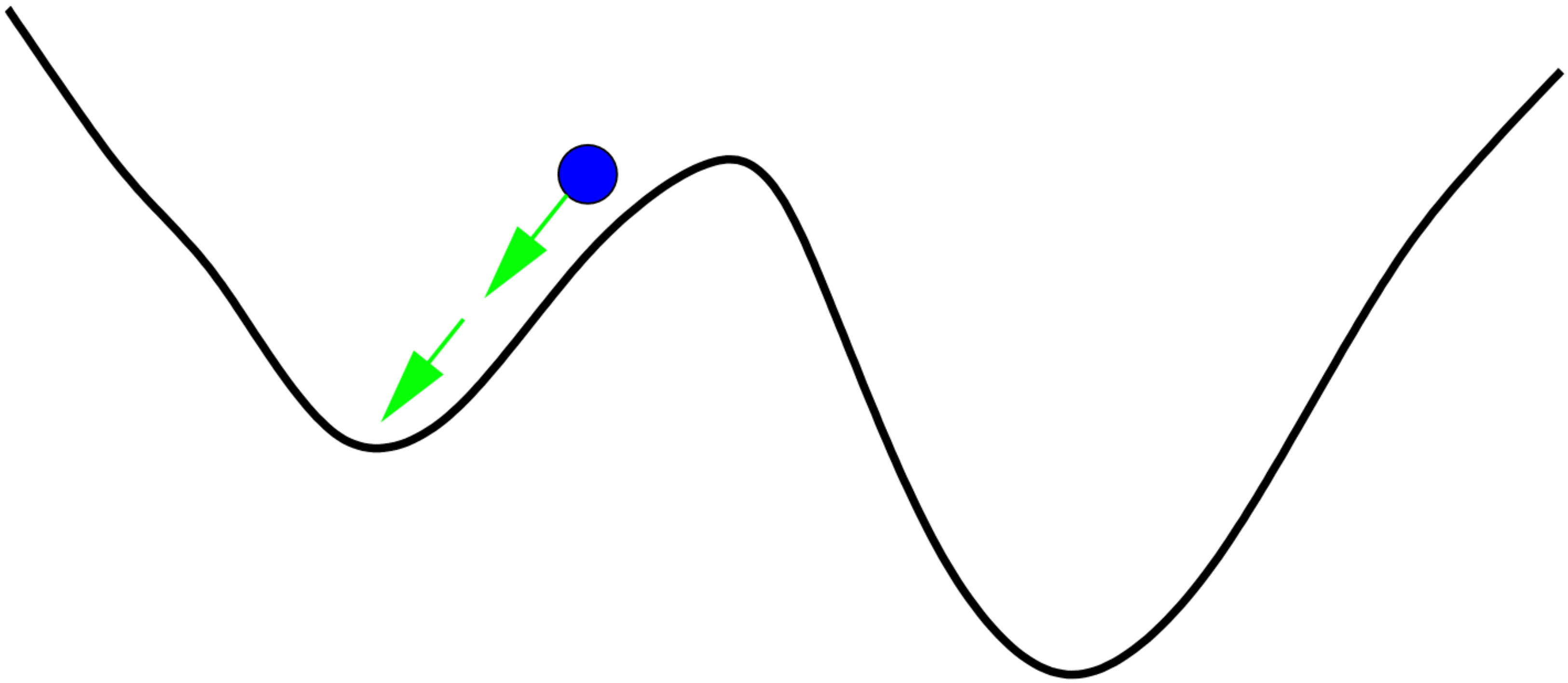
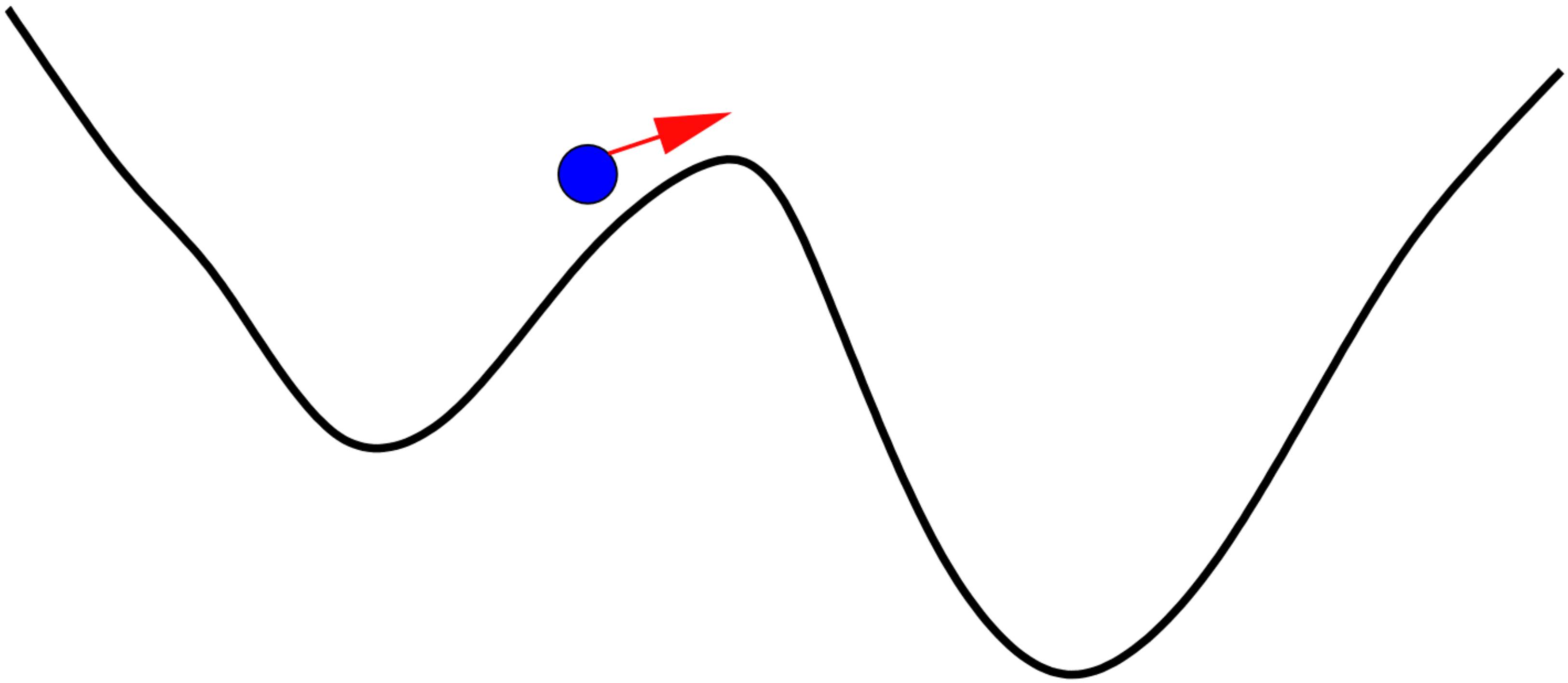
B.
Stochastic Neural Networks

(in particular, the stochastic Hopfield network)

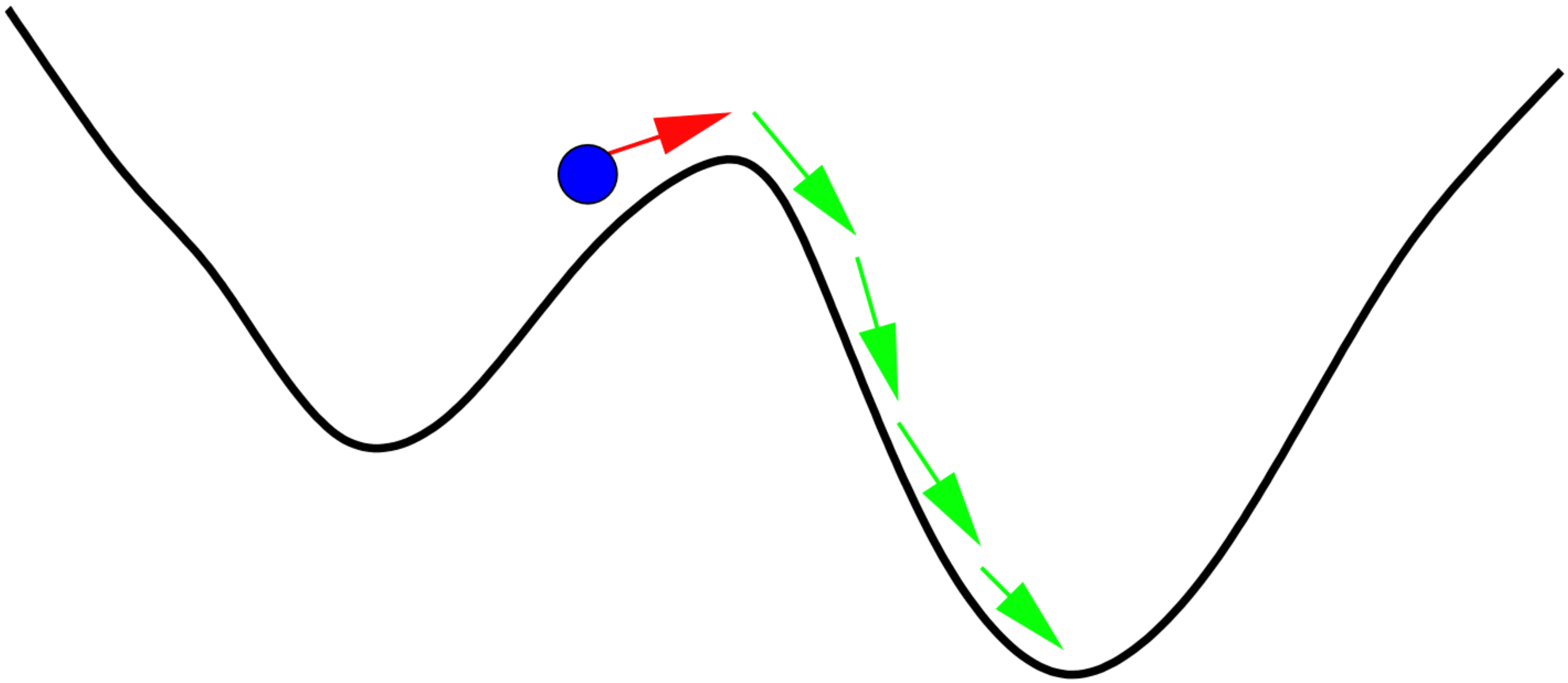
Trapping in Local Minimum



Escape from Local Minimum



Escape from Local Minimum



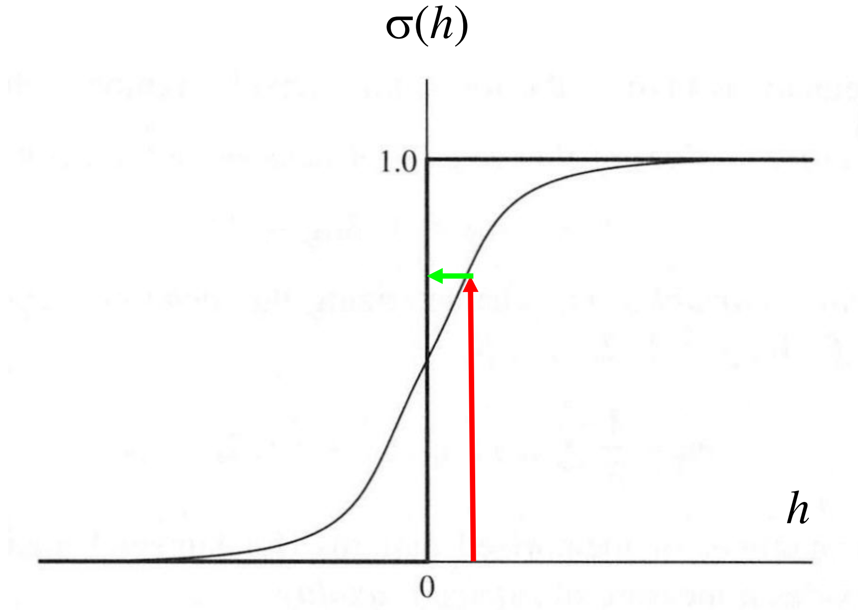
Motivation

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

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

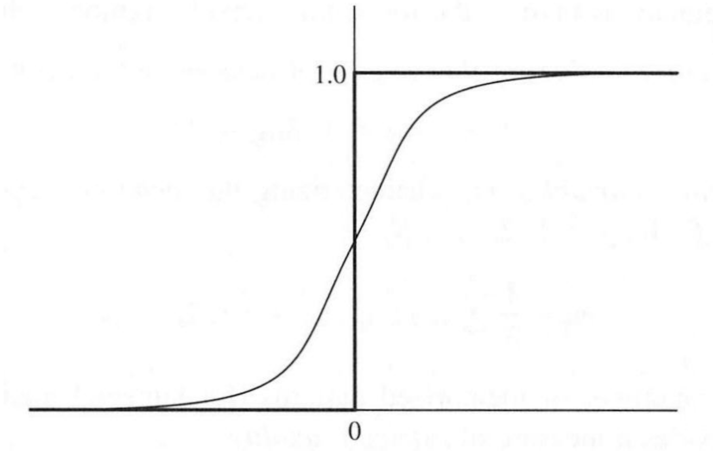


Stochastic neuron:

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

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

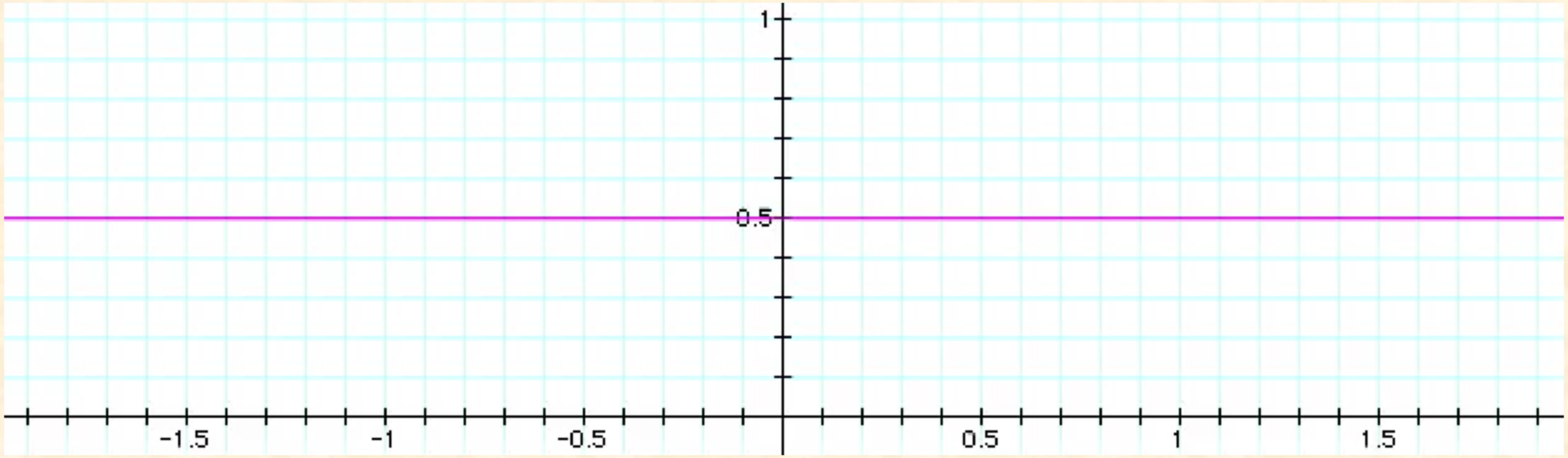
Properties of Logistic Sigmoid



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

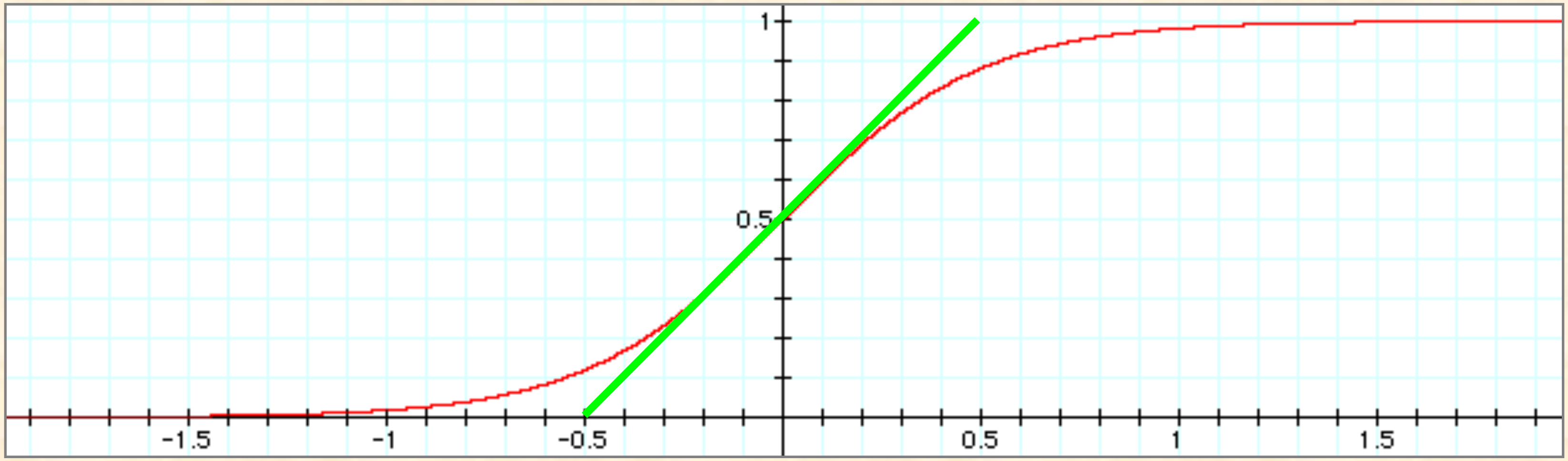
- As $h \to +\infty$, $\sigma(h) \to 1$
- As $h \to -\infty$, $\sigma(h) \to 0$
- $\sigma(0) = 1/2$

Logistic Sigmoid With Varying *T*



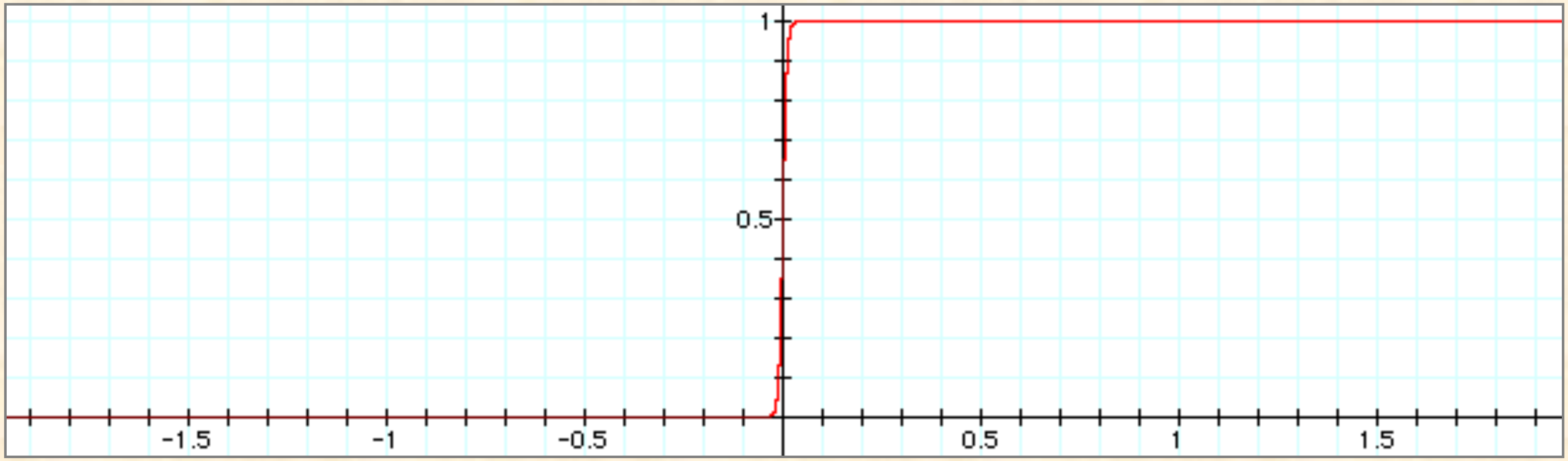
T varying from 0.05 to ∞ ($1/T = \beta = 0, 1, 2, \ldots, 20$)

Logistic Sigmoid
$T = 0.5$

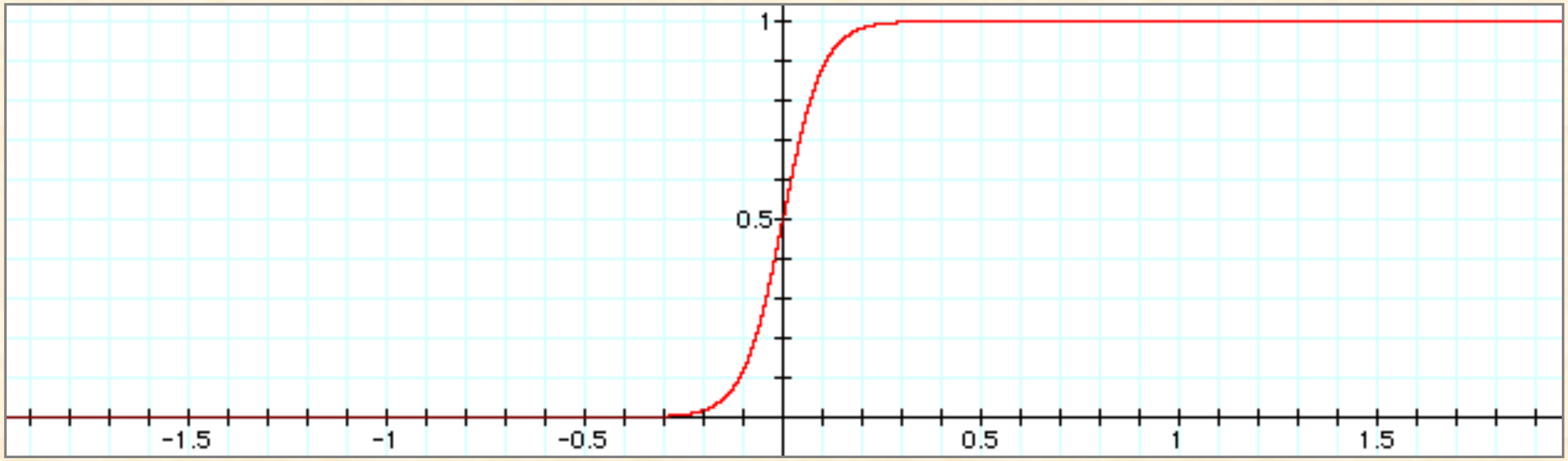


Slope at origin = 1 / 2*T*

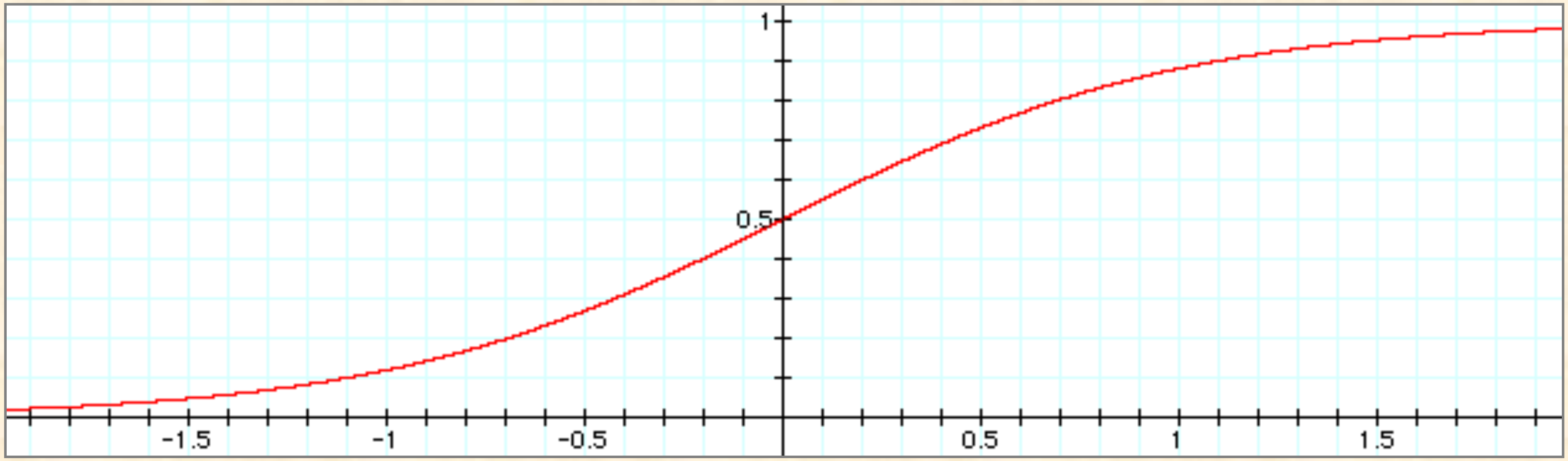
Logistic Sigmoid
$T = 0.01$



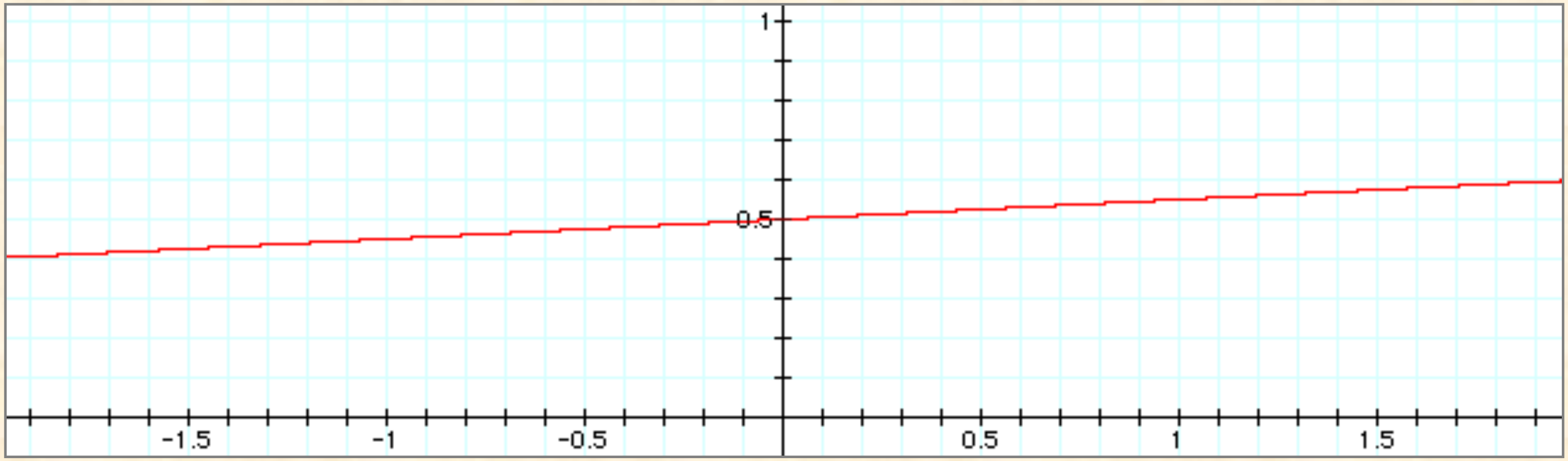
Logistic Sigmoid
$T = 0.1$



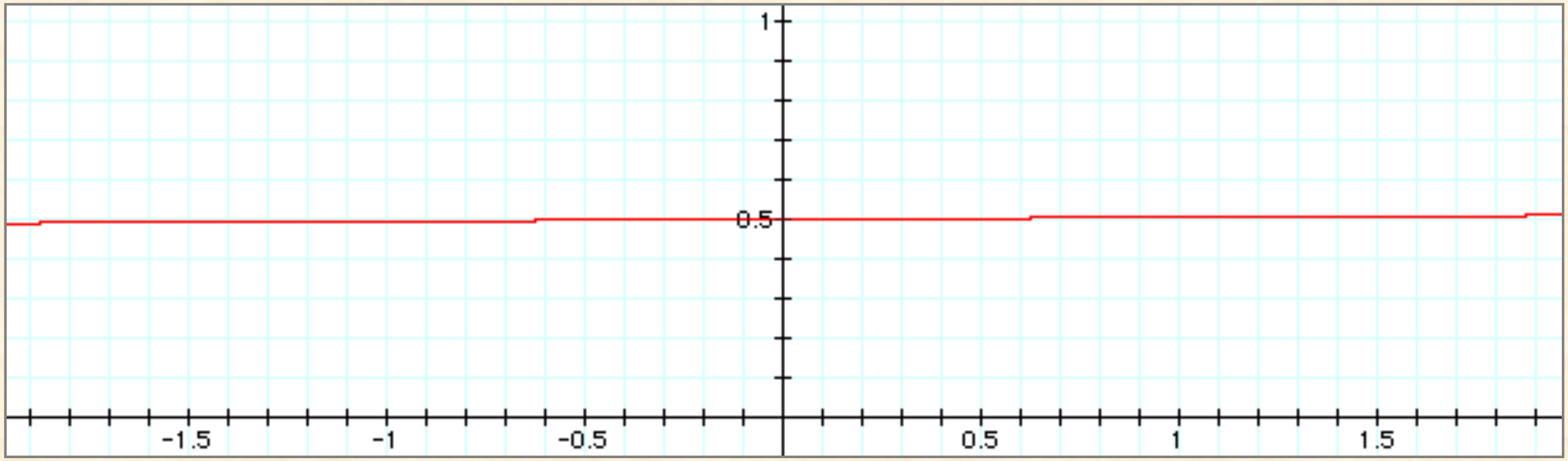
Logistic Sigmoid
$T = 1$



Logistic Sigmoid
$T = 10$



Logistic Sigmoid
$T = 100$



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 $\Delta E = -\Delta s_k h_k$

$$= 2 s_k h_k$$

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

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

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

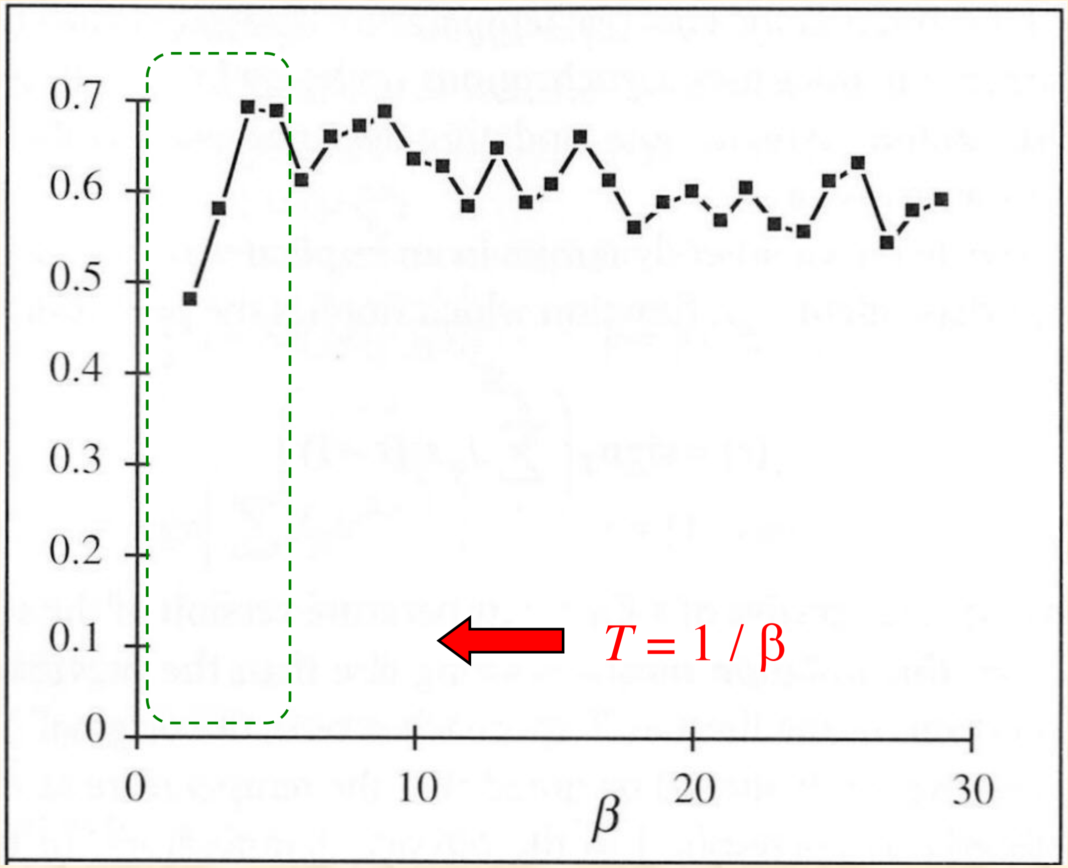
Stability

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

Does "Thermal Noise" Improve Memory Performance?

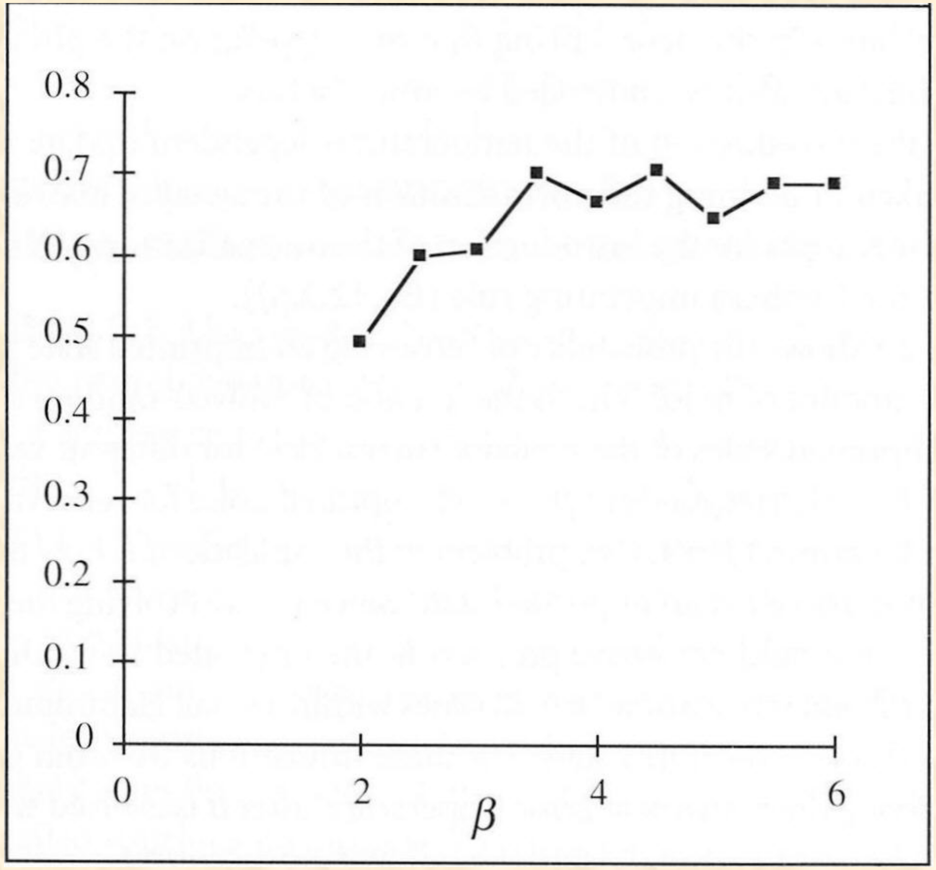
- Experiments by Bar-Yam (pp. 316–20):
  - $n = 100$
  - $p = 8$
- Random initial state
- To allow convergence, after 20 cycles set $T = 0$
- How often does it converge to an imprinted pattern?

Probability of Random State Converging on Imprinted State (n=100, p=8)



(fig. from Bar-Yam)

Probability of Random State Converging on Imprinted State (n=100, p=8)

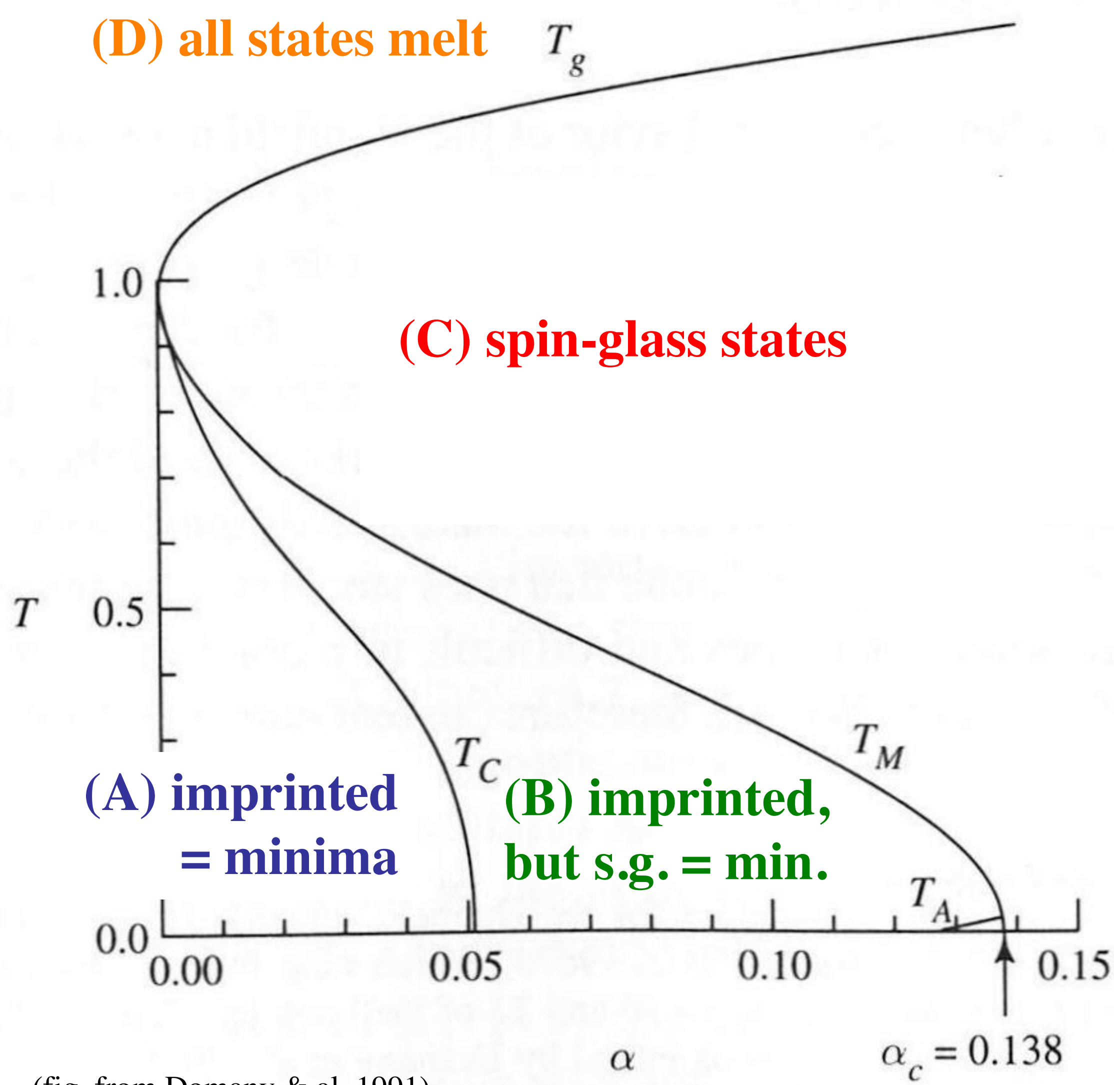


(fig. from Bar-Yam)

Analysis of Stochastic Hopfield Network

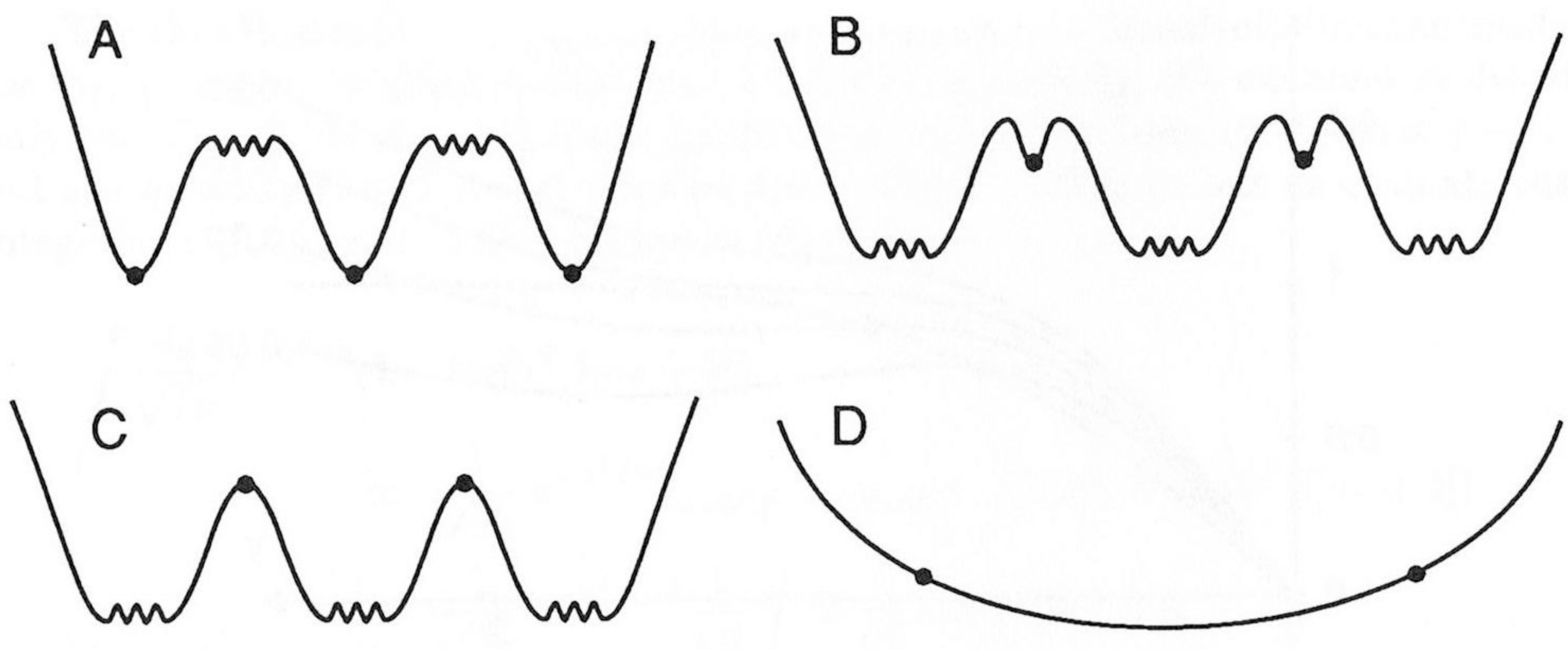
- Complete analysis by Daniel J. Amit & colleagues in mid-80s
- See D. J. Amit, *Modeling Brain Function: The World of Attractor Neural Networks*, Cambridge Univ. Press, 1989.
- The analysis is beyond the scope of this course

Phase Diagram



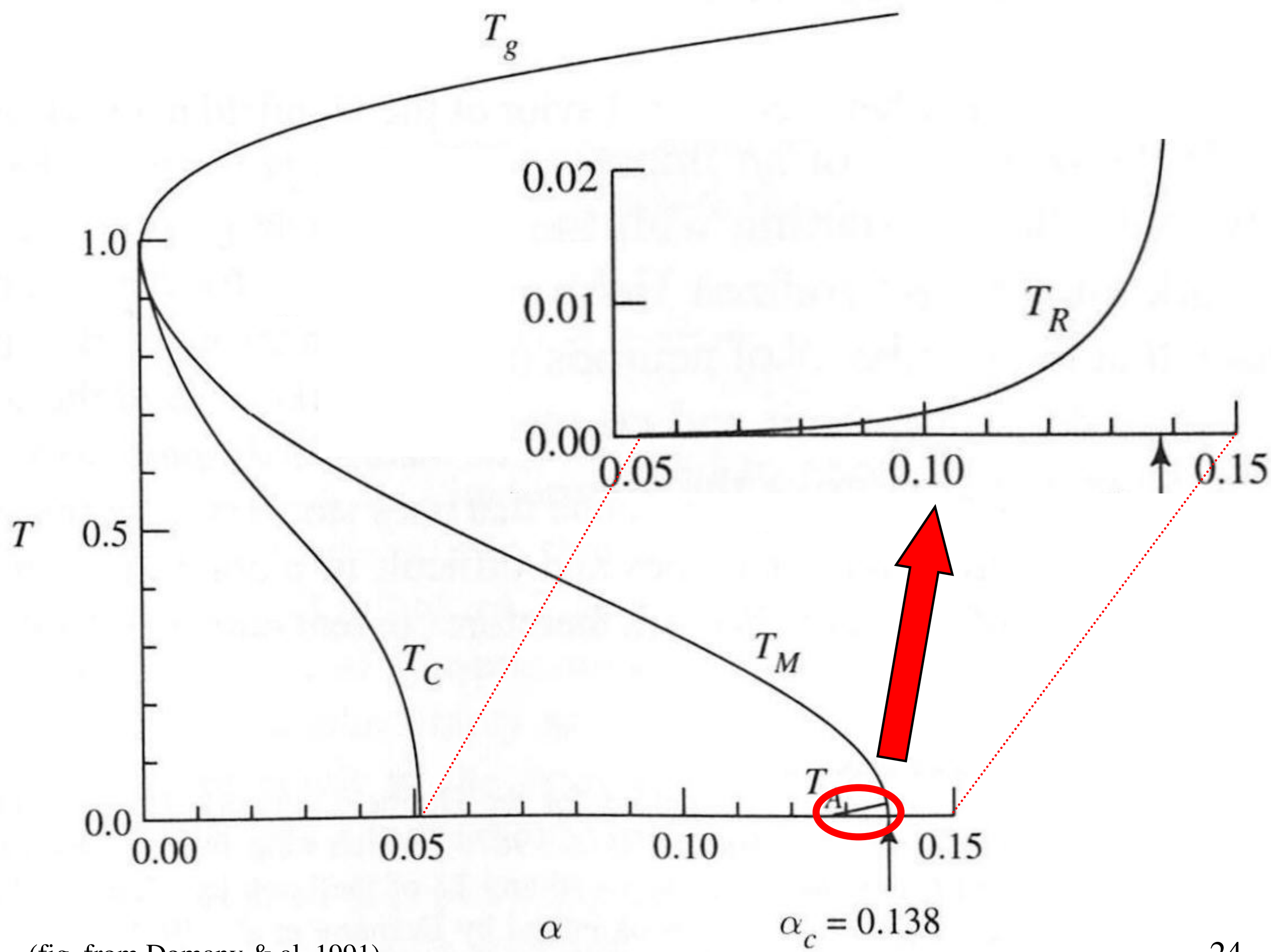
(fig. from Domany & al. 1991)

Conceptual Diagrams of Energy Landscape



(fig. from Hertz & al. *Intr. Theory Neur. Comp*.)

Phase Diagram Detail



 (fig. from Domany & al. 1991)

Simulated Annealing

(Kirkpatrick, Gelatt & Vecchi, 1983)

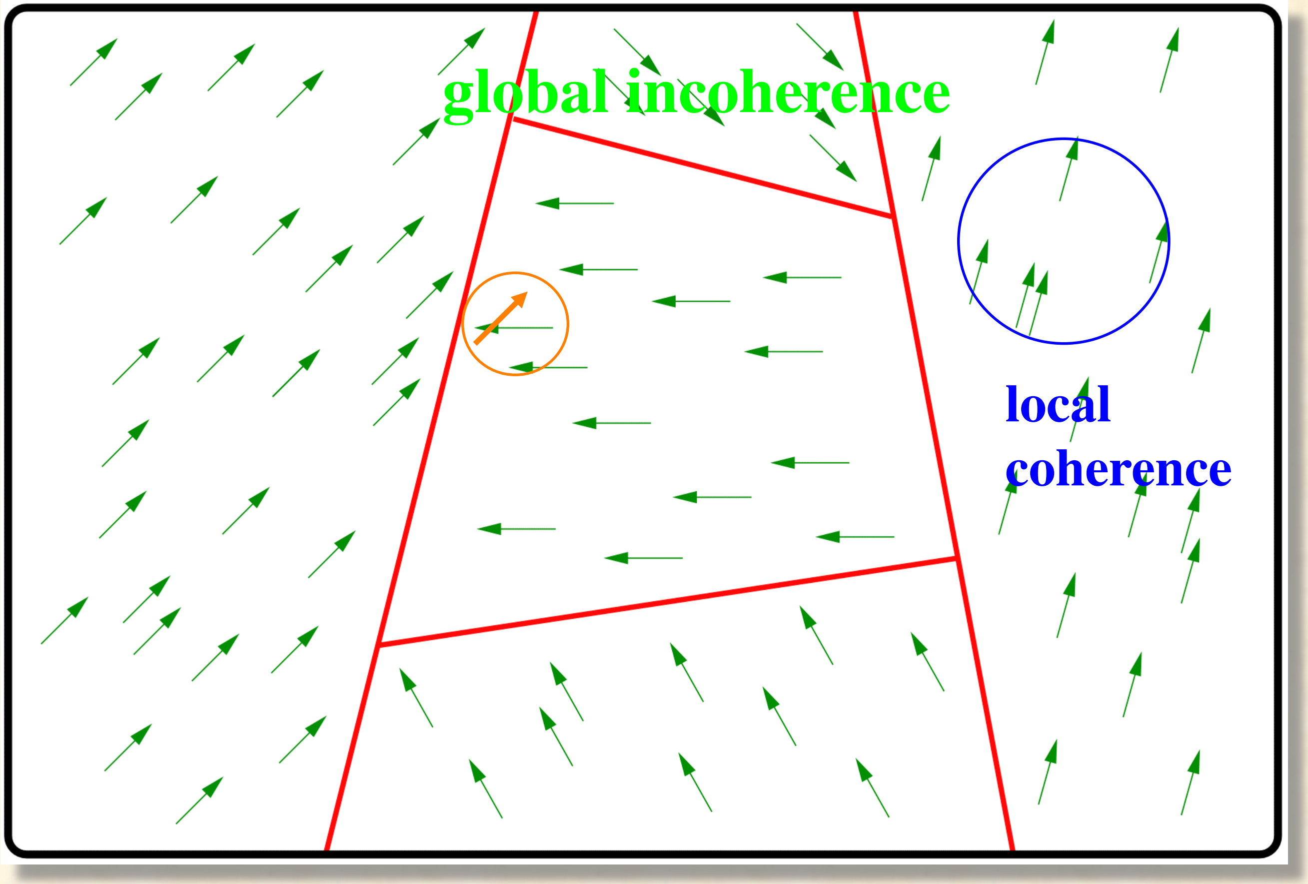
Dilemma

- 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

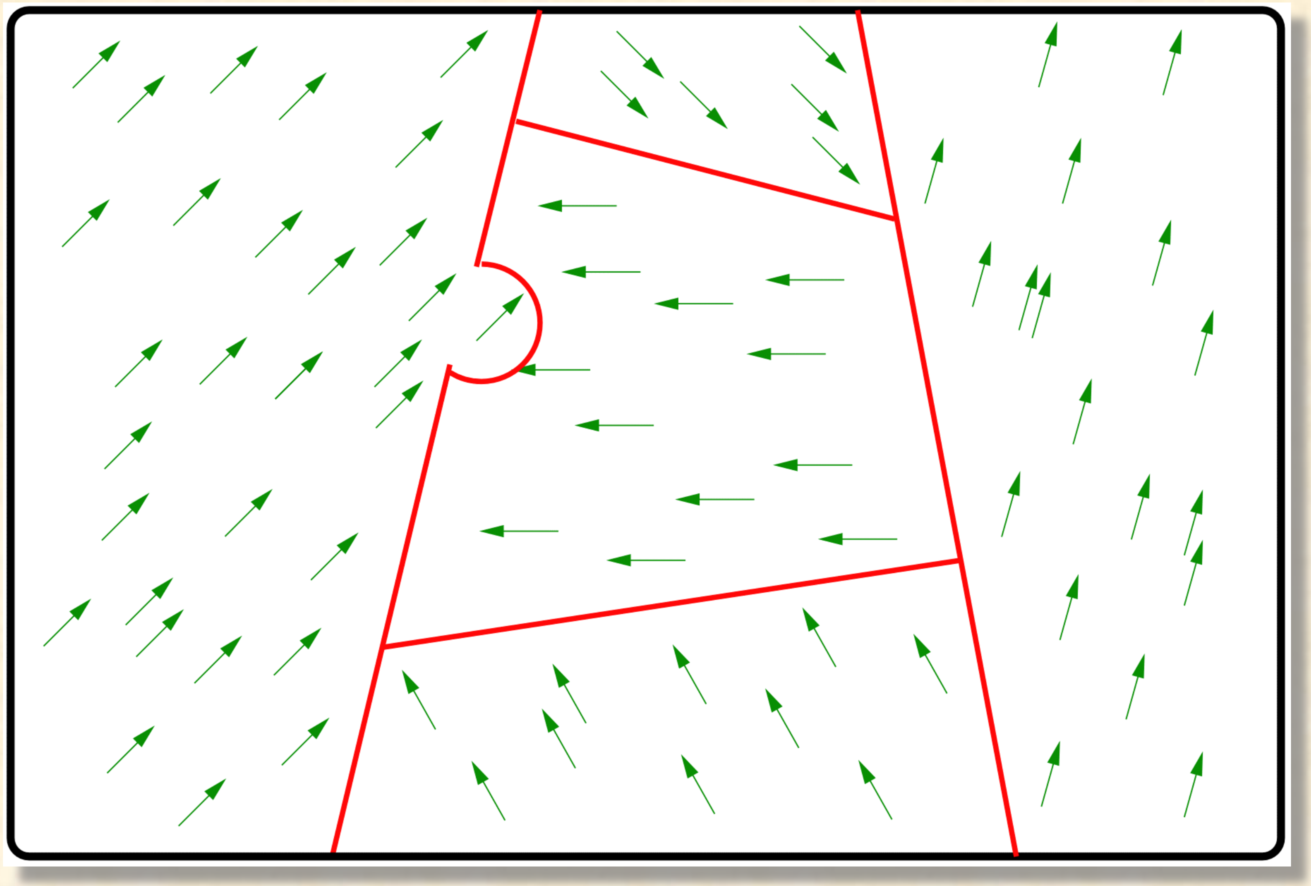
Quenching vs. Annealing

- 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

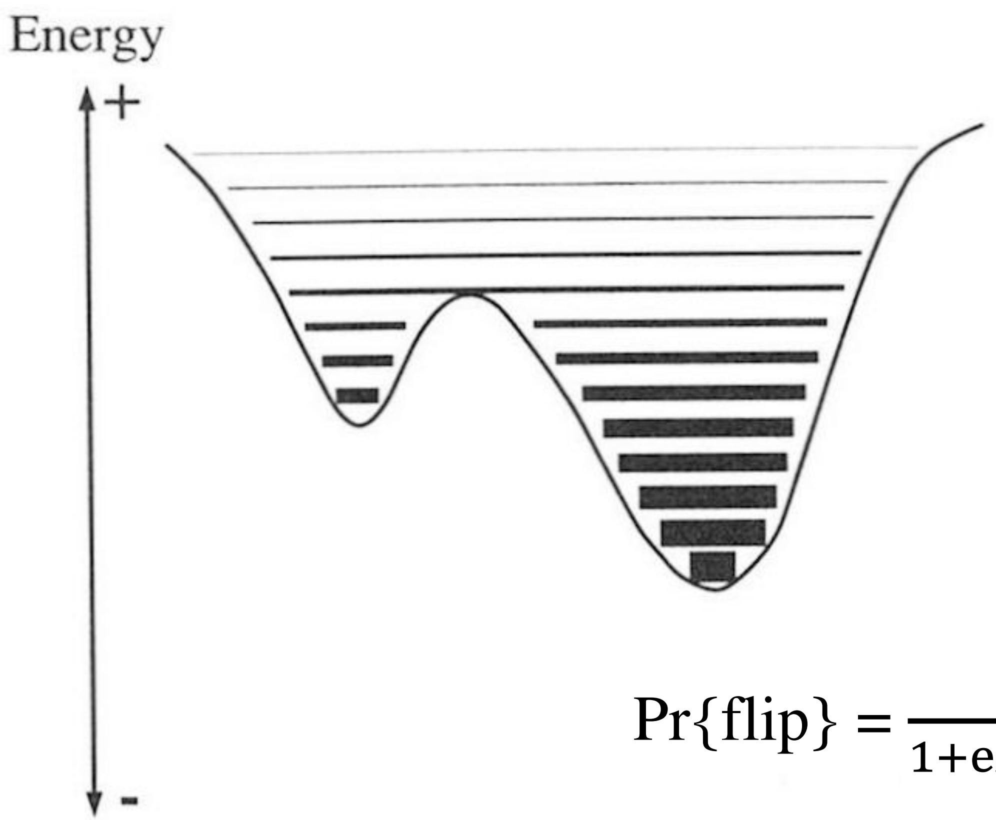
Multiple Domains



Moving Domain Boundaries



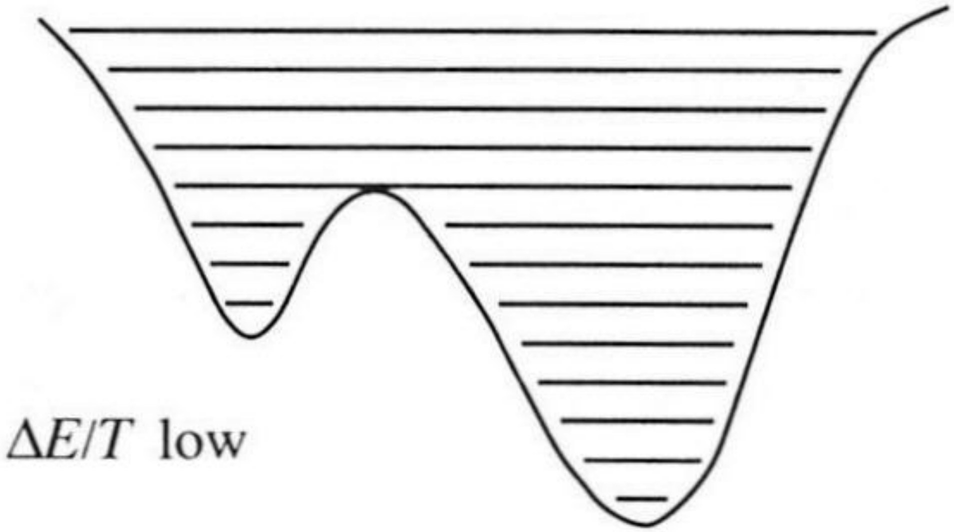
Effect of Moderate Temperature



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

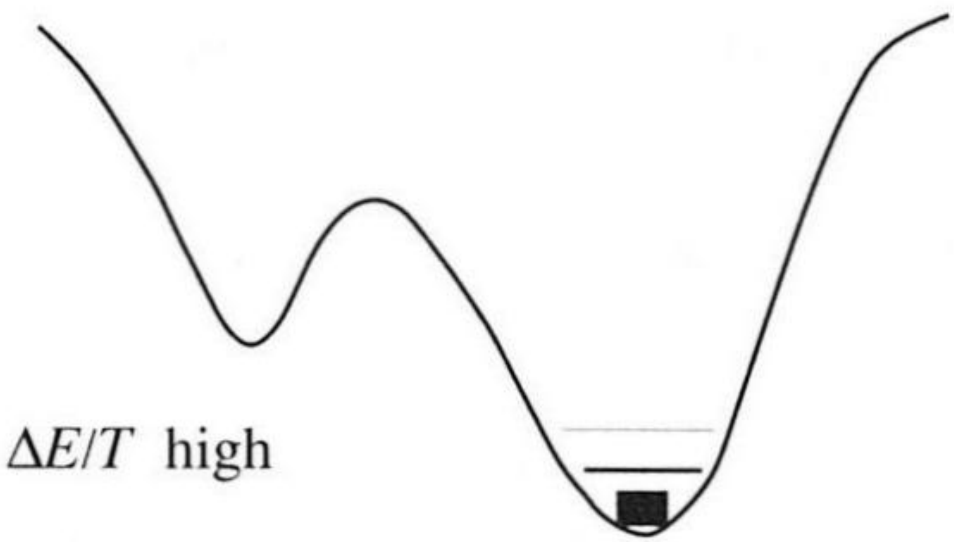
(fig. from Anderson *Intr. Neur. Comp.*)

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

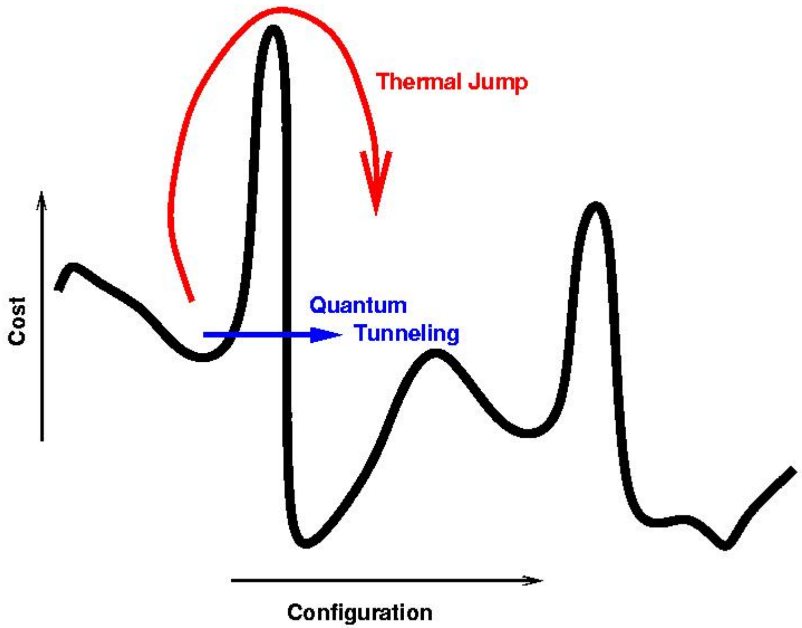
Typical Practical Annealing Schedule

- 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

Quantum Annealing



- See for example D-wave Systems <www.dwavesys.com>

Hopfield Network for 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

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

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

Energy function:

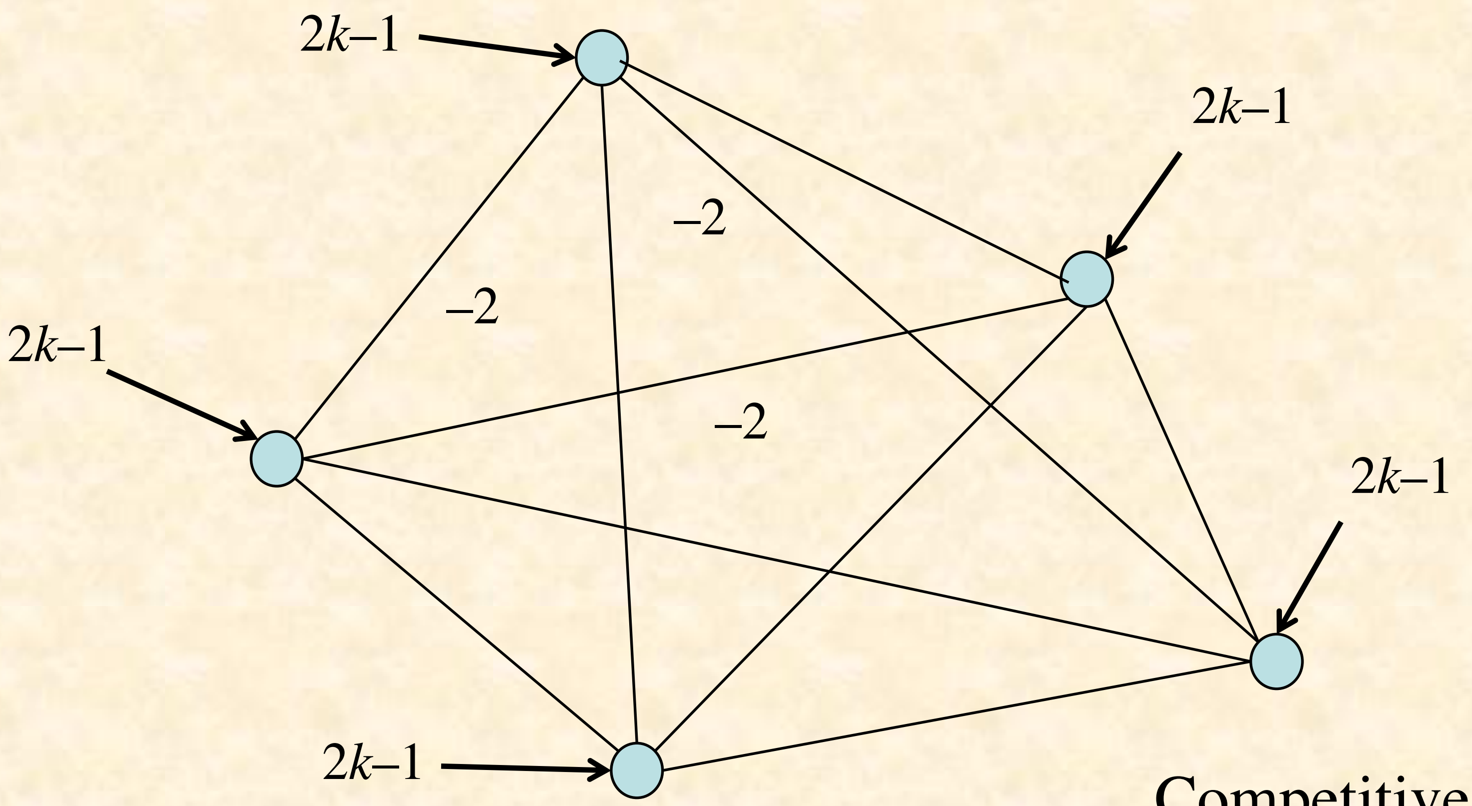
$$E = -\frac{1}{2} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} T_{ij} V_i V_j - \sum_{i=1}^{n} V_i I_i$$

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:

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

k-out-of-n Rule



Competitive Network

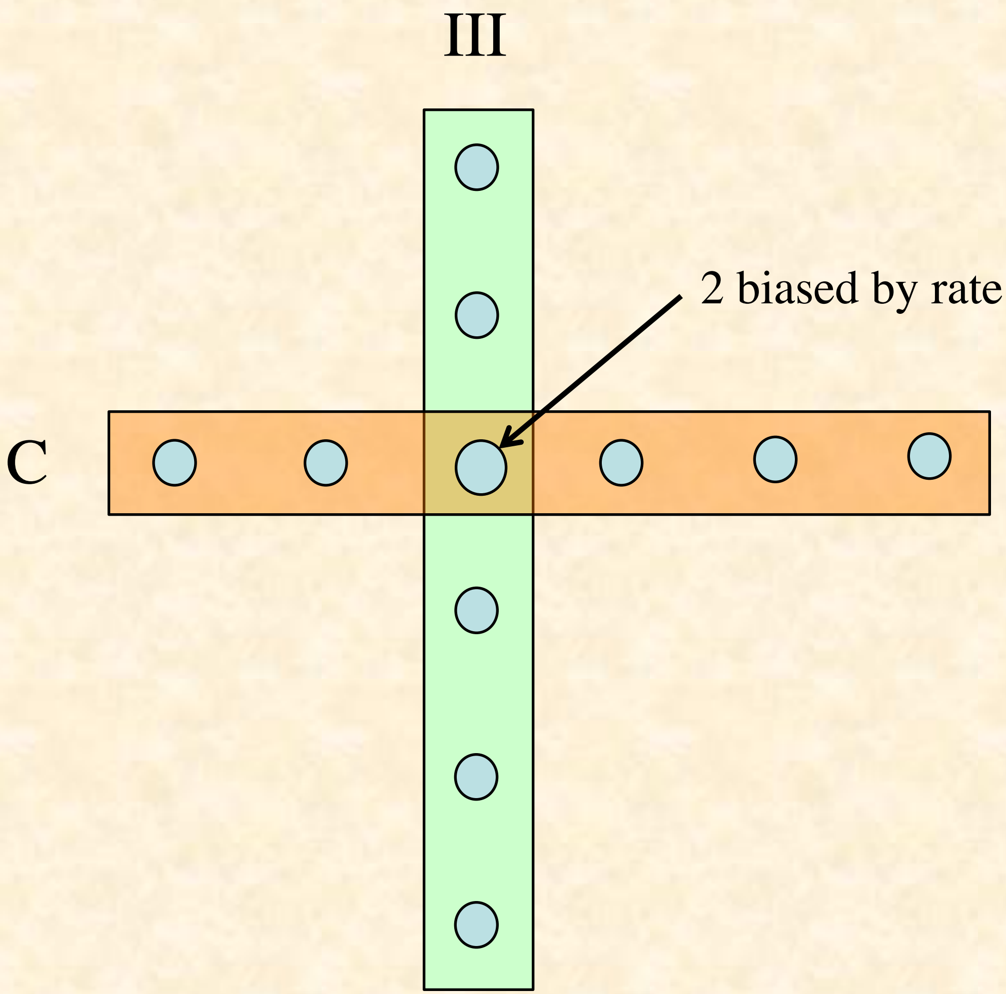
k-out-of-n Competitive Network

- 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]

Part IV