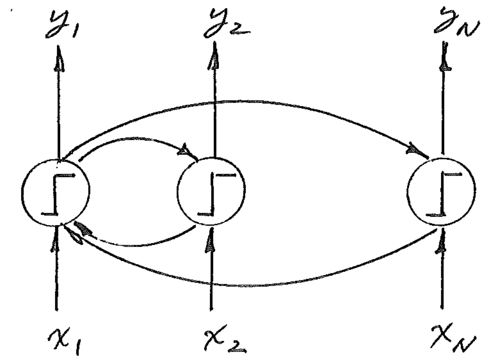
Hopfield Model



Hopfield networks are a special case of recurrent systems that use threshold PEs, do not have hidden PEs, and where interconnection matrix is symmetric.

The original Hopfield model did not have self-recurrent connections.

We can describe the discrete Hopfield model in discrete time as

$$y_i(n+1) = \mathrm{sgn}\left(\sum_{j=1}^{N} w_{ij}\, y_j(n) + b_i + x_i(n) \right), \quad i=1,\cdots,N$$

where sgn represents the threshold nonlinearity $(-1, 1)$, and b is a bias.

We assume that the update is done sequentially by PE number. The input is a binary number

$$x = [x_1, x_2, \cdots, x_N]^T$$

With $x_i = \pm 1$, the state of neuron i represents one bit of information, and the $N \times 1$ state vector represents a binary word of N bits of information. The input works as an initial condition; it is presented and then taken away.

The induced net input (local field) n_i of neuron i is defined by

$$n_i = \sum_{j=1}^{N} w_{ij}\, x_j + b_i$$

Hence, neuron i modifies its state x_i according to the <u>deterministic rule</u>

$$y_i = \begin{cases} +1 & \text{if } n_i > 0 \\ -1 & \text{if } n_i < 0 \end{cases}$$

i.e.,

$$y_i = \mathrm{sgn}(n_i)$$

Discrete Hopfield Model as a Content-Addressable Memory:

Suppose that we have a binary input pattern x^p, and we want this input pattern to be stable at the output, i.e., we want the system to be an autoassociator and produce a stable output pattern

$$y = x^p.$$

The asynchronous (serial) updating procedure is continued until there are no further changes to report. That is, starting with the probe vector x^p, the network finally produces a time invariant state vector y whose individual elements satisfy the condition for stability:

$$y_i = \mathrm{sgn}\left(\sum_{j=1}^{N} w_{ij} y_j + b_i \right), \quad i = 1, 2, \cdots, N$$

or, in a matrix form,

$$y = \mathrm{sgn}(Wy + b)$$

where W is the synaptic weight matrix of the network, and b is the externally applied bias vector.

The stability condition described here is also called the alignment condition. The state vector y that satisfies it is called a stable state or fixed point of the state space of the system.

Hopfield network will always converge to a stable state when the retrieval operation is performed asynchronously.

Storage Phase: Suppose we wish to store the pattern

$$x^p = [x_1^p, x_2^p, \cdots, x_N^p]^T$$

Then the weight can be created by the Hebbian learning:

$$w_{ij} \propto x_i^p x_j^p$$

In a matrix form,

$$W = x^p x^{p^T} = \begin{bmatrix} x_1^p \\ x_2^p \\ \vdots \\ x_N^p \end{bmatrix} [x_1^p \ x_2^p \ \cdots \ x_N^p]$$

which is the <u>outer product</u> of patterns.

Note that the weight matrix W is a <u>symmetric</u> matrix. But notice that this associative memory is <u>dynamic</u>; when we present the input x to the network as an initial condition, the system dynamics produce an output sequence $y(n)$ that takes some time to stabilize but will approximate the stored pattern.

The attractive feature of Hopfield networks is that even when the input is partially deleted or corrupted by noise, the system dynamics still take the output sequence $y(n)$ to x^p.

Since the system dynamics are converging to x^p, we call this solution a <u>point attractor</u> for the dynamics.

The system can also store multiple patterns P with a weight matrix given by

$$w_{ij} = \frac{1}{N} \sum_{p=1}^{P} x_i^p x_j^p$$

In the normal operation, we let

$$w_{ii} = 0 \quad \text{for all } i$$

which means that the neurons have no self-feedback. In matrix form,

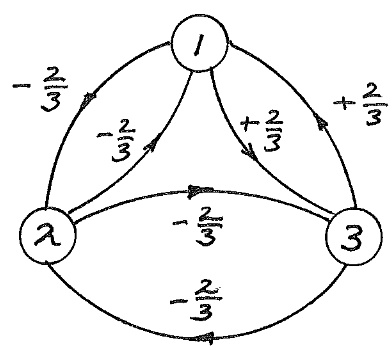
$$W = \frac{1}{N} \left(\sum_{p=1}^{P} x^p x^{p^T} - PI \right)$$

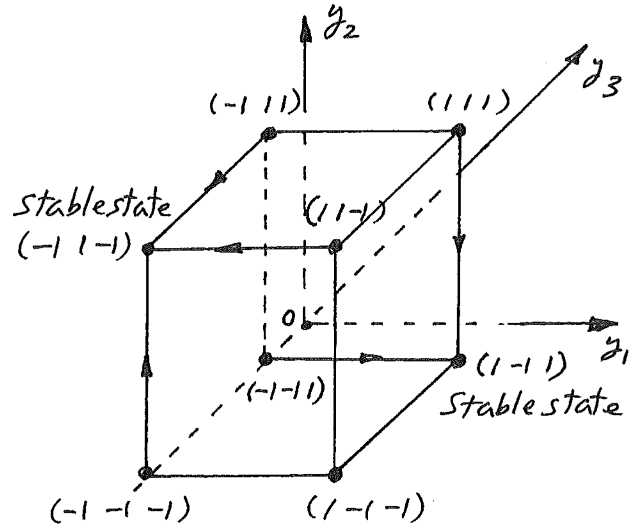
where $x^p x^{p^T}$ is the outer product and I denotes the identity matrix.

Note that there is <u>crosstalk</u> between the memory patterns, and recall is possible only if the number of (random) input patterns is smaller than 0.15N, where N is the number of inputs.

This is rather disappointing, but the intriguing property of Hopfield network is <u>pattern completion</u> and <u>noise removal</u>, which is obtained by the convergence of the system state to the attractor.

<u>Example</u>: Hopfield Model





The weight matrix is

$$W = \frac{1}{3}\begin{bmatrix} 0 & -2 & +2 \\ -2 & 0 & -2 \\ +2 & -2 & 0 \end{bmatrix}$$

symmetric & $w_{ii} = 0$

Assume bias is zero.

With 3 neurons, there are $2^3 = 8$ possible states.
Of these 8 states, only 2 states (1 −1 1) & (−1 1 −1) are stable; the remaining 6 states are all unstable.

The two stable states satisfy the alignment condition.
For the state vector (1 −1 1),

$$Wy = \frac{1}{3}\begin{bmatrix} 0 & -2 & +2 \\ -2 & 0 & -2 \\ +2 & -2 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 4 \\ -4 \\ 4 \end{bmatrix}$$

Hard limiting this result yields

$$\mathrm{sgn}(Wy) = \begin{bmatrix} +1 \\ -1 \\ +1 \end{bmatrix} = y$$

Similarly, for $(-1\ 1\ -1)$,

$$Wy = \frac{1}{3}\begin{bmatrix} 0 & -2 & 2 \\ -2 & 0 & -2 \\ 2 & -2 & 0 \end{bmatrix}\begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} -4 \\ 4 \\ -4 \end{bmatrix}$$

which yields

$$\text{sgn}(Wy) = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = y$$

Hence, both of the state vectors satisfy the alignment condition.

The flow map exhibits symmetry with respect to the two stable states.

The unstable states converge to stable states in one iteration.

The network therefore has two <u>fundamental memories</u>, $(1\ -1\ 1)$ and $(-1\ 1\ -1)$, representing the two stable states.

Check for the synaptic weight matrix:

$$W = \frac{1}{3}\begin{bmatrix} +1 \\ -1 \\ +1 \end{bmatrix}\begin{bmatrix} +1 & -1 & +1 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} -1 \\ +1 \\ -1 \end{bmatrix}\begin{bmatrix} -1 & +1 & -1 \end{bmatrix} - \frac{2}{3}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \frac{1}{3}\begin{bmatrix} 0 & -2 & +2 \\ -2 & 0 & -2 \\ +2 & -2 & 0 \end{bmatrix}$$

<u>NOTE</u>: The error correcting capability of the Hopfield network is readily seen by examing the flow map:

1. If the probe vector applied to the network equals $(-1\ -1\ 1)$, $(1\ 1\ 1)$, or $(1\ -1\ -1)$, then the resulting output is the fundamental memory $(1\ -1\ 1)$.

   Each of these values of the probe represents a single error, compared to the stored pattern.

2. Similary, $(1\ 1\ -1)$, $(-1\ -1\ -1)$, or $(-1\ 1\ 1)$ produce the output $(-1\ 1\ -1)$.

Spurious States:

The weight matrix W of a discrete Hopfield network is symmetric. The eigenvalues of W are therefore all real. For large P, the eigenvalues are ordinarily degenerate, which means there are several eigenvectors with the same eigenvalue. The eigenvectors associated with a degenerate eigenvalue form a subspace. Furthermore, the weight matrix W has a degenerate eigenvalue with a value of zero, in which case the subspace is called the null space.

The null space exists by virtue of the fact that the number of fundamental memories, P, is smaller than the number of neurons, N. The presence of a null space is an intrinsic characteristic of the Hopfield network.

By Aiyer et al. (1990),

1. The discrete Hopfield network acts as a vector projector in the sense that it projects a probe vector onto a subspace $\mathcal{M}$ spanned by the fundamental vectors.

2. The underlying dynamics of the network drive the resulting projected vector to one of the corners of a unit hypercube where the energy function is minimized.

The unit hypercube is N-dimensional. The P fundamental memory vectors, spanning the subspace $\mathcal{M}$, constitute a set of fixed points (stable states) represented by certain corners of the unit hypercube.

The other corners of the unit hypercube that lie in or near subspace $\mathcal{M}$ are potential locations for spurious states, also referred to as spurious attractors (Amit, 1989). Spurious states represent stable states of the Hopfield network that are different from fundamental memories of the network.

<u>Example:</u> A computer experiment to illustrate the behavior of the discrete Hopfield network as a content-addressable memory. The network has $N = 120$ neurons, and therefore $N^2 - N = 12{,}280$ synaptic weights.

It is trained to retrieve the eight digitlike black and white patterns, with each pattern containing 120 pixels.

The inputs applied to the network assume the value +1 for black pixels and −1 for white pixels.

The 8 patterns were used as fundamental memories in the storage (learning) phase to create the synaptic weight matrix W.

To demonstrate the error-correcting capability of the Hopfield network, a pattern was distorted by randomly and independently reversing each pixel with a probability of 0.25, and then using the corrupted pattern as a probe.

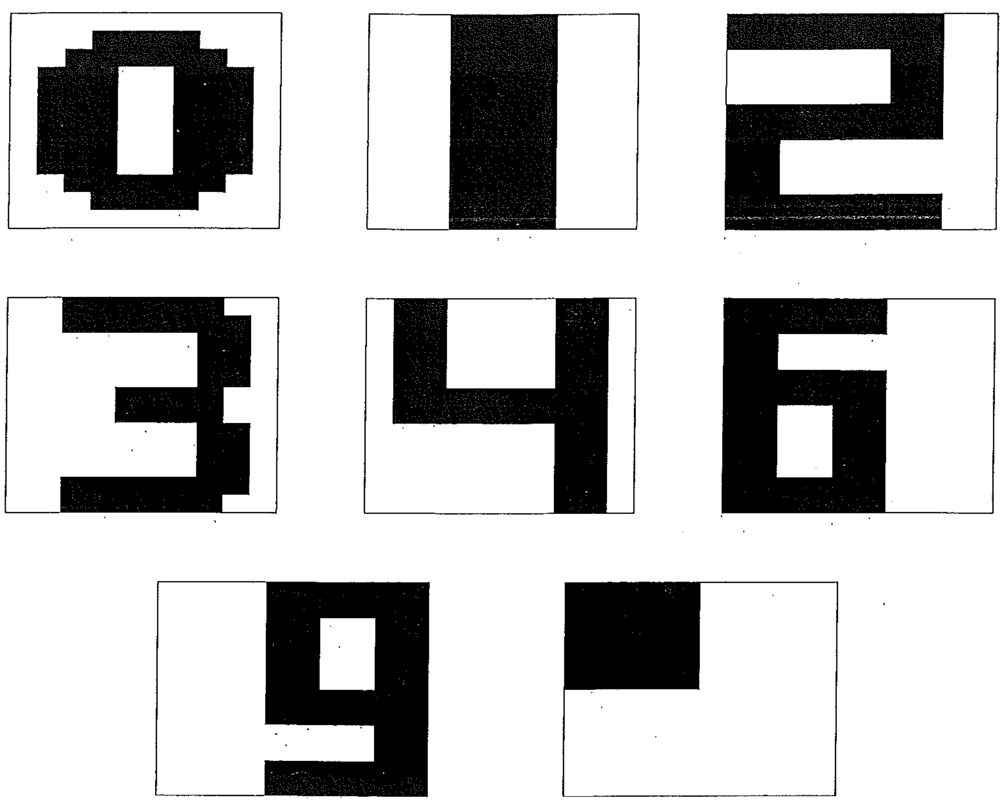


FIGURE 14.17 Set of handcrafted patterns for computer experiment on the Hopfield network.

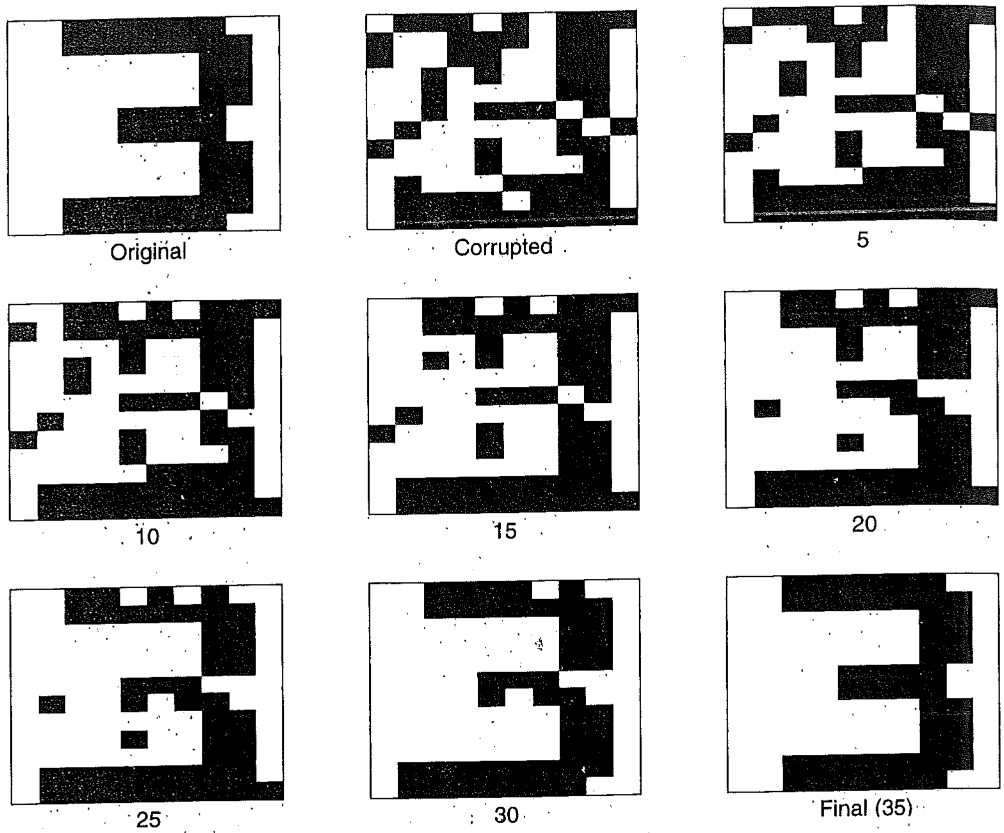


FIGURE 14.18 Correct recollection of corrupted pattern 3.

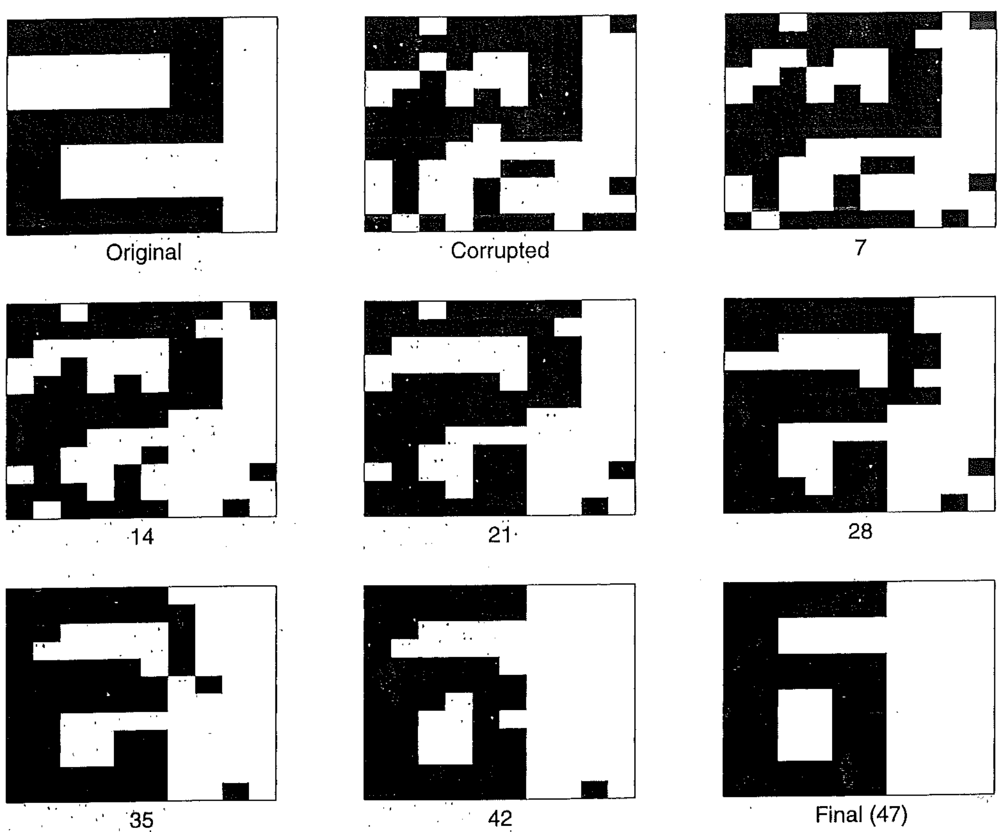


FIGURE 14.19 Incorrect recollection of corrupted pattern 2.

Found 108 spurious attractors in 43,097 tests of randomly selected digits corrupted with the probability of flipping a bit set at 0.25. They may be grouped as:

I. Reversed fundamental memories (1,1) position: −6

II. Mixture states (6×4): −1, 4, 9

III. Spin-glass states (7×6): not correlated with any.

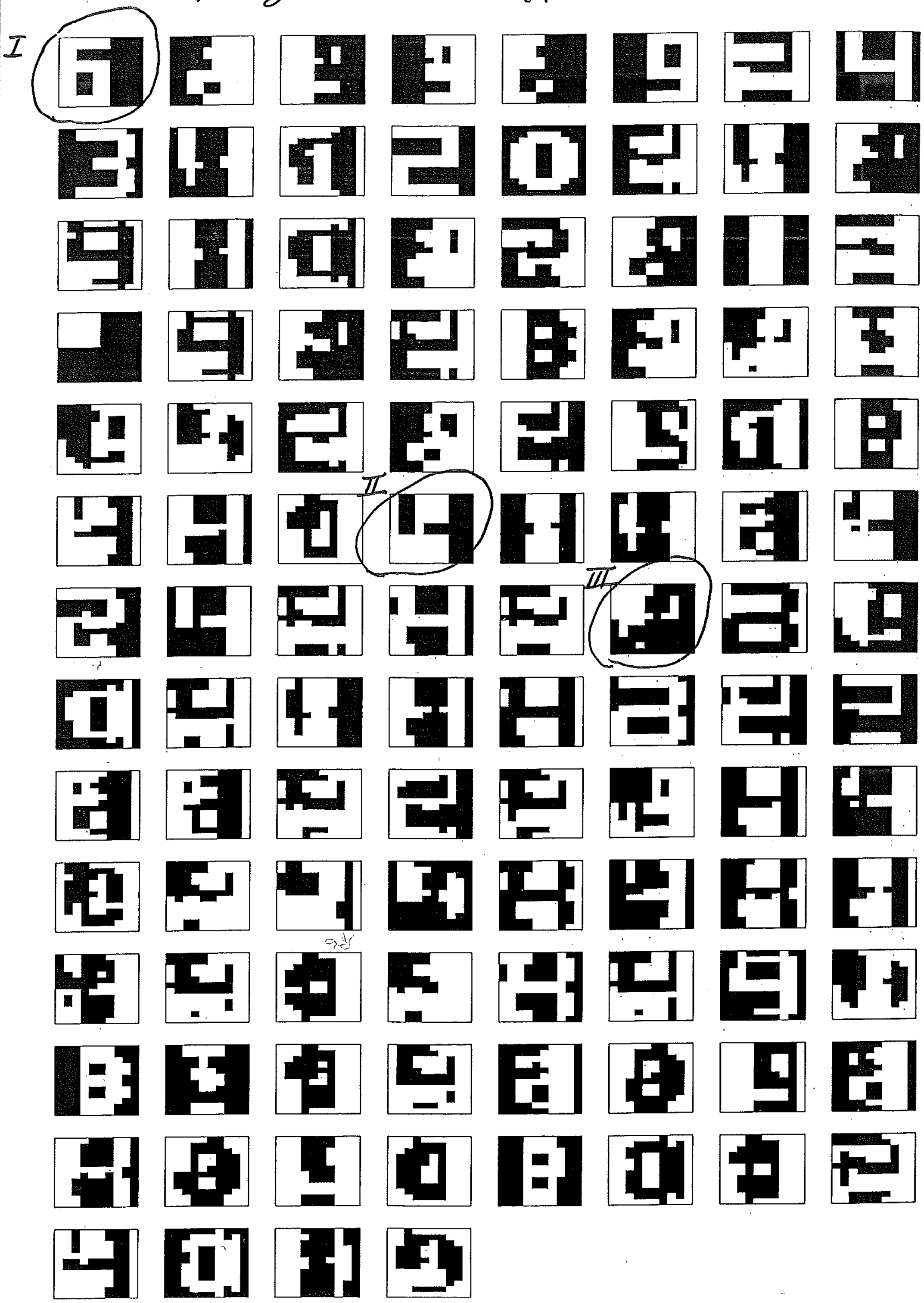
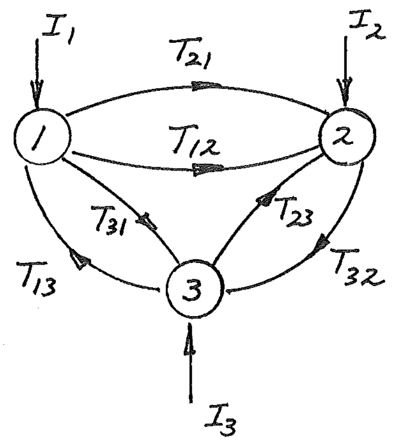


FIGURE 14.20 Compilation of the spurious states produced in the computer experiment on the Hopfield network.

Energy Function:

So far we assumed that the dynamics of the Hopfield model converge to a point attractor.

We now prove that this is true by defining an Energy Function.



Let

v_i : output of neuron i

u_i : threshhold "

Define the Energy function,

$$E = -\frac{1}{2} \sum_{i \neq j} \sum T_{ij} v_i v_j - \sum_i I_i v_i + \sum_i u_i v_i$$

The state of neuron is changed by Δv_i

$$\Delta E = -\left[\sum_{j \neq i} T_{ij} v_j + I_i - u_i \right] \Delta v_i$$

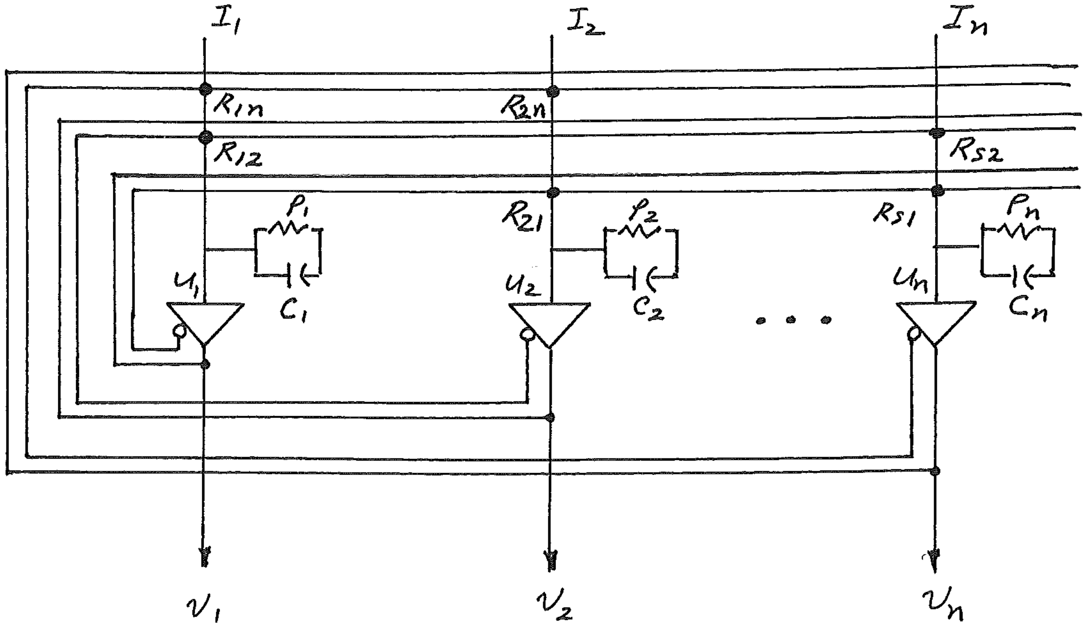
Therefore,

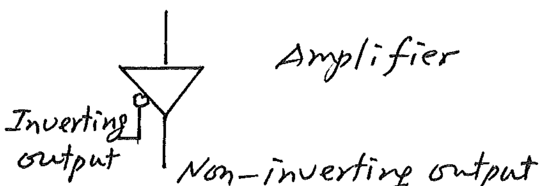
$$\sum_{j \neq i} T_{ij} v_j + I_i - u_i > 0 \ : \ \text{increase } V_i$$

$$\sum_{j \neq i} T_{ij} v_j + I_i - u_i < 0 \ : \ \text{decrease } V_i$$

If E is bounded, then the dynamics will reach to stable state.

Hopfield Model (Continuous)





- R_{ij} : resistor connecting the output of the jth amplifier to the input of the ith amplifier

$|T_{ij}| = \frac{1}{R_{ij}}$, allow T_{ij} to have both negative & positive.

τ_i : equivalent resistance,

$$\frac{1}{\tau_i} = \frac{1}{\rho_i} + \sum_{j=1}^{n} \frac{1}{R_{ij}}$$

Amplifier dynamics:

$$C_i \frac{du_i}{dt} = \sum_{j=1}^{n} T_{ij} v_j - \frac{u_i}{\tau_i} + I_i$$

$$v_i = g_i(u_i)$$

where u_i : input to the ith amplifier
v_i : output of the ith amplifier
I_i : external input

Assume:

1. $T_{ij} = T_{ij}$, symmetric
2. Inverse of the nonlinear activation function exists,

$$u_i = g_i^{-1}(v_i)$$

Example: $v = g_i(u) = \tanh\left(\frac{a_i u}{2}\right) = \frac{1 - \exp(-a_i u)}{1 + \exp(a_i u)}$

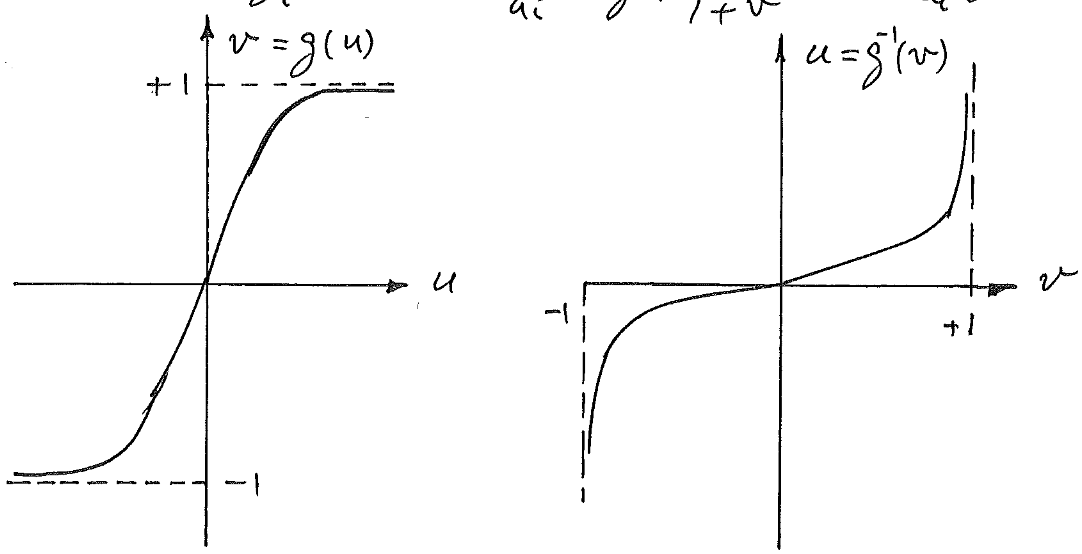
which has a lo e at the origin:

$$\left.\frac{d g_i}{du}\right|_{u=0} = \frac{a_i}{2}$$

Hence we refer to a_i as the gain of neuron i.

The inverse of the sigmoid function g_i is

$$u = g_i^{-1}(v) = -\frac{1}{a_i} \log\left(\frac{1-v}{1+v}\right) = \frac{1}{a_i} g^{-1}(v)$$



Standard sigmoidal nonlinearity $g(u)$ and its inverse ($a_i = 1$).

Define N-dim. state vector

$$v = [v_1, v_2, \cdots, v_N]^T$$

Define an Energy Function

$$E = -\frac{1}{2}\sum_i \sum_j T_{ij} v_j v_i + \sum \frac{1}{\tau_i}\int_0^{v_i} g_i^{-1}(v_i)\, dv_i - \sum_i I_i v_i$$

Note,

$$\frac{d}{dt}\int_0^{v_i} g_i^{-1}(v_i)\,dv_i = \frac{d}{dt}\int_0^{v_i} u_i\, dv_i = \frac{d}{dt}\int_0^{t} u_i \frac{dv_i}{dt}\, dt$$

$$= u_i \frac{dv_i}{dt}$$

Thus,

$$\frac{dE}{dt} = -\sum_i \frac{dv_i}{dt}\Big(\underbrace{\sum_j T_{ij} v_j - \frac{u_i}{\tau_i} + I_i}_{C_i \frac{du_i}{dt}} \Big)$$

$$= -\sum_i \frac{dv_i}{dt} C_i \frac{du_i}{dt}$$

$$= -\sum_i C_i \left[\frac{d}{dt} g_i^{-1}(v_i)\right] \frac{dv_i}{dt}$$

$$= -\sum_i C_i \underbrace{\left[\frac{d}{dv_i} g_i^{-1}(v_i)\right]}_{\geq 0} \left(\frac{dv_i}{dt}\right)^2$$

Since $g_i^{-1}(\cdot)$ is a monotonically increasing function,

$$\frac{d}{dv_i} g_i^{-1}(v_i) \geq 0 \quad \text{for all } i.$$

Hence, $\frac{dE}{dt} \leq 0$, and $\frac{dE}{dt} = 0$ only when $\frac{dv_i}{dt} = 0 \ \forall i$.

$v_i \ni \frac{dv_i}{dt} = 0$: asymptotically stable equilibrium points, fixed points

<u>Note</u>, E is bounded. Accordingly, we make the following statements:

1. The energy function E is a <u>Lyapunov function</u> of the continuous Hopfield model.
2. The model is <u>stable</u> in accordance with the Lyapunov's Theorem.

In other words, the Hopfield model represents a trajectory in the state space, which seeks out the minima of the energy (Lyapunov) function E and comes to a stop at such fixed points, i.e.,

$$\frac{dE}{dt} < 0 \quad \text{except at a fixed point.}$$

Thus, we have:

<u>Thm:</u> The (Lyapunov) energy function E of the Hopfield network is a monotonically decreasing function of time.

Accordingly, the Hopfield network is <u>globally asymptotically stable</u>; the attractor fixed-points are the minima of the energy function, and vice versa.

Relation between the Discrete and Continous Models

Continuous model : based on additive model
Discrete model : based on McCulloch-Pitts model

Stable states of both models satisfy the following simplifying characteristics:

1. The output of a neuron has the asymptotic values

$$v_j = \begin{cases} +1 & \text{for } u_j = \infty \\ -1 & \text{for } u_j = -\infty \end{cases}$$

2. The midpoint of the activation function of a neuron lies at the origin,

$$g_j(0) = 0.$$

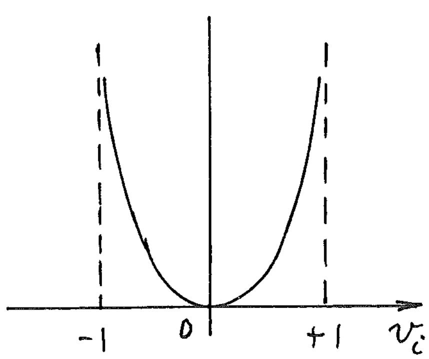
Correspondingly, we may set the bias I_j equal to zero for all j.

The neurons need not have self-loops, i.e., $T_{jj} = 0$.

The energy function of a continuous Hopfield model is then,

$$E = -\frac{1}{2} \sum_{i} \sum_{\substack{j \\ i \neq j}} T_{ij} v_j v_i + \sum_{j} \frac{1}{\tau_i} \int_0^{v_i} g_i^{-1}(v_i)\, dv_i$$

$$= -\frac{1}{2} \sum_{i} \sum_{\substack{j \\ i \neq j}} T_{ij} v_j v_i + \sum_{j} \frac{1}{a_i \tau_i} \int_0^{v_i} g^{-1}(v)\, dv$$



$\int_0^{v_i} g^{-1}(v)\,dv$: very large as $v_i \to \pm 1$

If $a_i \to \infty$, (i.e., the sigmoidal nonlinearity approaches the idealized hard-limiting form), then the 2nd term in E becomes negligibly small. In the limiting case when $a_i = \infty$ for all i, the maxima and minima of the continous Hopfield model become identical with those of the corresponding discrete model.

Mapping of an Optimization Problem

The Hopfield network is now known to minimize the Energy function to arrive at a stable solution. It remains to determine T_{ij} to arbitrarily place equilibrium points anywhere desired in the state space. This requires a mapping of a given optimization problem onto the Hopfield network model.

Economic Load Dispatch:

1. ELD Problem:

$$\text{Min} \quad C = \sum_i (a_i + b_i P_i + c_i P_i^2)$$

where C : total fuel cost

a_i, b_i, c_i : cost coefficient of generator i

P_i : power output of generator i

Constraints:

a) Power balance:

$$D + L = \sum_i P_i$$

where D : total demand

L : transmission loss

$$L = \sum_i \sum_j a_{ij} P_i P_j$$

where a_{ij} : transmission loss coefficients

b) Max & Min of power

$$\underline{P_i} \le P_i \le \overline{P_i}$$

2. Mapping of the ELD into the Hopfield model.
We can augment the constraints to the cost function:

$$E = \frac{A}{2}\left(D + L - \sum_i P_i\right)^2 + \frac{B}{2}\left(\sum_i (a_i + b_i P_i + c_i P_i^2)\right)$$

where $A \ge 0$, $B \ge 0$ are weighting factors.

Assume L is constant. Then

$$E = \frac{A}{2}\left[(D+L)^2 - 2(D+L)\sum_i P_i + \left(\sum_i P_i\right)^2 \right] + \frac{B}{2}\sum_i \left(a_i + b_i P_i + c_i P_i^2 \right)$$

$$= \frac{A}{2}(D+L)^2 - \sum_i \left[A(D+L) + B b_i/2 \right] P_i$$

$$+ \sum_i \sum_i (A + B c_i) P_i P_i/2 + \sum_{i \neq j}\sum A P_i P_j/2 + B \sum_i a_i/2$$

Then, we can define the parameters of Hopfield network as:

$$T_{ij} = -A$$

$$T_{ii} = -A - B c_i$$

$$I_i = A(D+L) - B b_i/2$$

The dynamics of neuron becomes:

$$u_i(k) - u_i(k-1) = \sum T_{ij} V_j(k) + I_i$$

$$V_i(k+1) = g_i\left[u_i(k) \right]$$

The sigmoidal function can be defined as

$$V_i = g(u_i) = (\bar{P}_i - \underline{P}_i)\frac{1}{1 + \exp(-u_i/u_0)} + \underline{P}_i$$

which satisfies the inequality constraint. b).

<u>Steps</u>: 1. Initialize P_i

2. Compute the loss L
3. Compute P_i via Hopfield model
4. Repeat steps 2 & 3.

Optimization with Inequality Constraints

In general, an optimization problem is defined as:

$$\text{Min} \quad f(x), \qquad x \in R^n$$

$$\text{such that} \quad h_j(x) \geq 0, \quad j = 1, \cdots, m$$

$$\underline{x} \leq x \quad \bar{x}$$

Define the Energy function:

$$E = f(x) + \sum_j U(h_j(x)) + \sum_i \frac{1}{R_i} \int_0^{x_i} g^{-1}(x_i)\, dx_i$$

where

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

Then

$$\frac{dU}{dz} = u(z) = \begin{cases} 0 & \text{if } z \geq 0 \\ -1 & \text{if } z < 0 \end{cases}$$

Also,

$$x_i = g(y_i), \quad \text{sigmoidal function}$$

$$y_i = g^{-1}(x_i)$$

Then

$$\frac{dE}{dt} = \sum_i \frac{dx_i}{dt} \underbrace{\left[\frac{\partial f}{\partial x_i} + \sum_j \frac{\partial h_j}{\partial x_i} u(h_j(x)) + \frac{y_i}{R_i} \right]}_{-C_i \frac{dy_i}{dt}}$$

Thus,

$$-C_i \frac{dy_i}{dt} = \frac{\partial f}{\partial x_i} + \sum_j \frac{\partial h_j}{dx_i} u(h_j(x)) + \frac{y_i}{R_i}$$

The <u>steady-state solution</u> can be obtained by

$$-\frac{y_i}{R_i} = \frac{\partial f}{\partial x_i} + \sum_j \frac{\partial h_j}{\partial x_i} u(h_j(x)) \triangleq H(x)$$

The steady-state solution can be solved by the Gauss-Seidel iteration:

$$H(x^0) \longrightarrow y_i \xrightarrow{g} x_i$$

$$x_i = g(y_i) \rightarrow \quad \underline{x}_i \le x_i \le \bar{x}_i$$

$$\Rightarrow \quad x_i = \underline{x}_i + \frac{(\bar{x}_i - \underline{x}_i)}{1 + e^{-y_i}}$$