**“Artificial Intelligence (LAB-5)”**

**Name:** Ahmad Amin

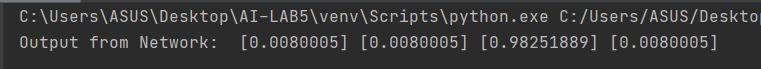
**Class:** BSCS-8C

**CMS:** 247956

**Task 1**

import numpy as np  
  
def sig(x):  
 return 1 / (1 + np.exp(-x))  
  
def sigDeri(x):  
 return x \* (1 - x)  
  
inputs = np.array([[1, 1], [1, 1], [1, 0], [1, 1]])  
output = np.array([[0], [0], [1], [0]])  
  
epochs = 10000  
lr = 0.5  
  
iln, hln, oln = 2, 2, 1  
hWeights = np.random.uniform(size=(iln, hln))  
oWeights = np.random.uniform(size=(hln, oln))  
  
hBias = np.random.uniform(size=(1, hln))  
oBias = np.random.uniform(size=(1, oln))  
  
for \_ in range(epochs):  
  
 hlActivate = np.dot(inputs, hWeights)  
 olActivate = np.dot(hlOutput, oWeights)  
  
 hlActivate += hBias  
 olActivate += oBias  
  
 hlOutput = sig(hlActivate)  
 pred\_output = sig(olActivate)  
  
 err = output - pred\_output  
  
 derPredictOutput = err \* sigDeri(pred\_output)  
  
 errHL = derPredictOutput.dot(oWeights.T)  
 derHL = errHL \* sigDeri(hlOutput)  
 oWeights += hlOutput.T.dot(derPredictOutput) \* lr  
 hWeights += inputs.T.dot(derHL) \* lr  
  
 oBias += np.sum(derPredictOutput, axis=0, keepdims=True) \* lr  
 hBias += np.sum(derHL, axis=0, keepdims=True) \* lr  
  
print("Output --> : ", \*pred\_output)

**output:**

****

**Task 2**

**Code:** (This code is written on google collab platform)

import pandas as pd

import csv

from sklearn.neural\_network import MLPClassifier

from google.colab import files

trainData = pd.read\_csv('https://raw.githubusercontent.com/Ahmad-Amin/lab5AI/main/TrainingSet.csv');

dataFeatures = trainData.iloc[:,0:4].values

dataLabels = trainData.iloc[:,4].values

clf = MLPClassifier(solver= 'lbfgs', alpha=1e-5, hidden\_layer\_sizes=100, random\_state=1)

clf.fit(dataFeatures, dataLabels)

testData = pd.read\_csv('https://raw.githubusercontent.com/Ahmad-Amin/lab5AI/main/TestSet1.csv')

testFeatures = testData.iloc[:,0:4].values

testLabels = clf.predict(testFeatures)

testData = pd.DataFrame(testLabels)

testData.to\_csv("result.csv", index=False, header=None)

files.download('result.csv')

**Output:**

****