(Brest State Technical University 2005 Spring Semester: Corse Practice)

Associative Memory by Hopfield Neural Network

Akira Imada
(e-mail akira@bstu.by)

This document is still under construction and was lastly modified on

May 11, 2005

1 Create your own patterns to be stored!

The first task of this practice is to create your own p patterns each of which is made up of N^2 binary pixels. Examples are shown in Figure 1.

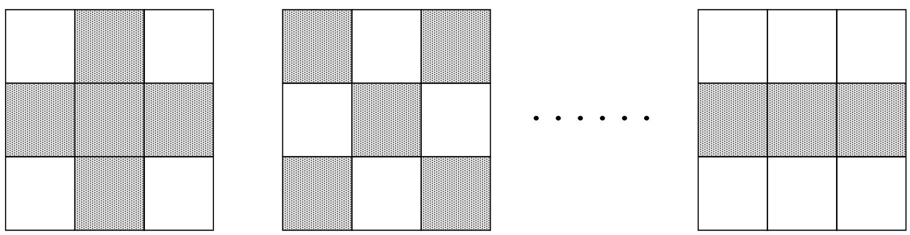

Figure 1: An example for $N = 3$. White pixcel is represented by -1 and white pixel is by $+1$. That is, patterns are represented with 9-bit *bipolar* vectors.

To be more formal, our p patterns are expressed as

$$(\xi_1^1, \xi_2^1, \cdots, \xi_N^1), \quad (\xi_1^2, \xi_2^2, \cdots, \xi_N^2), \quad (\xi_1^3, \xi_2^3, \cdots, \xi_N^3), \quad \cdots, \quad (\xi_1^p, \xi_2^p, \cdots, \xi_N^p). \tag{1}$$

where $\xi_i^\mu \in \{-1, 1\}, \quad \mu = 1, 2, \cdots, p \quad \text{and} \quad i = 1, 2, \cdots, N^2$.

For example, the first example in the Figure 1 is

$$(-1, +1, -1, +1, +1, +1, -1, +1, -1)$$

Excersize 1 (10 Patterns to be Stored) *Create your own 10 patterns each of which is made up of* 10×10 *binary pixels.*

2 Construct a Hopfield Neural Network to store your patterns!

Now Let's create a Hopfield Neural Network with $N \times N$ neurons. All neurons are connected with each other via synapse.

Each of neurons are influenced by other $(N \times N - 1)$ neurons. The strength of an influence from one neuron is according to the weight value of the synapse. Thus, state of neuron i at time $t+1$ $s_i(t+1)$is given as

$$s_i(t+1) = sgn\left(\sum_{j \neq i}^{N} w_{ij} \cdot s_j(t)\right). \tag{2}$$

2.0.1 How we update state of neurons? — Sinchronous and Asynchronous update

Now a question arise, how are neurons updated? Simultaneously-at-a-time or one-by-one? The former is called *Sinchronous update* and the latter is *Asynchronous update.* In this practice course we use the asynchronous update.

In other words, at time t, neuron-1 is firstly updated, and, this new state will be used to update other neurons here after; next, neuron-2, neuron-3, and so on, until all the neurons are updated. Then this updated procedures are repeated at time $t+1$, and on and on. [1]

2.1 An experiment with random weight values:

Now we have 10^2 synapses and each snyapse has a strength called *weight.* The weight value from neuron-j to neuron-i is denoted here as $w_{ij} \quad i, j = 1, 2, \cdots 100$.

We must assign these 100 weight values appropriately to store our patterns. However, we start with a set of 100 random values each of which is in $[-1, 1]$, such as -0.534, are assigned.

Excersize 2 (Observe the behaviours of your Hopfield Network.) *Give one of your patterns to your Hopfield Network with random weight values, and observe what will happen. That is, set* $S_i(0) = \xi_i^\mu$ *for all* $i = 1, 2, \cdots, N^2$ *with a specific* μ *you like. Then observe the pattern for* $t = 1, 2, \cdots$*, namely, how the pattern changes as time proceeds.*

3 Let's store your patterns!

3.1 Hebbian Learning

Here we calculate the weight according to the following equation, called Hebbian Learning.

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \ \ (i \neq j), \ \ w_{ii} = 0. \tag{3}$$

Excersize 3 (Observe the behaviours of your Hopfield Network Part II.) *Repeat the previous excersize with the weights calculated with the above equation, instead of random values.*

4 Let's observe how it is clever (or fool)!

Now if your work is successful Network recognize your patterns in the following way.

(1) If you give one of your patterne as it is, then the initial state never be changed. [2]

(2) If you give a noisy version of your pattern, then initial state changes several steps and converges to the original pattern. [3]

(3) If you give a very different pattern from any of your patters, then network shows a *chaotic* behaviour, e.g., repeat a same cycle of patterns or changes forever in a random way.

Then you may test your Hopfield Network to know how it is clever or fool.by changing (i) the number of patterns stored and/or (2) number of noisy bit.

Excersize 4 (Observe the behaviours of your Hopfield Network Part III.) *Starting with 1 pattern, test the capability of the Network by incrementing the number of noisy bits from zero, and observe when the Network become to be chaotic. Plot this critical number as a function of number of noise with a parameter of the number of patterns stored.*

---

[1] We can update neurons in a random order different order at every time t, but here we update from the 1st neuron to the last neuron one by one.

[2] In this case the pattern is called *fixed point.*

[3] We call it "An initial state in the vasin of attraction is attracted to its fixed point.