Lecture 13

Artificial Neural Networks

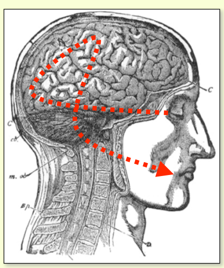
11/2/06 1

The Cognitive Inversion

- Computers can do some things very well that are difficult for people
  - e.g., arithmetic calculations
  - playing chess & other board games
  - doing proofs in formal logic & mathematics
  - handling large amounts of data precisely
- But computers are very bad at some things that are easy for people (and even some animals)
  - e.g., face recognition & general object recognition
  - autonomous locomotion
  - sensory-motor coordination
- Conclusion: *brains work very differently from Von Neumann computers*

11/2/06 2

The 100-Step Rule



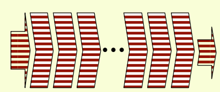
- Typical recognition tasks take less than one second
- Neurons take several milliseconds to fire
- Therefore then can be at most about 100 sequential processing steps

11/2/06 3

Two Different Approaches to Computing

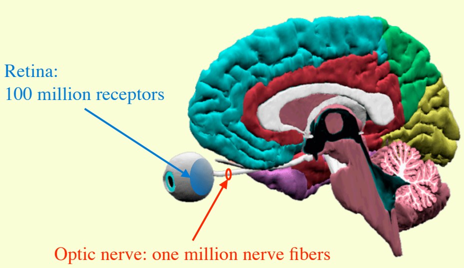
Von Neumann: *Narrow but Deep*

Neural Computation: *Shallow but Wide*



11/2/06 4

How Wide?



11/2/06 5

Neurons are Not Logic Gates

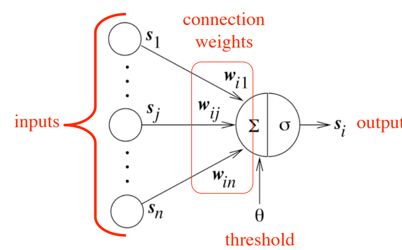
- Speed
  - electronic logic gates are very fast (nanoseconds)
  - neurons are comparatively slow (milliseconds)
- Precision
  - logic gates are highly reliable digital devices
  - neurons are imprecise analog devices
- Connections
  - logic gates have few inputs (usually 1 to 3)
  - many neurons have >100 000 inputs

11/2/06 6

Artificial Neural Networks

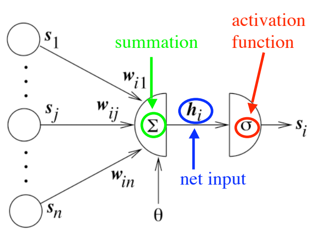
11/2/06 7

Typical Artificial Neuron



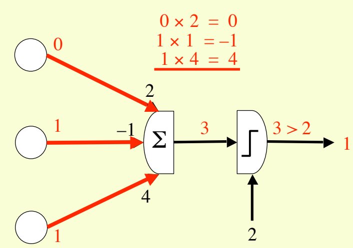
11/2/06 8

Typical Artificial Neuron



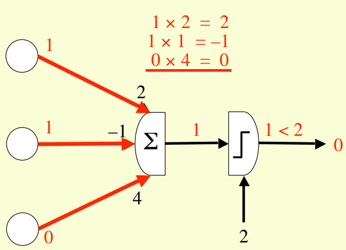
11/2/06 9

Operation of Artificial Neuron



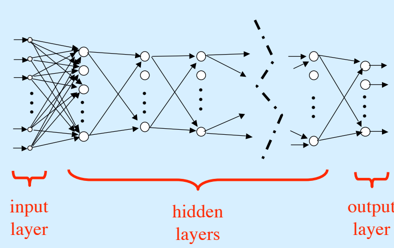
11/2/06 10

Operation of Artificial Neuron



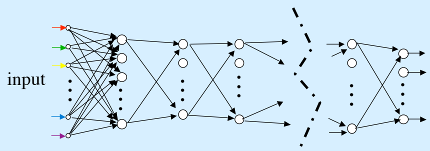
11/2/06 11

Feedforward Network

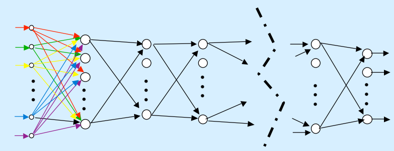


11/2/06 12

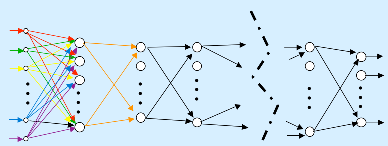
Feedforward Network In Operation (1)



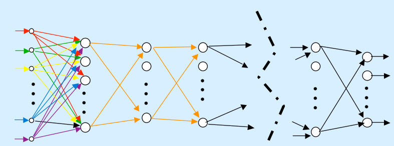
Feedforward Network In Operation (2)



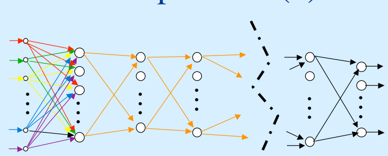
Feedforward Network In Operation (3)



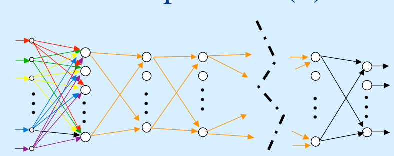
Feedforward Network In Operation (4)



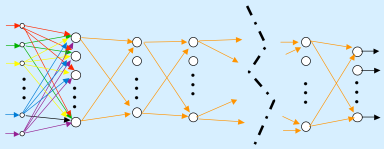
Feedforward Network In Operation (5)



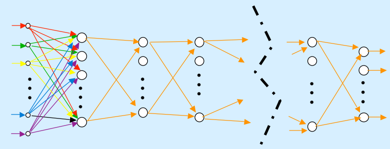
Feedforward Network In Operation (6)



Feedforward Network In Operation (7)



Feedforward Network In Operation (8)



Comparison with Non-Neural Net Approaches

- Non-NN approaches typically decide output from a small number of dominant factors
- NNs typically look at a large number of factors, each of which weakly influences output
- NNs permit:
  - subtle discriminations
  - holistic judgments
  - context sensitivity

Connectionist Architectures

- The knowledge is implicit in the *connection weights* between the neurons
- Items of knowledge are not stored in dedicated memory locations, as in a Von Neumann computer
- "Holographic" knowledge representation:
  - each knowledge item is distributed over many connections
  - each connection encodes many knowledge items
- Memory & processing is robust in face of damage, errors, inaccuracy, noise, …

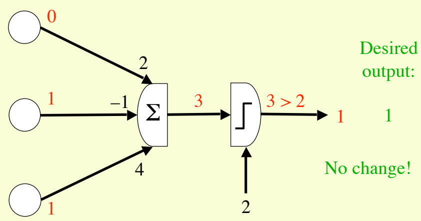
Differences from Digital Calculation

- Information represented in continuous images (rather than language-like structures)
- Information processing by continuous image processing (rather than explicit rules applied in individual steps)
- Indefiniteness is inevitable (rather than definiteness assumed)

Supervised Learning

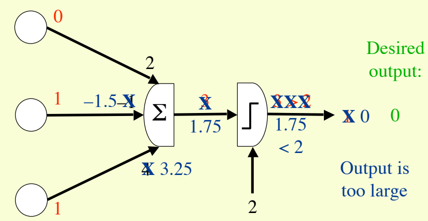
- Produce desired outputs for training inputs
- Generalize reasonably & appropriately to other inputs
- Good example: pattern recognition
- Neural nets are *trained* rather than *programmed*
  - another difference from Von Neumann computation

Learning for Output Neuron (1)



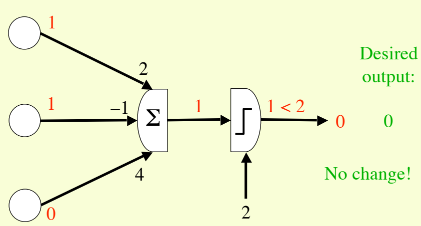
11/2/06 25

Learning for Output Neuron (2)



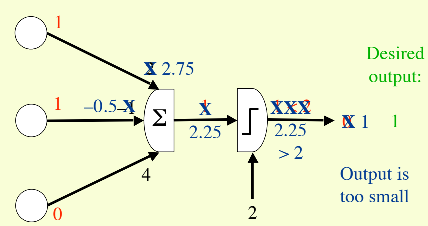
11/2/06 26

Learning for Output Neuron (3)



11/2/06 27

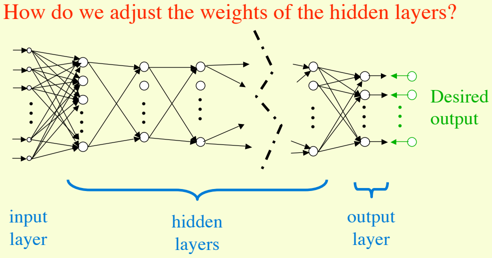
Learning for Output Neuron (4)



11/2/06 28

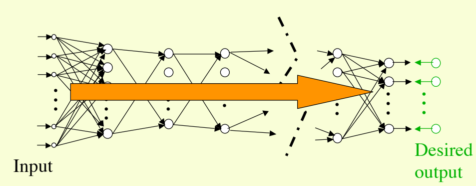
Credit Assignment Problem

How do we adjust the weights of the hidden layers?



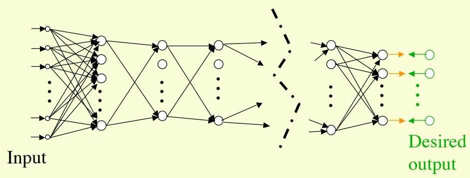
11/2/06 29

Back-Propagation: Forward Pass

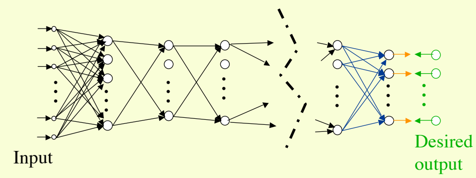


11/2/06 30

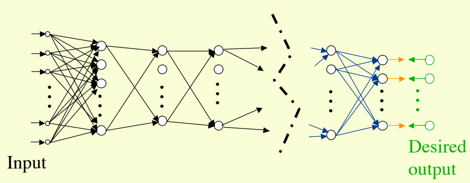
Back-Propagation: Actual vs. Desired Output



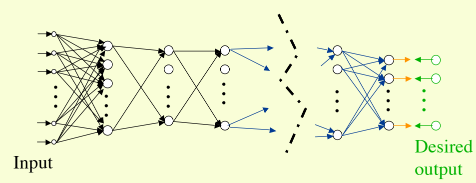
Back-Propagation: Correct Output Neuron Weights



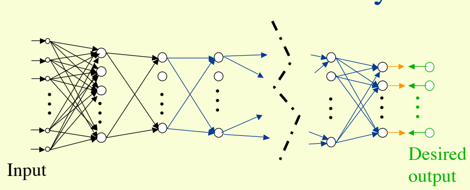
Back-Propagation: Correct Last Hidden Layer



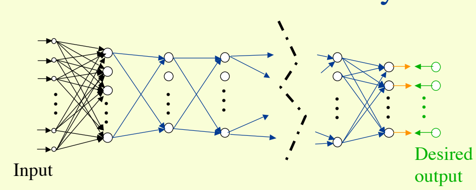
Back-Propagation: Correct All Hidden Layers



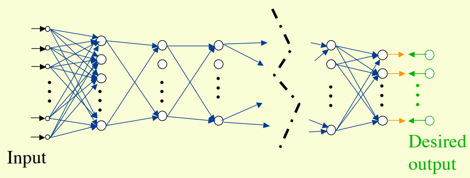
Back-Propagation: Correct All Hidden Layers



Back-Propagation: Correct All Hidden Layers



Back-Propagation: Correct First Hidden Layer



Use of Back-Propagation (BP)

- Typically the weights are changed slowly
- Typically net will *not* give correct outputs for all training inputs after one adjustment
- Each input/output pair is used repeatedly for training
- BP may be slow
- But there are many better ANN learning algorithms

ANN Training Procedures

- *Supervised training*: we show the net the output it should produce for each training input (e.g., BP)
- *Reinforcement training*: we tell the net if its output is right or wrong, but not what the correct output is
- *Unsupervised training*: the net attempts to find patterns in its environment without external guidance

Applications of ANNs

- "Neural nets are the second-best way of doing everything"
- If you really understand a problem, you can design a special purpose algorithm for it, which will beat a NN
- However, if you don't understand your problem very well, you can generally train a NN to do it well enough

The Hopfield Network

(and constraint satisfaction)

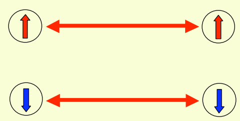
Hopfield Network

- 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:

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

Positive Coupling

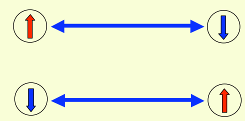
- Positive *sense* (sign)
- Large *strength*



11/2/06 43

Negative Coupling

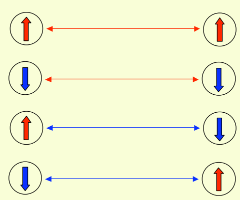
- Negative *sense* (sign)
- Large *strength*



11/2/06 44

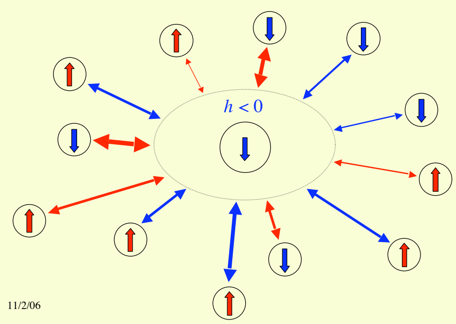
Weak Coupling

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



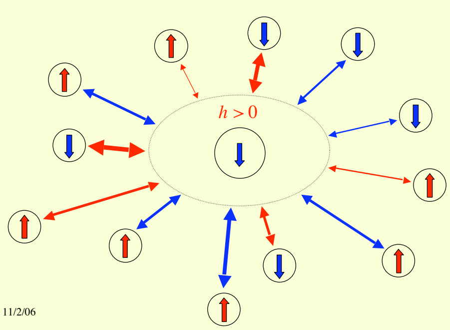
11/2/06 45

State = –1 & Local Field < 0



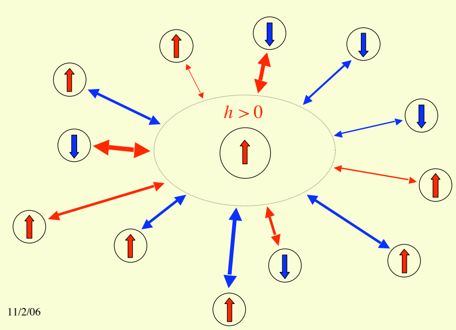
11/2/06 46

State = –1 & Local Field > 0



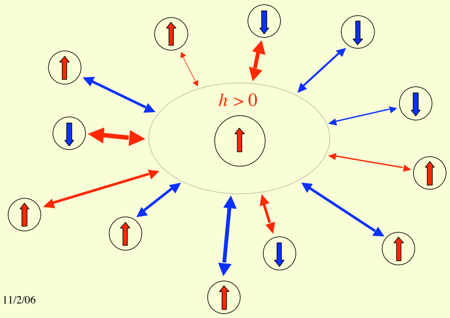
11/2/06 47

State Reverses

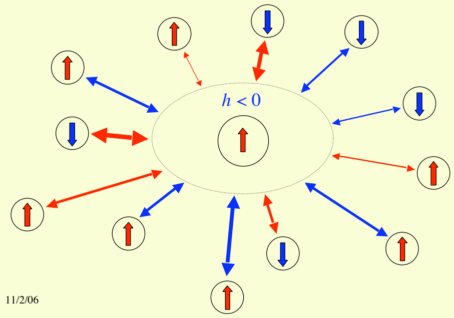


11/2/06 48

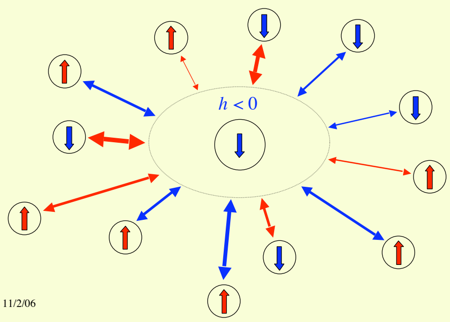
State = +1 & Local Field > 0



State = +1 & Local Field < 0



State Reverses



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?

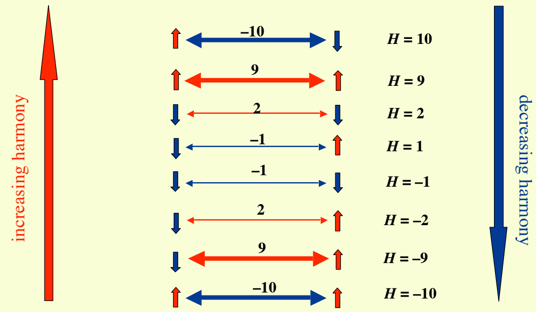
Demonstration of Hopfield Net

Run Hopfield Demo

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 (or increases its "harmony")
- But could a relaxation of one neuron lead to greater tension in other places?

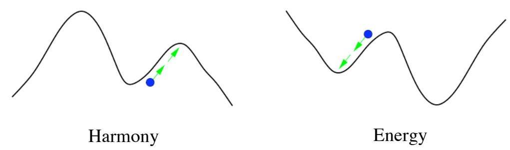
Quantifying Harmony



11/2/06 55

Energy

- "Energy" (or "tension") is the opposite of harmony
- $E = -H$



11/2/06 56

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? (Yes)

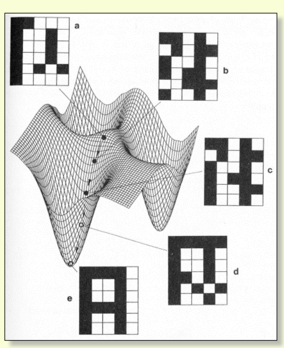
11/2/06 57

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

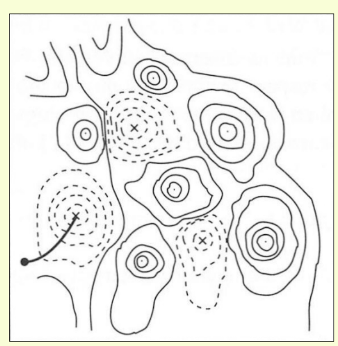
11/2/06 58

Conceptual Picture of Descent on Energy Surface

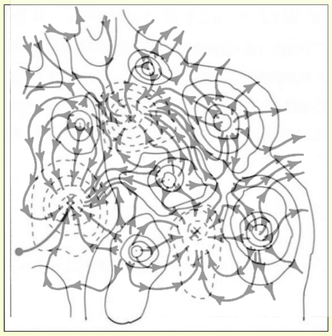


11/2/06 (fig. from Solé & Goodwin) 59

Energy Surface

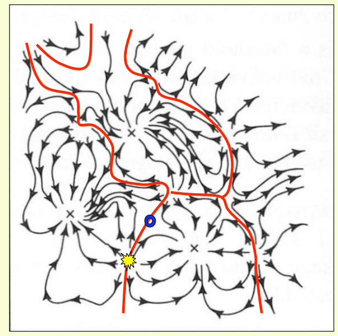


11/2/06 (fig. from Haykin *Neur. Netw.*) 60



Energy Surface + Flow Lines

11/2/06 (fig. from Haykin *Neur. Netw.*) 61

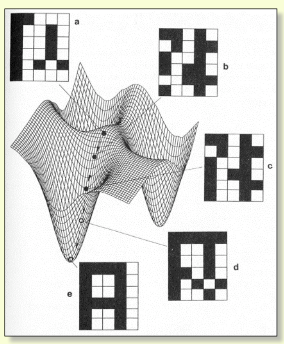


Flow Lines

Basins of Attraction

11/2/06 (fig. from Haykin *Neur. Netw.*) 62

Storing Memories as Attractors



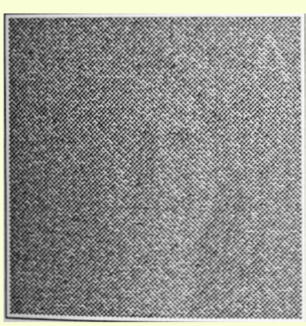
11/2/06 (fig. from Solé & Goodwin) 63

Demonstration of Hopfield Net

Run Hopfield Demo

11/2/06 64

Example of Pattern Restoration



11/2/06 (fig. from Arbib 1995) 65

Example of Pattern Restoration

11/2/06 (fig. from Arbib 1995) 66

Example of Pattern Restoration

(fig. from Arbib 1995)

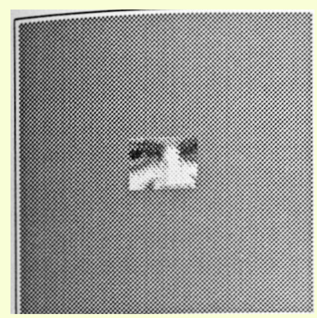
Example of Pattern Restoration

(fig. from Arbib 1995)

Example of Pattern Restoration

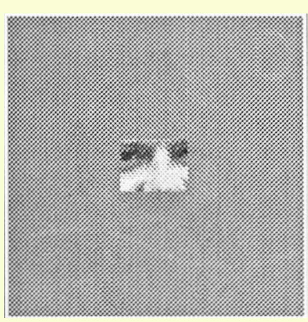
(fig. from Arbib 1995)

Example of Pattern Completion



(fig. from Arbib 1995)

Example of Pattern Completion



(fig. from Arbib 1995)

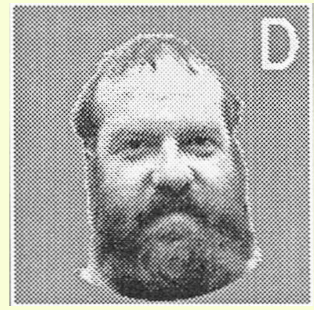
Example of Pattern Completion

(fig. from Arbib 1995)

Example of Pattern Completion

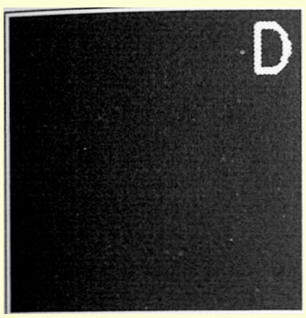
(fig. from Arbib 1995)

Example of Pattern Completion



(fig. from Arbib 1995)

Example of Association



(fig. from Arbib 1995)

Example of Association

(fig. from Arbib 1995)

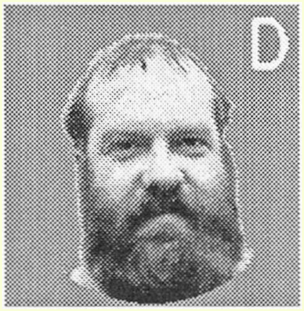
Example of Association

(fig. from Arbib 1995)

Example of Association

(fig. from Arbib 1995)

Example of Association



(fig. from Arbib 1995)

Applications of Hopfield Memory

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

Hopfield Net for Optimization and for Associative Memory

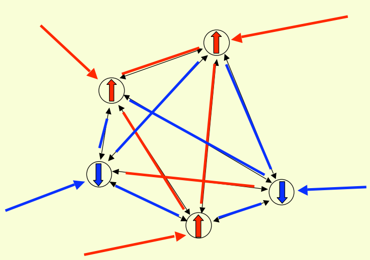
- For optimization:
  - we know the weights (couplings)
  - we want to know the minima (solutions)
- For associative memory:
  - we know the minima (retrieval states)
  - we want to know the weights

Hebb's Rule

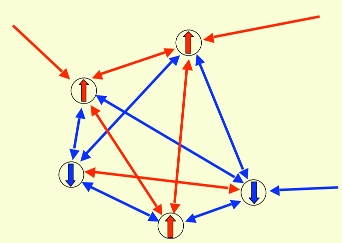
"When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."

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

Example of Hebbian Learning: Pattern Imprinted



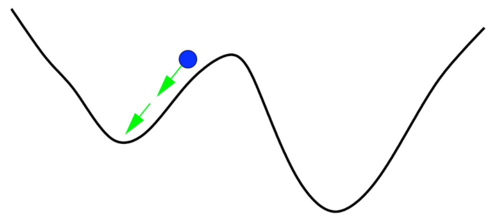
Example of Hebbian Learning: Partial Pattern Reconstruction



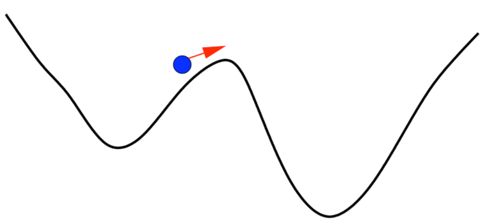
Stochastic Neural Networks

(in particular, the stochastic Hopfield network)

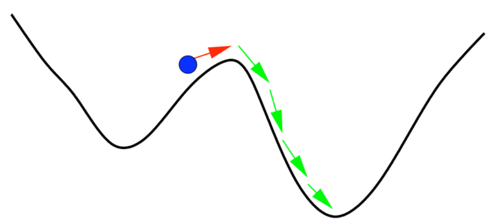
Trapping in Local Minimum



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

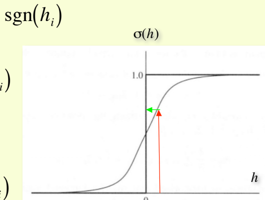
The Stochastic Neuron

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

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

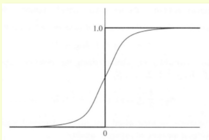
Stochastic neuron:

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



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

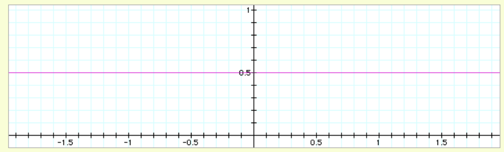
Properties of Logistic Sigmoid



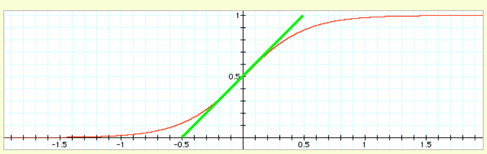
$$\sigma(h) = \frac{1}{1 + e^{-2h/T}}$$

- 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

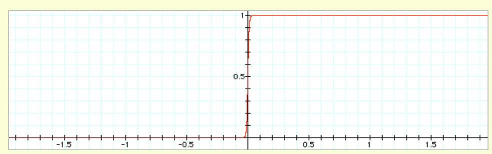


Logistic Sigmoid $T = 0.5$

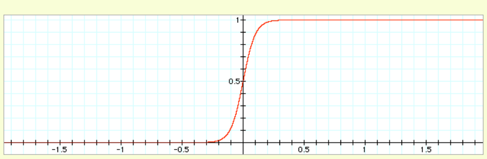


Slope at origin = 1 / 2T

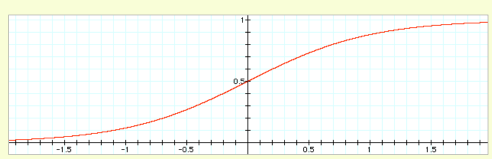
Logistic Sigmoid $T = 0.01$



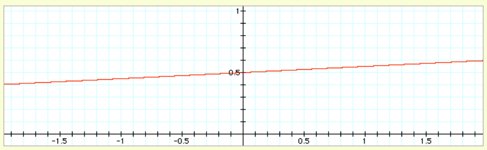
Logistic Sigmoid $T = 0.1$



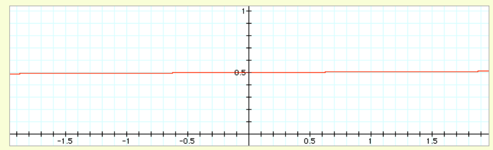
Logistic Sigmoid $T = 1$



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

Simulated Annealing

(Kirkpatrick, Gelatt & Vecchi, 1983)

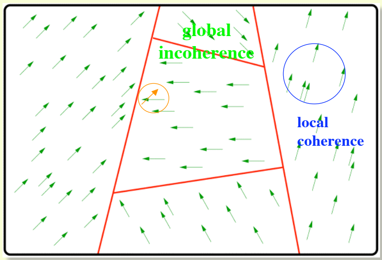
Dilemma

- 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

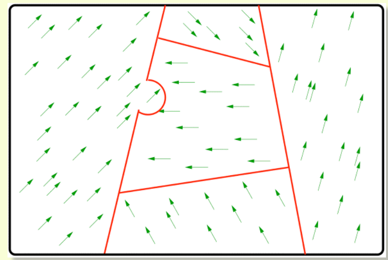
- Quenching:
  - rapid cooling of a hot material
  - may result in defects & brittleness
  - local order but global disorder
  - locally low-energy, globally frustrated
- Annealing:
  - slow cooling (or alternate heating & cooling)
  - reaches equilibrium at each temperature
  - allows global order to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

Multiple Domains



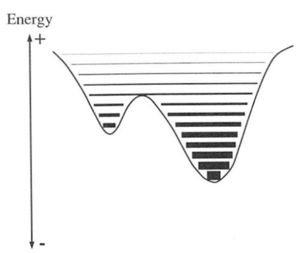
11/2/06 103

Moving Domain Boundaries



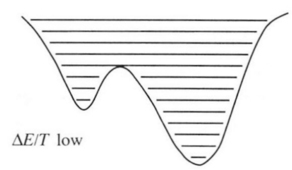
11/2/06 104

Effect of Moderate Temperature



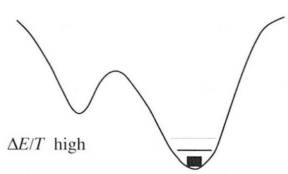
11/2/06 (fig. from Anderson *Intr. Neur. Comp.*) 105

Effect of High Temperature



11/2/06 (fig. from Anderson *Intr. Neur. Comp.*) 106

Effect of Low Temperature



11/2/06 (fig. from Anderson *Intr. Neur. Comp.*) 107

Annealing Schedule

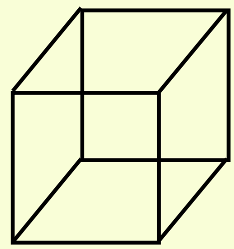
- Controlled decrease of temperature
- Should be sufficiently slow to allow equilibrium to be reached at each temperature
- With sufficiently slow annealing, the global minimum will be found with probability 1
- Design of schedules is a topic of research

11/2/06 108

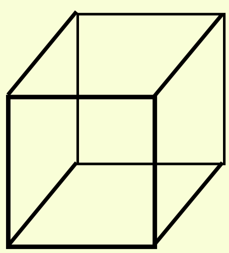
Demonstration of Boltzmann Machine & Necker Cube Example

Run ~mclennan/pub/cube/cubedemo

Necker Cube



Biased Necker Cube



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