**Practical No: 6(B)**

**Aim: To implement perceptron concept.**

**Code:**

class Perceptron : # With 2 inputs and 1 output

def \_\_init\_\_(self, a,b, c, tval):

self.x = a # input vector

self.result = b

self.cresult = c

self.threshold = tval by activation function

self.w = []

def h(self, tw): # calculating summation(hypothesis function)

hresult= []

for i in range(0 , len(self.result)):

hresult.append(0)

#print("index - ", i, ";", hresult)

for j in range(0,len(tw)):

#print("i=",i, ",j=",j)

hresult[i] = hresult[i] + ( tw[j][i]\*self.x[j][i] )

return hresult

def checkthreshold(self, hresult):

actfun =[]

for i in range(0 , len(self.result)) :

if (hresult[i] <= self.threshold ):

actfun.append(0)

else :

actfun.append( 1)

print("Ans :", hresult)

print("result of act fun:", actfun)

for i in range(0 , len(self.x)) :

if (actfun[i] != self.result[i]) :

return False

return True

def training(self, tw, alpha):

i=1

while i<=100 : # Max 100 attempts

print("Attempt :", i)

hresult = self.h(tw)

if(self.checkthreshold(hresult)) : self.w = tw

**Roll No:04,37**

print("In Attempt number ", i, ", i got it! I think i have learnt enough. Your w's are --" )

for x in range(0,len(self.w)):

print("w", x, " --> ", self.w[x])

break

i = i +1

for j in range(0,len(self.result)) :

for k in range(0, len(tw)):

sum = 0

for n in range(0, len(tw)):

sum = sum + (self.cresult[j] - hresult[j]) \*self.x[n][j]

tw[k][j] = tw[k][j] + alpha\*sum

if(i>=100):

print("I am exhausted, tried 100 iterations! plz change something else...")

a = [ [1,1,1,1], [0,0,1,1] , [0,1,0,1] ] # x vector, x0 is dummy

b = [0,1,1,1]

c = [0.5, 0.7, 1.3, 1.5] # sample h values

p = Perceptron(a,b,c, 0.5) # threshold = 0.5

print("Whether reservation is done =", p.x[0])

print("Whether raining outside =", p.x[1])

print("with threshold value :", p.threshold)

r = p.h([ [0.5,0.5,0.5,0.5], [0.8, 0.8, 0.8, 0.8], [0.2, 0.2, 0.2, 0.2]])

print("status :", p.checkthreshold(r))

print("Example 1 -->") #with alpha as 0.01, you will not get result

p.training( [ [0.7,0.7,0.7,0.7], [0.5, 0.5, 0.5, 0.5], [0.4, 0.4, 0.4, 0.4]], 0.01)

print("Example 2 -->") #with alpha as 0.5, you will not get result

p.training( [ [0.7,0.7,0.7,0.7], [0.5, 0.5, 0.5, 0.5], [0.4, 0.4, 0.4, 0.4]], 0.5)

print("Example 3 -->")

p.training( [ [0.2,0.2,0.2,0.2], [0.3, 0.3, 0.3, 0.3], [0.5, 0.5, 0.5, 0.5]], 0.01)

**Output:**

Whether reservation is done = [1, 1, 1, 1]

Whether raining outside = [0, 0, 1, 1]

with threshold value : 0.5

Ans : [0.5, 0.7, 1.3, 1.5]

result of act fun: [0, 1, 1, 1]

status : True

Example 1 -->

Attempt : 1

Ans : [0.7, 1.1, 1.2, 1.6]

result of act fun: [1, 1, 1, 1]

Attempt : 2

Ans : [0.698, 1.084, 1.204, 1.591]

result of act fun: [1, 1, 1, 1]

Attempt : 3

Ans : [0.69602, 1.0686399999999998, 1.20784, 1.5828099999999998]

result of act fun: [1, 1, 1, 1]

Attempt : 4

Ans : [0.6940598, 1.0538944, 1.2115264000000001, 1.5753571]

result of act fun: [1, 1, 1, 1]

Attempt : 5

Ans : [0.692119202, 1.039738624, 1.215065344, 1.568574961]

result of act fun: [1, 1, 1, 1]

Example 2 -->

Attempt : 1

Ans : [0.7, 1.1, 1.2, 1.6]

result of act fun: [1, 1, 1, 1]

Attempt : 2

Ans : [0.6, 0.2999999999999997, 1.4000000000000001, 1.1499999999999995]

result of act fun: [1, 0, 1, 1]

Attempt : 3

Ans : [0.55, 1.1, 1.2, 2.725000000000002]

result of act fun: [1, 1, 1, 1]

Attempt : 4

Ans : [0.525, 0.29999999999999993, 1.4000000000000001, -2.7875000000000068]

result of act fun: [1, 0, 1, 0]

Attempt : 5

Ans : [0.5125, 1.1, 1.2, 16.506250000000023]

result of act fun: [1, 1, 1, 1]

Attempt : 53

Ans : [0.5, 1.1, 1.2, 1.9567633533440096e+27]

result of act fun: [0, 1, 1, 1]

In Attempt number 53 , i got it! I think i have learnt enough. Your w's are --

w 0 --> [0.5, 0.7, 0.7, 6.522544511146698e+26]

w 1 --> [0.30000000000000004, 0.5, 0.5, 6.522544511146698e+26]

w 2 --> [0.20000000000000007, 0.4, 0.4, 6.522544511146698e+26]

Example 3 -->

Attempt : 1

Ans : [0.2, 0.7, 0.5, 1.0]

result of act fun: [0, 1, 0, 1]

Attempt : 2

Ans : [0.203, 0.7, 0.532, 1.045]

result of act fun: [0, 1, 1, 1]

In Attempt number 2 , i got it! I think i have learnt enough. Your w's are --

w 0 --> [0.203, 0.2, 0.21600000000000003, 0.21500000000000002]

w 1 --> [0.303, 0.3, 0.316, 0.315]

w 2 --> [0.503, 0.5, 0.516, 0.515]