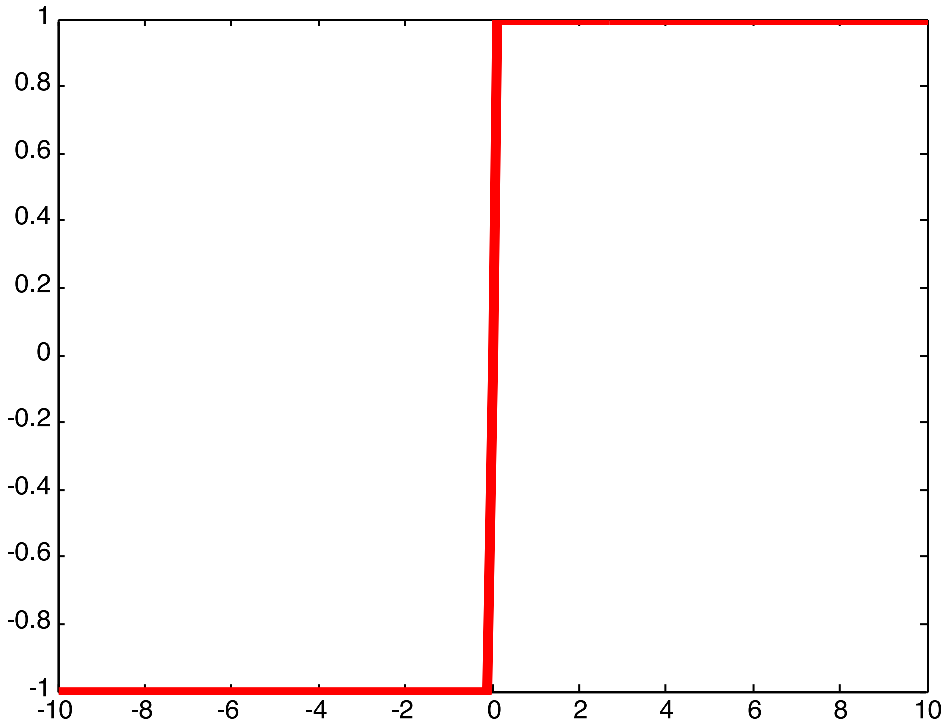


Typical transfer functions:

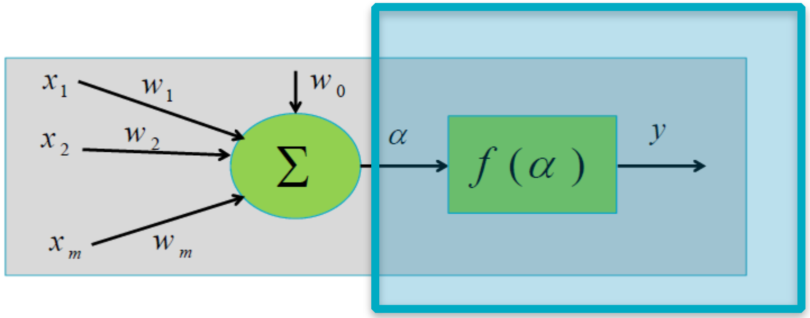
Let $\alpha = f(x)$ be the intermediate result from the input function.

- signum function

$$y = \begin{cases} -1, & \alpha < 0 \\ 1, & otherwise \end{cases}$$



Neural networks

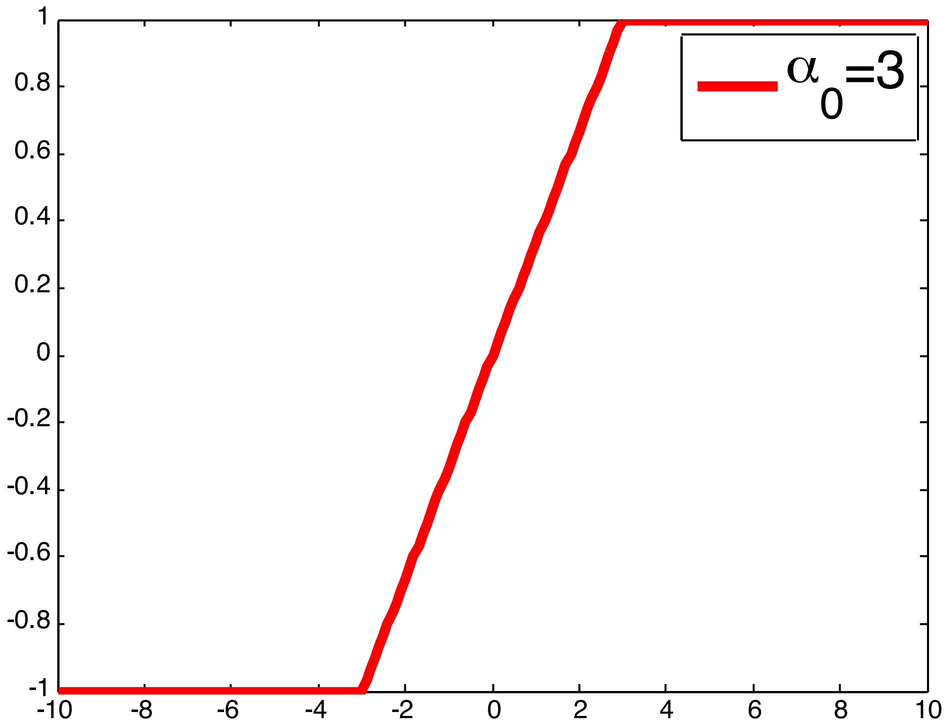


Typical transfer functions:

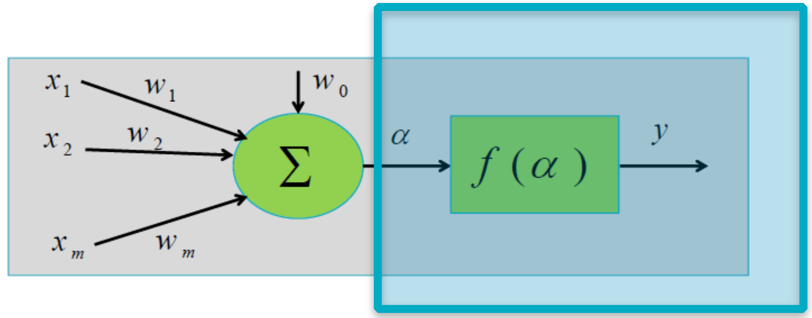
Let $\alpha = f(x)$ be the intermediate result from the input function.

- ramp function

$$y = \begin{cases} -1, & \alpha < -\alpha_0 \\ \dfrac{\alpha}{\alpha_0}, & -\alpha_0 \leq \alpha < \alpha_0 \\ 1, & \alpha > \alpha_0 \end{cases}$$



Neural networks

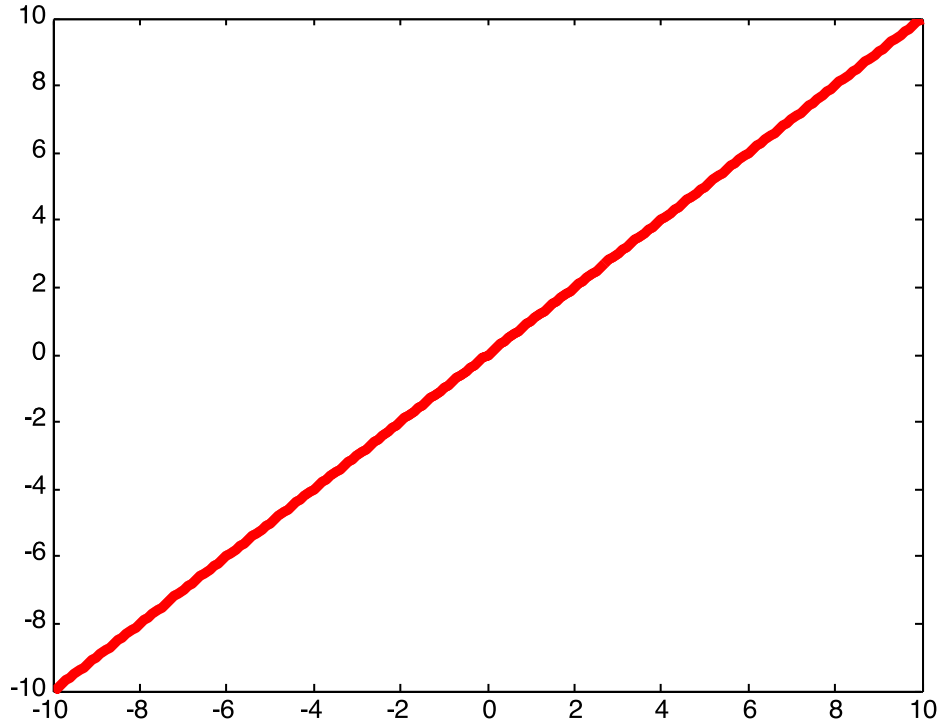


Typical transfer functions:

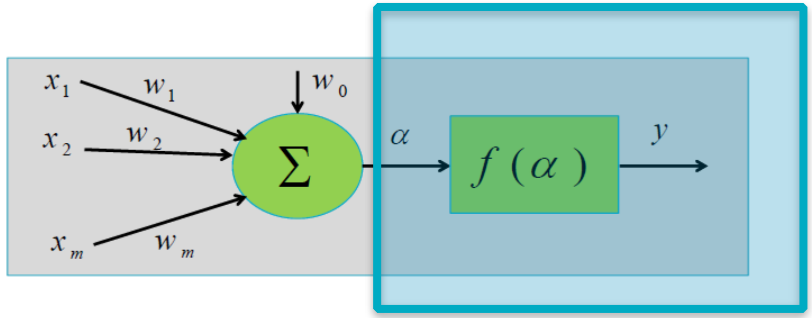
Let $\alpha = f(x)$ be the intermediate result from the input function.

- linear function

$$y = \alpha$$



Neural networks

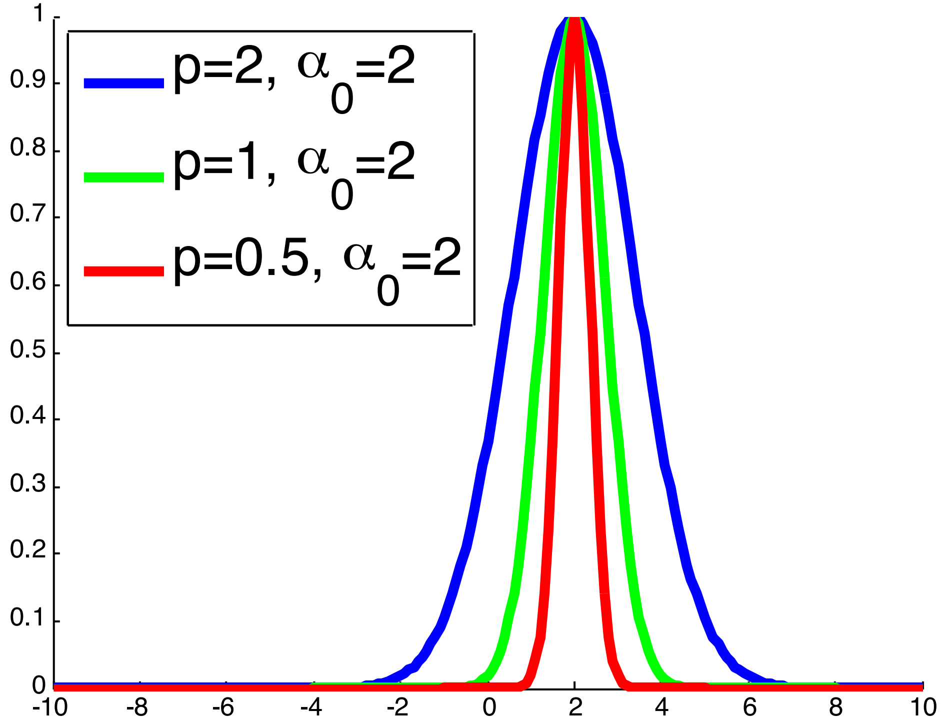


Typical transfer functions:

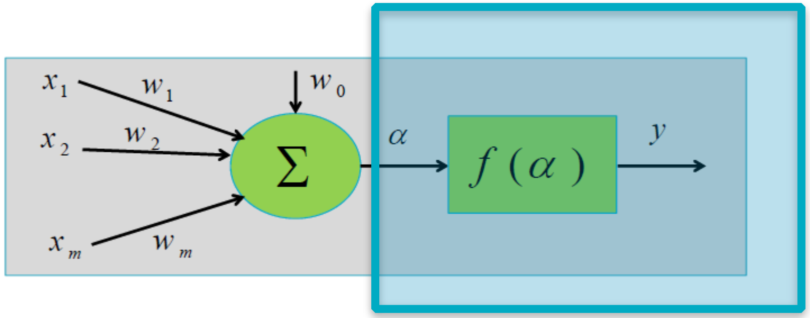
Let $\alpha = f(x)$ be the intermediate result from the input function.

- radial basis function

$$y = \exp(-\frac{(\alpha - \alpha_0)^2}{p})$$



Neural networks

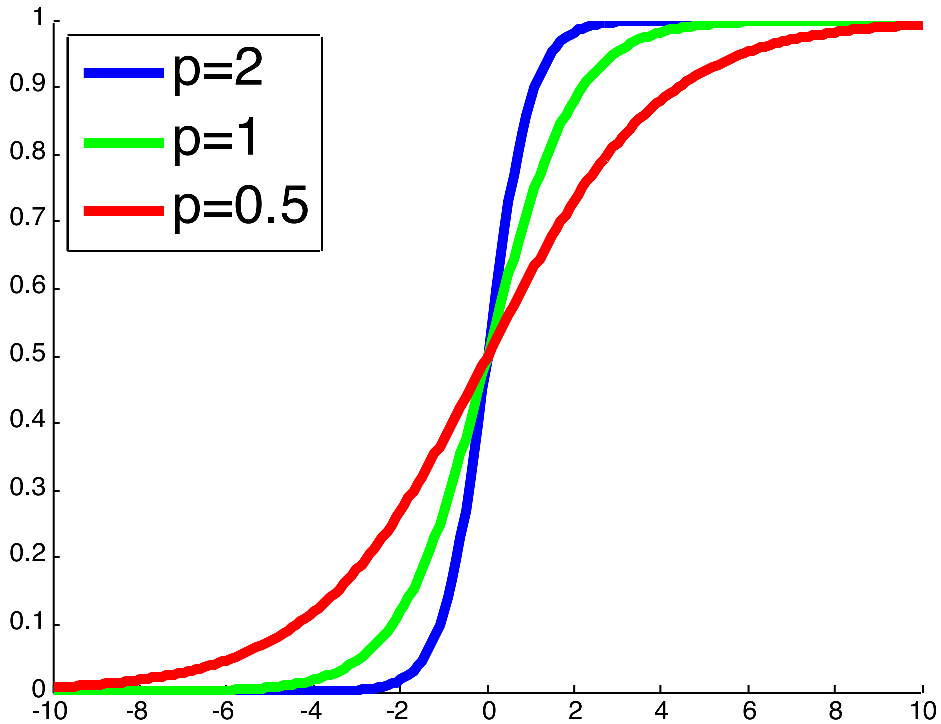


Typical transfer functions:

Let $\alpha = f(x)$ be the intermediate result from the input function.

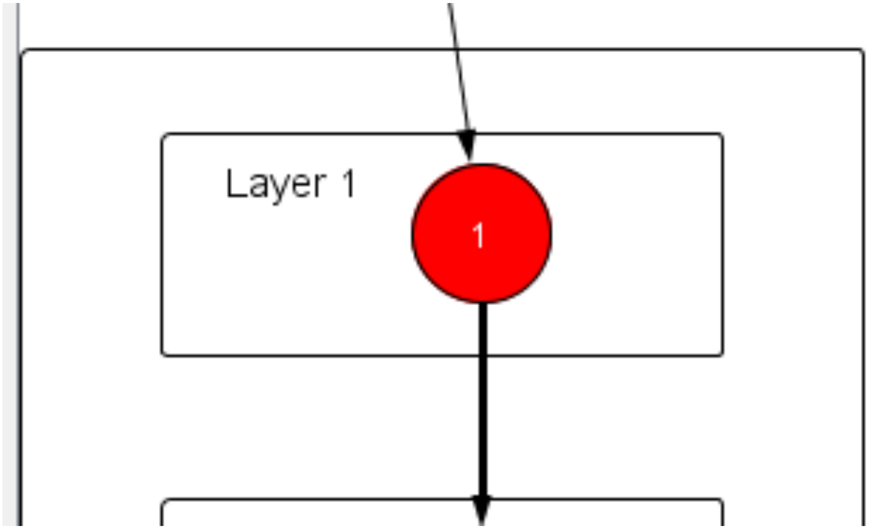
- sigmoid function

$$y = \frac{1}{1 + \exp(-p\alpha)}$$



Neural networks

Example:



Calculate the output to the following inputs:

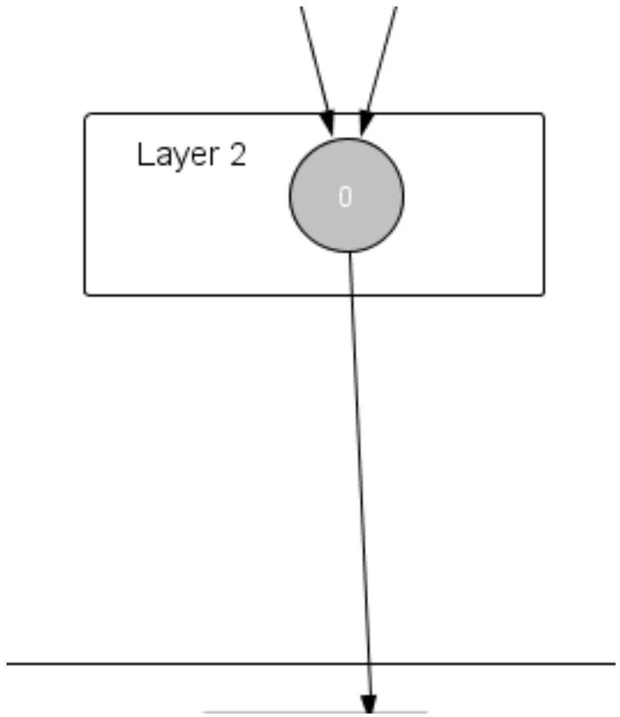
a) 0.5
b) 1.3
c) -3.1

if the neuron has

1. a single input, combined with the "sum" function
2. the step function as transfer function

$$step(\alpha) = \begin{cases} 0, & \alpha \leq 0 \\ 1, & otherwise \end{cases}$$

Neural networks



Example:

Calculate the output to the following inputs:

a) (0.5, 1.2)
b) (1.3, 0)
c) (-3.1, 1)

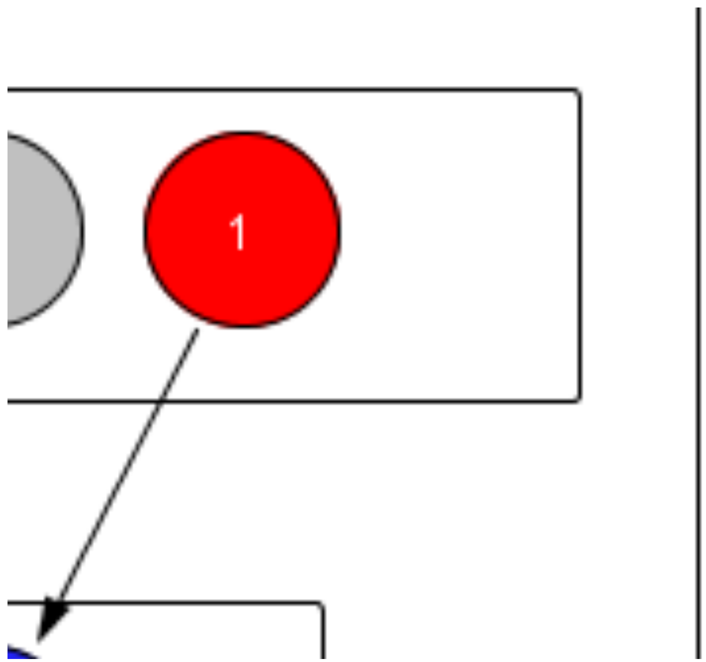
if the neuron has

1. two inputs, combined with the "weighted sum" function
$$ws(x_1, x_2) = \alpha_1 x_1 + \alpha_2 x_2$$
where $\alpha_1 = 0.1$ and $\alpha_2 = -1$
2. the step function as transfer function
$$step(\alpha) = \begin{cases} 0, & \alpha \leq 0 \\ 1, & otherwise \end{cases}$$

Neural Networks

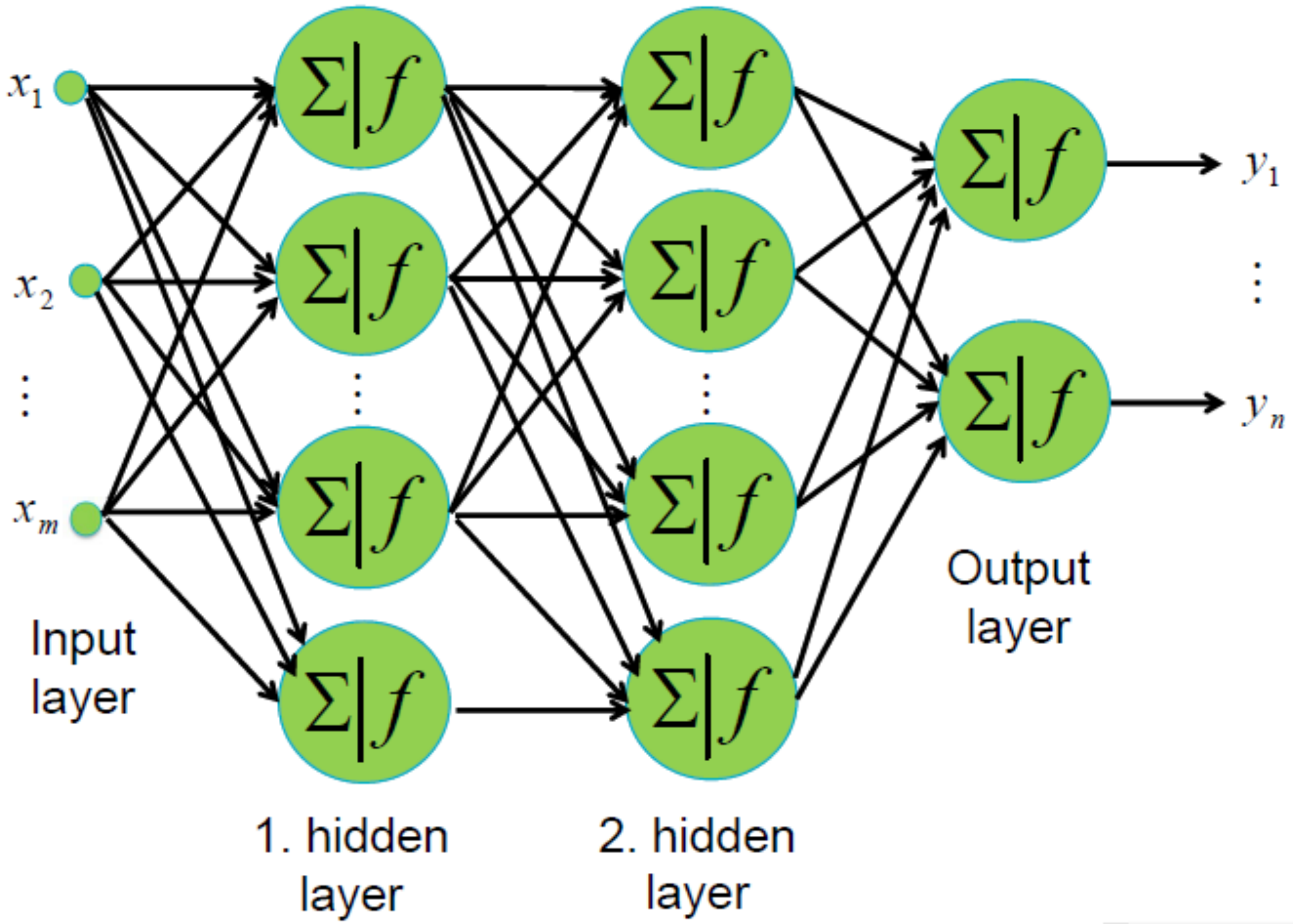
In addition to other artificial neurons, there can be *bias neurons*:

- Their output is always 1
- They don't receive input



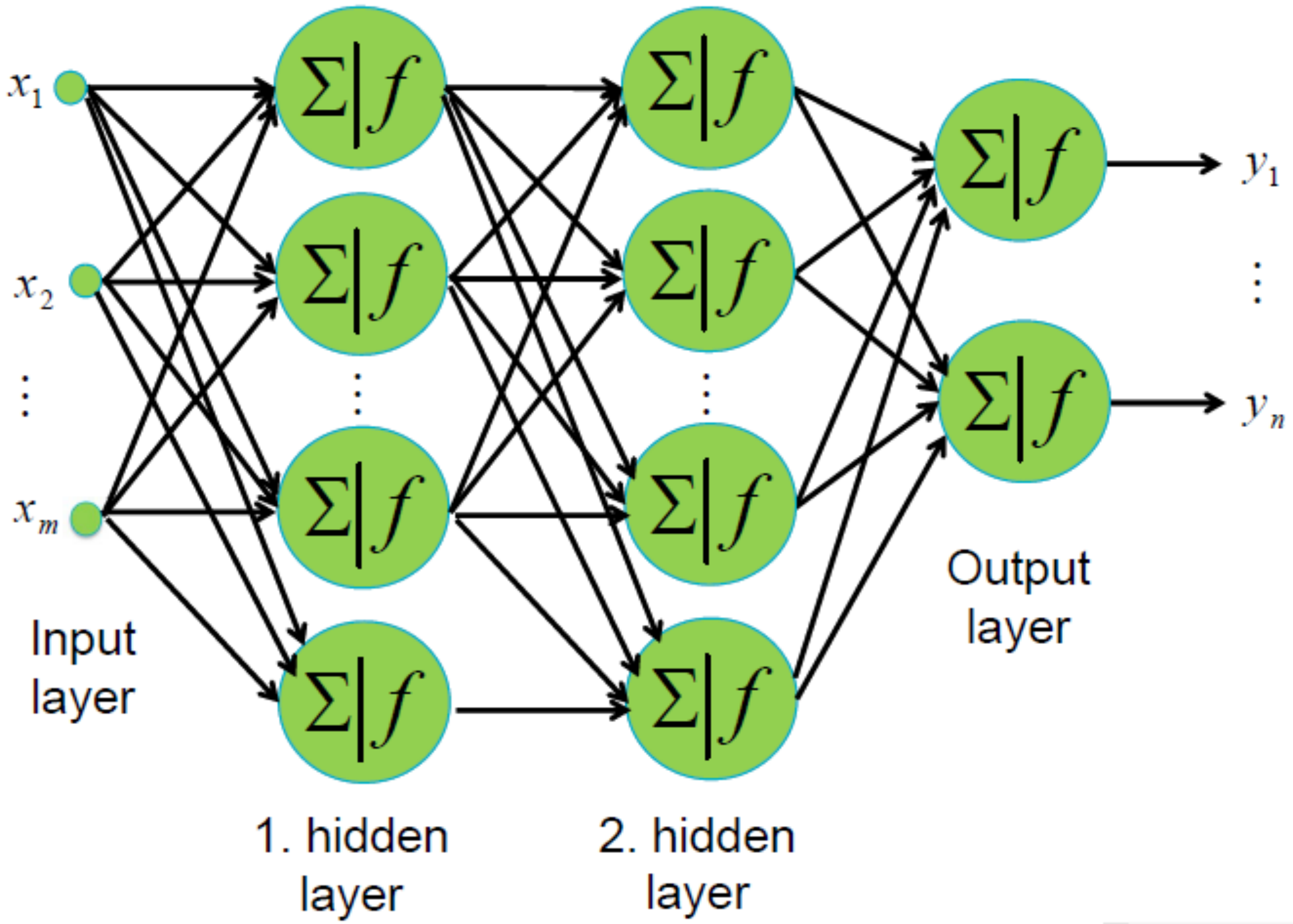
Neural Networks

- Neurons can be connected in many different ways
- Arrows "only pointing forward": *Feedforward network*

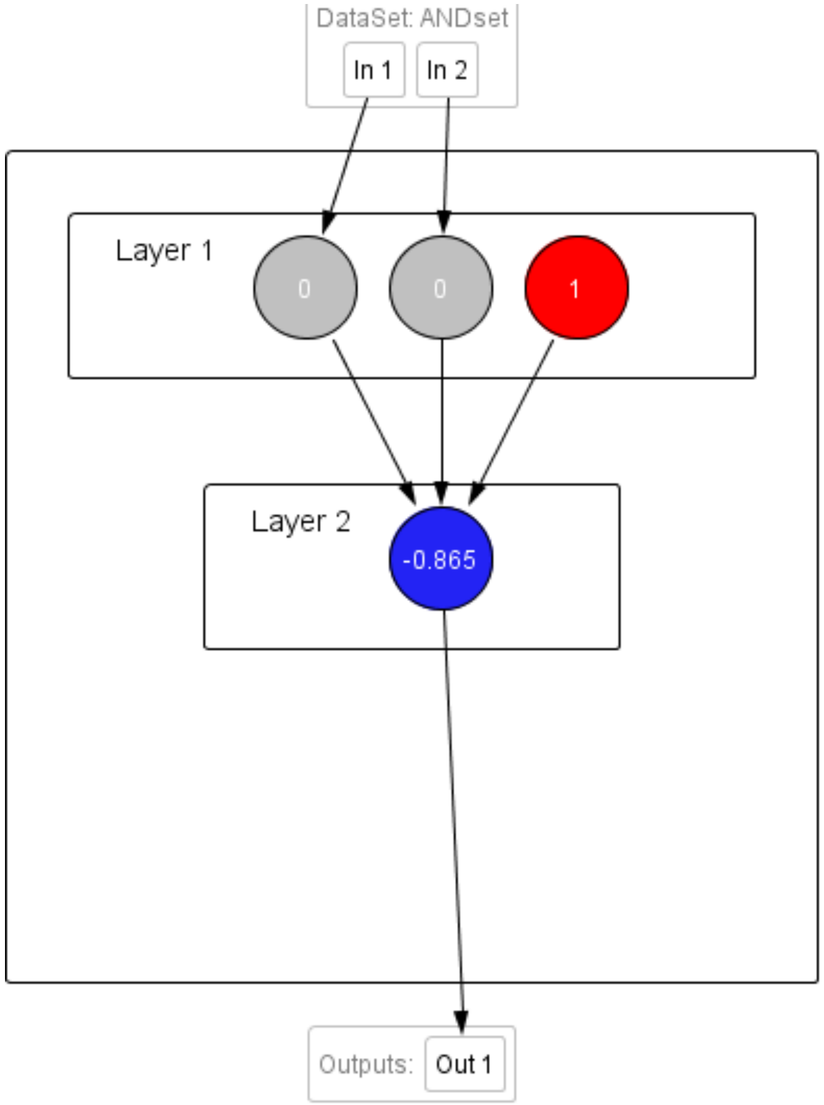


Neural Networks

- If each layer in a feedforward network is fully connected to the subsequent layer, the NN is called *Multilayer Perceptron*



Neural Networks

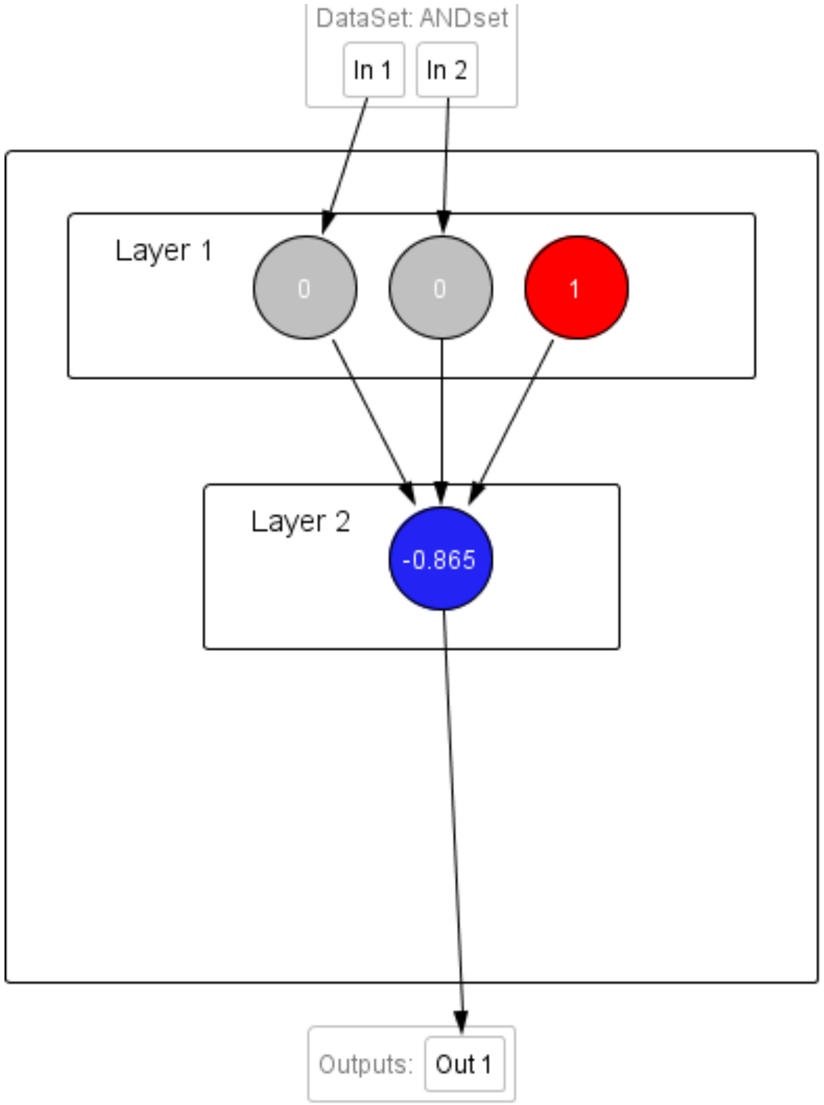


Can such a network act as a classifier function?

Example:

- two input neurons
- one bias neuron
- input function: weighted sum (weights: $\alpha_1, \alpha_2, \alpha_3$)
- transfer function: step

Neural Networks



Can such a network act as a classifier function?

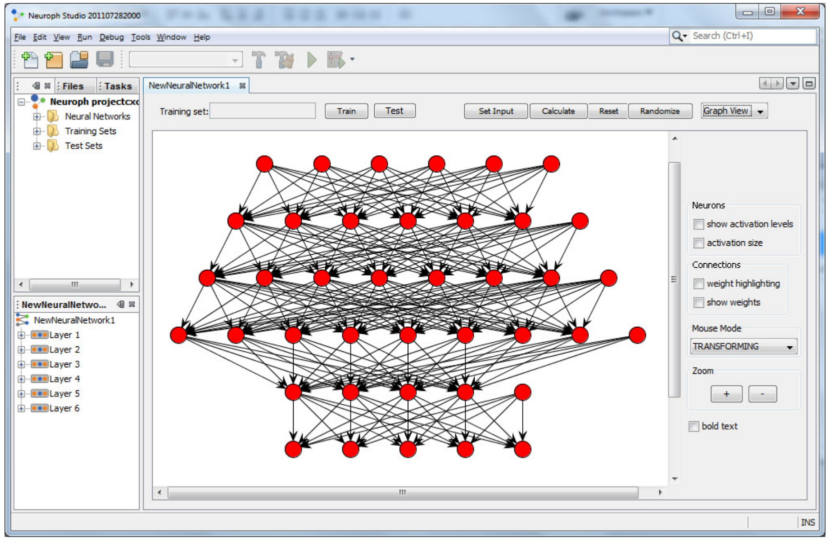
Result:

Already such a simple MLP-NN can be any linear classifier.

It just depends on the choice of the weight parameters.

This choice is typically done via an algorithm called *backpropagation*. This algorithm minimizes the cost function (typically squared error loss) on the training set.

Neural Networks



Software for NN studies:

- Neuroph studio (http://neuroph.sourceforge.net/)
  Freeware for various sorts of neural networks
- WEKA (http://www.cs.waikato.ac.nz/ml/weka/)
  freeware classification tools (among others also MLP-NNs), with GUI support
- MATLAB neural net toolbox (http://www.mathworks.de/products/neural-network/index.html)
  commercial tool, requires MATLAB installation

HOCHSCHULE RHEIN-WAAL
Rhine-Waal University of Applied Sciences