Name: Ayush Chanchal

Sap id: 500097569

Roll no: R2142211432

Batch: B6

Application of ML

Experiment: 07

Source 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, 15000)

print("Ending weights after training")

print(neural\_network.synaptic\_weights)

user\_input\_one = float(input("user input one:"))

user\_input\_two = float(input("user input two:"))

user\_input\_three = float(input("user input three:"))

print("Considering new situation:", 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])))

print('Yes, we did it!!!!')

Output:

