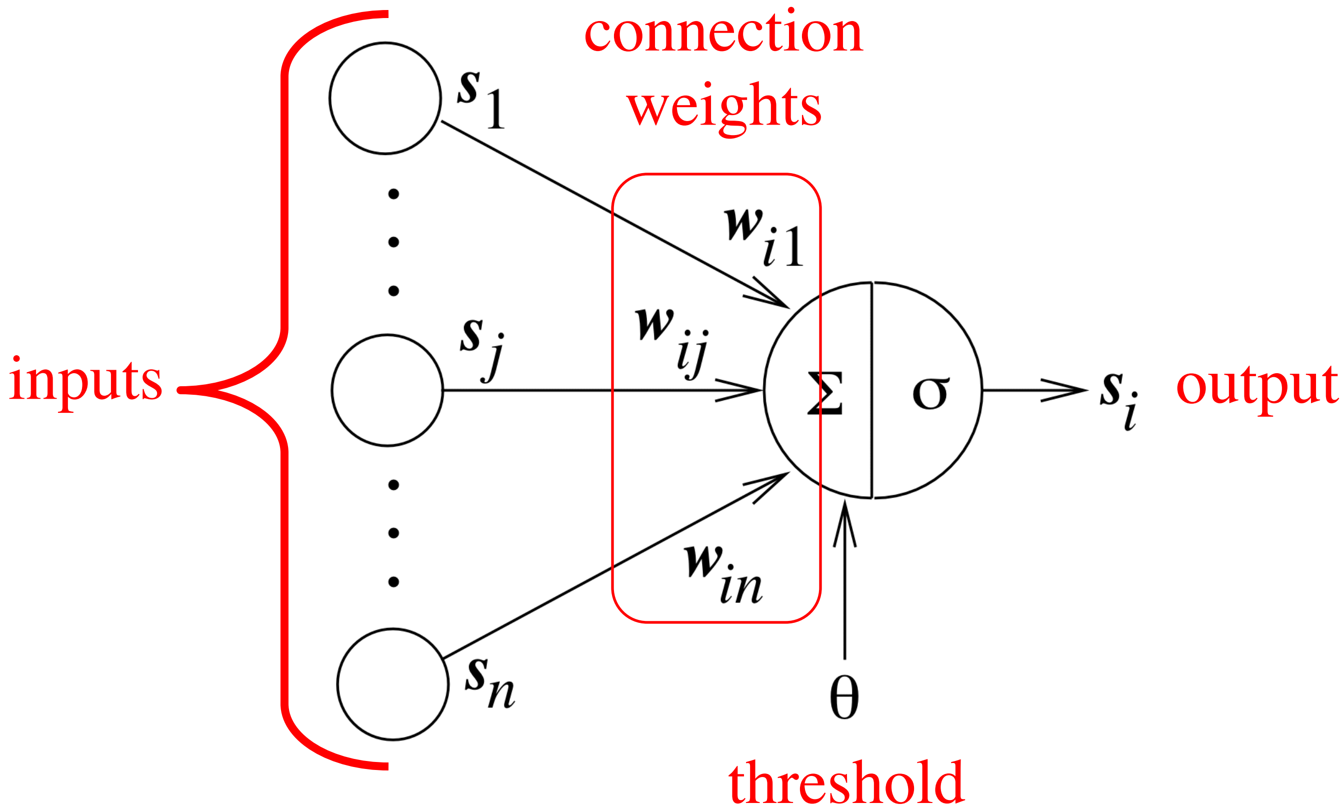
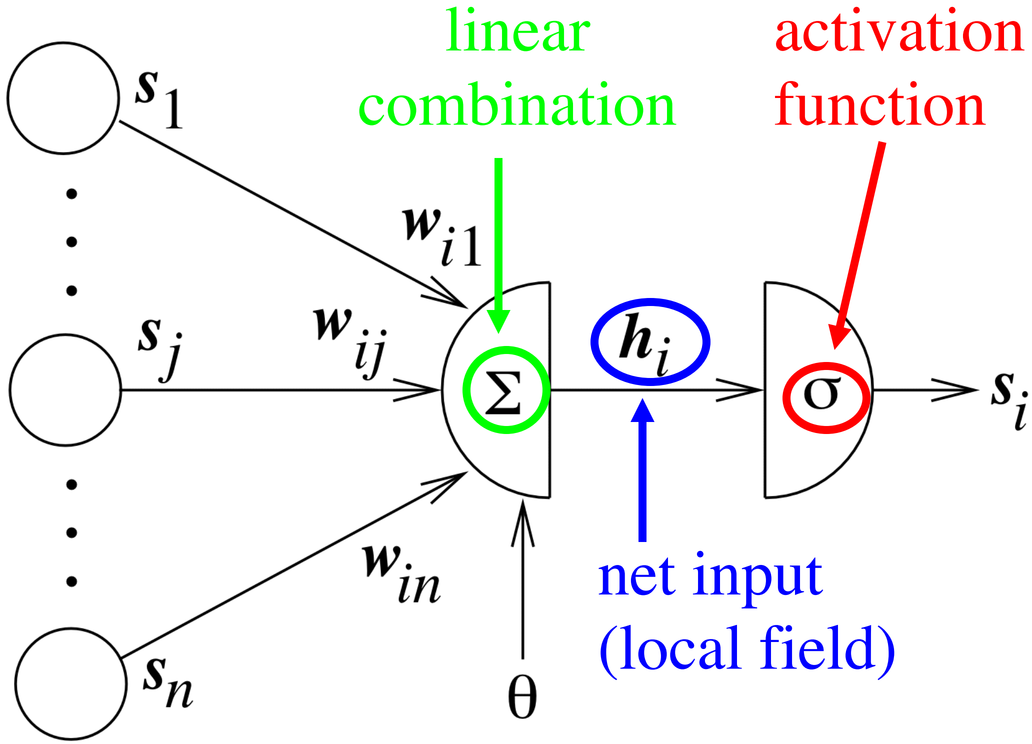
III. Quantum Annealing

A.
The Hopfield Network

Typical Artificial Neuron



Typical Artificial Neuron



Equations

Local field:

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

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

New neural state:

$$s_i' = \sigma(h_i)$$

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

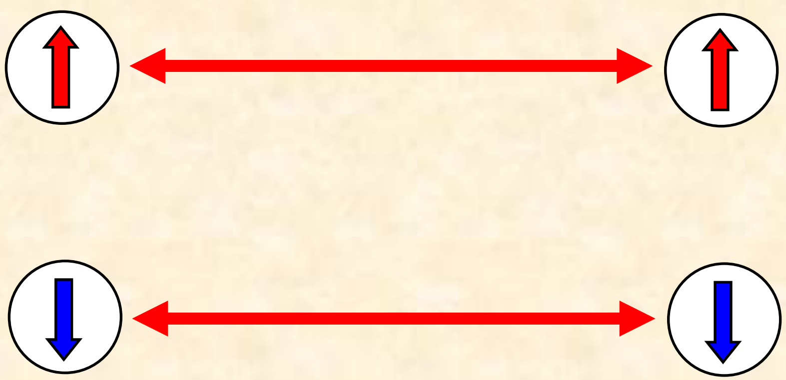
Hopfield Network

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

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

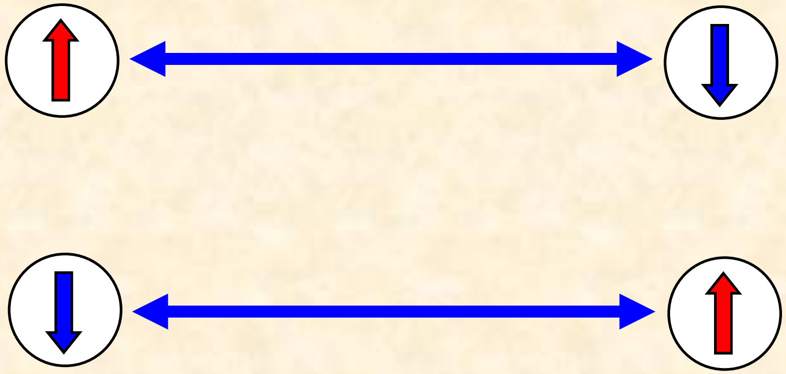
Positive Coupling

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



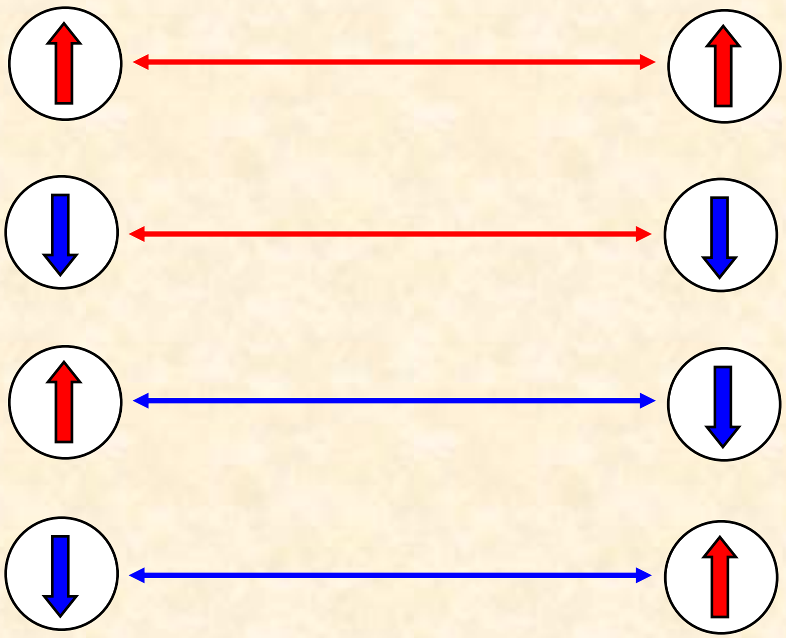
Negative Coupling

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

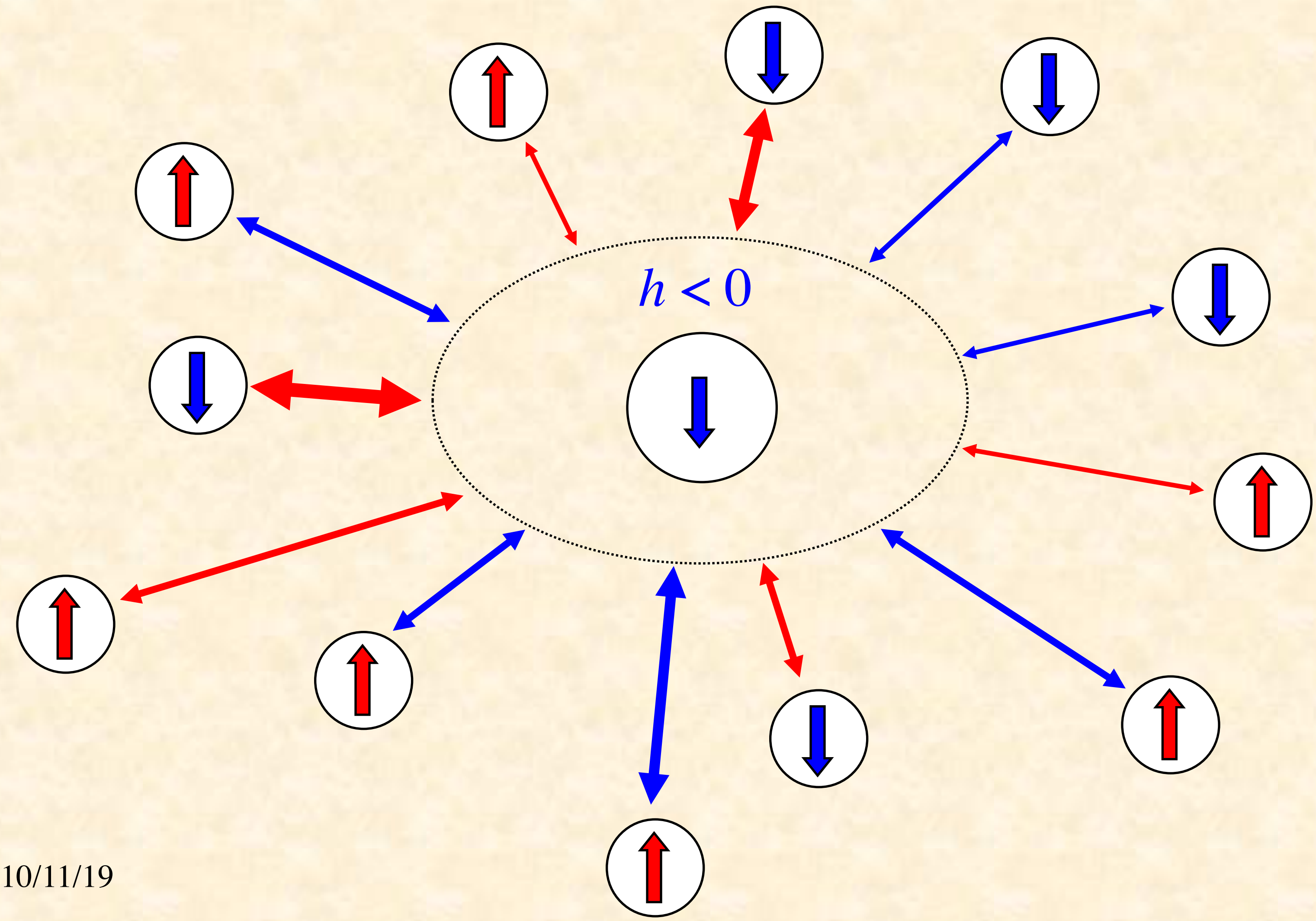


Weak Coupling

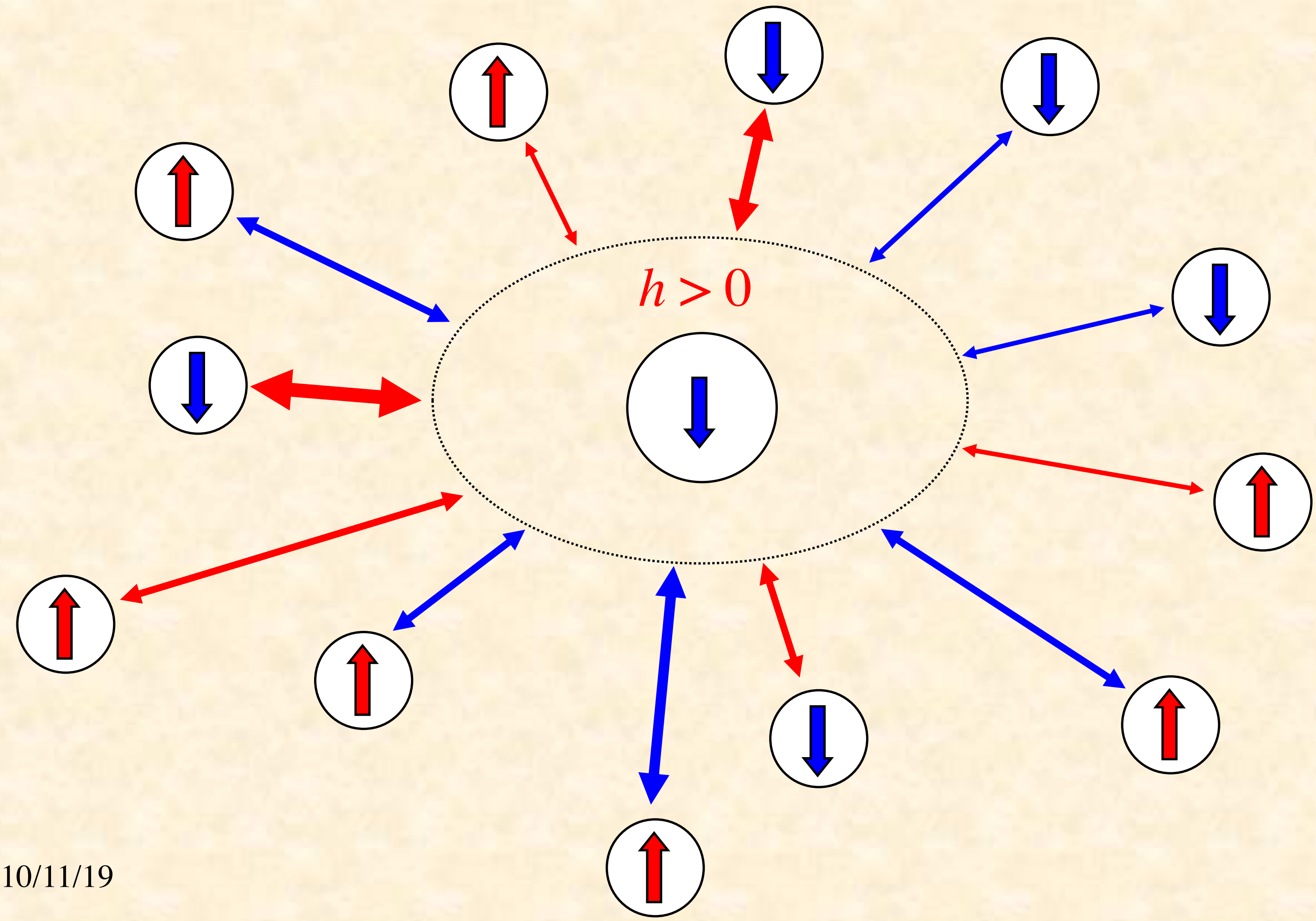
- Either *sense* (sign)
- Little *strength*



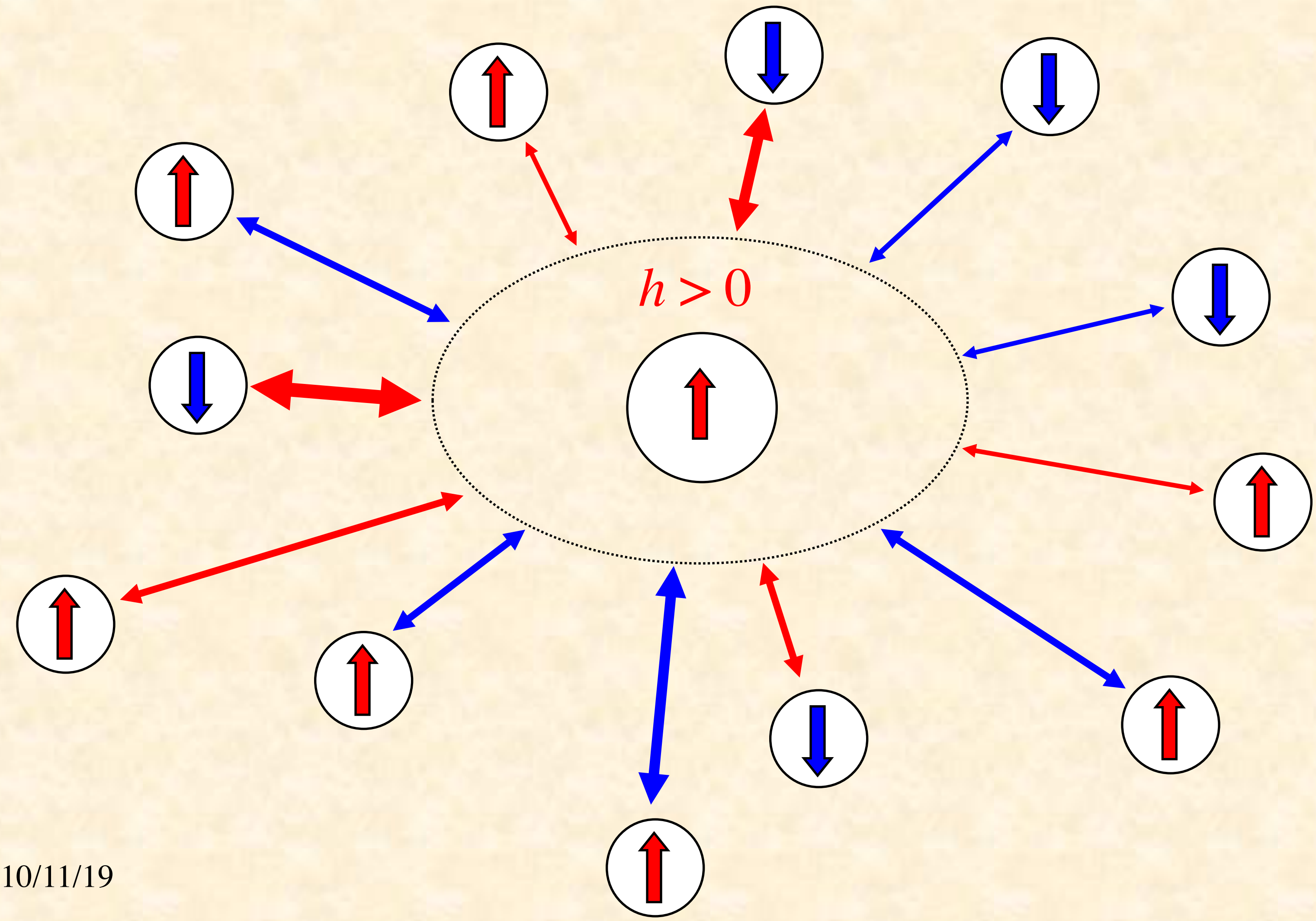
State = –1 & Local Field < 0



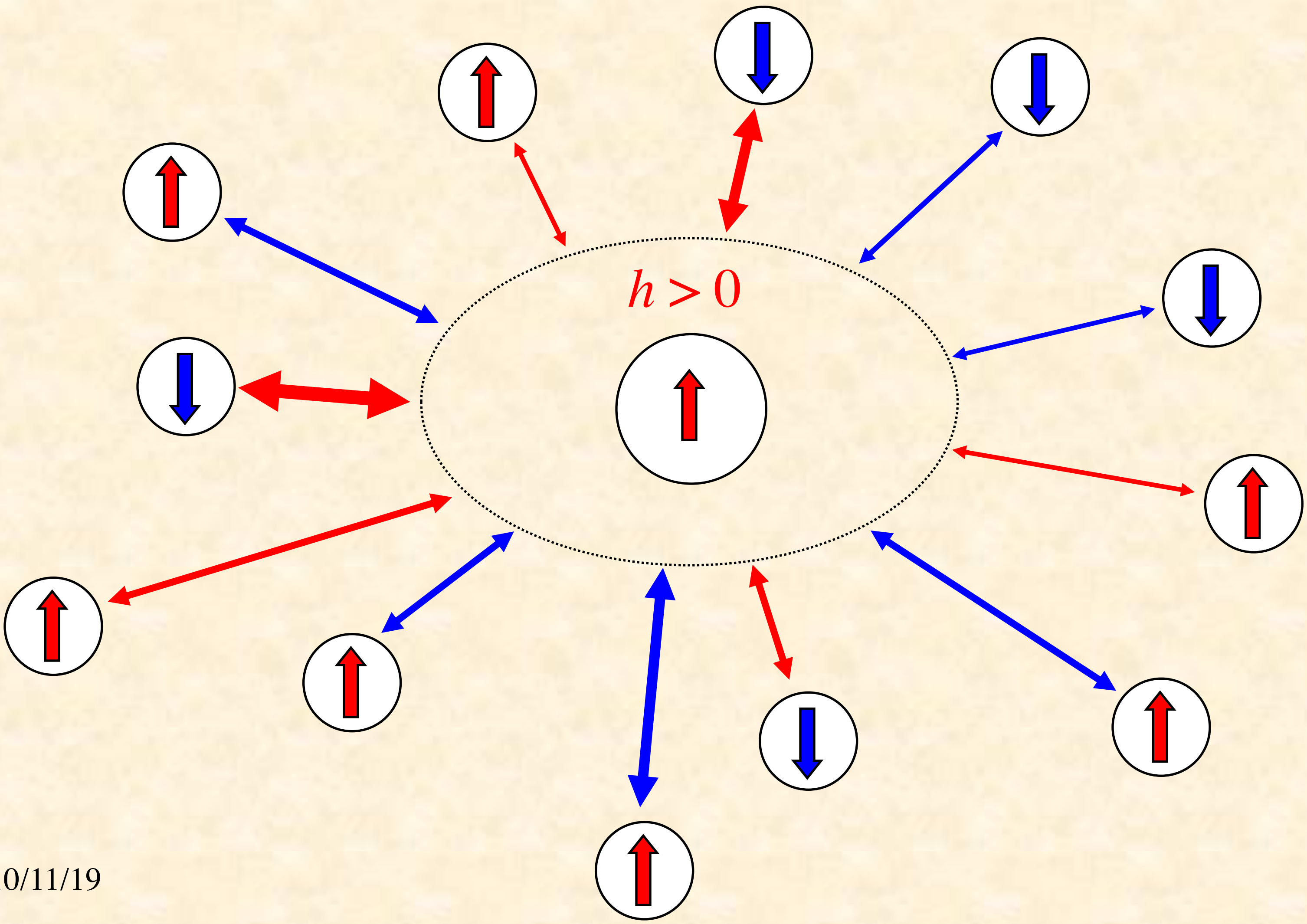
State = –1 & Local Field > 0



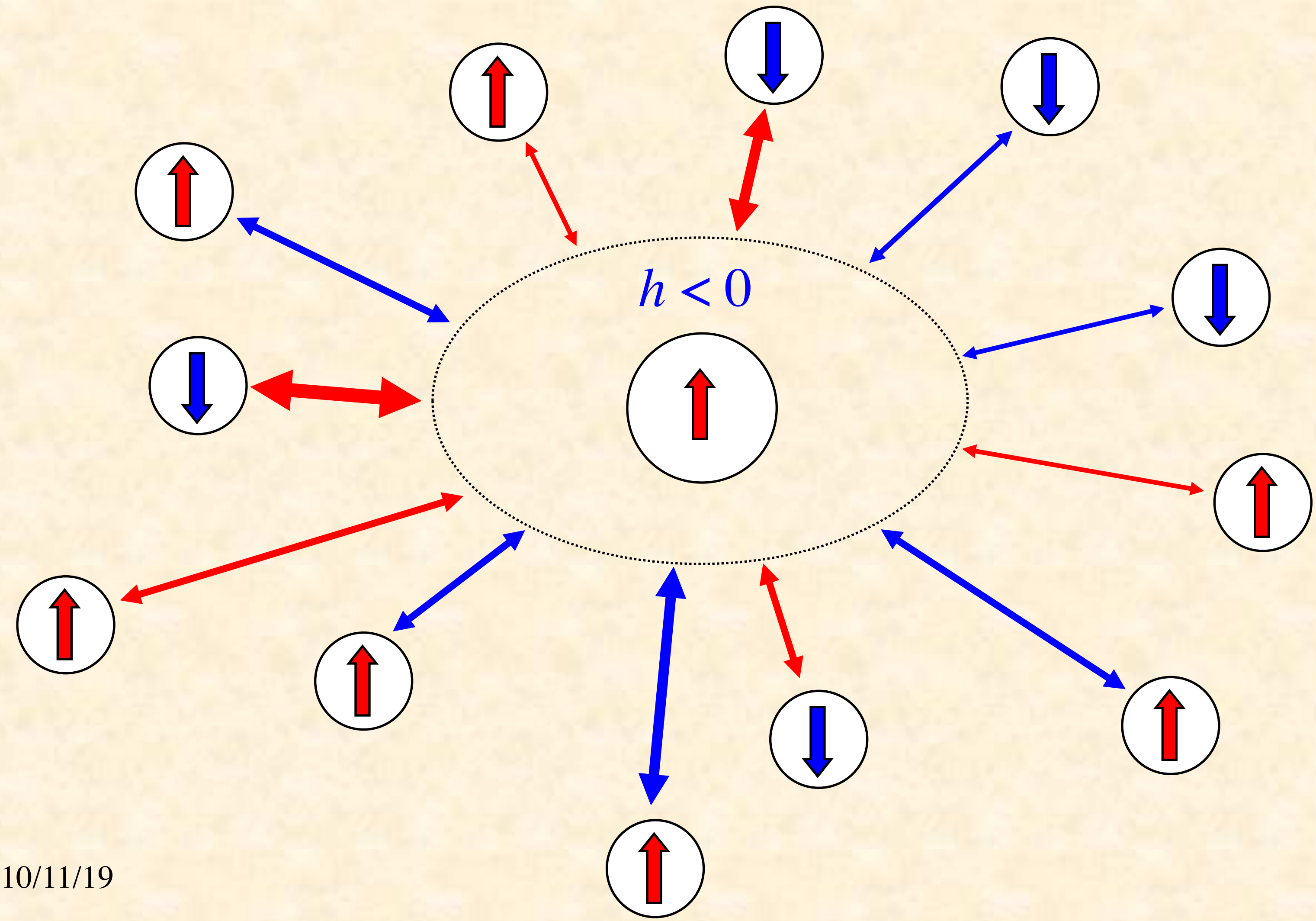
State Reverses



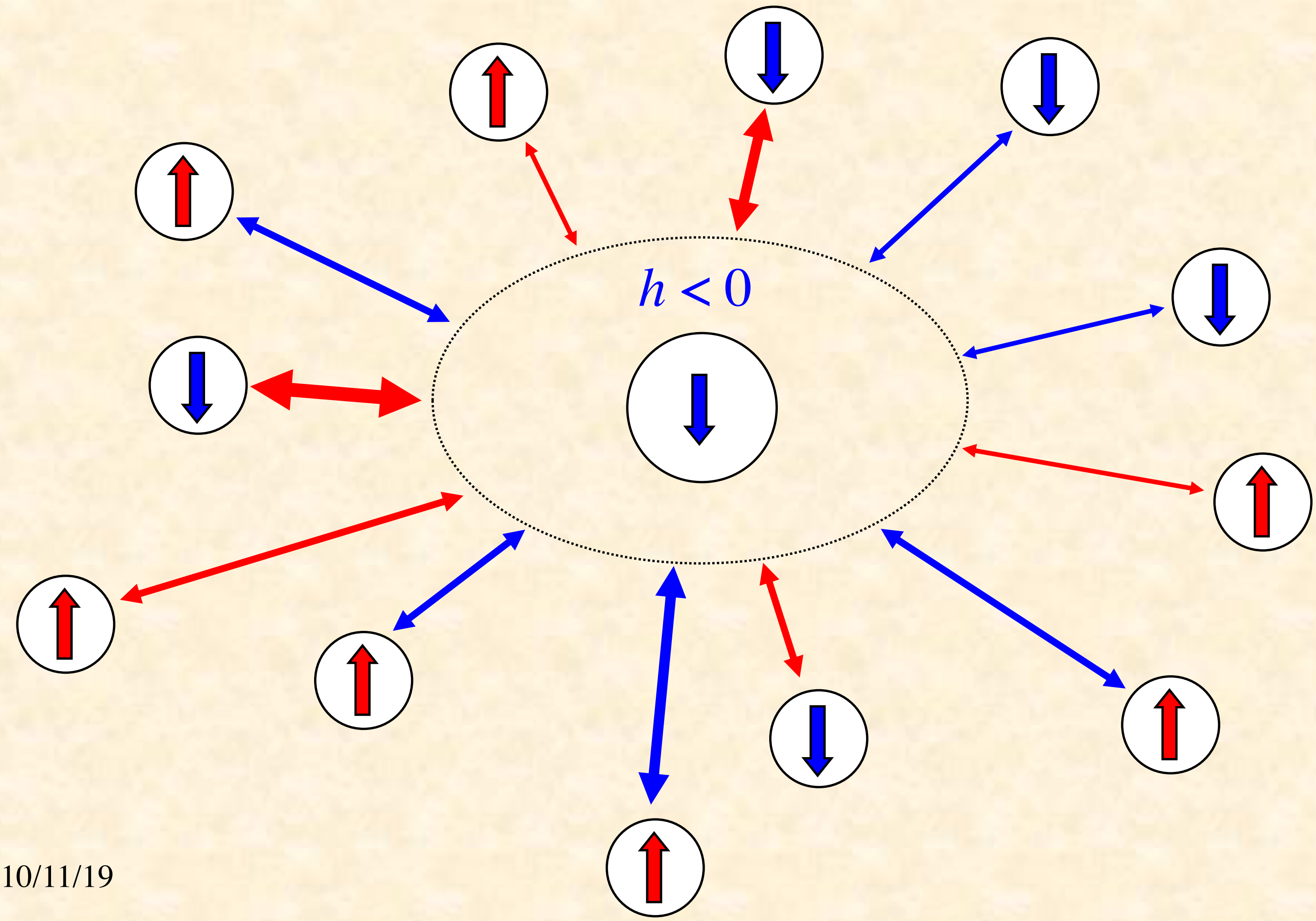
State = +1 & Local Field > 0



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 "tension" T_{ij} between neurons i and j:

$$T_{ij} = -\, s_i\, w_{ij}\, s_j$$

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

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

Total Energy of System

The "energy" of the system is the total "tension" in it:

$$E\{\mathbf{s}\} = \sum_{\langle ij \rangle} T_{ij}$$

$$= -\sum_{\langle ij \rangle} s_i w_{ij} s_j$$

$$= -\frac{1}{2} \sum_i \sum_{j \neq i} s_i W_{ij} s_j$$

$$= -\frac{1}{2} \sum_i \sum_j s_i W_{ij} s_j \text{, if } W_{ij} = 0$$

$$= -\frac{1}{2} \mathbf{s}^{\mathrm{T}} \mathrm{W} \mathbf{s}$$

Another View of Energy

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

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

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:
  $E\{\mathbf{s}(t+1)\} - E\{\mathbf{s}(t)\} \le 0$
- Energy cannot increase
- Energy decreases if any neuron changes
- Must it stop?

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

B.
Hopfield Network for Task Assignment Problem

(and the continuous Hopfield network)

Task Assignment Problem

- Six tasks to be done (I, II, …, VI)
- Six agents to do tasks (A, B, …, F)
- They can do tasks at various rates
  - A (10, 5, 4, 6, 5, 1)
  - B (6, 4, 9, 7, 3, 2)
  - etc
- What is the optimal assignment of tasks to agents?

Continuous Hopfield Net

$$\dot{U}_i = \sum_{j=1}^{n} J_{ij} V_j + B_i - \frac{U_i}{\tau}$$

$$V_i = \sigma(U_i) \in (0,1)$$

Energy function:

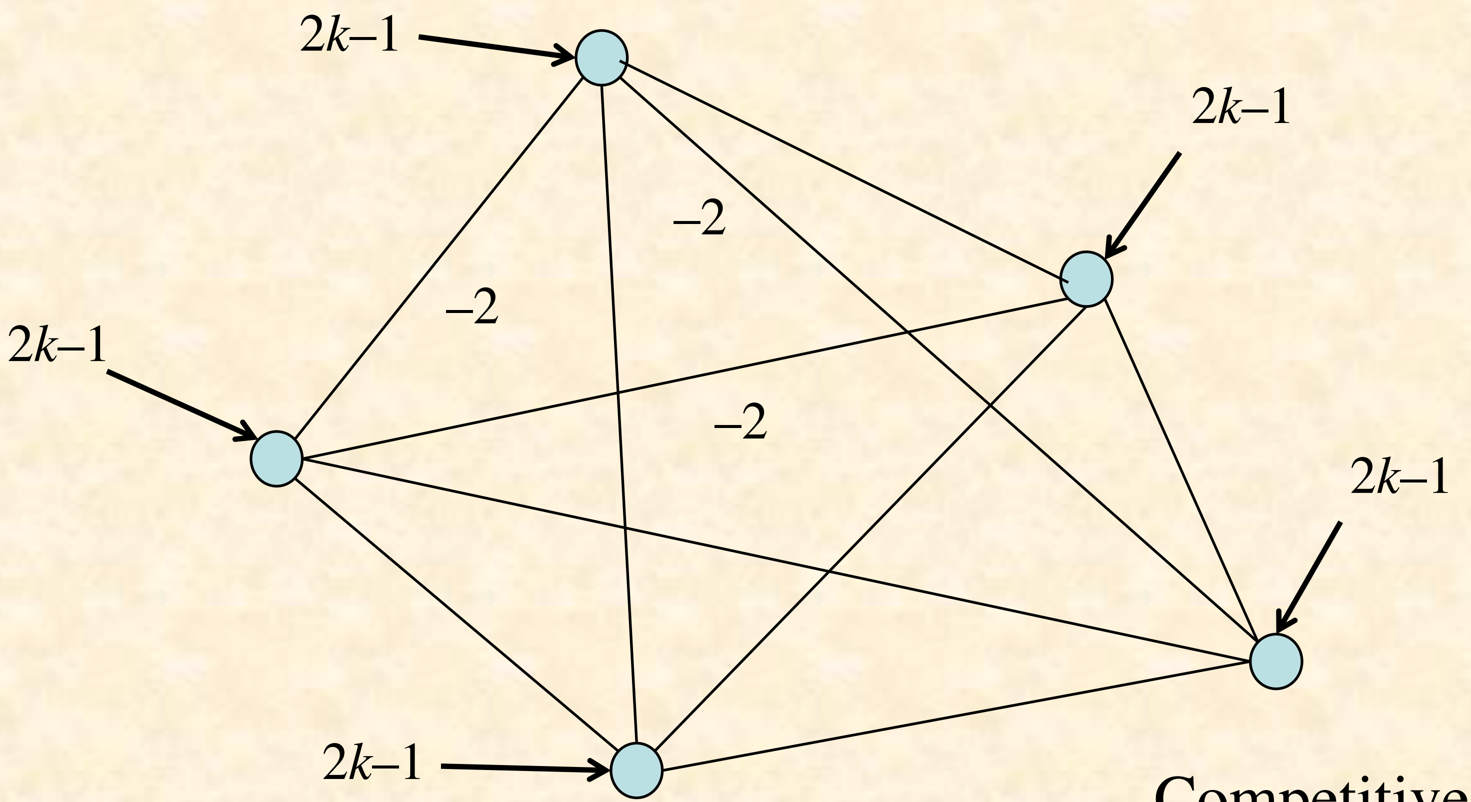
$$E = -\frac{1}{2} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} V_i J_{ij} V_j - \sum_{i=1}^{n} V_i B_i = -\tfrac{1}{2} \mathbf{V}^{\mathrm{T}} \mathbf{J} \mathbf{V} - \mathbf{B}^{\mathrm{T}} \mathbf{V}$$

Derivation of *k*-out-of-*n* Rule

- Suppose we want exactly *k* of *n* neurons = 1
  - That is, $\sum_{i=1}^{n} V_i = k$
- Therefore, minimize $E_o = [k - \sum_{i=1}^{n} V_i]^2$
- Want values of V_i to be integral 0 or 1
- Therefore, minimize $E_c = \sum_{i=1}^{n} V_i(1 - V_i)$
- Minimize total energy function:
  $E = [k - \sum_{i=1}^{n} V_i]^2 + \sum_{i=1}^{n} V_i(1 - V_i)$
- Rearrange to get:

$$E = -\frac{1}{2} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} V_i(-2)V_j - \sum_{i=1}^{n} V_i(2k - 1)$$

k-out-of-n Rule



Competitive Network

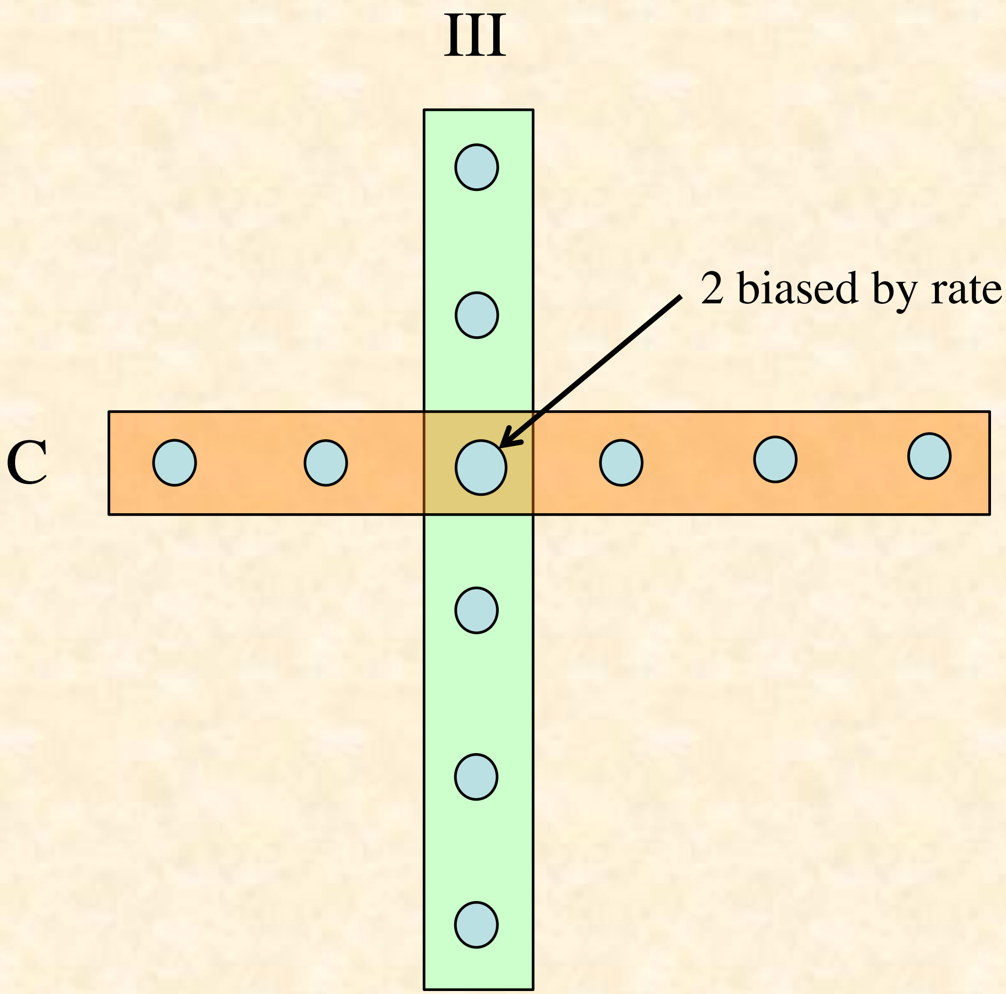
k-out-of-*n* Competitive Network

- With equal bias, it is essentially random which *k* will win
- With unequal bias, the *k* with strongest input win
- To bias neurons, make sure the inputs average to $2k–1$
- For k=1 it is a *winner-takes-all* network
- Macrocolumns in cortex seem to be *k*-out-of-*n* competitive feature detectors

Task Assignment Problem

- Six different tasks (I to VI)
- Six different agents (A to F)
- Agents can perform tasks at different rates
- What is the optimal assignment of tasks to agents (maximum rate)?
  (one task per agent, one agent per task)

Network for Task Assignment



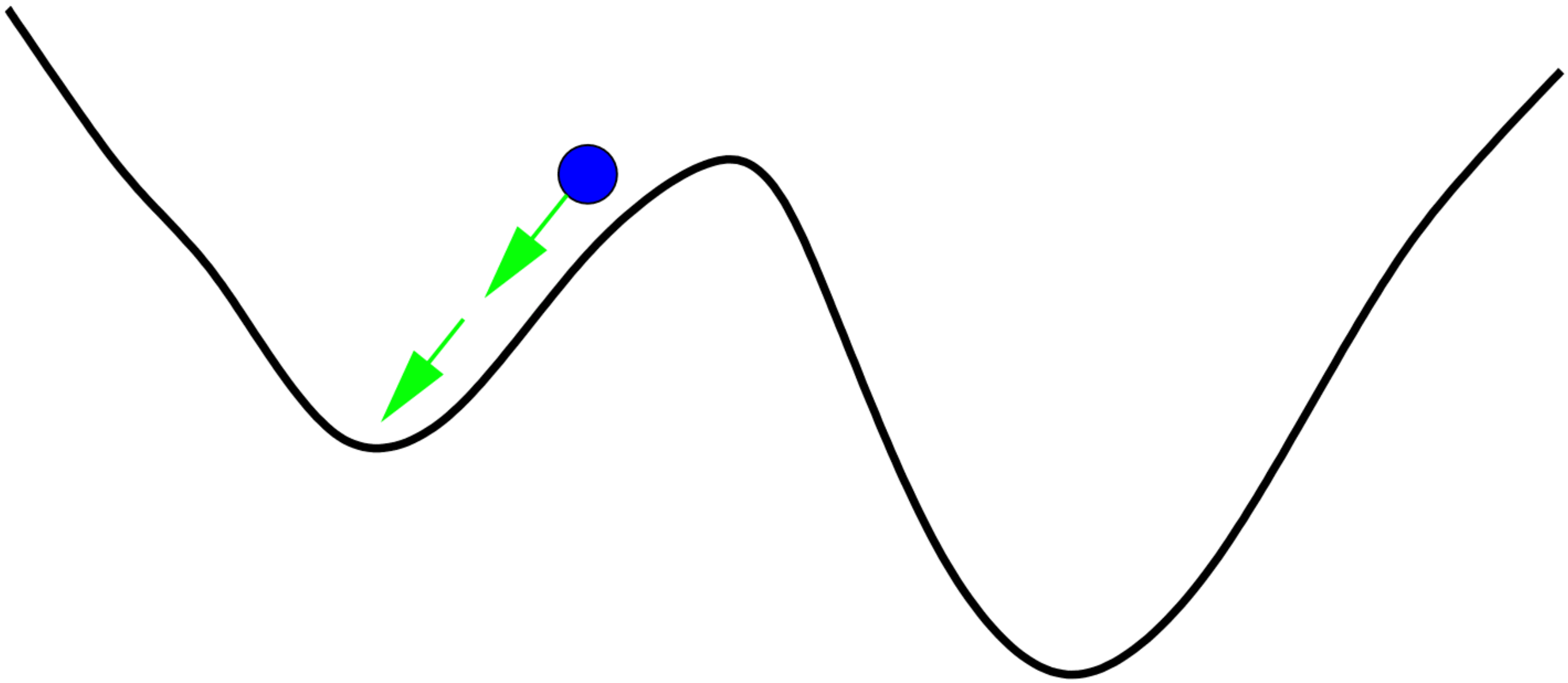
NetLogo Implementation of Task Assignment Problem

Run TaskAssignment.nlogo

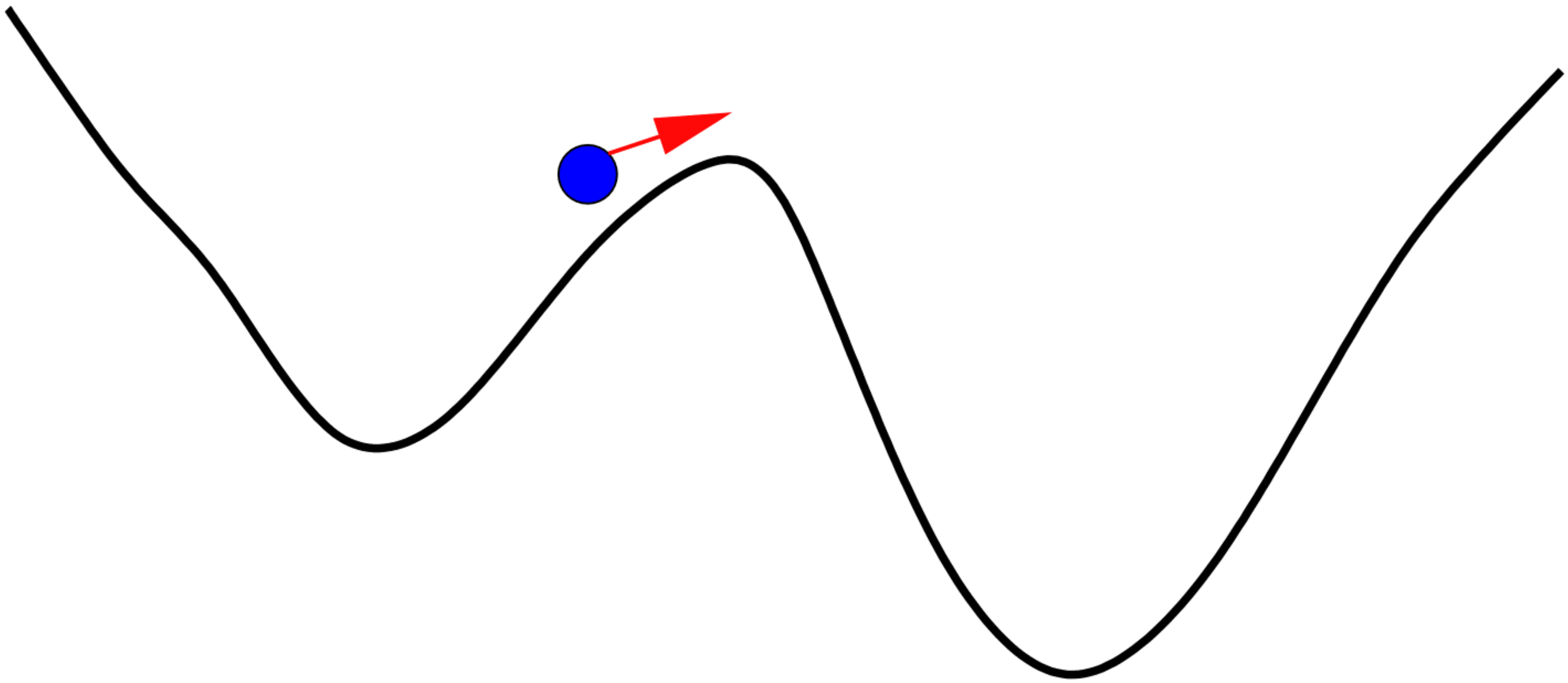
C.
Stochastic Neural Networks

(in particular, the stochastic Hopfield network)

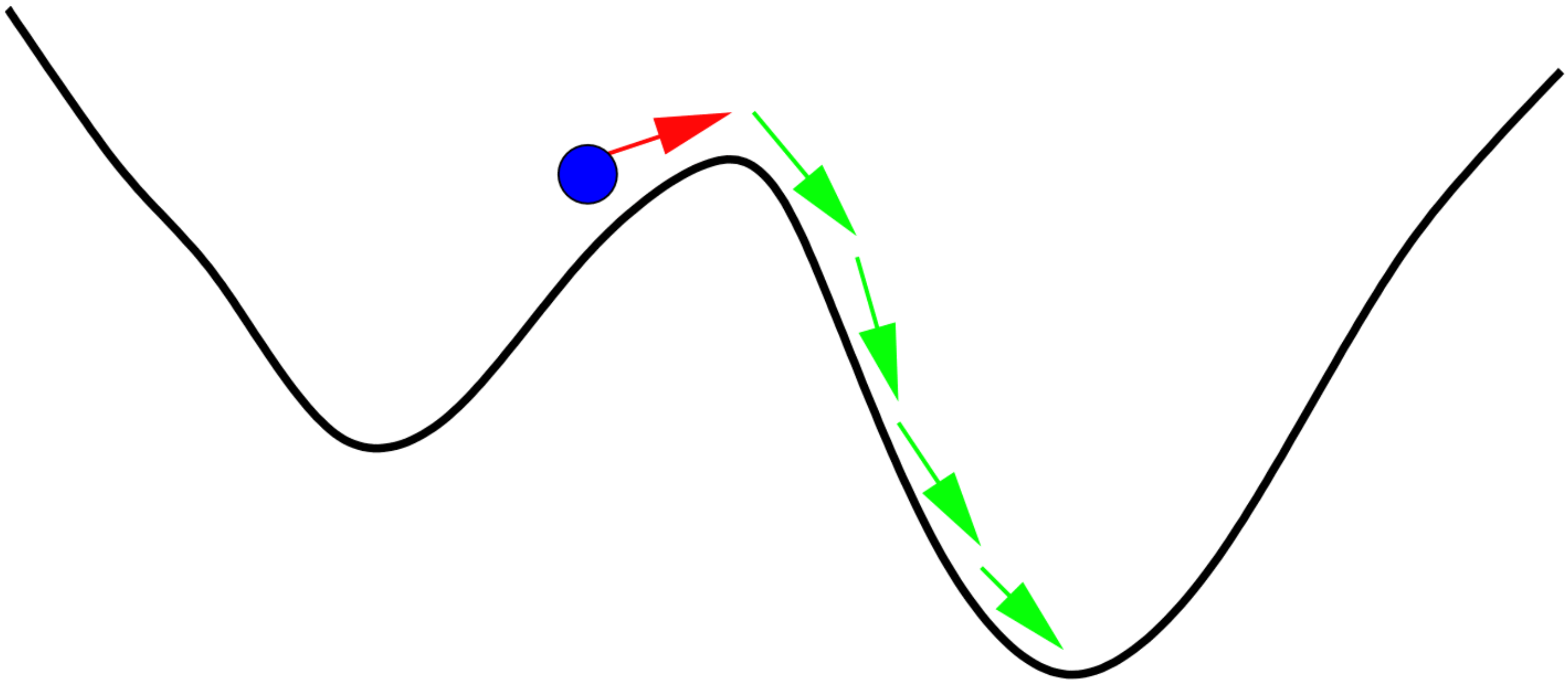
Trapping in Local Minimum



Escape from Local Minimum



Escape from Local Minimum



Motivation

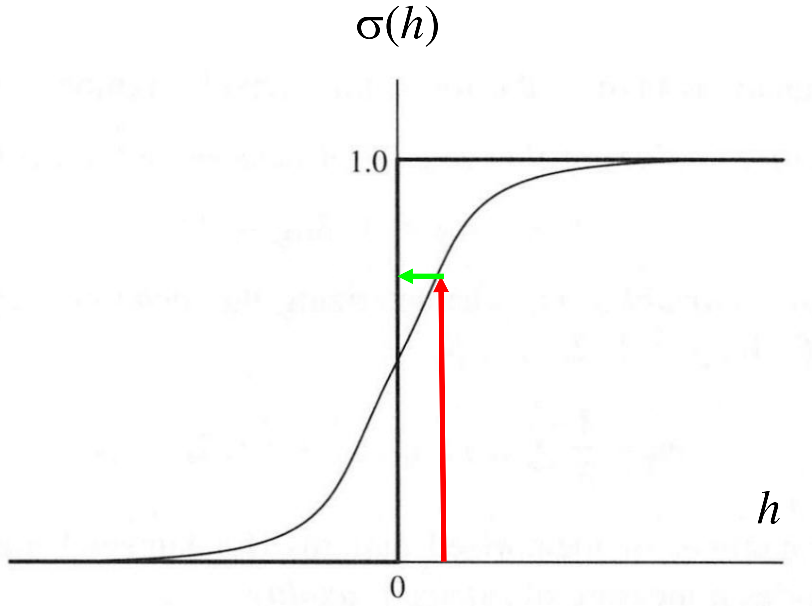
- Idea: with low probability, go against the local field
  - move up the energy surface
  - make the "wrong" microdecision
- Potential value for optimization: escape from local optima
- Potential value for associative memory: escape from spurious states
  - because they have higher energy than imprinted states

The Stochastic Neuron

Deterministic neuron: $s_i' = \operatorname{sgn}(h_i)$

$$\Pr\{s_i' = +1\} = \Theta(h_i)$$

$$\Pr\{s_i' = -1\} = 1 - \Theta(h_i)$$



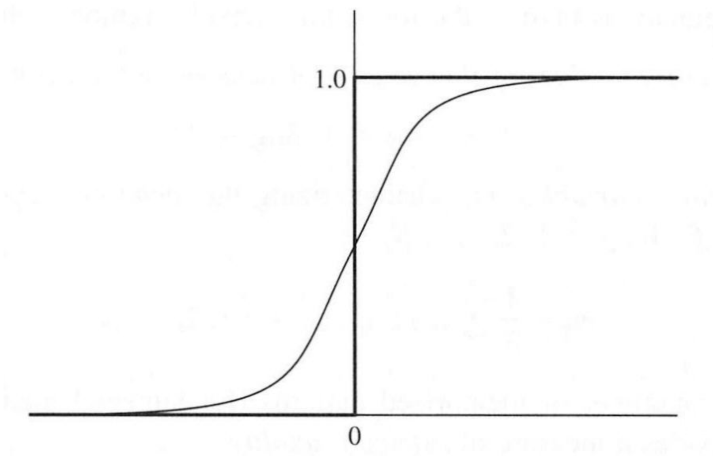
Stochastic neuron:

$$\Pr\{s_i' = +1\} = \sigma(h_i)$$

$$\Pr\{s_i' = -1\} = 1 - \sigma(h_i)$$

Logistic sigmoid: $\sigma(h) = \dfrac{1}{1 + \exp(-2h/T)}$

Properties of Logistic Sigmoid



$$\sigma(h) = \frac{1}{1 + e^{-2h/T}}$$

- As $h \to +\infty$, $\sigma(h) \to 1$
- As $h \to -\infty$, $\sigma(h) \to 0$
- $\sigma(0) = 1/2$

Pseudo-Temperature

- Temperature = measure of thermal energy (heat)
- Thermal energy = vibrational energy of molecules
- A source of random motion
- Pseudo-temperature = a measure of nondirected (random) change
- Logistic sigmoid gives same equilibrium probabilities as Boltzmann-Gibbs distribution
- Thermodynamic perk or coldness: $\beta = 1/T$

Transition Probability

Recall, change in energy $\Delta E = -\Delta s_k h_k$

$$= 2 s_k h_k$$

$$\Pr\{s_k' = \pm 1 | s_k = \mp 1\} = \sigma(\pm h_k) = \sigma(-s_k h_k)$$

$$\Pr\{s_k \to -s_k\} = \frac{1}{1 + \exp(2 s_k h_k / T)}$$

$$= \frac{1}{1 + \exp(\Delta E / T)}$$

Stability

- Are stochastic Hopfield nets stable?
- Thermal noise prevents absolute stability
- But with symmetric weights average values $\langle s_i \rangle$ become time-invariant

D.
Simulated Annealing

(Kirkpatrick, Gelatt & Vecchi, 1983)

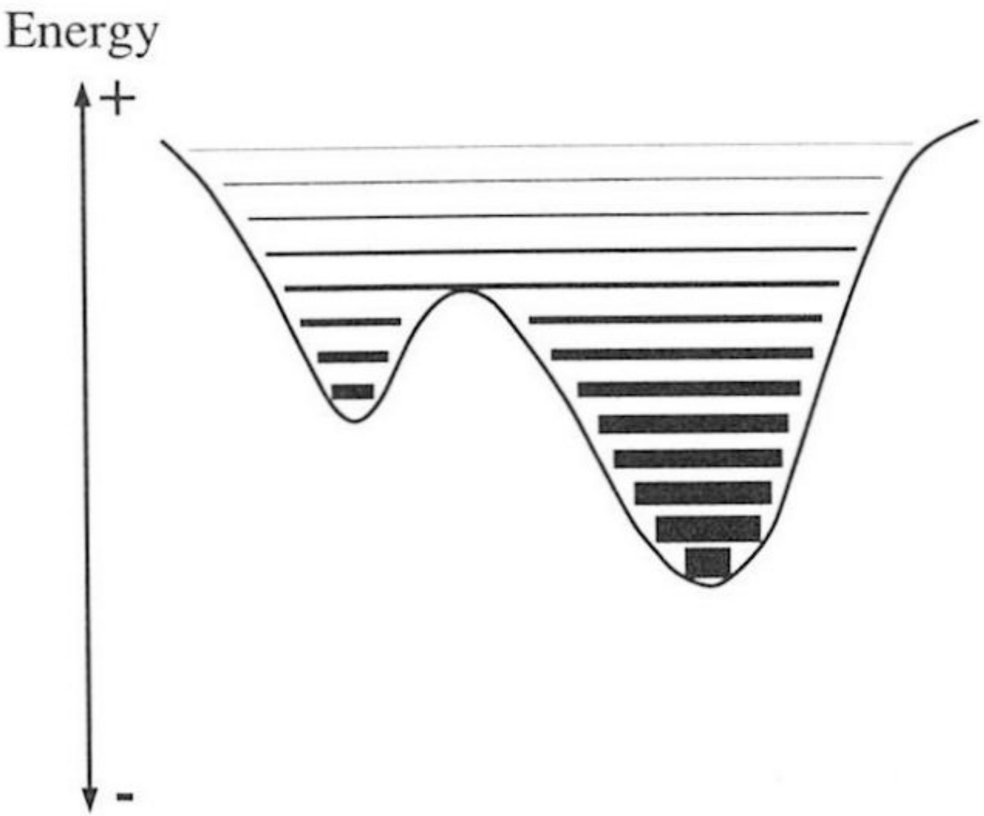
Dilemma

- In the early stages of search, we want a high temperature, so that we will explore the space and find the basins of the global minimum
- In the later stages we want a low temperature, so that we will relax into the global minimum and not wander away from it
- Solution: decrease the temperature gradually during search

Quenching vs. Annealing

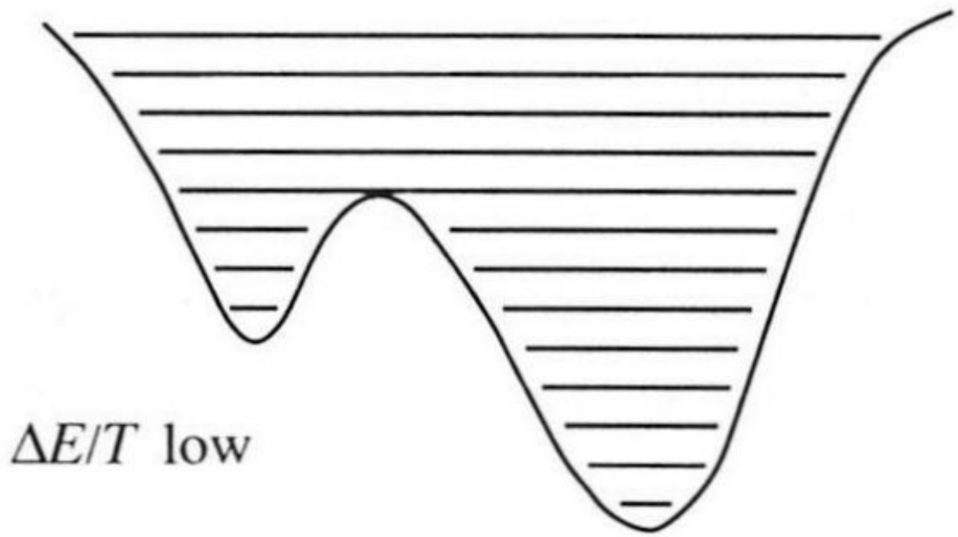
- Quenching:
  - rapid cooling of a hot material
  - may result in defects & brittleness
  - local order but global disorder
  - locally low-energy, globally frustrated
- Annealing:
  - slow cooling (or alternate heating & cooling)
  - reaches equilibrium at each temperature
  - allows global order to emerge
  - achieves global low-energy state

Effect of Moderate Temperature



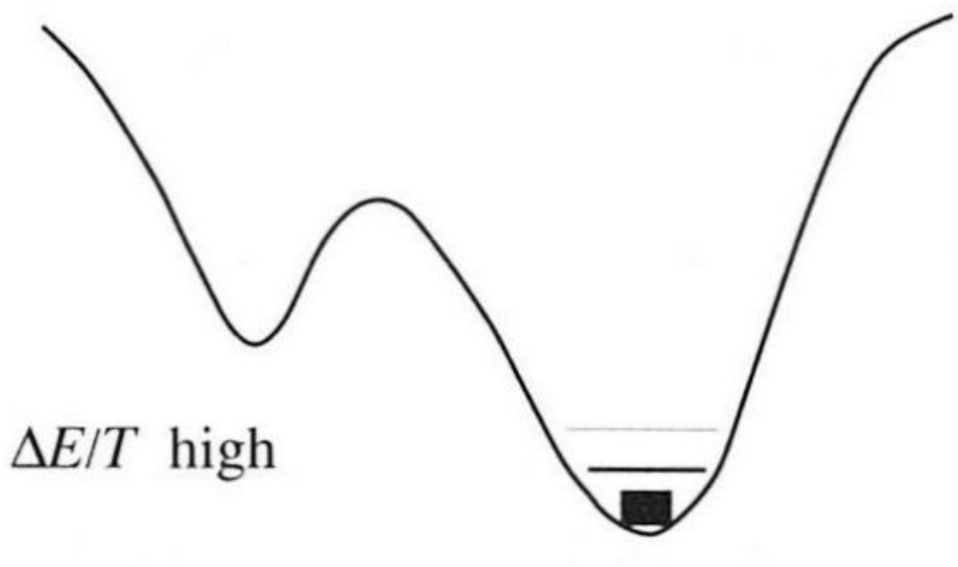
(fig. from Anderson *Intr. Neur. Comp.*)

Effect of High Temperature
(Low Perk)



(fig. from Anderson *Intr. Neur. Comp.*)

Effect of Low Temperature
(High Perk)



(fig. from Anderson *Intr. Neur. Comp.*)

Annealing Schedule

- Controlled decrease of temperature
- Should be sufficiently slow to allow equilibrium to be reached at each temperature
- With sufficiently slow annealing, the global minimum will be found with probability 1
- Design of schedules is a topic of research

Typical Practical Annealing Schedule

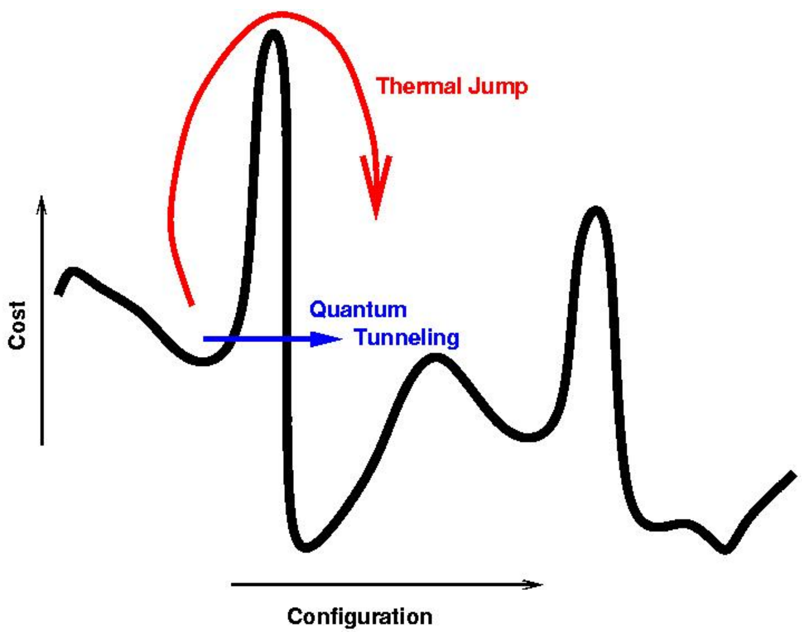
- Initial temperature T_0 sufficiently high so all transitions allowed
- Exponential cooling: $T_{k+1} = \alpha T_k$
  - typical $0.8 < \alpha < 0.99$
  - fixed number of trials at each temp.
  - expect at least 10 accepted transitions
- Final temperature: three successive temperatures without required number of accepted transitions

Summary

- Non-directed change (random motion) permits escape from local optima and spurious states
- Pseudo-temperature can be controlled to adjust relative degree of exploration and exploitation

E.
Quantum Annealing

Quantum Annealing



- Often quicker to go *through* than go *over*
- Start in disordered quantum state
- Slowly evolve to state that minimizes energy
- Can be simulated (inefficiently) on classical computer

Hamiltonian Quantum Mechanics

- Schrödinger's equation:
$$i\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle$$
- $H(t)$ is a Hamiltonian matrix, which is Hermitian and can be diagonalized:
$$H(t) = \sum_i E_i \left|E_i\right\rangle\!\left\langle E_i\right|$$
- where eigenvalues E_i are energies of eigenstates $\left|E_i\right\rangle$
- The smallest E_g defines the ground state $\left|E_g\right\rangle$

Problem Hamiltonian

- For problem P, determine $\mathbf{J}$ and $\mathbf{b}$ such
$$E_P = -\tfrac{1}{2}\mathbf{s}^{\mathrm{T}}\mathbf{J}\mathbf{s} - \mathbf{b}^{\mathrm{T}}\mathbf{s}$$
is minimized for solution $\mathbf{s} \in \{-1, +1\}^n$ (examples later)
- Define problem Hamiltonian:
$$H_P = -\sum_{\langle ij \rangle} J_{ij}\left(\mathbf{Z}_i \otimes \mathbf{Z}_j\right) - \sum_i b_i \mathbf{Z}_i$$
- Note: $\mathbf{Z}_i|0\rangle = \mathbf{Z}_i|\uparrow\rangle = +1|\uparrow\rangle$,
$\mathbf{Z}_i|1\rangle = \mathbf{Z}_i|\downarrow\rangle = -1|\downarrow\rangle$

Disordering Hamiltonian

- For example:

$$H_D = -\sum_i \mathbf{X}_i$$

- Since $\mathbf{X}_i|+\rangle = 0|+\rangle$ and $\mathbf{X}_i|-\rangle = -1|-\rangle$, the ground state is $|+\rangle^{\otimes n}$
- Note $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle)$
- H_D does not commute with H_P

Quantum Annealing Algorithm

- Define the time-dependent Hamiltonian:
$$H(t) = H_P + \Gamma(t) H_D$$
- $\Gamma(t)$ is the *transverse field coefficient*
- $\Gamma(t)$ starts large and $\Gamma(t) \to 0$ as $t \to 0$
- Typical annealing schedule:
$$\Gamma(k) = \frac{b}{(k+1)^{c/n}}$$

F.
Adiabatic Quantum Computing