7 Rate-Based Recurrent Networks of Threshold Neurons: Basis for Associative Memory

7.1 A recurrent network with threshold elements

The basic challenge in associative networks is this:

> Store a set of p patterns $\vec{\xi}^{\mu}$ in such a way that when presented with a new pattern $\vec{\mathbf{S}}^{test}$, the network responds by producing whichever one of the stored patterns most closely resembles $\vec{\mathbf{S}}^{test}$. Close is defined in terms if the Hamming distance, the number of different "bits" in the pattern.

The patterns are labelled by $\mu = 1,\ 2,\ \dots\ ,\ p$, while the neurons or units in the network are labelled by i = 1, 2, ... , N. We donate the activity of the $i-th$ neuron by S_i. The dynamics of the network are:

$$S_i \equiv sgn\left(\sum_{j=1}^{N} W_{ij} S_j - \theta_i\right) \tag{7.7}$$

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

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

where

$$h_i \equiv \sum_{j=1}^{N} W_{ij} S_j - \theta; \tag{7.8}$$

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

$$S_i \equiv sgn\left(\sum_{j=1}^{N} W_{ij} S_j\right). \tag{7.9}$$

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 from 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.

7.2 Storing 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

$$sgn\left(\sum_{j=1}^{N} W_{ij}\xi_j\right) = \xi_i \qquad \forall i \tag{7.10}$$

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

$$W_{ij} \propto \xi_i\xi_j \tag{7.11}$$

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

$$W_{ij} = \frac{1}{N}\xi_i\xi_j \quad . \tag{7.12}$$

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

$$h_i = \sum_{j=1}^{N} W_{ij}S_j \tag{7.13}$$

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.

7.3 Storing 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

$$W_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \quad . \tag{7.14}$$

Here p is the total number of stored patterns 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

$$sgn(h_i^\nu) = \xi_i^\nu \qquad \forall i \tag{7.15}$$

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

$$h_i^\nu \equiv \sum_{j=1}^{N} W_{ij} \xi_j^\nu = \frac{1}{N} \sum_{j=1}^{N} \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \xi_j^\nu \quad . \tag{7.16}$$

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

$$h_i^\nu = \xi_i^\nu + \frac{1}{N} \sum_{j=1}^{N} \sum_{\mu \neq \nu}^{p} \xi_i^\mu \xi_j^\mu \xi_j^\nu \quad . \tag{7.17}$$

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.

7.4 Scaling for error-free storage of many patterns

We consider a Hopfield network with the standard Hebb-like learning rule and ask how many memories we can imbed in a network of N neurons with the constraint that we will accept at most one bit (one neuron's output in only one memory state) of error. The input h_i is

$$\begin{aligned} h_i &= \sum_{j \neq i}^{N} W_{ij} S_j \\ &= \frac{1}{N} \sum_{\mu=1}^{p} \sum_{j \neq i}^{N} \xi_i^{\mu} \xi_j^{\mu} S_j \end{aligned} \tag{7.18}$$

where p is the number of stored memories, N is the number of neurons and

$$W_{ij} \equiv \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^{\mu} \xi_j^{\mu} \tag{7.19}$$

is the synaptic weight matrix given by the Hebb rule.

Now, check the stability of a stored state. Make $S_j = \xi_j^1$, one of the stored memory states, so that

$$\begin{aligned} h_i &= \frac{1}{N} \sum_{\mu=1}^{p} \sum_{j \neq i}^{N} \xi_i^{\mu} \xi_j^{\mu} \xi_j^1 \\ &= \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^{\mu} \sum_{j \neq i}^{N} \xi_j^{\mu} \xi_j^1 \\ &= \frac{1}{N}\ \xi_i^1 \sum_{j \neq i}^{N} \xi_j^1 \xi_j^1 + \frac{1}{N} \sum_{\mu \neq 1}^{p} \xi_i^{\mu} \sum_{j \neq i}^{N} \xi_j^{\mu} \xi_j^1 \end{aligned} \tag{7.20}$$

On average, the second term is zero, so that the average input is

$$< h_i >= \frac{1}{N}\ \xi_i^1 (N-1)\ \simeq\ \xi_i^1 \tag{7.21}$$

What is the variance, denoted σ^2? The second term, summed over random vectors with zero mean, consists of the sum of $(p-1)$ inner products of vectors with $(N-1)$ terms. Each term is +1 or -1, *i..e.*, binomially distributed, so that the fluctuation to the input is

$$\begin{aligned} \sigma &= \frac{1}{N} \cdot \sqrt{p-1} \cdot \sqrt{N-1} \\ &\simeq \sqrt{\frac{p}{N}}. \end{aligned} \tag{7.22}$$

This results in a fluctuation to the input with a standard deviation, σ. Noise hurts only if the magnitude of the noise term exceeds 1. The noise becomes Gaussian for large p and N, but constant p/N, which is the limit of interest, Thus the probability of an error in the recall of all stored states is

$$\begin{aligned} P_{error} \quad &= \frac{1}{\sqrt{2\pi}\ \sigma}\ \left[\ \int -_{\infty}^{-1}\ e^{-x^2/2\sigma^2}\ dx\ +\int_{+1}^{\infty}\ e^{-x^2/2\sigma^2}\ dx\ \right] \qquad (7.23) \\ &= \frac{\sqrt{2}}{\sqrt{\pi}\ \sigma}\ \int_{+1}^{\infty}\ e^{-x^2/2\sigma^2}\ dx \\ &= \frac{2}{\sqrt{\pi}}\ \int_{\sqrt{\frac{N}{2p}}}^{\infty}\ e^{-x^2}\ dx \\ &\equiv \operatorname{erfc}\left(\sqrt{\frac{N}{2p}}\right) \end{aligned}$$

where efrc(x) is the complementary error function and we again note that the average of the error term is zero. Note that for $\frac{N}{2p} >> 1$ the complementary error function may be approximated by an asymptotic form given by

$$P_{error} \simeq \frac{2}{\sqrt{\pi}} \frac{p}{N} e^{-N/2p} \qquad (7.24)$$

We have a nice and closed expression in a relevant limit!

Now $N \cdot p$ is total number of "bits" in the network. Suppose only less than one bit can be in error. Then we equate probabilities of correct to within a factor of one bit, or $\frac{1}{Np}$. Thus

$$(1 - P_{error})^{Np} \geq 1 - \frac{1}{Np} \qquad (7.25)$$

But Np is large and P_{error} will be small by construction, so $1 - Np \times P_{error} \geq 1 - \frac{1}{Np}$ and thus

$$P_{error} < \frac{1}{(Np)^2} \qquad (7.26)$$

From the above expansion of the gaussian error:

$$log\left[P_{error}\right] \simeq -\frac{1}{2}\ log\ \pi - \frac{N}{2p} - log\ \frac{N}{2p} \qquad (7.27)$$

From the constraint on the desired error:

$$log\,[P_{error}] < -2\ log(Np) \tag{7.28}$$

Thus

$$-\frac{1}{2}\ log\ \pi - \frac{N}{2p} - log\ \frac{N}{2p}\ < -2\ log\ (Np) \tag{7.29}$$

We now let $N \to \infty$ with N/p constant. Keeping zero-th and first order terms, we have:

$$\frac{N}{2p} > 2\ log\ (Np) > 2\ log\ (N) \tag{7.30}$$

so

$$p\ <\ \frac{1}{4}\ \frac{N}{log\ N} \tag{7.31}$$

Note that p has a similar scaling for the choice of a fixed, nonzero error rate.

Thus we see that an associate memory based on a recurrent Hopfield network stores a number of memories that scales more weakly than the number of neurons if one cannot tolerate any errors upon recall. Keep a mind that a linear network stores only one stable state, e.g., an integrator state . So things are looking good.

7.5 Energy description and convergence

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

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

$$E = -\frac{1}{2}\sum_{ij}^{N} W_{ij} S_i S_j \quad . \tag{7.32}$$

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

$$E = -\sum_{(ij)}^{N} W_{ij} S_i S_j + \text{constant} \tag{7.33}$$

where (ij) means all the distinct pairs of ij, counting for example "1,2" as the same pair as "2,1". 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:

$$S_i' = sgn\left(\sum_{j=1}^{N} W_{ij} S_j\right) \quad . \tag{7.34}$$

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

$$\begin{aligned} E' - E \qquad &= -\sum_{j \neq i}^{N} W_{ij} S_i' S_j + \sum_{j \neq i}^{N} W_{ij} S_i S_j \\ &= 2S_i \sum_{j \neq i}^{N} W_{ij} S_j \\ &= 2S_i \sum_{j=1}^{N} W_{ij} S_j - 2W_{ii}. \end{aligned} \tag{7.35}$$

Now the first term is negative from the update rule, and the second term is negative because the Hebb rule gives $W_{ii} = p/N \; \forall \; i$. 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.

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

$$E = -\frac{1}{2N} \sum_{\mu=1}^{p} \left(\sum_{i=1}^{N} S_i \xi_i^\mu\right)^2 \quad . \tag{7.36}$$

Multiplying this out gives

$$
\begin{aligned}
E \quad &= -\frac{1}{2N}\sum_{\mu=1}^{p}\left(\sum_{i=1}^{N} S_i\xi_i^\mu\right)\left(\sum_{j=1}^{N} S_j\xi_j^\mu\right) \qquad (7.37)\\
&= -\frac{1}{2}\sum_{ij}^{N}\left(\frac{1}{N}\sum_{\mu=1}^{p}\xi_i^\mu\xi_j^\mu\right)S_iS_j
\end{aligned}
$$

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_iS_j.

7.6 The issue of spurious attractors

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

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, as discovered by by Amit, Gottfried and Sompolinsky (1985).

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, $S_i \leftrightarrow - S_i \; \forall \; i$. 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:

$$\xi_i^{mix} = sgn(\pm\xi_i^1 \pm \xi_i^2 \pm \xi_i^3) \quad . \qquad (7.38)$$

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 ξ_i^1 three times out of four; only if ξ_i^2 and ξ_i^3 both have the opposite sign can the overall sign be reversed? So ξ_i^{mix} is Hamming distance $N/4$ from ξ_i^1, and of course from ξ_i^2 and ξ_i^3 too; the mixture states lie at points equidistant from their components. This also implies that $\sum_i \xi_i^1\xi_i^{mix} = N/2$ on average. To check the stability pick out the three special states with $\mu = 1, 2, 3$, still with all + signs, to find:

$$h_i^{mix} = \frac{1}{N} \sum_{j=1}^{N} \sum_{\mu} \xi_i^\mu \xi_j^\mu \xi_j^{mix} = \frac{1}{2}\xi_i^1 + \frac{1}{2}\xi_i^2 + \frac{1}{2}\xi_i^3 + \mathrm{cross-terms} \quad . \tag{7.39}$$

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 have nonzero inputs to have defined outputs of ±1.

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

7.7 The phase diagram of the Hopfield model

A statistical mechanical analysis by Amit, Gottfried and Sompolinsky (1985) shows that there is a crucial value α_c of $\alpha \equiv p/N$ where memory states no longer exist. A numerical evaluation gives

$$\alpha_c \approx 0.138 \quad . \tag{7.40}$$

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 input. 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').

7.8 Noise and spontaneous excitatory states

Before we leave the subject of the Hopfield model, it it worth stepping back and asking if, by connection with ferromagnetic systems, rate equations of the form used for the Hopfield model naturally go into an epileptic state of continuous firing, but not necessarily with every cell firing. This exercise also allows us to bring up the issue of fast noise that is uncorrelated from cell to cell.

We consider N binary neurons, with $N >> 1$, each of which is connected to all other neighboring neurons. For simplicity, we assume that the synaptic weights W_{ij} are the same for each connections, *i.e.*, $W_{ij} = W_0$. Then there is no spatial structure in the network and the total input to a given cell has two contributions. One term from the neighboring cells and one from an external input, which we also take to be the same for all cells and denote h_0. Then the input is

$$\text{Input} \;=\; W_0 \sum_{j=1}^{N} S_j \;+\; h_0. \tag{7.41}$$

The Energy per neuron, denoted ϵ, is then

$$\epsilon_i \;=\; -S_i\, W_0 \sum_{j=1}^{N} S_j \;-\; S_i\, h_0. \tag{7.42}$$

The insight for solving this system is the mean-field approach. We replace the sum of all neurons by the mean value of S_i, denoted $< S >$, where

$$< S > = \frac{1}{N} \sum_{j=1}^{N} S_j. \tag{7.43}$$

so that

$$\epsilon_i \;=\; -S_i\, (W_0 N < S > \; + h_0). \tag{7.44}$$

We can now use the expression for the value of the energy in term of the average spike rat, $< S >$, to solve self consistently for $< S >$. We know that the average rate is given by a Boltzman factor over all of the S_i. Thus

$$\begin{aligned} < S > \quad &= \frac{\sum_{S_i=-1}^{+1} e^{-\epsilon_i/k_BT}\; S_i}{\sum_{S_i=-1}^{+1} e^{-\epsilon_i/k_BT}} \qquad (7.45) \\ &= \frac{\sum_{S_i=-1}^{+1} e^{S_i (W_0 N <S> + h_0)/k_BT}\; S_i}{\sum_{S_i=-1}^{+1} e^{S_i (W_0 N <S> + h_0)/k_BT}} \\ &= \frac{-\, e^{-(W_0 N <S> + h_0)/k_BT} \; + e^{(W_0 N <S> + h_0)/k_BT}}{e^{-(W_0 N <S> + h_0)/k_BT} \; + \; e^{(W_0 N <S> + h_0)/k_BT}} \\ &= \tanh\left(\frac{W_0 N < S > + h_0}{k_BT}\right). \end{aligned}$$

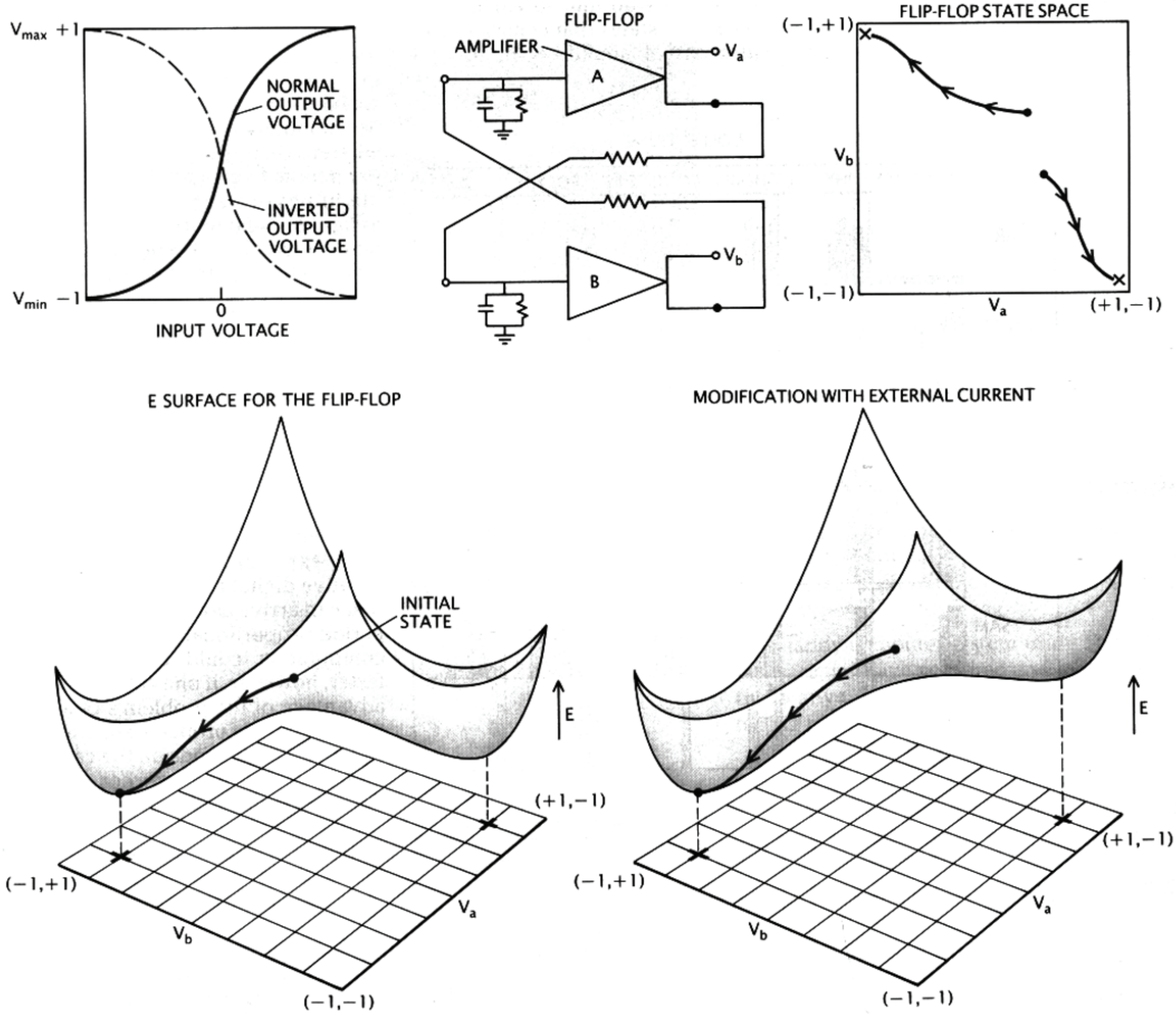
where we made of of the fact that $S_i = \pm 1$. This is the neuronal equivalent of the famous Weiss equation for ferromagnetism. The properties of the solution clearly depend on the ratio $\frac{W_0 N}{k_B T}$, which pits the connection strength W_0 against the noise level T/N.

- For $\frac{W_0 N}{k_B T} < 1$, the high noise limit, there is only the solution $< S >= 0$ in the absence of an external input h_0.

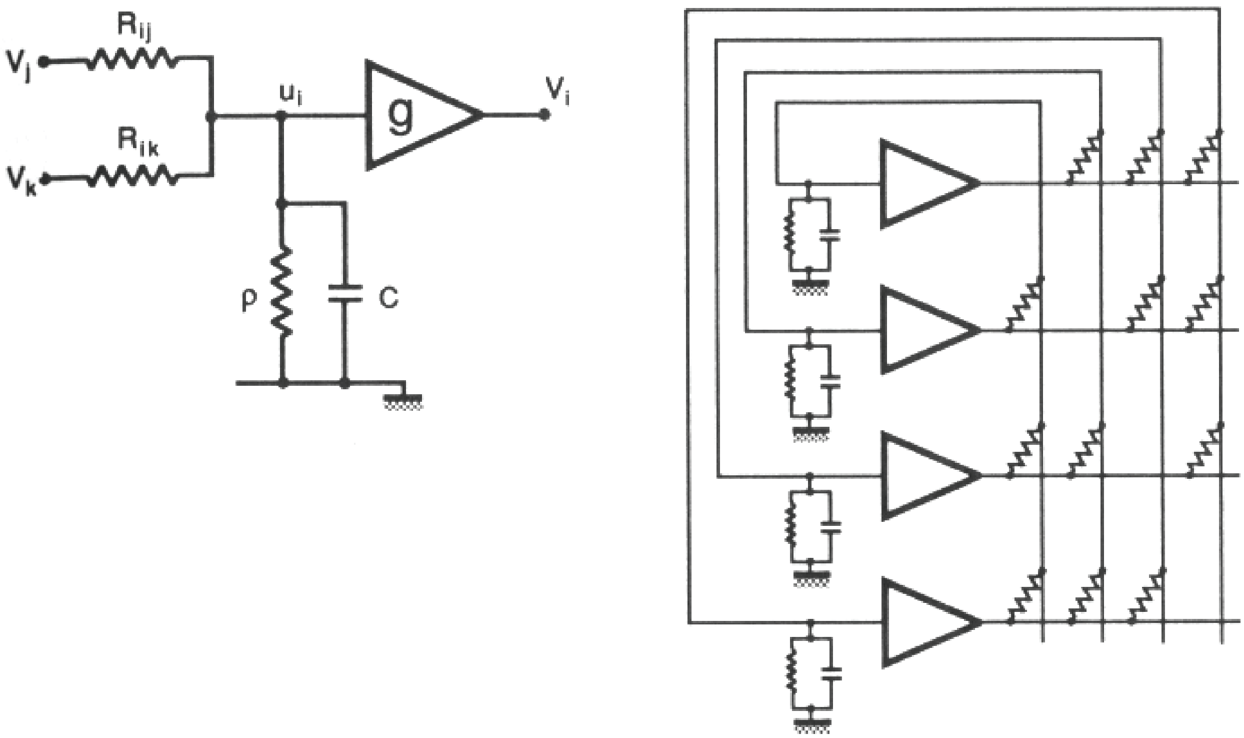
- For $\frac{W_0 N}{k_B T} > 1$, the low noise limit, there are three solutions in the absence of an external input h_0. One has $< S > = 0$ but is unstable. The other two solutions have $< S > \neq 0$ and must be found graphically or numerically.

- For sufficiently large $|h_0|$ the network is pushed to a state with $< S >= sgn(h_0/k_B T)$ independent of the interactions.

We see that there is a critical noise level for the onset of an active state and that this level depends on the strength of the connections and the number of cells. We also see that an active state can occur spontaneously for $\frac{W_0 N}{k_B T} > 1$ or $T < \frac{W_0 N}{k_B}$. This is a metaphor for epilepsy, in which recurrent excitatory connections maintain a spiking output.



V_max +1
V_min −1
NORMAL OUTPUT VOLTAGE
INVERTED OUTPUT VOLTAGE
0
INPUT 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_b
V_a
(−1,−1)
MODIFICATION WITH EXTERNAL CURRENT
E
(+1,−1)
(−1,+1)
V_b
V_a
(−1,−1)



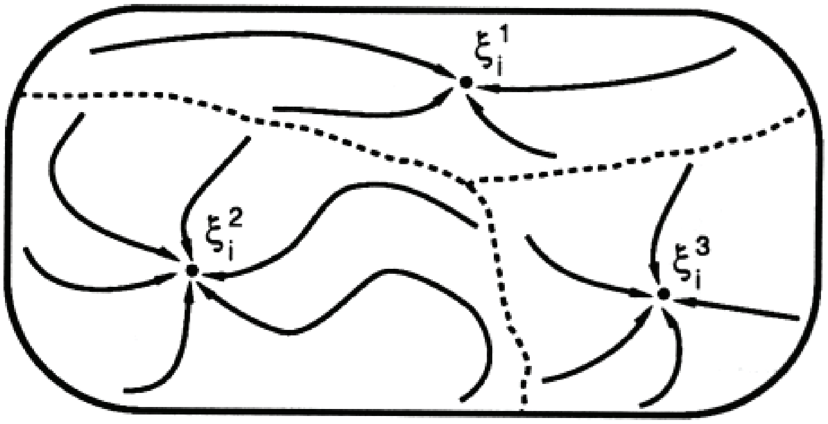
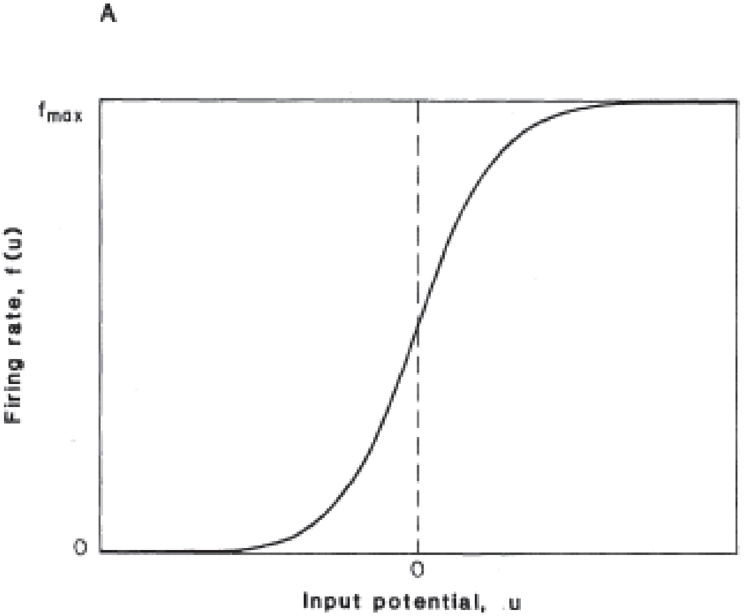


FIGURE 2.2 Schematic configuration space of a model with three attractors.

Herz_Fig.2.2



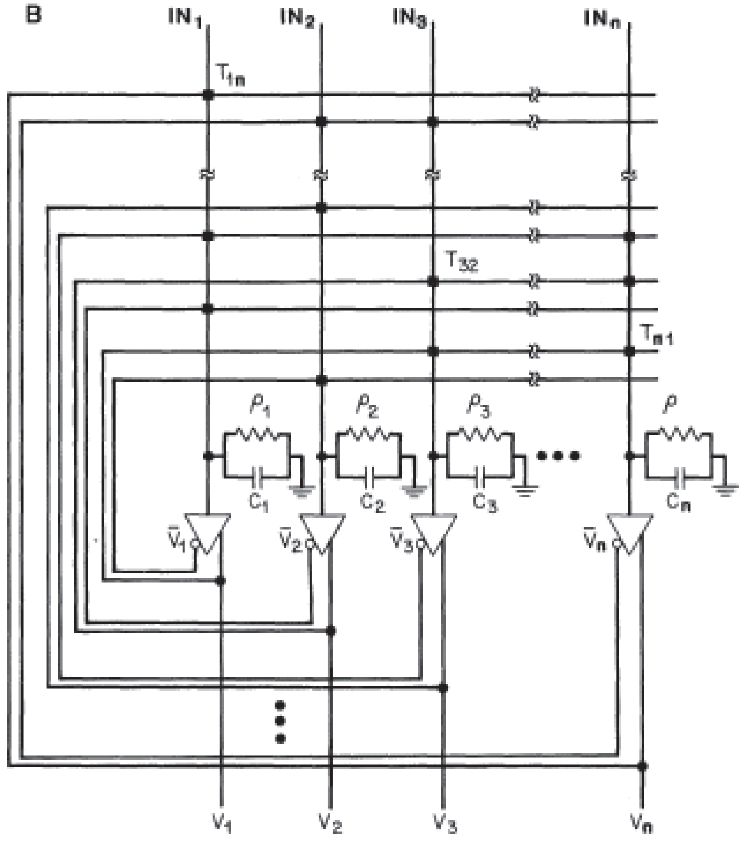


Fig. 3. (A) The sigmoid monotonic input-output relation used for the model neurons. (B) The model neural circuit in electrical components. The output of any neuron can potentially be connected to the input of any other neuron. Black squares at intersections represent resistive connections (with conductance T_{ij}) between outputs and inputs. Connections between inverted outputs (represented by the circles on the amplifiers) and inputs represent negative (inhibitory) connections.

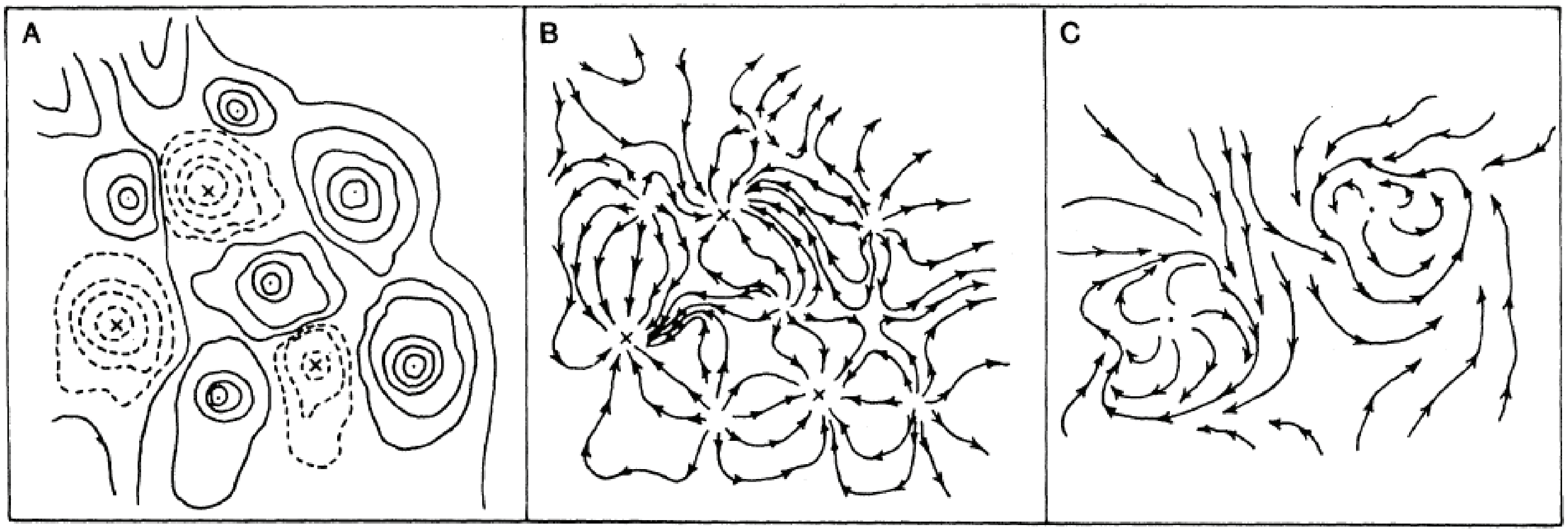
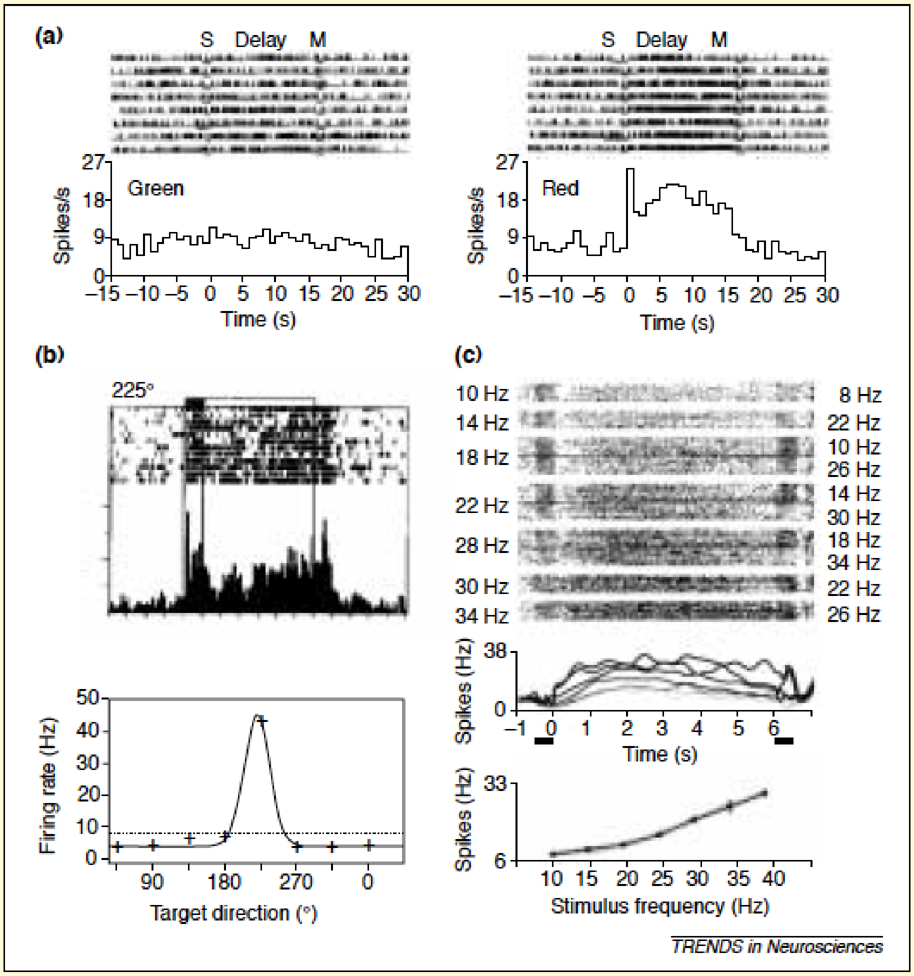


Fig. 4. (A) Energy-terrain contour map for the flow map shown in (B). (B) Typical flow map of neural dynamics for the circuit of Fig. 3 for symmetric connections ($T_{ij} = T_{ji}$). (C) More complicated dynamics that can occur for unrestricted (T_{ij}). Limit cycles are possible.

Box 1. Persistent activity as neural correlate of working memory

Cortical 'memory neurons' that show persistent activity are typically recorded during a delayed response task, in which a monkey is required to retain the information of a sensory cue across a delay period between the stimulus and behavioral response. Memory cells were first found, and seem to be especially abundant, in the prefrontal cortex (PFC)[a–e]. The crucial role of the PFC in working memory is also supported by lesion[f,g] and brain imaging studies[h]. However, neural persistent activity is a widespread phenonemon in association cortices, including the posterior parietal cortex[i–k], and the inferotemporal cortex[l,m]. According to the type of sensory stimulus that is encoded for storage, one can distinguish three kinds of working memory.



TRENDS in Neurosciences

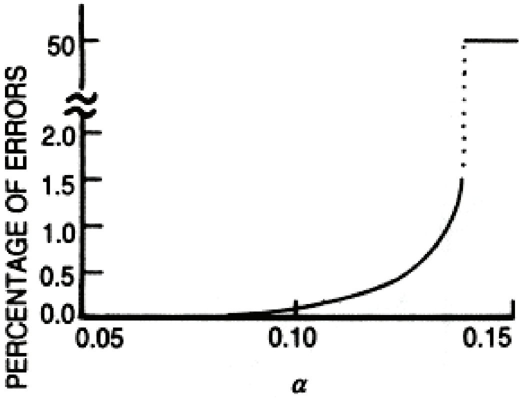
Fig. I. Three types of working memory encoding. (a) Discrete working memory. In a delayed match (M)-to-sample (S) experiment, an inferotemporal neuron shows sustained high activity for the color red (but not green) of a visual cue, during a delay period of 16 s. Redrawn, with permission, from Fuster and Jervey[l]. (b) Spatial working memory. In a delayed saccade experiment, a prefrontal neuron shows persistent activity that is tuned to a preferred location of a visual cue. Upper panel: rasters and cumulative spike histogram for a preferred cue; lower panel: spatial tuning curve of delay period activity. Redrawn, with permission, from Funahashi *et al.*[c] (c) Parametric working memory. In a delayed somatosensory discrimination task, a neuron in the inferior convexity shows persistent activity with a firing rate proportional to the cue frequency. Upper panel: rasters. Cue stimulus frequency indicated on the left, comparison stimulus frequency indicated on the right. Middle panel: trial-averaged firing rates as a function of time. Lower panel: mean firing rates, averaged across the entire delay period, as a function of the cue frequency. Redrawn, with permission, from Romo *et al.*[e]

Discrete working memory

Figure Ia shows a delayed match-to-sample experiment, in which the behavioral response depends on the memory of one of the two items (the stimulus color, red or green). An inferotemporal neuron displays elevated activity through the entire delay period (16 s), which is selective to the color red. Such tasks engage a working memory circuit in which the stored information is a categorical feature of the stimulus or an object (a face, color or word). Arguably, the items form a discrete collection, and a given neuron or neural assembly is selective to one or a few items.

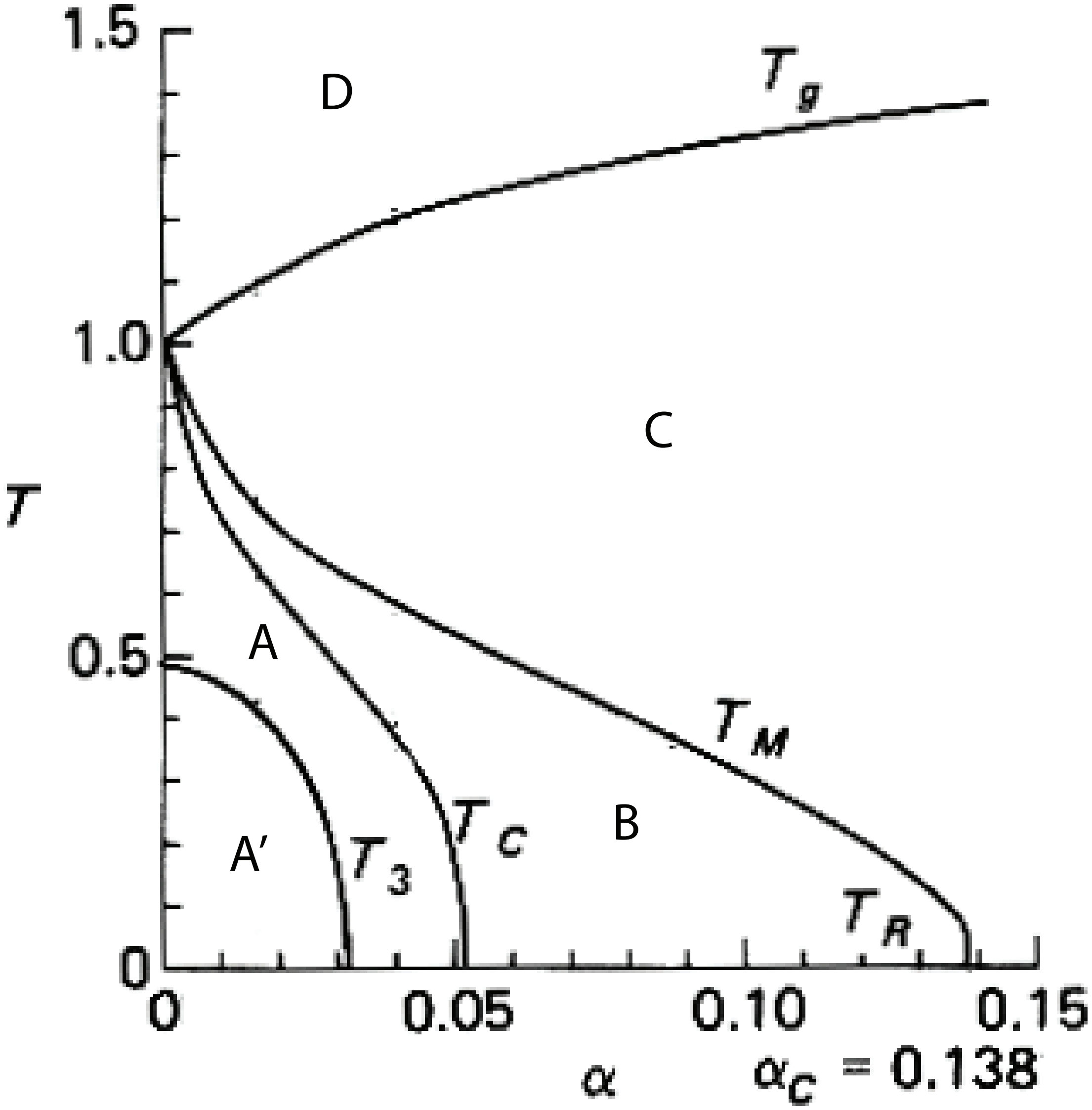
Spatial working memory

Figure Ib illustrates a delayed oculomotor experiment, in which a saccadic eye movement is guided by the memory of a spatial stimulus. In this case, the stored information is spatial location, which is an analog quantity. Neurons in the dorsolateral PFC display persistent delay activity that is spatially selective. The 'memory field' of a cell is characterized by a smooth tuning curve, peaked at a preferred spatial cue, which is different from cell to cell. The memory of a given spatial location is stored by the neural population in the form of a spatially localized persistent firing pattern, or a 'bump attractor'. Bump states are common to



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


T
D
C
A
A′
B
T_g
T_M
T_C
T_3
T_R
α
$\alpha_C = 0.138$