

# **Neural networks**

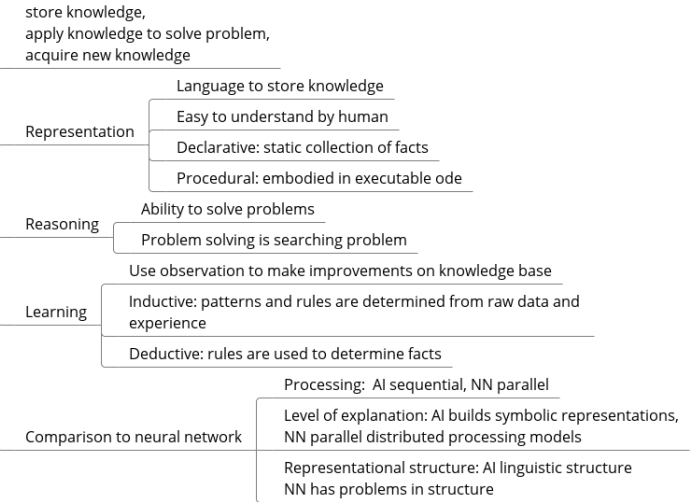
## **- Assignment 1 -**

Jens Weidmann, Markus Wiktorin

October 9, 2017

# Neural networks

## Artificial intelligence



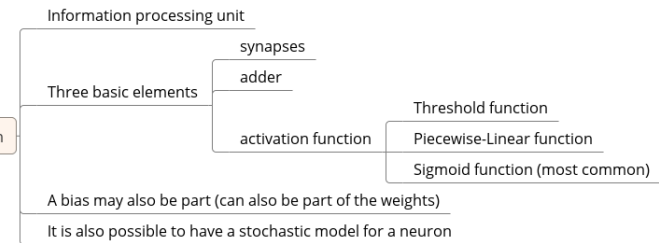
## Benefits

- Nonlinearity
- Input-Output mapping
- Adaptivity
- Evidential Response
- Contextual Information
- Fault Tolerance
- VLSI Implementability
- Uniformity of Analysis and Design
- Neurobiological Analogy

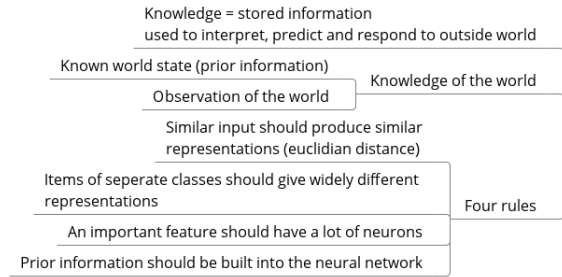
## Human brain as neural network

- Stimulus -> receptors <=> neural net (brain) <=> effectors -> response
- Neural events in millisecond range (relatively slow)
- Efficient because highly parallel

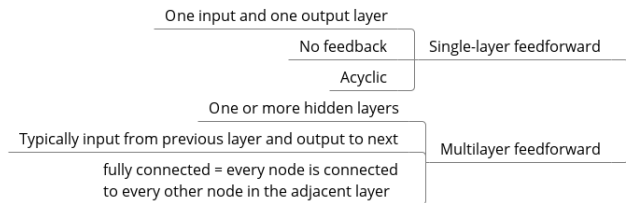
## A Neuron



## Knowledge representation



## Architectures



## Feedback

- Output of an element influences the input of an element
- Networks with feedback are called recurrent networks
- $A/(1-AB)$  is closed-loop operator
- $AB$  is open-loop operator
- unit delay operator  $z^{-1}$  (delays output by one time unit)
  - If  $|w| < 1$  then system is stable (convergent)
  - if  $|w| > 1$  then system is unstable (divergent)
  - if  $|w| = 1$  the divergence is linear, else exponential

## Directed graphs

- Network of directed links and nodes
- Node has signal
- A link originates in a node and terminates in a node and has a transfer function
- A signal flows only in the direction of the link
- A node sums up all incoming signals
- The signal of a node is transmitted to each outgoing link

In [3]:

```
import numpy as np
import sympy as sp
sp.init_printing()
```

In [4]:

```
a,v = sp.symbols("a, v")
```

## 1.2

The following cell shows the sigmoid function:

In [5]:

```
f1 = (1 - sp.exp(-a*v)) / (1 + sp.exp(-a*v))
f1
```

Out[5]:

$$\frac{1 - e^{-av}}{1 + e^{-av}}$$

The derivative is given in the next cell:

In [6]:

```
sp.diff(f1, v)
```

Out[6]:

$$\frac{a(1 - e^{-av})e^{-av}}{(1 + e^{-av})^2} + \frac{ae^{-av}}{1 + e^{-av}}$$

When we divide  $[1 - \varphi^2(v)]$  from the derivative we get the following result:

In [7]:

```
sp.simplify(sp.diff(f1, v) / (1 - f1**2))
```

Out[7]:

$$\frac{a}{2}$$

Since  $\frac{a}{2}$  is the result of dividing  $[1 - \varphi^2(v)]$  from the derivative, the derivative can be written as  $\frac{a}{2}[1 - \varphi^2(v)]$ .

The value of the derivative at the origin is computed in the next cell:

In [8]:

```
sp.diff(f1, v).subs(v, 0)
```

Out[8]:

$$\frac{a}{2}$$

The value of the derivative at the origin is  $\frac{a}{2}$ .

If  $a$  is infinitely large then  $e^{-av}$  will converge to 0 so the whole function  $\varphi(v)$  will be a straight line parallel to the x axis at the y value 1.

## 1.3

The following cell shows the sigmoid function:

In [12]:

```
f3 = v / (sp.sqrt(1 + v**2))  
f3
```

Out[12]:

$$\frac{v}{\sqrt{v^2 + 1}}$$

If we differentiate this function and compare it to the given  $\frac{\varphi^3(v)}{v^3}$  by subtracting both, we see that the result is 0:

In [34]:

```
sp.simplify(sp.diff(f3, v) - (f3**3 / v**3))
```

Out[34]:

0

That proves, that the derivative of  $\varphi(v)$  is given by  $\frac{\varphi^3(v)}{v^3}$ .

In the following cell the value of the derivative at  $v = 0$  is computed:

In [11]:

```
sp.diff(f3, v).subs(v, 0)
```

Out[11]:

1

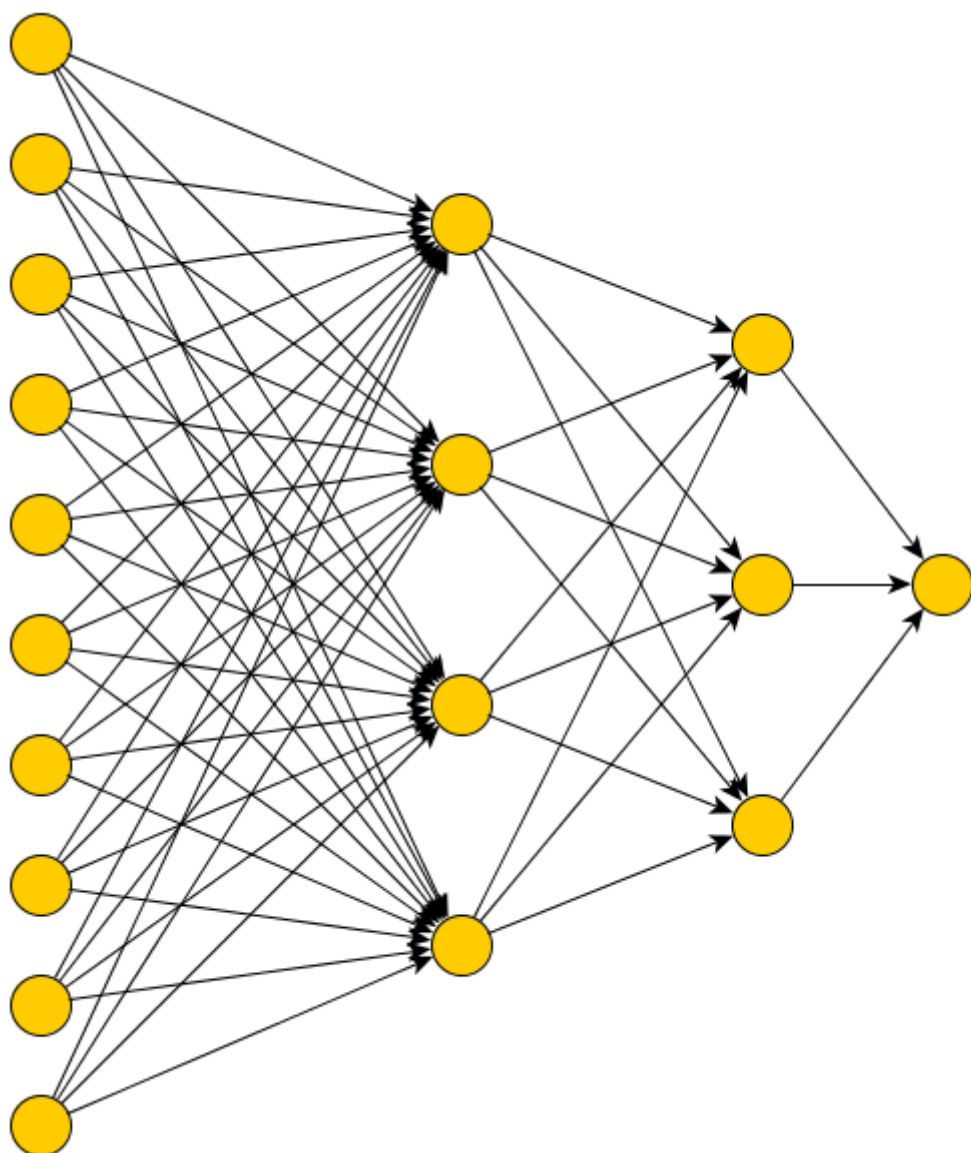
The result is 1, but we should keep in mind that the derivate is undefined for  $v = 0$  since  $v$  is in the denominator and dividing by 0 is not allowed. However, the function converges to 1 for  $v \rightarrow 0$ .

## 1.12

In [32]:

```
from IPython.display import Image  
Image("1_12_graph.png")
```

Out[32]:



## 1.13

a)

In [30]:

```
a, b = sp.symbols("a, b")  
phi = sp.Function("varphi")
```

In [37]:

```
a1 = a * 5 + b
b1 = -3 * b + 2 * a
a2 = phi(a1)
b2 = phi(b1)
a3 = 3 * a2 - b2
b3 = 6 * b2 + 4 * a2
a4 = phi(a3)
b4 = phi(b3)

phi(-2 * a4 + b4)
```

Out[37]:

$$\varphi(-2\varphi(-\varphi(2a - 3b) + 3\varphi(5a + b)) + \varphi(6\varphi(2a - 3b) + 4\varphi(5a + b)))$$

For two given inputs  $a$  and  $b$  the input-output mapping is given in the previous cell.

## b)

If the last neuron is linear instead of the function  $\varphi$  then the input-output mapping is calculated in the next cell, where  $w$  is the weight of the last neuron.

In [43]:

```
w = sp.Symbol("w")
w * (-2 * a4 + b4)
```

Out[43]:

$$w(-2\varphi(-\varphi(2a - 3b) + 3\varphi(5a + b)) + \varphi(6\varphi(2a - 3b) + 4\varphi(5a + b)))$$