```
import numpy as np
X=np.array(([2,9],[1,5],[3,6]),dtype=float)
y=np.array(([92],[86],[89]),dtype=float)
X=X/np.amax(X,axis=0)
y=y/100
def sigmoid(x):
  return 1/(1+np.exp(-x))
def derivatives_sigmoid(x):
  return 1*(1-x)
epoch=5000
lr=0.1
inputlayer_neurons=2
hiddenlayer_neurons=3
outputlayer_neurons=1
wh=np.random.uniform(size=(inputlayer_neurons,hiddenlayer_neurons))
bh=np.random.uniform(size=(1,hiddenlayer_neurons))
wout=np.random.uniform(size=(hiddenlayer_neurons,outputlayer_neurons))
bout=np.random.uniform(size=(1,outputlayer_neurons))



for i in range(epoch):
  hinp1=np.dot(X,wh)
  hinp=hinp1+bh
  hlayer_act=sigmoid(hinp)
  outinp1=np.dot(hlayer_act,wout)
  outinp=outinp1+bout
  output=sigmoid(outinp)
  EO=y-output
  outgrad=derivatives_sigmoid(output)
  d_output=EO*outgrad
  EH=d_output.dot(wout.T)
  hiddengrad=derivatives_sigmoid(hlayer_act)
  d_hiddenlayer=EH*hiddengrad
  wout+=hlayer_act.T.dot(d_output)*lr
  wh+=X.T.dot(d_hiddenlayer)*lr
print("Input:\n",str(X))
print("Actual Output:\n",str(y))
print("Predicted Output:\n",output)
```

```
Input:
 [[0.66666667 1.        ]
 [0.33333333 0.55555556]
 [1.         0.66666667]]
Actual Output:
 [[0.92]
 [0.86]
 [0.89]]
Predicted Output:
```

```
[[0.89596322]
 [0.8738003 ]
 [0.89930584]]
```

```
19881A1267
C.PREETHI
```

✓ 0s completed at 20:56