**Experiment-7**

**Aim:** **Build an Artificial Neural Network by implementing the Back**

**propagation algorithm and test the same using appropriate data sets.**

**Program:**

import numpy as np

def sigmoid(x):

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

def derivatives\_sigmoid(x):

return x \* (1 - x)

X\_train = np.array([[2, 9], [1, 5], [3, 6]], dtype=float)

y\_train = np.array([[92], [86], [89]], dtype=float)

X\_test = np.array([[4, 8], [5, 3]], dtype=float)

y\_test = np.array([[95], [82]], dtype=float)

X\_train\_normalized = X\_train / np.amax(X\_train, axis=0)

X\_test\_normalized = X\_test / np.amax(X\_test, axis=0)

epoch = 7000

lr = 0.1

inputlayer\_neurons = 2

hiddenlayer\_neurons = 3

output\_neurons = 1

wh = np.random.uniform(size=(inputlayer\_neurons, hiddenlayer\_neurons))

bh = np.random.uniform(size=(1, hiddenlayer\_neurons))

wout = np.random.uniform(size=(hiddenlayer\_neurons, output\_neurons))

bout = np.random.uniform(size=(1, output\_neurons))

for i in range(epoch):

hinp1 = np.dot(X\_train\_normalized, wh)

hinp = hinp1 + bh

hlayer\_act = sigmoid(hinp)

outinp1 = np.dot(hlayer\_act, wout)

outinp = outinp1 + bout

output = sigmoid(outinp)

EO = y\_train - output

outgrad = derivatives\_sigmoid(output)

d\_output = EO \* outgrad

EH = d\_output.dot(wout.T)

hiddengrad = derivatives\_sigmoid(hlayer\_act)

d\_hiddenlayer = EH \* hiddengrad

wout += hlayer\_act.T.dot(d\_output) \* lr

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

wh += X\_train\_normalized.T.dot(d\_hiddenlayer) \* lr

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

hinp1\_test = np.dot(X\_test\_normalized, wh)

hinp\_test = hinp1\_test + bh

hlayer\_act\_test = sigmoid(hinp\_test)

outinp1\_test = np.dot(hlayer\_act\_test, wout)

outinp\_test = outinp1\_test + bout

output\_test = sigmoid(outinp\_test)

print("Input Test Data:\n", X\_test\_normalized)

print("Actual Output Test:\n", y\_test)

print("Predicted Output Test:\n", output\_test)

mse = np.mean((output\_test - y\_test) \*\* 2)

print("Mean Squared Error:", mse)

Output:

Input Test Data:

[[0.8 1. ]

[1. 0.375]]

Actual Output Test:

[[95.]

[82.]]

Predicted Output Test:

[[0.99999965]

[0.99999946]]

Mean Squared Error: 7698.500076612787