CSE 5526: Introduction to Neural Networks

Hopfield Network for Associative Memory

The next few units cover unsupervised models

- Goal: learn the distribution of a set of observations
- Some observations are a better “fit” than others
- Hopfield networks store a set of observations
  - Deterministic, non-linear dynamical system
- Boltzmann machines can behave similarly
  - Stochastic, non-linear dynamical system
- Boltzmann machines with hidden units have a much greater capacity for learning the data distribution

Content-addressable memory basic task

- Store a set of "fundamental memories" $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_M\}$
- So that when presented with a new pattern **x**
- The system outputs the stored memory that is most similar to **x**
- The first content-addressable memory we will consider is the Hopfield network
  - Introduced in the influential (14,000 citations) paper Hopfield (1982). "Neural networks and physical systems with emergent collective computational abilities." PNAS.

Is this possible? How good can it be?

- Is this possible to implement as a neural network?
  - For a single pattern?
- Does it work equally well for any pattern?
- How many patterns can such a system store?
  - How do its storage requirements compare to other sys's?
- How much corruption can it tolerate?
  - And still retrieve the correct pattern?
  - Corruption of noise or of partial information

Hopfield (1982) describes the problem

- “Any physical system whose dynamics in phase space is dominated by a substantial number of locally stable states to which it is attracted can therefore be regarded as a general content-addressable memory. The physical system will be a potentially **useful** memory if, in addition, **any** prescribed set of states can readily be made the stable states of the system.”

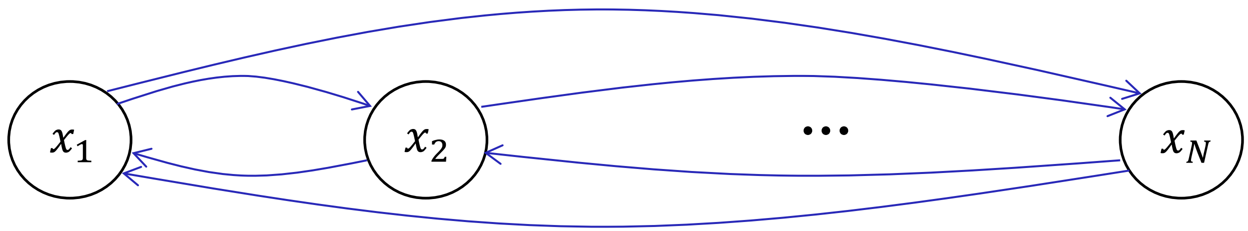
One associative memory: the Hopfield network

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

- The network is initialized with a (corrupted) pattern

One associative memory: the Hopfield network

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



- Then runs recurrently until it reaches a fixed point

State of each neuron defines the "state space"

- The network is in state $\boldsymbol{x}_t$ at time t
- The state of the network evolves according to
$$\boldsymbol{x}_{t+1} = \varphi(W\boldsymbol{x}_t + \boldsymbol{b})$$
  - Where we set $\boldsymbol{b} = 0$ without loss of generality
  - Meaning that each state leads to at most one next state
- $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_t\}$ is called a state trajectory
- Goal: set W so that state trajectory of corrupted memory $\boldsymbol{\xi}_i + \Delta$ converges to true memory $\boldsymbol{\xi}_i$

One-shot storage phase uses Hebbian learning

- Hopfield nets set W using the outer-product rule, one choice for doing so.

$$W = \frac{1}{N}\sum_{\mu=1}^{M} \boldsymbol{\xi}_\mu \boldsymbol{\xi}_\mu^T - I$$

Where N is the number of bits. Or, equivalently

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

- The $-\mathrm{I}$ and $-\delta_{ij}$ terms enforce $W_{ii} = 0$
  - no self-feedback

Hebbian learning

- “Neurons that fire together, wire together”
- In the Hopfield network, increase the weights of neurons that receive correlated inputs
- This notion is symmetric between neurons
  - And since $w_{ji} = w_{ij}$, the weight matrix is symmetric

Retrieval phase

- Play out dynamics $\boldsymbol{x}_{t+1} = \varphi(W\boldsymbol{x}_t)$
  - Until reaching a stable state $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$
  - If argument to $\varphi(\cdot)$ is 0, neuron stays in previous state
    - Leads to symmetric flow diagrams
- Can also use “asynchronous” updates
  - Pick one neuron at random
  - Update it based on the others
  - Repeat

With one memory, that memory is stable

- Let the input $\mathbf{x}_0$ be the same as the single memory $\boldsymbol{\xi}$

$$\boldsymbol{x}_1 = \varphi(W\boldsymbol{x}_0) = \varphi\left(\frac{1}{N}(\boldsymbol{\xi}\boldsymbol{\xi}^T - I)\boldsymbol{\xi}\right)$$

$$= \varphi\left(\frac{1}{N}\boldsymbol{\xi}(\boldsymbol{\xi}^T\boldsymbol{\xi} - 1)\right)$$

$$= \varphi\left(\frac{\|\boldsymbol{\xi}\|^2 - 1}{N}\boldsymbol{\xi}\right)$$

$$= \varphi\left(\frac{N-1}{N}\boldsymbol{\xi}\right) = \boldsymbol{\xi}$$

Therefore the memory is stable

Aside: Hamming distance is the number of differing bits between two patterns

- Hamming distance of 1 from $\{+1, +1, +1\}$
  - $\{+1, +1, -1\}$, $\{+1, -1, +1\}$, $\{-1, +1, +1\}$
- Hamming distance of 2 from $\{+1, +1, +1\}$
  - $\{+1, -1, -1\}$, $\{-1, +1, -1\}$, $\{-1, -1, +1\}$
- Hamming distance of 3 from $\{+1, +1, +1\}$
  - $\{-1, -1, -1\}$
- For $x_1, x_2 \in \{\pm 1\}^N$, $x_1^T x_2 = N - 2 d_H(x_1, x_2)$
  - So $-N \leq x_1^T x_2 \leq N$

With one memory, Hopfield net converges to the closer of $\boldsymbol{\xi}$ or $-\boldsymbol{\xi}$

- For input of $\boldsymbol{x}_0$

$$\boldsymbol{x}_1 = \varphi\left(\frac{1}{N} W \boldsymbol{x}_0\right) = \varphi\left(\frac{1}{N}(\boldsymbol{\xi}\boldsymbol{\xi}^T - I)\boldsymbol{x}_0\right)$$

$$= \varphi\left(\frac{1}{N}(\boldsymbol{\xi}\boldsymbol{\xi}^T\boldsymbol{x}_0 - \boldsymbol{x}_0)\right)$$

$$= \pm\boldsymbol{\xi}$$

  - Assuming that $|\boldsymbol{\xi}^T\boldsymbol{x}_0| > 1$
- Closer is measured by inner product
  - Or equivalently in this case, by Hamming distance

Example: Hopfield net with one memory

- Let's use $\boldsymbol{\xi} = [-1, +1, -1]^T$, then

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

- Test memory stability

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

  - So $\varphi(W\boldsymbol{\xi}) = \boldsymbol{\xi}$

Example: Hopfield net with one memory

- Follow state trajectory from $\boldsymbol{x}_1 = [-1, -1, +1]^T$

$$W\boldsymbol{x}_1 = \frac{1}{3}\begin{bmatrix} 0 & -1 & +1 \\ -1 & 0 & -1 \\ +1 & -1 & 0 \end{bmatrix}\begin{bmatrix} -1 \\ -1 \\ +1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$

  - So $\varphi(W\boldsymbol{x}_1) = [+1, -1, +1]^T = -\boldsymbol{\xi}$

- Follow state trajectory from $\boldsymbol{x}_2 = [+1, +1, -1]^T$

$$W\boldsymbol{x}_2 = \frac{1}{3}\begin{bmatrix} 0 & -1 & +1 \\ -1 & 0 & -1 \\ +1 & -1 & 0 \end{bmatrix}\begin{bmatrix} +1 \\ +1 \\ -1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

  - So $\varphi(W\boldsymbol{x}_2) = [-1, +1, -1]^T = \boldsymbol{\xi}$

For a Hopfield net with multiple memories

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

$$\boldsymbol{\xi}_\vartheta = \varphi(W\boldsymbol{\xi}_\vartheta)$$

$$= \varphi\left(\left(\frac{1}{N}\sum_{\mu}\boldsymbol{\xi}_\mu\boldsymbol{\xi}_\mu^T - I\right)\boldsymbol{\xi}_\vartheta\right)$$

$$= \varphi\left(\frac{N-1}{N}\boldsymbol{\xi}_\vartheta + \underbrace{\frac{1}{N}\sum_{\mu\neq\vartheta}\boldsymbol{\xi}_\mu\left(\boldsymbol{\xi}_\mu^T\boldsymbol{\xi}_\vartheta - 1\right)}_{\text{crosstalk}}\right)$$

Multiple memories can be stored if $M \ll N$

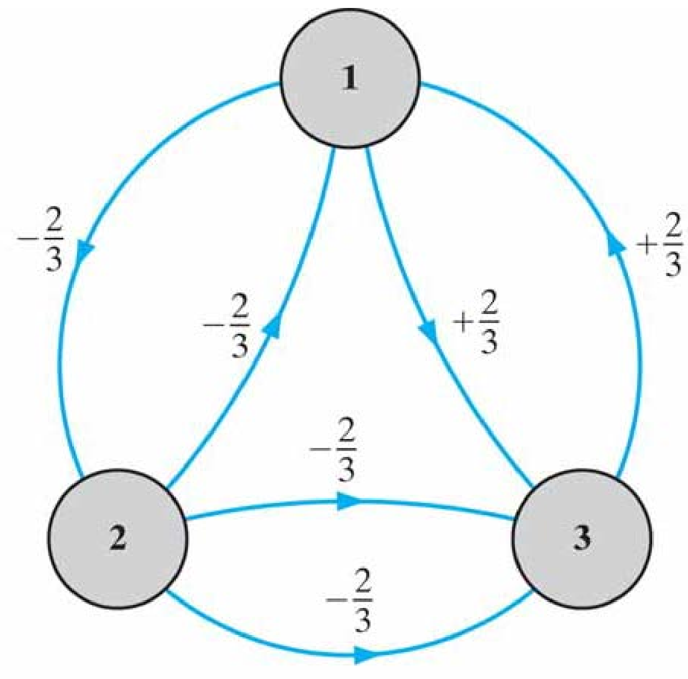
- Crosstalk is a weighted sum of the memories
- If memories are random variables (i.e., uncorrelated with each other)
  - Then this is a sum of $N(M-1)$ random ± 1 variables
  - Which is asymptotically Gaussian
- If the crosstalk is small, compared to the $\boldsymbol{\xi}_{\vartheta}$ term
  - Then the memory system is stable
  - In general, fewer memories are more likely stable
- More on this shortly

Example 2 from textbook

- Consider the Hopfield network with

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

- 8 possible states
  - See where each goes



(a)

Two states are stable

- Two states are stable
  - $\boldsymbol{\xi}_1 = [+1, -1, +1]^T$ and $\boldsymbol{\xi}_2 = [-1, +1, -1]^T = -\boldsymbol{\xi}_1$

$$W\boldsymbol{\xi}_1 = \frac{1}{3}\begin{bmatrix} 0 & -2 & +2 \\ -2 & 0 & -2 \\ +2 & -2 & 0 \end{bmatrix}\begin{bmatrix} +1 \\ -1 \\ +1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} +4 \\ -4 \\ +4 \end{bmatrix}$$

  - So $\varphi(W\boldsymbol{\xi}_1) = \boldsymbol{\xi}_1$

$$W\boldsymbol{\xi}_2 = \frac{1}{3}\begin{bmatrix} 0 & -2 & +2 \\ -2 & 0 & -2 \\ +2 & -2 & 0 \end{bmatrix}\begin{bmatrix} -1 \\ +1 \\ -1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} -4 \\ +4 \\ -4 \end{bmatrix}$$

  - So $\varphi(W\boldsymbol{\xi}_2) = \boldsymbol{\xi}_2$

Weight matrix agrees with the one calculated from the two stable states

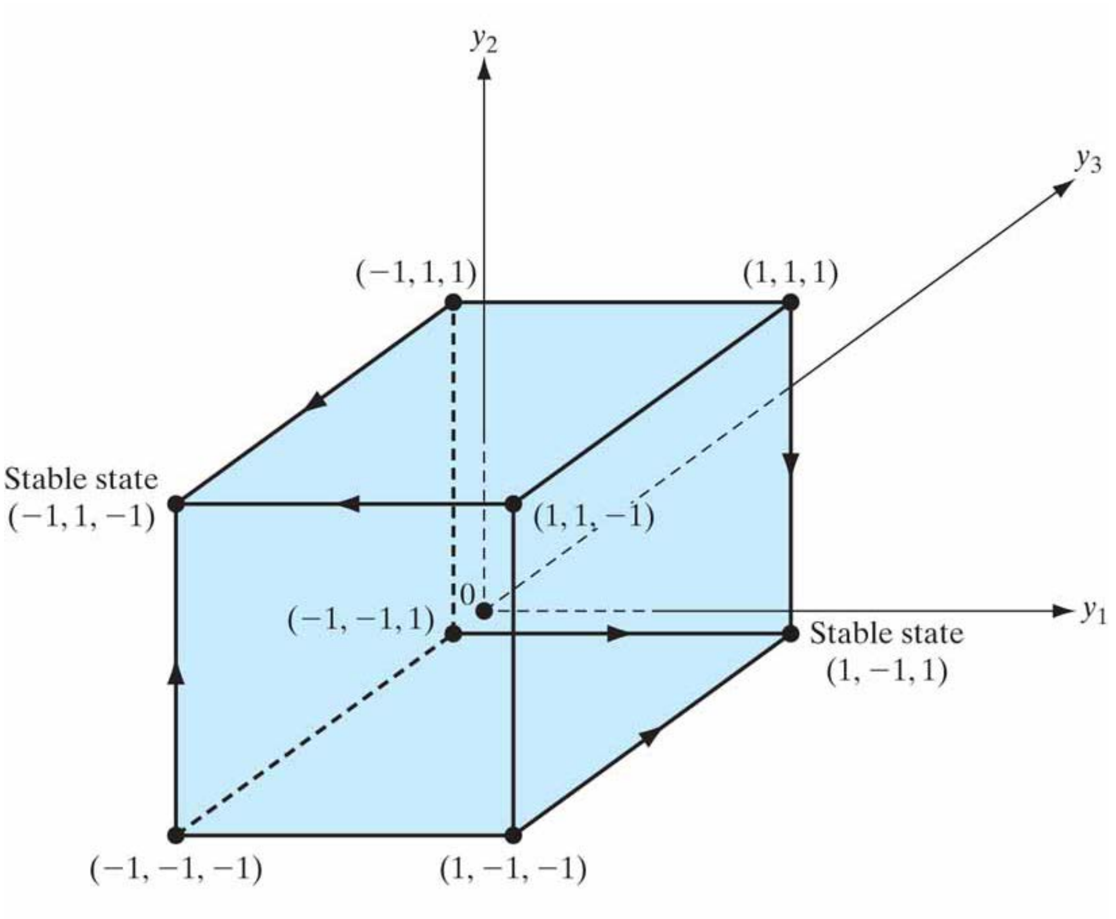
$$W = \frac{1}{3}\xi_1\xi_1^T + \frac{1}{3}\xi_2\xi_2^T - \frac{2}{3}I$$

$$= \frac{1}{3}\begin{bmatrix}+1\\-1\\+1\end{bmatrix}[+1,-1,+1] + \frac{1}{3}\begin{bmatrix}-1\\+1\\-1\end{bmatrix}[-1,+1,-1] - \frac{2}{3}I$$

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

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

Asynchronous updates follow this flow diagram



Memory capacity for a single bit: Prob of error is defined by amount of cross-talk

- Define

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

- Amount cross-talk pushes bit *j* in the wrong direction

$$C_j^{\vartheta} < 0 \quad \Rightarrow \quad \text{stable}$$

$$0 \leq C_j^{\vartheta} < N \quad \Rightarrow \quad \text{stable}$$

$$C_j^{\vartheta} > N \quad \Rightarrow \quad \text{unstable}$$

Capacity: Crosstalk is approximately Gaussian

- Consider random memories where each element takes $+1$ or -1 with equal probability.
- For random patterns, ${C_j}^{\vartheta}$ 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 $\sigma^2 = NM$
- Capacity M_{max} is defined by an error criterion
  - Acceptable level of $P_{\text{error}} = \text{Prob}({C_j}^{\vartheta} > N)$

Capacity: Prob of error is a function of N/M

- So

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

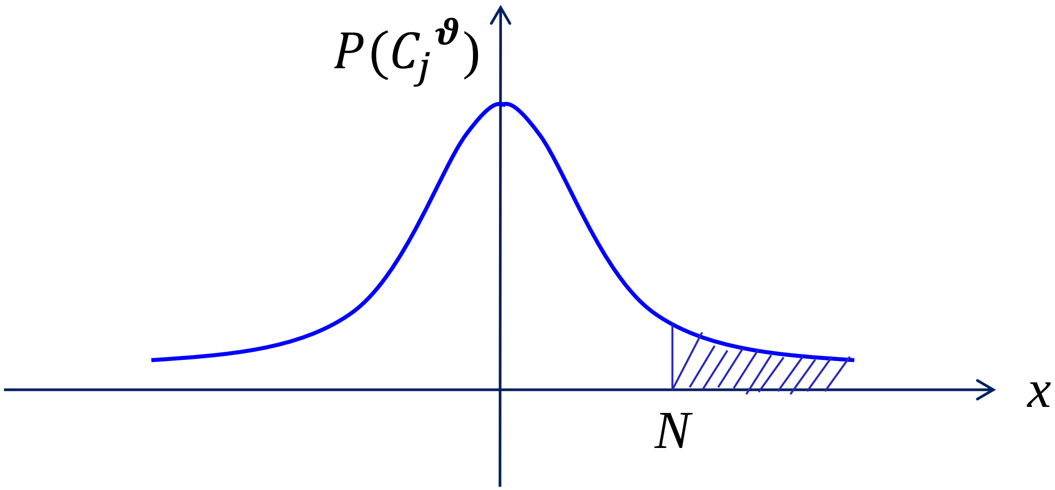
$$= \frac{1}{2} - \frac{1}{\sqrt{2\pi}\sigma}\int_0^N \exp\left(-\frac{x^2}{2\sigma^2}\right)\mathrm{d}x$$

define $\mu = \dfrac{x}{\sigma\sqrt{2}}$

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

Capacity: Visualizing prob of error

- So $P_{\text{error}} = \frac{1}{2}\left(1 - \text{erf}\left(\sqrt{\frac{N}{2M}}\right)\right)$



Capacity: Lower error prob requires smaller M

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 $P_{\text{error}} < 0.01 \Rightarrow M_{\max} = 0.185N$, an upper bound
- Or $0.138N$ just to be safe

To get all *N* bits correct requires smaller *M*

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

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

  - Approximately $P_{\text{error}} < \frac{0.01}{N}$
- For this case $M_{\max} = \frac{N}{2\log N}$
  - See (McEliece, Posner, Rodemich, and Venkatesh, 1987)
- This is a bit disappointing compared to various error correction codes

Non-random memories modify capacity

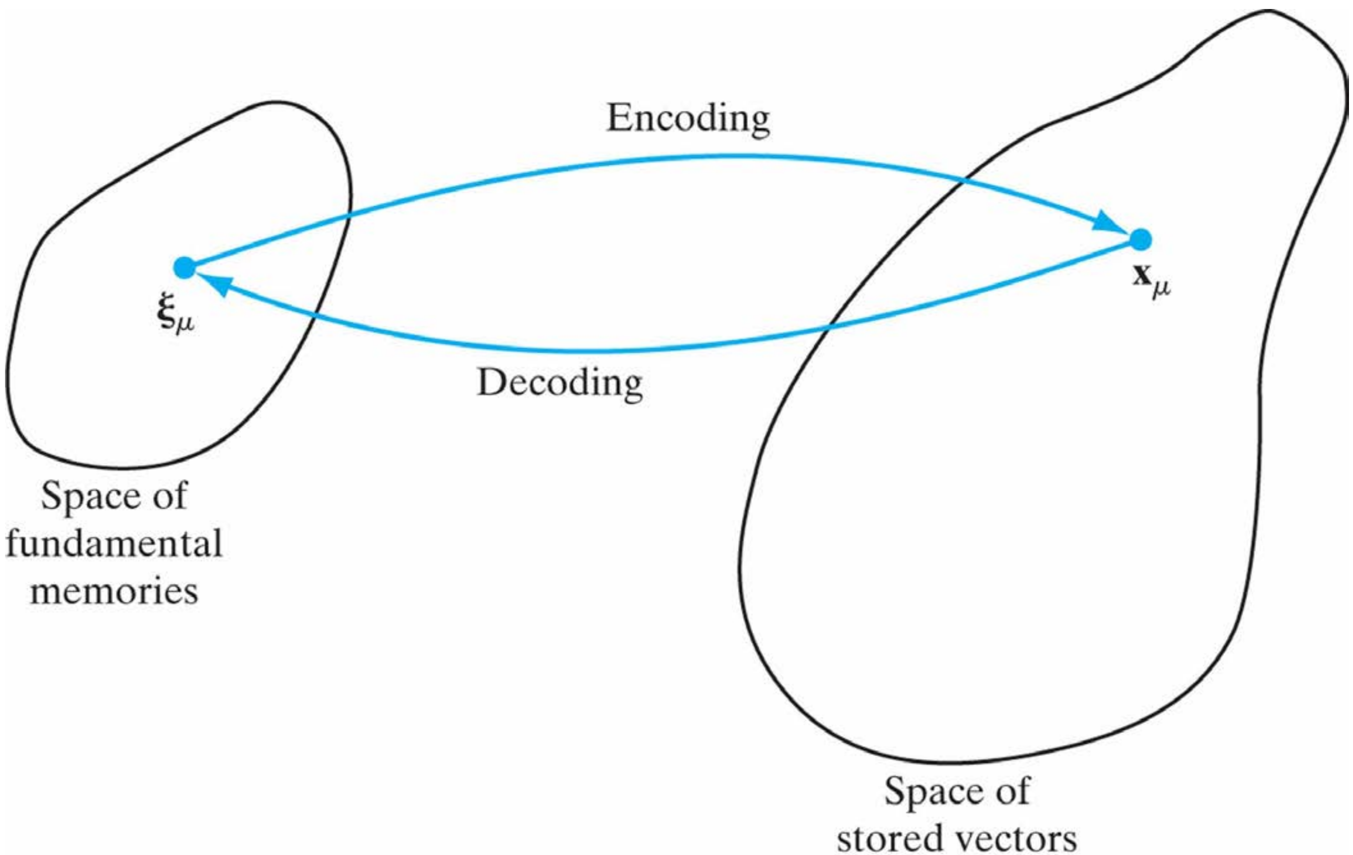
- Real patterns are not random
  - Although they could be encoded to be almost random
  - The capacity is worse for correlated patterns
- At the favorable extreme, for orthogonal memories

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

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

- This is the maximum number of orthogonal patterns
- Use fewer memories to allow some evolution, otherwise, why bother?

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

$$E(\mathbf{x}) = -\frac{1}{2}\sum_i \sum_j w_{ji} x_i x_j = -\frac{1}{2}\boldsymbol{x}^T W \boldsymbol{x}$$

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

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.)

- Otherwise, $x_j' = -x_j$:
  - Let s be a vector of 1s except for $s_j = -1$

$$E(x') - E(x) = -\frac{1}{2}\sum_i \sum_k w_{ki} x_i x_k s_i s_k + \frac{1}{2}\sum_i \sum_k w_{ki} x_i x_k$$

$$= -\sum_{i \neq j} w_{ji} x_i x_j s_j + \sum_{k \neq j} w_{kj} x_j x_k s_j$$

since $W = W^T$

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

$$= 2 x_j \sum_{i \neq j} w_{ji} x_i < 0$$

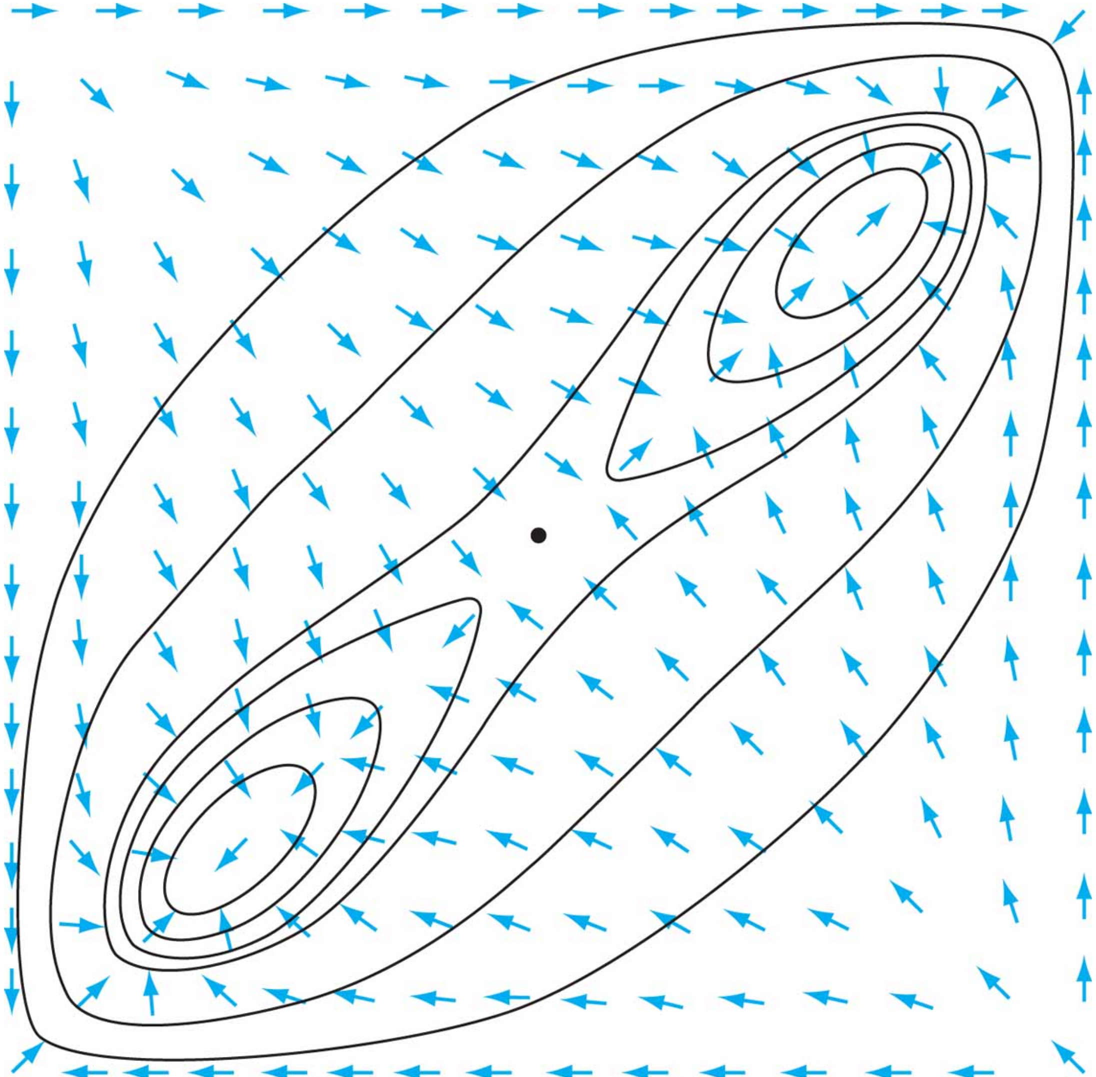
different signs
by assumption

Energy function (cont.)

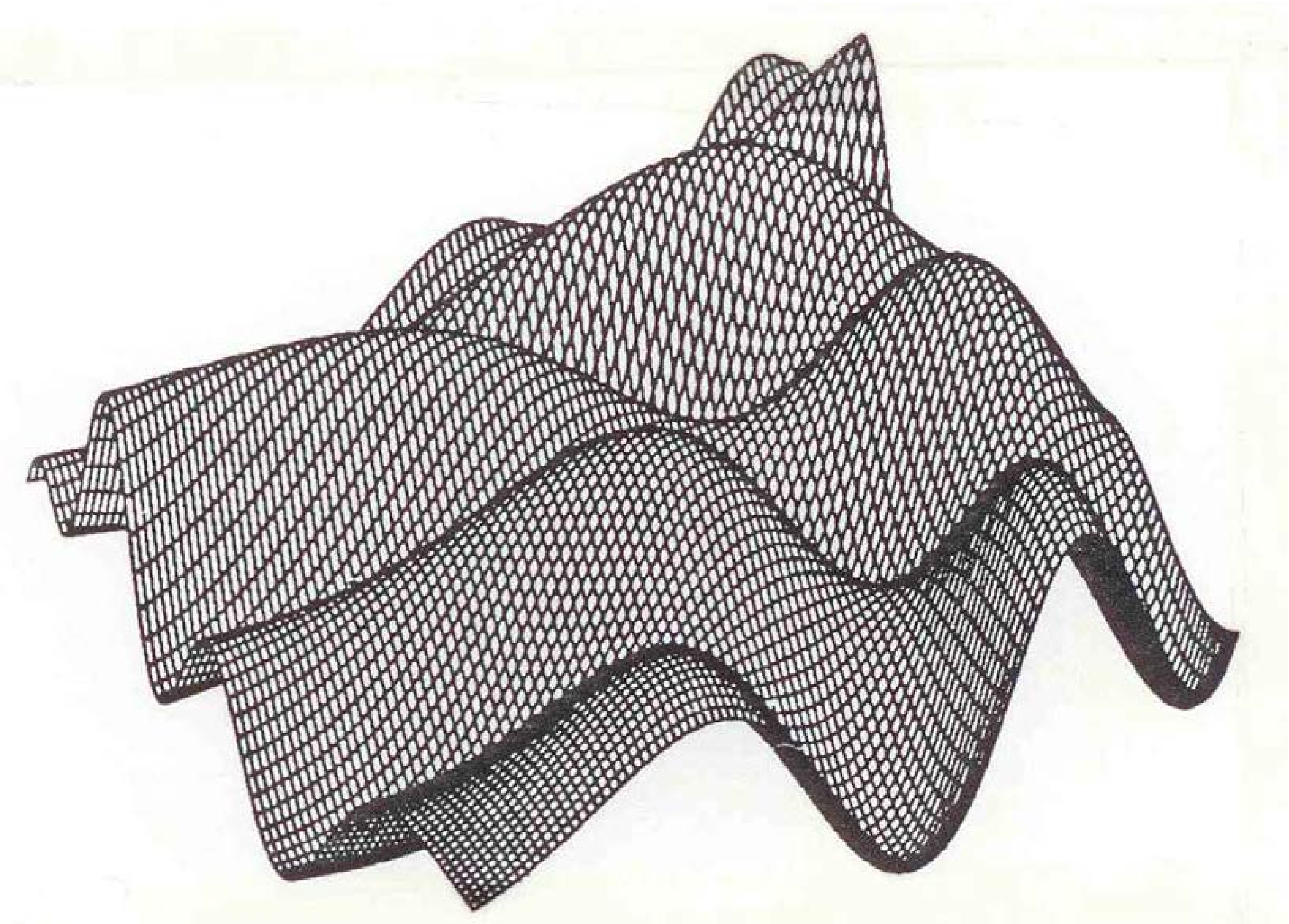
- Thus, $E(\mathbf{x})$ decreases every time x_j flips. Since E is bounded, the Hopfield net is always stable

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

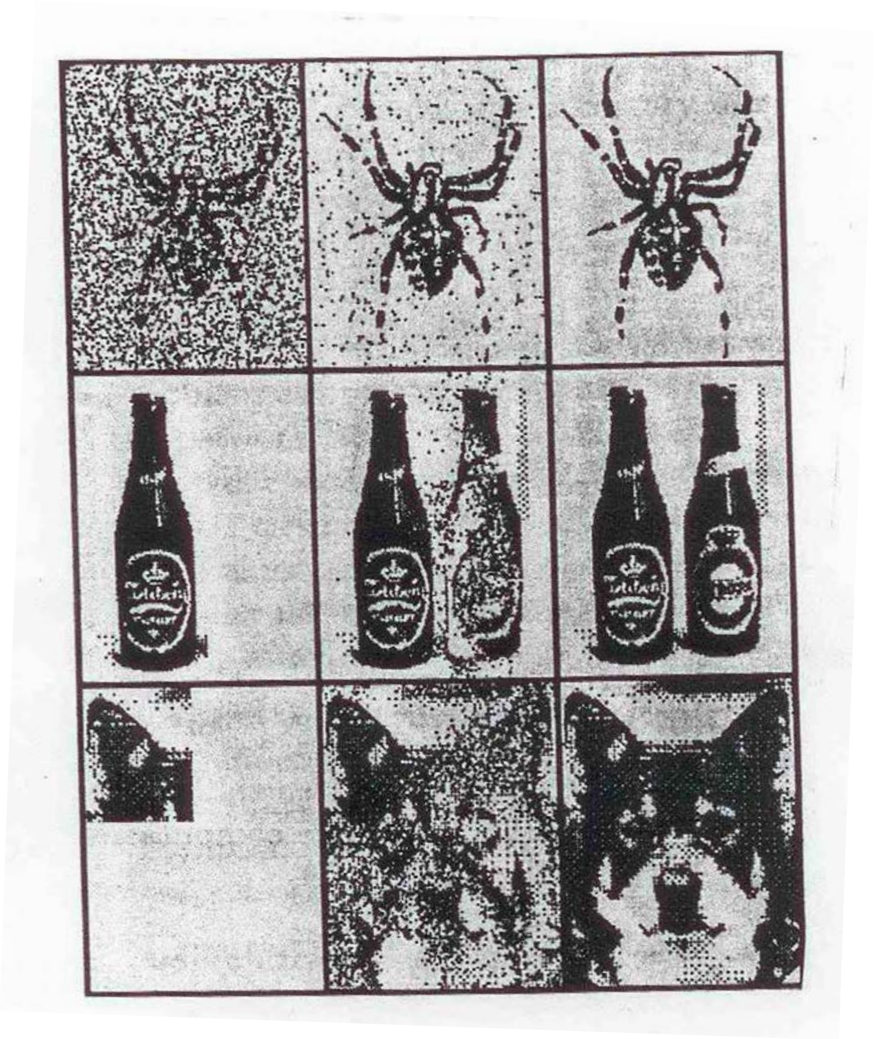
Energy contour map



2-D energy surface



Memory recall illustration



Hertz, Krogh, and Palmer (1991), Ch 2

Remarks (cont.)

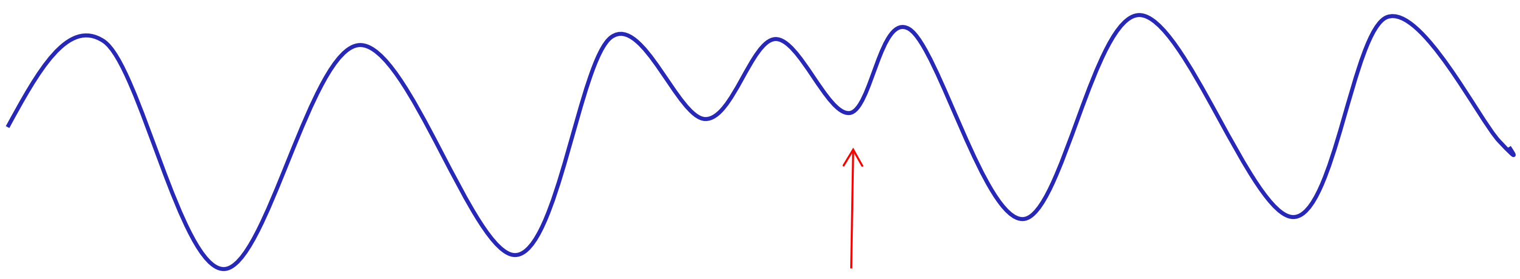
- Bipolar neurons can be extended to continuous-valued neurons by using hyperbolic tangent activation function, and discrete update can be extended to continuous-time dynamics (good for analog VLSI implementation)
- The concept of energy minimization has been applied to optimization problems (neural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

Spurious states

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

Spurious states (cont.)

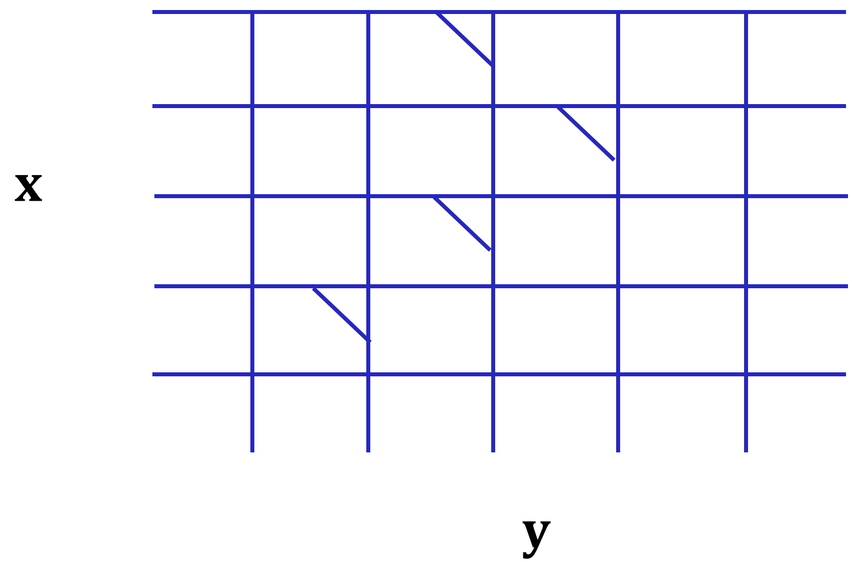
- Spurious states tend to have smaller basins and occur higher on the energy surface



local minima
for spurious states

Kinds of associative memory

- $\begin{cases} \text{Autoassociative (e.g. Hopfieled net)} \\ \text{Heteroassociative: store pairs } \langle x_\mu, y_\mu \rangle \text{ explicitly} \end{cases}$



matrix memory
(Anderson 1972)

holographic memory
(van Heerden, 1963)