## LAB 2

```
import numpy as np
import matplotlib.pyplot as plt

# Training Data set and define hyper parameters
#  Assume 4 neurons in input and hidden layer
X=np.array([[0,0],[0,1],[1,0], [1,1]])
y=np.array([[0],[1],[1],[0]])
w1=np.random.randn(2,4)
w2=np.random.randn(4,1)
learning_rate= 0.1   #0.05
costs=[]
num_epoch=10000

def sigmoid_func(x):
 y=1+np.exp(-x)
 y=1/y
 return y

# Feedforward Network

def fwd_pass(X_training, wt1, wt2):
 b=np.ones((4,4))
 a1=np.matmul(X_training,wt1)+b
 z1=sigmoid_func(a1)
# print(z1)
 a2=np.matmul(z1,wt2)
 z2=sigmoid_func(a2)
 #print(z2)
 return a1,z1,a2,z2

#Backpropagation Learning

def back_propagation(a2, z0, z1, z2, y):
 diff2 =(z2-y)*z2*(1-z2)
 Derivative2=np.matmul(z1.T,diff2)
 #print(Derivative2)
 diff1=(diff2.dot( w2.T ))*sigmoid_func(a1)*(1-sigmoid_func(a1))
 Derivative1=np.matmul(z0.T,diff1)
 return diff2,Derivative1, Derivative2

 #Weight Updation

def updateWeights(Derivative1,Derivative2,learning_rate,m,w1,w2):
 dw1= learning_rate*(1/m)*Derivative1
 w1=w1-dw1
 dw2= learning_rate*(1/m)*Derivative2
 w2= w2-dw2
 return w1,w2

# Prediction
def predict(X_test,weight1,weight2):
 a1,z1,a2,z2=fwd_pass(X_test,weight1,weight2)
 return z2

 # Testing
def test(X_test,y_test):
 y_predicted=predict(X_test,w1,w2)
 print('Test set is :')
 print(X_test[:,:])
 print('\nPredicted values for Test set are')
 print(np.round(y_predicted))
 print('\n And actual y values for test set are')
 print(y_test)
```

```
X=np.array([[0,0],[0,1],[1,0], [1,1]])
len_z1=len(X)
b=np.ones((len_z1,1))
z1=np.concatenate((X,b),axis=1)
print(z1)
```

```
     [[0. 0. 1.]
      [0. 1. 1.]
      [1. 0. 1.]
      [1. 1. 1.]]
```

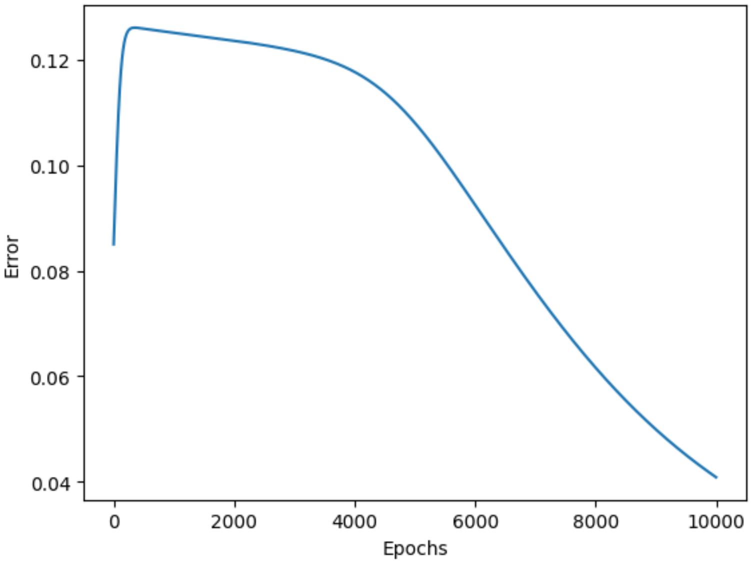
```
m=len(X)
for i in range(num_epoch):
  a1,z1,a2,z2=fwd_pass(X,w1,w2)
  diff2, Derivative1,Derivative2=back_propagation(a2,X,z1,z2,y)
  w1,w2=updateWeights(Derivative1,Derivative2,learning_rate,m,w1,w2)
  cost_i=np.mean(np.abs(diff2))
  costs.append(cost_i)
if i == 0 or i==num_epoch-1:
 print('In Iterartion: '+ str(i+1))
 print('the error is :'+str(cost_i)+'\n')
```

```
In Iterartion: 10000
the error is :0.040869144245690325
```

```
print('After the completion of Training :\n')
z3=predict(X,w1,w2)
print('Y value predicted: ')
print(np.round(z3))
print('\n')
plt.plot(costs)
plt.ylabel("Error")
plt.xlabel("Epochs")
plt.show()
```

```
After the completion of Training :

Y value predicted:
[[0.]
 [1.]
 [1.]
 [0.]]
```



```
X_test = np.array([[1,0],[1,1],[0,1 ],[0,0]])
y_test = np.array([[1.],[0.],[1],[0.]])
test(X_test,y_test)
```

```
Test set is :
[[1 0]
 [1 1]
 [0 1]
 [0 0]]

Predicted values for Test set are
[[1.]
 [0.]
 [1.]
 [0.]]

 And actual y values for test set are
[[1.]
 [0.]
 [1.]
 [0.]]
```