In [1]:
```
# Implementation of Back Propagation algorithm...
```

In [2]:
```
import numpy as np
```

In [3]:
```
inputsize=2
hiddensize=3
outputsize=1
lr=0.1
w1=np.random.randn(inputsize,hiddensize)*0.01
w2=np.random.randn(hiddensize,outputsize)*0.01
b1=np.random.randn(hiddensize,1)*0.01
b2=np.random.randn(outputsize,1)*0.01
```

In [4]:
```
w1
```

Out[4]:
```
array([[ 0.02067298,  0.00739844, -0.00310455],
       [ 0.00173518, -0.00022919, -0.01506263]])
```

In [5]:
```
print("w1=")
print(w1)
print("w2=")
print(w2)
print("b1=")
print(b1)
print("b2=")
print(b2)
```

```
w1=
[[ 0.02067298  0.00739844 -0.00310455]
 [ 0.00173518 -0.00022919 -0.01506263]]
w2=
[[-0.01105828]
 [-0.01992644]
 [ 0.01008439]]
b1=
[[-0.0129636 ]
 [ 0.01222187]
 [-0.01205409]]
b2=
[[-0.00537605]]
```

In [6]:
```
#   DEFINE ACTIVATION FUNCTION

def sigm(v):
    y=1/(1+np.exp(-v))
    return y

def derivative(y):
    return(y*(1-y))

def msqerror(yp,y):
    E= ((yp-y)**2)
    return E
```

In [7]:
```
#   DEFINE A DATASET

x=np.array([[7,2],[2,6],[9,1]])
y=np.array([[80],[50],[95]])

#  NORMALIZE
```

```
x=x/np.amax(x,axis=0)
y=y/100
```

In [8]:
```
#  CREATE A NEURAL NETWORK AND TRAIN IT

def train(x,y):
    global w1,w2,b1,b2,lr
    #FORWARD PHASE
    vh=np.dot(x,w1)+b1
    yh= sigm(vh)
    vo=np.dot(yh,w2)+b2
    yo=sigm(vo)
    #   BACKPROPAGATION PHASE
    e1 = yo -y
    delta_o = e1 * yo *(1-yo)
    e2 = delta_o @w2.T
    delta_h = e2 * yh * (1-yh)
    go = yh.T @delta_o
    gh = x.T @delta_h
    w1 = w1- (lr*gh)
    w2 = w2 - (lr*go)
    b1 = b1-(lr* delta_h)
    b2 = b2 - (lr * delta_o)
    return(yo)
```

In [11]:
```
train(x,y)
```

Out[11]:
```
array([[0.50148365],
       [0.49961438],
       [0.5024223 ]])
```

In [12]:
```
# iteration algorithm
for i in range(5500):
    train(x,y)
```

In [13]:
```
def forwardtest(x,y):
    global w1,w2, b1,b2
    vh=np.dot(x,w1)+b1
    yh=sigm(vh)
    vo=np.dot(yh,w2)+b2
    yo=sigm(vo)
    return(yo)
```

In [14]:
```
forwardtest(x,y)
```

Out[14]:
```
array([[0.80049095],
       [0.50022354],
       [0.9462993 ]])
```

In [ ]: