Worksheet For Exercises From Chapter 5

Exercise 5.1

1. What is the total SSE for the network after training has finished?

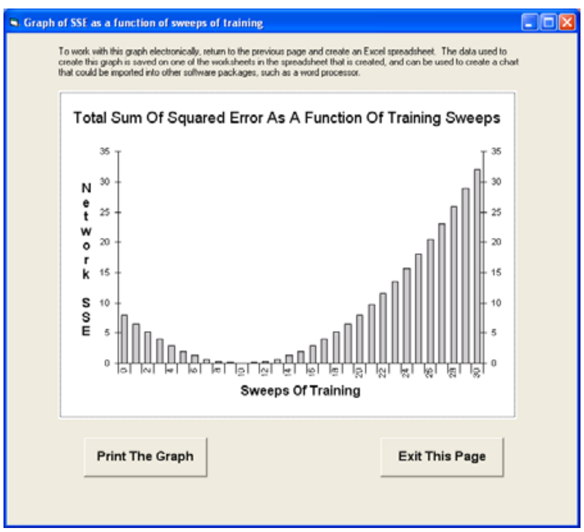
After training was finished, the total SSE was equal to 32.

2. How does this value for SSE compare to the same value that was observed in Exercise 4.1? What can one conclude from this comparison?

In the previous exercise, the total SSE was 0. In other words, by training the network for 20 more epochs, performance went from being perfect to being full of error!

3. Examine how SSE for this network changed over time. Compare and contrast the performance in this simulation to that observed for the same training set in Exercise 4.1. What are the implications of this comparison for Hebb rule learning?

As the graph below shows, over the first 10 trials error decreased as was observed in the previous exercise. However, after reaching a minimum level, there is a dramatic rise in error over the next 20 sweeps of training.



4. Describe the kind of errors that the network made. What is the relationship between these errors and the training procedure?

The first table below provides the desired responses:

Pattern	OUT 1	OUT 2	OUT 3	OUT 4	OUT 5	OUT 6	OUT 7	OUT 8
PAT 1	0.51	-0.14	-0.23	0.77	0.11	0.24	0.07	0.08
PAT 2	-0.52	0.37	-0.29	0.43	-0.01	-0.10	-0.41	-0.38
PAT 3	-0.04	-0.05	-0.83	-0.34	-0.16	0.38	0.01	0.15
PAT 4	0.25	0.45	-0.01	-0.05	-0.37	0.06	0.54	-0.55
PAT 5	-0.33	0.25	0.20	0.01	0.53	0.65	0.30	0.05
PAT 6	0.32	0.62	0.19	-0.08	-0.23	0.24	-0.47	0.38

PAT 7	0.35	-0.22	0.06	-0.28	0.29	0.28	-0.47	-0.61
PAT 8	0.27	0.39	-0.31	-0.16	0.64	-0.48	0.10	0.07

The second table provides the observed responses:

Pattern	ACT 1	ACT 2	ACT 3	ACT 4	ACT 5	ACT 6	ACT 7	ACT 8
PAT 1	1.54	-0.41	-0.69	2.30	0.32	0.73	0.22	0.23
PAT 2	-1.55	1.10	-0.87	1.30	-0.02	-0.29	-1.24	-1.15
PAT 3	-0.13	-0.15	-2.48	-1.02	-0.49	1.14	0.04	0.44
PAT 4	0.75	1.34	-0.02	-0.16	-1.12	0.19	1.61	-1.66
PAT 5	-0.98	0.75	0.61	0.04	1.59	1.94	0.89	0.16
PAT 6	0.95	1.86	0.58	-0.24	-0.69	0.71	-1.41	1.13
PAT 7	1.05	-0.65	0.19	-0.83	0.88	0.83	-1.40	-1.83
PAT 8	0.82	1.17	-0.92	-0.47	1.91	-1.45	0.29	0.21

The errors made by the network appear to indicate an increase in "extremeness" in the responses of the output units. That is, when a small positive response is desired, the network makes a positive response, but it is larger than needed. Similarly, when a small negative response is desired, the network makes a negative response that is more negative than needed.

5. **In some sense, there is a degree of correctness in the network's responses to the stimulus vectors. In what sense are the responses correct, and in what sense are the responses incorrect?**

It would appear that the absolute values of the network responses are wrong – more extreme than is required – as was indicated in the previous question. However, the direction of the responses is correct. If one views the entire set of output unit responses as a vector, it would appear that the response vector is pointing in exactly the same direction as the desired vector, but it is too long. One way to test this is to take corresponding rows of each of the two matrices given above, and correlate them. Even though the absolute values of the corresponding rows are different, the correlations between pairs of rows (e.g. the first row of the desired table, and the first row of the actual table) is equal to 1. So, the network makes the right "directional" response, but makes it to the wrong magnitude.

EXERCISE 5.2

1. **What is the total SSE for the network after training has finished?**

After training, total SSE was very large: 33,465.21

2. **Examine how SSE for this network changed over time. In general, what can be said about the performance of this network on this problem?**

This network starts out reasonably well on this problem (near zero error), but then error grows dramatically larger as training proceeds. The more training that is conducted, the poorer is the network's performance. This is shown in the graph of SSE data below:

3. **Continuing with an examination of total SSE, did this value ever decrease during training? How close did this value approach to 0? What are the implications of these observations?**

SSE did not really seem to decrease with training – indeed, the reverse is true. The implications would appear to be that for this problem, using this learning rule, with more training there is poorer performance.

4. **Describe the kind of errors that the network made. Is the network generating errors to a small number of problems, or are errors for all of the training patterns uniformly large?**

In terms of actual responses, the network responses are usually far more extreme than desired. However, unlike the previous exercise, there is less of an observable regularity in these errors.

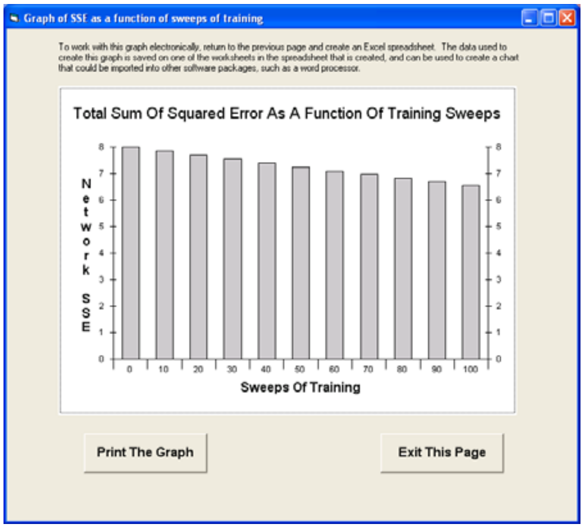
If one correlates each vector of desired outputs with each vector of actual outputs, the following numbers are obtained:

0.359719
0.737921
0.855116
0.602038
0.859827
0.937138
0.886416
0.754933

This shows several things. First, network performance is moderately correct from a "directional" sense, because most of the correlations are very high. However, performance is poorer than the previous exercise – none of the correlations equal 1. Second, not all associations are treated equally – the first pair produces very poor performance, but the 6^{th} pair nearly generates a correlation of 1. Linear independence is definitely causing a problem for this network.

5. **Rerun the network on the independ8.net problem, with a maximum number of sweeps set to 100, training with the Hebb rule, and printing out information every sweep. Play with the learning rate a bit, and examine the SSE curve when the program stops training. Are you able to improve the performance of the network in any significant way? What are the implications of these observations? (To answer this question, you should provide some information about what you settings you used to run the study.)**

If the learning rate is dropped to a value of 0.001, then at the end of training total SSE is a very small value (6.56). Here is the graph of SSE during this learning, which shows that error slowly decreases, but even after 100 sweeps with this small learning rate not much change in error has been observed:



Playing around with a number of different learning rates can change the overall appearance of this graph, but I was unable to generate one in which the entire set of associations was learned. Linearly independent sets of vectors appear to give Hebb learning extreme difficulty when used as the basis for associations!