Raiid Ahmed Homework 3 + 4 9/28/20

Probloml:

Given:

Hetero associative net

Hebb rule by outer products

$x = 9 \times 7$

$y = 5 \times 3$

example 3.9

figure 2.20

project 3.4

Find:

number of pattern pair capacity

Can this net respond to noisy input?

Testing # of pairs:

Define pair as an individual S and T

1 through 7 patterns: 1/1, 2/2, 3/3, 1/4,

5/5, 3/6, 4/7 correct

Analysis:

After reaching 3 pairs stored, accuracy greatly decreases. Inputs don't seem to be orthogonal, so selecting distinct pairs could help performance. For testing noise effects, we will use the best case scenario of five patterns.

Testing Noisy input:

Use case with 5 patterns

25% noise, 5 trials: 4/5, 3/5, 4/5,
1/5, 4/5 correct

Analysis:

With 25% noise, we get an average accuracy of 4/5. This is a performance decrease, but not as big as I was expecting. This is probably due to the small set of patterns. Future noise tests will report the average.

Problem 2:

Given:

Hebb rule outer product

input 7x5 → vector

35 tuple

training mode

test mode

output 7x5

input set

Find:

- # of patterns to store and recall successfully
  - which ones work better?
  - orthogonal?
- how much noise?
  - is this dependent on pattern count?
- how does batch vs one at a time change things?

Testing Binary / Bipolar!

1 through 10 patterns ; 1/1, 0/all forresh

Testing # of patterns!

Defining pattern as an input S

1 through 10 patterns: 1/1, 2/2, 3/3, 2/4,
2/5, 1/6, 0/7, 0/8,
0/9, 0/10 correct

Testing orthogonal!

Qualitative assesment by MC shows that 1, 2, 4, 6, 7 should be distinct

Results:

5 distinct patterns: 4/5 correct

Analysis:

Our inputs are numbers, and a lot of numbers look very similar. This is why I was not surprised by the large drop of performance when loading >3 patterns in the net. Selecting distinct looking numbers improved performance for 5 patterns loaded. The net failed to work with binary input.

Testing noisy input w/ # of patterns:

10% noise:

1 through 5 distinct patterns: 1/1, 2/2, 3/3, 4/4, 3/5

25% noise:

1 through 5 distinct patterns: 1/1, 2/2, 3/3, 3/4, 3/5

Analysis:

There was a noticeable drop in performance when adding noise. From this test, we can observe that decreasing the noise level improves accuracy for a larger # of patterns.

Testing batch iteration:

0% noise

1 through 10 patterns: 1/1, 2/2, 3/3, 4/4, 2/5, 1/6, 0/7
3/8, 0/9, 0/10

25% noise

1 through 10 patterns: 1/1, 2/2, 3/3, 3/4, 1/5, 1/6, 1/7,
2/8, 1/9, 0/9, 0/10

5 distinct patterns: 5/5

Testing 1 by 1 iteration:

0% noise

1 through 10 patterns: 1/1, 2/2, 3/3, 4/4, 2/5, 1/6, 0/7, 0/8, 0/9, 0/10 correct

25% noise

1 through 10 patterns: 1/1, 2/2, 3/3, 3/4, 2/5, 1/6, 0/7, 0/8, 0/9, 0/10 correct

5 distinct patterns! 5/5 correct

Analysis:

Both one by one and batch iteration improved net performance over the non-iterative net. Of the two, Batch iteration was more effective.

Problem 3:

Given:

Figure 2.20 input set

Discrete hopfield architecture

Find:

- # of patterns that can be stored and recalled
- Can net take noisy input?

Test # of patterns w/ Noisy input:

0% noise

Testing 1 through 7 patterns: 1/1, 2/2, 3/3, 1/4, 1/5, 1/6, 0/7 correct

10% noise:

Testing 1 through 7 patterns: 1/1, 2/2, 3/3, 1/4, 1/5, 0/6, 0/7 correct

25% noise:

Testing 1 through 7 patterns: 1/1, 2/2, 3/3, 1/4, 1/5, 0/6, 0/7 correct

Analysis:

There was a huge dropoff in performance after 3 patterns were stored. The dropoff was so dramatic that noise had a small effect. This is likely due to reaching the net's storage capacity, as there were many unique inputs.

Problem 4:

Given:

Bipolar
BAM
15 unit X layer
3 unit Y layer
input sets
Example 3.23

Find:

- Test cases for A and C w/ noise
- Can we store 8 patterns?
  - Test for noise

Find hammings for Y layer
- Corellate and find best letter pairs

- Test cases past max allowable patterns

Test part A:

Noisy Y vector, No X vector: 0/1 correct

Noisy Y vector, Noisy X vector w/ 25% noise: 1/1 correct

Test part B # of patterns w/noise:

* correct when s and t both match

0% noise on s and t:

Testing 1 through 8 patterns: 1/1, 2/2, 3/3, 3/4, 3/5, 5/6, 3/7, 5/8

10% noise on s and t:

correct

Testing 1 through 8 patterns: 1/1, 2/2, 3/3, 3/4, 3/5, 3/6, 5/7, 4/8

25% noise on s and t:

correct

Testing 1 through 8 patterns: 1/1, 2/2, 2/3, 2/4, 2/5, 3/6, 4/7, 3/8

Analysis:

I was able to replicate test cases from the example in part 3.

In part B, accuracy dropped off incrementally as patterns were added as opposed to a massive drop off in previous problems. Noise affected the net, but again only incrementally as expected. Accuracy was 100% before reaching the upper limit of min(15, 3).

Hamming distance:

Y layer

	A	B	C	D	E	F	G	H
A	0	1	1	2	1	2	2	3
B		0	2	2	2	1	3	2
C			0	1	2	3	1	1
D				0	3	2	2	1
E					0	1	1	2
F						0	2	1
G							0	1
H								0

From table in book

Pairs with hamming distance > seven!

CA, CB, DC, GB, GF, HC

From here!

Pairs with hamming distance = 3!

ED, FC, GB, HA

Therefore, GB should be the easiest pair to store