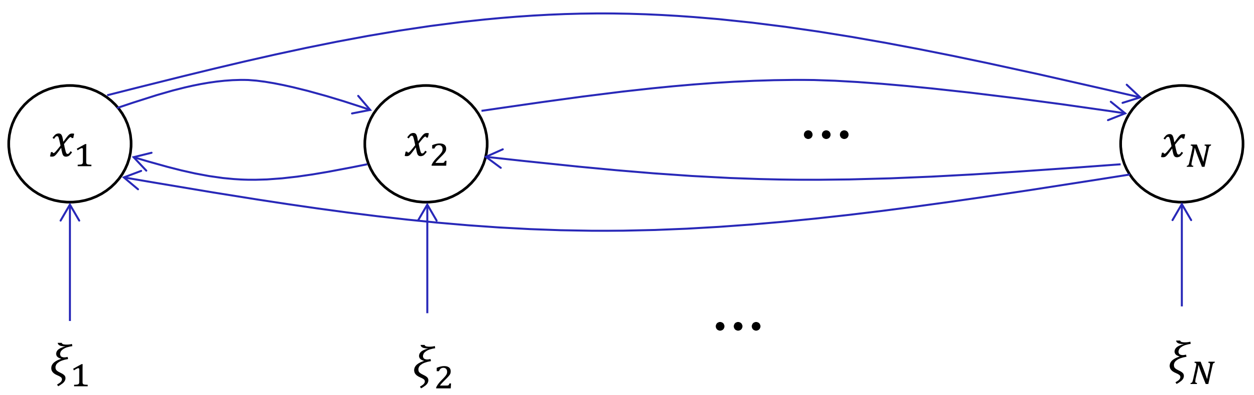
Hopfield Network for Associative Memory

The basic task

- Store a set of fundamental memories $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_M\}$ so that, when presented a new pattern **x**, the system outputs one of the stored memories that is most similar to **x**
  - Such a system is called content-addressable memory

Architecture

- The Hopfield net consists of *N* McCulloch-Pitts neurons, recurrently connected among themselves



Definition

- Each neuron is defined as

$$x_j = \varphi(v_j)$$

where $v_j = \sum_{i=1}^{N} w_{ji} x_i + b_j$

and $\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ -1 & \text{otherwise} \end{cases}$

  - Without loss of generality, let $b_j = 0$

Storage phase

- To store fundamental memories, the Hopfield model uses the outer-product rule, a form of Hebbian learning:

$$w_{ji} = \frac{1}{N} \sum_{\mu=1}^{M} \xi_{\mu,j}\, \xi_{\mu,i}$$

- Hence $w_{ji} = w_{ij}$, i.e., $\mathbf{w} = \mathbf{w}^T$, so the weight matrix is symmetric

Retrieval phase

- The Hopfield model sets the initial state of the net to the input pattern. It adopts asynchronous (serial) update, which updates one neuron at a time

Hamming distance

- Hamming distance between two binary/bipolar patterns is the number of differing bits
  - Example (see blackboard)

One memory case

- Let the input **x** be the same as the single memory $\boldsymbol{\xi}$

$$
\begin{aligned}
x_j &= \varphi\left(\sum_i w_{ji}\, x_i\right) \\
&= \varphi\left(\frac{1}{N}\sum_i \xi_j \xi_i \xi_i\right) \\
&= \varphi(\xi_j) \\
&= \xi_j
\end{aligned}
$$

Therefore the memory is stable

One memory case (cont.)

- Actually for any input pattern, as long as the Hamming distance between $\mathbf{x}$ and $\boldsymbol{\xi}$ is less than $N/2$, the net converges to $\boldsymbol{\xi}$. Otherwise it converges to $-\boldsymbol{\xi}$
  - Think about it

Multiple memories

- The stability (alignment) condition for any memory $\boldsymbol{\xi}_\vartheta$ is

$$\varphi(v_j) = \xi_{\vartheta,j}$$

where

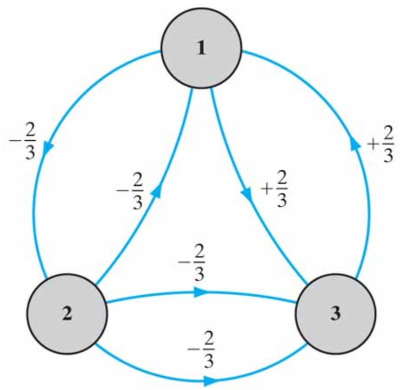
$$v_j = \sum_i w_{ji}\xi_{\vartheta,i} = \frac{1}{N}\sum_i \sum_\mu \xi_{\mu,j}\xi_{\mu,i}\xi_{\vartheta,i}$$

$$= \xi_{\vartheta,j} + \underbrace{\frac{1}{N}\sum_i \sum_{\mu\neq\vartheta} \xi_{\mu,j}\xi_{\mu,i}\xi_{\vartheta,i}}_{\text{crosstalk}}$$

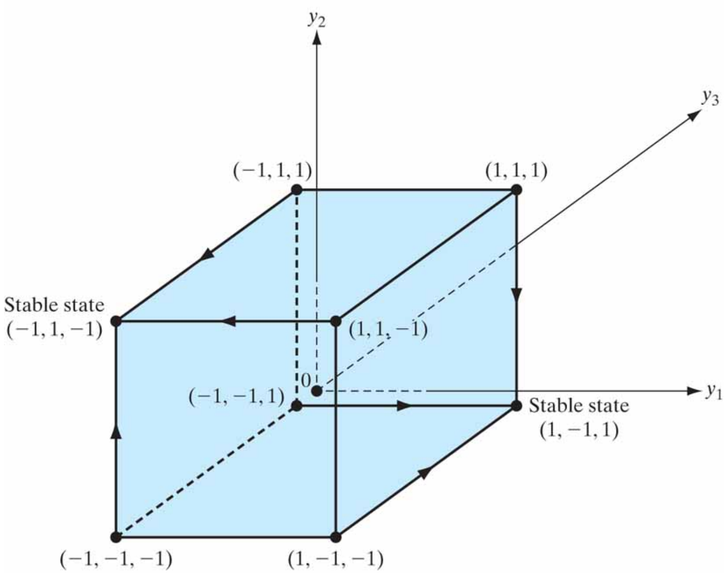
Multiple memories (cont.)

- If |crosstalk| $< N$, the memory system is stable. In general, fewer memories are more likely stable
- Example 2 in book (see blackboard)

Example (cont.)



(a)



(b)

Memory capacity

- Define

$$C_j^{\vartheta} = -\xi_{\vartheta,j} \sum_i \sum_{\mu \neq \vartheta} \xi_{\mu,j} \xi_{\mu,i} \xi_{\vartheta,i}$$

$$C_j^{\vartheta} < 0 \quad \Rightarrow \quad \text{stable}$$
$$0 \le C_j^{\vartheta} < N \quad \Rightarrow \quad \text{stable}$$
$$C_j^{\vartheta} > N \quad \Rightarrow \quad \text{unstable}$$

- What if $C_j^{\vartheta} = N$?

Memory capacity (cont.)

- Consider random memories where each element takes +1 or -1 with equal probability (prob.). We measure

$$P_{\text{error}} = \text{Prob}(C_j^{\vartheta} > N)$$

- To compute capacity $M_{\max}$, decide on an error criterion
- For random patterns, C_j^{ϑ} is proportional to a sum of $N(M-1)$ random numbers of +1 or -1. For large NM, it can be approximated by a Gaussian distribution (central limit theorem) with zero mean and variance $\sigma^2 = NM$

Memory capacity (cont.)

- So

$$P_{\text{error}} = \frac{1}{\sqrt{2\pi}\sigma}\int_N^{\infty} \exp(-\frac{x^2}{2\sigma^2})\,\mathrm{d}x$$

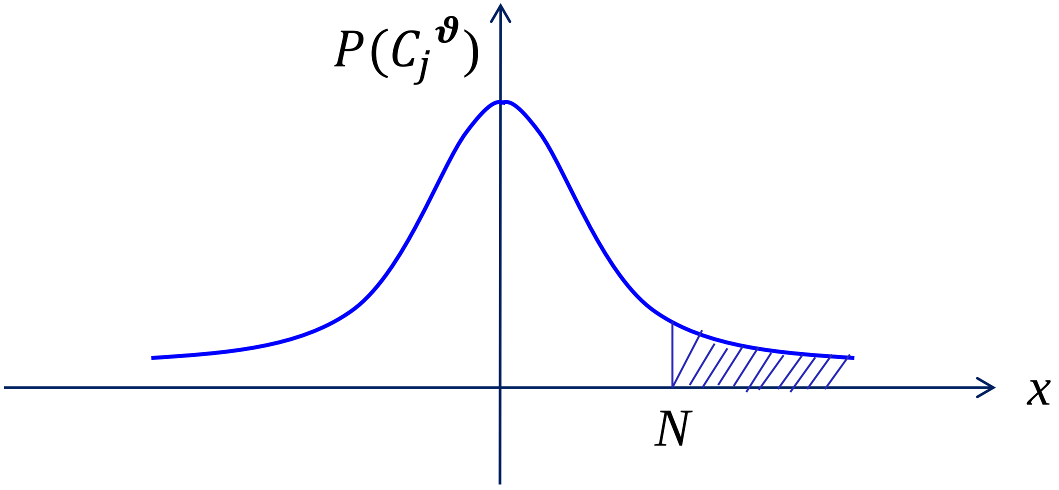
$$= \frac{1}{2} - \frac{1}{\sqrt{2\pi}\sigma}\int_0^{N} \exp\left(-\frac{x^2}{2\sigma^2}\right)\mathrm{d}x$$

$(x = \sqrt{2}\sigma\mu)$

$$= \frac{1}{2}\left(1 - \underbrace{\frac{2}{\sqrt{\pi}}\int_0^{\sqrt{N/(2M)}} \exp(-\mu^2)\,\mathrm{d}\mu}_{\text{error function}}\right)$$

Memory capacity (cont.)

- Error probability plot



Memory capacity (cont.)

P_{error}	M_{max}/N
0.001	0.105
0.0036	0.138
0.01	0.185
0.05	0.37
0.1	0.61

- So $P_{\text{error}} < 0.01 \Rightarrow M_{\text{max}} = 0.185N$, an upper bound

Memory capacity (cont.)

- What if 1% flips occur? Avalanche effect?
- Real upper bound: $0.138N$ to prevent the avalanche effect

Memory capacity (cont.)

- The above analysis is for one bit. If we want perfect retrieval for $\boldsymbol{\xi}^{\vartheta}$ with probability of 0.99:

$$(1 - P_{\text{error}})^{N} > 0.99$$

  - Approximately $P_{\text{error}} < \frac{0.01}{N}$

- For this case

$$M_{\text{max}} = \frac{N}{2\log N}$$

Memory capacity (cont.)

- But real patterns are not random (they could be encoded) and the capacity is worse for correlated patterns

- At one favorable extreme, if memories are orthogonal

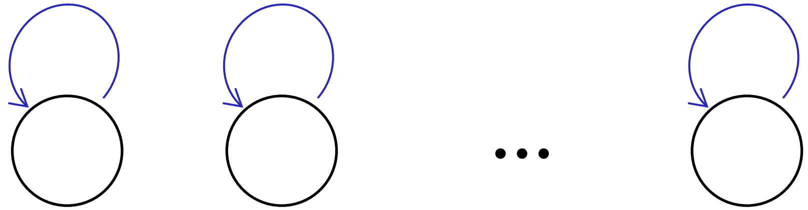
$$\sum_i \xi_{\mu,i}\xi_{\vartheta,i} = 0 \quad \text{for } \vartheta \neq \mu$$

then ${C_j}^{\vartheta} = 0$ and $M_{\text{max}} = N$

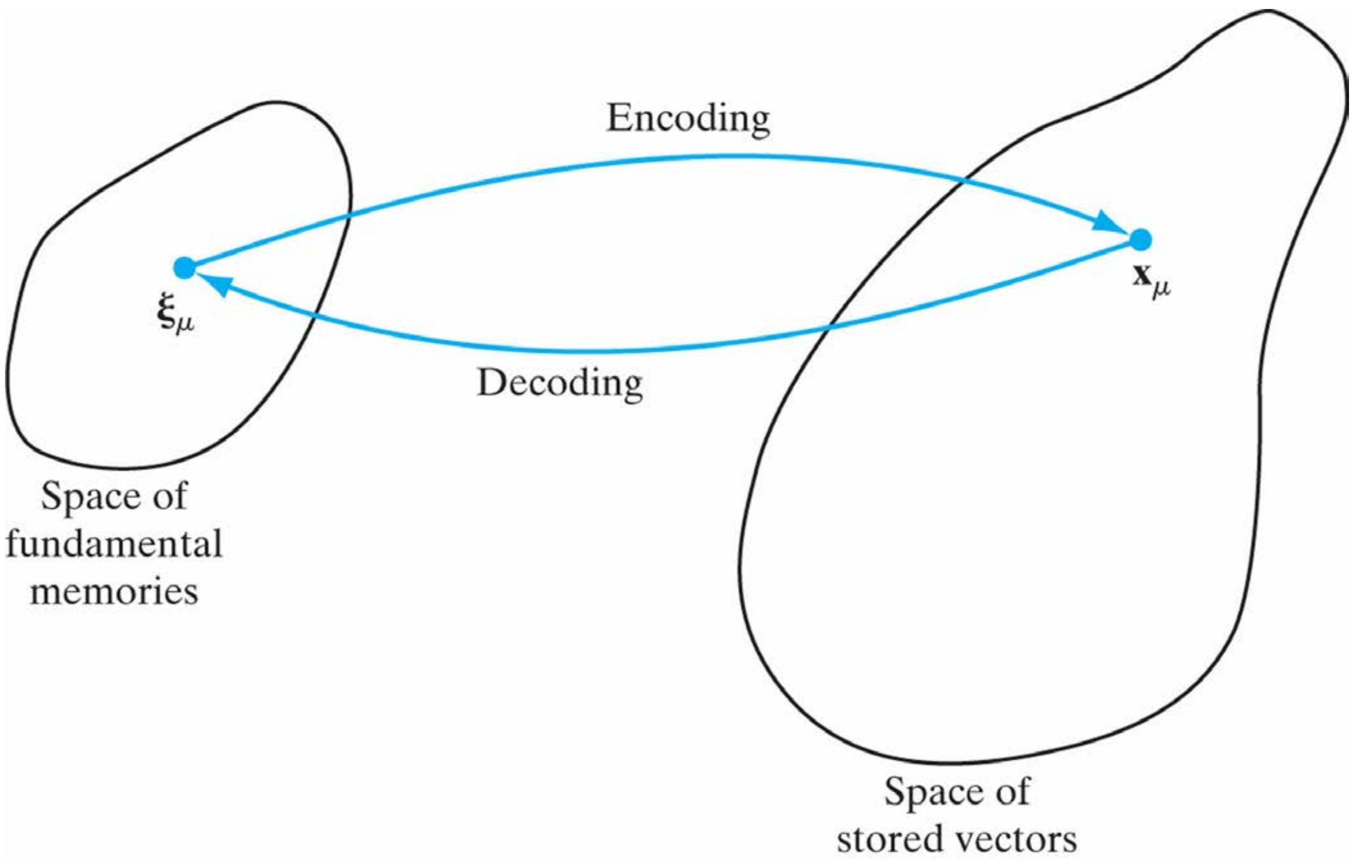
  - This is the maximum number of orthogonal patterns

Memory capacity (cont.)

- But in reality a useful system stores a little less; otherwise the memory is useless as it does not evolve



Coding illustration



Energy function (Lyapunov function)

- The existence of an energy (Lyapunov) function for a dynamical system ensures its stability
- The energy function for the Hopfield net is

$$E(\mathbf{x}) = -\frac{1}{2}\sum_{i}\sum_{j} w_{ji} x_i x_j$$

- **Theorem**: Given symmetric weights, $w_{ji} = w_{ij}$, the energy function does not increase as the Hopfield net evolves

Energy function (cont.)

- Let ${x_j}'$ be the new value of x_j after an update

$$
{x_j}' = \varphi\left(\sum_i w_{ji} x_i\right)
$$

- If $x_j' = x_j$, E remains the same

Energy function (cont.)

- In the other case, $x_j' = -x_j$:

$$E(x_j') - E(x_j) = -\frac{1}{2}\sum_i \sum_j w_{ji} x_i x_j' + \frac{1}{2}\sum_i \sum_j w_{ji} x_i x_j$$

since $w_{jj} = {}^{M}/_{N}$ $$= -\frac{1}{2}\sum_{i \neq j} \sum_j w_{ji} x_i x_j' + \frac{1}{2}\sum_{i \neq j} \sum_j w_{ji} x_i x_j$$

since $w_{ij} = w_{ji}$ $$= -x_j' \sum_{i \neq j} w_{ji} x_i + x_j \sum_{i \neq j} w_{ji} x_i$$

$$= 2x_j \sum_{i \neq j} w_{ji} x_i$$

$$= 2x_j \sum_i w_{ji} x_i - 2w_{jj} < 0$$

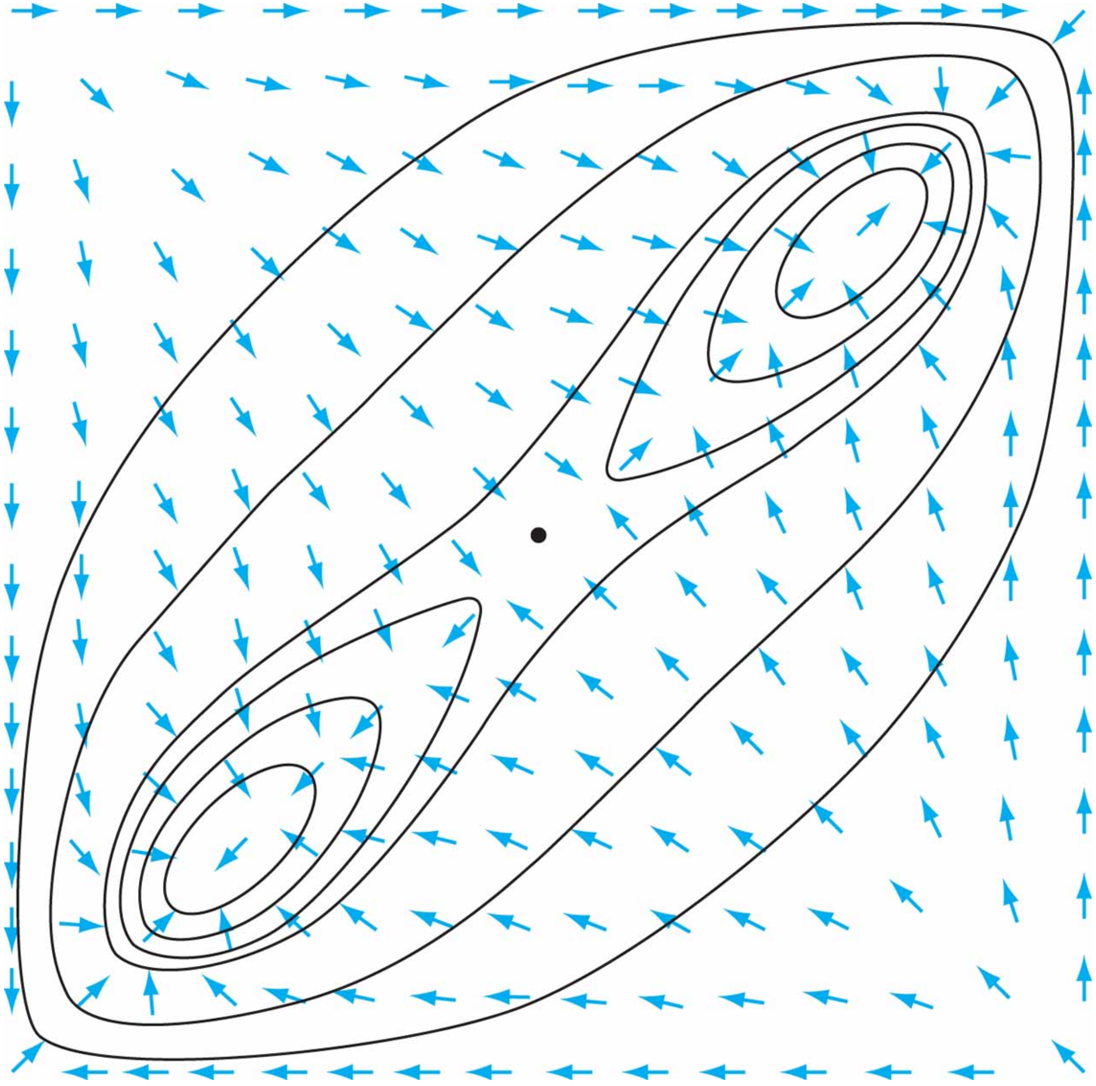
different signs

Energy function (cont.)

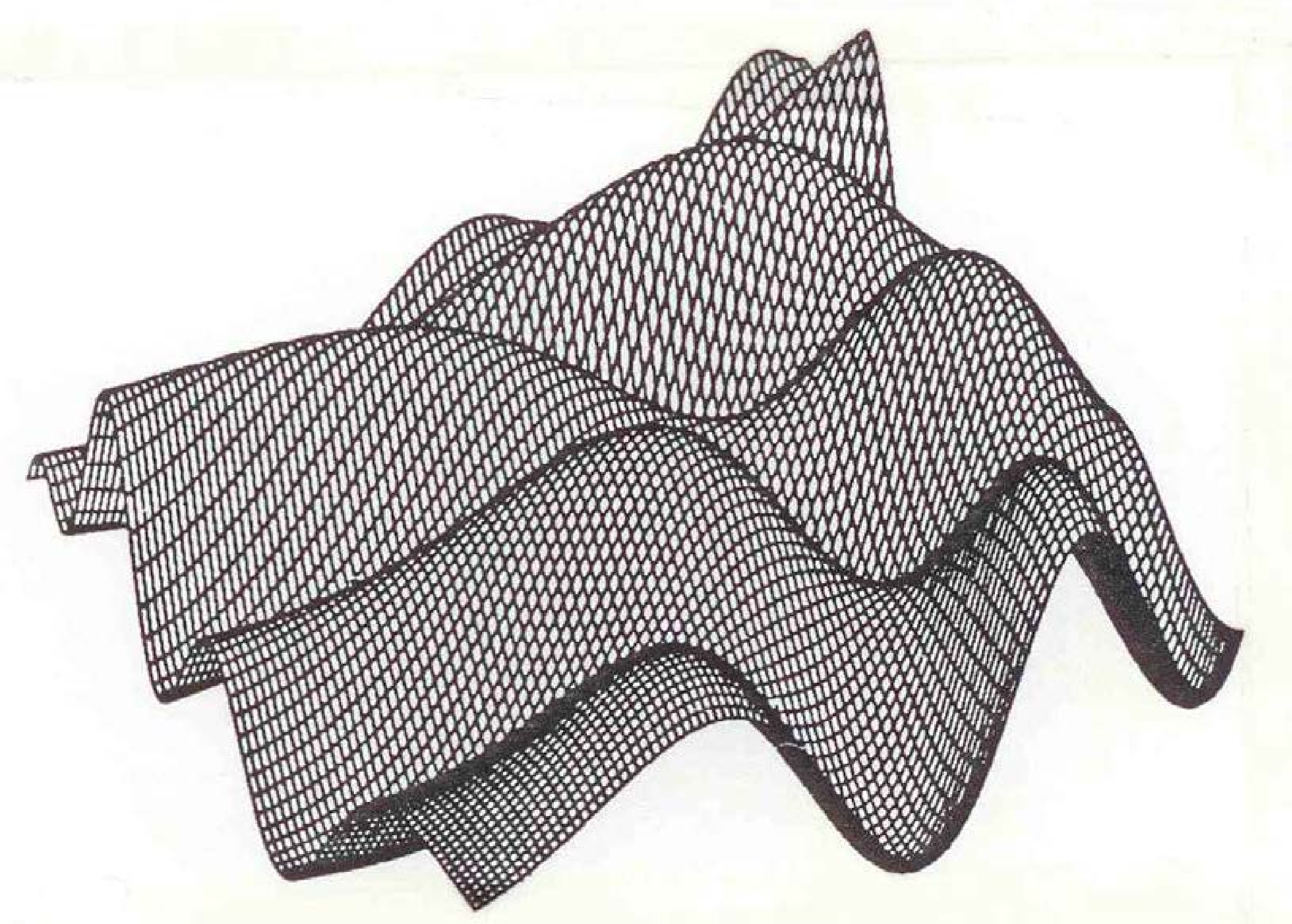
- Thus, $E(\mathbf{x})$ decreases every time x_j flips. Since E is bounded, the Hopfield net is always stable

- **Remarks**:
  - Useful concepts from dynamical systems: attractors, basins of attraction, energy (Lyapunov) surface or landscape

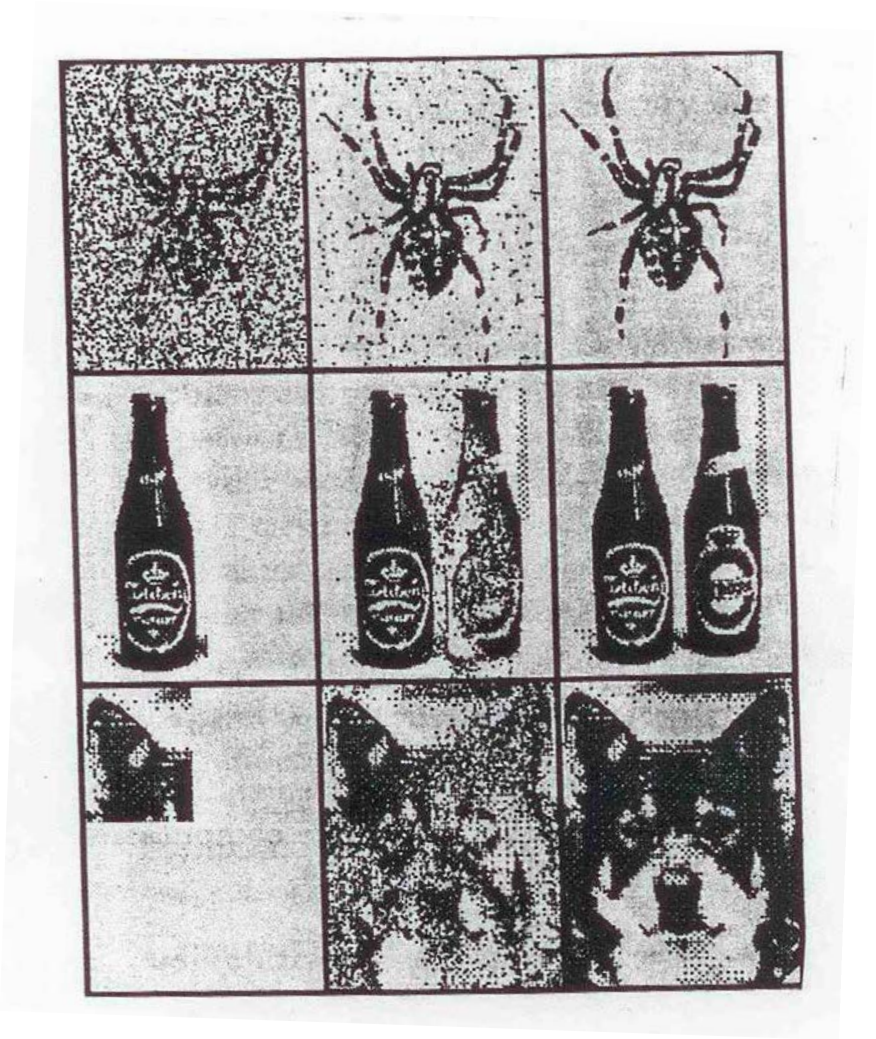
Energy contour map



2-D energy landscape



Memory recall illustration



Remarks (cont.)

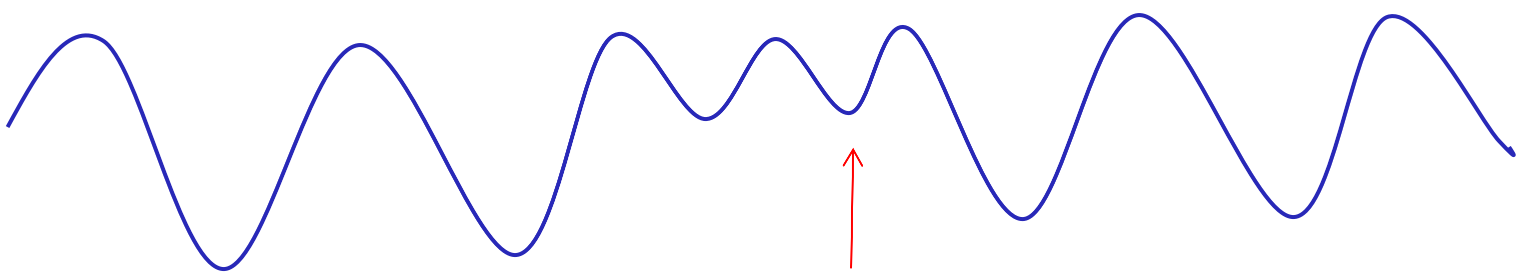
- Bipolar neurons can be extended to continuous-valued neurons by using hyperbolic tangent activation function, and discrete update can be extended to continuous-time dynamics (good for analog VLSI implementation)
- The concept of energy minimization has been applied to optimization problems (neural optimization)

Spurious states

- Not all local minima (stable states) correspond to fundamental memories. Typically, $-\boldsymbol{\xi}_\mu$, linear combination of odd number of memories, or other uncorrelated patterns, can be attractors
  - Such attractors are called spurious states

Spurious states (cont.)

- Spurious states tend to have smaller basins and occur higher on the energy surface

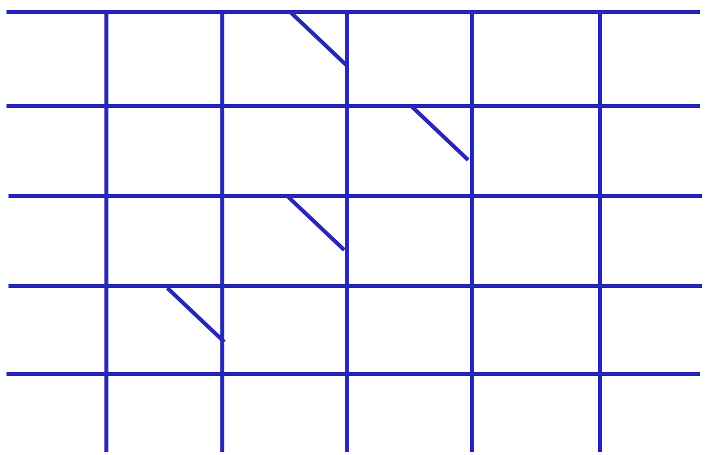


local minima
for spurious states

Kinds of associative memory

- $\begin{cases} \text{Autoassociative} \quad \text{(e.g. Hopfiled net)} \\ \\ \text{Heteroassociative: store pairs of } < \mathbf{x}_\mu, \mathbf{y}_\mu > \text{ explicitly} \end{cases}$

x



y

matrix memory

holographic memory