**INTRODUCTION TO NEURAL NETWORKS (CS 537-01)**

**ASSIGNMENT 3**

**SHIVANGI GUPTA**

**A25266618**

[sg0097@uah.edu](mailto:sg0097@uah.edu)

|  |  |  |  |
| --- | --- | --- | --- |
| **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** |

**DataSet**

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)**

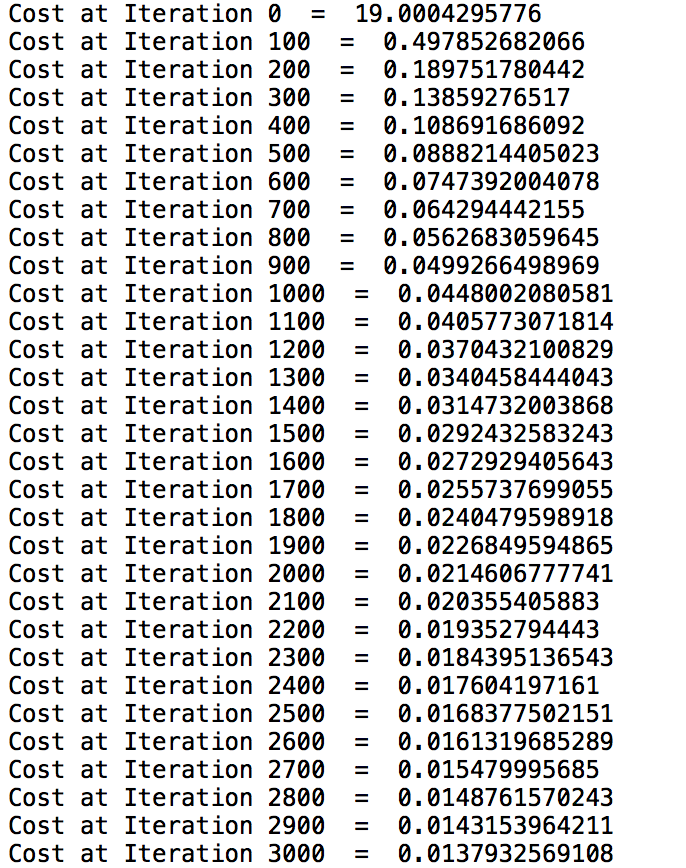
**plt.xlim(0,1000)**

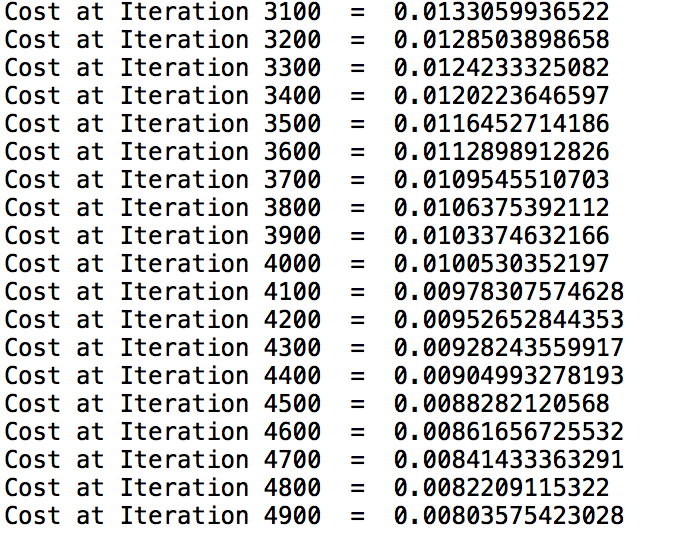
**plt.xlabel("Number of Iterations")**

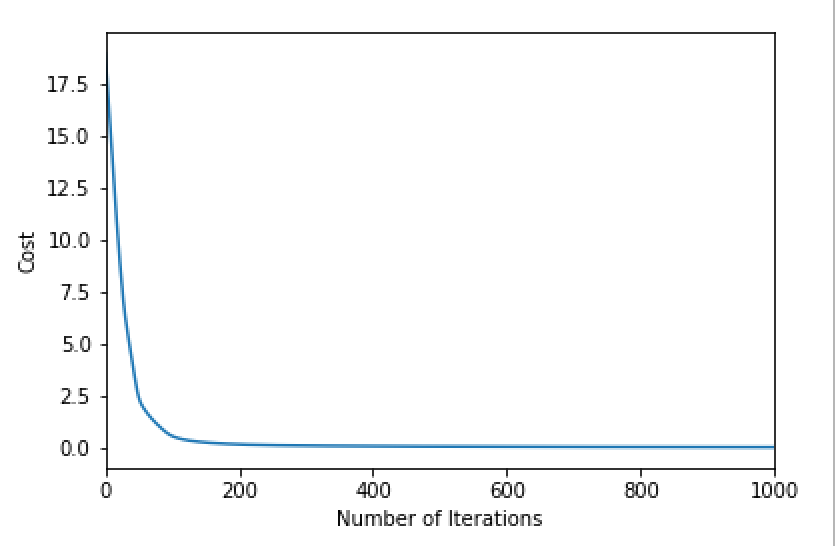
**plt.ylabel("Cost")**

**plt.show()**

**Output:**

****

****

****