Practical No. 8

Artificial Neural Network

Q.1] Implement a basic ANN in Python using numpy.

**Code:**

import numpy as np

class NeuralNetwork():

def \_\_init\_\_(self):

np.random.seed(1)

self.synaptic\_weights = 2 \* np.random.random((3, 1)) - 1

def sigmoid(self, x):

return 1 / (1 + np.exp(-x))

def sigmoid\_derivative(self, x):

return x \* (1 - x)

def train(self, training\_inputs, training\_outputs, training\_iterations):

for iteration in range(training\_iterations):

output = self.think(training\_inputs)

error = training\_outputs - output

adjustments = np.dot(training\_inputs.T, error \* self.sigmoid\_derivative(output))

self.synaptic\_weights += adjustments

def think(self, inputs):

inputs = inputs.astype(float)

output = self.sigmoid(np.dot(inputs, self.synaptic\_weights))

return output

if \_\_name\_\_ == "\_\_main\_\_":

neural\_network = NeuralNetwork()

print("Beginning Randomly Generated Weights: ")

print(neural\_network.synaptic\_weights)

training\_inputs = np.array([[0,0,1],[1,1,1],[1,0,1],[0,1,1]])

training\_outputs = np.array([[0,1,1,0]]).T

neural\_network.train(training\_inputs, training\_outputs, 1000000)

print('Ending Weights After Training: ')

print(neural\_network.synaptic\_weights)

user\_input\_one = str(input("User input one: "))

user\_input\_two = str(input("User input two: "))

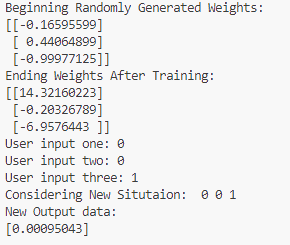
user\_input\_three = str(input("User input three: "))

print("Considering New Situtaion: ",user\_input\_one,user\_input\_two,user\_input\_three)

print("New Output data: ")

print(neural\_network.think(np.array([user\_input\_one, user\_input\_two, user\_input\_three])))

**Output:**



Q.2] Implement a ANN for predicting “pass” or “fail” using tensorflow and keras library in Python.

**Note:** Need to clarify the keras import issue

**Code:**

import numpy as np

import tensorflow as tf

#from tensorflow.keras.models import Sequential

#from tensorflow.keras.layers import Dense

from tensorflow import keras;

from keras.api.models import Sequential

from keras.api.layers import Dense

# [Hours Studied, Hours Slept]

X = np.array([[10, 7], [9, 8], [3, 7], [2, 6], [5, 6], [8, 6], [1, 5], [7, 7]])

y = np.array([1, 1, 0, 0, 1, 0, 0, 1]) # 1 = Pass, 0 = Fail

model = Sequential()

model.add(Dense(8, input\_dim=2, activation="relu"))

model.add(Dense(4, activation="relu"))

model.add(Dense(1, activation="sigmoid"))

model.compile(loss="binary\_crossentropy", optimizer="adam", metrics=["accuracy"])

model.fit(X, y, epochs=100, batch\_size=2)

# Testing

new\_data = np.array([[9, 8], [4, 4]])

predictions = model.predict(new\_data)

for i, prediction in enumerate(predictions):

print(f"Sample {i + 1}: {'Pass' if prediction > 0.5 else 'Fail'} with probability {prediction[0]:.2f}")

**Output:**

****