The University of Aizu

Neural Network I - Fundamental Theory and Applications

Project 4

Lecturer:	Q. F. Zhao
ID:	M5128110
Name:	Fu, Yu-Hsiang
Date:	2009/05/15

Contents

1. Environment
2. Given problem
3. Patterns
4. Schema
5. Operation step
6. Results
7. Improve idea

Reference

1. Environment

The experiment environment in this report is HP Presario CQ45. The hardware is performed on Intel Dual Core Processor P7350 CPU, 4GB RAM and Microsoft Windows Vista Ultimate SP1 operating system. The software is performed on Java 1.6.0 Update 11 with IDE Gel RC40.

2. Given problem

Suppose that a Hopfield neural network contains 12x10 neurons. We want to store some patterns given as follows (not necessarily the same) into the network, and recall any of them with a pattern corrupted by noise.

Write a computer program to implement the algorithm:

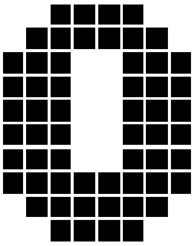
1. Try to recall the patterns with the noise level being 0%, 10% or 15%.
2. Try to recall the patterns when part of the image is missing.
3. The percentage of missing data is 10% or 15%.

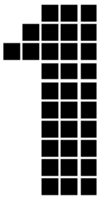
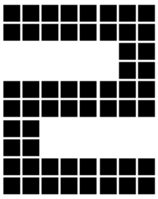
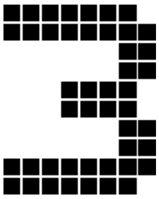
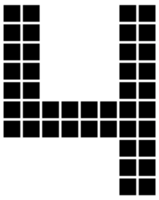
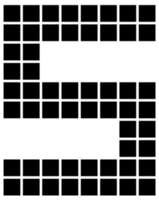
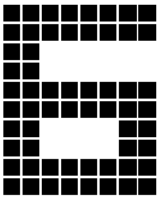
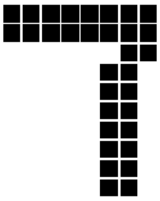
3. Patterns

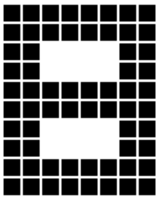
The patterns of project 4 that are the numbers which is 0 to 9 are shown as table 1. In the pattern, the “1” denotes white; the “0” denotes black.

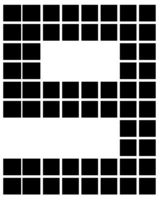
Table 1. The patterns of project 4

Pattern number	Input form	Binary form

0	1 1 1 1 0 0 0 0 1 1 1 1
	1 1 1 0 0 0 0 0 0 1 1 1
	1 1 0 0 0 1 1 0 0 0 1 1
	1 1 0 0 0 1 1 0 0 0 1 1
	1 1 0 0 0 1 1 0 0 0 1 1
	1 1 0 0 0 1 1 0 0 0 1 1
	1 1 0 0 0 1 1 0 0 0 1 1
	1 1 0 0 0 0 0 0 0 0 1 1
	1 1 1 0 0 0 0 0 0 1 1 1
	1 1 1 1 0 0 0 0 1 1 1 1
		

1	111110001111 111100001111 111000001111 111110001111 111110001111 111110001111 111110001111 111110001111 111110001111 111110001111	
2	111000000001 111000000001 111111111001 111111111001 111000000001 111000000001 111001111111 111001111111 111000000001 111000000001	
3	111100000001 111100000000 111111111100 111111111100 111111100001 111111100001 111111111100 111111111100 111100000000 111100000001	
4	110011110011 110011110011 110011110011 110011110011 110011110011 110000000011 110000000011 111111110011 111111110011 111111110011	
5	110000000011 110000000011 110011111111 110011111111 110000000011 110000000011 111111110011 111111110011 110000000011 110000000011	
6	000000001111 000000001111 001111111111 001111111111 000000001111 000000001111 001111001111 001111001111 000000001111 000000001111	
7	110000000011 110000000011 111111110011 111111100111 111111100111 111111100111 111111100111 111111100111 111111100111 111111100111	

8	1 1 0 0 0 0 0 0 0 0 1 1
	1 1 0 0 0 0 0 0 0 0 1 1
	1 1 0 0 1 1 1 1 0 0 1 1
	1 1 0 0 1 1 1 1 0 0 1 1
	1 1 0 0 0 0 0 0 0 0 1 1
	1 1 0 0 0 0 0 0 0 0 1 1
	1 1 0 0 1 1 1 1 0 0 1 1
	1 1 0 0 1 1 1 1 0 0 1 1
	1 1 0 0 0 0 0 0 0 0 1 1
	1 1 0 0 0 0 0 0 0 0 1 1
		

9	1 1 1 1 0 0 0 0 0 0 0 0
	1 1 1 1 0 0 0 0 0 0 0 0
	1 1 1 1 0 0 1 1 1 1 0 0
	1 1 1 1 0 0 1 1 1 1 0 0
	1 1 1 1 0 0 0 0 0 0 0 0
	1 1 1 1 0 0 0 0 0 0 0 0
	1 1 1 1 1 1 1 1 1 1 0 0
	1 1 1 1 1 1 1 1 1 1 0 0
	1 1 1 1 0 0 0 0 0 0 0 0
	1 1 1 1 0 0 0 0 0 0 0 0
		

4. Schema

Due to the given input is formed by 0 and 1. Therefore, in the store phase, the formula should be used as equation (1).

$$w_{ij} = (1-\delta_{ij}) \cdot \sum_{m=1}^{p} (2s_i^m - 1)(2s_j^m - 1) \quad (1)$$

In the results of project 4, I choose the Euclidean distance to be the recall measurement to calculate the quality of recalling a pattern that means how close the distance between current state vector and original pattern vector is. The Euclidean distance and the normalization are as equation (2) and (3).

Euclidean distance:

$$D(p_i, v_j) = \sqrt{\sum_{k=1}^{m} (p_{ki} - v_{kj})^2} \quad (2)$$

where p_i is original pattern vector and v_j is current state vector

Normalization:

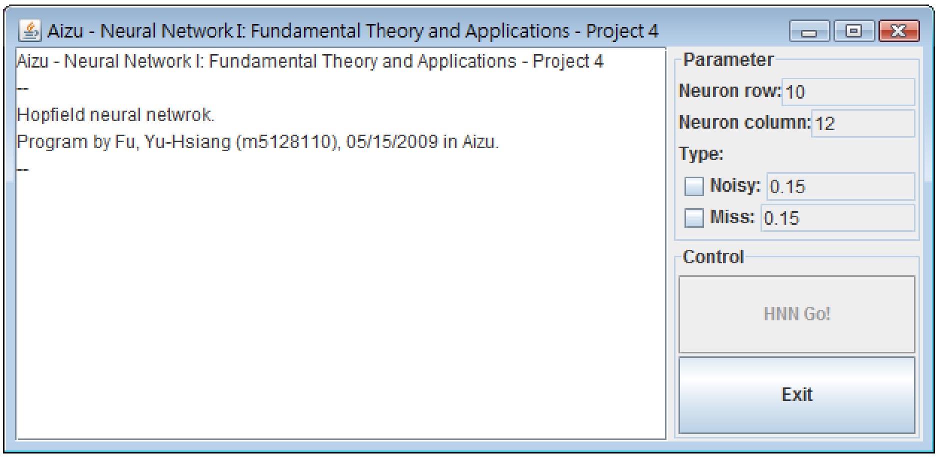
$$ND(p_i, v_j) = \sqrt{\frac{\sum_{k=1}^{m} (p_{ki} - v_{kj})^2}{m}} \quad (3)$$

Determining the grade of recalled pattern, I define the 6 grades that are based on the similarity (distance) which is calculated by Euclidean distance formula and denotes how the recalled degree of given input is. The reason to define the 6 grade is that it can be used to understand how recall effect which is affected by noisy or missing is. The 6 grades are defined as follows:

```
If (similarity>=0.9) then
    Grade is A
Else if (similarity>=0.8) then
    Grade is B
Else if (similarity>=0.7) then
    Grade is C
Else if (similarity>=0.6) then
    Grade is D
Else if (similarity>=0.5) then
    Grade is E
Else if (similarity<0.5) then
    Grade is F
```

5. Operation step

The programs of project 4 have two kinds of program. The first is the program with GUI, as figure 1, which can be used for testing the specific given patters set, for example, $Pattern = \{0,1,2,3\}$ which is the 4 patterns case. The second is the program of testing all combinations of 1 to 9 patterns.



Aizu - Neural Network I: Fundamental Theory and Applications - Project 4

Aizu - Neural Network I: Fundamental Theory and Applications - Project 4
--
Hopfield neural netwrok.
Program by Fu, Yu-Hsiang (m5128110), 05/15/2009 in Aizu.
--

Parameter
Neuron row: 10
Neuron column: 12
Type:
☐ Noisy: 0.15
☐ Miss: 0.15

Figure 1. The screenshot of program 1 of project 4

The program 1 can be executed by using double-click on the “HopfieldNetwork.jar” at the directory “/NNI/proj-04/code/”. The patterns can be found at the directory “/NNI/proj-04/code/pattern/”. After patterns are chosen, then move them to the directory “/NNI/proj-04/code/input/”. The operation steps of program are as follows:

Step1. Choose type
Step2. Set parameter of chosen type
Step3. Click “HNN Go!”
Step4. “Exit”

The second program of project 4 is at the directory “/proj-04 /code2/”. The follow result shows the partial result of combinations of choosing 4 patterns. The testing result can be found at the directory “/proj-04 /result/ result_code2_4 patterns_noisy 0%.txt”.

The partial result of total combinations of choosing 4 patterns:

```
…
Pattern:4679, Avg. recall quality: 95.44%, A
Pattern:4689, Avg. recall quality: 96.77%, A
Pattern:4789, Avg. recall quality: 90.87%, A
Pattern:5678, Avg. recall quality: 90.94%, A
Pattern:5679, Avg. recall quality: 83.34%, B
Pattern:5689, Avg. recall quality: 93.55%, A
Pattern:5789, Avg. recall quality: 77.83%, C
Pattern:6789, Avg. recall quality: 85.93%, B
A:53
B:75
C:55
D:22
E:5
F:0
```

In the example result as above, the first value is the combination of chosen patterns. The second value is the average recall quality (similarity) of recalling each pattern of chosen patterns. For example, the chosen pattern is as the second row of example result, $Pattern = \{4,6,7,9\}$. The average recall quality is 95.44% which is the average of recalling pattern 4, 6, 7 and 9. The last value, A, is the grade of average recall quality.

6. Results

In the section 6, it is briefly discussing about the results of project 4. The results include two parts. The first part is show the result of given specific patterns by using program 1. The results of given patterns, $Pattern = \{0,2,6,9\}$ which is one of the best cases, are shown as table 2 to 6. The detail results can be found at the directory "/proj-04 /result/4 patterns/a_0269/". The results with different missing rates are "result_miss_10%_0269.txt" and "result_miss_15%_0269.txt". The results with different noisy rates are "result_moisy_0%_0269.txt", "result_moisy_10%_0269.txt" and "result_moisy_15%_0269.txt".

Table 2. The result of 0246 with noisy rate 0%

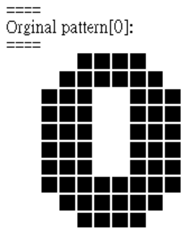
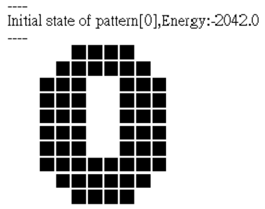
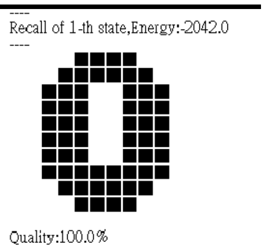
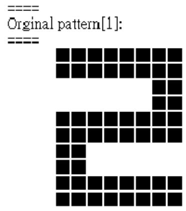
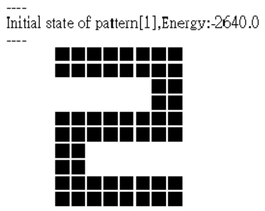
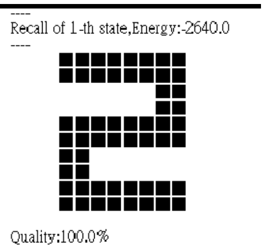
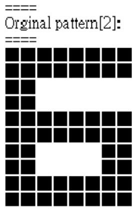
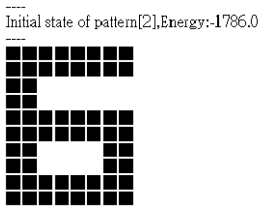
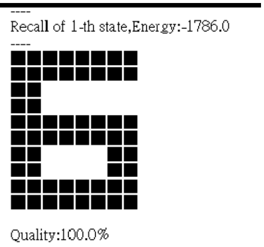
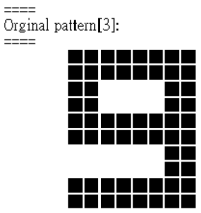
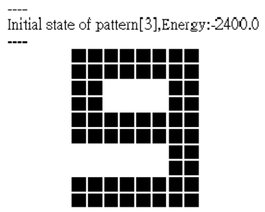
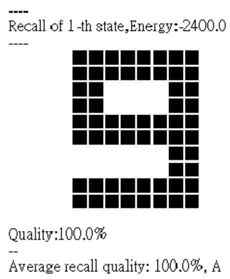
Original	Initial state	Recall 1-th
 ==== Orginal pattern[0]: ====	 ---- Initial state of pattern[0],Energy:-2042.0 ----	 ---- Recall of 1-th state,Energy:-2042.0 ---- Quality:100.0%
 ==== Orginal pattern[1]: ====	 ---- Initial state of pattern[1],Energy:-2640.0 ----	 ---- Recall of 1-th state,Energy:-2640.0 ---- Quality:100.0%
 ==== Orginal pattern[2]: ====	 ---- Initial state of pattern[2],Energy:-1786.0 ----	 ---- Recall of 1-th state,Energy:-1786.0 ---- Quality:100.0%
 ==== Orginal pattern[3]: ====	 ---- Initial state of pattern[3],Energy:-2400.0 ----	 ---- Recall of 1-th state,Energy:-2400.0 ---- Quality:100.0% -- Average recall quality: 100.0%, A

Table 3. The result of 0246 with noisy rate 10%

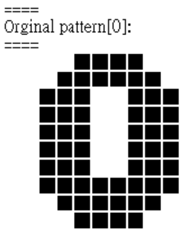
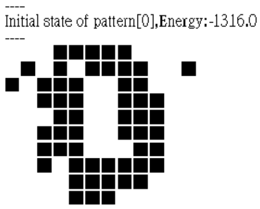
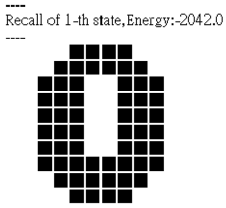
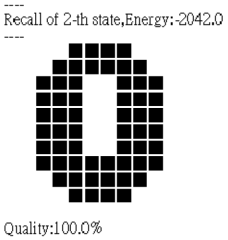
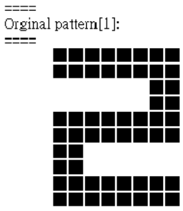
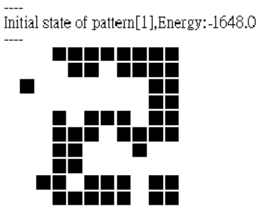
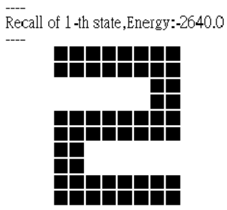
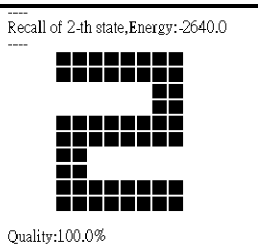
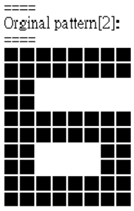
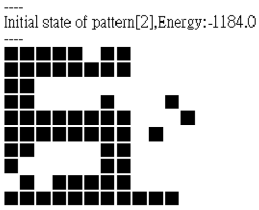
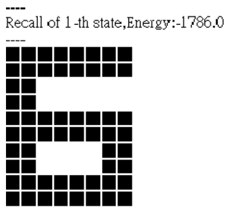
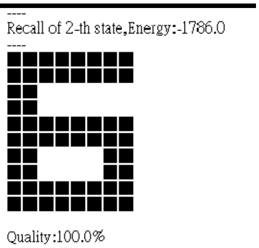
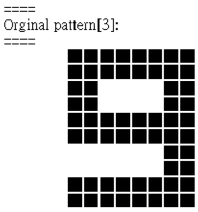
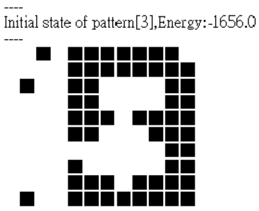
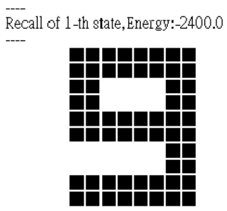
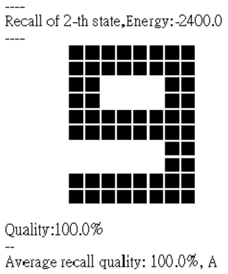
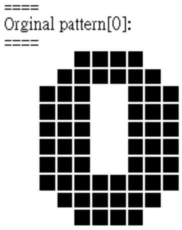
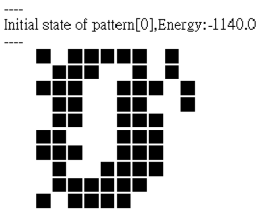
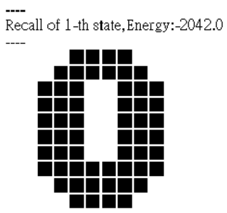
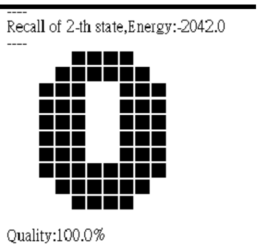
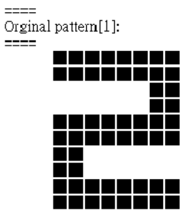
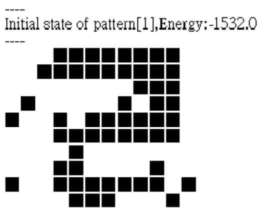
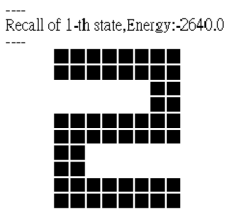
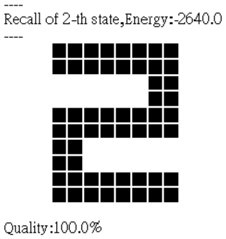
Original	Initial state	Recall 1-th	Recall 2-th
 ==== Orginal pattern[0]: ====	 ---- Initial state of pattern[0],Energy:-1316.0 ----	 ---- Recall of 1-th state,Energy:-2042.0 ----	 ---- Recall of 2-th state,Energy:-2042.0 ---- Quality:100.0%
 ==== Orginal pattern[1]: ====	 ---- Initial state of pattern[1],Energy:-1648.0 ----	 ---- Recall of 1-th state,Energy:-2640.0 ----	 ---- Recall of 2-th state,Energy:-2640.0 ---- Quality:100.0%
 ==== Orginal pattern[2]: ====	 ---- Initial state of pattern[2],Energy:-1184.0 ----	 ---- Recall of 1-th state,Energy:-1786.0 ----	 ---- Recall of 2-th state,Energy:-1786.0 ---- Quality:100.0%
 ==== Orginal pattern[3]: ====	 ---- Initial state of pattern[3],Energy:-1656.0 ----	 ---- Recall of 1-th state,Energy:-2400.0 ----	 ---- Recall of 2-th state,Energy:-2400.0 ---- Quality:100.0% -- Average recall quality: 100.0%, A

Table 4. The result of 0246 with noisy rate 15%

Original	Initial state	Recall 1-th	Recall 2-th
 ==== Orginal pattern[0]: ====	 ---- Initial state of pattern[0],Energy:-1140.0 ----	 ---- Recall of 1-th state,Energy:-2042.0 ----	 ---- Recall of 2-th state,Energy:-2042.0 ---- Quality:100.0%
 ==== Orginal pattern[1]: ====	 ---- Initial state of pattern[1],Energy:-1532.0 ----	 ---- Recall of 1-th state,Energy:-2640.0 ----	 ---- Recall of 2-th state,Energy:-2640.0 ---- Quality:100.0%

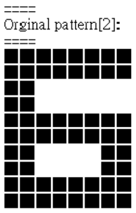
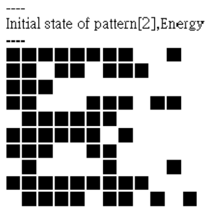
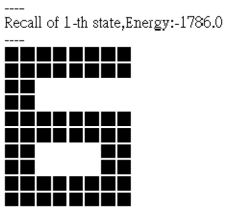
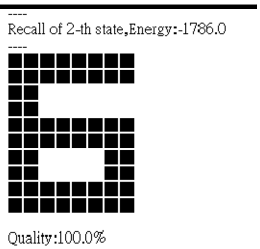
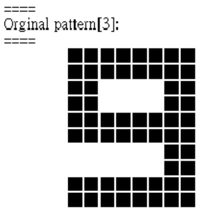
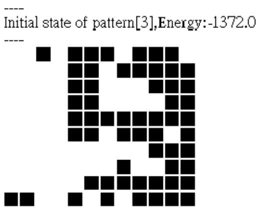
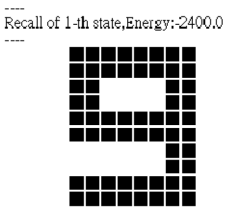
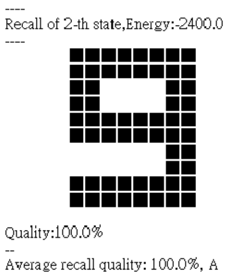
 ==== Orginal pattern[2]: ====	 ---- Initial state of pattern[2],Energy:-600.0 ----	 ---- Recall of 1-th state,Energy:-1786.0 ----	 ---- Recall of 2-th state,Energy:-1786.0 ---- Quality:100.0%
 ==== Orginal pattern[3]: ====	 ---- Initial state of pattern[3],Energy:-1372.0 ----	 ---- Recall of 1-th state,Energy:-2400.0 ----	 ---- Recall of 2-th state,Energy:-2400.0 ---- Quality:100.0% -- Average recall quality: 100.0%, A

Table 5. The result of 0246 with missing rate 10%

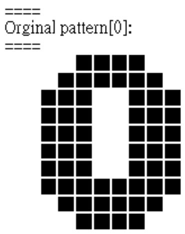
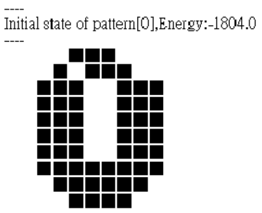
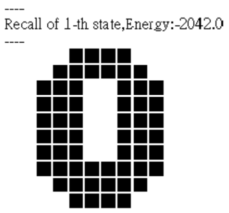
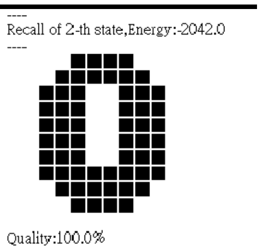
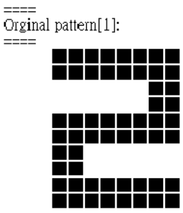
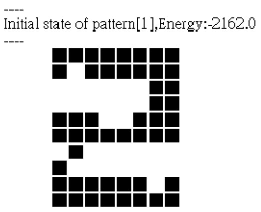
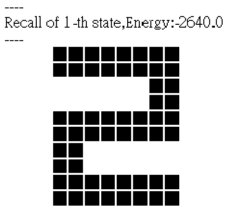
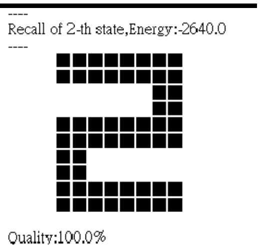
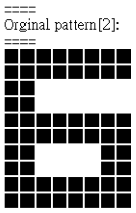
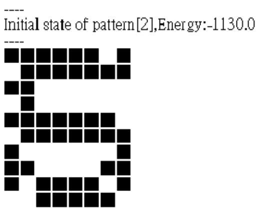
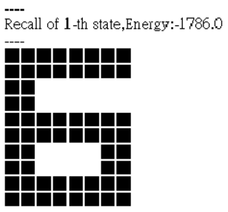
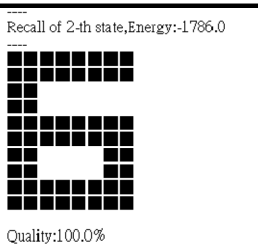
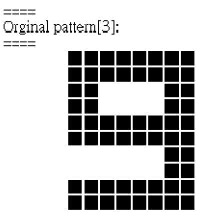
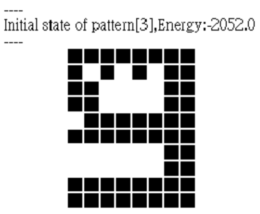
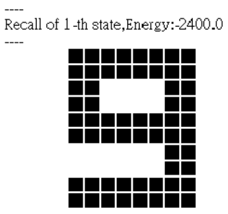
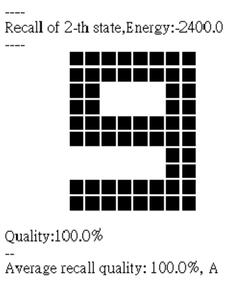
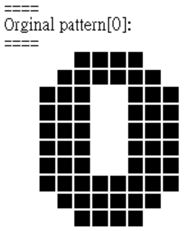
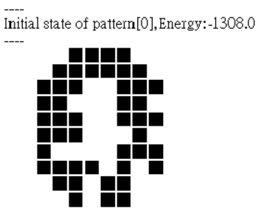
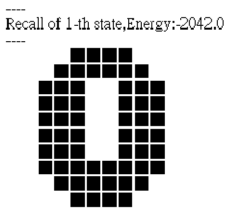
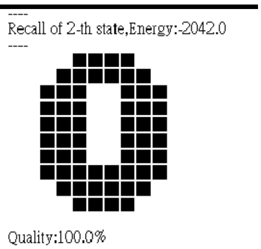
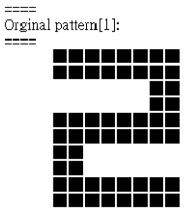
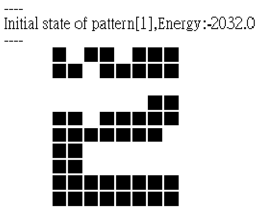
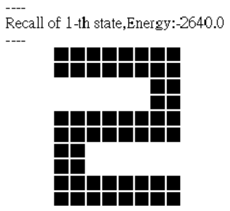
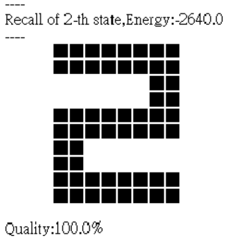
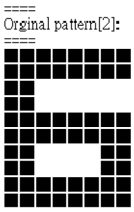
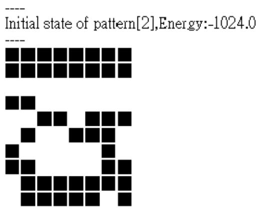
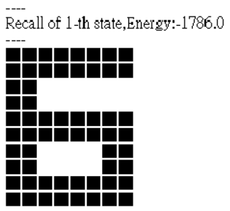
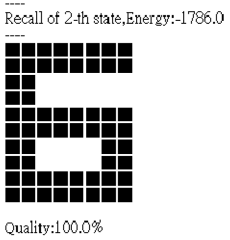
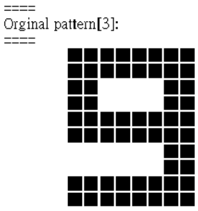
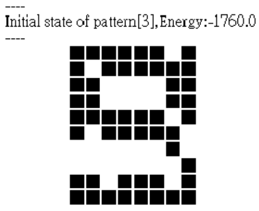
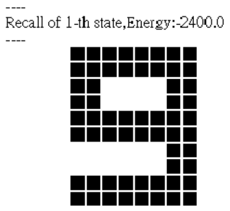
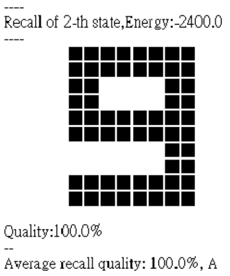
Original	Initial state	Recall 1-th	Recall 2-th
 ==== Orginal pattern[0]: ====	 ---- Initial state of pattern[0],Energy:-1804.0 ----	 ---- Recall of 1-th state,Energy:-2042.0 ----	 ---- Recall of 2-th state,Energy:-2042.0 ---- Quality:100.0%
 ==== Orginal pattern[1]: ====	 ---- Initial state of pattern[1],Energy:-2162.0 ----	 ---- Recall of 1-th state,Energy:-2640.0 ----	 ---- Recall of 2-th state,Energy:-2640.0 ---- Quality:100.0%
 ==== Orginal pattern[2]: ====	 ---- Initial state of pattern[2],Energy:-1130.0 ----	 ---- Recall of 1-th state,Energy:-1786.0 ----	 ---- Recall of 2-th state,Energy:-1786.0 ---- Quality:100.0%
 ==== Orginal pattern[3]: ====	 ---- Initial state of pattern[3],Energy:-2052.0 ----	 ---- Recall of 1-th state,Energy:-2400.0 ----	 ---- Recall of 2-th state,Energy:-2400.0 ---- Quality:100.0% -- Average recall quality: 100.0%, A

Table 6. The result of 0246 with missing rate 15%

Original	Initial state	Recall 1-th	Recall 2-th
 ==== Orginal pattern[0]: ====	 ---- Initial state of pattern[0],Energy:-1308.0 ----	 ---- Recall of 1-th state,Energy:-2042.0 ----	 ---- Recall of 2-th state,Energy:-2042.0 ---- Quality:100.0%
 ==== Orginal pattern[1]: ====	 ---- Initial state of pattern[1],Energy:-2032.0 ----	 ---- Recall of 1-th state,Energy:-2640.0 ----	 ---- Recall of 2-th state,Energy:-2640.0 ---- Quality:100.0%
 ==== Orginal pattern[2]: ====	 ---- Initial state of pattern[2],Energy:-1024.0 ----	 ---- Recall of 1-th state,Energy:-1786.0 ----	 ---- Recall of 2-th state,Energy:-1786.0 ---- Quality:100.0%
 ==== Orginal pattern[3]: ====	 ---- Initial state of pattern[3],Energy:-1760.0 ----	 ---- Recall of 1-th state,Energy:-2400.0 ----	 ---- Recall of 2-th state,Energy:-2400.0 ---- Quality:100.0% -- Average recall quality: 100.0%, A

For the others cases of grade B, C, D, E and F, it is also choosing one set of patterns of each grade. The detail results of each chosen set of patterns of each grade can be found at the directory “/proj-04 /result/4 patterns/b_0358/”, “/proj-04 /result/4 patterns/c_1259/”, “/proj-04 /result/4 patterns/d_1237/”, “/proj-04 /result/4 patterns/e_0357/” and “/proj-04 /result/6 patterns/e_015679/”.

Next, the second part is to show the results of program 2. The results are the all combinations of each kind of set of patterns which is 1 pattern to 9 patterns. The detail results of each kind of set of patterns can be found at the directory “/proj-04 /result/1 pattern/” to “/proj-04 /result/9 patterns/”. The table 7 shows each kind of set of patterns that has how many combinations are. The summarized result of each kind of set of patterns is shown as table 8. The formula of combination is as follows:

$$C_n^m = \frac{m!}{n!(m-n)!}$$

Table 7. The combinations of each kind of set of patterns

Combination	Total
C_1^{10}	10
C_2^{10}	45
C_3^{10}	120
C_4^{10}	210
C_5^{10}	252
C_6^{10}	210
C_7^{10}	120
C_8^{10}	45
C_9^{10}	10

Table 8. The summarized result of all combinations

Combination	Grade	Original	Noisy		Miss	
		0%	10%	15%	10%	15%
C_1^{10}	A	10	10	10	10	10
	B	0	0	0	0	0
	C	0	0	0	0	0
	D	0	0	0	0	0
	E	0	0	0	0	0
	F	0	0	0	0	0
C_2^{10}	A	41	42	40	41	41
	B	2	2	2	2	2
	C	2	1	1	2	2
	D	0	0	2	0	0
	E	0	0	0	0	0
	F	0	0	0	0	0
C_3^{10}	A	65	53	40	58	51
	B	30	34	33	31	33
	C	25	31	42	31	35
	D	0	1	4	0	0
	E	0	1	1	0	1
	F	0	0	0	0	0
C_4^{10}	A	53	32	29	44	32
	B	75	71	59	64	70

	C	55	61	62	62	65
	D	22	38	44	31	32
	E	5	8	16	9	11
	F	0	0	0	0	0
C_5^{10}	A	19	10	9	13	9
	B	71	53	34	54	50
	C	66	67	70	72	73
	D	58	71	79	65	63
	E	38	51	59	48	57
	F	0	0	1	0	0
C_6^{10}	A	0	0	0	0	0
	B	30	10	9	11	10
	C	51	43	33	44	45
	D	67	75	74	75	71
	E	58	78	89	76	79
	F	4	4	5	4	5
C_7^{10}	A	0	0	0	0	0
	B	5	0	0	3	3
	C	11	7	6	6	7
	D	38	29	28	37	29
	E	65	83	85	73	80
	F	1	1	1	1	1
C_8^{10}	A	0	0	0	0	0
	B	0	0	0	0	0
	C	0	1	1	1	0
	D	6	6	4	4	6
	E	36	36	37	37	36
	F	3	2	3	3	3
C_9^{10}	A	0	0	0	0	0
	B	0	0	0	0	0
	C	0	0	0	0	0
	D	0	0	0	0	0
	E	9	9	9	9	9
	F	1	1	1	1	1

In this homework report, it is choosing the set of 4 patterns for analysis as figure 2 to 5. In the analysis, it is to analyze how the noisy and missing data affect the result of recalling pattern and compare the results of noisy and missing data. As figure 2, we can easy understand when the noisy data is increasing; the effect of recalling pattern will get worst. In the case of 10% noisy data, the first two grades, A and B, have more better recalled patterns than the case of 15% noisy data; and the last 4 grades have less bad recalled pattern than the case of 15% noisy data. As figure 3, the case of missing data, the case of 10% missing data is also better than the case of the case of 15% missing data. Because the case of 10% missing data has more good recalled patterns in the grade A and less bad recalled pattern in the last 5 grades.

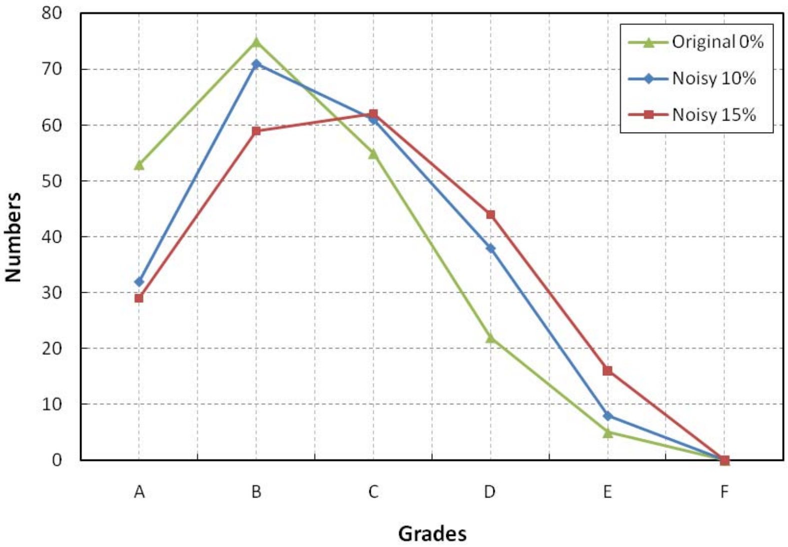


Figure 2. 4 patterns with noisy data

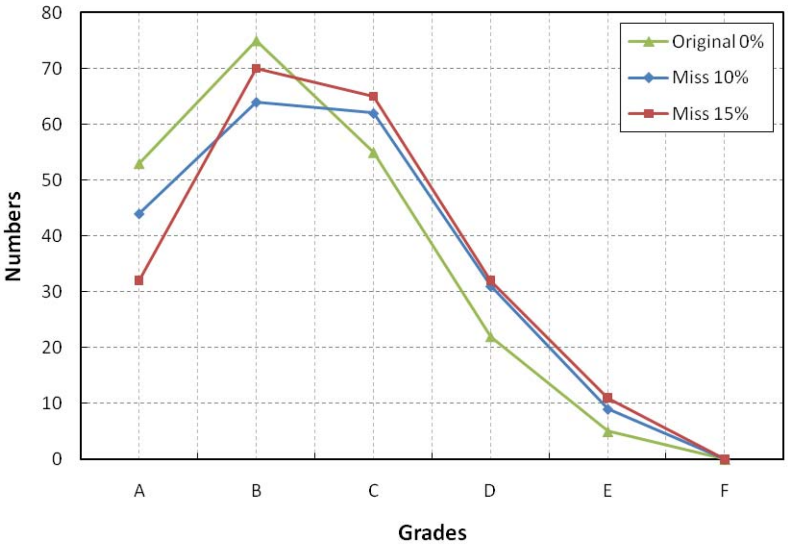


Figure 3. 4 patterns with missing data

As figure 4 and 5, it is comparing the noisy data and missing data with 10% and 15% rates. We can figure out the result, the missing data has good recalling quality in the grade A and less bad recalling quality in the last 4 grades. As figure 5, the result of recalling quality of missing data is also better noisy data.

In the brief conclusion, we can guess why the case of missing data can be better than the case of noisy data. Is that the noisy data affecting the recalling quality of recalled patterns? In the opposite side, the case of missing data has no noisy data to affect the recalling quality. So, the case of missing data can restore the pattern without the affecting by noisy.

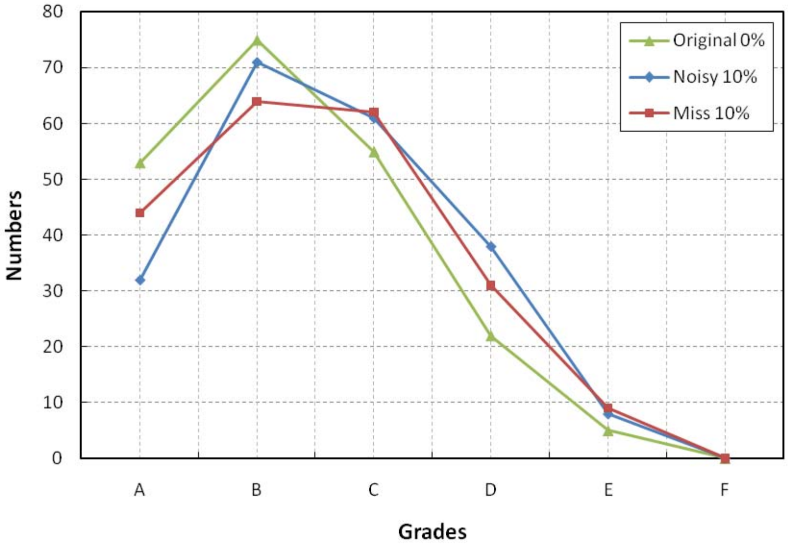


Figure 4. Compare 10% noisy and missing data

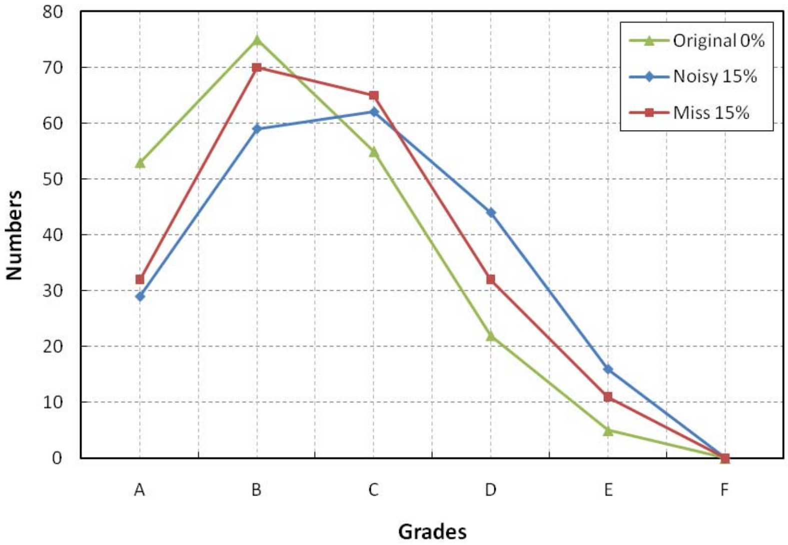


Figure 5. Compare 15% noisy and missing data

7. Improve idea

In the section 7, it is to show the improving idea briefly. Maybe, the idea can really improve the recalling quality of Hopfield neural network. As figure 6, it is using the hierarchical structure trying to improve the recalling quality. Generally, we can say that the Hopfield network is input the pattern in just one level. The improving idea is called Hierarchical Hopfield Neural Network (H^2NN). It is to recall patterns in two or more levels. Firstly, we divide the pattern into 4 sub-patterns. Secondly, do the local recall by using distance function to choose the closest sub-pattern in the store matrix. Thirdly, after local recall phase, it is to combine the sub-patterns into one major pattern and do the global recall. Finally, output the recalled result.

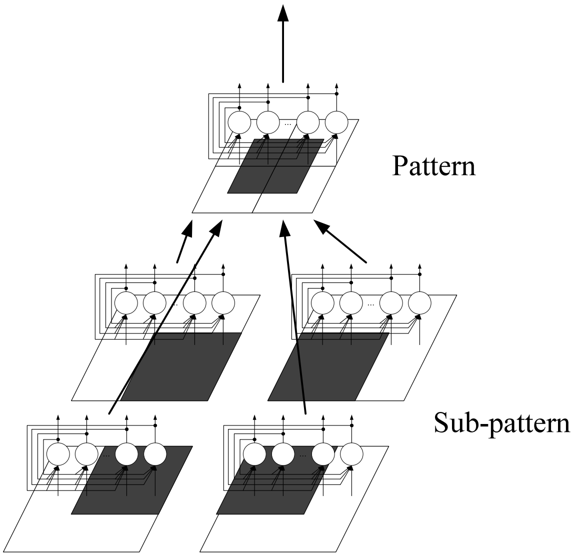


Figure 6. Hierarchical structure of Hopfield network

Why does it use the local recall? The reason is that it is trying to void the attracting by the attractor when we use the while pattern as input. Guess that if we separate the pattern into sub-patterns, can the attracting forces of attractor be reduce? Before separating the pattern, the pattern is attracted by the wrong pattern (attractor). Maybe, in the 4 sub-patterns case, the 2 sub-patterns are attracted by the wrong pattern and the other 2 sub-patterns are attracted by the right pattern.

Reference

[1] A. K. Jain, J. Mao and K.M. Mohiuddin, “Artificial neural networks: a tutorial,”

IEEE Computer, pp.31-44, Mar. 1996.

[2] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *PNAS*, Vol. 79, pp.2554-2558, 1982.

[3] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *PNAS*, Vol. 81, pp.3088-3092, May 1984

[4] J. J. Hopfield, "'Neural' computation of decisions in optimization problems," *Biological Cybernetics*, Vol. 52, pp.141-152 1985

[5] J. J. Hopfield, “Artificial neural networks,” *IEEE Circuits and Devices Magazine* pp.3-10, Sep. 1988.

[6] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE Magasine*, Vol. 3, No4, pp.4-22, Apr. 1987.