Team Members :

Hammam Abdelwahab

Lemrabet Najlae

Jamali Hatim

In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
import sympy as sp
sp.init_printing(use_latex=True)
```

Models of a neuron

1.1

Given the logistic function, use sympy for proof as follows: $f(v) = \frac{1}{1+exp(-av)}$

$$\frac{\partial f}{\partial v} = af(v)[1 - f(v)]$$

In [15]:

```
def f(a,v):
    """
    The sigmoid function
    """
    return 1 / (1 + sy.exp(-a*v))

v = sp.Symbol('v')
a = sp.Symbol('a')

derivative = sy.diff(f(a,v),v)
display("The derivation is:")
display(derivative)
```

'The derivation is:'

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

And from the derivation we substitute $\frac{1}{(1+e^{-av})} = f(v)$.

Also we substitute $\frac{1}{(1+e^{-av})}$ = $1 - \frac{1}{1+e^{-av}}$

Therefore,

$$\frac{\partial f}{\partial v} = af(v)(1 - \frac{1}{1+e^{-av}}) = af(v)[1 - f(v)]$$

In [16]:

```
display(derivative.subs(v,0))
```

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

1.3

$g(v) = \frac{v}{\sqrt{1+v^2}}$

$\frac{\partial f}{\partial v} = f^3(v)/v^3$

Same here we use sympy to find the derivation as follows

In [22]:

```
v = sp.symbols('v')
a = sp.symbols('a')
f = v/sp.sqrt(1+v**2) #defines f as a symbolic function
df=sp.diff(f, v)
display(df)
```

$$-\frac{v^2}{\left(v^2+1\right)^{\frac{3}{2}}} + \frac{1}{\sqrt{v^2+1}}$$

Now adding this result together to become:

$\frac{v^2}{(v^2+1)^{\frac{3}{2}}}$ + $\frac{1}{\sqrt{v^2+1}}$ = $\frac{-v^2+v^2+1}{(\sqrt{v^2+1})^2}$ = $\frac{\partial f}{\partial v} = \frac{v^3}{(\sqrt{v^2}+1)^3} * \frac{1}{v^3} = \frac{\varphi^3}{v^3}$

Now we subtitute for at the origin as follows:

In [23]:

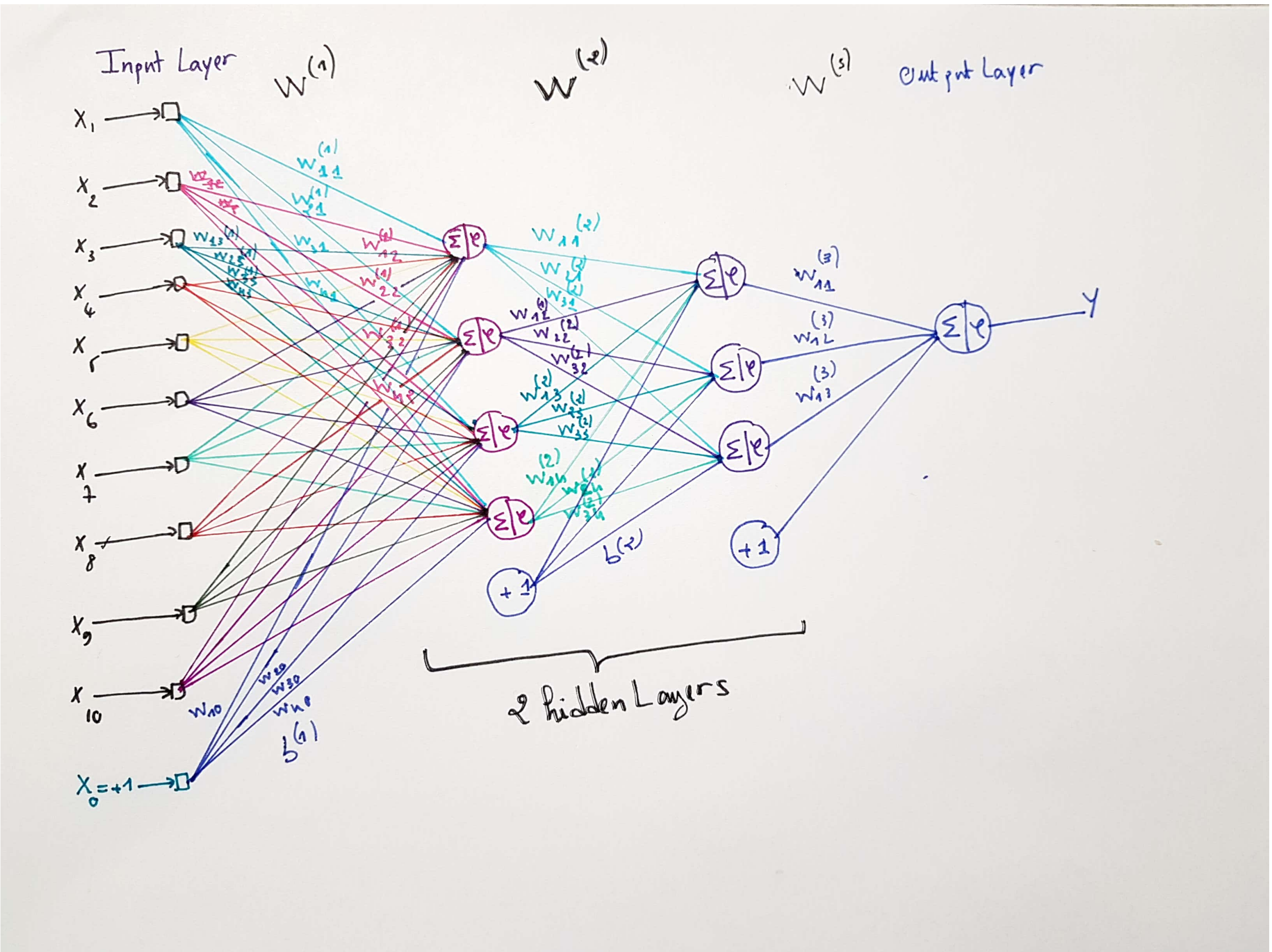
```
display(df.subs(v,0))
```

$$1$$

Network architectures

1.12

The architectural graph for the neural network is as follows:



1.13. a

The input-output mapping for the network is depicted as follows:

$$Y = \varphi(-2\varphi(3\varphi(5X_1 + X_2) - \varphi(2X_1 - 3X_2)) + \varphi(4\varphi(5X_1 + 1X_2) + 6\varphi(2X_1 - 3X_2)))$$

1.13. b

supposing that the output neuron operates in the linear region. Then:

$\varphi(x) \approx \alpha x$

and the output vector $y_k \approx \alpha w_k^T x$

$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \alpha \begin{pmatrix} w_1^T \\ \vdots \\ w_m^T \end{pmatrix} x = \alpha W x$$

and the output becomes:

$$Y = \alpha(-2\alpha(3\alpha(5X_1 + X_2) - \alpha(2X_1 - 3X_2)) + \alpha(4\alpha(5X_1 + 1X_2) + 6\alpha(2X_1 - 3X_2)))$$

knowledge Representation

1.21

According to (simard 1992) the transformation s(P,a) can be depicted by the Taylor expansion of the transformation s as follows:

$$s(x, \alpha) = s(x, 0) + \partial \frac{s(x, \alpha)}{\partial \alpha} + O(\alpha^2) \approx x + \alpha T$$

This equation shows that the linear approximation is characterized completely by the point p (x) and the tangent vector. If we consider alpha as a rotation angle, ||a|| < 1 shows good linear approximation, and if a = 0 then transformation will result to the same input (image).According to (simard 1992) the reason that for infinitesimal (or an isomorphism) transformation, there exists a direct correspondence between the tangent vectors of the tangent plane and the compositions of transformations.For example, if we take three tangent vectors for rotation, we will generate a tangent plane corresponding to all possible compositions of rotations. Hencefore, the resulting tangent distance is considered locally invariant to all rotations (of any center). This also applies to X-translations and Y-translations.

reference: Transformation Invariance for Pattern Recognition-Tangent Distance and Tangent Probagation.

http://yann.lecun.com/exdb/publis/pdf/simard-00.pdf (http://yann.lecun.com/exdb/publis/pdf/simard-00.pdf)