# Assignment No-08

**Name-** Hule Samiksha Anil

**Roll No-** 24

**Title-** Program for creating a back propagation feed-forward neural network.

**Program:**

import numpy as np

def sigmoid(x):

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

def sigmoid\_derivative(x):

return x \* (1 - x)

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

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

learning\_rate = 0.1

num\_epochs = 100000

hidden\_weights = 2 \* np.random.random((2, 2)) - 1

output\_weights = 2 \* np.random.random((2, 1)) - 1

for \_ in range(num\_epochs):

hidden\_layer = sigmoid(np.dot(X, hidden\_weights))

output\_layer = sigmoid(np.dot(hidden\_layer, output\_weights))

output\_error = y - output\_layer

output\_delta = output\_error \* sigmoid\_derivative(output\_layer)

hidden\_error = output\_delta.dot(output\_weights.T)

hidden\_delta = hidden\_error \* sigmoid\_derivative(hidden\_layer)

output\_weights += hidden\_layer.T.dot(output\_delta) \* learning\_rate

hidden\_weights += X.T.dot(hidden\_delta) \* learning\_rate

print("Input:")

print(X)

print("Output:")

print(output\_layer)

# Output:

# Input:

# [[00]

# [0 1]

# [1 0]

# [1 1]]

# Output:

# [[0 61385986]

# [0.63944088]

# [0.8569871]

# [0.11295854]]