What is good about back-propagation

Discovering useful internal representations

- Tradeoff between *memorization* and *generalization*
- Networks generalize on the basis of *similarity*
- Similarity in environmental input often does not indicate similarity in the appropriate behavioral (or internal) response
- Networks must learn to re-represent inputs (over internal "hidden" units) in a way that corresponds more closely to output similarity

Temporal processing

- Can learn to interpret time-varying input and produce time-varying output, including long-distance temporal dependencies

Efficiency

- Relatively efficient in terms of amount of experience required for learning (well, not really, but it's the best we've got)

What is bad about back-propagation

Aspects of unit functions, connectivity, architecture are inaccurate
(e.g., real-valued activations, no separation of excitation/inhibition)

- Approximations are *hypotheses* subject to revision/elaboration
- Can incorporate additional constraints into network structure and function

Back-propagation is implausible as a neural learning mechanism

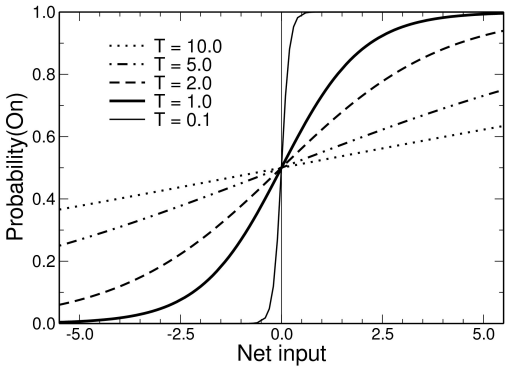
- Propagates "error signals" backwards across connections
  - ⇒ Other error-correcting learning proceedures only propagate activation (e.g., Contrastive Hebbian Learning)
- Requires explicit "target" activations for output units
  - ⇒ Effective feedback can come from multiple sources
    - Internal supervision (one part of the network teaching another)
    - Statistical distribution of inputs (unsupervised learning)
    - Evaluative feedback (reinforcement learning)
    - Implicit prediction ("self-supervised" learning)

Boltzmann Machines

Stochastic units (s_j is binary "state" of unit j)

$$p(s_j = 1) = \frac{1}{1 + \exp(-n_j/T)}$$

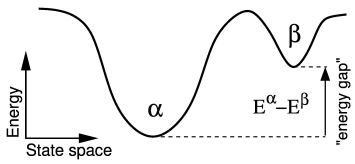
- temperature T adjusts steepness of sigmoid
- gradually lower T ("simulated annealing")



Thermal equilibrium: stable unit *probabilities*

- unit states still change; exhibits *probability distribution* over all possible global patterns

Energy (for global pattern α) $E^\alpha = -\sum_{i<j} s_i^\alpha \, s_j^\alpha \, w_{ij}$



Boltzmann distribution

$$\frac{p^\alpha}{p^\beta} = \frac{\exp(-E^\alpha/T)}{\exp(-E^\beta/T)} = \exp\left(-\left(E^\alpha - E^\beta\right)/T\right)$$

Boltzmann Machine learning

Run network in two "phases"

Negative phase: Clamp inputs only; run hidden and outputs units [cf. forward pass]

Positive phase: Clamp both inputs and outputs (targets); run hidden units [cf. backward pass]

Error function: Information gain $G = \sum_{\alpha,\beta} p^+(I_\alpha, O_\beta) \log \frac{p^+(O_\beta|I_\alpha)}{p^-(O_\beta|I_\alpha)}$

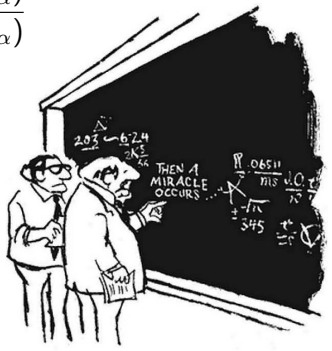
I_α	input units in pattern α
O_β	output units in pattern β
p^+, p^+	probabilities in positive, negative phases

$\Rightarrow$ "distance" between two probability distributions (pos. vs. neg. phases)

Contrastive Hebbian learning (CHL)

$$\frac{\partial G}{\partial w_{ij}} = -\frac{1}{T}\left(\langle s_i\ s_j\rangle^+ - \langle s_i\ s_j\rangle^-\right) \qquad \left[\textbf{Energy} = -\sum_{i<j} s_i\ s_j\ w_{ij}\right]$$

- Raise energy of negative phase (network's own performance)
- Lower energy of positive phase (performance based on "teacher")



Performance (Ackley, Hinton & Sejnowski, 1985)

- Works (for the most part) but is **very slow**
  - 4-2-4 encoder: average of 110 epochs
  - 8-3-8 encoder: average of 1570 epochs to solution; fails on 4/20
  - 40-10-40: 98.6% correct at 1200 epochs
  - Shifter: 9000 epochs; performance is far from perfect

- Sampling binary states at thermal equilibrium leads to noisy derivatives
  - Learning rule based on the *difference* between two noisy values

$$\frac{\partial G}{\partial w_{ij}} = -\frac{1}{T}\left(\langle s_i\ s_j\rangle^{+} - \langle s_i\ s_j\rangle^{-}\right)$$

$\Rightarrow$ "For every complex problem there's a solution that's simple, clean, and wrong."
—H. L. Mencken

Deterministic Boltzman Machines (DBMs)

Mean-field approximation: Replace binary stochastic variables (s_j) with real-valued variables (a_j) that represent the "expected values" (means) of the stochastic variables

$$a_j = \langle s_j \rangle = \frac{1}{1 + \exp\left(-n_j / T\right)}$$

standard sigmoid w/ temperature T

- Loses high-order structure (e.g., perfectly correlated vs. anticorrelated units)

s_i	s_j	vs.	s_i	s_j
1	1		0	1
0	0		1	0
1	1		0	1
0	0		1	0
⋮	⋮		⋮	⋮

$a_i = a_j = 0.5$

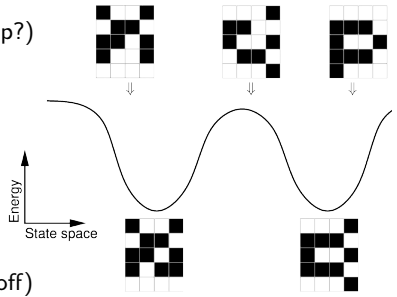
Contrastive Hebbian learning (CHL)

- Still run two phases ($+$ and $-$) but no sampling

$$\Delta w_{ij} = \epsilon \, \frac{1}{T} \left(a_i^+ a_j^+ - a_i^- a_j^- \right)$$

Comparison of DBM/CHL with recurrent back-propagation

- CHL only sends activations across weights, and only forwards (not derivatives backwards)
  - BP: Backward pass requires stored activations of forward pass
  - CHL: Can run two phases at different times (+ awake, − asleep?)
- Weights in DBM remain (or become) symmetric
  - CHL makes symmetric changes
  - weight decay would eliminate any small initial differences
- DBM intrinsically settles to stable activity patterns
  - Enforced by symmetric weights
  - Final pattern tends to be *fairly binary* (units completely on or off)

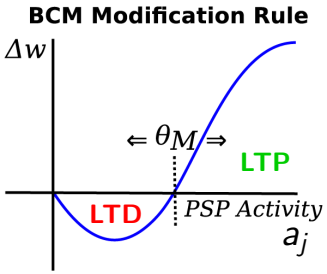


- DBM tends to need more hidden units to achieve comparable level of performance
  - Back-propagation is better at using real-valued activation differences
- Learning speed (number of epochs) and power (what problems can be solved) fairly comparable to recurrent back-propagation
  - However, DBM can only learn to form stable fixed-points/attractors; cannot generate continous trajectories as output (e.g. articulation)

Other generalizations of Hebbian learning

Bienenstock-Cooper-Munro (BCM) rule

- Strong post-synaptic response (a_j):
  long-term potentiation (LTP) *[increase (positive) weight]*
- Weak post-synaptic response (a_j):
  long-term depression (LTD) *[decrease (positive) weight]*



Spike-timing dependent plasticity (STDP)

- Presynaptic spike before postsynaptic spike:
  long-term potentiation (LTP) *[A could have caused B]*
- Presynaptic spike after postsynaptic spike:
  long-term depression (LTD) *[A didn't cause B]*