INTRODUCTION TO NEURAL NETWORKS (CS 537-01)
ASSIGNMENT 3

SHIVANGI GUPTA
A25266618
sg0097@uah.edu

DataSet

X1	X2	X3	Y
-1	-1	-1	0
-1	-1	1	1
-1	1	-1	1
-1	1	1	0
1	-1	-1	1
1	-1	1	0
1	1	-1	0
1	1	1	1

There is 1 input layer with 3 units, 2 hidden layer and a Output Layer with 1 unit. The first hidden layer uses ReLu(Rectified linear Unit) as the activation function and the second layer uses Sigmoid Function as the activation function.

SOURCE CODE

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

n_x=3
n_y=1

def sigmoid(x):
    return 1/(1+np.exp(-x))

def ReLU(x):
    return x * (x > 0)

def dReLU(x):
    return 1 * (x > 0)
```

```
def Backpropogation(alpha,X,Y,num_of_iterations,n):
    W1=np.random.randn(n,3)-0.5
    b1 = np.zeros(shape=(n, 1))
    W2=np.random.randn(1,n)-0.5
    b2 = np.zeros(shape=(n_y, 1))
    cost=[]

    for i in range(0,num_of_iterations):
        Z1 = np.dot(W1, X) + b1
        A1 = ReLU(Z1)
        Z2 = np.dot(W2, A1) + b2
        A2 = sigmoid(Z2)

        log = np.multiply(Y,np.log(A2)) + np.multiply((1 - Y), np.log(1 - A2))
        cost.append(- np.sum(log) / 8)

         if i%100==0:
            print("Cost at Iteration",i," = ",cost[i])

        dZ2= (A2 - Y)/8
        dW2 = (np.dot(dZ2, A1.T))/8
        db2 = (dZ2)/8
        dZ1 = (np.multiply(np.dot(W2.T, dZ2), dReLU(A1)))/8
        dW1 = (np.dot(dZ1, X.T))/8
        db1 = dZ1/8

        W1 = W1 - alpha * dW1
        b1 = b1 - alpha  * db1
        W2 = W2 - alpha  * dW2
        b2 = b2 - alpha  * db2

    return cost

x1=[-1,-1,-1,-1,1,1,1,1]
x2=[-1,-1,1,1,-1,-1,1,1]
x3=[-1,1,-1,1,-1,1,-1,1]
X=np.array([x1,x2,x3])
Y=np.array([0,1,1,0,1,0,0,1])

C=Backpropogation(alpha=0.2,X=X,Y=Y,num_of_iterations=5000,n=100)
iterations=[i for i in range(len(C))]
plt.plot(iterations,C)
plt.xlim(0,1000)
```

```
plt.xlabel("Number of Iterations")
plt.ylabel("Cost")
plt.show()
```

Output:

```
Cost at Iteration 0  =  19.0004295776
Cost at Iteration 100  =  0.497852682066
Cost at Iteration 200  =  0.189751780442
Cost at Iteration 300  =  0.13859276517
Cost at Iteration 400  =  0.108691686092
Cost at Iteration 500  =  0.0888214405023
Cost at Iteration 600  =  0.0747392004078
Cost at Iteration 700  =  0.064294442155
Cost at Iteration 800  =  0.0562683059645
Cost at Iteration 900  =  0.0499266498969
Cost at Iteration 1000  =  0.0448002080581
Cost at Iteration 1100  =  0.0405773071814
Cost at Iteration 1200  =  0.0370432100829
Cost at Iteration 1300  =  0.0340458444043
Cost at Iteration 1400  =  0.0314732003868
Cost at Iteration 1500  =  0.0292432583243
Cost at Iteration 1600  =  0.0272929405643
Cost at Iteration 1700  =  0.0255737699055
Cost at Iteration 1800  =  0.0240479598918
Cost at Iteration 1900  =  0.0226849594865
Cost at Iteration 2000  =  0.0214606777741
Cost at Iteration 2100  =  0.020355405883
Cost at Iteration 2200  =  0.019352794443
Cost at Iteration 2300  =  0.0184395136543
Cost at Iteration 2400  =  0.017604197161
Cost at Iteration 2500  =  0.0168377502151
Cost at Iteration 2600  =  0.0161319685289
Cost at Iteration 2700  =  0.015479995685
Cost at Iteration 2800  =  0.0148761570243
Cost at Iteration 2900  =  0.0143153964211
Cost at Iteration 3000  =  0.0137932569108
```

```
Cost at Iteration 3100  =  0.0133059936522
Cost at Iteration 3200  =  0.0128503898658
Cost at Iteration 3300  =  0.0124233325082
Cost at Iteration 3400  =  0.0120223646597
Cost at Iteration 3500  =  0.0116452714186
Cost at Iteration 3600  =  0.0112898912826
Cost at Iteration 3700  =  0.0109545510703
Cost at Iteration 3800  =  0.0106375392112
Cost at Iteration 3900  =  0.0103374632166
Cost at Iteration 4000  =  0.0100530352197
Cost at Iteration 4100  =  0.00978307574628
Cost at Iteration 4200  =  0.00952652844353
Cost at Iteration 4300  =  0.00928243559917
Cost at Iteration 4400  =  0.00904993278193
Cost at Iteration 4500  =  0.0088282120568
Cost at Iteration 4600  =  0.00861656725532
Cost at Iteration 4700  =  0.00841433363291
Cost at Iteration 4800  =  0.0082209115322
Cost at Iteration 4900  =  0.00803575423028
```