Boltzmann Machines: supplement to Lectures 2 and 3

Iasonas Kokkinos

October 23, 2015

This note follows the notation of David MacKay's book, `http://www.inference.phy.cam.ac.uk/itprnn/book.pdf` as this was also used in the class slides. Please be wary of any deviations with respect to the notation of Kevin Murphy's book, included in your supplementary course handouts.

1 Boltzmann Machine, Ising Model, RBMs

We use the following notation for the random variables in the distribution of the Boltzmann machine:

$\mathbf{x}$	$N \times 1$	Observable variables
$\mathbf{h}$	$K \times 1$	Hidden variables
$\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{h} \end{bmatrix}$	$(N+K) \times 1$	Joint hidden and observable variable vector

The energy of a general Boltzmann Machine (BM) is defined as:

$$E_{BM}(\mathbf{y}; \mathbf{W}) = -\frac{1}{2}\mathbf{y}^T \mathbf{W} \mathbf{y} \quad (1)$$

$$E_{BM}(\mathbf{x}, \mathbf{h}; \mathbf{W}) = -\frac{1}{2}[\mathbf{x}^T \mathbf{h}^T] \begin{bmatrix} \mathbf{W}_{\mathbf{x},\mathbf{x}} & \mathbf{W}_{\mathbf{x},\mathbf{h}} \\ \mathbf{W}_{\mathbf{x},\mathbf{h}}^T & \mathbf{W}_{\mathbf{h},\mathbf{h}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{h} \end{bmatrix} \quad (2)$$

$$= -\frac{1}{2}\mathbf{x}^T \mathbf{W}_{\mathbf{x},\mathbf{x}} \mathbf{x} - \mathbf{x}^T \mathbf{W}_{\mathbf{x},\mathbf{h}} \mathbf{y} - \frac{1}{2}\mathbf{y}^T \mathbf{W}_{\mathbf{h},\mathbf{h}} \mathbf{y} \quad (3)$$

The general Ising, or, 'Hopfield' model is a special case where no hidden variables are involved, $\mathbf{y} = \mathbf{x}$:

$$E_{Ising}(\mathbf{x}; \mathbf{W}_{\mathbf{x},\mathbf{x}}) = -\frac{1}{2}\mathbf{x}^T \mathbf{W}_{\mathbf{x},\mathbf{x}} \mathbf{x}. \quad (4)$$

The Restricted Boltzmann Machine (RBM) is a special case of Eq. 3 for $\mathbf{W}_{\mathbf{x},\mathbf{x}} = \mathbf{0}, \mathbf{W}_{\mathbf{h},\mathbf{h}} = \mathbf{0}$:

$$E_{RBM}(\mathbf{x}, \mathbf{h}; \mathbf{W}_{\mathbf{x},\mathbf{h}}) = -\mathbf{x}^T \mathbf{W}_{\mathbf{x},\mathbf{h}} \mathbf{y}, \quad (5)$$

i.e. no 'lateral' interactions between the observable, or the hidden variables are included.

In all of the above equations the quantities $\mathbf{W}_{\mathbf{x},\mathbf{h}}, \mathbf{W}_{\mathbf{x},\mathbf{x}}, \mathbf{W}$ appear after a semicolon, indicating that these are the parameters of the respective models. Below we will use $E(\mathbf{y}; \mathbf{W})$ as a common notation for any of the above energy models.

Under the Boltzmann-Gibbs distribution we have:

$$P(\mathbf{y}; \mathbf{W}) = \frac{1}{Z}\exp(-E(\mathbf{y}; \mathbf{W})) = \frac{1}{Z}\exp(\sum_{i,j>i} \mathbf{W}_{i,j}\mathbf{y}_i\mathbf{y}_j), \quad (6)$$

where $E(\mathbf{y};\mathbf{W}) \doteq E(\mathbf{x},\mathbf{h};\mathbf{W})$ and the $\frac{1}{2}$ factor in Eq. 1 has vanished from the last summation because we only sum over pairs of i,j where $j>i$ (i.e. we do not double-count network edges).

If we want to define a distribution over $\mathbf{x}$ we use marginalization: $P(\mathbf{x}) = \sum_{\mathbf{h}} P(\mathbf{x},\mathbf{h})$, which yields:

$$P(\mathbf{x}';\mathbf{W}) = \sum_{\mathbf{h}} P(\mathbf{x}',\mathbf{h};\mathbf{W}) = \sum_{\mathbf{h}} \frac{1}{Z}\exp(-E(\mathbf{x}',\mathbf{h};\mathbf{W})) = \sum_{\mathbf{y}:\mathbf{y}_{|\mathbf{x}}=\mathbf{x}'} P(\mathbf{y};\mathbf{W}), \tag{7}$$

where the last summation indicates that we are summing over all vectors $\mathbf{y}$ whose first N elements equal the respective elements of $\mathbf{x}'$.

2 Maximum Likelihood Parameter Estimation

Given a set of M training examples $\mathbf{X} = \{\mathbf{x}^1,\ldots,\mathbf{x}^M\}$, Boltzmann machine training aims at estimating $\mathbf{W}$ (or $\mathbf{W}_{\mathbf{x},\mathbf{x}}, \mathbf{W}_{\mathbf{x},\mathbf{h}}$ for Ising/RBM training respectively) so as to maximize the log-likelihood $\mathbf{X}$:

$$\begin{aligned} S(\mathbf{X},\mathbf{W}) &= \sum_{m=1}^{M} \log P(\mathbf{x}^m;\mathbf{W}) &(8)\\ &= \sum_{m=1}^{M} \log \sum_{\mathbf{h}} \frac{1}{Z}\exp(-E(\mathbf{x}^m,\mathbf{h};\mathbf{W})) &(9)\end{aligned}$$

The partial derivative of this quantity with respect to parameter $\mathbf{W}_{k,m}$ is given by [1]:

$$\frac{\partial S(\mathbf{X},\mathbf{W})}{\partial w_{k,m}} = \sum_{m=1}^{M} \Big[\langle \mathbf{y}_k\mathbf{y}_m\rangle_{P(\mathbf{h},\mathbf{x}^m;\mathbf{W})} - \langle \mathbf{y}_k\mathbf{y}_m\rangle_{P(\mathbf{h},\mathbf{x};\mathbf{W})}\Big]. \tag{10}$$

The computation of the first summand is commonly referred to as the 'awake' phase when the network 'sees' the data, while for the second summand it is referred to as the 'dream' phase when the network 'imagines' data without any observations; in loose language, we can say that learning makes the network's dreams conform with reality: we have convergence when the two expectations are identical.

In particular, the first summand in Eq. 10 above writes:

$$\langle \mathbf{y}_k\mathbf{y}_l\rangle_{P(\mathbf{h},\mathbf{x}^m;\mathbf{W})} = \sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{x}^m;\mathbf{W})\mathbf{y}_k\mathbf{y}_l, \quad \mathbf{y} = \begin{bmatrix} \mathbf{x}^m \\ \mathbf{h} \end{bmatrix}, \tag{11}$$

which means that for any m, the $\mathbf{x}$ components of all $\mathbf{y}$ vectors are 'clamped' to the observed value $\mathbf{x}^m$, while the $\mathbf{h}$ components are being summed over; this means that for any sample m we have 2^K summations, where K is the number of hidden variables.

The weights in this summation are equal to the conditional probability of $\mathbf{h}$ given that $\mathbf{x}^m$ has been observed,

[1] The proof is given in Section 19.5.2 of K. Murphy's book for the most general case of the exponential family

given by:

$$P(\mathbf{h}|\mathbf{x}^m;\mathbf{W}) = \frac{P(\mathbf{h},\mathbf{x}^m;\mathbf{W})}{P(\mathbf{x}^m;\mathbf{W})} \tag{12}$$

$$\overset{Eq.\ 7}{=} \frac{P(\mathbf{h},\mathbf{x}^m;\mathbf{W})}{\sum_{\mathbf{h}} P(\mathbf{h},\mathbf{x}^m;\mathbf{W})} \tag{13}$$

$$= \frac{\exp(-E(\mathbf{x}^m,\mathbf{h};\mathbf{W})}{\sum_{\mathbf{h}} \exp(-E(\mathbf{x}^m,\mathbf{h};\mathbf{W})}. \tag{14}$$

The second term in Eq. 10 writes:

$$\langle \mathbf{y}_k \mathbf{y}_l \rangle_{P(\mathbf{h},\mathbf{x};\mathbf{W})} = \sum_{\mathbf{h},\mathbf{x}} P(\mathbf{h},\mathbf{x};\mathbf{W}) \mathbf{y}_k \mathbf{y}_l, \quad \mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{h} \end{bmatrix}, \tag{15}$$

namely we are now using the 'unclamped' version of the distribution, where $\mathbf{x}$ is allowed to vary. This means that for any sample m we have 2^{K+N} summations, where K is the number of hidden variables and N the number of observed variables. Given that we are performing this summation with brute-force summation, the result will be the same irrespective of m - i.e. we only need to perform it once. Things change however as soon as we turn to Monte Carlo approximations, described below.

3 Fast sampling of RBMs

For larger models it is impossible to perform the brute-force summation described above, and one needs to resort to Monte Carlo approximations to estimate the quantities involved in Eq. 10.

Generic Gibbs sampling could be used, but it can be excruciatingly slow. If however we work with Restricted Boltzmann Machines (RBMs), we can use 'Block-Gibbs' sampling, which consists in the simultaneous sampling of sets ('blocks') of nodes.

3.1 Factorization property of RBMs

To see how we do this, we observe that since in an RBM we have no lateral connections between the hidden, or the observed nodes, the energy is bilinear in $\mathbf{h}, \mathbf{v}$:

$$E(\mathbf{x},\mathbf{h};\mathbf{W}) = -\mathbf{x}^T \mathbf{W}_{\mathbf{x},\mathbf{h}} \mathbf{h}, \tag{16}$$

meaning that if we know $\mathbf{x}$ our energy is a linear function of $\mathbf{h}$, and vice versa. To make this more explicit we introduce the following notation for the rows and columns of $\mathbf{W}_{\mathbf{x},\mathbf{h}}$:

$$\mathbf{W}_{\mathbf{x},\mathbf{h}} = \begin{bmatrix} \mathbf{W}_1 & \cdots & \mathbf{W}_K \end{bmatrix} = \begin{bmatrix} \mathbf{W}^1 \\ \vdots \\ \mathbf{W}^N \end{bmatrix}. \tag{17}$$

We can now write our energy function as follows:

$$E(\mathbf{x},\mathbf{h};\mathbf{W}) = -\mathbf{x}^T\mathbf{W}_{\mathbf{x},\mathbf{h}}\mathbf{h} \tag{18}$$

$$= -\sum_{k=1}^{K}\underbrace{(\mathbf{x}'^T\mathbf{W}_k)}_{\mathbf{f}_k^{\mathbf{x}}}\mathbf{h}_k = \sum_{k=1}^{K}\mathbf{h}_k\mathbf{f}_k^{\mathbf{x}} \tag{19}$$

$$= -\sum_{n=1}^{N}\mathbf{x}_n\underbrace{\mathbf{W}^n\mathbf{h}}_{c_n^{\mathbf{h}}} = \sum_{n=1}^{N}\mathbf{x}_n c_n^{\mathbf{h}} \tag{20}$$

The 'decoupling' of the contributions of $\mathbf{h}$ to the energy in Eq. 19 is also reflected in the resulting form of their conditional distribution given $\mathbf{x}$:

$$P(\mathbf{h}|\mathbf{x};\mathbf{W}) = \frac{\exp(-E(\mathbf{x},\mathbf{h};\mathbf{W}))}{\sum_{\mathbf{h}}\exp(-E(\mathbf{x},\mathbf{h};\mathbf{W}))} \tag{21}$$

In particular, for the denominator we have:

$$\sum_{\mathbf{h}}\exp(-E(\mathbf{x},\mathbf{h};\mathbf{W})) = \sum_{\mathbf{h}}\prod_{i=1}^{K}\exp(\mathbf{h}_i\mathbf{f}_i^{\mathbf{x}}) \tag{22}$$

$$= \sum_{\mathbf{h}_1=\{-1,1\}}\cdots\sum_{\mathbf{h}_K=\{-1,1\}}\prod_{i=1}^{K}\exp(\mathbf{h}_i\mathbf{f}_i^{\mathbf{x}}) \tag{23}$$

$$= \left(\sum_{\mathbf{h}_1=\{-1,1\}}\exp(\mathbf{h}_1\mathbf{f}_1^{\mathbf{x}})\right)\cdots\left(\sum_{\mathbf{h}_K=\{-1,1\}}\exp(\mathbf{h}_K\mathbf{f}_K^{\mathbf{x}})\right) \tag{24}$$

$$= \prod_i\sum_{\mathbf{h}_i=\{-1,1\}}\exp(\mathbf{h}_i\mathbf{f}_i^{\mathbf{x}}) \tag{25}$$

Inserting this expression in Eq. 21 we have:

$$P(\mathbf{h}|\mathbf{x};\mathbf{W}) = \frac{\prod_i\exp(\mathbf{h}_i\mathbf{f}_i^{\mathbf{x}})}{\prod_i\sum_{\mathbf{h}_i=\{-1,1\}}\exp(\mathbf{h}_i\mathbf{f}_i^{\mathbf{x}})} \tag{26}$$

$$= \prod_i\frac{\exp(\mathbf{h}_i\mathbf{f}_i^{\mathbf{x}})}{\sum_{\mathbf{h}_i=\{-1,1\}}\exp(\mathbf{h}_i\mathbf{f}_i^{\mathbf{x}})} \tag{27}$$

$$= \prod_i\frac{1}{1+\exp(-2\mathbf{h}_i\langle\mathbf{x},\mathbf{W}_i\rangle)}, \tag{28}$$

namely, given $\mathbf{x}$, the elements of $\mathbf{h}$ are independent -as indicated by the factorized form of their posterior distribution. Similarily, starting from Eq. 20 we can show that

$$P(\mathbf{x}|\mathbf{h};\mathbf{W}) = \prod_n\frac{1}{1+\exp(2\mathbf{x}_n\langle\mathbf{h},\mathbf{W}^n\rangle)}, \tag{29}$$

where now n runs over the N elements of $\mathbf{x}$, and $\mathbf{W}^n$ indicates the n-th row of $\mathbf{W}_{\mathbf{x},\mathbf{h}}$

3.2 Block-Gibbs Sampling

We now turn to two simplifications of learning that stem from this factorization property. The first simplification relates to the computation in Eq. 11, corresponding to the 'awake' phase, where the nodes $\mathbf{x}$ are

clamped to the observations. We note that since we have an RBM, we only consider pairs k, l of one observed (say $\mathbf{y}_k = \mathbf{x}_k$) and one hidden variable (say $\mathbf{y}_l = \mathbf{h}_{l-N} = \mathbf{h}_{l'}$). For such a pair of variables we have:

$$\begin{aligned}
\langle \mathbf{y}_k \mathbf{y}_l \rangle_{P(\mathbf{h},\mathbf{x}^m;\mathbf{W})} &= \sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{x}^m;\mathbf{W}) \mathbf{y}_k \mathbf{y}_l && (30)\\
&\overset{Eq.\ 28}{=} \sum_{\mathbf{h}_{-\{l'\}}} \sum_{\mathbf{h}'_l} P(\mathbf{h}_{-\{l'\}}|\mathbf{x}^m;\mathbf{W}) P(\mathbf{h}_{l'}|\mathbf{x}^m;\mathbf{W}) \mathbf{y}_k \mathbf{y}_l && (31)\\
&= \sum_{\mathbf{h}_{-\{l'\}}} P(\mathbf{h}_{-\{l'\}}|\mathbf{x}^m;\mathbf{W}) \sum_{\mathbf{h}'_l} P(\mathbf{h}_{l'}|\mathbf{x}^m;\mathbf{W}) \mathbf{y}_k \mathbf{y}_l && (32)\\
&= \sum_{\mathbf{h}'_l} P(\mathbf{h}_{l'}|\mathbf{x}^m;\mathbf{W}) \mathbf{y}_k \mathbf{y}_l && (33)\\
&= \mathbf{y}_k 1 P(\mathbf{h}_{l'} = 1|\mathbf{x}^m;\mathbf{W}) + \mathbf{y}_k (-1) P(\mathbf{h}_{l'} = -1|\mathbf{x}^m;\mathbf{W}) && (34)\\
&= \mathbf{y}_k P(\mathbf{h}_{l'} = 1|\mathbf{x}^m;\mathbf{W}) - \mathbf{y}_k (1 - P(\mathbf{h}_{l'} = 1|\mathbf{x}^m;\mathbf{W})) && (35)\\
&\overset{Eq.\ 28}{=} \mathbf{y}_k \left[2P(\mathbf{h}_{l'} = 1|\mathbf{x}^m;\mathbf{W}) - 1 \right]. && (36)
\end{aligned}$$

Namely, for an RBM we can compute the expectation of this quantity in $O(K)$ operations without having to perform all of the 2^K summations that would be required for a generic Boltzmann machine. As such, rather than performing a Monte Carlo approximations we can efficiently compute the exact expectation.

3.3 Contrastive Divergence [covered in Lecture 4]

The second simplification relates to the quantity in Eq. 15, corresponding to the 'dream' phase, where $\mathbf{x}$ is not fixed - but rather summed over. For this quantity we will need to resort to Monte Carlo approximations, since it requires considering all $\mathbf{x}, \mathbf{h}$ combinations. Here, RBMs make sampling faster: since the conditional distribution in Eq. 28 is factorized, we can sample *all* elements of $\mathbf{h}$ given $\mathbf{x}$ *in a single step*, rather than updating one element of $\mathbf{h}$ at a time. Similarily, based on Eq. 29 we can update all elements of $\mathbf{x}$ in a single step given $\mathbf{h}$.

This suggests the following 'Block-Gibbs' sampling scheme:

$$\mathbf{x}^0 \to \mathbf{h}^0 \to \mathbf{x}^1 \to \mathbf{h}^1 \to \cdots \to \mathbf{h}^{l-1} \to \mathbf{x}^l \to \mathbf{h}^l \tag{37}$$

which means that we start by conditioning on $\mathbf{x}^0$ and sampling $\mathbf{h}^0$ from its conditional distribution, then conditioning on $\mathbf{h}^0$ and sampling $\mathbf{x}^1$ and then repeating. In just two iterations we update all elements of our network, irrespective of the network's size; this can result in large savings when compared to plain Gibbs sampling, that updates one node per iteration.

A third acceleration trick that we can use (which does not necessarily relate to RBMs) is to use the training data $\mathbf{X}$ to guide this sampling process. Namely, even though Block-Gibbs sampling may be fast, if we initialize it from a random vector $\mathbf{x}^0$ a certain number of iterations may be required until we have burn-in, i.e. until $\mathbf{x}^l \sim P(\mathbf{x};\mathbf{W})$ - and multiple Block-Gibbs sampling updates should be skipped in order to make consecutive samples $\mathbf{x}^l, \mathbf{x}^{l+t}$ independent.

Instead, we can use each training vector $\mathbf{x}^m$ as a starting point for Block-Gibbs sampling; namely setting $\mathbf{x}^{m,0} = \mathbf{x}^m$ and then using $\mathbf{x}^{m,L}$ as a sample for a Monte Carlo approximation to Eq. 15, where L indicates

the length of the Block-Gibbs sampling scheme used. We then get:

$$\langle \mathbf{y}_k \mathbf{y}_l \rangle_{P(\mathbf{h},\mathbf{x};\mathbf{W})} = \sum_{\mathbf{h},\mathbf{x}} P(\mathbf{h},\mathbf{x};\mathbf{W}) \mathbf{y}_k \mathbf{y}_l, \quad \mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{h} \end{bmatrix}, \tag{38}$$

$$\simeq \sum_{m=1}^{M} \mathbf{y}_k^{m,L} \mathbf{y}_l^{m,L}, \quad \mathbf{y}^{m,L} = \begin{bmatrix} \mathbf{x}^{m,L} \\ \mathbf{h}^{m,L} \end{bmatrix}, \tag{39}$$

$$\simeq \sum_{m=1}^{M} \mathbf{y}_k^{m,L} \left[2P(\mathbf{h}_{l'} = 1 | \mathbf{x}^{m,L}; \mathbf{W}) - 1 \right], \tag{40}$$

where M is the size of the training set, and for Eq. 40 we have used the same kind of reasoning as to get to Eq. 36. The second Monte-Carlo approximation will be better than the first, since we are analytically dealing with the distribution over $\mathbf{h}$.

Using this scheme amounts to using the Contrastive Divergence training algorithm, detailed in Sec. 27.7.4 of K. Murphy's book, which can be understood as approximate Maximum Likelihood estimation (note that here the term 'approximate' does not relate to the Monte Carlo part, but rather the approximation of the Maximum Likelihood training objective). The quality of the approximation depends on the length L and for $L \to \infty$ we get back the original Maximum Likelihood estimation problem.

3.4 Technical note

One technicality is that while in the gradient expression for the log-likelihood we have been summing over m, e.g. in Eq. 10, in the code of your assignments we are averaging. This allows us to use the same update step for gradient descent, irrespective of the size M of the training set. This can be formally treated by saying that our objective is the *average* log-likelihood of the data, which would give us an $1/M$ factor for the gradient.