Assigment1

October 14, 2017

1 Assigment 1

```
In [24]: import numpy as np
         import sympy as sp
         sp.init_printing()
```

Import of libraries and initializing for printing

```
In [25]: a, v = sp.symbols("a, v")
```

Creating symbols for functions

1.1 1.1

```
In [8]: f1 = 1/(1+sp.exp(-a*v))
        f1
```

Out[8]:

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

Creating the function

```
In [13]: f2 = sp.diff(f1, v)
         f2
```

Out[13]:

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

Calculating the derivation

```
In [15]: f3 = f2 - a * f1 * (1 -f1)
         f3 = sp.simplify(f3)
         f3
```

Out[15]:

$$0$$

To verify both derivations (given from exercise and the caculated one) are equal, the difference is caculated. The result is zero. That means both derivations are equal.

```
In [16]: f2.subs(v,0)
```

Out[16]:

$$\frac{a}{4}$$

The Value of the derivation at the origin is $\frac{a}{4}$

1.2 1.2

```
In [20]: g1 = (1 - sp.exp(-a*v)) / (1 + sp.exp(-a*v))
         g1 = sp.simplify(g1)
         g1
```

Out[20]:

$$\tanh\left(\frac{av}{2}\right)$$

Creating the function

```
In [23]: g2 = sp.diff(g1, v)
         g3 = sp.simplify(g2 - (a/2)*(1 - g1**2))
         g3
```

Out[23]:

$$0$$

Calculating the derivation and verifying the calculated derivation and the given derivation are equal

```
In [27]: g2.subs(v, 0)
```

Out[27]:

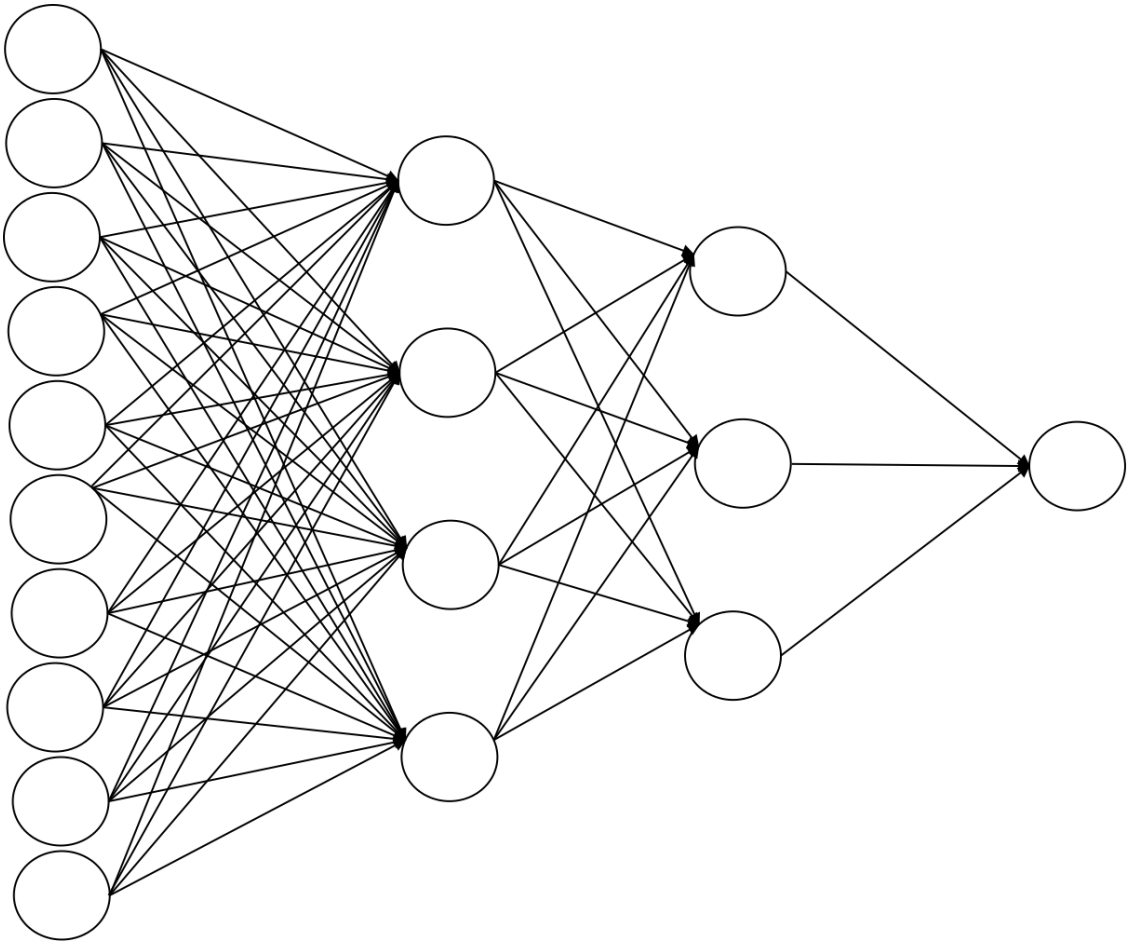
$$\frac{a}{2}$$

$\frac{a}{2}$ is the value of the derivation at the origin
Making the slope parameter a infinite large results in $e^{-av} = 0$. It follows $\phi = \frac{1}{1}$

1.3 1.12

```
In [28]: from IPython.display import Image
         Image("112.png")
```

Out[28]:



1.4 1.13

a

```
In [32]: b = sp.symbols("b")
         phi = sp.Function("phi")
```

Creating a placeholder-fuction for the input-output mapping
All nodes at the top line are named a1,...,an whereas the very first node is just a
All nodes at the bottom line are named b1,...,bn whereas the very first node is just b

```
In [35]: a1 = a * 5 + b
         b1 = b * (-3) + a * 2
```

```
a2 = phi(a1)
b2 = phi(b1)
a3 = a2 * 3 + b2 * (-1)
b3 = b2 * 6 + a2 * 4
a4 = phi(a3)
b4 = phi(b3)
c = a4 * (-2) + b4
out = phi(c)
out
```

Out[35]:

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

b

```
In [37]: w = sp.symbols("w")
```

Using a weight w instead of the function ϕ so the incoming values are linear combined.

```
In [39]: out2 = w * c
         out2
```

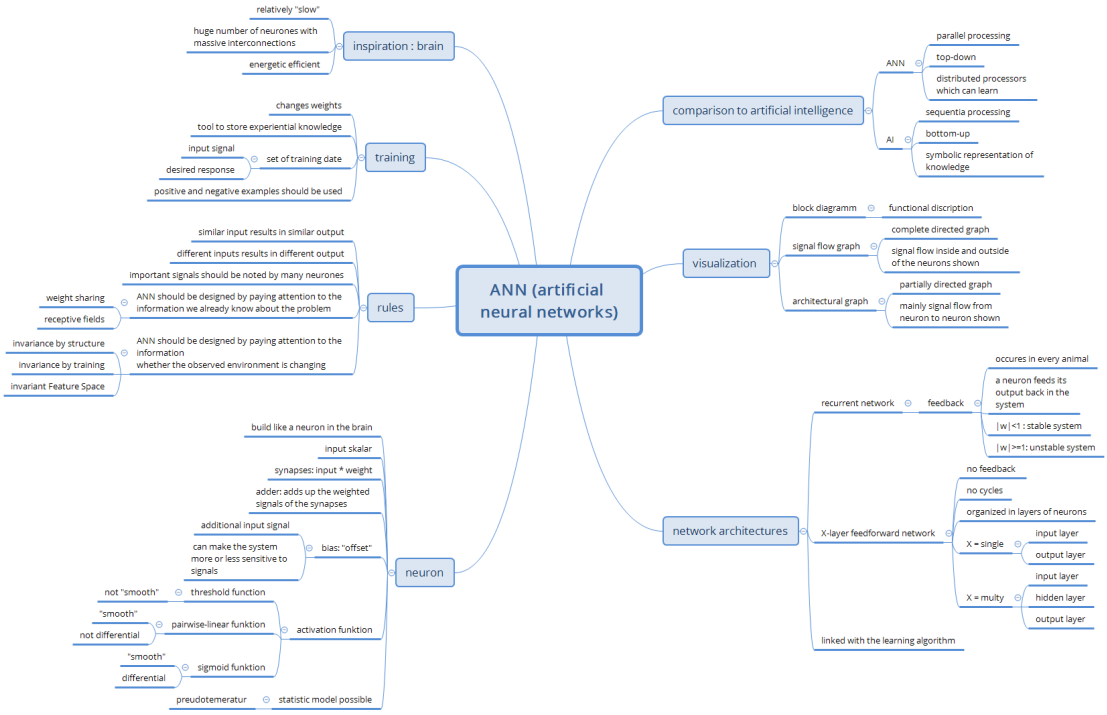
Out[39]:

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

1.5 Mindmap

```
In [44]: Image("mindmap.png")
```

Out[44]:


ANN (artificial neural networks)
inspiration : brain
relatively "slow"
huge number of neurones with massive interconnections
energetic efficient
training
changes weights
tool to store experiential knowledge
set of training date
input signal
desired response
positive and negative examples should be used
rules
similar input results in similar output
different inputs results in different output
important signals should be noted by many neurones
ANN should be designed by paying attention to the information we already know about the problem
weight sharing
receptive fields
ANN should be designed by paying attention to the information whether the observed environment is changing
invariance by structure
invariance by training
invariant Feature Space
neuron
build like a neuron in the brain
input skalar
synapses: input * weight
adder: adds up the weighted signals of the synapses
bias: "offset"
additional input signal
can make the system more or less sensitive to signals
activation funktion
threshold function
not "smooth"
pairwise-linear funktion
"smooth"
not differential
sigmoid funktion
"smooth"
differential
statistic model possible
preudotemeratur
comparison to artificial intelligence
ANN
parallel processing
top-down
distributed processors which can learn
AI
sequentia processing
bottom-up
symbolic representation of knowledge
visualization
block diagramm
functional discription
signal flow graph
complete directed graph
signal flow inside and outside of the neurons shown
architectural graph
partially directed graph
mainly signal flow from neuron to neuron shown
network architectures
recurrent network
feedback
occures in every animal
a neuron feeds its output back in the system
|w|<1 : stable system
|w|>=1: unstable system
X-layer feedforward network
no feedback
no cycles
organized in layers of neurons
X = single
input layer
output layer
X = multy
input layer
hidden layer
output layer
linked with the learning algorithm