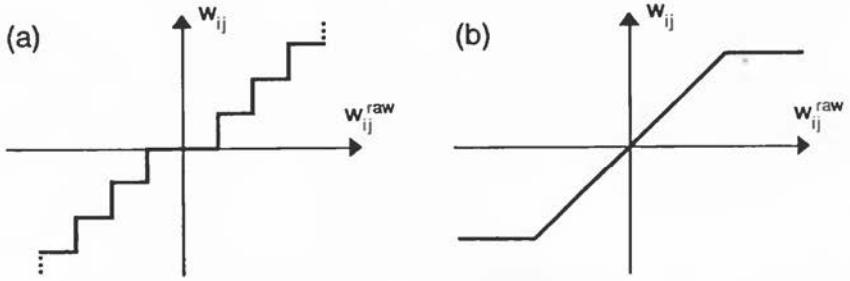


FIGURE 3.1 Examples of (a) discretization and (b) clipping. The raw connection weight w_{ij}^{raw} calculated from the Hebb rule is transformed to the actual w_{ij} by the function shown.

The addition of a small random number to each w_{ij} has only a qualitative effect, reducing α_c [Sompolinsky, 1987]. Similarly with **discretizing** or **clipping** the allowed values. Discretizing—allowing only a discrete set of values—may be useful when building circuits using a fixed number of standard resistors. Clipping means restricting all connections w_{ij} to some fixed range, say $|w_{ij}| \leq A$, and is clearly also useful (if not essential) in practical implementations. Figure 3.1 shows examples of both processes.

In the most extreme case of discretization and clipping we allow only two values for w_{ij}, sometimes referred to as **binarizing** the connections:

$$w_{ij} = \mathrm{sgn}(w_{ij}^{raw}) = \mathrm{sgn}\left(\sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu\right). \tag{3.1}$$

This model can be solved exactly [van Hemmen and Kühn, 1986]; the result is that α_c is reduced from 0.138 to about 0.1. This represents a rather efficient use of the single bit of information retained per connection, compared to the $\log_2 p$ bits necessary to specify one of the p possible values of each w_{ij} with the full Hebb rule.

Clipping may also be viewed in the context of successively learning new patterns. We can imagine using the Hebb rule *incrementally* to continue adding new terms to each w_{ij}, so that

$$w_{ij}^{new} = w_{ij}^{old} + \eta \xi_i^\mu \xi_j^\mu \tag{3.2}$$

to add pattern μ. Here η is an **acquisition rate**. Applying clipping to this means restricting w_{ij}^{new} to a range $[-A, A]$ at all times; values outside these limits are immediately replaced by the appropriate limit value. This is called **learning within bounds** [Parisi, 1986; Nadal et al., 1986]. The most recently added memory patterns are then always recalled well, while older ones gradually decay away. The number of patterns that can be remembered depends on the value of η compared to A; if η is very large only the most recent pattern can be recalled, while for very small η the

theory as before, replacing h_i by

$$\langle h_i \rangle = c \sum_j w_{ij}^{\text{Hebb}} \langle S_j \rangle . \tag{3.6}$$

Thus the previous mean field results apply (for p/N small), with a simple scaling of the temperature by a factor of c.

At larger p/N the situation is more complicated, but the qualitative feature of a capacity $p_{\max}$ of order N persists for the case of symmetric dilution $C_{ij} = C_{ji}$ [Sompolinsky, 1987; Canning and Gardner, 1988].

Strong Dilution

There is another limit of the dilution problem which can be solved exactly and rather simply. This is the case of **strong dilution**, where only an infinitesimal fraction of the original number of connections remain. Defining K as the average number of connections to and from each unit, the precise condition is that K not exceed something proportional to $\log N$ as N goes to infinity. The exact solution also requires another twist: the dilution must be performed independently on w_{ij} and w_{ji}, so that the factors C_{ij} and C_{ji} in (3.4) are independent random variables. The w_{ij} matrix is then no longer symmetric.

This model, first studied by Derrida, Gardner, and Zippelius [1987], can then be solved. We use

$$w_{ij} = \frac{1}{K} C_{ij} \sum_\mu \xi_i^\mu \xi_j^\mu \tag{3.7}$$

for the connection strengths, with $1/K$ rather than $1/N$ for the normalization so as to give sensible values of order unity. For any state $\{S_j\}$ of the network, we now break up the field h_i in (3.5) into a term coming from a particular pattern ν and a remaining crosstalk term:

$$\begin{aligned} h_i &= \sum_j w_{ij} S_j = \frac{1}{K} \sum_j C_{ij} \sum_\mu \xi_i^\mu \xi_j^\mu S_j \\ &= \frac{1}{K} \xi_i^\nu \sum_j C_{ij} \xi_j^\nu S_j + \eta_i^\nu \end{aligned} \tag{3.8}$$

where

$$\eta_i^\nu = \frac{1}{K} \sum_{\mu \neq \nu} \xi_i^\mu \sum_j C_{ij} \xi_j^\mu S_j . \tag{3.9}$$

Note that the crosstalk term η_i^ν depends on the state $\{S_j\}$.

If we set $S_i = \xi_i^\nu$ in (3.8) and (3.9) we can see that ξ_i^ν is stable for small enough p. The first term on the right-hand side of (3.8) gives just ξ_i^ν on average, since $\langle \sum_j C_{ij} \rangle = K$. Meanwhile the second term η_i^ν, given by (3.9), becomes $1/K$

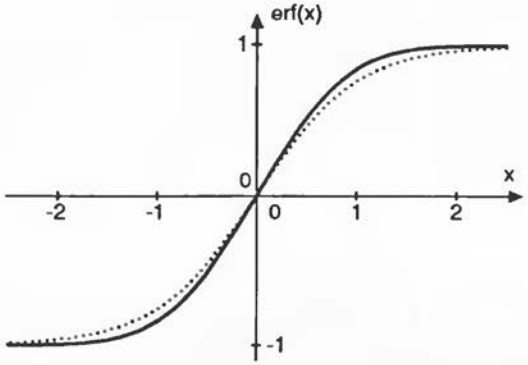


FIGURE 3.2 The function $\mathrm{erf}(x)$. The dotted line shows $\tanh(x)$ for comparison.

There is a critical value $\alpha_c' = 2/\pi$ of α' beyond which the only solution is $m_\nu = 0$, but below which there are solutions with $m_\nu \neq 0$. Thus the crosstalk acts in a way similar to thermal noise. Note however that m_ν goes continuously to zero as α' approaches α_c', in contrast to what happened in the fully connected case as α approached α_c, where the jump down to zero was discontinuous. The origin of this difference can be understood by comparing (3.13) with the corresponding equation (2.73) for the fully connected case. The latter had an extra factor of $1/\sqrt{r}$ in the argument of the erf, and r (and q) had to be determined self-consistently with m.

We can generalize the treatment to finite temperature simply by replacing the $\mathrm{sgn}(x)$ in (3.12) by a $\tanh(x)$:

$$m_\nu = \int d\eta\, P(\eta) \tanh[\beta(m_\nu + \eta)]\,. \tag{3.14}$$

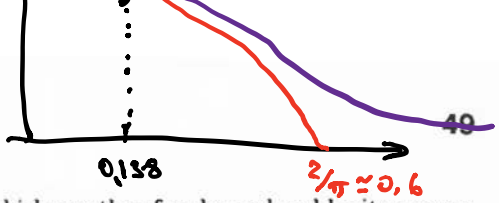
As in the full connectivity case, the effect of finite temperature is to reduce the capacity from its zero-temperature value.

The model can be solved for both synchronous and asynchronous updating (with the same result for the capacity), but is apparently not so easy to solve if the connection matrix is constrained to be symmetric.

Random Asymmetric Connections

The densely connected model may also be studied when the connections are allowed to be asymmetric, $w_{ij} \neq w_{ji}$. If the asymmetry is systematic, or very strong, it can produce limit cycles or chaotic behavior, as we will study later. But if it is random and not too strong it mainly plays the role of noise. Random asymmetry can be introduced by random dilution or by adding a random number to each connection, independently for ij and ji in both cases.

For $p \ll N$ there seems to be no difference from the symmetric case. As in the case of weak dilution, the argument relies on the fact that there are still of order N



terms in the weighted input sum h_i, which can therefore be replaced by its average value without approximation in the large N limit.

At finite α, the asymmetric fully connected model differs from the symmetric one in that it does not have any of the spurious spin glass states at any $T > 0$ [Hertz et al., 1986, 1987; Crisanti and Sompolinsky, 1987]. But retrieval of the stored patterns is not qualitatively different from that in the symmetric model. Thus random asymmetry may improve performance by removing the spin glass states. On the other hand, the asymmetry introduces some fluctuations and slows down the approach to an attractor [Parisi, 1986].

Unipolar Connections

For some applications it is inconvenient to require both positive and negative connection weights w_{ij}. This is particularly true when implementing a network in electronic or optical hardware, as discussed in Section 3.4. It is however easy to modify the design so that all the weights are positive; i.e., we can replace *bipolar* connections by *unipolar* ones [Denker, 1986].

The trick is simply to add a constant κ to every connection w_{ij},

$$w'_{ij} = w_{ij} + \kappa \tag{3.15}$$

and compensate for this with an extra term $-\kappa \sum_j S_j$ in the input h_i at every unit. Then the total input h'_i is given by

$$h'_i = \sum_j (w_{ij} + \kappa) S_j - \kappa \sum_j S_j = \sum_j w_{ij} S_j = h_i \tag{3.16}$$

exactly as before. Thus there is *no* overall effect on the network's behavior.

We choose κ large enough to make w'_{ij} positive (or perhaps zero) for all ij. For the usual Hebb rule (2.9) the value $\kappa = 1$ suffices. The compensating term $-\kappa \sum_j S_j$ is the same for all units, and may be calculated by one extra unit. It is sometimes referred to as an **adaptive threshold** because in effect it changes the threshold of every unit by an amount depending on the total activity $\sum_j S_j$.

3.2 Correlated Patterns

The Pseudo-Inverse

The crosstalk term in h_i which sets the fundamental limit on the network capacity comes from the overlap between random patterns. This overlap is much more of a problem when the different patterns are correlated. Then a standard network may not even recall patterns reliably in the limit $p \ll N$. There is however a general

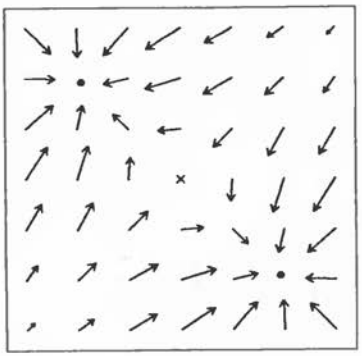


FIGURE 3.3 Motion towards attractors in a two-unit network. There are two attractors shown by dots. The center point is a saddle point of the energy, not an attractor. The system moves in the direction of the arrows to one of the attractors; *which* one depends on the starting point.

- **Asynchronous updating.** One unit at a time is selected to have its output set according to (3.30).
- **Synchronous updating.** At each time step all units have their outputs set according to (3.30).
- **Continuous updating.** All units continuously and simultaneously change their outputs towards the values given by (3.30). The u_i's change continuously too, according to $u_i = \sum_j w_{ij} V_j$.

The third possibility is the new one [Cohen and Grossberg, 1983; Hopfield, 1984], and is of particular interest for the circuit implementations discussed in Section 3.4. It can be represented by the set of differential equations

$$\tau_i \frac{dV_i}{dt} = -V_i + g(u_i) = -V_i + g\Big(\sum_j w_{ij} V_j\Big) \tag{3.31}$$

where τ_i are suitable time constants.

If $g(u)$ has a saturation nonlinearity and the w_{ij}'s are symmetric, the solution $V_i(t)$ to these equations always settles down to a stable equilibrium solution, as we will prove in the next subsection. At an equilibrium $V_i(t)$ ceases to change, so $dV_i/dt = 0$ for all i. Then the right-hand side of (3.31) shows that (3.30) is obeyed on all units. Thus the desired state satisfying (3.30) is an **attractor** of the dynamical rule (3.31).

Figure 3.3 shows a simple example for a system with two units. A state of the system corresponds to a point in the V_1–V_2 plane illustrated. At any such point, the equations (3.31) (one for $i = 1$, one for $i = 2$) give a velocity vector $d\mathbf{V}/dt$, shown in the figure by an arrow. The state of the system moves from its initial point in the direction of the arrows, faster for larger arrows. Thus it ends up at one of the two attractors shown, where $d\mathbf{V}/dt = 0$.

A very similar dynamical rule with the same end result arises from letting the inputs u_i continuously approach their correct values $\sum_j w_{ij} V_j$, with $V_i = g(u_i)$