

Boltzmann Machines

Stochastic units

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

where temperature T adjusts steepness of sigmoid

- gradually lower temperature: simulated annealing

Energy (for global activation pattern α)

$$E^\alpha = - \sum_{i < j} a_i^\alpha a_j^\alpha w_{ij}$$

Thermal equilibrium

- exhibits *probability distribution* over all possible global states

Boltzmann distribution

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

Performance (Ackley, Hinton & Sejnowski, 1985)

Works (for the most part) but is VERY SLOW (Ackley et al, 1985)

- 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

For every problem there's a solution that's simple, clean, and wrong.

—H. L. Mencken

Boltzmann Machine learning

Objective function: Information gain (G)

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

I_α input units in pattern α

O_β output units in pattern β

p^+ probabilities in *positive* phase (both inputs and outputs clamped)

p^- probabilities in *negative* phase (only inputs clamped)

Contrastive Hebbian learning

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

Unsupervised version

$$G = \sum_{\alpha} p^+(V_\alpha) \log \frac{p^+(V_\beta)}{p^-(V_\beta)}$$

V_α visible units in pattern α

Mean-field approximation (Petersen & Anderson, 1987)

Replace binary stochastic variables with real-valued variables that represent the means (averages) of the stochastic variables

$$p_j = \langle s_j \rangle = \frac{1}{1 + \exp(-n_j/T)} \quad \text{standard sigmoid w/ temperature } T$$

Lose 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
\vdots	\vdots		\vdots	\vdots
$p_i = p_j = 0.5$				

Mean-field learning

$$\Delta w_{ij} = \varepsilon \frac{1}{T} \left(p_i^+ p_j^+ - p_i^- p_j^- \right)$$

Learning speed and power comparable to back-propagation; running takes longer

GRAIN (McClelland, 1993; Movellan & McClelland, 1993)

Graded Random Adaptive Interactive Networks

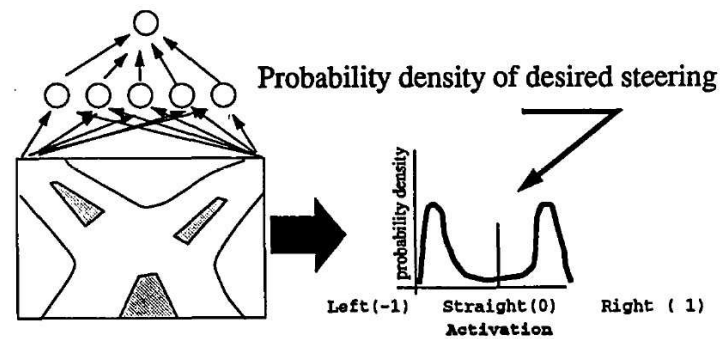
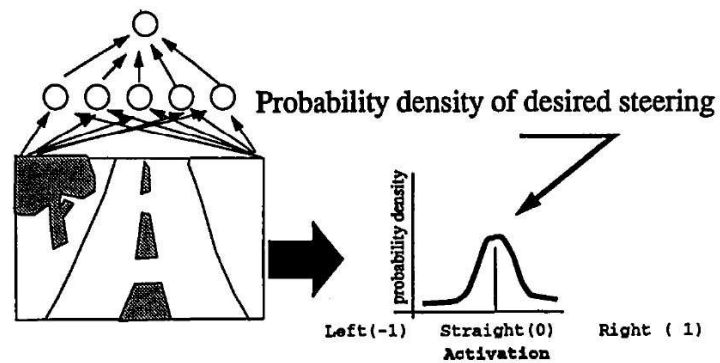
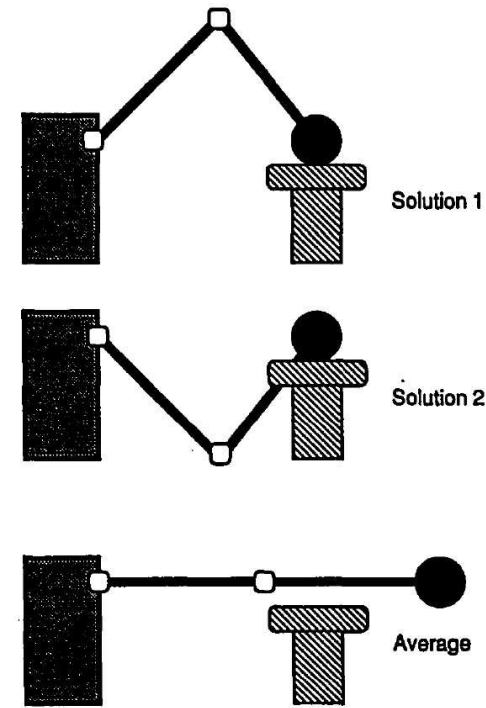
Real-valued stochastic units

$$n_j = \sum_i a_i w_{ij} + N(0, sd) \quad (\text{add zero - mean Gaussian noise})$$

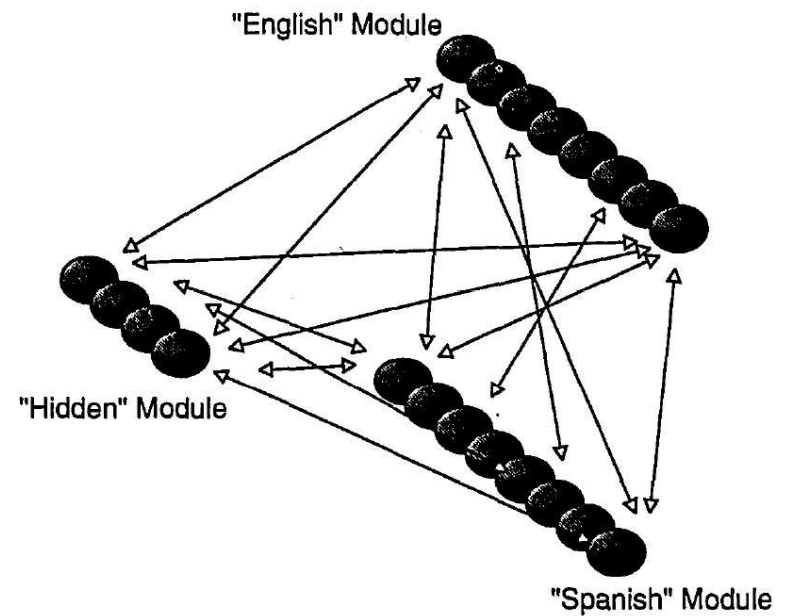
$$a_j = \frac{1}{1 + \exp(-n_j)} \quad (\text{tend not to use temperature/annealing})$$

$$\Delta w_{ij} = \varepsilon \left(\langle p_i p_j \rangle^+ - \langle p_i p_j \rangle^- \right)$$

Have to sample like Boltzmann machines; can represent and learn continuous probability distributions



Word Translation Problem



Translation Problem

Input	Translation				
house	casa	1.000	[1.000]		
home	casa	1.000	[1.000]		
do	hacer	1.000	[1.000]		
make	hacer	1.000	[1.000]		
olive	aceituna	0.700	[0.657]	oliva	0.300 [0.289]
be	ser	0.500	[0.495]	estar	0.500 [0.486]
casa	house	0.700	[0.674]	home	0.300 [0.303]
hacer	do	0.500	[0.464]	make	0.500 [0.531]
aceituna	olive	1.000	[1.000]		
oliva	olive	1.000	[1.000]		
ser	be	1.000	[1.000]		
estar	be	1.000	[1.000]		

