Name : Vishal Shrirang Madle

Prn : 2020BTECS00092

Lab : AIML Lab

ASSIGNMENT 07

Q1. Implement a simple perceptron learning algorithm for logical NOR. Use learning rate =0.2, W1 = 0.3, W2 = -0.2, bias = 0.4

X1	X2	Y
0	0	1
0	1	0
1	0	0
1	1	0

```
import numpy as np

# define Unit Step Function
def Avtivation_func(v):
    if v >= 0:
        return 1
    else:
        return 0

w = [0.3,-0.2]
b = 0.4
alpha = 0.2
def perceptron(x,w,b,val,alpha):
    sum = np.dot(w,x)+b
    ans = Avtivation_func(sum)
    err = val-ans
    for j in range(len(x)):
```

```
        w[j]=w[j]+alpha * err *x[j]

    return err

err = 1
num_of_iteration=0

Xor_set = [
[0,0,1],
[0,1,0],
[1,0,0],
[1,1,0]
]


# Training->
while(err):
    err = 0
    for i in range(len(Xor_set)):
        x = Xor_set[i][:-1]  #array of first two value from first list
        real_outPut = Xor_set[i][-1]    # only last value as realOutput
        ans = perceptron(x,w,b,real_outPut,alpha)
        err = err or ans
        num_of_iteration+=1


# Testing->
print("-----------after training Updated Weight will be -------------")
print(w)
print("final number of iteration for to update weight value : ",num_of_iteration-
1)
print("-----------------------------------------------------------------")


# ----------------------------testing with real output
def perceptronModel(x, w, b):
    v = np.dot(w, x) + b
    y = Avtivation_func(v)
    return y

# NOT Logic Function
# wNOT = -1, bNOT = 0.5
def NOT_logicFunction(x):
    wNOT = 1
    # bNOT = 0.4
```

```
    bNOT = -0.5
    return perceptronModel(x, wNOT, bNOT)

# OR Logic Function
# w1 = 1, w2 = 1, bOR = -0.5
def OR_logicFunction(x):
    # w = np.array([0.3, -0.2])
    # w=w;
    w=[-0.5, -0.6000000000000001]
    bOR = 0.4
    # bOR = -1
    return perceptronModel(x, [-0.5, -0.6000000000000001], bOR)

# NOR Logic Function
# with OR and NOT
# function calls in sequence
def NOR_logicFunction(x):

    output_OR = OR_logicFunction(x)
    # print("or:",output_OR)

    output_NOT = NOT_logicFunction(output_OR)
    # print("not:",output_NOT)
    return output_NOT

# testing the Perceptron Model

for i in range(len(Xor_set)):
    input_x1_x2 = Xor_set[i][:-1]  #array of first two value from first list
    print("NOR({}, {}) = {}".format(Xor_set[i][0],Xor_set[i][1],
NOR_logicFunction([Xor_set[i][0],Xor_set[i][1]])))
```

OUTPUT:

----------after training Updated Weight will be ----------

[-0.5, -0.6000000000000001]

final number of iteration for to update weight value : 15

NOR(0, 0) = 1

NOR(0, 1) = 0

NOR(1, 0) = 0

NOR(1, 1) = 0

---

2. Implement a multilayer perceptron learning algorithm for the dataset given below.

X1	X2	X3	X4	Y
1	1	0	1	1

W15	W16	W25	W26	W35	W36	W45	W46	W57	W67
0.3	0.1	-0.2	0.4	0.2	-0.3	0.1	0.4	-0.3	0.2

θ5	θ6	θ7
0.2	0.1	-0.3

Perform at least 2 iterations and show that the model is learning.

Code:

```
import math
Dataset ={
"W15":0.3,
"W16":0.1,
"W25":-0.2,
"W26":0.4,
"W35":0.2,
"W36":-0.3,
"W45":0.1,
"W46":0.4,
"W57":-0.3,
"W67":0.2,
"05":0.2,
"06":0.1,
"07":-0.3
}
alpha = 0.8
def perceptron(Dataset,num_or_iteration,Y,INPUT,alpha,j):
    print("Running Iteration ->", j)
```

```
    X5
=Dataset["W15"]*INPUT[0]+Dataset["W25"]*INPUT[1]+Dataset["W35"]*INPUT[2]+Dataset[
"W45"]*INPUT[3]+Dataset["05"]
    X6
=Dataset["W16"]*INPUT[0]+Dataset["W26"]*INPUT[1]+Dataset["W36"]*INPUT[2]+Dataset[
"W46"]*INPUT[3]+Dataset["06"]

#    formulae------->
    O5 = 1/(1+math.exp(-X5))
    O6 = 1/(1+math.exp(-X6))
    # print("O5",O5,"O6",O6)
    X7 = Dataset["W57"]*O5+Dataset["W67"]*O6+Dataset["07"]
    O7 = 1/(1+math.exp(-X7))
    # print("O7",O7)

    Error7 = O7*(1-O7)*(Y-O7)
    Error6 = O6*(1-O6)*Error7*Dataset["W67"]
# ------------------------above are formulae

    Dataset["W67"] += alpha*Error7*O6
    Error5 = O5*(1-O5)*Error7*Dataset["W57"]

    Dataset["W57"] += alpha*Error7*O5
    #    formulae------->
    Dataset["05"] = Dataset["05"]+alpha * Error5
    Dataset["06"] = Dataset["06"]+alpha * Error6
    Dataset["07"] = Dataset["07"]+alpha * Error7
    Dataset["W15"] +=alpha*Error5*INPUT[0]
    Dataset["W25"] +=alpha*Error5*INPUT[1]
    Dataset["W35"] +=alpha*Error5*INPUT[2]
    Dataset["W45"] +=alpha*Error5*INPUT[3]
    j+=1
    Dataset["W16"] +=alpha*Error6*INPUT[0]
    Dataset["W26"] +=alpha*Error6*INPUT[1]
    Dataset["W36"] +=alpha*Error6*INPUT[2]
    Dataset["W46"] +=alpha*Error6*INPUT[3]
    print("The ERROR values in the iteration: ")
    print("ERROR7 = ",Error7,"ERROR6",Error6,"ERROR5",Error5)
    print("Final Error Result : ",1-O7)
Y=1
num_or_iteration=0
INPUT = [1,1,0,1]
num_or_iteration = int(input("Enter the number of num_or_iteration:"))
print(num_or_iteration)
```

```
for i in range(num_or_iteration):
    perceptron(Dataset,num_or_iteration,Y,INPUT,alpha,i+1)
```

OUTPUT:

Enter the number of num_or_iteration:20

20

Running Iteration -> 1

The ERROR values in the iteration:

ERROR7 = 0.14167287072891396 ERROR6 0.005570915400376464 ERROR5 -0.010211528871801637

Final Error Result : 0.5825851010449373

Running Iteration -> 2

The ERROR values in the iteration:

ERROR7 = 0.13180040525292394 ERROR6 0.007269233392444829 ERROR5 -0.007396899236691631

Final Error Result : 0.5289784623791298

Running Iteration -> 3

The ERROR values in the iteration:

ERROR7 = 0.11942813186633403 ERROR6 0.00829852607062217 ERROR5 -0.004924401336639702

Final Error Result : 0.4785900437984745

Running Iteration -> 4

The ERROR values in the iteration:

ERROR7 = 0.10637076084065905 ERROR6 0.008727022374376907 ERROR5 -0.0029502403707565893

Final Error Result : 0.4332125349020165

Running Iteration -> 5

The ERROR values in the iteration:

ERROR7 = 0.09392262314725901 ERROR6 0.008717117066402663 ERROR5 -0.001476479275600442

Final Error Result : 0.3935334871953189

Running Iteration -> 6

The ERROR values in the iteration:

ERROR7 = 0.08275090698767996 ERROR6 0.0084358307828496 ERROR5 -0.00042367078301662383

Final Error Result : 0.35941707117777233

Running Iteration -> 7

The ERROR values in the iteration:

ERROR7 = 0.07305984490305194 ERROR6 0.008011727757550892 ERROR5 0.0003077408679799331

Final Error Result : 0.3302905613061029

Running Iteration -> 8

The ERROR values in the iteration:

ERROR7 = 0.0647966563164903 ERROR6 0.007529114582634309 ERROR5 0.0008064353638435985

Final Error Result : 0.3054359454414888

Running Iteration -> 9

The ERROR values in the iteration:

ERROR7 = 0.05779901635078246 ERROR6 0.007037652444611893 ERROR5 0.0011411565831470237

Final Error Result : 0.2841512641945476

Running Iteration -> 10

The ERROR values in the iteration:

ERROR7 = 0.0518768739942165 ERROR6 0.0065640179660181016 ERROR5 0.0013617554032637118

Final Error Result : 0.26581848339447933

Running Iteration -> 11

The ERROR values in the iteration:

ERROR7 = 0.04685012853754923 ERROR6 0.006120968114224145 ERROR5 0.0015032444614182625

Final Error Result : 0.249920406250412

Running Iteration -> 12

The ERROR values in the iteration:

ERROR7 = 0.042562357722380165 ERROR6 0.005713324448591781 4 ERROR5 0.0015898857677551746

Final Error Result : 0.2360348079729122

Running Iteration -> 13

The ERROR values in the iteration:

ERROR7 = 0.038883248966194085 ERROR6 0.005341616488268253 ERROR5 0.001638436421273457

Final Error Result : 0.2238206971257355

Running Iteration -> 14

The ERROR values in the iteration:

ERROR7 = 0.03570641447820967 ERROR6 0.0050041988705639764 ERROR5 0.001660496182904316

Final Error Result : 0.21300348912810896

Running Iteration -> 15

The ERROR values in the iteration:

ERROR7 = 0.032945763395341165 ERROR6 0.004698442304386055 ERROR5 0.0016641358155529667

Final Error Result : 0.2033617176018082

Running Iteration -> 16

The ERROR values in the iteration:

ERROR7 = 0.03053179761602063 ERROR6 0.004421383409359903 ERROR5 0.0016550063708678615

Final Error Result : 0.19471600660481214

Running Iteration -> 17

The ERROR values in the iteration:

ERROR7 = 0.028408339410907753 ERROR6 0.004170065543523569 ERROR5 0.0016370906677086394

Final Error Result : 0.1869202492011206

Running Iteration -> 18

The ERROR values in the iteration:

ERROR7 = 0.026529818279853096 ERROR6 0.00394170615664092 ERROR5 0.0016132130183343517

Final Error Result : 0.17985467489294527

Running Iteration -> 19

The ERROR values in the iteration:

ERROR7 = 0.024859090802884995 ERROR6 0.0037337684866590244 ERROR5 0.0015853867378272494

Final Error Result : 0.17342044256107558

Running Iteration -> 20

The ERROR values in the iteration:

ERROR7 = 0.023365715879304413 ERROR6 0.003543981818633227 ERROR5 0.001555052767596455

Final Error Result : 0.16753543071216026
