EEL-5813 Neural Networks SUMMER 'C' 2019

INDIVIDUAL PROJECT I **Perceptron and Hebbian Associator**

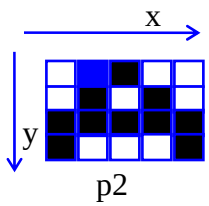
PART I: PERCEPTRON
I.1 "A DETECTOR"

Following the instructions in FIGURE 1, you will create 25 TRAINING PATTERNS p1, p2 … p25. Each of the patterns will be transformed into a column vector by reading the pixels of the 4 x 5 letter image, column, by column, starting with the left-most column (from top to bottom), encoding each "ON" PIXEL as a "1" and each "OFF" PIXEL as a "-1".

So, for example:

p1 = [-1 -1 1 1 -1 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 -1 1 1]^T

According to FIGURE 1, p2 (as a matrix) is obtained from p1 by turning ON pixel x=2, y = 1. This results in the following character:



Expressed as a column vector:
p2 = [-1 -1 1 1 **+1** 1 1 -1 1 -1 1 -1 -1 1 1 -1 -1 -1 1 1]^T

Follow the same process to:
Create p3, p4, p5 changing 1 pixel in p1 (This yields 5 "A" TRAINIG PATTERNS)
Create p7, p8, p9, p10 changing 1 pixel in p6 (This yields 5 "E" TRAINIG PATTERNS)
Create p12, p13, p14, p15 changing 1 pixel in p11 (This yields 5 "I" TRAINIG PATTERNS)
Create p17, p18, p19, p20 changing 1 pixel in p16 (This yields 5 "O" TRAINIG PATTERNS)
Create p22, p23, p24, p25 changing 1 pixel in p21 (This yields 5 "U" TRAINIG PATTERNS)

You will train and test a PERCEPTRON with a single output activation. This will be an "A DETECTOR" Therefore the 5 targets for the training patterns that represent variations of "A" will be +1. The remaining targets (those for patterns that represent variations of "E", "I" "O" or "U") will be -1: That is:

t1 = t2 = t3 = t4 = t5 = +1

t6 = t7 = t8 = t9 = t10 = t11 = t12 = t13 = t14 = t15 = t16 = t17 = t18 = t19 = t20 =t21=t22 = t23 = t24 = t25 = -1 .

INDICATE how many epochs of training took place and TABULATE your final weights and bias.

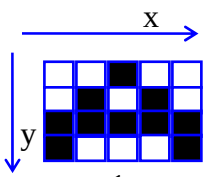
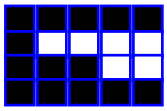
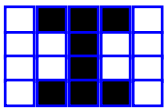
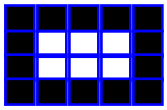
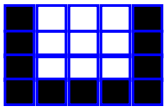
	Turn ON for training set: (x,y)
 p1	(2,1) (3,2) (3,4) (4,4)
 p6	(2,2) (4,2) (5,3) (4,3)
 p11	(1,1) (4,2) (1,4) (4,3)
 p16	(2,2) (4,2) (2,3) (4,3)
 p21	(2,1) (4,1) (2,3) (4,3)

FIGURE 1: Creation of the TRAINING (INPUT) PATTERNS

CREATING 3 PROGRESSIVELY MORE CHALLENGING TEST SETS

Once you have trained your "A DETECTOR" with the 25 {p | t} pairs, you will TEST its performance three different times, with 3 different TEST SETS (TSET1, TSET2 TSET3), each containing 5 distorted A s, 5 distorted E s, 5 distorted I s, 5 distorted O s, , and 5 distorted U s.(Because you will create these "pt" patterns, you will know which letter they are supposed to be, and you will be able to say if the perceptron classified each one correctly (a "hit") or incorrectly (a "miss").

INSTRUCTIONS FOR THE CREATION OF TEST SET TSET1 (Lowest noise set):

The 5 A s in this test set will be obtained as follows:

pt101 = p1 (unchanged). That is, the "perfect A" is allowed to be one of the 5 As in TSET1
pt102 is created FROM p1, by TOGGLING the value of the pixel indicated in the leftmost column of FIGURE 2. That is pt102 is the same as p1 except that it has the pixel at x = 4, y = 1 ON, instead of it being OFF.
pt103 is created FROM p1, by TOGGLING the value of the pixel indicated in the leftmost column of FIGURE 2. That is pt103 is the same as p1 except that it has the pixel at x = 5, y = 3 OFF, instead of it being ON.
pt104 is created FROM p1, by TOGGLING the value of the pixel indicated in the leftmost column of FIGURE 2. That is pt104 is the same as p1 except that it has the pixel at x = 2, y = 4 ON, instead of it being OFF.
pt105 is created FROM p1, by TOGGLING the value of the pixel indicated in the leftmost column of FIGURE 2. That is pt105 is the same as p1 except that it has the pixel at x = 1, y = 2 ON, instead of it being OFF.

Follow the same process to:
Crete pt106, pt107, pt108, pt109 and pt110 from p6 (Noisy E s)
Crete pt111, pt112, pt113, pt114 and pt115 from p11 (Noisy I s)
Crete pt116, pt117, pt118, pt119 and pt120 from p16 (Noisy O s)
Crete pt121, pt122, pt123, pt124 and pt125 from p21 (Noisy U s)

INSTRUCTIONS FOR THE CREATION OF TEST SET TSET2 (medium noise set):

For the creation of TSET2 we will ADD ONE MORE PIXEL TOGGLE to most of the patterns in TSET1, except for the "perfect" characters (p1 , p6, p11, p16, 21), which will be included into TSET2 unchanged.

The ADDITIONAL PIXELS TO BE TOGGLED are listed in the second column of Figure 2.

So, for example:
pt201 = pt101 = p1
pt202 is obtained from pt102 toggling the pixel at x = 3, y=2 (was OFF, it will now be ON)
pt203 is obtained from pt103 toggling the pixel at x =1, y=3 (was ON, it will now be OFF)
pt204 is obtained from pt104 toggling the pixel at x = 3, y=3 (was ON, it will now be OFF)
pt205 is obtained from pt105 toggling the pixel at x = 2, y=1 (was OFF, it will now be ON)

Follow the same process to create the remaining 20 patterns in TSET2 (pt206 …. pt225).

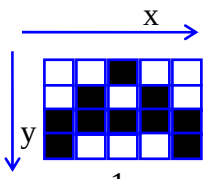
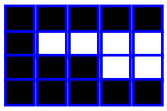
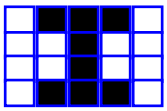
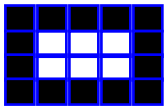
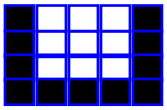
	TOGGLE for Test Set:		
	TSET1	TSET2	TSET3
 p1	(4,1) (5,3) (2,4) (1,2)	(3,2) (1,3) (3,3) (2,1)	(5,2) (3,3) (1,3) (5,1)
 p6	(2,4) (5,2) (4,3) (1,3)	(3,2) (3,3) (5,4) (1,2)	(3,4) (1,4) (5,2) (1,3)
 p11	(4,4) (2,4) (2,2) (3,4)	(1,3) (2,2) (4,3) (1,4)	(1,1) (1,2) (2,4) (2,1)
 p16	(1,3) (3,2) (5,4) (3,3)	(4,3) (4,3) (5,1) (2,4)	(1,2) (4,2) (3,3) (1,4)
 p21	(5,3) (3,4) (1,4) (5,4)	(3,3) (2,2) (2,1) (1,3)	(3,2) (5,2) (5,4) (4,2)

FIGURE 2: Creation of the THREE TEST SETS

INSTRUCTIONS FOR THE CREATION OF TEST SET TSET3 (highest noise set):

For the creation of TSET3 we will ADD ONE MORE PIXEL TOGGLE to most of the patterns in TSET2, except for the "perfect" characters (p1 , p6, p11, p16, 21), which will be included into TSET2 unchanged.

The ADDITIONAL PIXELS TO BE TOGGLED are listed in the right-most column of Figure 2.

So, for example:
pt301 = pt201 = pt101 = p1
pt302 is obtained from pt202 toggling the pixel at x = 5, y=2 (was OFF, it will now be ON)
pt303 is obtained from pt203 toggling the pixel at x = 3, y=3 (was ON, it will now be OFF)
pt304 is obtained from pt204 toggling the pixel at x = 1, y=3 (was ON, it will now be OFF)
pt305 is obtained from pt205 toggling the pixel at x = 5, y=1 (was OFF, it will now be ON)

Follow the same process to create the remaining 20 patterns in TSET3 (pt306 …. pt325).

TEST RESULTS:
You will apply the 3 test sets, separately. You will indicate the result ('HIT" or "MISS") for each of the 25 patterns in each set (in a table) , and you will calculate the "Hit Ratio" (HR) for each test set:

HR = (Number of hits) / 25

(NOTE: ALSO tabulate the hits/misses for the 25 TRAINIG PATTERNS and calculate the HR.)

I.2 "E DETECTOR"

To create a perceptron "E Detector" you will use the same architecture as for I.1 and the same set of 25 training patterns (p1, …p25).

THE ONLY CHANGE NEEDED is a change of the target values:

t1 = t2 = t3 = t4 = t5 = -1 (because p1, …, p5 are A s)

t6 = t7 = t8 = t9 = t10 = +1(because p6, ..p10 are the "E patterns" this network should detect)

t11 = t12 = t13 = t14 = t15 = t16 = t17 = t18 = t19 = t20 =t21=t22 = t23 = t24 = t25 = -1 (I,O,U)

You will test and report this "E DETECTOR" as you did for the "A DETECTOR".

I.3 COMMENTS: Write comments and observations about the training and testing of the "A DETECTOR" and the "E DECTECTOR". Comment on any differences between the 2. Provide overall comments on the complete perceptron experimentation.

PART II: HEBBIAN ASSOCIATOR

To train the associator we will only use the "perfect characters", associating each one of them to a 3-feature (3 "bit") pattern:

p1 associated to t1 = $[-1 \ -1 \ +1]^T$ (Character A association)
p6 associated to t6 = $[-1 \ +1 \ -1]^T$ (Character E association)
p11 associated to t11 = $[-1 \ +1 \ +1]^T$ (Character I association)
p16 associated to t16 = $[+1 \ -1 \ -1]^T$ (Character O association)
p21 associated to t21 = $[+1 \ -1 \ +1]^T$ (Character U association)

II.1 Training the associator with the "Standard Hebb" method.

You will use the "Standard Hebb Learning" to determine the matrix of weights for the associator.

Testing: You will test with each of the 3 test sets (TSET1, TSET2 and TSET3) separately
For each of the tests sets, create a table including in each row:

- The "name" of the pattern applied to the associator (for example "pt101").
- The output pattern obtained, transposed (for example [-1 +1 -1])
- The number of "wrong bits" ("Hamming Distance") in the output pattern obtained
- Whether the pattern was a "HIT" (0 wrong bits) or a "MISS" (otherwise)

At the bottom of each table calculate the "Hit Ratio" (HR).

II.2 Training the associator with the "Pseudo-Inverse" method.

You will use the "Pseudo-Inverse method" to determine the matrix of weights for the associator.

Testing: You will test and report your results in the same way as for II.1 ,

II.3 COMMENTS: Write comments on the training and testing of the associator using the "Standard Hebb" method. Write comments on the training and testing of the associator using the "Pseudo-Inverse" method. Comment on any differences between the 2. Provide overall comments on the complete Hebbian associator experimentation.