8 Rate-based Recurrent Networks: Basis for Associative Memory

These notes are abstracted directly from the presentation in chapter 2 of the book by Hertz, Krogh and Palmer (Introduction to the Theory of Neural Computation, Addison Wesley, 1991)

The basic problem is this:

> Store a set of p patterns ξ_i^u in such a way that when presented with a new pattern ς_i, the network responds by producing whichever one of the stored patterns most closely resembles ς_i.

The patterns are labelled by μ = 1, 2, ... , p, while the units in the network are labelled by i = 1, 2, ... , N. Both the stored patterns ς_i^μ and the test patterns ς_i can be taken to be either 0 or 1 on each site i, though we will adopt a different convention shortly.

For mathematical convenience we use a formulation where the activation values of the units are +1 (firing) and -1 (not firing), as opposed to 1 and 0. We denote them by S_i. The dynamics of the network are:

$$S_i \equiv sgn\left(\sum_{j=1}^{N} W_{ij}S_j - \theta_i\right) \tag{8.1}$$

where we take the sign function sign $sgn(x)$ to be

$$sgn(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

where

$$\mu_i \equiv \sum_{j=1}^{N} W_{ij}S_j - \theta; \tag{8.2}$$

is the input to the neuron. In the rest of this chapter we drop the threshold terms, taking $\theta_i = 0$ as befits the case of random patterns. Thus we have

$$S_i \equiv \left(\sum_{j=1}^{N} W_{ij}S_j\right). \tag{8.3}$$

There are at least two ways in which we might carry out the updating specified by the above equation. We could do it *synchronously*, updating all units simultaneously

at each time step. Or we could do it *asynchronously*, updating them one at a time. Both kinds of models are interesting, but the asynchronous choice is more natural for both brains and artificial networks. The synchronous choice requires a central clock or pacemaker, and is potentially sensitive to timing errors. In the asynchronous case, which we adopt henceforth, we can proceed in either of two ways:

- At each time step, select at random a unit i to be updated, and apply the update rule.
- Let each unit independently choose to update itself according to the update rule, with some constant probability per unit time.

These choices are equivalent, except for the distribution of update intervals, because the second gives a random sequence; there is vanishing small probability of two units choosing to update at exactly the same moment.

Rather than study a specific problem such as memorizing a particular set of pictures, we examine the more generic problem of a *random* set of patterns drawn form a distribution. For convenience we will usually take the patterns to be made up of independent bits ξ_i that can each take on the values +1 and -1 with equal probability.

Our procedure for testing whether a proposed form of w_{ij} is acceptable is first to see whether the patterns to be memorized are themselves stable, and then to check whether small deviations from these patterns are corrected as the network evolves.

One Pattern

To motivate our choice for the connection weights, we consider first the simple case whether there is just one pattern ξ_i that we want to memorize. The condition for this pattern to be stable is just

$$sgn\left(\sum_{j=1}^{N} W_{ij}\xi_j\right) = \xi_i \qquad (for\ all\ i) \tag{8.4}$$

because then the update rule produces no changes. It is easy to see that this is true if we take

$$W_{ij} \propto \xi_i\xi_j \tag{8.5}$$

since $\xi_j^2 = 1$. We take the constant of proportionality to be $1/N$, where N is the number of units in the network, giving

$$W_{ij} = \frac{1}{N}\xi_i\xi_j \quad . \tag{8.6}$$

Furthermore, it is also obvious that even if a number (fewer than half) of the bits of the starting pattern S_i are wrong (i.e., not equal to ξ_i), they will be overwhelmed in the sum for the net input

$$h_i = \sum_{j=1}^{N} W_{ij} S_j \tag{8.7}$$

by the majority that are right, and sgn(h_i) will still give ξ_i. An initial configuration near to ξ_i will therefore quickly relax to ξ_i. This means that the network will correct errors as desired, and we can say that the pattern ξ_i is an **attractor**.

Actually there are two attractors in this simple case; the other one is at $-\xi_i$. This is called a **reversed state**. All starting configurations with *more* than half the bits different from the original pattern will end up in the reversed state.

Many Patterns

This is fine for one pattern, but how do we get the system to recall the most similar of many patterns? The simplest answer is just to make w_{ij} by an outer product rule, which corresponds to

$$W_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \quad . \tag{8.8}$$

Here p is the total number of stored patters labelled by μ.

This is called the "Hebb rule" because of the similarity with a hypothesis made by Hebb (1949) about the way in which synaptic strengths in the brain change in response to experience: Hebb suggested changes proportional to the correlation between the firing of the pre- and post-synaptic neurons.

Let us examine the stability of a particular pattern ξ_i^ν. The stability condition generalizes to

$$sgn(h_i^\nu) = \xi_i^\nu \qquad\qquad (for\ all\ i) \tag{8.9}$$

where the net input h_i^ν to unit i in pattern ν is

$$h_i^\nu \equiv \sum_{j=1}^{N} W_{ij} \xi_j^\nu = \frac{1}{N} \sum_{j=1}^{N} \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \xi_j^\nu \quad . \tag{8.10}$$

We now separate the sum on μ into the special term $\mu = \nu$ and all the rest:

$$h_i^\nu = \xi_i^\nu + \frac{1}{N} \sum_{j=1}^{N} \sum_{\mu \neq \nu}^{p} \xi_i^\mu \xi_j^\mu \xi_j^\nu \quad . \tag{8.11}$$

If the second term were zero, we could immediately conclude that pattern number ν was stable according to the stability condition. This is still true if the second term is small enough: *if its magnitude is smaller than 1 it cannot change the sign of* h_i^ν.

It turns out that the second term *is* less than 1 in many cases of interest if p, the number of patterns, is small enough. Then the stored patterns are all stable – if we start the system from one of them it will stay there. Furthermore, a small fraction of bits different from a stored pattern will be corrected in the same way as

in the single-pattern case; they are overwhelmed in the sum $\sum_j W_{ij}S_j$ by the vast majority of correct bits. A configuration near to ξ_i^ν thus relaxes to ξ_i^ν. This shows that the chosen patterns are truly attractors of the system . The system works as a content-addressable memory.

The Energy Function

One of the most important contributions of Hopfield was to introduce the idea of an *energy function* into neural network theory. For the networks we are considering, the energy function E is

$$E = -\frac{1}{2}\sum_{ij}^{N} W_{ij}S_iS_j \quad . \tag{8.12}$$

The double sum is over all i and all j. The $i = j$ terms are of no consequence because $S_i^2 = 1$; they just contribute a constant to E, and in any case we could choose $W_{ii} = 0$. The energy function is a function of the configuration S_i of the system, where S_i means the set of all the S_i's. Typically this surface is quite hilly.

The central property of an energy function is that it *always decreases (or remains constant) as the system evolves according to its dynamical rule.* Thus the attractors (memorized patterns) are at local minima of the energy surface.

For neural networks in general an energy function exists if the connection strengths are *symmetric*, i.e., $W_{ij} = W_{ji}$. In real networks of neurons this is an unreasonable assumption, but it is useful to study the symmetric case because of the extra insight that the existence of an energy function affords us. The Hebb prescription that we are now studying automatically yields symmetric W_{ij}'s.

For symmetric connections we can write the energy in the alternative form

$$E = -\sum_{(ij)}^{N} w_{ij}S_iS_j + \text{constant} \tag{8.13}$$

where (ij) means all the distinct pairs of ij, counting for example 12 as the same pair as 21. We exclude the ii terms from (ij); they give the constant.

It now is easy to show that the dynamical rule can only decrease the energy. Let S_i' be the new value of S_i for some particular unit i:

$$S_i' = sgn\left(\sum_{j=1}^{N} W_{ij}S_j\right) \quad . \tag{8.14}$$

Obviously if $S_i' = S_i$ the energy is unchanged. In the other case $S_i' = -S_i$ so, picking out the terms that involve S_i

$$\begin{aligned} E' - E \quad &= -\sum_{j\neq i}^{N} W_{ij}S_i'S_j + \sum_{j\neq i}^{N} W_{ij}S_iS_j \\ &= 2S_i\sum_{j\neq i}^{N} W_{ij}S_j \end{aligned} \tag{8.15}$$

$$= 2S_i \sum_{j=1}^{N} W_{ij} S_j - 2W_{ii}.$$

Now the first term is negative from the update rule, and the second term is negative because the Hebb rule gives $W_{ii} = p/N$ for all i. Thus the energy decreases every time an S_i changes, as claimed.

The self-coupling terms W_{ii} may actually be omitted altogether, both from the Hebb rule (where we can simply define $W_{ii} = 0$) and from the energy function as they make no appreciable difference to the stability of the ξ_i^ν patterns in the large N limit.

Starting from an Energy Function

The idea of the energy function as something to be minimized in the stable states gives us an alternate way to derive the Hebb prescription. Let us start again with the single-pattern case. We want the energy to be minimized when the overlap between the network configuration and the stored pattern ξ_i is largest. So we choose

$$E = -\frac{1}{2N} \sum_{\mu=1}^{p} \left(\sum_{i=1}^{N} S_i \xi_i^\mu \right)^2 \quad . \tag{8.16}$$

Multiplying this out gives

$$\begin{aligned} E \quad &= -\frac{1}{2N} \sum_{\mu=1}^{p} \left(\sum_{i=1}^{N} S_i \xi_i^\mu \right) \left(\sum_{j=1}^{N} S_j \xi_j^\mu \right) \\ &= -\frac{1}{2} \sum_{ij}^{N} \left(\frac{1}{N} \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \right) S_i S_j \end{aligned} \tag{8.17}$$

which is exactly the same as our original energy function if w_{ij} is given by the Hebb rule.

This approach to finding appropriate W_{ij}'s is generally useful. If we can write down an energy function whose minimum satisfies a problem of interest, then we can multiply it out and identify the appropriate strength W_{ij} from the coefficient of $S_i S_j$.

Spurious States (Amit, Gottfried and Sompolinsky 1985)

We have shown that the Hebb prescription gives us (for small enough p) a dynamical system that has attractors – local minima of the energy function – at the desired points ξ_i^μ. These are sometimes called the **retrieval states**. But we have not shown that these are the only attractors. And indeed there are others.

First of all, the reversed states $-\xi_i^\mu$ are minima and have the same energy as the original patterns. The dynamics and the energy function both have a perfect symmetry, $S_i \leftrightarrow -S_i$ for all i. This is not too troublesome for the retrieved patterns;

we could agree to reverse all the remaining bits when a particular "sign bit" is –1 for example.

Second, there are stable **mixture states** ξ_i^{mix}, which are not equal to any single pattern, but instead correspond to linear combinations of an odd number of patterns. The simplest of these are symmetric combinations of three stored patterns:

$$\xi_i^{mix} = sgn(\pm\xi_i^{\mu 1} \pm \xi_i^{\mu 2} \pm \xi_i^{\mu 3}) \quad . \tag{8.18}$$

All $2^3 = 8$ sign combinations are possible, but we consider for definiteness the case where all the signs are chosen as +'s. The other cases are similar. Observe that on average ξ_i^{mix} has the same sign at $\xi_i^{\mu 1}$ three times out of four; only if $\xi_i^{\mu 2}$ and $\xi_i^{\mu 3}$ both have the opposite sign can the overall sign be reversed? So ξ_i^{mix} is Hamming distance $N/4$ from $\xi_i^{\mu 1}$, and of course from $\xi_i^{\mu 2}$ and $\xi_i^{\mu 3}$ too; the mixture states lie at points equidistant from their components. This also implies that $\sum_i \xi_i^{\mu 1}\xi_i^{mix} = N/2$ on average. To check the stability pick out the three special μ's; still with all + signs, to find:

$$h_i^{mix} = \frac{1}{N}\sum_{j=1}^{N}\sum_{\mu}\xi_i^{\mu}\xi_j^{\mu}\xi_j^{mix} = \frac{1}{2}\xi_i^{\mu 1} + \frac{1}{2}\xi_i^{\mu 2} + \frac{1}{2}\xi_i^{\mu 3} + cross - terms \quad . \tag{8.19}$$

Thus the stability condition is satisfied for the mixture state. Similarly 5, 7, ... patterns may be combined. The system does not choose an *even* number of patterns because they can add up to zero on some sites, whereas the units have to be ± 1.

Third, for large p there are local minima that are not correlated with any finite number of the original patters ξ_i^{μ}.

Phase Diagram of the Hopfield Model (Amit, Gottfried and Sompolinsky 1985)

A statistical mechanical analysis shows that there is a crucial value α_c of $\alpha \equiv P/N$ where memory states no longer exist. A numerical evaluation gives

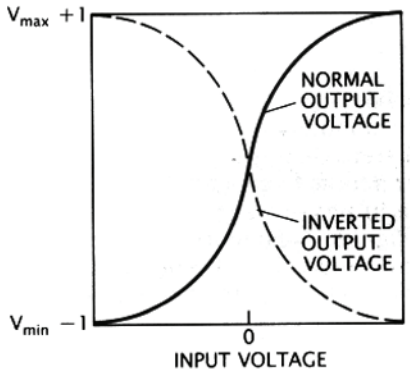
$$\alpha_c \approx 0.138 \quad . \tag{8.20}$$

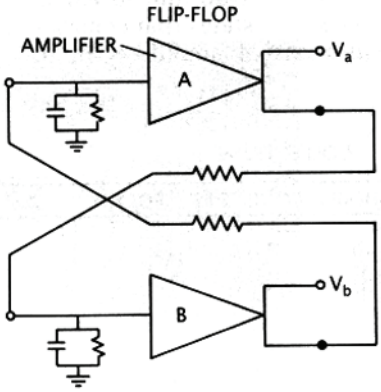
The jump in the number of memory states is considerable: from near-perfect recall to zero. This tells us that with no internal noise we go discontinuously from a very good working memory with only a few bits in error for $\alpha < \alpha_c$ to a useless one for a $\alpha > \alpha_c$.

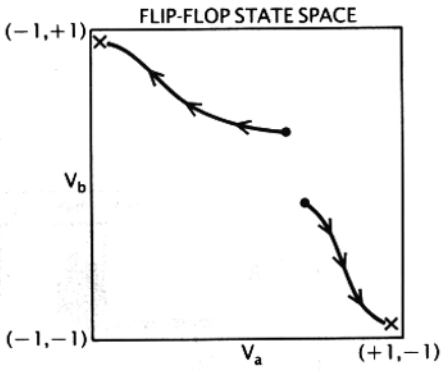
The attached figure shows the whole **phase diagram** for the Hopfield model, delineating different regimes of behavior in the $T - \alpha$ plane, where T is the variance of the random line. There is a roughly triangular region where the network is a good memory device, as indicated by regions A and a' of the figure. The result corresponds to the upper limit on the α axis, while the critical noise level $T_c = 1$ for the $p \ll N$ case sets the limit on the T axis. Between these limits there is a critical noise level $T_c(\alpha)$, or equivalently a critical load $\alpha_c(T)$, as shown. As $T \to 1, \alpha_c(T)$ goes to zero like $(1 - T)^2$.

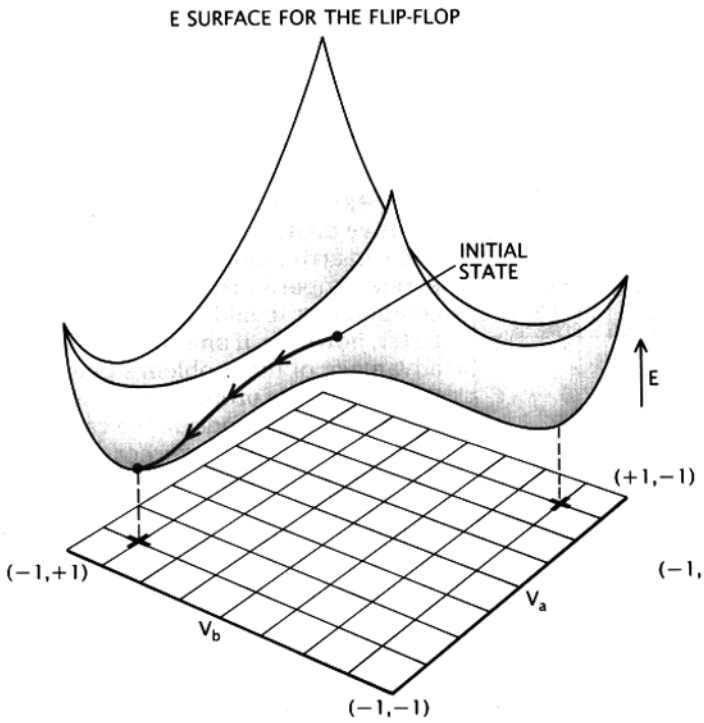
In region C the network still turns out to have many stable states, called **spin glass states**, but these are not correlated with any of the patterns ξ_i^μ. However, if T is raised to a sufficiently high value, into region D, the output of the network continuously fluctuates with $\langle S_i \rangle = 0$.

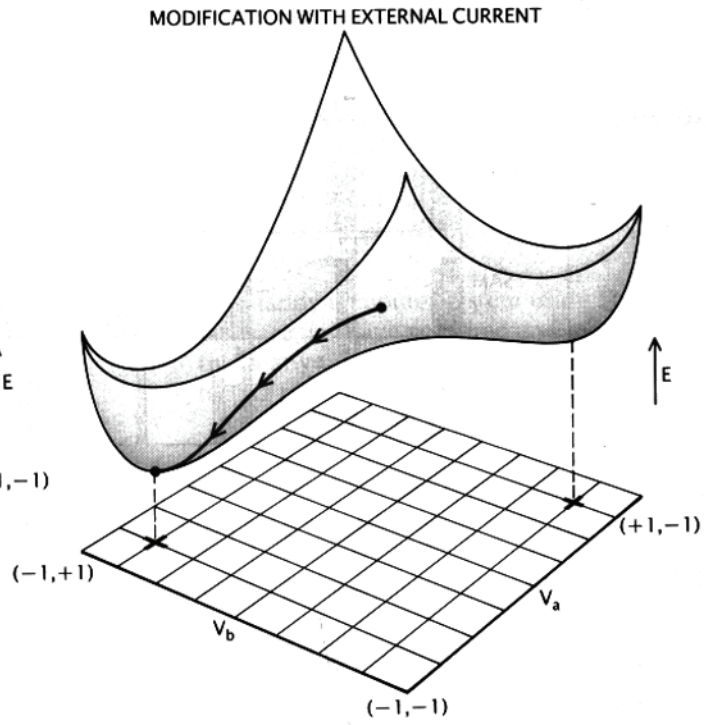
Regions A, A', and B both have the desired retrieval states, beside some percentage of wrong bits, but also have spin glass states. The spin states are the most stable states in region B, lower in energy than the desired states, whereas in region A the desired states are the global minima. For small enough α and T there are also mixture states that are correlated with an odd number of the patterns as discussed earlier. These always have higher free energy than the desired states. Each type of mixture state is stable in a triangular region (A, A' and B), but with smaller intercepts on both axes. The most stable mixture states extend to 0.46 on the T axis and 0.03 on the α axis (region A').

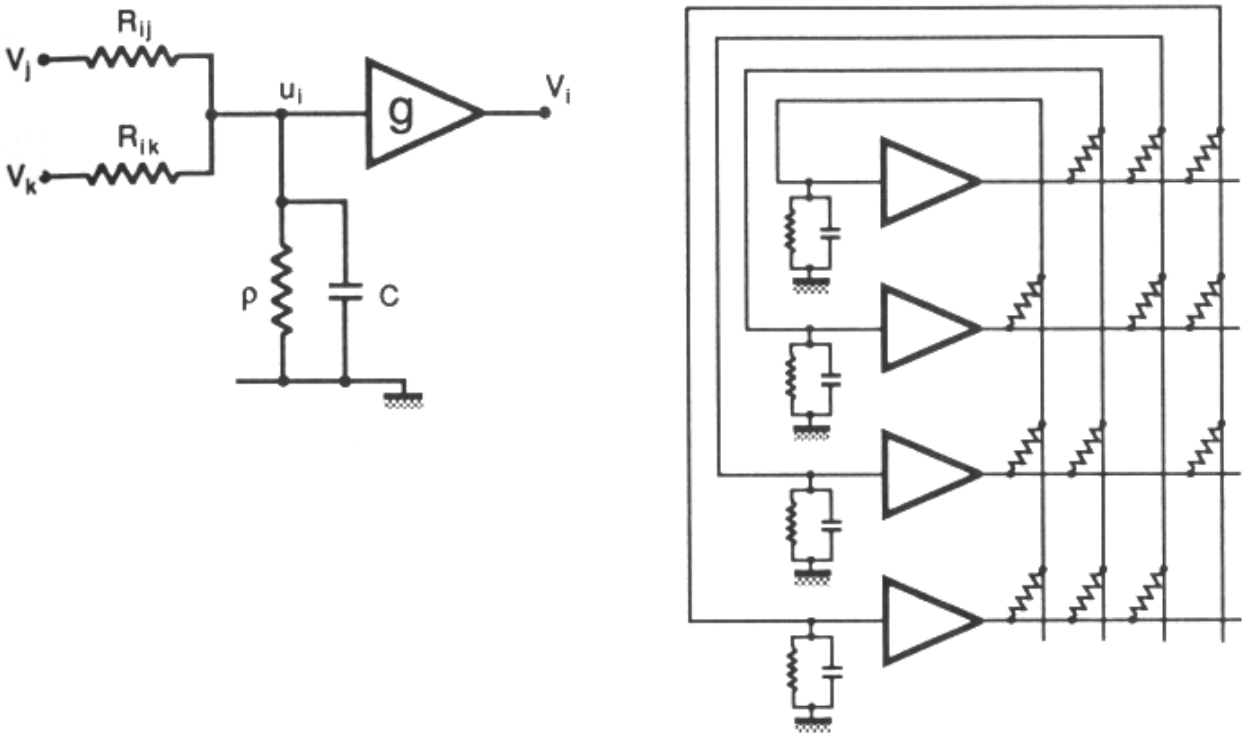

V_{max} +1
V_{min} −1
0
INPUT VOLTAGE
NORMAL OUTPUT VOLTAGE
INVERTED OUTPUT VOLTAGE


FLIP-FLOP
AMPLIFIER
A
B
V_a
V_b


FLIP-FLOP STATE SPACE
(−1,+1)
(−1,−1)
(+1,−1)
V_b
V_a


E SURFACE FOR THE FLIP-FLOP
INITIAL STATE
E
(+1,−1)
(−1,+1)
V_a
V_b
(−1,−1)


MODIFICATION WITH EXTERNAL CURRENT
E
(+1,−1)
(−1,+1)
V_a
V_b
(−1,−1)



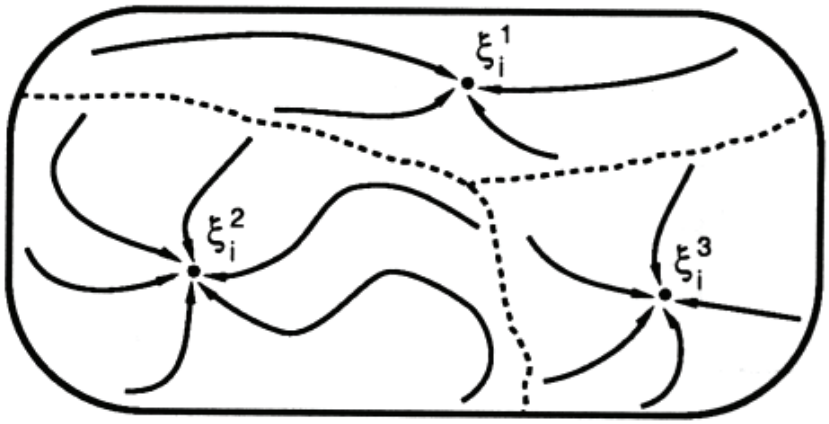
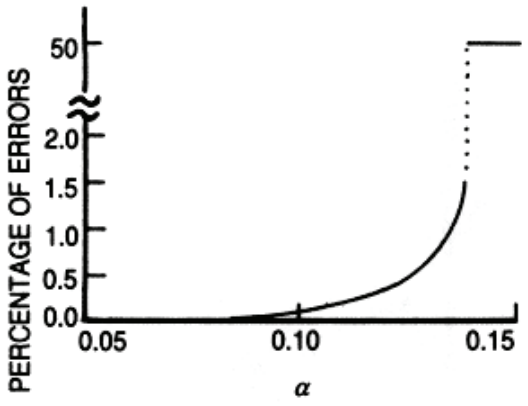


FIGURE 2.2 Schematic configuration space of a model with three attractors.
Herz_Fig.2.2



Errors per neuron increase discontinuously as $T \to 0$ in the Hopfield model, signaling a complete loss of memory, when the parameter $\alpha = p/N$ exceeds the critical value 0.14. Here p is the number of random memories stored in a Hopfield network of N neurons.

Sompolinsky(1988)_Fig.2