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

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

\[ m_i = \tanh\left(\frac{\beta}{n} \sum_{\mu,j} \xi_i^\mu \xi_j^\mu m_j\right) \]

We analyse the system close to one pattern \( m_i = m \xi_i^\nu \)

\[ m \xi_i^\nu = \tanh\left(\frac{\beta}{n} \sum_{\mu,j} \xi_i^\mu \xi_j^\mu m \xi_j^\nu\right) \]

\[ = \tanh(\beta m \xi_i^\nu + \text{cross terms}) \]
To understand better the network behavior for different connection patterns we consider a fully connected network with Gaussian distributed weights (mean $J_0/n$, $\sigma = J/\sqrt{n}$, $\theta_i = 0$).

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

\[ m = \frac{1}{\sqrt{2\pi}} \int dz e^{-\frac{z^2}{2}} \tanh(\sqrt{qJ^2}z + J_0m) \]

\[ q = \frac{1}{\sqrt{2\pi}} \int dz e^{-\frac{z^2}{2}} \tanh^2(\sqrt{qJ^2}z + J_0m) \]

Paramagnetic
\[ m=q=0 \]

Ferromagnetic
\[ m=q='1' \]

Spin glass
\[ m=0, \ q > 0 \]
SK phase plot

\[ J \]

\[ m=0 \]
\[ q>0 \]

\[ m=0 \]
\[ q=0 \]

\[ m=1 \]
\[ q=1 \]

1

1

\( J_0 \)
Phase plot Hopfield model

many patterns → large weights → frustration

\[ \alpha = \frac{p}{n} \]

- 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

\[
p(s) = \frac{1}{Z} \exp(-E(s))
\]

\[
-E(s) = \frac{1}{2} \sum_{ij} w_{ij} s_i s_j + \sum_i \theta_i s_i.
\]

\[
Z = \sum_s \exp(-E(s))
\]
Label the states separately for hidden and visible units $s = (\alpha, \beta)$. Thus,

$$s_i^{\alpha\beta} \quad E(s) \rightarrow E_{\alpha\beta} \quad p(s) \rightarrow p_{\alpha\beta}$$

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 occurrence 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 $w, \theta$ 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 \( \alpha \) 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

$$\langle s_i s_j \rangle = \sum_{\alpha\beta} s_i^{\alpha\beta} s_j^{\alpha\beta} p^{\alpha\beta}$$

$$\langle s_i s_j \rangle_c = \sum_{\alpha\beta} s_i^{\alpha\beta} s_j^{\alpha\beta} q^{\alpha\beta} p^{\alpha\beta}_{|\alpha}$$
Note, that when $i$ is visible, $s_i^{\alpha\beta} = s_i^\alpha$ and

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

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

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

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

Since computation of averages is intractable we need approximations

- MCMC, a sampling technique
- mean field theory
Mean field learning

Naive approach (Peterson 1987):

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

\[ m_i = \tanh \left( \sum_j w_{ij} m_j + \theta_i \right) \]
Linear response correction

\[
\langle s_i \rangle = \frac{\partial \log Z}{\partial \theta_i}
\]

\[
\chi_{ij} = \langle s_is_j \rangle - \langle s_i \rangle \langle s_j \rangle = \frac{\partial^2 \log Z}{\partial \theta_i \partial \theta_j}
\]

implies

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

with

\[
m_i = \tanh(\sum_k w_{ik} m_k + \theta_i)
\]

\[
\frac{\partial m_i}{\partial \theta_j} = (1 - m_i^2) \left( \sum_k w_{ik} \frac{\partial m_k}{\partial \theta_j} + \delta_{ij} \right)
\]

\[
\delta_{ij} = \sum_k \left( \frac{\delta_{ik}}{1 - m_i^2} - w_{ik} \right) \frac{\partial m_k}{\partial \theta_j}
\]

\[
(\chi^{-1})_{ij} = \frac{\delta_{ij}}{1 - m_i^2} - w_{ij}
\]
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

\[
\begin{align*}
    w_{ij} &= \frac{\delta_{ij}}{1 - m_i^2} - (c^{-1})_{ij} \\
    \theta_i &= \tanh^{-1}(m_i) - \sum_j w_{ij} m_j
\end{align*}
\]

Complexity is \( O(Pn^3) \). Success depends on

• need for hidden units

• inversion of \( c \)
8 × 8 handwritten digits
7000 training patterns and 4000 test patterns
Train one Boltzmann distribution per class, no hidden units

\[ p(s) = \frac{1}{Z} \exp(-E(s)) \]

\[ \log Z \approx -\langle E \rangle_q - \langle \log q \rangle_q \]

\[ -\langle E \rangle_q = \frac{1}{2} \sum_{ij} w_{ij} m_i m_j + \sum_i \theta_i m_i \]

\[ \langle \log q \rangle_q = \frac{1}{2} \sum_i \left( (1 + m_i) \log \frac{1}{2}(1 + m_i) + (1 - m_i) \log(\frac{1}{2}(1 - m_i)) \right) \]

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

\[ q_\alpha \rightarrow (1 - \lambda) q_\alpha + \lambda \frac{1}{2^n} \]

\[ \langle s_i \rangle_c \rightarrow (1 - \lambda) \langle s_i \rangle_c \]

\[ \langle s_is_j \rangle_c \rightarrow (1 - \lambda) \langle s_is_j \rangle_c + \lambda \delta_{ij} \]

<table>
<thead>
<tr>
<th>Method</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>nearest neighbor</td>
<td>6.7 %</td>
</tr>
<tr>
<td>back-propagation</td>
<td>5.6 %</td>
</tr>
<tr>
<td>wake-sleep</td>
<td>4.8 %</td>
</tr>
<tr>
<td>sigmoid belief</td>
<td>4.6 %</td>
</tr>
<tr>
<td>Boltzmann Machine</td>
<td>4.6 %</td>
</tr>
</tbody>
</table>