Neurophysics lecture 4

October 22, 2012

- Storage capacity of Hopfield model
  - paramagnetism, ferromagnetism
  - frustration, spin glass
- Boltzmann Machines

Bert Kappen

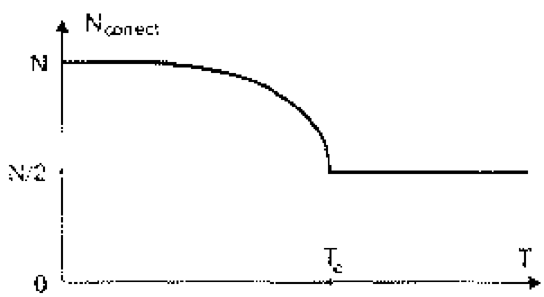
Mean field description of Hopfield model

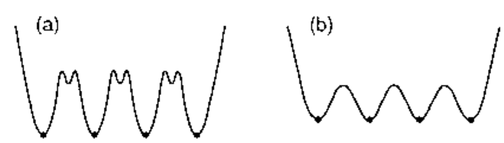
$$w_{ij} = \frac{\beta}{n} \sum_{\mu} \xi_i^{\mu} \xi_j^{\mu}$$

$$m_i = \tanh(\frac{\beta}{n} \sum_{\mu,j} \xi_i^{\mu} \xi_j^{\mu} m_j)$$

We analyse the system close to one pattern $m_i = m\xi_i^{\nu}$

$$\begin{aligned} m\xi_i^{\nu} &= \tanh(\frac{\beta}{n} \sum_{\mu,j} \xi_i^{\mu} \xi_j^{\mu} m \xi_j^{\nu}) \\ &= \tanh(\beta m \xi_i^{\nu} + \text{cross terms}) \end{aligned}$$

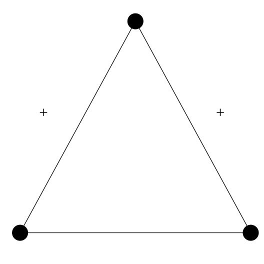




The SK model

To understand better the network behavior for different connection patterns we consider a fully connected network with Gaussian distributed weights (mean= $\frac{J0}{n}$, $\sigma = \frac{J}{\sqrt{n}}$, $\theta_i = 0$).

Large weights of opposite sign leads to *frustration*



One can define two *order parameters*

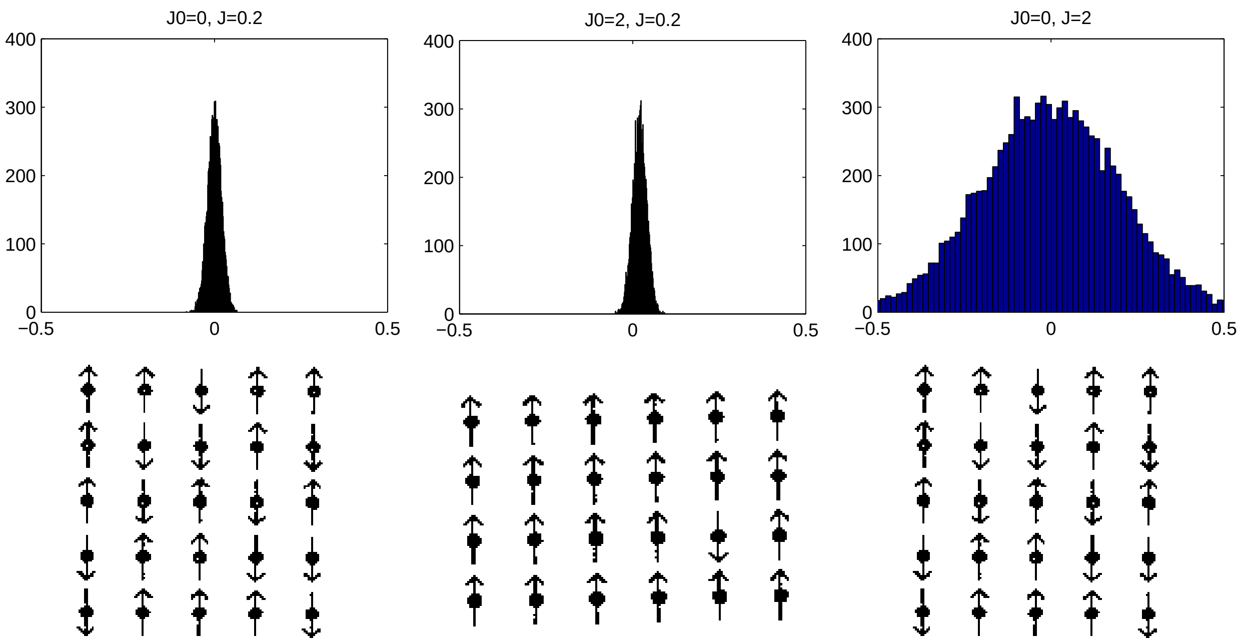
$$m = \frac{1}{n}\sum_i \langle s_i \rangle$$

$$q = \frac{1}{n}\sum_i \langle s_i \rangle^2$$

which obey the equations

$$m = \frac{1}{\sqrt{2\pi}} \int dz e^{-\frac{z^2}{2}} \tanh(\sqrt{qJ^2}z + J_0 m)$$

$$q = \frac{1}{\sqrt{2\pi}} \int dz e^{-\frac{z^2}{2}} \tanh^2(\sqrt{qJ^2}z + J_0 m)$$

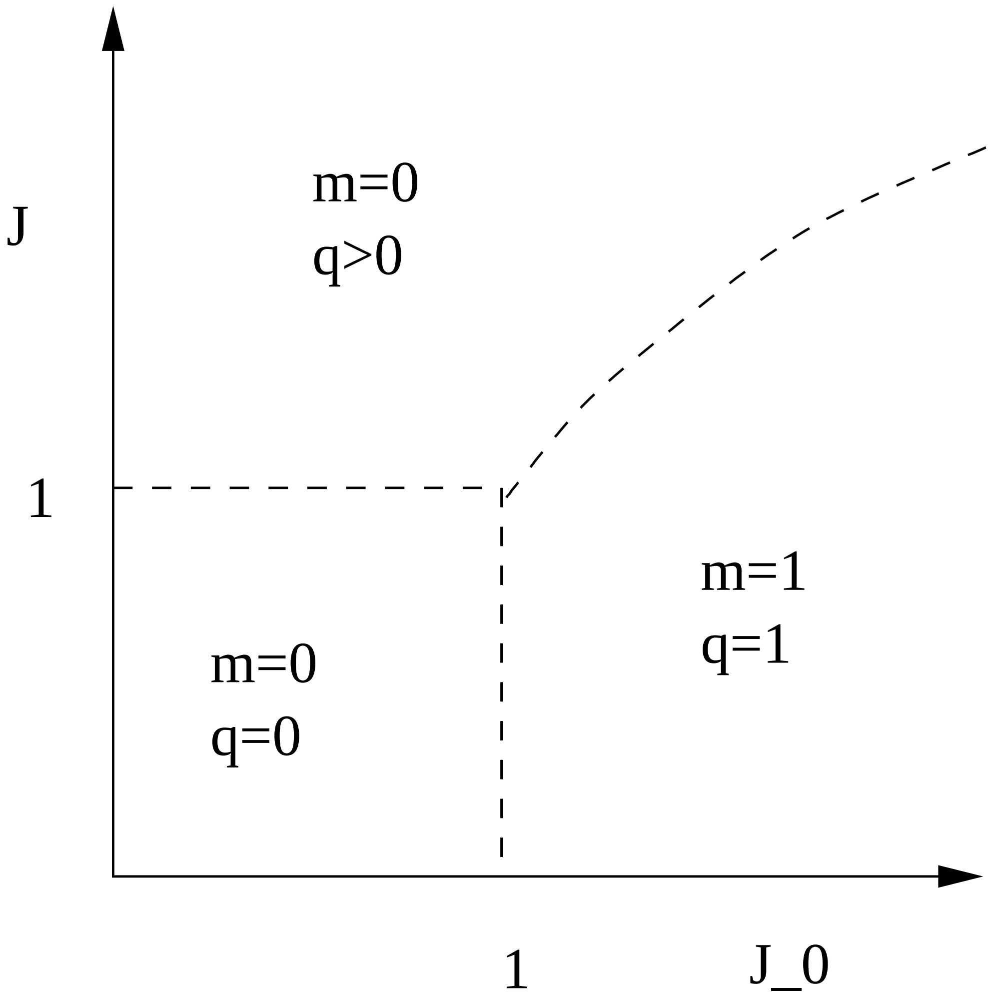


Paramagnetic
m=q=0

Ferromagnetic
m=q='1'

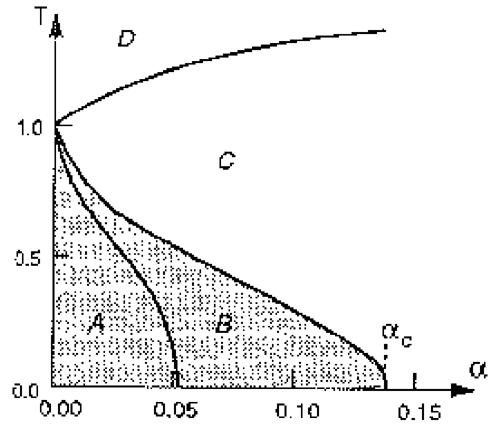
Spin glass
m=0, $q > 0$

SK phase plot

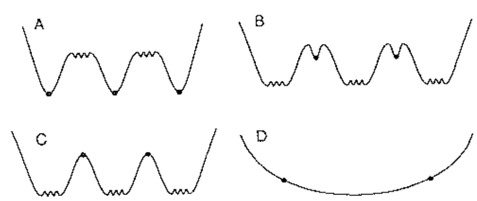


Phase plot Hopfield model

many patterns → large weights → frustration

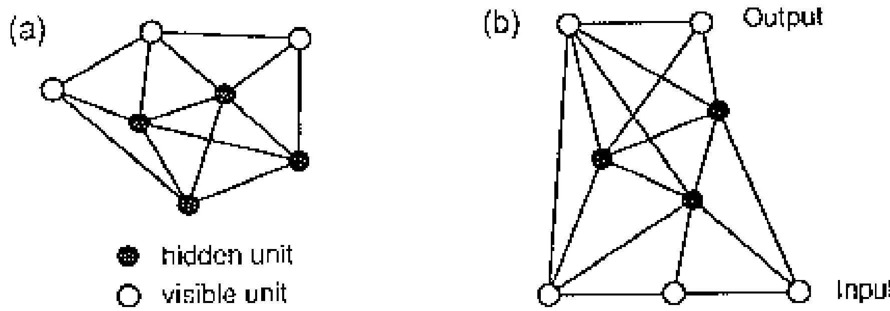


$$\alpha = p/n$$



- for $T = 0$ $\alpha > 0.138$ frustration
- for $\alpha = 0$ $T < 1$ paramagnetism

Learning in stochastic networks: Boltzmann Machines



Hebb learning only works well for random patterns. How about storing some given data set? One can distinguish *unsupervised* and *supervised* learning.

Sequential dynamics and symmetric weights yields

$$
\begin{aligned}
p(s) &= \frac{1}{Z}\exp(-E(s)) \\
-E(s) &= \frac{1}{2}\sum_{ij} w_{ij}s_i s_j + \sum_i \theta_i s_i. \\
Z &= \sum_s \exp(-E(s))
\end{aligned}
$$

Label the states separately for hidden and visible units $s = (\alpha, \beta)$. Thus,

$$s_i^{\alpha\beta} \qquad E(s) \to E_{\alpha\beta} \qquad p(s) \to p_{\alpha\beta}$$

The probability distribution restricted to the visible units is

$$p_\alpha = \sum_\beta p_{\alpha\beta}$$

Consider unsupervised learning. We wish to train the network on a data set given by patterns

$$\xi_i^\mu, \quad i = 1, \ldots, n_{\text{visible}}, \quad \mu = 1, \ldots, P$$

When the occurance of each pattern is equally likely, one can equivalently consider the *target distribution*

$$q_\alpha = \frac{1}{P} \sum_{\mu=1}^{P} \delta_{\alpha, \xi^\mu}$$

Learning consists of finding w, θ such that q and p are as similar as possible. A good measure for similarity of probability distributions is the Kullback-Leibler divergence:

$$K = \sum_\alpha q_\alpha \log \left(\frac{q_\alpha}{p_\alpha} \right)$$

Properties: $K \geq 0$ and $K = 0$ $p_\alpha = q_\alpha$ for all α The proof is easy from Jensen's inequality:

$$-\sum_\alpha q_\alpha \log \left(\frac{q_\alpha}{p_\alpha} \right) = \left\langle \log \left(\frac{p_\alpha}{q_\alpha} \right) \right\rangle_q$$

$$\leq \log \left\langle \frac{p_\alpha}{q_\alpha} \right\rangle_q = \log \sum_\alpha p_\alpha = 0$$

BM learning rules

Learning consists of *gradient descent* on the KL divergence.

1. start with random w_{ij}
2. compute the gradients $\Delta w_{ij} = -\eta \frac{\partial K}{\partial w_{ij}}$
3. $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}$
4. stop or goto 2.

Similar for $\theta_i = w_{i0}$ with $s_0 = 1$.

The gradients are easily computed and are given by

$$\Delta w_{ij} = \eta \left(\langle s_i s_j \rangle_c - \langle s_i s_j \rangle \right)$$

with

$$\begin{aligned} \langle s_i s_j \rangle &= \sum_{\alpha\beta} s_i^{\alpha\beta} s_j^{\alpha\beta} p_{\alpha\beta} \\ \langle s_i s_j \rangle_c &= \sum_{\alpha\beta} s_i^{\alpha\beta} s_j^{\alpha\beta} q_\alpha p_{\beta|\alpha} \end{aligned}$$

Note, that when i is visible, $s_i^{\alpha\beta} = s_i^\alpha$ and

$$\langle s_i \rangle_c = \sum_\alpha s_i^\alpha q_\alpha = \frac{1}{P} \sum_\mu \xi_i^\mu$$

is just the mean value of s_i in the training set.

In the supervised case, the KL divergence and learning rules become

$$\begin{aligned}
\Delta w_{ij} &= \eta \left(\langle s_i s_j \rangle_{I,O} - \langle s_i s_j \rangle_I \right) \\
\Delta \theta_i &= \eta \left(\langle s_i \rangle_{I,O} - \langle s_i \rangle_I \right)
\end{aligned}$$

Since computation of averages is intractable we need approximations

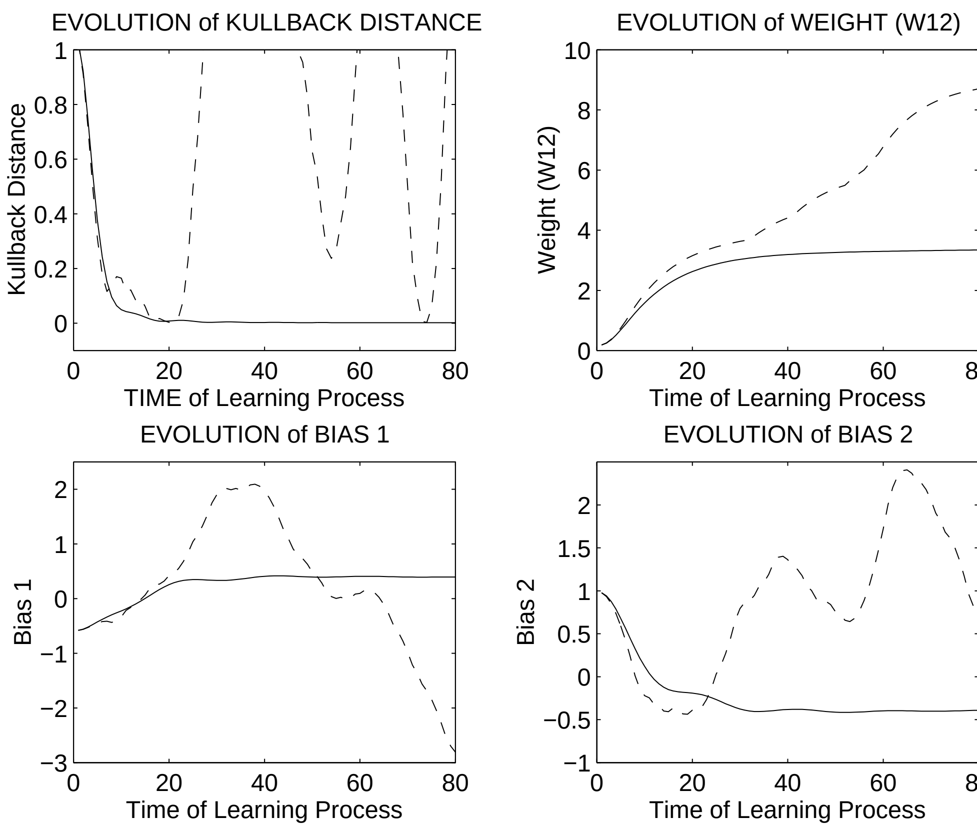
- MCMC, a sampling technique
- mean field theory

Mean field learning

Naive approach (Peterson 1987):

$$\Delta w_{ij} = \eta \left(\langle s_i s_j \rangle_c - m_i m_j \right), \quad \Delta\theta_i = \eta \left(\langle s_i \rangle_c - m_i \right).$$

$$m_i = \tanh\left(\sum_j w_{ij} m_j + \theta_i\right)$$



Linear response correction

$$
\begin{aligned}
\langle s_i \rangle &= \frac{\partial \log Z}{\partial \theta_i} \\
\chi_{ij} &= \langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle = \frac{\partial^2 \log Z}{\partial \theta_i \partial \theta_j}
\end{aligned}
$$

implies

$$
\chi_{ij} = \frac{\partial \langle s_i \rangle}{\partial \theta_j} = \frac{\partial m_i}{\partial \theta_j}
$$

with

$$
\begin{aligned}
m_i &= \tanh(\sum_k w_{ik} m_k + \theta_i) \\
\frac{\partial m_i}{\partial \theta_j} &= (1 - m_i^2) \left(\sum_k w_{ik} \frac{\partial m_k}{\partial \theta_j} + \delta_{ij} \right) \\
\delta_{ij} &= \sum_k \left(\frac{\delta_{ik}}{1 - m_i^2} - w_{ik} \right) \frac{\partial m_k}{\partial \theta_j} \\
\left(\chi^{-1} \right)_{ij} &= \frac{\delta_{ij}}{1 - m_i^2} - w_{ij}
\end{aligned}
$$

A (very) fast learning rule

In the absence of hidden units we compute

$$m_i = \langle s_i \rangle_c \,\text{and}\; c_{ij} = \langle s_i s_j \rangle_c - m_i m_j$$

directly from the training data.

We then compute

$$\begin{aligned} w_{ij} &= \frac{\delta_{ij}}{1 - m_i^2} - \left(c^{-1}\right)_{ij} \\ \theta_i &= \tanh^{-1}(m_i) - \sum_j w_{ij} m_j \end{aligned}$$

Complexity is $\mathcal{O}(Pn^3)$. Success depends on

- need for hidden units
- inversion of c

OCR

8×8 handwritten digits
7000 training patterns and 4000 test patterns
Train one Boltzmann distribution per class, no hidden units

$$
\begin{aligned}
p(s) &= \frac{1}{Z}\exp(-E(s)) \\
\log Z &\approx -\langle E\rangle_q - \langle \log q\rangle_q \\
-\langle E\rangle_q &= \frac{1}{2}\sum_{ij} w_{ij} m_i m_j + \sum_i \theta_i m_i \\
\langle \log q\rangle_q &= \frac{1}{2}\sum_i \Bigg((1+m_i)\log\frac{1}{2}(1+m_i) \\
&\quad + (1-m_i)\log(\frac{1}{2}(1-m_i))\Bigg)
\end{aligned}
$$

Matrix c is singular. We add a flat distribution to the training data:

$$
\begin{aligned}
q_\alpha &\rightarrow (1-\lambda) q_\alpha + \lambda \frac{1}{2^n} \\
\langle s_i \rangle_c &\rightarrow (1-\lambda) \langle s_i \rangle_c \\
\langle s_i s_j \rangle_c &\rightarrow (1-\lambda) \langle s_i s_j \rangle_c + \lambda \delta_{ij}
\end{aligned}
$$

nearest neighbor	6.7 %
back-propagation	5.6 %
wake-sleep	4.8 %
sigmoid belief	4.6 %
Boltzmann Machine	4.6 %