**Experiment No. 06**

**Aim:** Program to implement learning : Perceptron Learning

**Code:**

import numpy as np

X1 = np.array([1,0,0,1,0,0,1,1,1])

X2 = np.array([1,0,0,1,0,0,1,1,1])

X3 = np.array([1,1,0,1,0,0,1,1,1])

X4 = np.array([1,0,0,1,0,0,1,1,1])

X5 = np.array([1,0,0,1,0,1,1,1,1])

X6 = np.array([1,0,1,1,1,1,1,0,1])

X7 = np.array([1,0,1,1,1,1,1,0,1])

X8 = np.array([1,0,1,1,1,1,1,0,1])

X9 = np.array([1,0,1,1,1,1,1,1,1])

X10 = np.array([1,1,1,1,1,1,1,0,1])

X = np.array([X1, X2, X3, X4, X5,X6,X7,X8,X9,X10])

W = np.array([1, -1, 0, 0.5,0,1,1,0.5,1])

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

c = 1

epochs = 2

i=0

for i in range(epochs):

print("Iteration ", i+1)

for j in range(len(X)):

net = np.dot(X[j], W)

if (net <= 0):

op = 0

elif net > 0:

op = 1

error = d[j] - op

dW = c\*error\*X[j]

W += dW

print("W", j, W)

print("\nW after ", i+1, " epochs ", W)

# c=c/2

print("Final W after ", epochs, "epochs:")

print(W)

j=0

net = np.dot([1,0,0,1,0,0,1,1,0], W)

if (net <= 0):

op = 0

elif net > 0:

op = 1

print("op",":", op)

