Roll No: 5117060

Aim: To implement a program demonstrating the working of a perceptron.

Code:

```
def multiply(weight, x):
    sum = 0
    for i in range(len(weight)):
        sum += weight[i] * x[i]
    return sum
N = int(input('\n Enter number of inputs:'))
c = float(input('\n Enter learning constant: '))
desired_op=[]
input_x = []
for i in range (N):
    temp = list(map(float,input('\n Enter x vector:').split(',')))
    input_x.append(temp)
    t = float(input('\n Enter desired output:'))
    desired_op.append(t)
weight = list(map(float,input('\n Enter weights:').split(',')))
print('\n Input vectors:',input_x)
print('\n Desired outputs:',desired_op)
print('\n Learning rate:',c)
print('\n Weights:',weight)
iterate = int(input('\n Enter number of iterations:'))
for i in range(iterate):
    print('\n Iteration Number:',i+1)
    for j in range(N):
        print('\n Input number:',j+1)
        net = multiply(weight,input_x[j])
        print('\n Net[',j+1,']=',net)
        if (net <= 0):
            o = 0
        else:
            o = 1.0

        print('\n Actual Output:{0} Desired Output
{1}'.format(o,desired_op[j]))

        if o == desired_op[j]:

            break

        print("\n Since Actual Output is not equal to desired
output.\nTherefore, change Weights")

        delta =list(c*(desired_op[j] - o) * k for k in input_x[j])

        print('\n Delta_w =',delta)

        for m in range(len(weight)):

            weight[m] += delta[m]

        print('\n Updated weights:',weight)
```

Output:

```
Enter number of inputs:3

 Enter learning constant: 1

 Enter x vector:1,2

 Enter desired output:1

 Enter x vector:-1,2

 Enter desired output:0

 Enter x vector:0,-1

 Enter desired output:0

 Enter weights:1.0,-0.8

 Input vectors: [[1.0, 2.0], [-1.0, 2.0], [0.0, -1.0]]

 Desired outputs: [1.0, 0.0, 0.0]

 Learning rate: 1.0

 Weights: [1.0, -0.8]

 Enter number of iterations:1

 Iteration Number: 1

 Input number: 1

 Net[ 1 ]= -0.6000000000000001
 Actual Output:0 Desired Output 1.0

 Since Actual Output is not equal to desired output.
Therefore, change Weights

 Delta_w = [1.0, 2.0]

 Updated weights: [2.0, 1.2]

 Input number: 2

 Net[ 2 ]= 0.3999999999999999

 Actual Output:1.0 Desired Output 0.0

 Since Actual Output is not equal to desired output.
Therefore, change Weights

 Delta_w = [1.0, -2.0]

 Updated weights: [3.0, -0.8]

 Input number: 3

 Net[ 3 ]= 0.8

 Actual Output:1.0 Desired Output 0.0

 Since Actual Output is not equal to desired output.
Therefore, change Weights

 Delta_w = [-0.0, 1.0]

 Updated weights: [3.0, 0.19999999999999996]
>>>
```