**PRACTICAL NO.10**

**10. Write a program to implement Neural network.**

**Ans.**

import numpy as np

# sigmoid function

def nonlin(x,deriv=False):

if(deriv==True):

return x\*(1-x)

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

# input dataset

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

[0,1,1],

[1,0,1],

[1,1,1] ])

# output dataset

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

# seed random numbers to make calculation

# deterministic (just a good practice)

np.random.seed(1)

# initialize weights randomly with mean 0

syn0 = 2\*np.random.random((3,1)) - 1

for iter in range(10000):

# forward propagation

l0 = X

l1 = nonlin(np.dot(l0,syn0))

l1\_error = y - l1

# multiply how much we missed by the

# slope of the sigmoid at the values in l1

l1\_delta = l1\_error \* nonlin(l1,True)

# update weights

syn0 += np.dot(l0.T,l1\_delta)

print("Output After Training:")

print(l1)

Output:-



