Radial Basis Boltzmann Machines and learning with missing values

Hilbert J. Kappen, Marcel J. Nijman
RWCP Novel Function SNN Laboratory
Dept. of Medical Physics and Biophysics, University of Nijmegen
Geert Grooteplein 21, NL 6525 EZ Nijmegen, The Netherlands

Abstract

A Radial Basis Boltzmann Machine (RBBM) is a specialized Boltzmann Machine architecture that combines feed-forward mapping with probability estimation in the input space, and for which very fast learning rules exist. The hidden representation of the network displays symmetry breaking as a function of the noise in the Glauber dynamics. Thus generalization can be studied as a function of the noise in the neuron dynamics instead of as a function of the number of hidden units. For the special case of unsupervised learning, we show that this method is an elegant alternative of k nearest neighbor, leading to comparable performance without the need to store all data. We show that the RBBM has good classification performance compared to the MLP. The main advantage of the RBBM is that simultaneously with the input-output mapping, a model of the input space is obtained which can be used for learning with missing values. We show that the RBBM compares favorably to the MLP for large percentages of missing values.

1 Introduction

Recently, a specialized Boltzmann Machine architecture was developed that combines feed-forward mapping with probability estimation in the input space (Kappen, 1995). Because of its resemblance with Radial Basis Networks, it is called a Radial Basis Boltzmann Machine (RBBM). The architecture is shown in Fig. 1. The main differences in comparison with normal Boltzmann Machines (MBs) (Adley et al., 1985) are 1)

the presence of continuous valued neurons, and 2) lateral inhibition in the hidden layer. For the continuous valued neurons, the Glauber dynamics must be replaced by (for instance) a Langevin type dynamics:

\[ \dot{x}_i = -x_i + \sum_j w_{ij} s_j + \xi = -\frac{\partial E}{\partial x_i} + \xi \quad <\xi(t)\xi(t')> = \frac{1}{\beta} \delta(t-t'). \]  

(1)

\( \xi \) is Gaussian white noise with standard deviation \( \frac{1}{\sqrt{\beta}} \). The Glauber dynamics on the remaining discrete states \( s \) and \( \bar{y} \) is such, that they satisfy the detailed balance equation

\[ \frac{T(s', \bar{y} | \bar{s}', \bar{y}')}{T(s, \bar{y} | s', \bar{y})} = \exp(-\beta E(s', \bar{y}') / \exp(-\beta E(s, \bar{y})) \]  

(2)

---

1Submitted to World Congress on Neural Networks 1995, rp: galba.mbfys.kun.nl Kappen.RBBM.ps.Z 4 pages
with the same $E$ and $\beta$ as in Eq. 1:

$$E = \frac{1}{2} \sum_{i=1}^{n} x_i^2 - \sum_{i=0}^{n} \sum_{j=0}^{h} w_{ij} x_i s_j - \sum_{j=0}^{h} \sum_{k=0}^{m} v_{jk} s_j y_k + J \sum_{j,j'=1}^{h} s_j s_{j'}$$

The lateral inhibition in the hidden layer leads to additional terms in the local field contributions for the hidden neurons. These contributions can be chosen such that they result in the last term in $E$ above and that for infinitely large inhibition only a subset of the hidden states $\hat{s}_j = (0 \ldots 01 \ldots 0)$ and $\hat{s}_0 = (0 \ldots 0)$ are possible, and all other hidden states have vanishing small probabilities (see (Kappen, 1995) for extensions). It can be shown, that the combined dynamics Eqs. 1 and 2 leads to an equilibrium Boltzmann distribution $\propto \exp(-\beta E)$. Summing over all hidden states leads to the joint probability for $\vec{x}$ and $\vec{y}$

$$p(\vec{x}, \vec{y}) = \frac{1}{Z} \prod_{j=0}^{h} \exp\{-\beta \|\vec{a}_j - \vec{x}\|^2 + \sum_{k=0}^{m} v_{jk} y_k\}$$

$$Z = (\pi/\beta)^{n/2} \prod_{j=0}^{h} \cosh(v_{jk})$$

Note that $Z$ contains $h$ terms, whereas in the absence of inhibition it contains $2^h$ terms. The absence of this exponential sum solves the problem of excessive learning times, as is common in standard BMs (Kappen, 1995).

Learning in this network is defined as gradient descent on the Kullback-Leibler divergence, which for a finite training set of $P$ patterns is equivalent to the expected logarithmic probability:

$$K = \frac{1}{P} \sum_{p} \log(p(\vec{x}^p, \vec{y}^p))$$

(3)

2 Symmetry breaking and classification

The hidden representation of the network displays symmetry breaking as a function of the noise in the Glauber dynamics. This phenomenon was first described by (Rose et al., 1990). In fact, for unsupervised learning, $m = 0$ and $v_{j0} = 0$, minimization of $K$ is identical to the minimization of their free energy. For different $\beta$, qualitatively different solutions for $\vec{a}_j$ are obtained: for high noise the optimal weights satisfy $\vec{a}_1 = \ldots = \vec{a}_h$ and all weight vectors point to the center of gravity of the data set. Thus, all hidden units do the same and the effective number of hidden units is 1. For lower noise levels, hidden units specialize to subsets of the data. Thus generalization can be studied as a function of the noise in the neuron dynamics instead of as a function of the number of hidden units.

As an illustration, consider an unsupervised learning problem on a data set of 1-dimensional continuous data. The training set is $\{-8,-4,4,8\}$ and the test set is $\{-9,-5,-3,3,5,7,9\}$. The RBBM consists of 1 continuous external neuron and 20 hidden neurons. Figure 2 shows the learning process. For low values of $\beta$ all weights $w_{ij}$ are equal to the center of mass of the training set. At $\beta = 1.2$ the set of hidden units breaks into 2 groups, namely one with its center at $x = -6$ and one at $x = 6$. At $\beta = 8.5$ a second breaking occurs. For higher values of $\beta$ the hidden units code the training data, i.e., the network is overtrained. The optimal value of $K$ on the test set is found for $\beta = 5$, and so the BM concludes that there are 2 clusters of data.

Figure 3 shows the learning process for a 1-dimensional classification problem. The data consists of random drawings from 2 Gaussians centered at 0, but with different variance. The Gaussians represent different classes that are a-priori equally probable. The training set and test set both consist of 100 points. We used 1000 hidden units. For low values of $\beta$ the classification of the training set and the test set is both 50%, because the network represents each class by its centers, which coincide (LS). For high values of $\beta$, extreme specialization occurs, and the hidden unit that is closest to the data point determines the classification. With more hidden units than data points, this behaviour is identical to 1-nearest neighbor classification. Optimal Kullback and classification coincide for $\beta = 2$ and results in 68% correct classification. Similar performance can also be obtained with k-nearest neighbors, with $k = 5$.

The Kullback divergence provides an objective criterion by which different "clusterings" can be compared. This is an important advantage over more heuristic approaches such as Parzen window or nearest neighbor clustering (Duda and Hart, 1973). Another advantage of the RBBM is that it yields a set of optimal weights
with which subsequent classification can be performed, whereas for the k-nearest neighbor algorithm all training data must be saved.

The symmetry breaking phenomenon can be used effectively to obtain high quality solutions. For $\beta \to \infty$, the algorithm becomes identical to Vector Quantization, which is known to suffer from severe local minima problems. Continuous variation of $\beta$ from small to large values provides an annealing schedule that yields better solutions (Rose et al., 1992).

In Table 1 we compare the classification performance of the RBBM and the MLP on a number of well-known benchmarks (Prechelt, 1994). For each of these problems, half of the data was used for training and one quarter for optimizing $\beta$ or the number of hidden units. The remaining one quarter of the data was used as test set to evaluate the performance. Note that the RBBM simultaneously solves a classification problem

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
 & horse & glass & cancer \\
\hline
& h & test set & h & test set & h & test set \\
RBBM & 44 & 74.7 & 66 & 69.8 & 119 & 97.7 \\
mlp & 4 & 73.6 & 8 & 67.1 & 4+2 & 98.2 \\
\hline
\end{tabular}
\caption{Classification performance on independent test data of the RBBM with optimal $\beta$ and a MLP with optimal number of hidden units. The optimal number of hidden units and the % correct classification are shown. ‘Horse’ consists of 364 records with 58 input fields and 3 classes. ‘glass’ consists of 214 records with 9 input fields and 6 classes. ‘cancer’ consists of 699 records with 9 input fields and 2 classes.}
\end{table}

and an unsupervised learning problem in the input space (joint probability estimation). For this reason, far more hidden units are used. We conclude that the RBBM has good classification performance compared to the MLP, despite the fact that the joint optimization problem is far more complex.
3 Missing values

The main advantage of the RBBM is that simultaneously with the input-output mapping, a model of the input space is obtained which can be used for learning with missing values. In the RBBM, the missing values are treated as if they were hidden neurons and are integrated over during training. Let each training pattern consist of a ‘known’ and a ‘missing’ part: \((\vec{x}^\mu, \vec{y}^\mu) = (\vec{x}^\mu_{k(\mu)}, \vec{x}^\mu_{m(\mu)}, \vec{y}^\mu_{k(\mu)}, \vec{y}^\mu_{m(\mu)})\), where \(k(\mu)\) and \(m(\mu)\) denote the known and missing part of pattern \(\mu\), respectively. Learning with missing values simply consists of replacing equation (3) by

\[
K = -\frac{1}{P} \sum_{\mu} \log(p(\vec{x}^\mu_{k(\mu)}, \vec{y}^\mu_{k(\mu)})) \text{ with } p(\vec{x}^\mu_k, \vec{y}^\mu_k) = \int d\vec{x}_m \sum_{\vec{y}_m} p(\vec{x}^\mu_k, \vec{x}_m, \vec{y}^\mu_k, \vec{y}_m)
\]

and performing gradient descent on \(K\) (Nijman and Kappen, 1995). Conceptually, this method consists of substituting the missing values with expectation values generated by the RBBM during training. A similar method, based on EM, was developed by (Ghaemmaghami and Jordan, 1994). To test the performance of this method, we compared it with an alternative, where the missing values are filled with the corresponding value of the nearest neighbor from the same class (NN). Table 2 shows for the ‘iris’ and ‘glass’ data bases that the RBBM obtains better classification on independent test data than when the missing values are ‘repaired’ using a 1-nearest neighbor approach. The main reason is that the RBBM uses not only information from the nearest pattern, but from the complete joint probability distribution to fill in the missing values.

<table>
<thead>
<tr>
<th></th>
<th>iris</th>
<th>glass</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 %</td>
<td>70 %</td>
</tr>
<tr>
<td>RBBM</td>
<td>94</td>
<td>88</td>
</tr>
<tr>
<td>1-NN</td>
<td>94</td>
<td>85</td>
</tr>
</tbody>
</table>

Table 2: Classification performance (% correct) on an uncorrupted test set with training on ‘iris’ and ‘glass’ data with 0, 70 and 90 % missing values.

References


