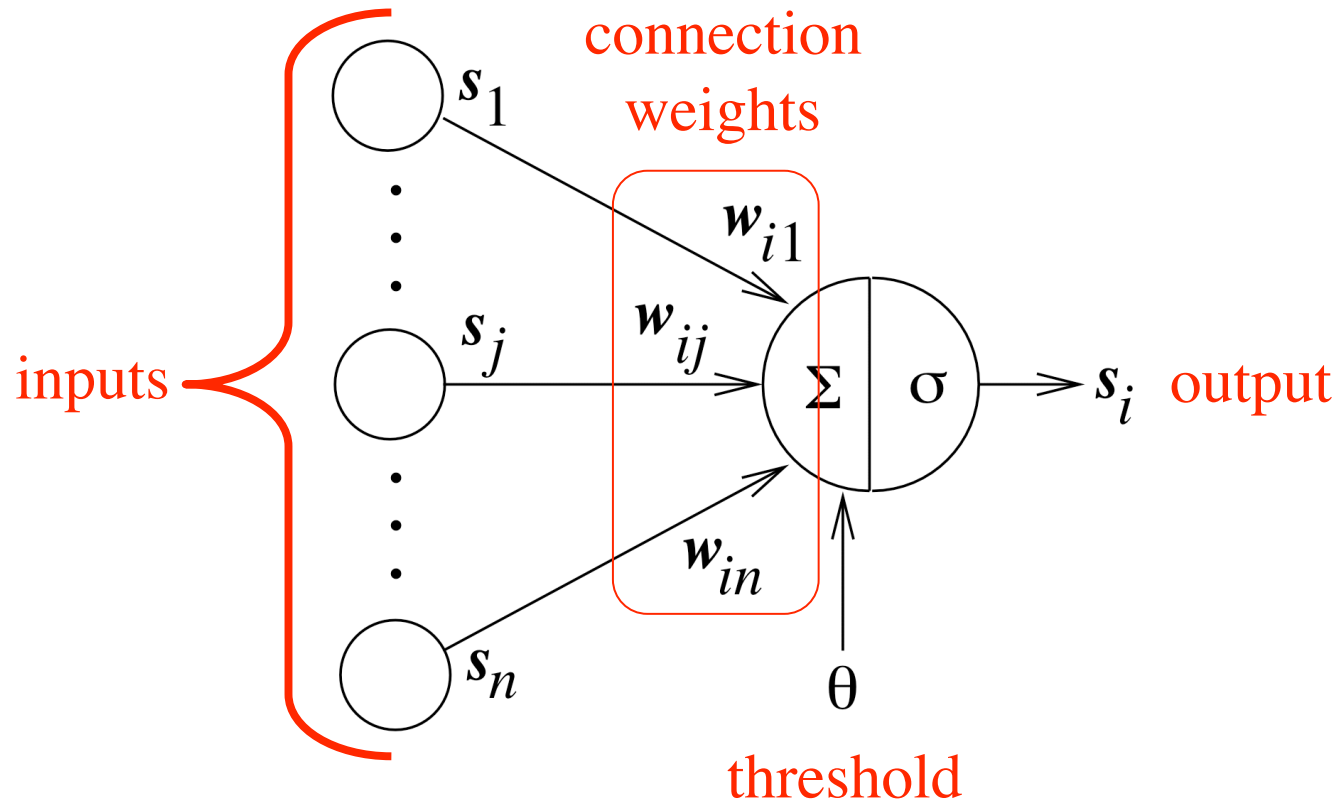


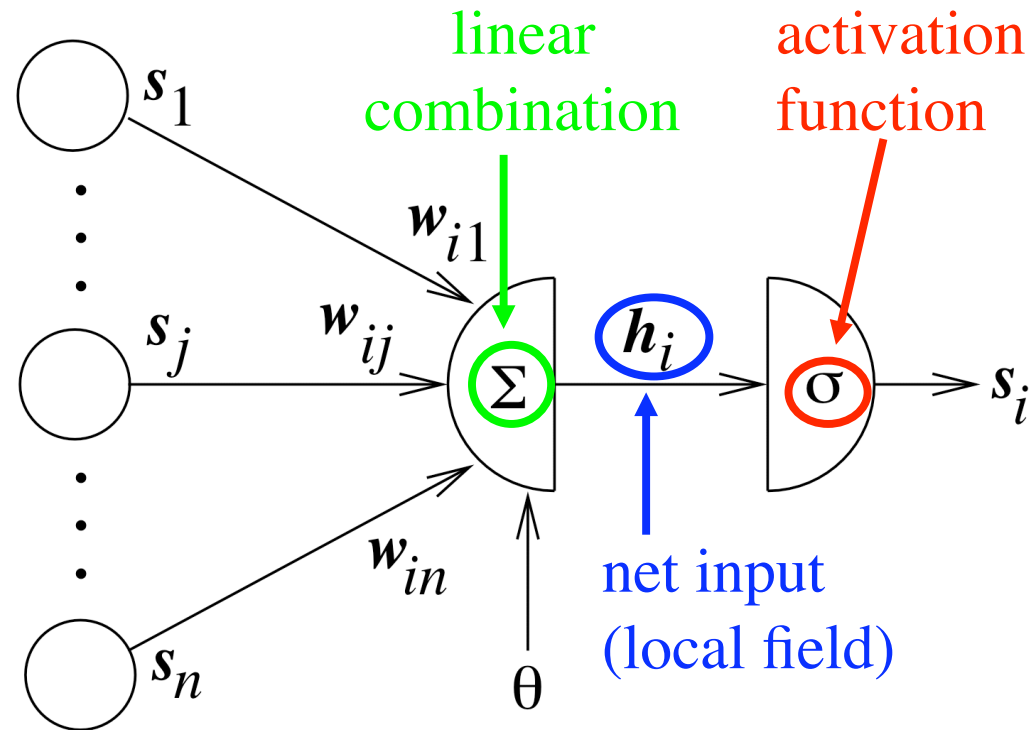
III. Recurrent Neural Networks

A.
The Hopfield Network

Typical Artificial Neuron



Typical Artificial Neuron



Equations

Net input:

$$h_i = \left(\sum_{j=1}^n w_{ij} s_j \right) - \theta$$

$$\mathbf{h} = \mathbf{W}\mathbf{s} - \theta$$

New neural state:

$$s'_i = \sigma(h_i)$$

$$\mathbf{s}' = \sigma(\mathbf{h})$$

Hopfield Network

- Symmetric weights: $w_{ij} = w_{ji}$
- No self-action: $w_{ii} = 0$
- Zero threshold: $\theta = 0$
- Bipolar states: $s_i \in \{-1, +1\}$
- Discontinuous bipolar activation function:

$$\sigma(h) = \text{sgn}(h) = \begin{cases} -1, & h < 0 \\ +1, & h > 0 \end{cases}$$

What to do about $h = 0$?

- There are several options:
 - $\sigma(0) = +1$
 - $\sigma(0) = -1$
 - $\sigma(0) = -1$ or $+1$ with equal probability
 - $h_i = 0 \Rightarrow$ no state change ($s_i' = s_i$)
- Not much difference, but be consistent
- Last option is slightly preferable, since symmetric

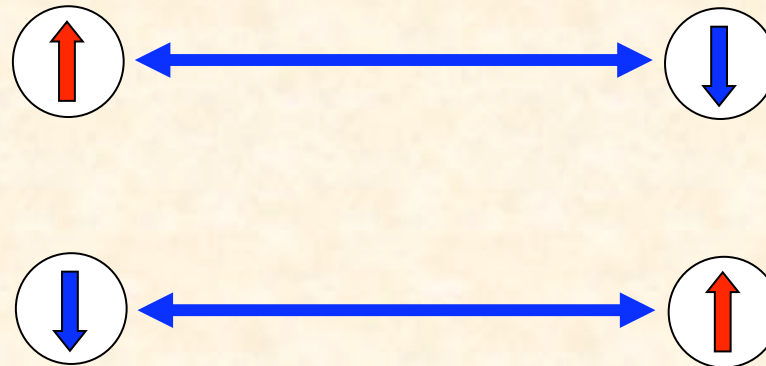
Positive Coupling

- Positive *sense* (sign)
- Large *strength*



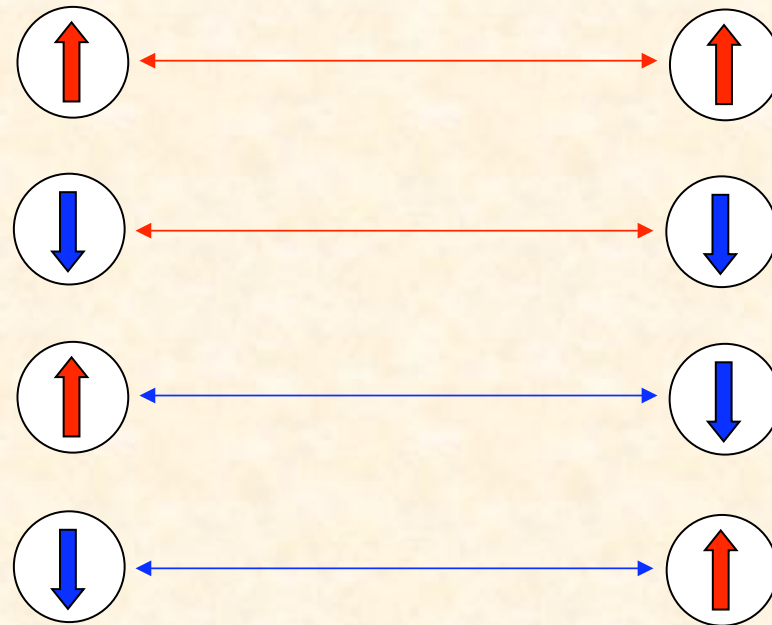
Negative Coupling

- Negative *sense* (sign)
- Large *strength*

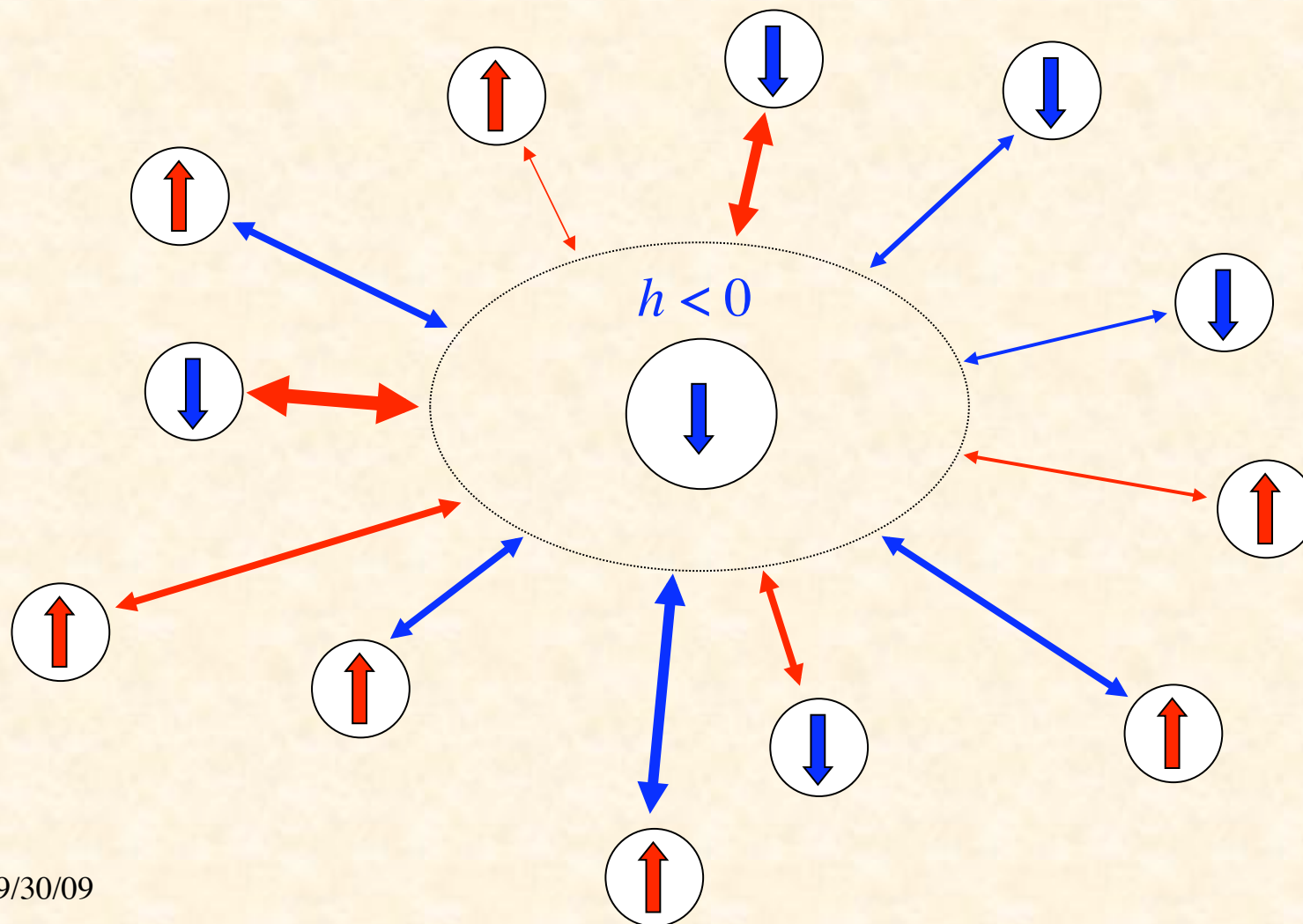


Weak Coupling

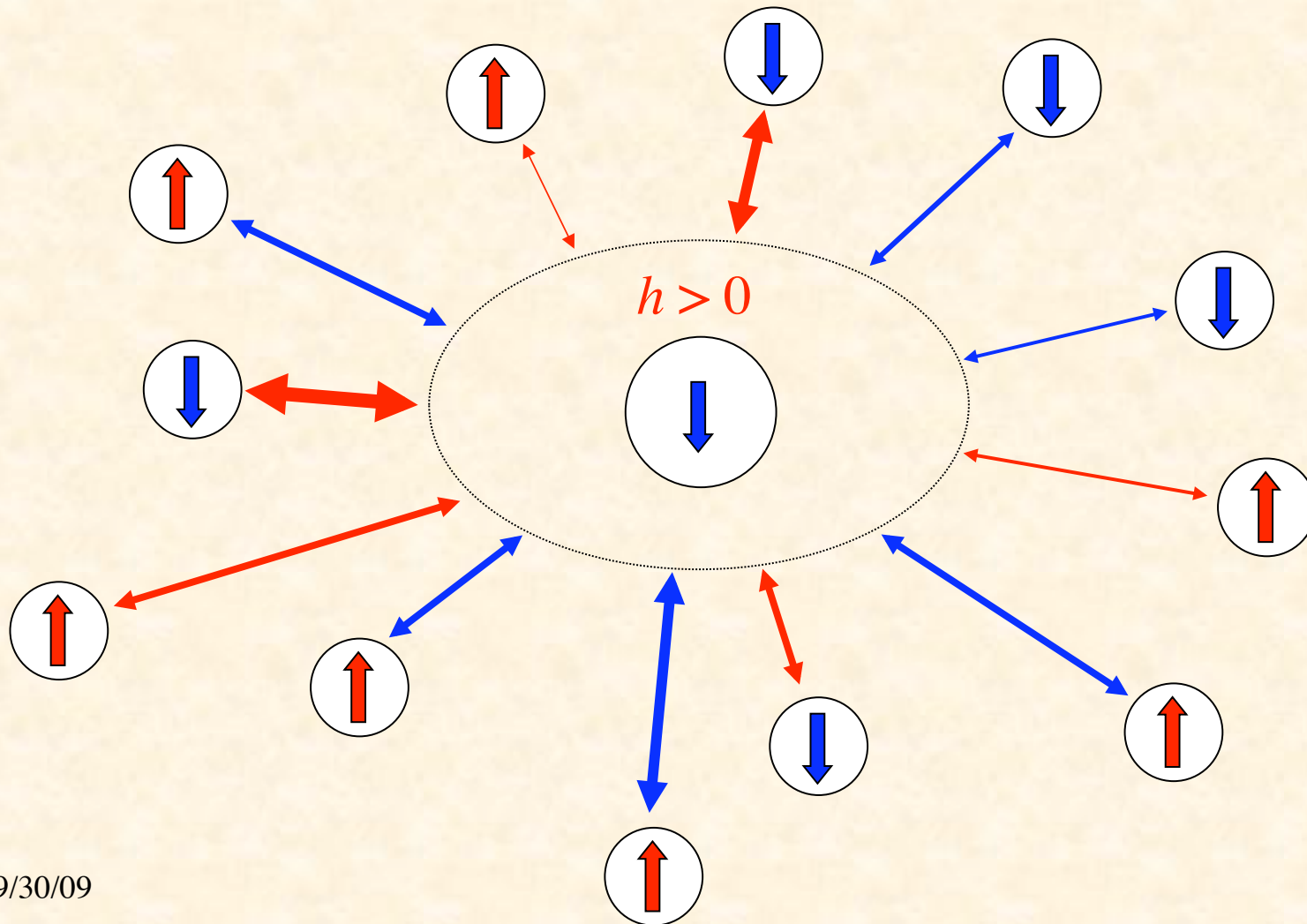
- Either *sense* (sign)
- Little *strength*



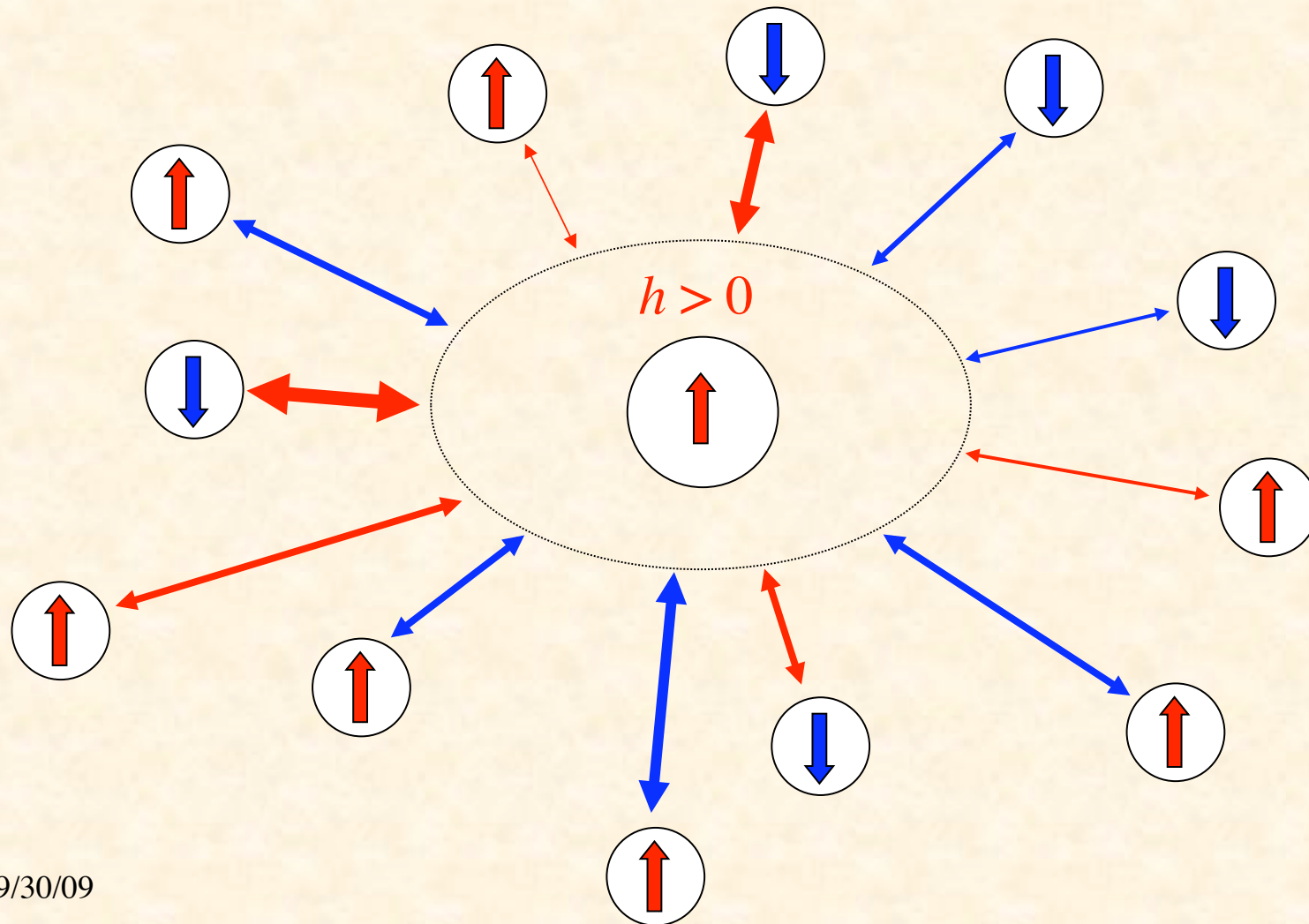
State = -1 & Local Field < 0



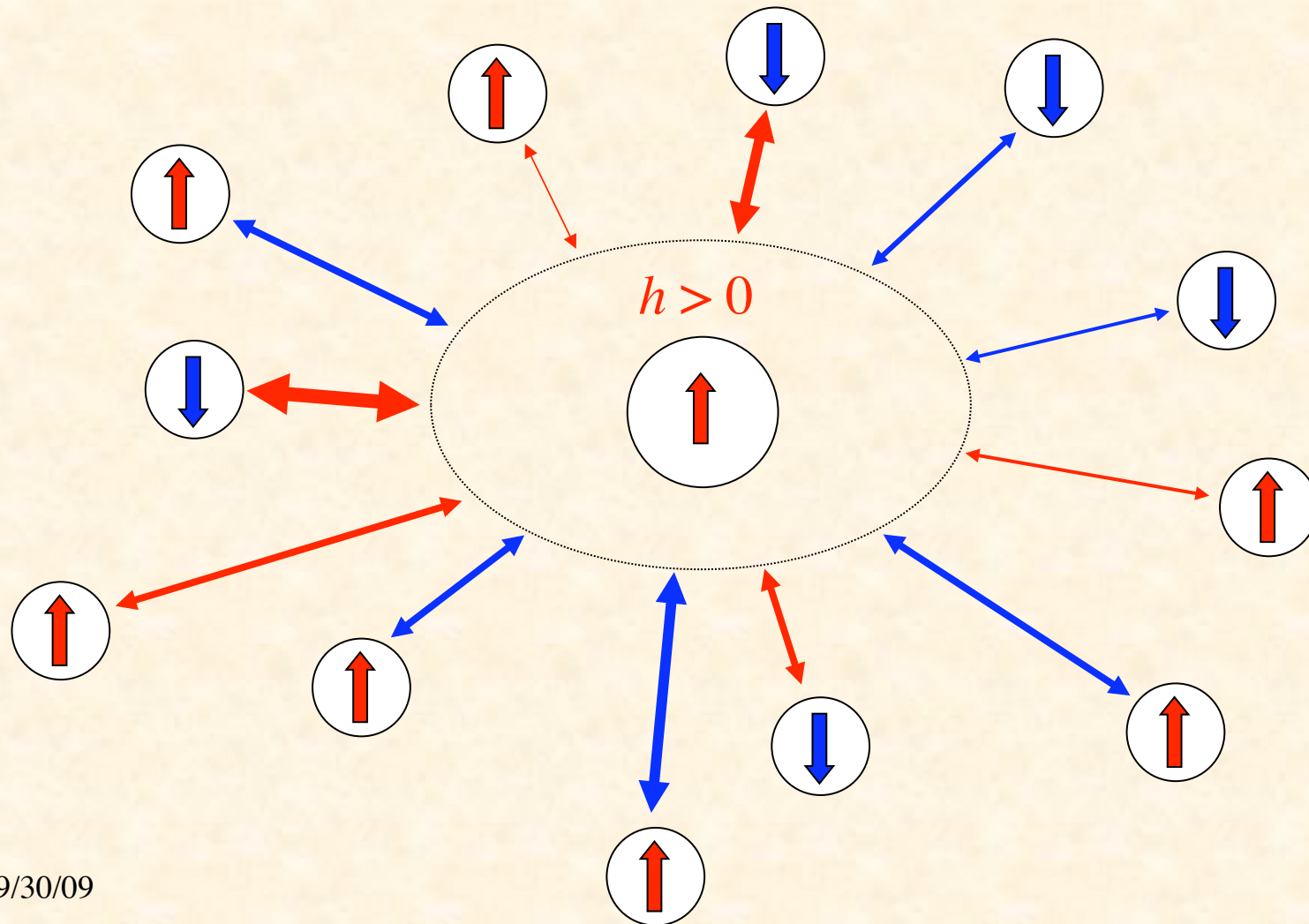
State = -1 & Local Field > 0



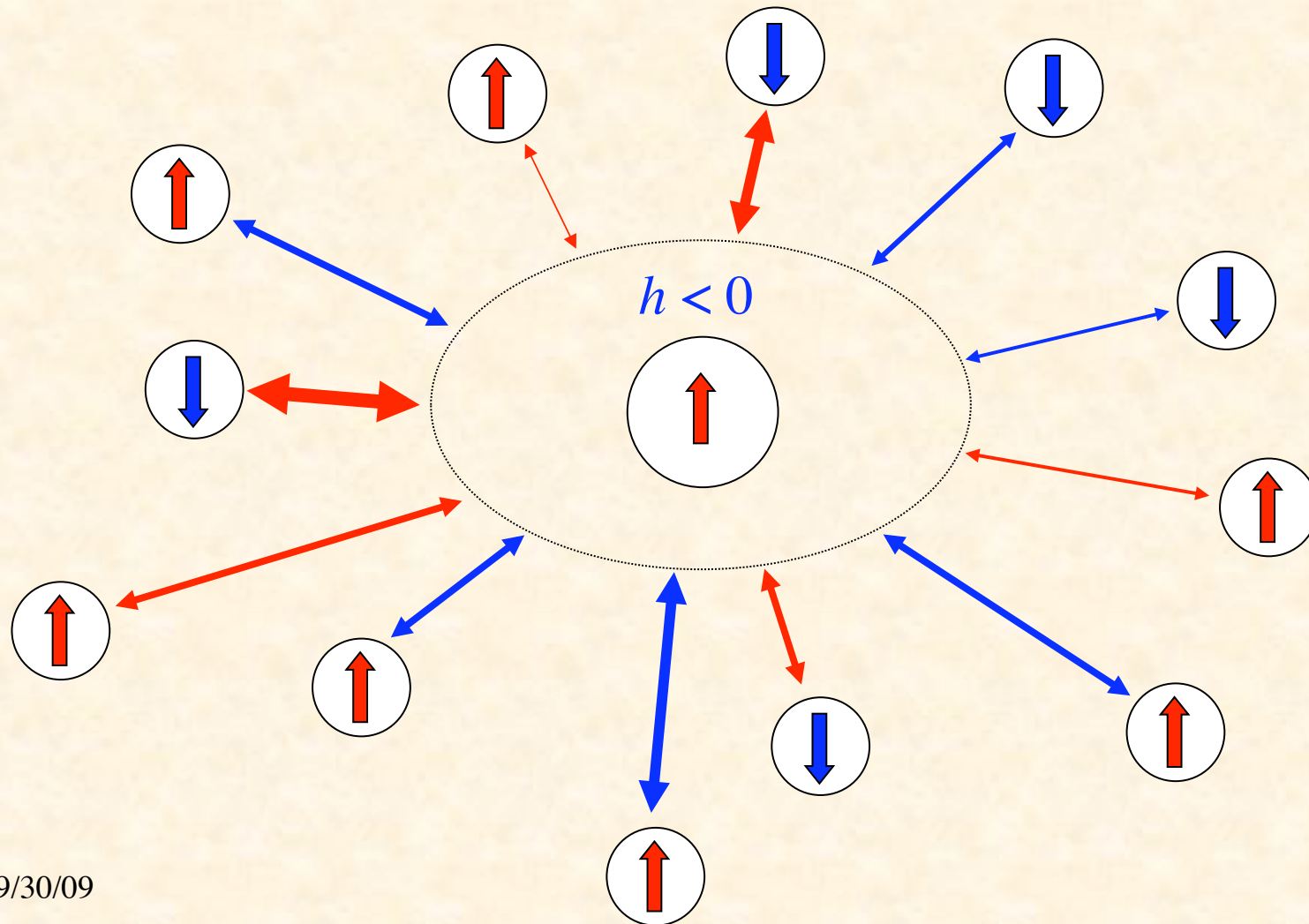
State Reverses



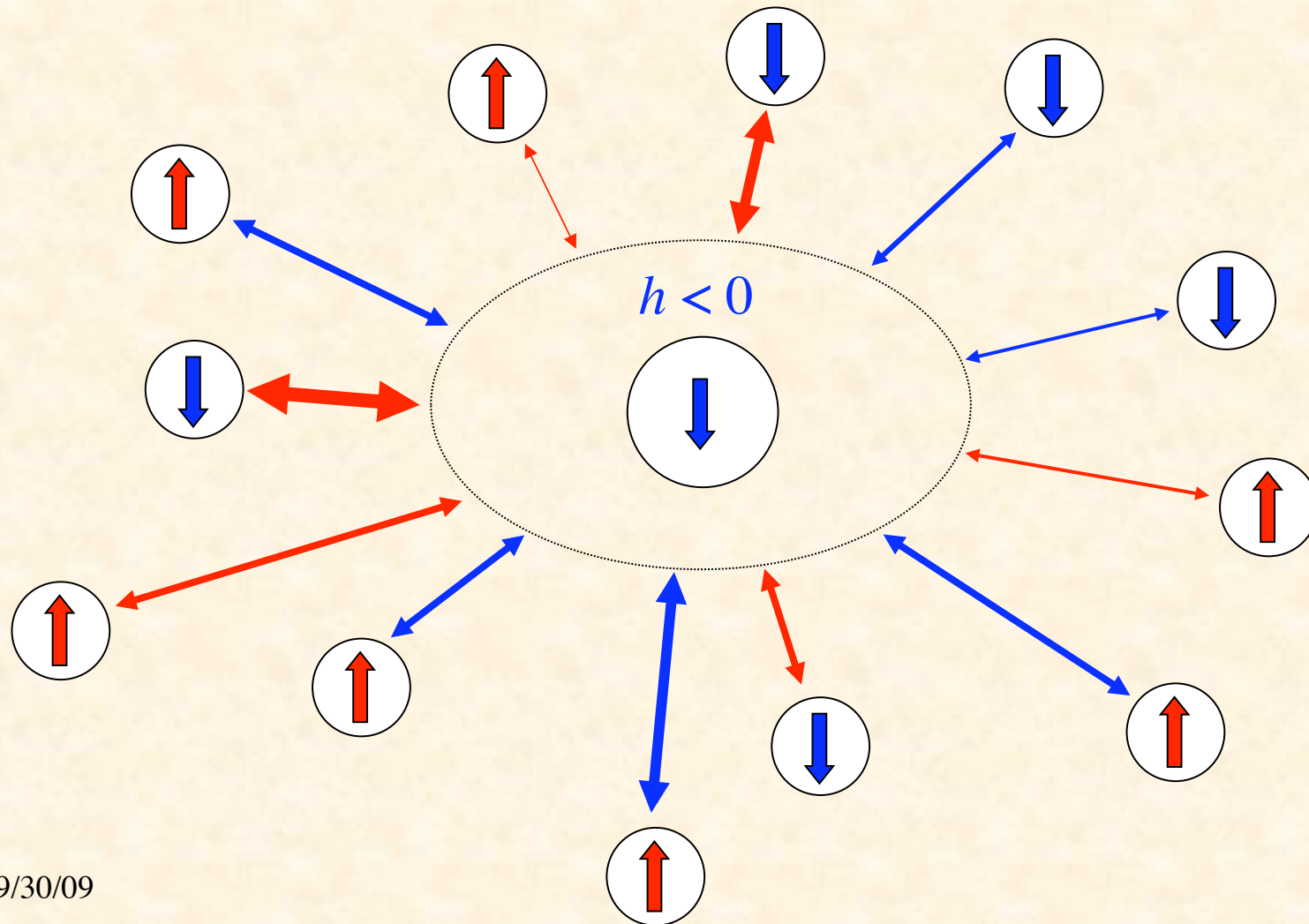
State = +1 & Local Field > 0



State = +1 & Local Field < 0



State Reverses



NetLogo Demonstration of Hopfield State Updating

[Run Hopfield-update.nlogo](#)

Hopfield Net as Soft Constraint Satisfaction System

- States of neurons as yes/no decisions
- Weights represent *soft constraints* between decisions
 - *hard* constraints *must* be respected
 - *soft* constraints have *degrees* of importance
- Decisions change to better respect constraints
- Is there an optimal set of decisions that best respects all constraints?

Demonstration of Hopfield Net Dynamics I

[Run Hopfield-dynamics.nlogo](#)

Convergence

- Does such a system converge to a stable state?
- Under what conditions does it converge?
- There is a sense in which each step relaxes the “tension” in the system
- But could a relaxation of one neuron lead to greater tension in other places?

Quantifying “Tension”

- If $w_{ij} > 0$, then s_i and s_j want to have the same sign ($s_i s_j = +1$)
- If $w_{ij} < 0$, then s_i and s_j want to have opposite signs ($s_i s_j = -1$)
- If $w_{ij} = 0$, their signs are independent
- Strength of interaction varies with $|w_{ij}|$
- Define disharmony (“tension”) D_{ij} between neurons i and j :

$$D_{ij} = -s_i w_{ij} s_j$$

$$D_{ij} < 0 \Rightarrow \text{they are happy}$$

$$D_{ij} > 0 \Rightarrow \text{they are unhappy}$$

Total Energy of System

The “energy” of the system is the total “tension” (disharmony) in it:

$$\begin{aligned} E\{\mathbf{s}\} &= \sum_{\langle ij \rangle} D_{ij} = - \sum_{\langle ij \rangle} s_i w_{ij} s_j \\ &= -\frac{1}{2} \sum_i \sum_{j \neq i} s_i w_{ij} s_j \\ &= -\frac{1}{2} \sum_i \sum_j s_i w_{ij} s_j \\ &= -\frac{1}{2} \mathbf{s}^T \mathbf{W} \mathbf{s} \end{aligned}$$

Review of Some Vector Notation

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = (x_1 \quad \cdots \quad x_n)^T \quad \text{(column vectors)}$$

$$\mathbf{x}^T \mathbf{y} = \sum_{i=1}^n x_i y_i = \mathbf{x} \cdot \mathbf{y} \quad \text{(inner product)}$$

$$\mathbf{xy}^T = \begin{pmatrix} x_1 y_1 & \cdots & x_1 y_n \\ \vdots & \ddots & \vdots \\ x_m y_1 & \cdots & x_m y_n \end{pmatrix} \quad \text{(outer product)}$$

$$\mathbf{x}^T \mathbf{M} \mathbf{y} = \sum_{i=1}^m \sum_{j=1}^n x_i M_{ij} y_j \quad \text{(quadratic form)}$$

Another View of Energy

The energy measures the number of neurons whose states are in disharmony with their local fields (i.e. of opposite sign):

$$\begin{aligned} E\{\mathbf{s}\} &= -\frac{1}{2} \sum_i \sum_j s_i w_{ij} s_j \\ &= -\frac{1}{2} \sum_i s_i \sum_j w_{ij} s_j \\ &= -\frac{1}{2} \sum_i s_i h_i \\ &= -\frac{1}{2} \mathbf{s}^T \mathbf{h} \end{aligned}$$

Do State Changes Decrease Energy?

- Suppose that neuron k changes state
- Change of energy:

$$\begin{aligned}\Delta E &= E\{\mathbf{s}'\} - E\{\mathbf{s}\} \\ &= -\sum_{\langle ij \rangle} s'_i w_{ij} s'_j + \sum_{\langle ij \rangle} s_i w_{ij} s_j \\ &= -\sum_{j \neq k} s'_k w_{kj} s_j + \sum_{j \neq k} s_k w_{kj} s_j \\ &= -(s'_k - s_k) \sum_{j \neq k} w_{kj} s_j \\ &= -\Delta s_k h_k \\ &< 0\end{aligned}$$

Energy Does Not Increase

- In each step in which a neuron is considered for update:

$$E\{\mathbf{s}(t + 1)\} - E\{\mathbf{s}(t)\} \leq 0$$

- Energy cannot increase
- Energy decreases if any neuron changes
- Must it stop?

Proof of Convergence in Finite Time

- There is a minimum possible energy:
 - The number of possible states $\mathbf{s} \in \{-1, +1\}^n$ is finite
 - Hence $E_{\min} = \min \{E(\mathbf{s}) \mid \mathbf{s} \in \{\pm 1\}^n\}$ exists
- Must show it is reached in a finite number of steps

Steps are of a Certain Minimum Size

$$h_{\min} = \min \left\{ |h_k| \text{ s.t. } h_k = \sum_{j \neq k} w_{kj} s_j \wedge s \in \{\pm 1\}^n \right\}$$

$$|\Delta E| = 2|h_k| \geq 2h_{\min}$$

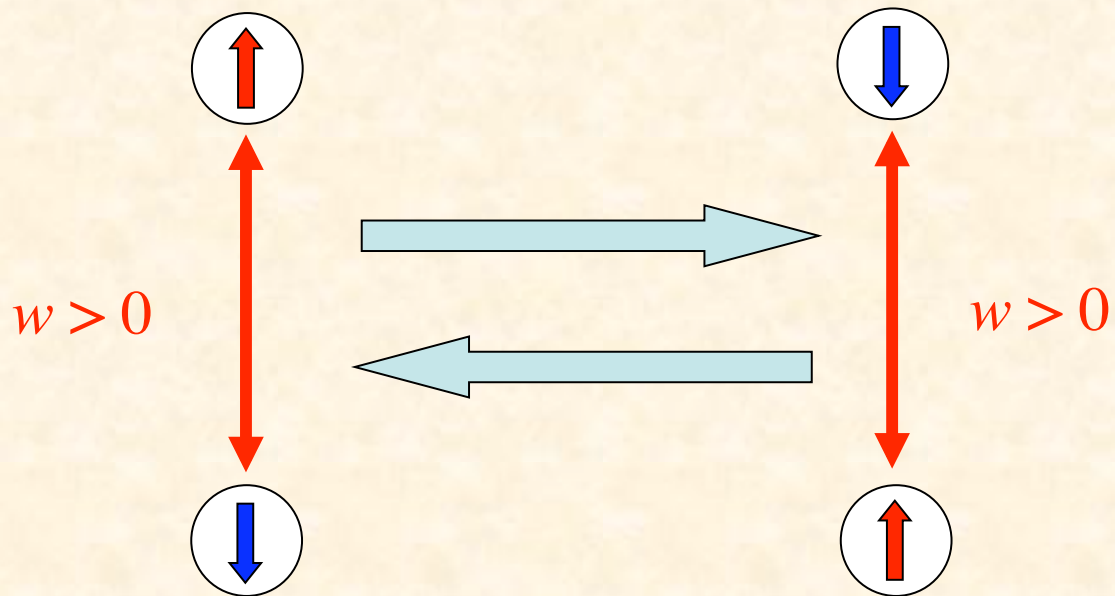
Conclusion

- If we do asynchronous updating, the Hopfield net must reach a stable, minimum energy state in a finite number of updates
- This does not imply that it is a global minimum

Lyapunov Functions

- A way of showing the convergence of discrete- or continuous-time dynamical systems
- For discrete-time system:
 - need a Lyapunov function E (“energy” of the state)
 - E is bounded below ($E\{\mathbf{s}\} > E_{\min}$)
 - $\Delta E < (\Delta E)_{\max} \leq 0$ (energy decreases a certain minimum amount each step)
 - then the system will converge in finite time
- Problem: finding a suitable Lyapunov function

Example Limit Cycle with Synchronous Updating

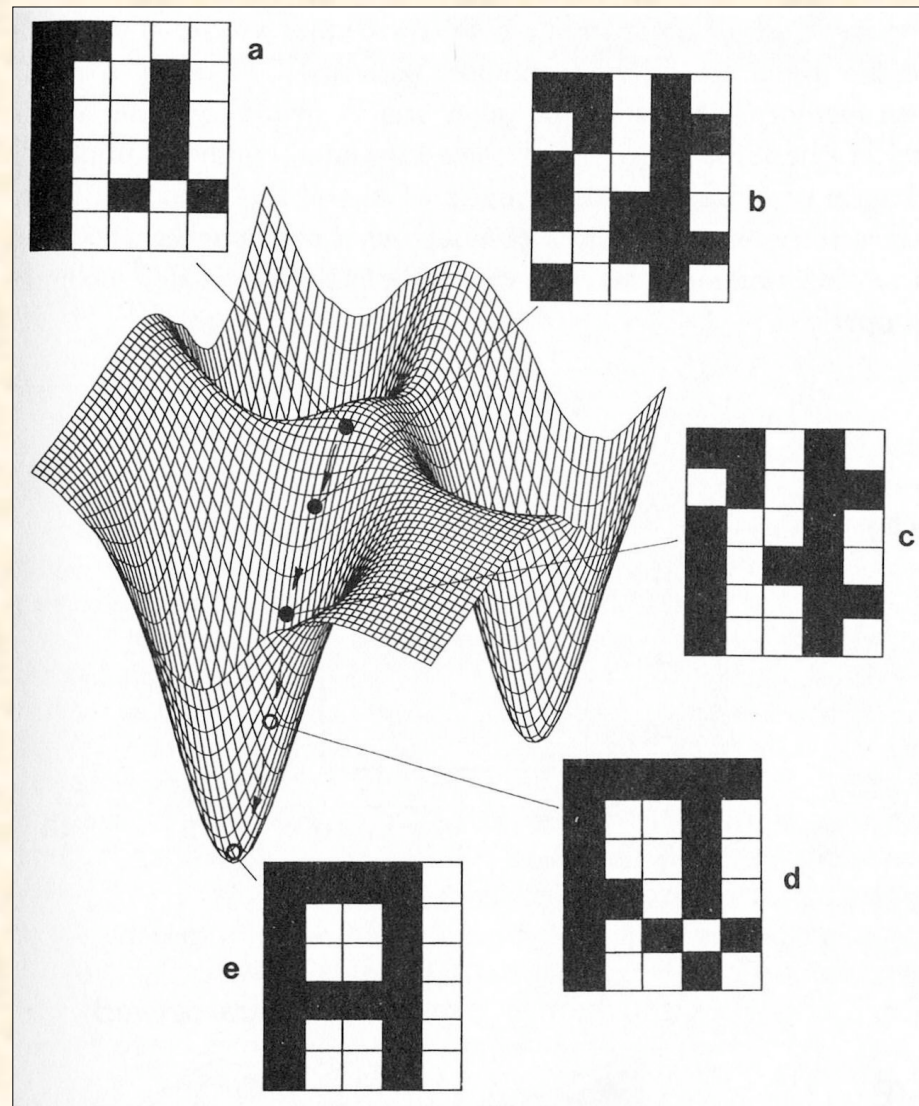


The Hopfield Energy Function is Even

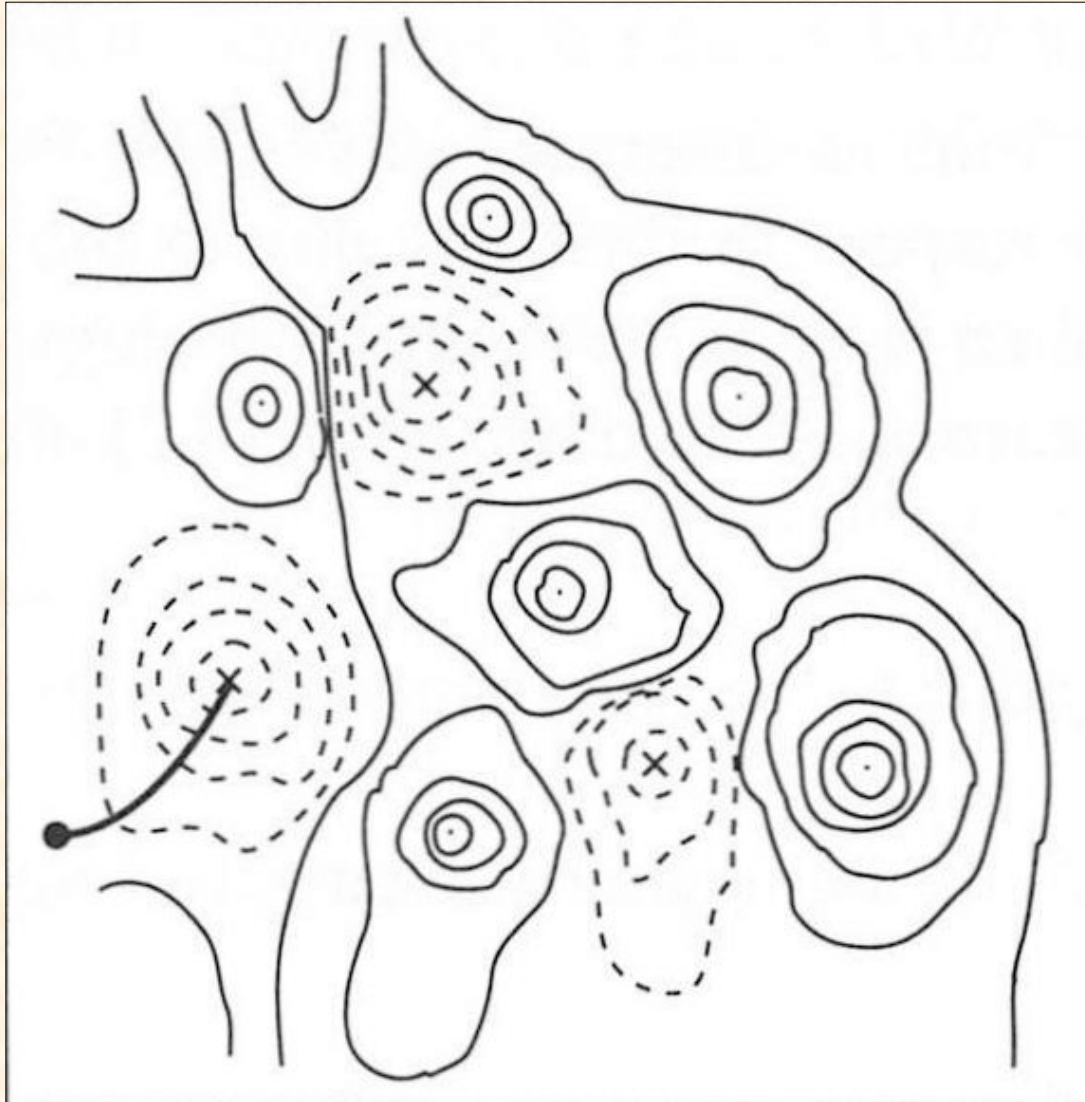
- A function f is **odd** if $f(-x) = -f(x)$, for all x
- A function f is **even** if $f(-x) = f(x)$, for all x
- Observe:

$$E\{-\mathbf{s}\} = -\frac{1}{2}(-\mathbf{s})^T \mathbf{W}(-\mathbf{s}) = -\frac{1}{2}\mathbf{s}^T \mathbf{W}\mathbf{s} = E\{\mathbf{s}\}$$

Conceptual Picture of Descent on Energy Surface



Energy Surface



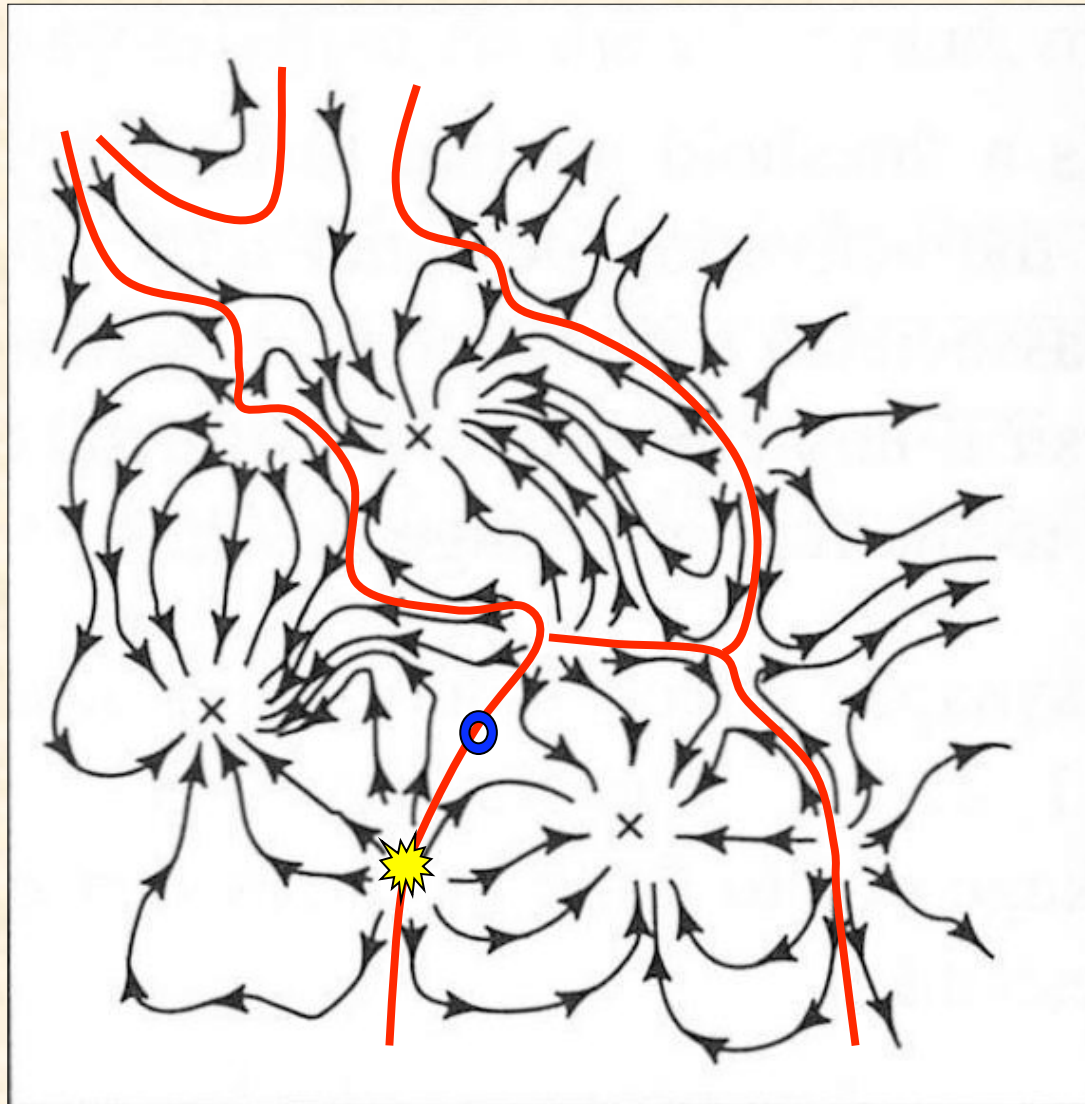
9/30/09

(fig. from Haykin *Neur. Netw.*)

34

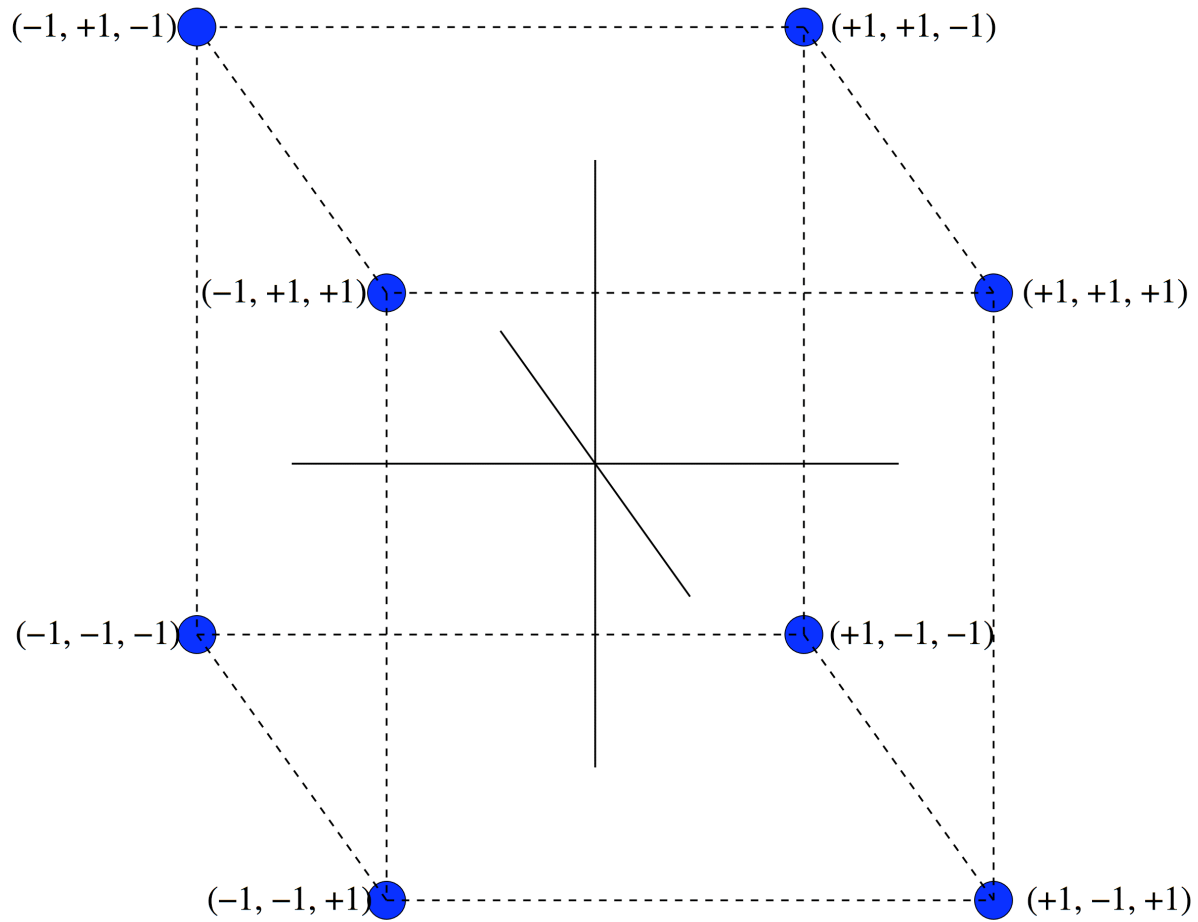


Energy
Surface
+
Flow
Lines



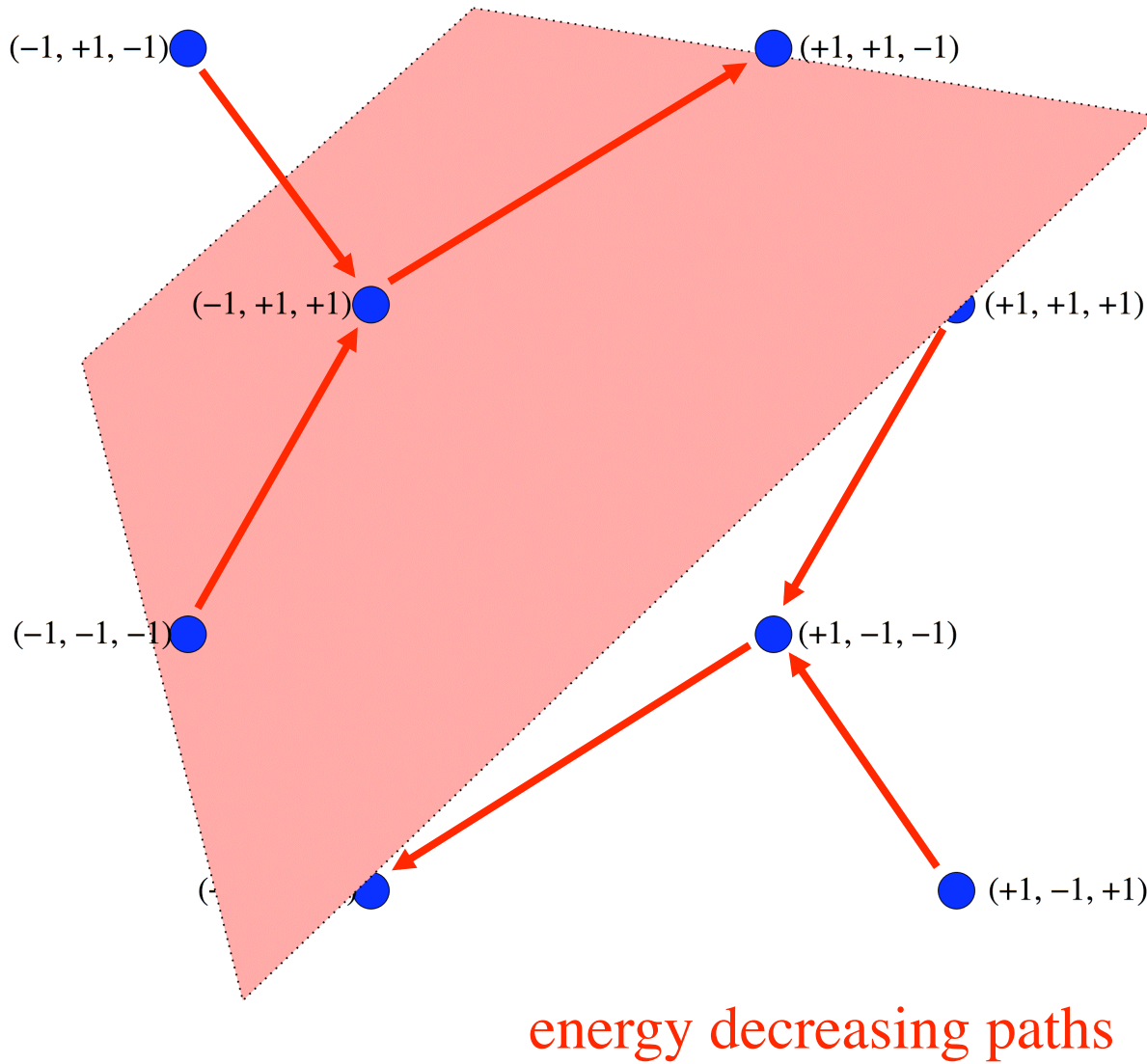
Flow
Lines

Basins of
Attraction



Bipolar State Space

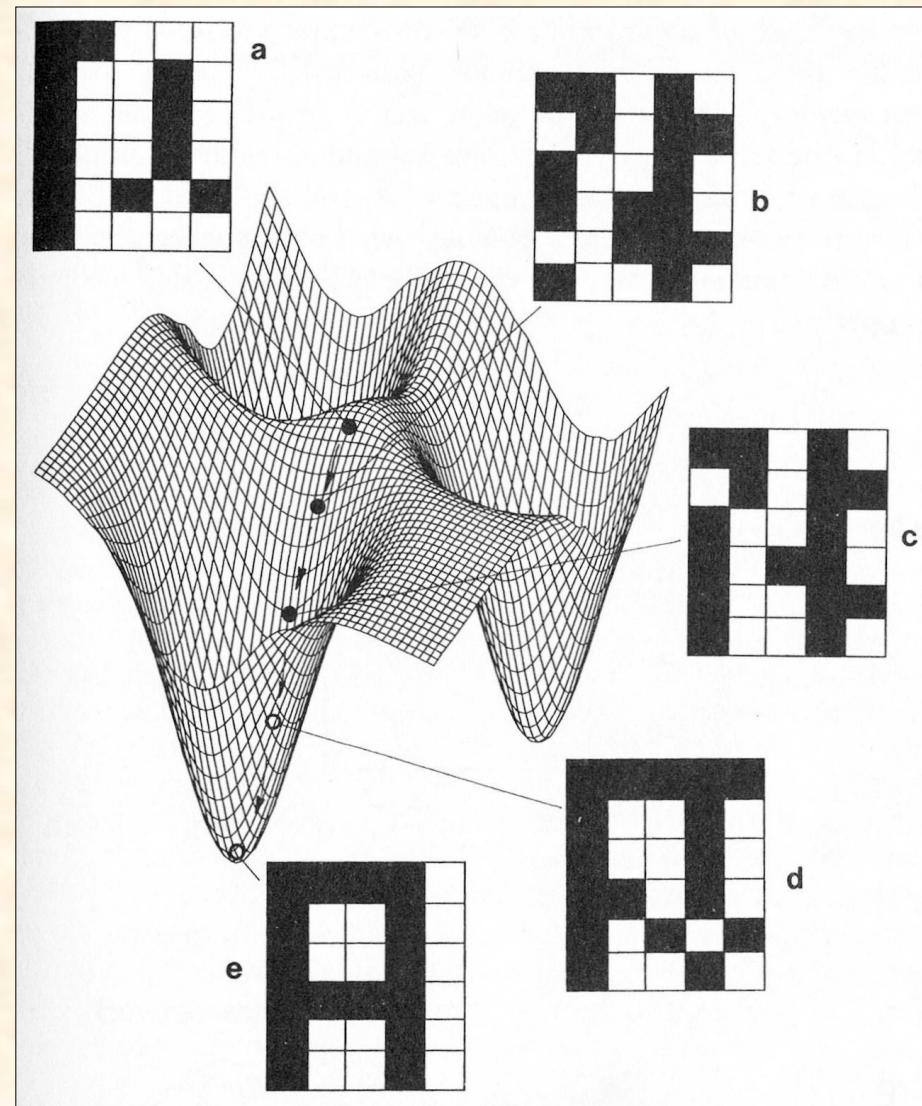
Basins in Bipolar State Space



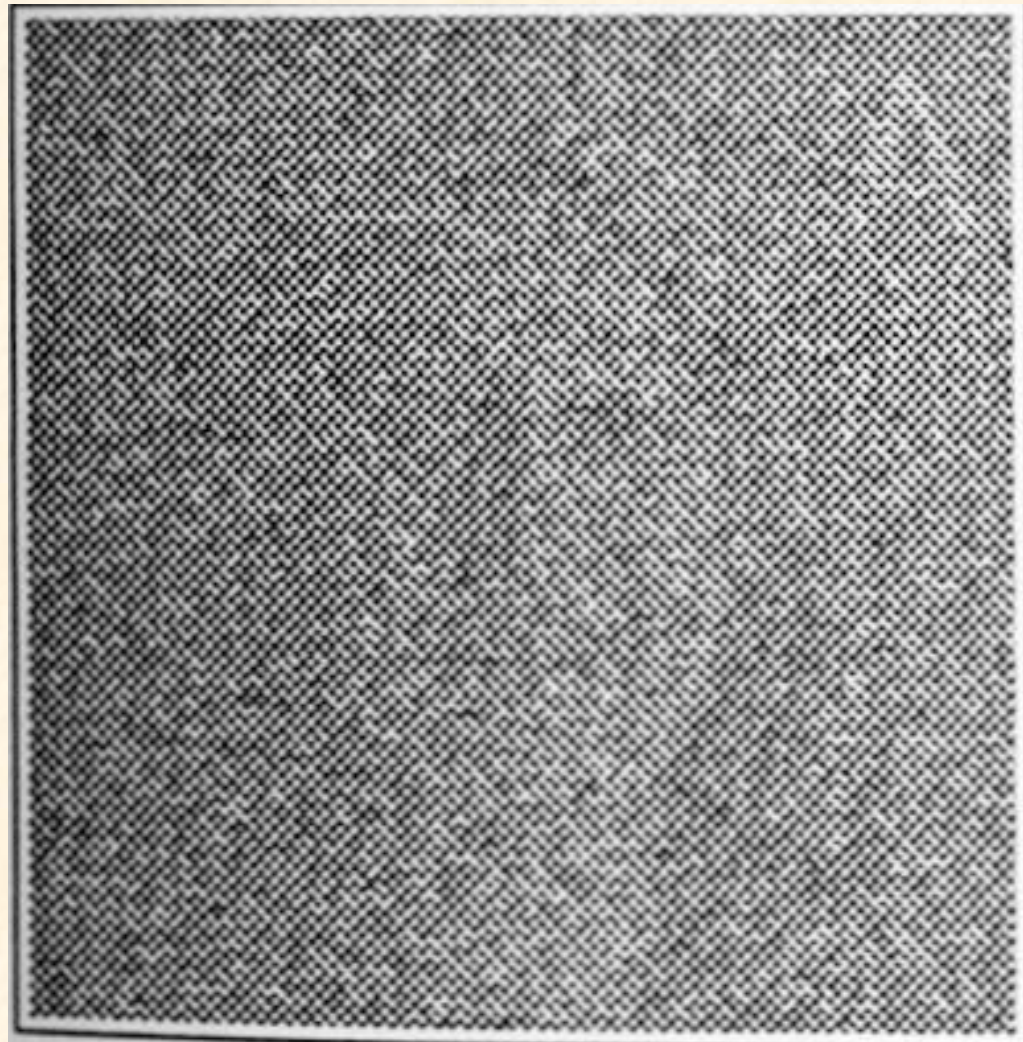
Demonstration of Hopfield Net Dynamics II

Run initialized Hopfield.nlogo

Storing Memories as Attractors



Example of Pattern Restoration



Example of Pattern Restoration



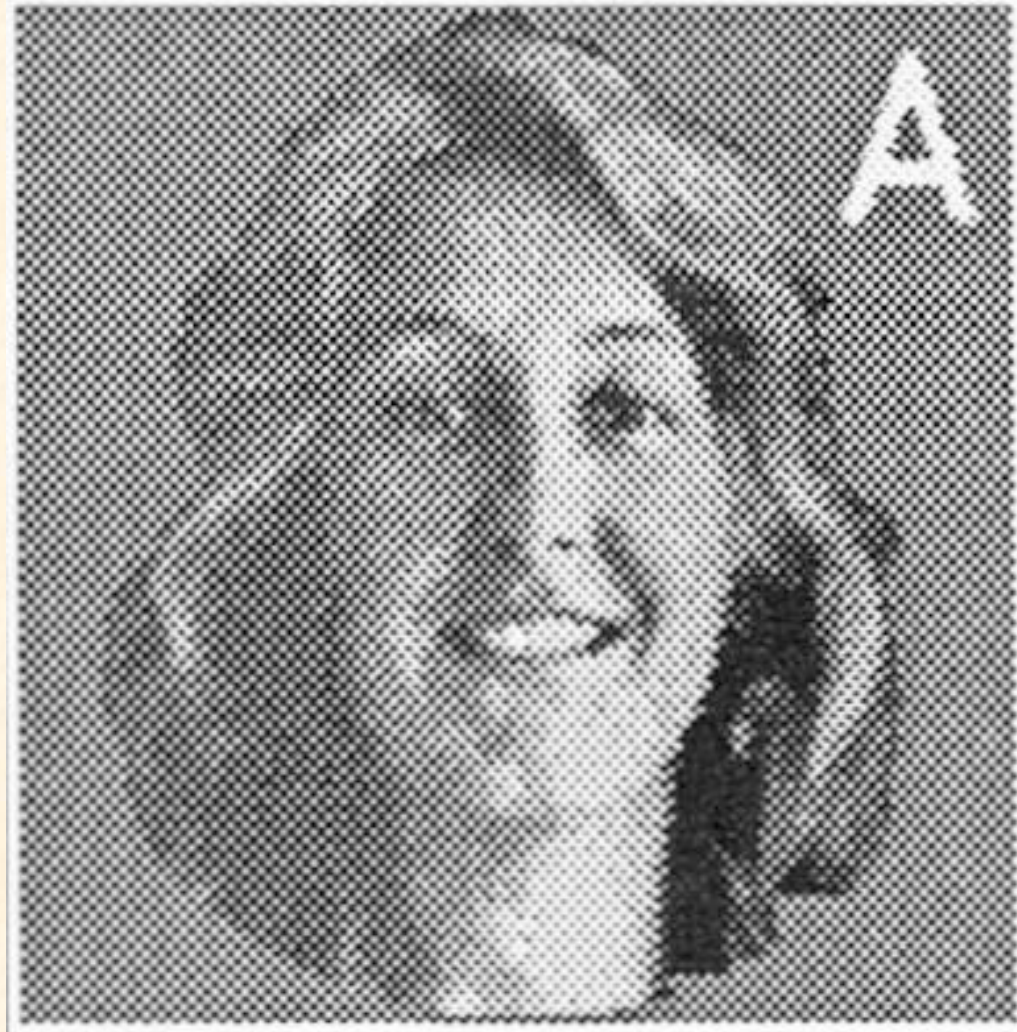
Example of Pattern Restoration



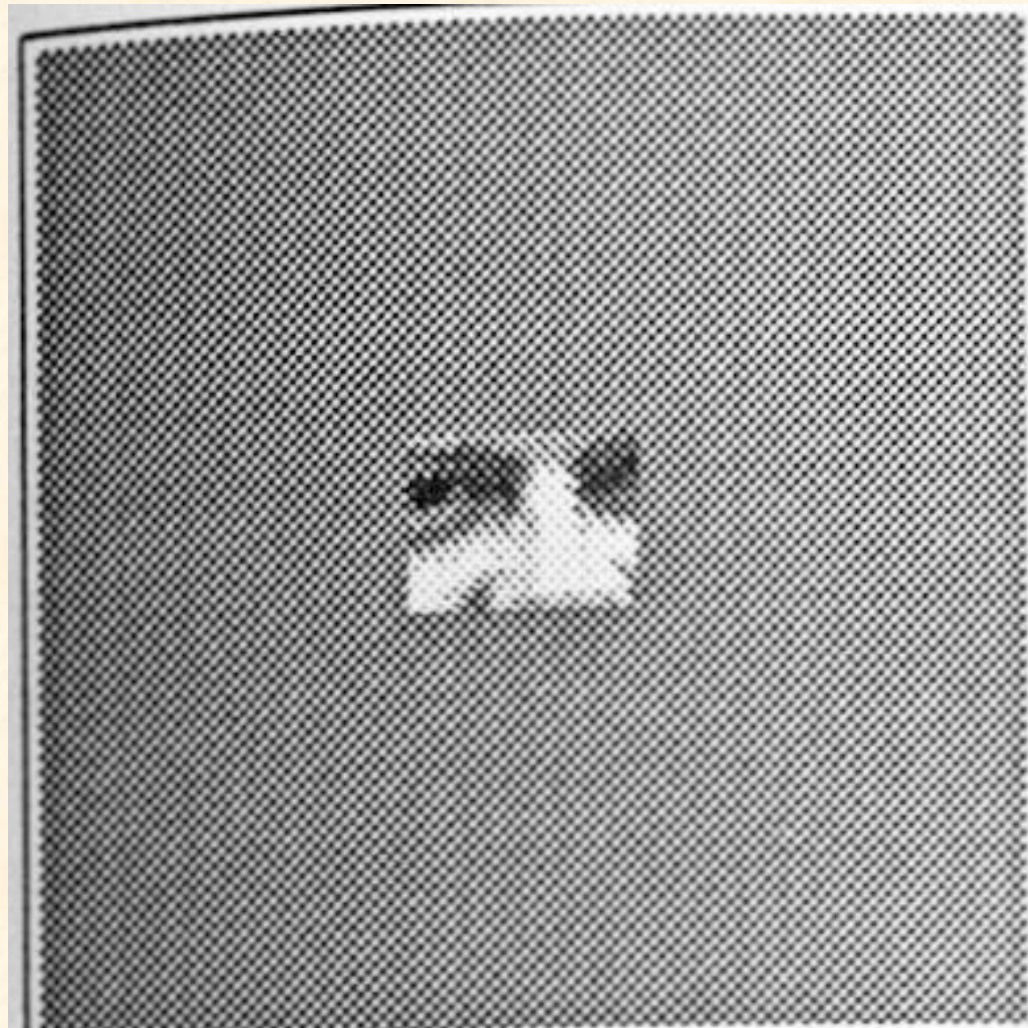
Example of Pattern Restoration



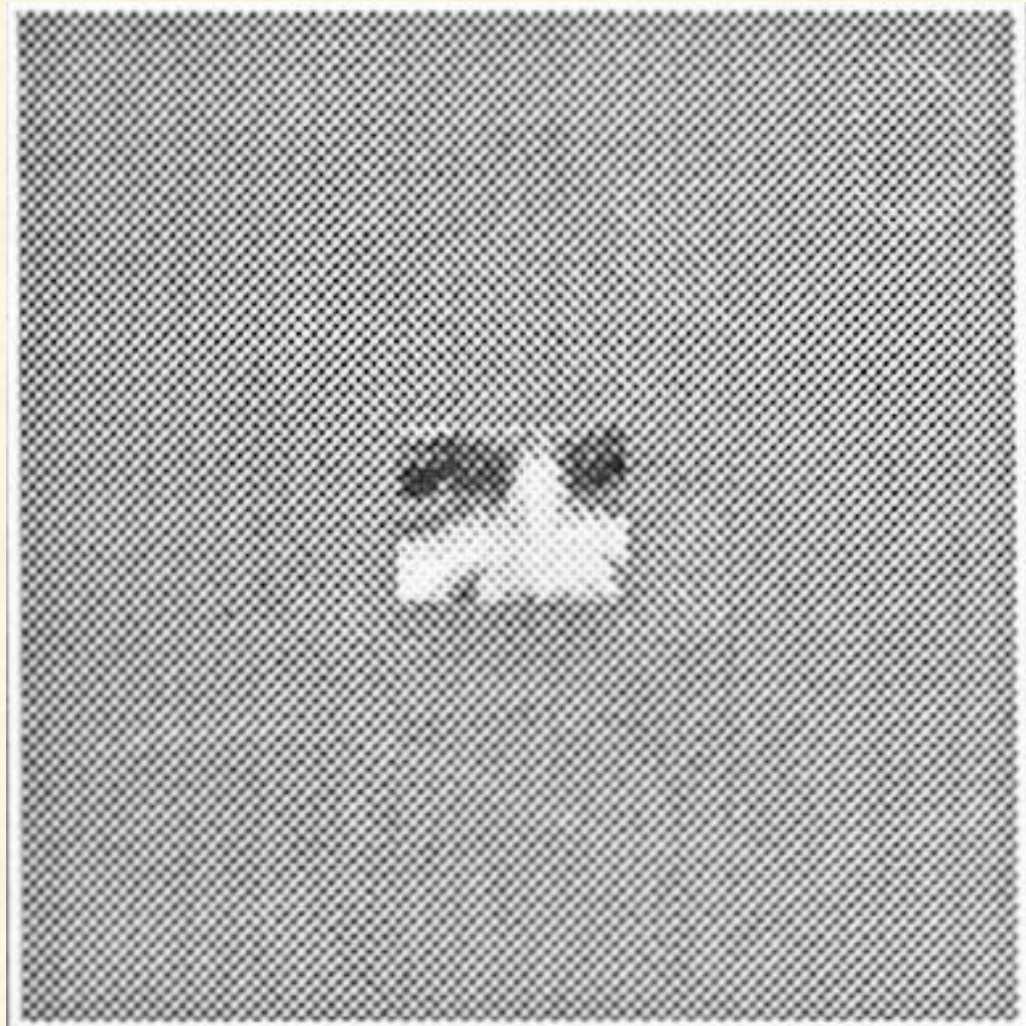
Example of Pattern Restoration



Example of Pattern Completion



Example of Pattern Completion



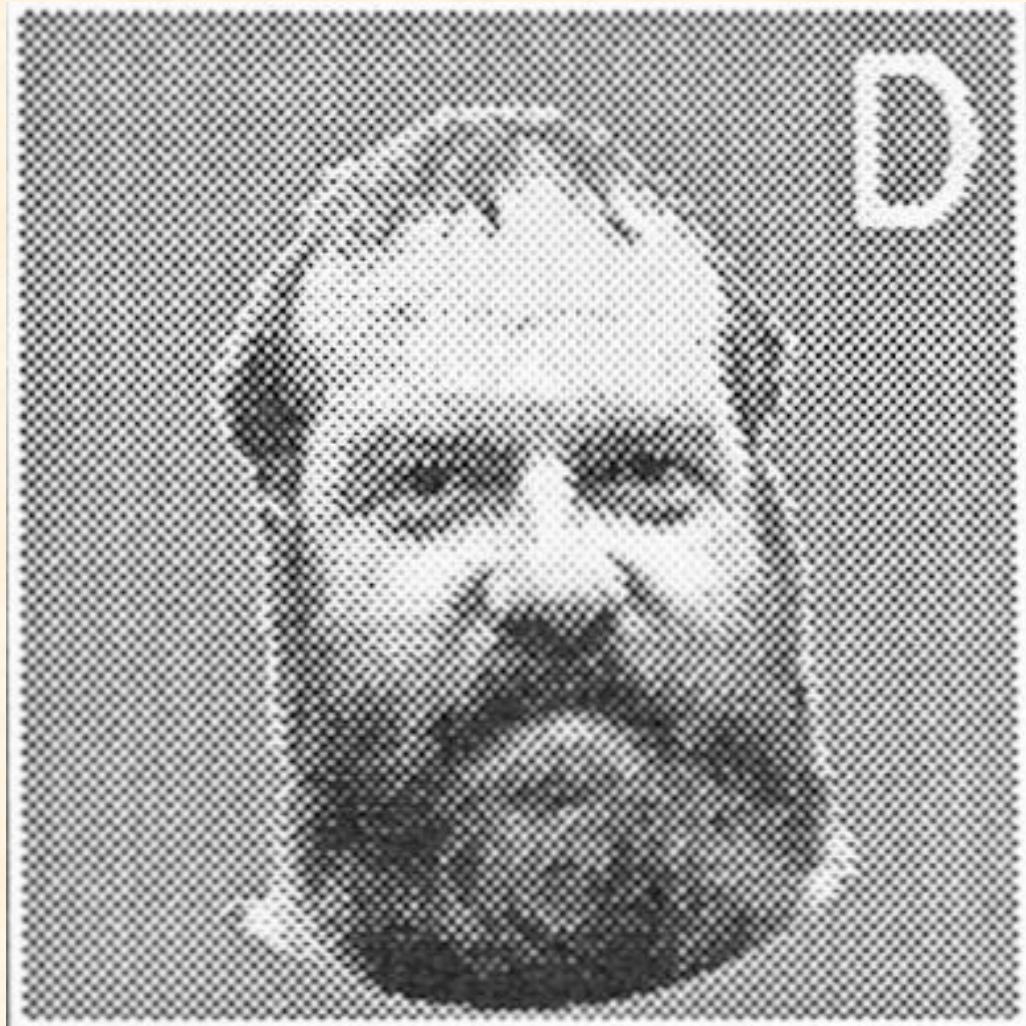
Example of Pattern Completion



Example of Pattern Completion



Example of Pattern Completion

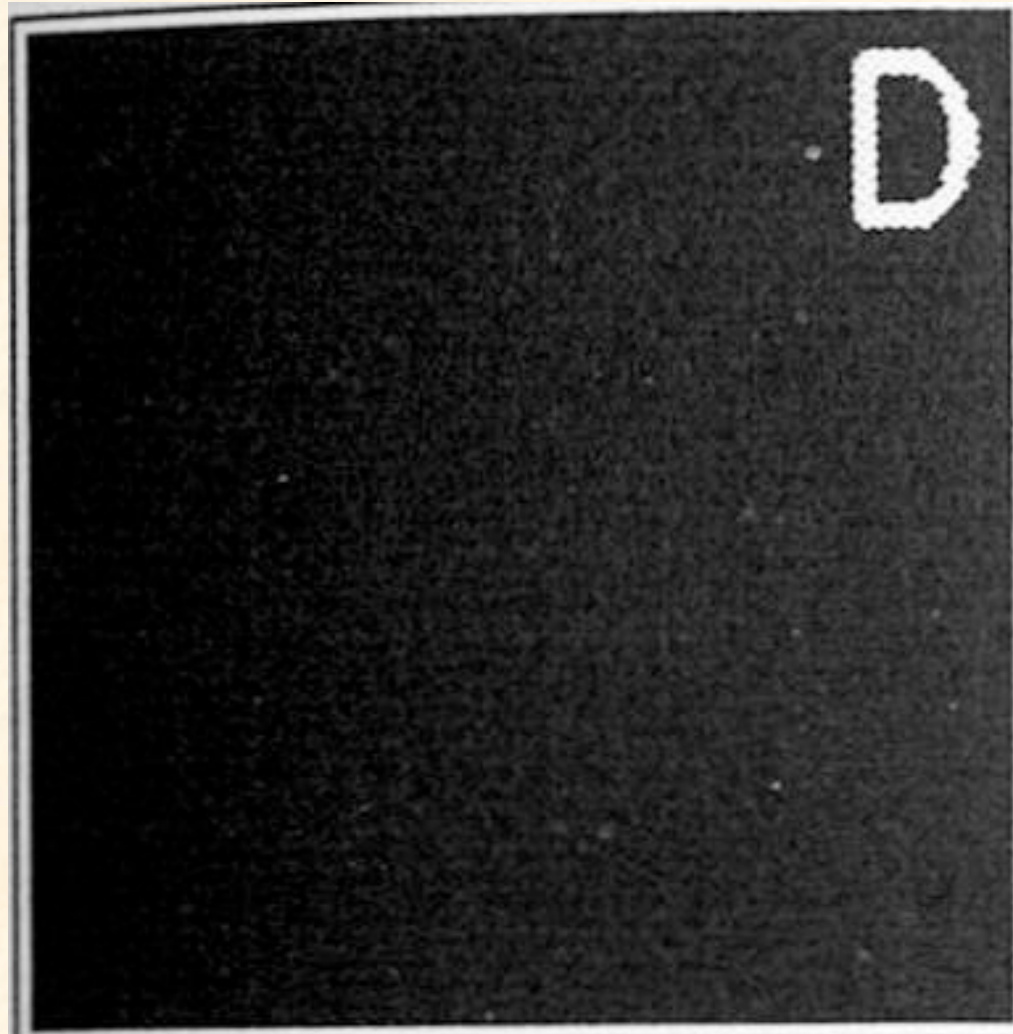


9/30/09

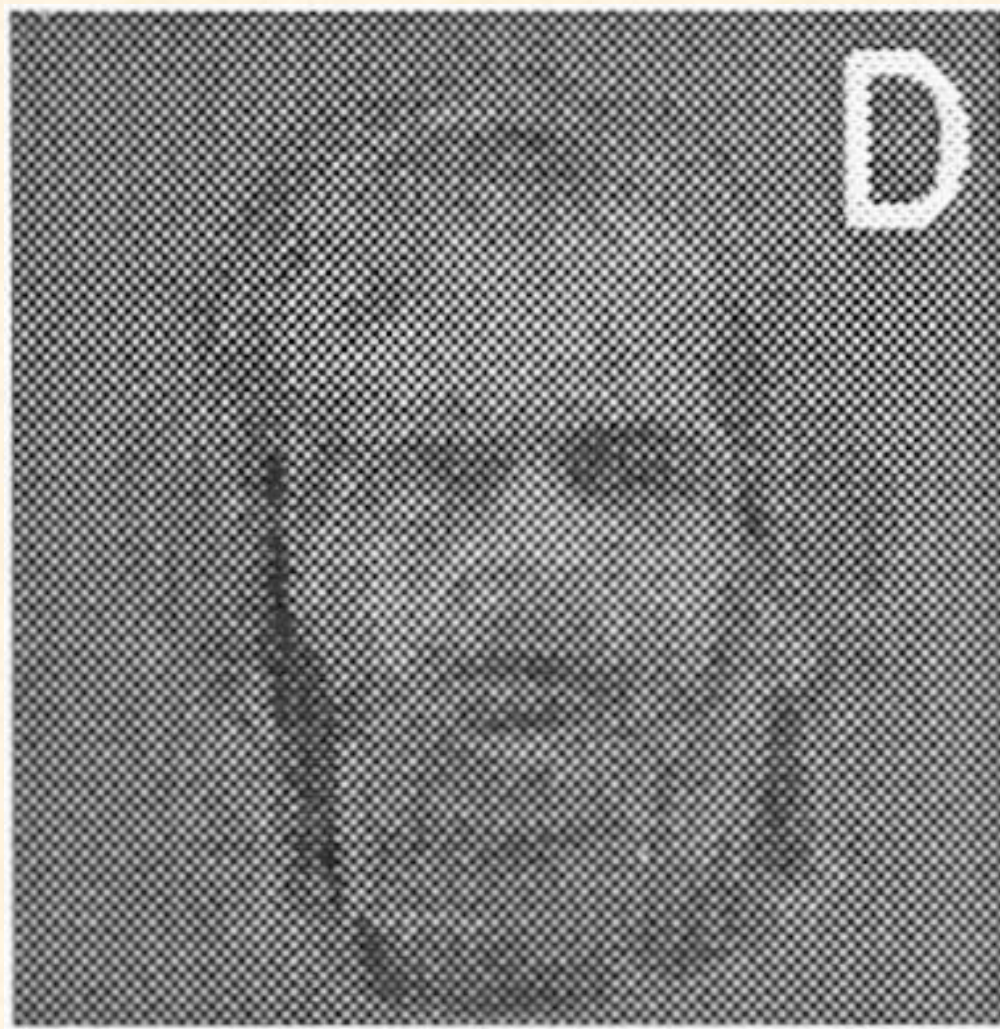
(fig. from Arbib 1995)

50

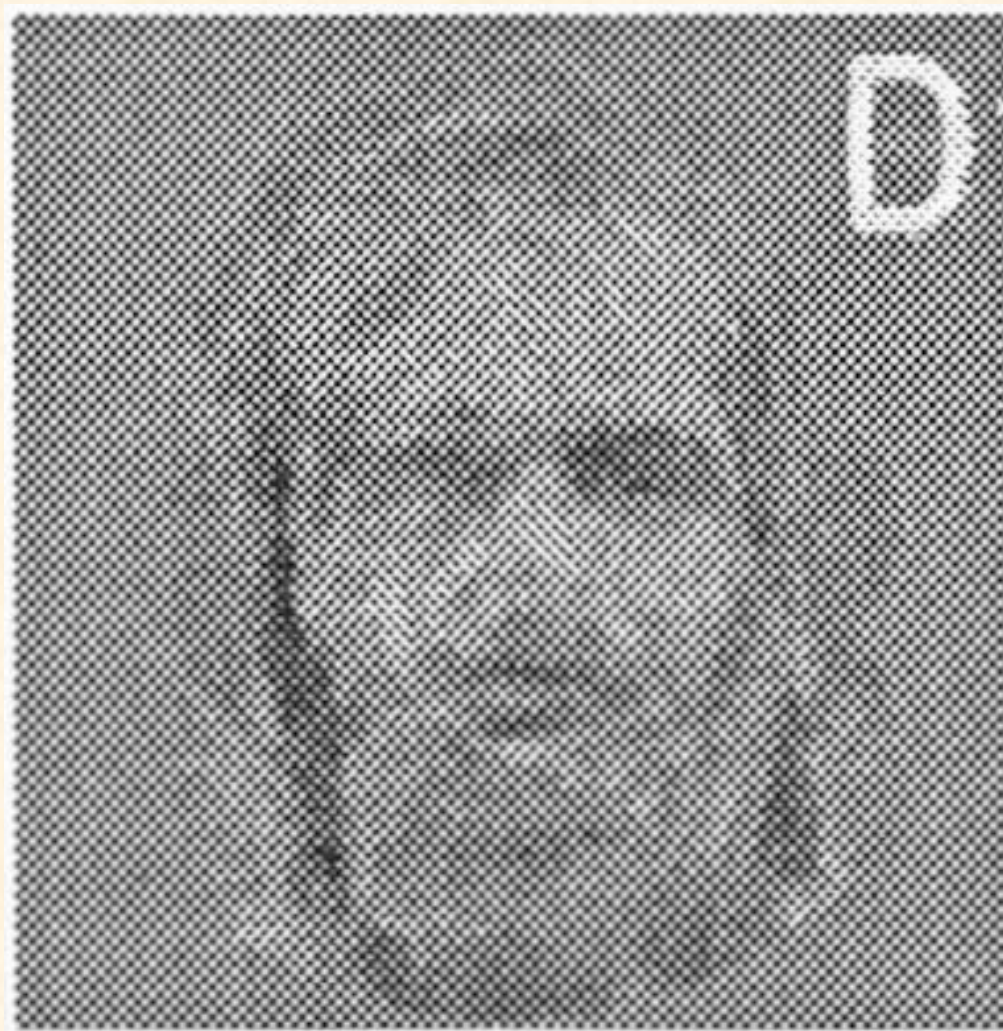
Example of Association



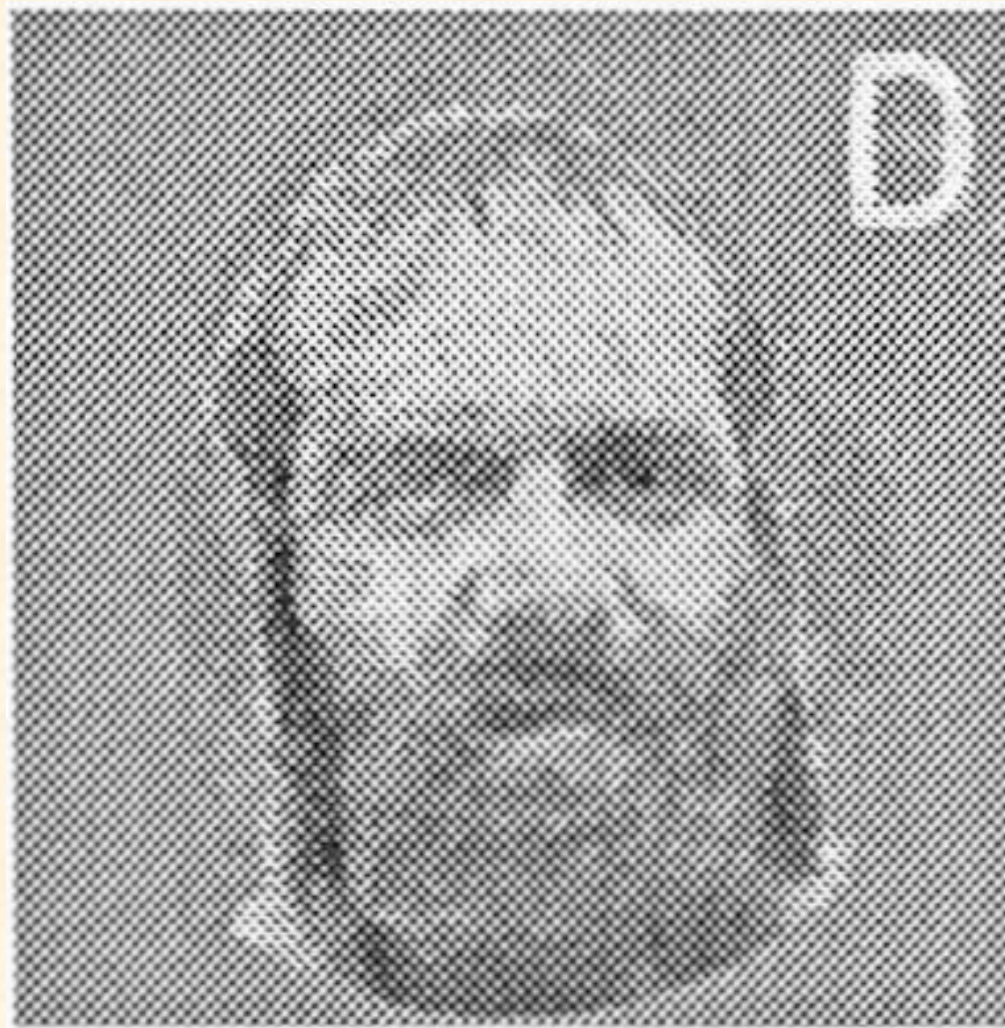
Example of Association



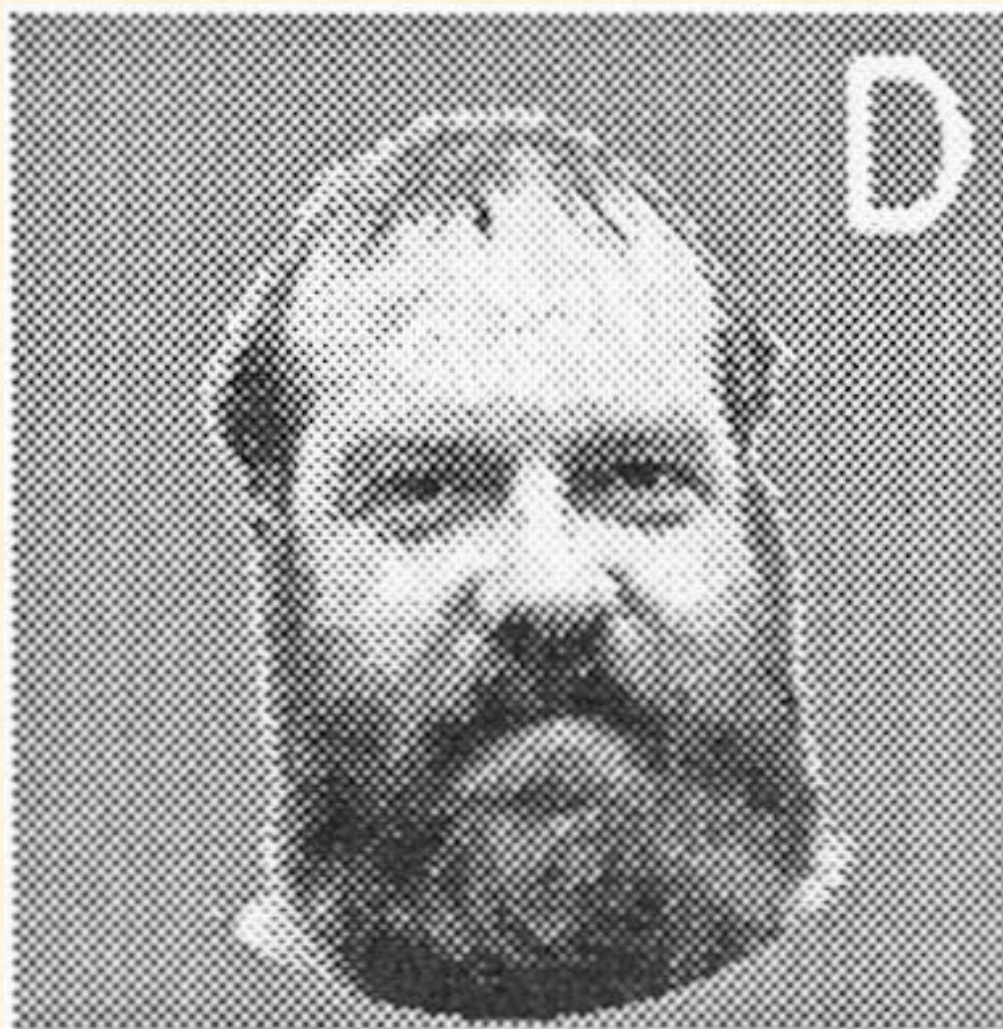
Example of Association



Example of Association



Example of Association



9/30/09

(fig. from Arbib 1995)

55

Applications of Hopfield Memory

- Pattern restoration
- Pattern completion
- Pattern generalization
- Pattern association

Hopfield Net for Optimization and for Associative Memory

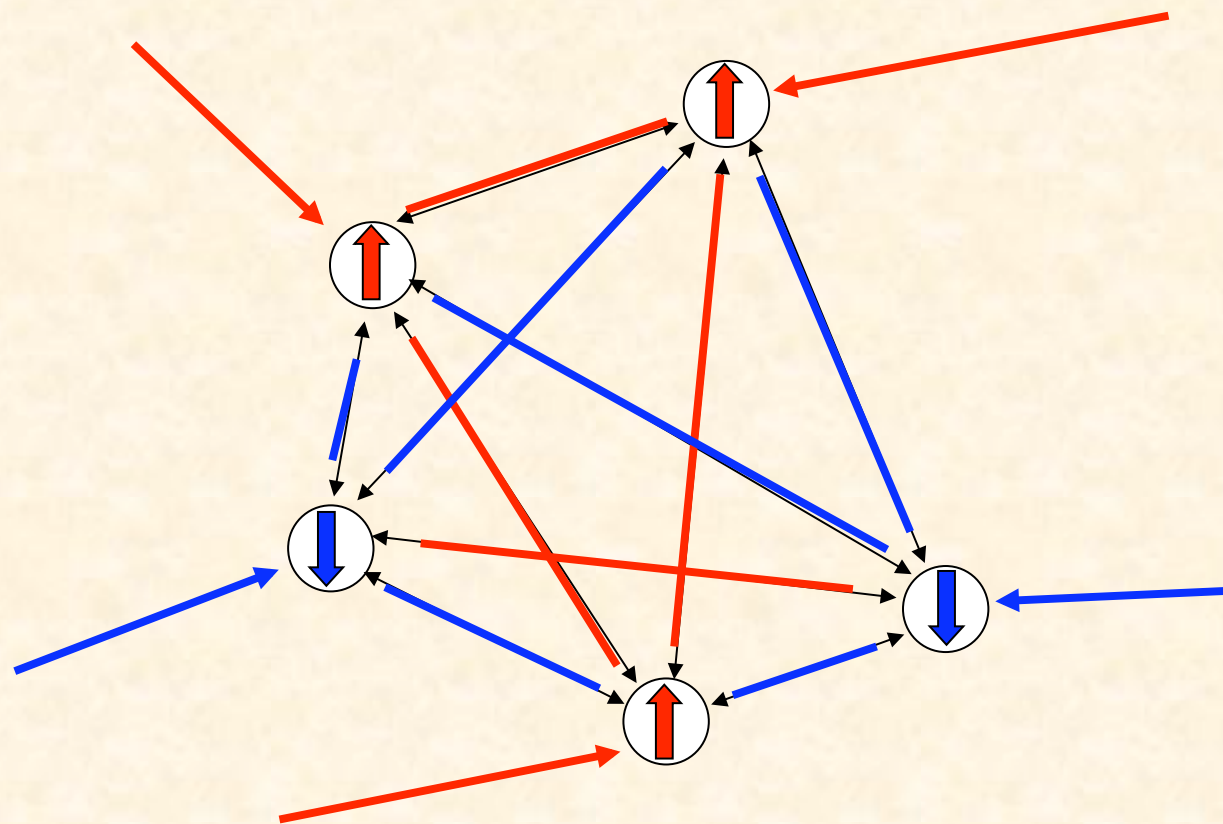
- For optimization:
 - we know the weights (couplings)
 - we want to know the minima (solutions)
- For associative memory:
 - we know the minima (retrieval states)
 - we want to know the weights

Hebb's Rule

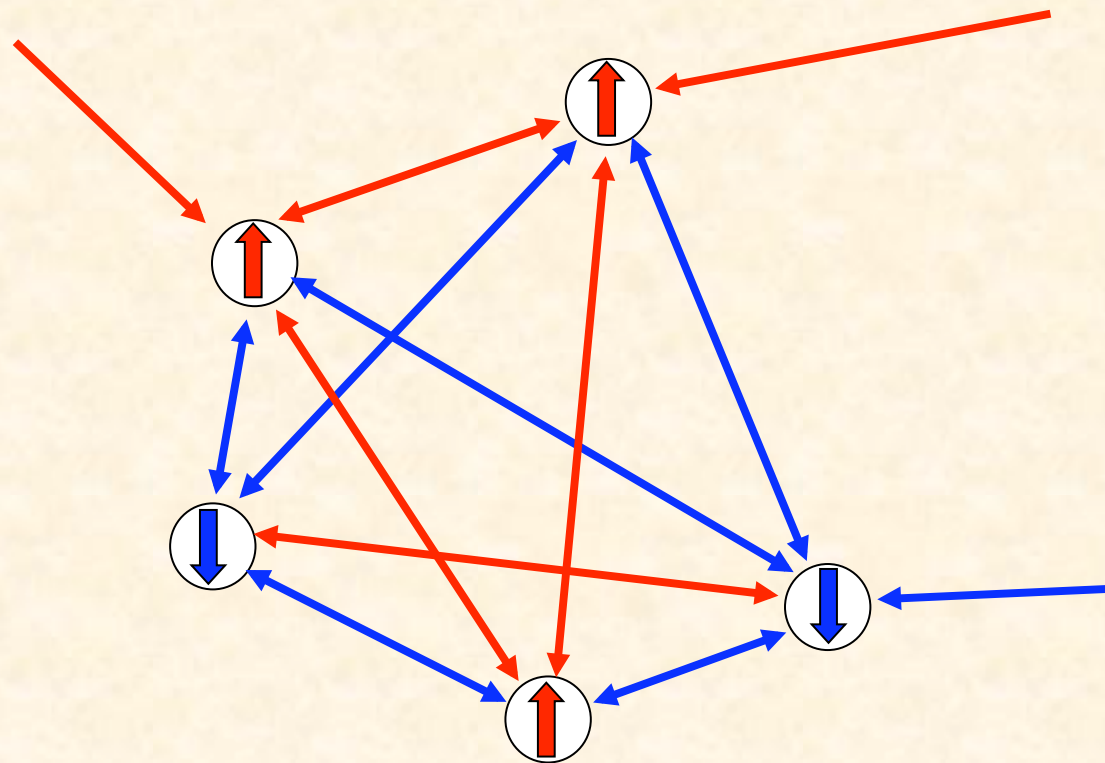
“When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.”

—Donald Hebb (*The Organization of Behavior*, 1949, p. 62)

Example of Hebbian Learning: Pattern Imprinted



Example of Hebbian Learning: Partial Pattern Reconstruction



Mathematical Model of Hebbian Learning for One Pattern

$$\text{Let } W_{ij} = \begin{cases} x_i x_j, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases}$$

$$\text{Since } x_i x_i = x_i^2 = 1, \quad \mathbf{W} = \mathbf{xx}^T - \mathbf{I}$$

For simplicity, we will include self-coupling:

$$\mathbf{W} = \mathbf{xx}^T$$

A Single Imprinted Pattern is a Stable State

- Suppose $\mathbf{W} = \mathbf{x}\mathbf{x}^T$
- Then $\mathbf{h} = \mathbf{W}\mathbf{x} = \mathbf{x}\mathbf{x}^T\mathbf{x} = n\mathbf{x}$

since

$$\mathbf{x}^T \mathbf{x} = \sum_{i=1}^n x_i^2 = \sum_{i=1}^n (\pm 1)^2 = n$$

- Hence, if initial state is $\mathbf{s} = \mathbf{x}$, then new state is $\mathbf{s}' = \text{sgn}(n\mathbf{x}) = \mathbf{x}$
- May be other stable states (e.g., $-\mathbf{x}$)

Questions

- How big is the basin of attraction of the imprinted pattern?
- How many patterns can be imprinted?
- Are there unneeded *spurious* stable states?
- These issues will be addressed in the context of multiple imprinted patterns

Imprinting Multiple Patterns

- Let $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^p$ be patterns to be imprinted
- Define the sum-of-outer-products matrix:

$$W_{ij} = \frac{1}{n} \sum_{k=1}^p x_i^k x_j^k$$

$$\mathbf{W} = \frac{1}{n} \sum_{k=1}^p \mathbf{x}^k (\mathbf{x}^k)^T$$

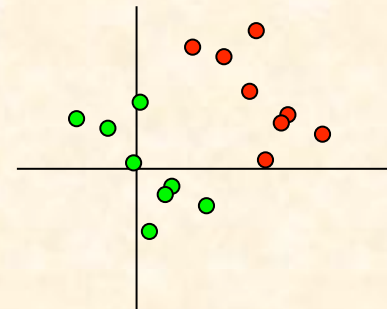
Definition of Covariance

Consider samples $(x^1, y^1), (x^2, y^2), \dots, (x^N, y^N)$

Let $\bar{x} = \langle x^k \rangle$ and $\bar{y} = \langle y^k \rangle$

Covariance of x and y values :

$$\begin{aligned} C_{xy} &= \langle (x^k - \bar{x})(y^k - \bar{y}) \rangle \\ &= \langle x^k y^k - \bar{x} y^k - x^k \bar{y} + \bar{x} \cdot \bar{y} \rangle \\ &= \langle x^k y^k \rangle - \bar{x} \langle y^k \rangle - \langle x^k \rangle \bar{y} + \bar{x} \cdot \bar{y} \\ &= \langle x^k y^k \rangle - \bar{x} \cdot \bar{y} - \bar{x} \cdot \bar{y} + \bar{x} \cdot \bar{y} \\ C_{xy} &= \langle x^k y^k \rangle - \bar{x} \cdot \bar{y} \end{aligned}$$



Weights & the Covariance Matrix

Sample pattern vectors: $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^p$

Covariance of i^{th} and j^{th} components:

$$C_{ij} = \langle x_i^k x_j^k \rangle - \bar{x}_i \cdot \bar{x}_j$$

If $\forall i : \bar{x}_i = 0$ (± 1 equally likely in all positions):

$$C_{ij} = \langle x_i^k x_j^k \rangle = \frac{1}{p} \sum_{k=1}^p x_i^k y_j^k$$

$$\therefore \mathbf{W} = \frac{p}{n} \mathbf{C}$$

Characteristics of Hopfield Memory

- Distributed (“holographic”)
 - every pattern is stored in every location (weight)
- Robust
 - correct retrieval in spite of noise or error in patterns
 - correct operation in spite of considerable weight damage or noise

Demonstration of Hopfield Net





Run Malasri Hopfield Demo

Stability of Imprinted Memories

- Suppose the state is one of the imprinted patterns \mathbf{x}^m

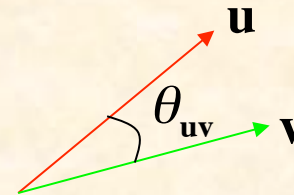
- Then:
$$\begin{aligned}\mathbf{h} &= \mathbf{W}\mathbf{x}^m = \left[\frac{1}{n} \sum_k \mathbf{x}^k (\mathbf{x}^k)^T \right] \mathbf{x}^m \\ &= \frac{1}{n} \sum_k \mathbf{x}^k (\mathbf{x}^k)^T \mathbf{x}^m \\ &= \frac{1}{n} \mathbf{x}^m (\mathbf{x}^m)^T \mathbf{x}^m + \frac{1}{n} \sum_{k \neq m} \mathbf{x}^k (\mathbf{x}^k)^T \mathbf{x}^m \\ &= \mathbf{x}^m + \frac{1}{n} \sum_{k \neq m} (\mathbf{x}^k \cdot \mathbf{x}^m) \mathbf{x}^k\end{aligned}$$

Interpretation of Inner Products

- $\mathbf{x}^k \cdot \mathbf{x}^m = n$ if they are identical 
 - highly correlated
- $\mathbf{x}^k \cdot \mathbf{x}^m = -n$ if they are complementary 
 - highly correlated (reversed)
- $\mathbf{x}^k \cdot \mathbf{x}^m = 0$ if they are orthogonal 
 - largely uncorrelated
- $\mathbf{x}^k \cdot \mathbf{x}^m$ measures the *crosstalk* between patterns k and m 

Cosines and Inner products

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos \theta_{\mathbf{uv}}$$



If \mathbf{u} is bipolar, then $\|\mathbf{u}\|^2 = \mathbf{u} \cdot \mathbf{u} = n$

Hence, $\mathbf{u} \cdot \mathbf{v} = \sqrt{n} \sqrt{n} \cos \theta_{\mathbf{uv}} = n \cos \theta_{\mathbf{uv}}$

Hence $\mathbf{h} = \mathbf{x}^m + \sum_{k \neq m} \mathbf{x}^k \cos \theta_{km}$

Conditions for Stability

Stability of entire pattern :

$$\mathbf{x}^m = \text{sgn} \left(\mathbf{x}^m + \sum_{k \neq m} \mathbf{x}^k \cos \theta_{km} \right)$$

Stability of a single bit :

$$x_i^m = \text{sgn} \left(x_i^m + \sum_{k \neq m} x_i^k \cos \theta_{km} \right)$$

Sufficient Conditions for Instability (Case 1)

Suppose $x_i^m = -1$. Then unstable if :

$$(-1) + \sum_{k \neq m} x_i^k \cos \theta_{km} > 0$$

$$\sum_{k \neq m} x_i^k \cos \theta_{km} > 1$$

Sufficient Conditions for Instability (Case 2)

Suppose $x_i^m = +1$. Then unstable if :

$$(+1) + \sum_{k \neq m} x_i^k \cos \theta_{km} < 0$$

$$\sum_{k \neq m} x_i^k \cos \theta_{km} < -1$$

Sufficient Conditions for Stability

$$\left| \sum_{k \neq m} x_i^k \cos \theta_{km} \right| \leq 1$$

The crosstalk with the sought pattern must be sufficiently small

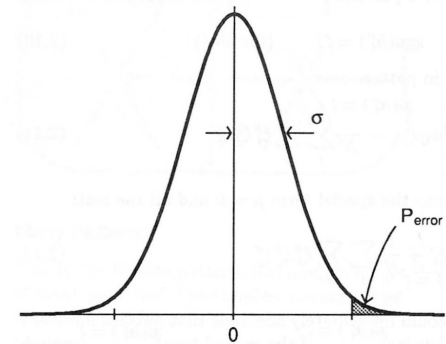
Capacity of Hopfield Memory

- Depends on the patterns imprinted
- If orthogonal, $p_{\max} = n$
 - but every state is stable \Rightarrow trivial basins
- So $p_{\max} < n$
- Let **load parameter** $\alpha = p / n$

Single Bit Stability Analysis

- For simplicity, suppose \mathbf{x}^k are random
- Then $\mathbf{x}^k \cdot \mathbf{x}^m$ are sums of n random ± 1
 - binomial distribution \approx Gaussian
 - in range $-n, \dots, +n$
 - with mean $\mu = 0$
 - and variance $\sigma^2 = n$
- Probability sum $> t$:

$$\frac{1}{2} \left[1 - \operatorname{erf} \left(\frac{t}{\sqrt{2n}} \right) \right]$$



[See “Review of Gaussian (Normal) Distributions” on course website]

Approximation of Probability

$$\text{Let crosstalk } C_i^m = \frac{1}{n} \sum_{k \neq m} x_i^k (\mathbf{x}^k \cdot \mathbf{x}^m)$$

$$\text{We want } \Pr\{C_i^m > 1\} = \Pr\{nC_i^m > n\}$$

$$\text{Note: } nC_i^m = \sum_{\substack{k=1 \\ k \neq m}}^p \sum_{j=1}^n x_i^k x_j^k x_j^m$$

A sum of $n(p-1) \approx np$ random ± 1 s

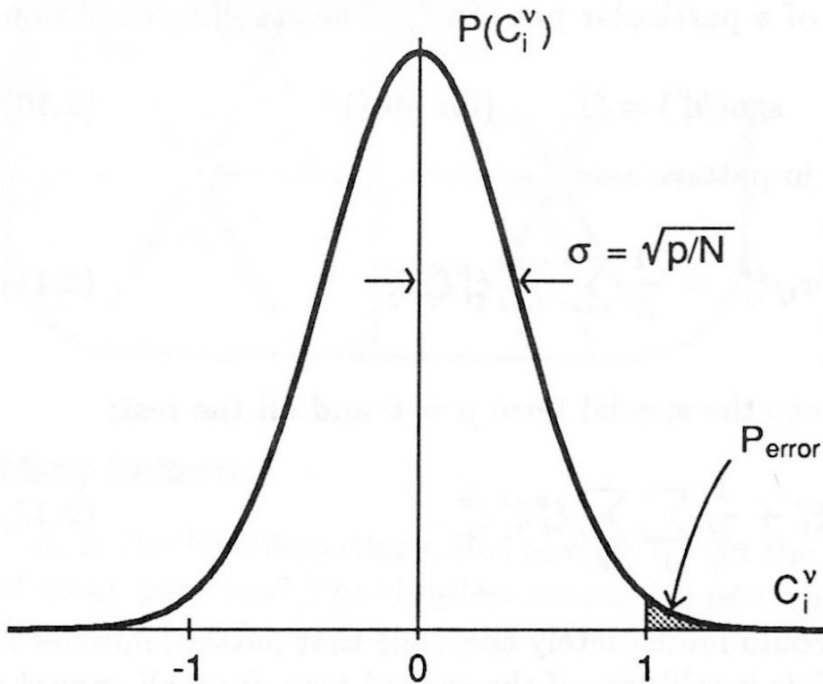
$$\text{Variance } \sigma^2 = np$$

Probability of Bit Instability

$$\Pr\{nC_i^m > n\} = \frac{1}{2} \left[1 - \operatorname{erf}\left(\frac{n}{\sqrt{2np}}\right) \right]$$

$$= \frac{1}{2} \left[1 - \operatorname{erf}\left(\sqrt{n/2p}\right) \right]$$

$$= \frac{1}{2} \left[1 - \operatorname{erf}\left(\sqrt{1/2\alpha}\right) \right]$$



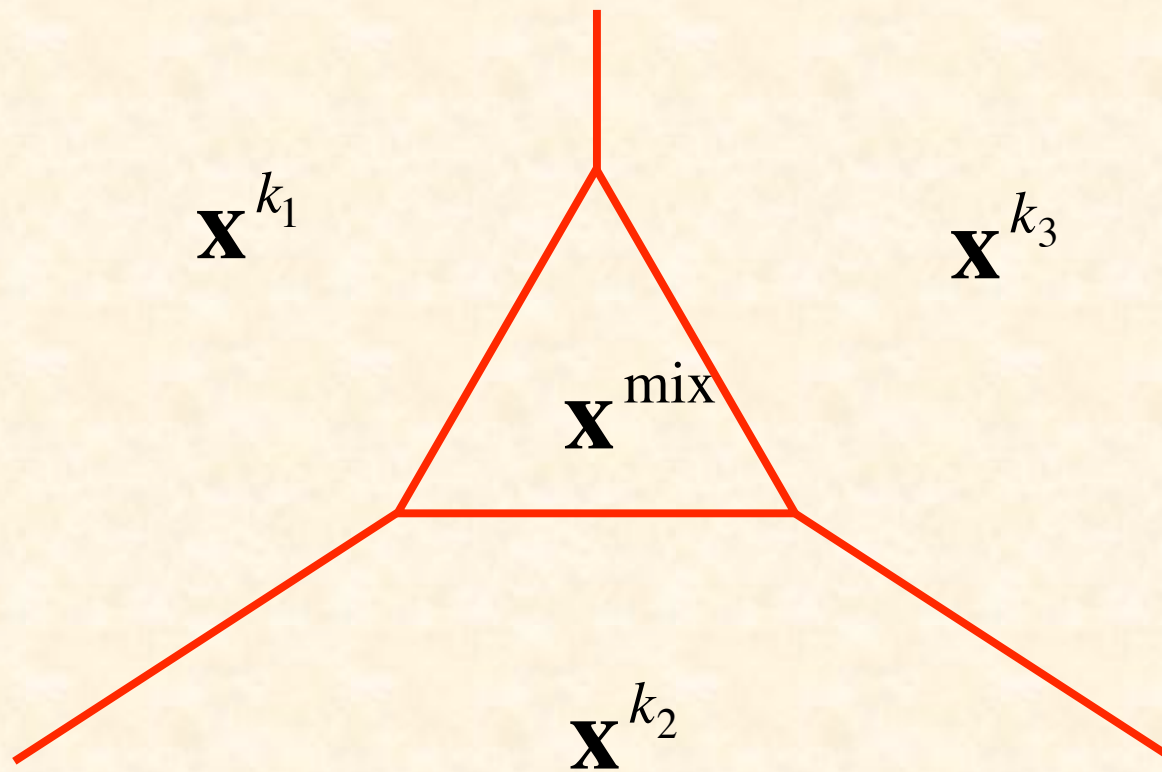
Tabulated Probability of Single-Bit Instability

P_{error}	α
0.1%	0.105
0.36%	0.138
1%	0.185
5%	0.37
10%	0.61

Spurious Attractors

- **Mixture states:**
 - sums or differences of odd numbers of retrieval states
 - number increases combinatorially with p
 - shallower, smaller [basins](#)
 - basins of mixtures swamp basins of retrieval states \Rightarrow overload
 - useful as combinatorial generalizations?
 - self-coupling generates spurious attractors
- **Spin-glass states:**
 - not correlated with any finite number of imprinted patterns
 - occur beyond overload because weights effectively random

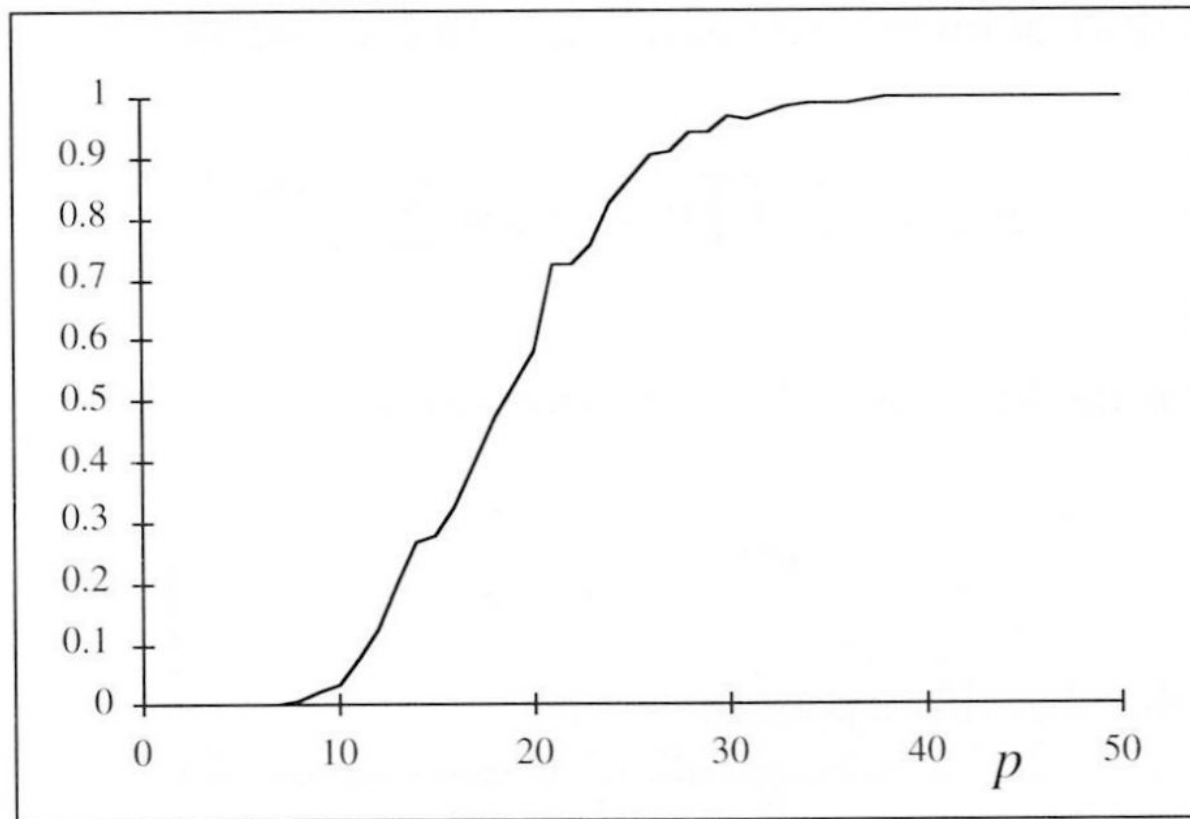
Basins of Mixture States



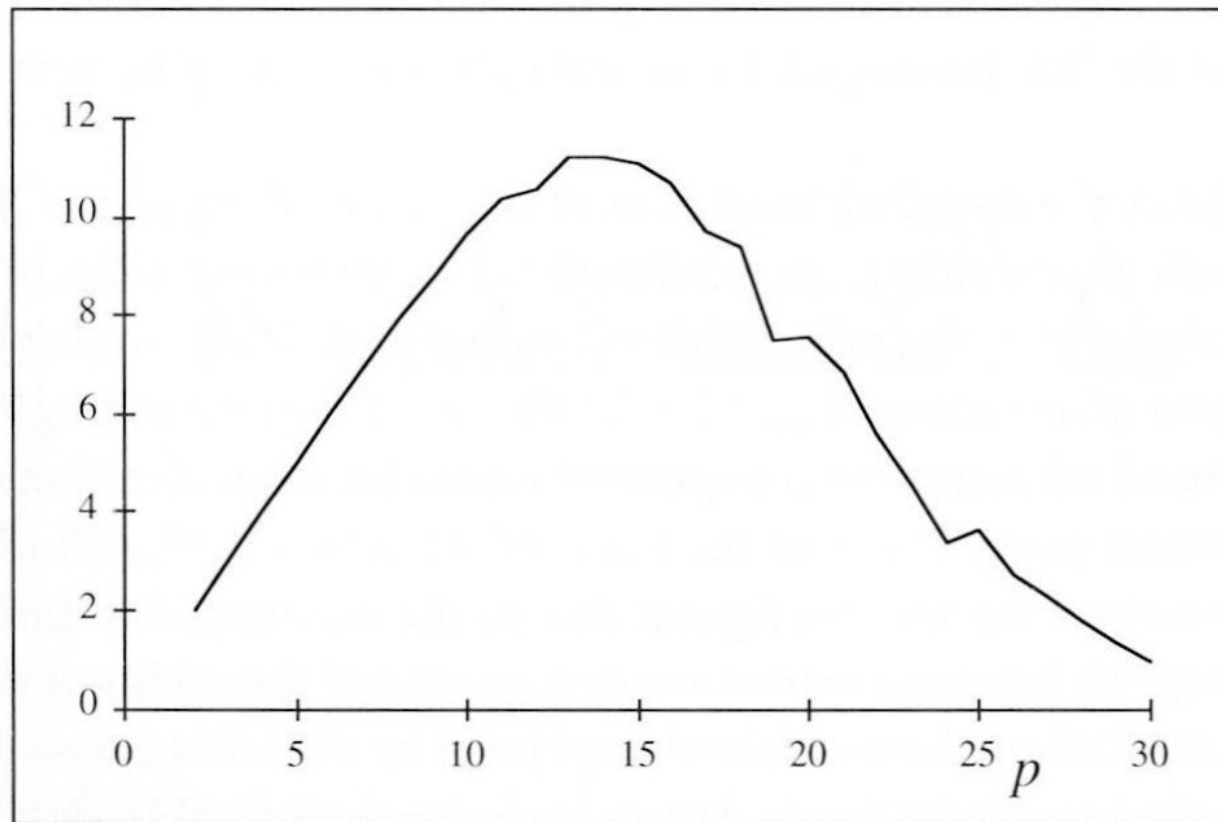
$$x_i^{\text{mix}} = \text{sgn}(x_i^{k_1} + x_i^{k_2} + x_i^{k_3})$$



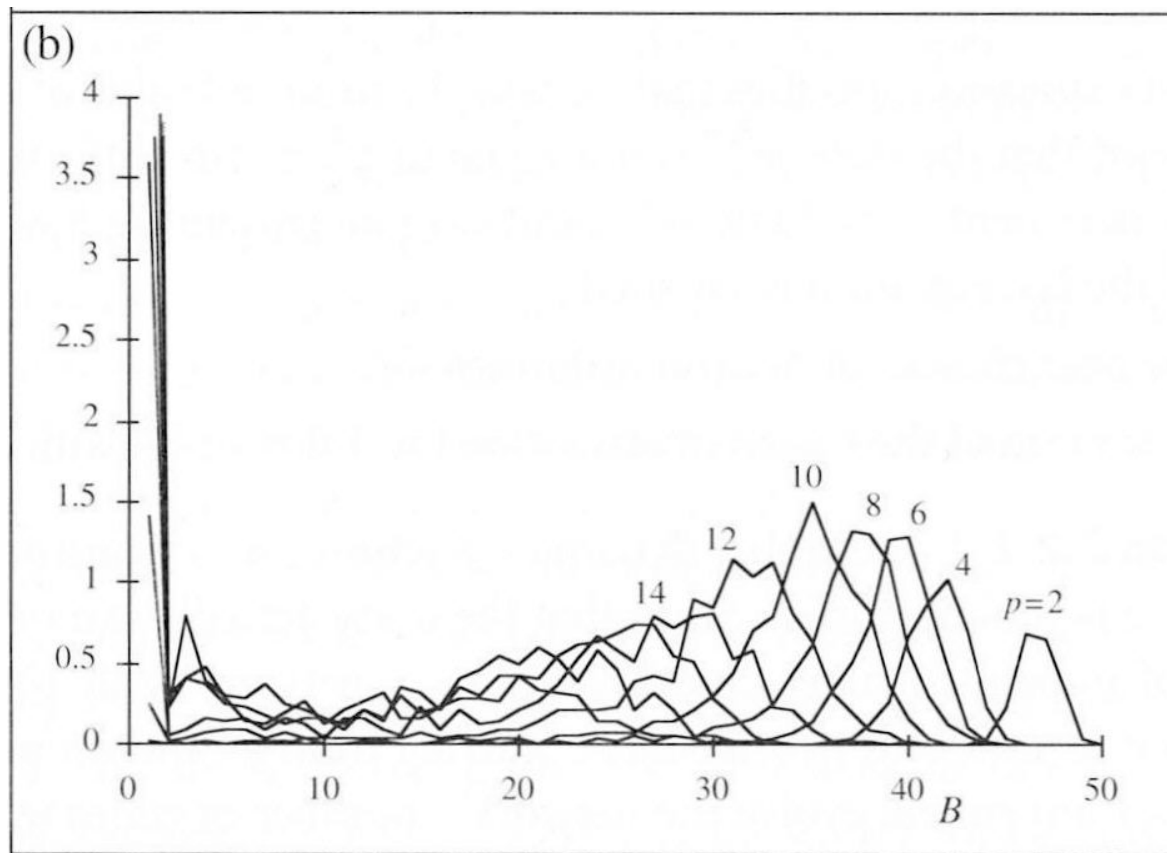
Fraction of Unstable Imprints ($n = 100$)



Number of Stable Imprints ($n = 100$)



Number of Imprints with Basins of Indicated Size ($n = 100$)



Summary of Capacity Results

- Absolute limit: $p_{\max} < \alpha_c n = 0.138 n$
- If a small number of errors in each pattern permitted: $p_{\max} \propto n$
- If all or most patterns must be recalled perfectly: $p_{\max} \propto n / \log n$
- Recall: all this analysis is based on *random* patterns
- Unrealistic, but sometimes can be arranged