**Exp16 :Write the python program to implement Feed forward neural Network**

**Input:**

import numpy as np

def sigmoid(x):

    return 1 / (1 + np.exp(-x))

def sigmoid\_deriv(x):

    return x \* (1 - x)

X = np.array([[0,0],

              [0,1],

              [1,0],

              [1,1]])

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

np.random.seed(1)

input\_size, hidden\_size, output\_size = 2, 4, 1

W1 = np.random.randn(input\_size, hidden\_size)

b1 = np.zeros((1, hidden\_size))

W2 = np.random.randn(hidden\_size, output\_size)

b2 = np.zeros((1, output\_size))

lr = 0.1

epochs = 10000

for epoch in range(epochs):

    z1 = np.dot(X, W1) + b1

    a1 = sigmoid(z1)

    z2 = np.dot(a1, W2) + b2

    a2 = sigmoid(z2)

    error = y - a2

    d\_a2 = error \* sigmoid\_deriv(a2)

    d\_a1 = np.dot(d\_a2, W2.T) \* sigmoid\_deriv(a1)

    W2 += lr \* np.dot(a1.T, d\_a2)

    b2 += lr \* np.sum(d\_a2, axis=0, keepdims=True)

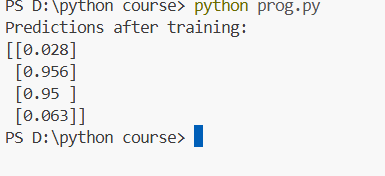
    W1 += lr \* np.dot(X.T, d\_a1)

    b1 += lr \* np.sum(d\_a1, axis=0, keepdims=True)

print("Predictions after training:")

print(a2.round(3))

**Output:**

****