

August 21, 2017

RT0979

Mathematics 36 pages

Research Report

Boltzmann machines and energy-based models

Takayuki Osogami

IBM Research - Tokyo

IBM Japan, Ltd.

19-21 Nihonbashi Hakozaiki-cho, Chuo-ku

Tokyo 103-8510, Japan

Boltzmann machines and energy-based models

Takayuki Osogami

Abstract

We review Boltzmann machines and energy-based models. A Boltzmann machine defines a probability distribution over binary-valued patterns. One can learn parameters of a Boltzmann machine via gradient based approaches in a way that log likelihood of data is increased. The gradient and Laplacian of a Boltzmann machine admit beautiful mathematical representations, although computing them is in general intractable. This intractability motivates approximate methods, including Gibbs sampler and contrastive divergence, and tractable alternatives, namely energy-based models.

1 Introduction

The Boltzmann machine has received considerable attention particularly after the publication of the seminal paper by Hinton and Salakhutdinov on autoencoder with stacked restricted Boltzmann machines [21], which leads to today's success of and expectation to deep learning [54, 13] as well as a wide range of applications of Boltzmann machines such as collaborative filtering [1], classification of images and documents [33], and human choice [45, 47]. The Boltzmann machine is a stochastic (generative) model that can represent a probability distribution over binary patterns and others (see Section 2). The stochastic or generative capability of the Boltzmann machine has not been fully exploited in today's deep learning. For further advancement of the field, it is important to understand basics of the Boltzmann machine particularly from probabilistic perspectives. In this paper, we review fundamental properties of the Boltzmann machines with particular emphasis on probabilistic representations that allow intuitive interpretations in terms of probabilities.

A core of this paper is in the learning rules based on gradients or stochastic gradients for Boltzmann machines (Section 3-Section 4). These learning rules maximize the log likelihood of given dataset or minimize the Kullback-Leibler (KL) divergence to a target distribution. In particular, Boltzmann machines admit concise mathematical representations for its gradients and Laplacians. For example, Laplacians can be represented with covariance matrices.

The exact learning rules, however, turn out to be computationally intractable for general Boltzmann machines. We then review approximate learning methods such as Gibbs sampler and contrastive divergence in Section 5.

We also review other models that are related to the Boltzmann machine in Section 6. For example, the Markov random field is a generalization of the Boltzmann machine. We also discuss how to deal with real valued distributions by modifying the Boltzmann machine.

The intractability of exact learning of the Boltzmann machine motivates tractable energy-based learning. Some of the approximate learning methods for the Boltzmann machine may be considered as a form of energy-based learning. As a practical example, we review an energy-based model for face detection in Section 7.

This survey paper is based on a personal note prepared for the first of the four parts of a tutorial given at the 26th International Joint Conference on Artificial Intelligence (IJCAI-17) held in Melbourne, Australia on August 21, 2017. See a tutorial webpage¹ for information about the tutorial.

¹https://researcher.watson.ibm.com/researcher/view_group.php?id=7834

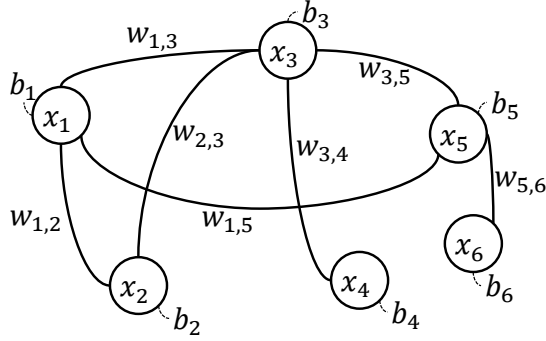


Figure 1: A Boltzmann machine.

A survey corresponding to the third part of the tutorial (Boltzmann machines for time-series) can be found in [43].

2 The Boltzmann machine

A Boltzmann machine is a network of units that are connected to each other (see Figure 1). Let N be the number of units. Each unit takes a binary value (0 or 1). Let X_i be the random variable representing the value of the i -th unit for $i \in [1, N]$. We use a column vector \mathbf{X} to denote the random values of the N units. The Boltzmann machine has two types of parameters: bias and weight. Let b_i be the bias for the i -th unit for $i \in [1, N]$, and let $w_{i,j}$ be the weight between unit i and unit j for $(i, j) \in [1, N-1] \times [i+1, N]$. We use a column vector \mathbf{b} to denote the bias for all units and a matrix \mathbf{W} to denote the weight for all pairs of units. Namely, the (i, j) -th element of \mathbf{W} is $w_{i,j}$. We let $w_{i,j} = 0$ for $i \geq j$ and for the pair of units (i, j) that are disconnected each other. The parameters are collectively denoted by

$$\theta \equiv (b_1, \dots, b_N, w_{1,2}, \dots, w_{N-1,N}), \quad (1)$$

which we also denote as $\theta = (\mathbf{b}, \mathbf{W})$.

The energy of the Boltzmann machine is defined by

$$E_\theta(\mathbf{x}) = -\sum_{i=1}^N b_i x_i - \sum_{i=1}^{N-1} \sum_{j=i+1}^N w_{i,j} x_i x_j \quad (2)$$

$$= -\mathbf{b}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{W} \mathbf{x}. \quad (3)$$

From the energy, the Boltzmann machine defines the probability distribution over binary patterns as follows:

$$\mathbb{P}_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{\sum_{\tilde{\mathbf{x}}} \exp(-E_\theta(\tilde{\mathbf{x}}))} \quad (4)$$

where the summation with respect to $\tilde{\mathbf{x}}$ is over all of the possible N bit binary values. Namely, the higher the energy of a pattern \mathbf{x} , the less likely that the \mathbf{x} is generated. For a moment, we do not address the computational aspect of the denominator, which involves a summation of 2^N terms. This denominator is also known as a partition function:

$$Z \equiv \sum_{\tilde{\mathbf{x}}} \exp(-E_\theta(\tilde{\mathbf{x}})). \quad (5)$$

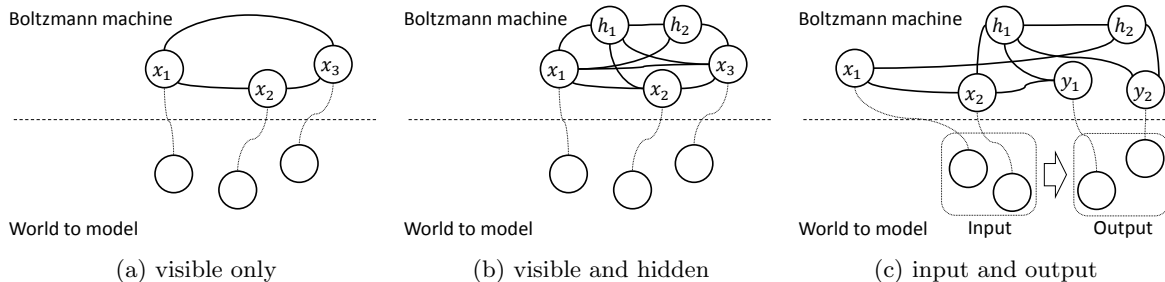


Figure 2: Boltzmann machines with hidden units, input, and output

A Boltzmann machine can be used to model the probability distribution, $\mathbb{P}_{\text{target}}(\cdot)$, of target patterns. Namely, by optimally setting the values of θ , we approximate $\mathbb{P}_{\text{target}}(\cdot)$ with $\mathbb{P}_{\theta}(\cdot)$. Here, some of the units of the Boltzmann machine are allowed to be hidden, which means that those units do not directly correspond to the target patterns (see Figure 2b). The units that directly correspond to the target patterns are called visible. The primary purpose of the hidden units is to allow particular dependency between visible units, which cannot be represented solely with visible units. The visible units may be divided into input and output (see Figure 2c). Then the Boltzmann machine can be used to model the conditional distribution of the output patterns given an input pattern.

3 Learning a generative model

Now we consider the problem of optimally setting the values of θ in a way that $\mathbb{P}_{\theta}(\cdot)$ best approximates a given $\mathbb{P}_{\text{target}}(\cdot)$. Specifically, we seek to minimize the Kullback-Leibler (KL) divergence from \mathbb{P}_{θ} to $\mathbb{P}_{\text{target}}$ [2]:

$$\text{KL}(\mathbb{P}_{\text{target}} \parallel \mathbb{P}_{\theta}) \equiv \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \log \frac{\mathbb{P}_{\text{target}}(\tilde{\mathbf{x}})}{\mathbb{P}_{\theta}(\tilde{\mathbf{x}})} \quad (6)$$

$$= \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \log \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \log \mathbb{P}_{\theta}(\tilde{\mathbf{x}}). \quad (7)$$

The first term of (7) is independent of θ . It thus suffices to maximize the negation of the second term:

$$f(\theta) \equiv \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \log \mathbb{P}_{\theta}(\tilde{\mathbf{x}}). \quad (8)$$

A special case of $\mathbb{P}_{\text{target}}$ is the empirical distribution of the patterns in a given training dataset:

$$\mathcal{D} = \{\mathbf{x}^{(d)}\}_{d=1}^D, \quad (9)$$

where D is the number of the patterns in \mathcal{D} . Then the objective function (8) becomes

$$f(\theta) = \frac{1}{D} \sum_{\mathbf{x} \in \mathcal{D}} \log \mathbb{P}_{\theta}(\mathbf{x}) \quad (10)$$

$$= \frac{1}{D} \log \prod_{\mathbf{x} \in \mathcal{D}} \mathbb{P}_{\theta}(\mathbf{x}), \quad (11)$$

which is the log-likelihood of \mathcal{D} with respect to \mathbb{P}_{θ} when multiplied by D . By defining

$$\mathbb{P}_{\theta}(\mathcal{D}) \equiv \prod_{\mathbf{x} \in \mathcal{D}} \mathbb{P}_{\theta}(\mathbf{x}), \quad (12)$$

we can represent $f(\theta)$ as follows:

$$f(\theta) = \frac{1}{D} \log \mathbb{P}_\theta(\mathcal{D}). \quad (13)$$

To find the optimal values of θ , we take the gradient of $f(\theta)$ with respect to θ :

$$\nabla f(\theta) = \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \nabla \log \mathbb{P}_\theta(\tilde{\mathbf{x}}). \quad (14)$$

3.1 All of the units are visible

We start with the simplest case where all of the units are visible (see Figure 2a). Then the energy of the Boltzmann machine is simply given by (3), and the probability distribution is given by (4).

3.1.1 Gradient

We will derive a specific representation of $\nabla \log \mathbb{P}_\theta(\mathbf{x})$ to examine the form of $\nabla f(\theta)$ in this case:

$$\nabla \log \mathbb{P}_\theta(\mathbf{x}) = \nabla \log \frac{\exp(-E_\theta(\mathbf{x}))}{\sum_{\hat{\mathbf{x}}} \exp(-E_\theta(\hat{\mathbf{x}}))} \quad (15)$$

$$= -\nabla E_\theta(\mathbf{x}) - \nabla \log \sum_{\hat{\mathbf{x}}} \exp(-E_\theta(\hat{\mathbf{x}})) \quad (16)$$

$$= -\nabla E_\theta(\mathbf{x}) + \frac{\sum_{\hat{\mathbf{x}}} \exp(-E_\theta(\hat{\mathbf{x}})) \nabla E_\theta(\hat{\mathbf{x}})}{\sum_{\hat{\mathbf{x}}} \exp(-E_\theta(\hat{\mathbf{x}}))} \quad (17)$$

$$= -\nabla E_\theta(\mathbf{x}) + \sum_{\hat{\mathbf{x}}} \mathbb{P}_\theta(\hat{\mathbf{x}}) \nabla E_\theta(\hat{\mathbf{x}}), \quad (18)$$

where the summation with respect to $\hat{\mathbf{x}}$ is over all of the possible binary patterns, similar to the summation with respect to $\tilde{\mathbf{x}}$. Here, (18) follows from (4) and (17).

Plugging the last expression into (14), we obtain

$$\nabla f(\theta) = - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \nabla E_\theta(\tilde{\mathbf{x}}) + \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \sum_{\hat{\mathbf{x}}} \mathbb{P}_\theta(\hat{\mathbf{x}}) \nabla E_\theta(\hat{\mathbf{x}}) \quad (19)$$

$$= - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \nabla E_\theta(\tilde{\mathbf{x}}) + \sum_{\tilde{\mathbf{x}}} \mathbb{P}_\theta(\tilde{\mathbf{x}}) \nabla E_\theta(\tilde{\mathbf{x}}) \quad (20)$$

$$= - \sum_{\tilde{\mathbf{x}}} (\mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) - \mathbb{P}_\theta(\tilde{\mathbf{x}})) \nabla E_\theta(\tilde{\mathbf{x}}). \quad (21)$$

The last expression allows intuitive interpretation of a gradient-based method for increasing the value of $f(\theta)$:

$$\theta \leftarrow \theta + \eta \nabla f(\theta), \quad (22)$$

where η is the learning rate (or the step size). Namely, for each pattern $\tilde{\mathbf{x}}$, we compare $\mathbb{P}_\theta(\tilde{\mathbf{x}})$ against $\mathbb{P}_{\text{target}}(\tilde{\mathbf{x}})$. If $\mathbb{P}_\theta(\tilde{\mathbf{x}})$ is greater than $\mathbb{P}_{\text{target}}(\tilde{\mathbf{x}})$, we update θ in a way that it increases the energy $E_\theta(\tilde{\mathbf{x}})$ so that the $\tilde{\mathbf{x}}$ becomes less likely to be generated with \mathbb{P}_θ . If $\mathbb{P}_\theta(\tilde{\mathbf{x}})$ is smaller than $\mathbb{P}_{\text{target}}(\tilde{\mathbf{x}})$, we update θ in a way that $E_\theta(\tilde{\mathbf{x}})$ decreases.

We will also write (20) as follows:

$$\nabla f(\theta) = -\mathbb{E}_{\text{target}}[\nabla E_\theta(\mathbf{X})] + \mathbb{E}_\theta[\nabla E_\theta(\mathbf{X})], \quad (23)$$

where $\mathbb{E}_{\text{target}}[\cdot]$ is the expectation with respect to $\mathbb{P}_{\text{target}}$, $\mathbb{E}_{\theta}[\cdot]$ is the expectation with respect to \mathbb{P}_{θ} , and \mathbf{X} is the vector of random variables denoting the values of the N units. Note that the expression of the gradient in (23) holds for any form of energy, as long as the energy is used to define the probability as in (4).

Now we take into account the specific form of the energy given by (3). Taking the derivative with respect to each parameter, we obtain

$$\frac{\partial}{\partial b_i} E_{\theta}(\mathbf{x}) = -x_i \quad (24)$$

$$\frac{\partial}{\partial w_{i,j}} E_{\theta}(\mathbf{x}) = -x_i x_j \quad (25)$$

for $i \in [1, N]$ and $(i, j) \in [1, N - 1] \times [i + 1, N]$. From (23), we then find

$$\frac{\partial}{\partial b_i} f(\theta) = \mathbb{E}_{\text{target}}[X_i] - \mathbb{E}_{\theta}[X_i] \quad (26)$$

$$\frac{\partial}{\partial w_{i,j}} f(\theta) = \mathbb{E}_{\text{target}}[X_i X_j] - \mathbb{E}_{\theta}[X_i X_j], \quad (27)$$

where X_i is the random variable denoting the value of the i -th unit for each $i \in [1, N]$. Notice that the expected value of X_i is the same as the probability of $X_i = 1$, because X_i is binary. In general, exact evaluation of $\mathbb{E}_{\theta}[X_i]$ or $\mathbb{E}_{\theta}[X_i X_j]$ is computationally intractable, but we will not be concerned with this computational aspect until Section 5.

A gradient ascent method is thus to iteratively update the parameters as follows:

$$b_i \leftarrow b_i + \eta (\mathbb{E}_{\text{target}}[X_i] - \mathbb{E}_{\theta}[X_i]) \quad (28)$$

$$w_{i,j} \leftarrow w_{i,j} + \eta (\mathbb{E}_{\text{target}}[X_i X_j] - \mathbb{E}_{\theta}[X_i X_j]) \quad (29)$$

for $i \in [1, N]$ and $(i, j) \in [1, N - 1] \times [i + 1, N]$. Intuitively, b_i controls how likely that the i -th unit takes the value 1, and $w_{i,j}$ controls how likely that the i -th unit and the j -th unit simultaneously take the value 1. For example, when $\mathbb{E}_{\theta}[X_i X_j]$ is smaller than $\mathbb{E}_{\text{target}}[X_i X_j]$, we increase $w_{i,j}$ to increase $\mathbb{E}_{\theta}[X_i X_j]$. This form of learning rule appears frequently in the context of Boltzmann machines. Namely, we compare our prediction $\mathbb{E}_{\theta}[\cdot]$ against the target $\mathbb{E}_{\text{target}}[\cdot]$ and update θ in a way that $\mathbb{E}_{\theta}[\cdot]$ gets closer to $\mathbb{E}_{\text{target}}[\cdot]$.

3.1.2 Stochastic gradient

We now rewrite (20) as follows:

$$\nabla f(\theta) = \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) (-\nabla E_{\theta}(\tilde{\mathbf{x}}) + \mathbb{E}_{\theta} [\nabla E_{\theta}(\mathbf{X})]). \quad (30)$$

Namely, $\nabla f(\theta)$ is given by the expected value of $-\nabla E_{\theta}(\mathbf{X}) + \mathbb{E}_{\theta} [\nabla E_{\theta}(\mathbf{X})]$, where the first \mathbf{X} is distributed with respect to $\mathbb{P}_{\text{target}}$. Recall that the second term is an expectation with respect to \mathbb{P}_{θ} . This suggests stochastic gradient methods [3, 28, 9, 61, 50]. At each step, we sample a pattern $\mathbf{X}(\omega)$ according to $\mathbb{P}_{\text{target}}$ and update θ according to the stochastic gradient:

$$\theta \leftarrow \theta + \eta g_{\theta}(\omega), \quad (31)$$

where

$$g_{\theta}(\omega) \equiv -\nabla E_{\theta}(\mathbf{X}(\omega)) + \mathbb{E}_{\theta} [\nabla E_{\theta}(\mathbf{X})]. \quad (32)$$

When the target distribution is the empirical distribution given by the training data \mathcal{D} , we only need to take a sample $\mathbf{X}(\omega)$ from \mathcal{D} uniformly at random.

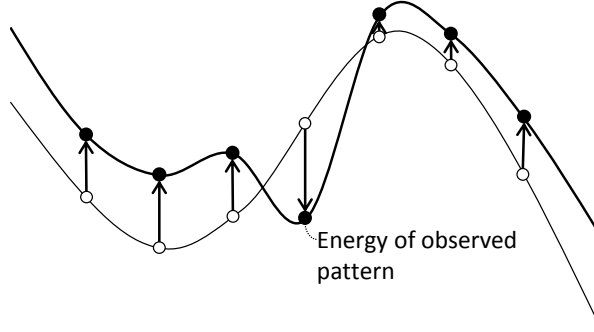


Figure 3: How the energy is updated

The stochastic gradient method based on (31)-(32) allows intuitive interpretation. At each step, we sample a pattern according to the target distribution (or from the training data) and update θ in a way that the energy of the sampled pattern is reduced. At the same time, the energy of every pattern is increased, where the amount of the increase is proportional to the probability for the Boltzmann machine with the latest parameter θ to generate that pattern (see Figure 3).

Taking into account the specific form of the energy given by (3), we can derive the specific form of the stochastic gradient:

$$\frac{\partial}{\partial b_i} E_{\theta}(\mathbf{x})(\omega) = X_i(\omega) - \mathbb{E}_{\theta}[X_i] \quad (33)$$

$$\frac{\partial}{\partial w_{i,j}} E_{\theta}(\mathbf{x})(\omega) = X_i(\omega) X_j(\omega) - \mathbb{E}_{\theta}[X_i X_j], \quad (34)$$

which suggests a stochastic gradient method of iteratively sampling a pattern $\mathbf{X}(\omega)$ according to the target probability distribution and updating the parameters as follows:

$$b_i \leftarrow b_i + \eta (X_i(\omega) - \mathbb{E}_{\theta}[X_i]) \quad (35)$$

$$w_{i,j} \leftarrow w_{i,j} + \eta (X_i(\omega) X_j(\omega) - \mathbb{E}_{\theta}[X_i X_j]) \quad (36)$$

for $i \in [1, N]$ and $(i, j) \in [1, N - 1] \times [i + 1, N]$.

3.1.3 Giving theoretical foundation for Hebb's rule

The learning rule of (36) has a paramount importance of providing a theoretical foundation for Hebb's rule of learning in biological neural networks [16]:

When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

In short, "neurons wire together if they fire together" [37].

A unit of a Boltzmann machine corresponds to a neuron, and $X_i = 1$ means that the i -th neuron fires. When two neurons, i and j , fire ($X_i(\omega) X_j(\omega) = 1$), the wiring weight $w_{i,j}$ between the two neurons gets stronger according to (36). Here, notice that we have $0 < \mathbb{E}_{\theta}[X_i X_j] < 1$ as long as the values of θ are finite.

The learning rule of the Boltzmann machine also involves a mechanism beyond what is suggested by Hebb's rule. Namely, the amount of the change in $w_{i,j}$ when the two neurons (i and j) fire depends on how likely those two neurons fire according to \mathbb{P}_{θ} at that time. More specifically, it the two neurons

are already expected to fire together (*i.e.*, $\mathbb{E}_\theta[X_i X_j] \approx 1$), we increase $w_{i,j}$ only by a small amount (*i.e.*, $\eta (1 - \mathbb{E}_\theta[X_i X_j])$) even if the two neurons fire together (*i.e.*, $X_i(\omega) X_j(\omega) = 1$).

Without this additional term ($-\mathbb{E}_\theta[X_i]$ or $-\mathbb{E}_\theta[X_i X_j]$) in (36), all of the parameters monotonically increases. If $X_i = 1$ with nonzero probability in $\mathbb{P}_{\text{target}}$, then b_i diverges to ∞ almost surely. Otherwise, b_i stays unchanged from the initial value. Likewise, if $X_i X_j = 1$ with nonzero probability in $\mathbb{P}_{\text{target}}$, then $w_{i,j}$ diverges to ∞ almost surely. Otherwise, $w_{i,j}$ stays unchanged.

What is important is that this additional term is formally derived instead of being introduced in an ad hoc manner. Specifically, the learning rule is derived from a stochastic model (*i.e.*, a Boltzmann machine) and an objective function (*i.e.*, minimizing the KL divergence to the target distribution or maximizing the log-likelihood of training data) by taking the gradient with respect to the parameters.

3.1.4 Laplacian

We now derive the Laplacian of $f(\theta)$ to examine its landscape. Starting from the expression in (27), we obtain

$$\frac{\partial}{\partial w_{k,\ell}} \frac{\partial}{\partial w_{i,j}} f(\theta) = -\frac{\partial}{\partial w_{k,\ell}} \mathbb{E}_\theta[X_i X_j] \quad (37)$$

$$= -\sum_{\tilde{\mathbf{x}}} \tilde{x}_i \tilde{x}_j \frac{\partial}{\partial w_{k,\ell}} \mathbb{P}_\theta(\tilde{\mathbf{x}}) \quad (38)$$

$$= -\sum_{\tilde{\mathbf{x}}} \tilde{x}_i \tilde{x}_j \mathbb{P}_\theta(\tilde{\mathbf{x}}) \frac{\partial}{\partial w_{k,\ell}} \log \mathbb{P}_\theta(\tilde{\mathbf{x}}) \quad (39)$$

$$= \left(\sum_{\tilde{\mathbf{x}}} \mathbb{P}_\theta(\tilde{\mathbf{x}}) \tilde{x}_i \tilde{x}_j \right) \left(\sum_{\tilde{\mathbf{x}}} \mathbb{P}_\theta(\tilde{\mathbf{x}}) \tilde{x}_k \tilde{x}_\ell \right) - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_\theta(\tilde{\mathbf{x}}) \tilde{x}_i \tilde{x}_j \tilde{x}_k \tilde{x}_\ell, \quad (40)$$

where the last expression is obtained from (18) and (25). The last expression consists of expectations with respect to \mathbb{P}_θ and can be represented conveniently as follows:

$$\frac{\partial}{\partial w_{k,\ell}} \frac{\partial}{\partial w_{i,j}} f(\theta) = \mathbb{E}_\theta[X_i X_j] \mathbb{E}_\theta[X_k X_\ell] - \mathbb{E}_\theta[X_i X_j X_k X_\ell] \quad (41)$$

$$= -\mathbb{C}\mathbb{O}\mathbb{V}_\theta[X_i X_j, X_k X_\ell], \quad (42)$$

where $\mathbb{C}\mathbb{O}\mathbb{V}_\theta[A, B]$ denotes the covariance between random variables A and B with respect to \mathbb{P}_θ . Likewise, we have

$$\frac{\partial}{\partial b_k} \frac{\partial}{\partial w_{i,j}} f(\theta) = -\mathbb{C}\mathbb{O}\mathbb{V}_\theta[X_i X_j, X_k] \quad (43)$$

$$\frac{\partial}{\partial b_j} \frac{\partial}{\partial b_i} f(\theta) = -\mathbb{C}\mathbb{O}\mathbb{V}_\theta[X_i, X_j]. \quad (44)$$

Therefore, the Laplacian of $f(\theta)$ is a covariance matrix:

$$\nabla^2 f(\theta) = -\mathbb{C}\mathbb{O}\mathbb{V}_\theta[X_1, \dots, X_N, X_1 X_2, \dots, X_{N-1} X_N], \quad (45)$$

where we use $\mathbb{C}\mathbb{O}\mathbb{V}_\theta$ to denote a covariance matrix with respect to \mathbb{P}_θ . When θ is finite, this covariance matrix is positive semidefinite, and $f(\theta)$ is concave. This justifies (stochastic) gradient based approaches to optimizing θ . This concavity has been known [19], but I am not aware of the literature that explicitly represent the Laplacian with a covariance matrix.

3.1.5 Summary

Consider a Boltzmann machine with parameters $\theta = (\mathbf{b}, \mathbf{W})$. When all of the N units of the Boltzmann machine are visible, the Boltzmann machine defines a probability distribution \mathbb{P}_θ of N -bit binary

patterns by

$$\mathbb{P}_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{\sum_{\tilde{\mathbf{x}}} \exp(-E_\theta(\tilde{\mathbf{x}}))}, \quad (46)$$

where the energy is

$$E_\theta(\mathbf{x}) \equiv -\mathbf{b}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{W} \mathbf{x}. \quad (47)$$

The KL divergence from \mathbb{P}_θ to $\mathbb{P}_{\text{target}}$ can be minimized (or the log-likelihood of the target data having the empirical distribution $\mathbb{P}_{\text{target}}$ can be maximized) by maximizing

$$f(\theta) \equiv \mathbb{E}_{\text{target}}[\log \mathbb{P}_\theta(\mathbf{X})]. \quad (48)$$

The gradient and the Laplacian of $f(\theta)$ is given by

$$\nabla f(\theta) = \mathbb{E}_{\text{target}}[\mathbf{S}] - \mathbb{E}_\theta[\mathbf{S}] \quad (49)$$

$$\nabla^2 f(\theta) = -\text{COV}_\theta(\mathbf{S}), \quad (50)$$

where \mathbf{S} denotes the vector of the random variables representing the value of a unit or the product of the values of a pair of units:

$$\mathbf{S} = (X_1, \dots, X_N, X_1 X_2, \dots, X_{N-1} X_N). \quad (51)$$

3.2 Some of the units are hidden

In this section, we consider Boltzmann machines that have both visible units and hidden units. Let N be the number of visible units and M be the number of hidden units.

3.2.1 Necessity of hidden units

We first study the necessity of hidden units [2]. The Boltzmann machine with N units have

$$N + \frac{1}{2}N(N-1) = \frac{1}{2}N(N+1) \quad (52)$$

parameters. This Boltzmann machine is used to model N -bit binary patterns. There are 2^N possible N -bit binary patterns, and the general distribution of N -bit patterns assigns probabilities to those 2^N patterns. We need

$$2^N - 1 \quad (53)$$

parameters to characterize this general distribution.

The number of parameters of the Boltzmann machine is smaller than the number of parameters needed to characterize the general distribution as long as $N > 2$. This suggests that the probability distribution that can be represented by the Boltzmann machine is limited. One way to extend the flexibility of the Boltzmann machine is the use of hidden units.

3.2.2 Free energy

Let \mathbf{x} denote the visible values (*i.e.*, the values of visible units), \mathbf{h} denote the hidden values, and (\mathbf{x}, \mathbf{h}) denote the values of all units. We write the marginal probability distribution of the visible values as follows:

$$\mathbb{P}_\theta(\mathbf{x}) = \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\mathbf{x}, \tilde{\mathbf{h}}), \quad (54)$$

where the summation is over all of the possible binary patterns of the hidden values, and

$$\mathbb{P}_\theta(\mathbf{x}, \mathbf{h}) = \frac{\exp(-E_\theta(\mathbf{x}, \mathbf{h}))}{\sum_{\tilde{\mathbf{x}}, \tilde{\mathbf{h}}} \exp(-E_\theta(\tilde{\mathbf{x}}, \tilde{\mathbf{h}}))}. \quad (55)$$

Here, we write energy as follows:

$$E_\theta(\mathbf{x}, \mathbf{h}) = -\mathbf{b}^\top \begin{pmatrix} \mathbf{x} \\ \mathbf{h} \end{pmatrix} - (\mathbf{x}^\top, \mathbf{h}^\top) \mathbf{W} \begin{pmatrix} \mathbf{x} \\ \mathbf{h} \end{pmatrix} \quad (56)$$

$$= -(\mathbf{b}^V)^\top \mathbf{x} - (\mathbf{b}^H)^\top \mathbf{h} - \mathbf{x}^\top \mathbf{W}^{VV} \mathbf{x} - \mathbf{x}^\top \mathbf{W}^{VH} \mathbf{h} - \mathbf{h}^\top \mathbf{W}^{HH} \mathbf{h}. \quad (57)$$

Now, we define free energy as follows:

$$F_\theta(\mathbf{x}) \equiv -\log \sum_{\tilde{\mathbf{h}}} \exp(-E_\theta(\mathbf{x}, \tilde{\mathbf{h}})). \quad (58)$$

We can then represent $\mathbb{P}_\theta(\mathbf{x})$ in a way similar to the case where all of the units are visible, replacing energy with free energy:

$$\mathbb{P}_\theta(\mathbf{x}) = \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\mathbf{x}, \tilde{\mathbf{h}}) \quad (59)$$

$$= \frac{\sum_{\tilde{\mathbf{h}}} \exp(-E_\theta(\mathbf{x}, \tilde{\mathbf{h}}))}{\sum_{\tilde{\mathbf{x}}, \tilde{\mathbf{h}}} \exp(-E_\theta(\tilde{\mathbf{x}}, \tilde{\mathbf{h}}))} \quad (60)$$

$$= \frac{\exp(-F_\theta(\mathbf{x}))}{\sum_{\tilde{\mathbf{x}}} \exp(-F_\theta(\tilde{\mathbf{x}}))}. \quad (61)$$

3.2.3 Gradient

In (20), we simply replace energy with free energy to obtain the gradient of our objective function when some of the units are hidden:

$$\nabla f(\theta) = -\sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \nabla F_\theta(\tilde{\mathbf{x}}) + \sum_{\tilde{\mathbf{x}}} \mathbb{P}_\theta(\tilde{\mathbf{x}}) \nabla F_\theta(\tilde{\mathbf{x}}) \quad (62)$$

$$= -\mathbb{E}_{\text{target}}[\nabla F_\theta(\mathbf{X})] + \mathbb{E}_\theta[\nabla F_\theta(\mathbf{X})] \quad (63)$$

What we then need is the gradient of free energy:

$$\nabla F_\theta(\mathbf{x}) = -\nabla \log \sum_{\tilde{\mathbf{h}}} \exp(-E_\theta(\mathbf{x}, \tilde{\mathbf{h}})) \quad (64)$$

$$= \frac{\sum_{\tilde{\mathbf{h}}} \exp(-E_\theta(\mathbf{x}, \tilde{\mathbf{h}})) \nabla E_\theta(\mathbf{x}, \tilde{\mathbf{h}})}{\sum_{\tilde{\mathbf{h}}} \exp(-E_\theta(\mathbf{x}, \tilde{\mathbf{h}}))} \quad (65)$$

$$= \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\tilde{\mathbf{h}} | \mathbf{x}) \nabla E_\theta(\mathbf{x}, \tilde{\mathbf{h}}), \quad (66)$$

where $\mathbb{P}_\theta(\mathbf{h} | \mathbf{x})$ is the conditional probability that the hidden values are \mathbf{h} given that the visible values are \mathbf{x} :

$$\mathbb{P}_\theta(\mathbf{h} | \mathbf{x}) \equiv \frac{\exp(-E_\theta(\mathbf{x}, \mathbf{h}))}{\sum_{\tilde{\mathbf{h}}} \exp(-E_\theta(\mathbf{x}, \tilde{\mathbf{h}}))} \quad (67)$$

$$= \frac{\exp(-E_\theta(\mathbf{x}, \mathbf{h}))}{\sum_{\tilde{\mathbf{x}}, \tilde{\mathbf{h}}} \exp(-E_\theta(\mathbf{x}, \tilde{\mathbf{h}}))} \frac{\sum_{\tilde{\mathbf{h}}} \exp(-E_\theta(\mathbf{x}, \tilde{\mathbf{h}}))}{\sum_{\tilde{\mathbf{h}}} \exp(-E_\theta(\mathbf{x}, \tilde{\mathbf{h}}))} \quad (68)$$

$$= \frac{\mathbb{P}_\theta(\mathbf{x}, \mathbf{h})}{\sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\mathbf{x}, \tilde{\mathbf{h}})} \quad (69)$$

$$= \frac{\mathbb{P}_\theta(\mathbf{x}, \mathbf{h})}{\mathbb{P}_\theta(\mathbf{x})}. \quad (70)$$

Observe in (66) that the gradient of free energy is expected gradient of energy, where the expectation is with respect to the conditional distribution of hidden values given the visible values.

We thus obtain

$$\nabla f(\theta) = - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\tilde{\mathbf{h}} | \tilde{\mathbf{x}}) \nabla E_\theta(\tilde{\mathbf{x}}, \tilde{\mathbf{h}}) + \sum_{\tilde{\mathbf{x}}} \mathbb{P}_\theta(\tilde{\mathbf{x}}) \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\tilde{\mathbf{h}} | \tilde{\mathbf{x}}) \nabla E_\theta(\tilde{\mathbf{x}}, \tilde{\mathbf{h}}) \quad (71)$$

$$= - \sum_{\tilde{\mathbf{x}}, \tilde{\mathbf{h}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \mathbb{P}_\theta(\tilde{\mathbf{h}} | \tilde{\mathbf{x}}) \nabla E_\theta(\tilde{\mathbf{x}}, \tilde{\mathbf{h}}) + \sum_{\tilde{\mathbf{x}}, \tilde{\mathbf{h}}} \mathbb{P}_\theta(\tilde{\mathbf{x}}, \tilde{\mathbf{h}}) \nabla E_\theta(\tilde{\mathbf{x}}, \tilde{\mathbf{h}}). \quad (72)$$

The first term in the last expression (except the minus sign) is the expected value of the gradient of energy, where the expectation is with respect to the distribution defined with $\mathbb{P}_{\text{target}}$ and \mathbb{P}_θ . Specifically, the visible values follow $\mathbb{P}_{\text{target}}$, and given the visible values, \mathbf{x} , the hidden values follow $\mathbb{P}_\theta(\cdot | \mathbf{x})$. We will write this expectation with $\mathbb{E}_{\text{target}}[\mathbb{E}_\theta[\cdot | \mathbf{X}]]$. The second term is expectation with respect to \mathbb{P}_θ , which we denote with $\mathbb{E}_\theta[\cdot]$. Because the energy (56) has the form equivalent to (47), $\nabla f(\theta)$ can then be represented analogously to (49):

$$\nabla f(\theta) = -\mathbb{E}_{\text{target}}[\mathbb{E}_\theta[\nabla E_\theta(\mathbf{X}, \mathbf{H}) | \mathbf{X}]] + \mathbb{E}_\theta[\nabla E_\theta(\mathbf{X}, \mathbf{H})] \quad (73)$$

$$= \mathbb{E}_{\text{target}}[\mathbb{E}_\theta[\mathbf{S} | \mathbf{X}]] - \mathbb{E}_\theta[\mathbf{S}], \quad (74)$$

where \mathbf{X} is the vector of random values of the visible units, \mathbf{H} is the vector of H_i for $i \in [1, N]$, and \mathbf{S} is defined analogously to (51) for all of the (visible or hidden) units:

$$\mathbf{S} = (U_1, \dots, U_{N+M}, U_1 U_2, \dots, U_{N+M-1} U_{N+M}), \quad (75)$$

where $U_i \equiv X_i$ for $i \in [1, N]$, and $U_{N+i} \equiv H_i$ for $i \in [1, M]$, where H_i is the random variable denoting the i -th hidden value.

3.2.4 Stochastic gradient

The expression with (72) suggests stochastic gradient analogous to (31)-(32). Observe that $\nabla f(\theta)$ can be represented as

$$\nabla f(\theta) = \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) (\mathbb{E}_\theta[\nabla E_\theta(\mathbf{X}, \mathbf{H})] - \mathbb{E}_\theta[\nabla E_\theta(\tilde{\mathbf{x}}, \mathbf{H}) | \tilde{\mathbf{x}}]), \quad (76)$$

where $\mathbb{E}_\theta [\nabla E_\theta(\mathbf{X}, \mathbf{H})]$ is the expected value of the gradient of the energy when both visible values and hidden values follow \mathbb{P}_θ , and $\mathbb{E}_\theta [\nabla E_\theta(\tilde{\mathbf{x}}, \mathbf{H}) | \tilde{\mathbf{x}}]$ is the corresponding conditional expectation when the hidden values follow $\mathbb{P}_\theta(\cdot | \tilde{\mathbf{x}})$ given the visible values $\tilde{\mathbf{x}}$.

A stochastic gradient method is then to sample visible values, $\mathbf{X}(\omega)$, according to $\mathbb{P}_{\text{target}}$ and update θ according to the stochastic gradient:

$$g_\theta(\omega) = \mathbb{E}_\theta [\nabla E_\theta(\mathbf{X}, \mathbf{H})] - \mathbb{E}_\theta [\nabla E_\theta(\mathbf{X}(\omega), \mathbf{H}) | \mathbf{X}(\omega)]. \quad (77)$$

By taking into account the specific form of the energy, we find the following specific update rule:

$$b_i \leftarrow b_i + \eta (\mathbb{E}_\theta[U_i | \mathbf{X}(\omega)] - \mathbb{E}_\theta[U_i]) \quad (78)$$

$$w_{i,j} \leftarrow w_{i,j} + \eta (\mathbb{E}_\theta[U_i U_j | \mathbf{X}(\omega)] - \mathbb{E}_\theta[U_i U_j]), \quad (79)$$

where each unit (i or j) may be either visible or hidden. Specifically, let M be the number of hidden units and N be the number of visible units. Then $(i, j) \in [1, N + M - 1] \times [i + 1, N + M]$. Here, U_i denotes the value of the i -th unit, which may be visible or hidden. When the i -th unit is visible, its expected value is simply $\mathbb{E}_\theta[U_i | \mathbf{X}(\omega)] = X_i(\omega)$, and $\mathbb{E}_\theta[U_i U_j | \mathbf{X}(\omega)] = X_i(\omega) \mathbb{E}_\theta[U_j | \mathbf{X}(\omega)]$. When both i and j are visible, we have $\mathbb{E}_\theta[U_i U_j | \mathbf{X}(\omega)] = X_i(\omega) X_j(\omega)$.

Namely, we have

$$b_i \leftarrow b_i + \eta (X_i(\omega) - \mathbb{E}_\theta[X_i]) \quad (80)$$

for a visible unit $i \in [1, N]$,

$$b_i \leftarrow b_i + \eta (\mathbb{E}_\theta[H_i | \mathbf{X}(\omega)] - \mathbb{E}_\theta[H_i]) \quad (81)$$

for a hidden unit $i \in [N + 1, N + M]$,

$$w_{i,j} \leftarrow w_{i,j} + \eta (X_i(\omega) X_j(\omega) - \mathbb{E}_\theta[X_i X_j]) \quad (82)$$

for a pair of visible units $(i, j) \in [1, N - 1] \times [i + 1, N]$,

$$w_{i,j} \leftarrow w_{i,j} + \eta (X_i(\omega) \mathbb{E}_\theta[H_j | \mathbf{X}(\omega)] - \mathbb{E}_\theta[X_i X_j]) \quad (83)$$

for a pair of a visible unit and a hidden unit $(i, j) \in [1, N] \times [N + 1, N + M]$, and

$$w_{i,j} \leftarrow w_{i,j} + \eta (\mathbb{E}_\theta[H_i H_j | \mathbf{X}(\omega)] - \mathbb{E}_\theta[H_i H_j]) \quad (84)$$

for a pair of hidden units $(i, j) \in [N + 1, N + M - 1] \times [i + 1, N + M]$.

3.2.5 Laplacian

We now derive the Laplacian of $f(\theta)$ when some of the units are hidden. From the gradient of $f(\theta)$ in (72), we can write the partial derivatives as follows:

$$\frac{\partial}{\partial w_{i,j}} f(\theta) = \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\tilde{\mathbf{h}} | \tilde{\mathbf{x}}) \tilde{u}_i \tilde{u}_j - \sum_{\tilde{\mathbf{u}}} \mathbb{P}_\theta(\tilde{\mathbf{u}}) \tilde{u}_i \tilde{u}_j \quad (85)$$

$$\frac{\partial}{\partial w_{k,\ell}} \frac{\partial}{\partial w_{i,j}} f(\theta) = \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \sum_{\tilde{\mathbf{h}}} \frac{\partial}{\partial w_{k,\ell}} \mathbb{P}_\theta(\tilde{\mathbf{h}} | \tilde{\mathbf{x}}) \tilde{u}_i \tilde{u}_j - \sum_{\tilde{\mathbf{u}}} \frac{\partial}{\partial w_{k,\ell}} \mathbb{P}_\theta(\tilde{\mathbf{u}}) \tilde{u}_i \tilde{u}_j, \quad (86)$$

where recall that $u_i \equiv x_i$ for $i \in [1, N]$ and $u_{N+i} \equiv h_i$ for $i \in [1, M]$.

When all of the units are visible, the first term in (27) is the expectation with respect to the target distribution and is independent of θ . Now that some of the units are hidden, the corresponding first

term in (85) depends on θ . Here, the first term is expectation, where the visible units follow the target distribution, and the hidden units follow the conditional distribution with respect to the Boltzmann machine with parameters θ given the visible values.

Notice that the second term in the right-hand side of (86) has the form equivalent to (38). Hence, we have

$$\sum_{\tilde{\mathbf{u}}} \frac{\partial}{\partial w_{k,\ell}} \mathbb{P}_\theta(\tilde{\mathbf{u}}) \tilde{u}_i \tilde{u}_j = \mathbb{C}\text{OV}_\theta[U_i U_j, U_k U_\ell]. \quad (87)$$

The first term in the right-hand side of (86) has a form similar to (38) with two differences. The first difference is that it is an expectation with respect to $\mathbb{P}_{\text{target}}$. The second difference is that the probability \mathbb{P}_θ is conditional probability given the visible values. We now show that there exists a Boltzmann distribution that has $\mathbb{P}_\theta(\cdot | \mathbf{x})$ as its probability distribution (see also Figure 4).

Lemma 1. *Consider a Boltzmann machine having visible units and hidden units whose energy is given by (57). The conditional probability distribution of the hidden units given the visible values \mathbf{x} is given by the probability distribution of a Boltzmann machine with bias $\mathbf{b}(\mathbf{x})$ and weight $\mathbf{W}(\mathbf{x})$ that are defined as follows:*

$$\mathbf{b}(\mathbf{x}) \equiv \mathbf{b}^H + (\mathbf{W}^{VH})^\top \mathbf{x} \quad (88)$$

$$\mathbf{W}(\mathbf{x}) \equiv \mathbf{W}^{HH}. \quad (89)$$

Proof. Recall the expression of the conditional probability distribution in (67):

$$\mathbb{P}_\theta(\mathbf{h} | \mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}, \mathbf{h}))}{\sum_{\tilde{\mathbf{h}}} \exp(-E_\theta(\mathbf{x}, \tilde{\mathbf{h}}))}, \quad (90)$$

where we rewrite the energy in (57) as follows:

$$E_\theta(\mathbf{x}, \mathbf{h}) = -((\mathbf{b}^V)^\top \mathbf{x} + \mathbf{x}^\top \mathbf{W}^{VV} \mathbf{x}) - ((\mathbf{b}^H)^\top + \mathbf{x}^\top \mathbf{W}^{VH}) \mathbf{h} - \mathbf{h}^\top \mathbf{W}^{HH} \mathbf{h}. \quad (91)$$

The first term in the right-hand side of the last expression is independent of \mathbf{h} and canceled out between the numerator and the denominator in (90). We thus consider the Boltzmann machine with parameters $\theta(\mathbf{x}) \equiv (\mathbf{b}(\mathbf{x}), \mathbf{W}(\mathbf{x}))$ as defined in (88)-(89). Then we have

$$\mathbb{P}_\theta(\mathbf{h} | \mathbf{x}) = \frac{\exp(-E_{\theta(\mathbf{x})}(\mathbf{h}))}{\sum_{\tilde{\mathbf{h}}} \exp(-E_{\theta(\mathbf{x})}(\tilde{\mathbf{h}}))} \quad (92)$$

$$= \mathbb{P}_{\theta(\mathbf{x})}(\mathbf{h}), \quad (93)$$

which completes the proof of the lemma. \square

By (86) and Lemma 1, we can represent the second partial derivative as follows:

$$\frac{\partial}{\partial w_{k,\ell}} \frac{\partial}{\partial w_{i,j}} f(\theta) = \mathbb{E}_{\text{target}} [\mathbb{C}\text{OV}_\theta[U_i U_j, U_k U_\ell | \mathbf{X}]] - \mathbb{C}\text{OV}_\theta[U_i U_j, U_k U_\ell], \quad (94)$$

where $\mathbb{C}\text{OV}_\theta[\cdot, \cdot | \mathbf{X}]$ denotes the conditional covariance with respect to $\mathbb{P}_{\theta(\mathbf{X})}(\cdot) \equiv \mathbb{P}_\theta(\cdot | \mathbf{X})$ given the visible values \mathbf{X} .

Therefore, the Laplacian of $f(\theta)$ is

$$\nabla^2 f(\theta) = \mathbb{E}_{\text{target}} [\mathbb{C}\text{OV}_\theta[\mathbf{S} | \mathbf{X}]] - \mathbb{C}\text{OV}_\theta[\mathbf{S}], \quad (95)$$

where we define \mathbf{S} as in (75), and $\mathbb{C}\text{OV}_\theta[\cdot, \cdot | \mathbf{X}]$ denotes the conditional covariance matrix with respect to $\mathbb{P}_{\theta(\mathbf{X})}$ given the visible values \mathbf{X} . In general, the Laplacian is not positive semidefinite, and $f(\theta)$ is not concave. Hence, (stochastic) gradient based approaches do not necessarily find globally optimal parameters.

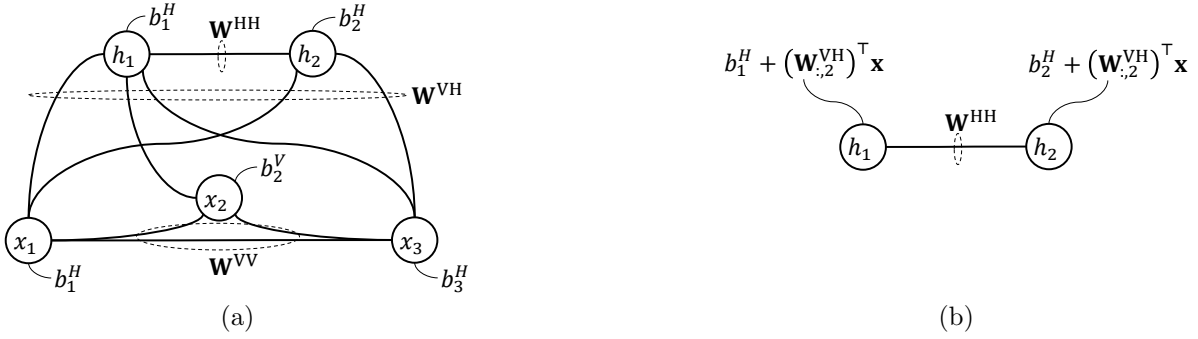


Figure 4: The conditional distribution of \mathbf{h} given \mathbf{x} in (a) is equal to the distribution of \mathbf{h} in (b), illustrating Lemma 1.

3.2.6 Summary

Consider a Boltzmann machine with parameters $\theta \equiv (\mathbf{b}, \mathbf{W})$, where at least one of the units are hidden. The Boltzmann machine defines a probability distribution \mathbb{P}_θ of the visible values \mathbf{x} and the hidden values \mathbf{h} by

$$\mathbb{P}_\theta(\mathbf{x}, \mathbf{h}) = \frac{\exp(-E_\theta(\mathbf{x}, \mathbf{h}))}{\sum_{\tilde{\mathbf{x}}, \tilde{\mathbf{h}}} \exp(-E_\theta(\tilde{\mathbf{x}}, \tilde{\mathbf{h}}))}, \quad (96)$$

where energy is given by

$$E_\theta(\mathbf{x}, \mathbf{h}) = -\mathbf{b}^\top \begin{pmatrix} \mathbf{x} \\ \mathbf{h} \end{pmatrix} - (\mathbf{x}^\top, \mathbf{h}^\top) \mathbf{W} \begin{pmatrix} \mathbf{x} \\ \mathbf{h} \end{pmatrix}. \quad (97)$$

The marginal probability distribution of the visible values is

$$\mathbb{P}_\theta(\mathbf{x}) = \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\mathbf{x}, \tilde{\mathbf{h}}) \quad (98)$$

$$= \frac{\exp(-F_\theta(\mathbf{x}))}{\sum_{\tilde{\mathbf{x}}} \exp(-F_\theta(\tilde{\mathbf{x}}))}, \quad (99)$$

where free energy is defined as follows:

$$F_\theta(\mathbf{x}) \equiv -\log \sum_{\tilde{\mathbf{h}}} \exp(-E_\theta(\mathbf{x}, \tilde{\mathbf{h}})). \quad (100)$$

The KL divergence from \mathbb{P}_θ to $\mathbb{P}_{\text{target}}$ can be minimized (or the log-likelihood of the target data having the empirical distribution $\mathbb{P}_{\text{target}}$ can be maximized) by maximizing

$$f(\theta) \equiv \mathbb{E}_{\text{target}} [\log \mathbb{P}_\theta(\mathbf{X})]. \quad (101)$$

The gradient and the Laplacian of $f(\theta)$ are given by

$$\nabla f(\theta) = \mathbb{E}_{\text{target}} [\mathbb{E}_\theta[\mathbf{S} | \mathbf{X}] - \mathbb{E}_\theta[\mathbf{S}]] \quad (102)$$

$$\nabla^2 f(\theta) = \mathbb{E}_{\text{target}} [\text{COV}_\theta[\mathbf{S} | \mathbf{X}] - \text{COV}_\theta[\mathbf{S}]], \quad (103)$$

where \mathbf{S} denotes the vector of the random variables representing the value of a unit ($U_i = X_i$ for $i \in [1, N]$ and $U_{N+i} = H_i$ for $i \in [1, M]$) or the product of the values of a pair of units:

$$\mathbf{S} \equiv (U_1, \dots, U_{N+M}, U_1 U_2, \dots, U_{N+M-1} U_{N+M}). \quad (104)$$

4 Learning a discriminative model

In this section, we study the Boltzmann machine whose visible units are divided into input units and output units (see Figure 2c). Such a Boltzmann machine is sometimes called a conditional Boltzmann machine [20]. Let N_{in} be the number of input units, N_{out} be the number of output units, $N = N_{\text{in}} + N_{\text{out}}$ be the number of visible units, and M be the number of hidden units.

Such a Boltzmann machine can be used to model a conditional probability distribution of the output values \mathbf{y} given the input values \mathbf{x} . Let $\mathbb{P}_\theta(\mathbf{y} | \mathbf{x})$ denote the conditional probability of \mathbf{y} given \mathbf{x} . When the Boltzmann machine has hidden units, we also write

$$\mathbb{P}_\theta(\mathbf{y} | \mathbf{x}) = \sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\mathbf{y}, \tilde{\mathbf{h}} | \mathbf{x}) \quad (105)$$

$$= \frac{\sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\mathbf{x}, \mathbf{y}, \tilde{\mathbf{h}})}{\sum_{\tilde{\mathbf{y}}, \tilde{\mathbf{h}}} \mathbb{P}_\theta(\mathbf{x}, \tilde{\mathbf{y}}, \tilde{\mathbf{h}})}. \quad (106)$$

4.1 Objective function

A training dataset for a discriminative model is a set of the pairs of input and output:

$$\mathcal{D} = \{(\mathbf{x}^{(d)}, \mathbf{y}^{(d)})\}_{d=1}^D. \quad (107)$$

When this training dataset is given, a natural objective function corresponding to the log-likelihood (10) for the generative model is

$$f(\theta) = \frac{1}{D} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} \log \mathbb{P}_\theta(\mathbf{y} | \mathbf{x}) \quad (108)$$

$$= \frac{1}{D} \log \prod_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} \mathbb{P}_\theta(\mathbf{y} | \mathbf{x}). \quad (109)$$

This objective function may also be related to a KL divergence. Let $\mathbb{P}_{\text{target}}(\cdot | \cdot)$ be the target conditional distribution, which we seek to model with $\mathbb{P}_\theta(\cdot | \cdot)$. Consider the following expected KL divergence from $\mathbb{P}_\theta(\cdot | \mathbf{X})$ to $\mathbb{P}_{\text{target}}(\cdot | \mathbf{X})$, where \mathbf{X} is the random variable denoting the input values that are distributed according to $\mathbb{P}_{\text{target}}$:

$$\begin{aligned} & \mathbb{E}_{\text{target}} [\text{KL}(\mathbb{P}_{\text{target}}(\cdot | \mathbf{X}) || \mathbb{P}_\theta(\cdot | \mathbf{X}))] \\ &= \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \sum_{\tilde{\mathbf{y}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{y}} | \tilde{\mathbf{x}}) \log \frac{\mathbb{P}_{\text{target}}(\tilde{\mathbf{y}} | \tilde{\mathbf{x}})}{\mathbb{P}_\theta(\tilde{\mathbf{y}} | \tilde{\mathbf{x}})} \end{aligned} \quad (110)$$

$$= \sum_{\tilde{\mathbf{x}}, \tilde{\mathbf{y}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \log \mathbb{P}_{\text{target}}(\tilde{\mathbf{y}} | \tilde{\mathbf{x}}) - \sum_{\tilde{\mathbf{x}}, \tilde{\mathbf{y}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \log \mathbb{P}_\theta(\tilde{\mathbf{y}} | \tilde{\mathbf{x}}). \quad (111)$$

The first term of the last expression is independent of θ . To minimize the expected KL divergence, it thus suffices to maximize the negation of the second term:

$$f(\theta) = \sum_{\tilde{\mathbf{x}}, \tilde{\mathbf{y}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \log \mathbb{P}_\theta(\tilde{\mathbf{y}} | \tilde{\mathbf{x}}) \quad (112)$$

$$= \sum_{\tilde{\mathbf{x}}, \tilde{\mathbf{y}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \log \frac{\mathbb{P}_\theta(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})}{\mathbb{P}_\theta(\tilde{\mathbf{x}})} \quad (113)$$

$$= \sum_{\tilde{\mathbf{x}}, \tilde{\mathbf{y}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \log \mathbb{P}_\theta(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \log \mathbb{P}_\theta(\tilde{\mathbf{x}}) \quad (114)$$

$$= \mathbb{E}_{\text{target}} [\log \mathbb{P}_\theta(\mathbf{X}, \mathbf{Y})] - \mathbb{E}_{\text{target}} [\log \mathbb{P}_\theta(\mathbf{X})] \quad (115)$$

When $\mathbb{P}_{\text{target}}$ is the empirical distribution of the pairs of input and output in the training dataset, (112) is reduced to (108).

4.2 Gradient, stochastic gradient, Laplacian

Observe that the first term of (115) is equivalent to the objective function of the generative model where both of input and output are visible (*i.e.*, input and output in (115) should be regarded as visible in (102)). The second term is analogous to the first term, but now only the input should be regarded as visible (*i.e.*, output and hidden in (115) should be regarded as hidden in (102)).

The gradient thus follows from (102):

$$\nabla f(\theta) = \mathbb{E}_{\text{target}} [\mathbb{E}_{\theta}[\mathbf{S} | \mathbf{X}, \mathbf{Y}]] - \mathbb{E}_{\theta}[\mathbf{S}] - (\mathbb{E}_{\text{target}} [\mathbb{E}_{\theta}[\mathbf{S} | \mathbf{X}]] - \mathbb{E}_{\theta}[\mathbf{S}]) \quad (116)$$

$$= \mathbb{E}_{\text{target}} [\mathbb{E}_{\theta}[\mathbf{S} | \mathbf{X}, \mathbf{Y}] - \mathbb{E}_{\theta}[\mathbf{S} | \mathbf{X}]], \quad (117)$$

where \mathbf{S} denotes the vector of random variables representing the value of a unit or the product of the values of a pair of units:

$$\mathbf{S} \equiv (U_1, \dots, U_{N+M}, U_1 U_2, \dots, U_{N+M-1} U_{N+M}), \quad (118)$$

where $U_i \equiv X_i$ for $i \in [1, N_{\text{in}}]$, $U_{N_{\text{in}}+i} \equiv Y_i$ for $i \in [1, N_{\text{out}}]$, and $U_{N+i} \equiv H_i$ for $i \in [1, M]$.

Because (117) is expressed as an expectation with respect to $\mathbb{P}_{\text{target}}$, it directly gives the following stochastic gradient:

$$g_{\theta}(\omega) = \mathbb{E}_{\theta}[\mathbf{S} | \mathbf{X}(\omega), \mathbf{Y}(\omega)] - \mathbb{E}_{\theta}[\mathbf{S} | \mathbf{X}(\omega)], \quad (119)$$

where $(\mathbf{X}(\omega), \mathbf{Y}(\omega))$ is sampled according to $\mathbb{P}_{\text{target}}$.

Specifically, we have the following learning rule of a stochastic gradient method:

$$b_i \leftarrow b_i + \eta (Y_i(\omega) - \mathbb{E}_{\theta}[Y_i | \mathbf{X}(\omega)]) \quad (120)$$

for an output unit $i \in [N_{\text{in}} + 1, N]$,

$$b_i \leftarrow b_i + \eta (\mathbb{E}_{\theta}[H_i | \mathbf{X}(\omega), \mathbf{Y}(\omega)] - \mathbb{E}_{\theta}[H_i | \mathbf{X}(\omega)]) \quad (121)$$

for a hidden unit $i \in [N + 1, N + M]$,

$$w_{i,j} \leftarrow w_{i,j} + \eta (X_i(\omega) Y_j(\omega) - X_i(\omega) \mathbb{E}_{\theta}[Y_j | \mathbf{X}(\omega)]) \quad (122)$$

for a pair of an input unit and an output unit $(i, j) \in [1, N_{\text{in}}] \times [N_{\text{in}} + 1, N]$,

$$w_{i,j} \leftarrow w_{i,j} + \eta (X_i(\omega) \mathbb{E}_{\theta}[H_j | \mathbf{X}(\omega), \mathbf{Y}(\omega)] - X_i(\omega) \mathbb{E}_{\theta}[H_j | \mathbf{X}(\omega)]) \quad (123)$$

for a pair of an input unit and a hidden unit $(i, j) \in [1, N_{\text{in}}] \times [N + 1, N + M]$,

$$w_{i,j} \leftarrow w_{i,j} + \eta (Y_i(\omega) Y_j(\omega) - \mathbb{E}_{\theta}[Y_i Y_j | \mathbf{X}(\omega)]) \quad (124)$$

for a pair of output units $(i, j) \in [N_{\text{in}} + 1, N - 1] \times [i + 1, N]$,

$$w_{i,j} \leftarrow w_{i,j} + \eta (Y_i(\omega) \mathbb{E}_{\theta}[H_j | \mathbf{X}(\omega), \mathbf{Y}(\omega)] - \mathbb{E}_{\theta}[Y_i H_j | \mathbf{X}(\omega)]) \quad (125)$$

for a pair of an output unit and a hidden unit $(i, j) \in [N_{\text{in}} + 1, N] \times [N + 1, N + M]$, and

$$w_{i,j} \leftarrow w_{i,j} + \eta (\mathbb{E}_{\theta}[H_i H_j | \mathbf{X}(\omega), \mathbf{Y}(\omega)] - \mathbb{E}_{\theta}[H_i H_j | \mathbf{X}(\omega)]) \quad (126)$$

for a pair of hidden units $(i, j) \in [N + 1, N + M - 1] \times [i + 1, N + M]$.

We also obtain trivial learning rules:

$$b_i \leftarrow b_i \quad (127)$$

for an input unit $i \in [1, N_{\text{in}}]$ and

$$w_{i,j} \leftarrow w_{i,j} \quad (128)$$

for a pair of input units $(i, j) \in [1, N_{\text{in}} - 1] \times [i + 1, N_{\text{in}}]$. One can easily see that these parameters are redundant and do not play any role in $\mathbb{P}_\theta(\mathbf{y} | \mathbf{x})$.

The Laplacian of the objective of discriminative learning follows from (103) and (115):

$$\nabla^2 f(\theta) = \mathbb{E}_{\text{target}} [\text{COV}_\theta[\mathbf{S} | \mathbf{X}, \mathbf{Y}]] - \text{COV}_\theta[\mathbf{S}] - (\mathbb{E}_{\text{target}} [\text{COV}_\theta[\mathbf{S} | \mathbf{X}]] - \text{COV}_\theta[\mathbf{S}]) \quad (129)$$

$$= \mathbb{E}_{\text{target}} [\text{COV}_\theta[\mathbf{S} | \mathbf{X}, \mathbf{Y}] - \text{COV}_\theta[\mathbf{S} | \mathbf{X}]], \quad (130)$$

where $\text{COV}_\theta[\cdot | \mathbf{X}, \mathbf{Y}]$ denotes the conditional covariance matrix with respect to the conditional distribution of the hidden values $\mathbb{P}_\theta(\cdot | \mathbf{X}, \mathbf{Y})$ given (\mathbf{X}, \mathbf{Y}) , and $\text{COV}_\theta[\cdot | \mathbf{X}]$ denotes the conditional covariance matrix with respect to the conditional distribution of the output and hidden values $\mathbb{P}_\theta(\cdot | \mathbf{X})$ given \mathbf{X} .

4.2.1 Summary

Consider a Boltzmann machine with parameters $\theta = (\mathbf{b}, \mathbf{W})$, where the units can be classified into input, output, and hidden. The Boltzmann machine defines a conditional probability distribution of the output values \mathbf{y} given the input values \mathbf{x} :

$$\mathbb{P}_\theta(\mathbf{y} | \mathbf{x}) = \frac{\sum_{\tilde{\mathbf{h}}} \mathbb{P}_\theta(\mathbf{x}, \mathbf{y}, \tilde{\mathbf{h}})}{\sum_{\tilde{\mathbf{y}}, \tilde{\mathbf{h}}} \mathbb{P}_\theta(\mathbf{x}, \tilde{\mathbf{y}}, \tilde{\mathbf{h}})}, \quad (131)$$

where $\mathbb{P}_\theta(\mathbf{x}, \mathbf{y}, \mathbf{h})$ is the probability distribution of the Boltzmann machine where (\mathbf{x}, \mathbf{y}) is visible, and \mathbf{h} is hidden.

The expected KL divergence from $\mathbb{P}_\theta(\cdot | \mathbf{X})$ to $\mathbb{P}_{\text{target}}(\cdot | \mathbf{X})$, where the expectation is with respect to the target distribution of the input values \mathbf{X} , can be minimized (or the sum of the conditional log-likelihood of the output values given the input values, where the input and output follow the empirical distribution $\mathbb{P}_{\text{target}}$, can be maximized) by maximizing

$$f(\theta) \equiv \mathbb{E}_{\text{target}} [\log \mathbb{P}_\theta(\mathbf{X}, \mathbf{Y})] - \mathbb{E}_{\text{target}} [\log \mathbb{P}_\theta(\mathbf{X})]. \quad (132)$$

The gradient and the Laplacian of $f(\theta)$ are given by

$$\nabla f(\theta) = \mathbb{E}_{\text{target}} [\mathbb{E}_\theta[\mathbf{S} | \mathbf{X}, \mathbf{Y}]] - \mathbb{E}_{\text{target}} [\mathbb{E}_\theta[\mathbf{S} | \mathbf{X}]] \quad (133)$$

$$\nabla^2 f(\theta) = \mathbb{E}_{\text{target}} [\text{COV}_\theta[\mathbf{S} | \mathbf{X}, \mathbf{Y}]] - \mathbb{E}_{\text{target}} [\text{COV}_\theta[\mathbf{S} | \mathbf{X}]], \quad (134)$$

where \mathbf{S} denotes the vector of the random variables representing the value of a unit ($U_i = X_i$ for $i \in [1, N_{\text{in}}]$, $U_{N_{\text{in}}+i} = Y_i$ for $i \in [1, N_{\text{out}}]$, and $U_{N+i} = H_i$ for $i \in [1, M]$) or the product of the values of a pair of units:

$$\mathbf{S} \equiv (U_1, \dots, U_{N+M}, U_1 U_2, \dots, U_{N+M-1} U_{N+M}). \quad (135)$$

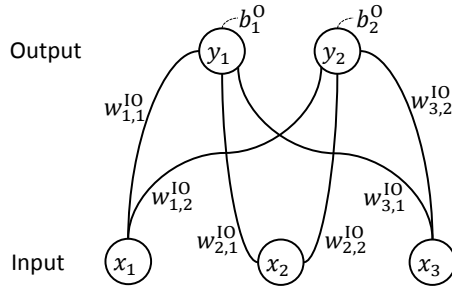


Figure 5: A discriminative model with a Boltzmann machine with no hidden units and no weight between output units.

4.3 Simplest case with relation to logistic regression

Here, we study the simplest but non-trivial case of a discriminative model of a Boltzmann machine. Specifically, we assume that the Boltzmann machine has no hidden units, and there are no weight between output units. As we have discussed in Section 4, the bias associated with the visible unit and the weight between visible units are redundant and do not play any role. Therefore, it suffices to consider the bias for output units and the weight between input units and output units (see Figure 5). Let \mathbf{b}^O be the bias associated with output units and \mathbf{W}^{IO} be the weight matrix where the (i, j) element denotes the weight between the i -th input unit and the j -th output unit.

4.3.1 Conditional probability

We first apply Lemma 1, where the input units correspond to the visible units in the lemma, and the output units correspond to the hidden units in the lemma. Then we can see that the probability distribution of output values given input values \mathbf{x} is given by a Boltzmann machine with the following bias and no weight:

$$\mathbf{b}(\mathbf{x}) \equiv \mathbf{b}^O + (\mathbf{W}^{IO})^\top \mathbf{x}. \quad (136)$$

The conditional probability of output values \mathbf{y} given input values \mathbf{x} is thus given by

$$\mathbb{P}_\theta(\mathbf{y} | \mathbf{x}) = \frac{\exp((\mathbf{b}(\mathbf{x}))^\top \mathbf{y})}{\sum_{\tilde{\mathbf{y}}} \exp((\mathbf{b}(\mathbf{x}))^\top \tilde{\mathbf{y}})}. \quad (137)$$

In this case, the partition function, which consists of $2^{N_{\text{out}}}$ terms, can be computed in $O(N_{\text{out}})$ time:

Lemma 2. *The partition function (the denominator) in the right-hand side of (137) can be written as follows:*

$$\sum_{\tilde{\mathbf{y}}} \exp((\mathbf{b}(\mathbf{x}))^\top \tilde{\mathbf{y}}) = \prod_{i=1}^{N_{\text{out}}} (1 + \exp(b_i(\mathbf{x}))), \quad (138)$$

where the summation with respect to $\tilde{\mathbf{y}}$ is over all of the possible binary patterns of length N_{out} .

Proof.

$$\sum_{\tilde{\mathbf{y}}} \exp((\mathbf{b}(\mathbf{x}))^\top \tilde{\mathbf{y}}) = \sum_{\tilde{\mathbf{y}}} \exp\left(\sum_{i=1}^{N_{\text{out}}} b_i(\mathbf{x}) \tilde{y}_i\right) \quad (139)$$

$$= \sum_{\tilde{\mathbf{y}}} \prod_{i=1}^{N_{\text{out}}} \exp(b_i(\mathbf{x}) \tilde{y}_i) \quad (140)$$

$$= \prod_{i=1}^{N_{\text{out}}} (1 + \exp(b_i(\mathbf{x}))). \quad (141)$$

□

Now, it can be easily shown that the output values are conditionally independent of each other given the input values:

Corollary 1. *The conditional probability of hidden values (137) can be written as the product of the conditional probabilities of each hidden value:*

$$\mathbb{P}_\theta(\mathbf{y} | \mathbf{x}) = \prod_{i=1}^{N_{\text{out}}} \mathbb{P}_\theta(y_i | \mathbf{x}), \quad (142)$$

where

$$\mathbb{P}_\theta(y_i | \mathbf{x}) = \frac{\exp(b_i(\mathbf{x}) y_i)}{1 + \exp(b_i(\mathbf{x}))}. \quad (143)$$

Proof. By Lemma 2, we have

$$\mathbb{P}_\theta(\mathbf{y} | \mathbf{x}) = \frac{\exp\left(\sum_{i=1}^{N_{\text{out}}} b_i(\mathbf{x}) y_i\right)}{\prod_{i=1}^{N_{\text{out}}} (1 + \exp(b_i(\mathbf{x})))} \quad (144)$$

$$= \prod_{i=1}^{N_{\text{out}}} \mathbb{P}_\theta(y_i | \mathbf{x}). \quad (145)$$

□

4.3.2 Relation to logistic regression

By (143), the probability that the i -th output unit takes the value 1 is

$$\mathbb{P}_\theta(Y_i = 1 | \mathbf{x}) = \frac{1}{1 + \exp(-b_i(\mathbf{x}))} \quad (146)$$

$$= \frac{1}{1 + \exp(-(b_i^O + \mathbf{w}_i^\top \mathbf{x}))}, \quad (147)$$

where \mathbf{w}_i is the i -th column of \mathbf{W}^{IO} . The last expression has the form of logistic regression, where the explanatory variables \mathbf{x} are binary.

Due to the conditional independence shown in Corollary 1, we can say that the Boltzmann machine shown in Figure 5 consists of N independent models of logistic regression that have common explanatory variables.

Algorithm 1 A Gibbs sampler for a Boltzmann machine with N units.

- 1: **Input** Parameters of the Boltzmann machine and K
 - 2: $\mathbf{x}^{(0)} \leftarrow$ Initialize the values of the N units
 - 3: **for** $k = 1, \dots, K$ **do**
 - 4: **for** $i = 1, \dots, N$ **do**
 - 5: Choose $x_i^{(k)}$ according to its conditional distribution given $x_j^{(k-1)}$ for $j \neq i$
 - 6: **end for**
 - 7: **end for**
 - 8: **Return** K samples: $\mathbf{x}^{(k)}$ for $k = 1, \dots, K$.
-

5 Evaluating expectation with respect to a model distribution

When we train a Boltzmann machine with a stochastic gradient method, we need to evaluate expected values with respect to the distribution defined by the Boltzmann machine. Such expected values appear for example as $\mathbb{E}_\theta[\mathbf{S}]$ in (49) and (102) or as $\mathbb{E}_\theta[\mathbf{S} | \mathbf{X}]$ in (133). In general, exact expressions for these expectations are unavailable in closed forms.

5.1 Gibbs sampler

A practical approach that can be used to estimate those expectations is Markov Chain Monte Carlo in general and Glauber dynamics [36] or Gibbs samplers [5] in particular. For example, we can sample K patterns according to the distribution given by a Boltzmann machine via a Gibbs sampler shown in Algorithm 1. The expected values can then be estimated using the K samples. In practice, we often ignore initial samples and consider only every n th sample for a sufficiently large n .

In Step 5 of Algorithm 1, the conditional distribution of $x_i^{(k)}$ given $x_j^{(k-1)}$ for $j \neq i$ is defined by

$$\mathbb{P}_\theta(x_i^{(k)} | \mathbf{x}^{(k-1)}) = \frac{\exp(-E_\theta(x_i^{(k)} | \mathbf{x}^{(k-1)}))}{\sum_{\tilde{x}_i^{(k)} \in \{0,1\}} \exp(-E_\theta(\tilde{x}_i^{(k)} | \mathbf{x}^{(k-1)}))}, \quad (148)$$

for $x_i^{(k)} \in \{0, 1\}$, where

$$E_\theta(x_i^{(k)} | \mathbf{x}^{(k-1)}) \equiv -b_i x_i^{(k)} - \sum_{j \neq i} x_i^{(k)} w_{i,j} x_j^{(k-1)}. \quad (149)$$

5.2 Contrastive divergence

Another approach to deal with the computational intractability of evaluation the expectations is to avoid it. Namely, we modify our objective function.

Recall, from (7) and (10), that our objective function has been the KL divergence from \mathbb{P}_θ to $\mathbb{P}_{\text{target}}$ (or equivalently the log likelihood of data with respect to \mathbb{P}_θ). The gradient of the KL divergence with respect to θ involves the computationally intractable term of expectation with respect to \mathbb{P}_θ .

Here, we study an alternative objective function of Contrastive divergence [18, 4]. Consider a Gibbs sampler (Algorithm 1) that initializes the values by sampling from $\mathbb{P}_{\text{target}}$ (or uniformly at random from the dataset \mathcal{D}). Because this is the distribution at the beginning of the Gibbs sampler, we write $\mathbb{P}_0 \equiv \mathbb{P}_{\text{target}}$. The distribution of the patterns sampled by the Gibbs sampler at step k is referred to as \mathbb{P}_k^θ . Because $\mathbb{P}_k^\theta \rightarrow \mathbb{P}_\theta$ as $k \rightarrow \infty$, we also write $\mathbb{P}_\infty^\theta \equiv \mathbb{P}_\theta$.

The KL divergence in (7) can now be written as

$$\text{KL}(\mathbb{P}_0 || \mathbb{P}_\infty^\theta) = \sum_{\tilde{\mathbf{x}}} \mathbb{P}_0(\tilde{\mathbf{x}}) \log \mathbb{P}_0(\tilde{\mathbf{x}}) - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_0(\tilde{\mathbf{x}}) \log \mathbb{P}_\infty^\theta(\tilde{\mathbf{x}}), \quad (150)$$

where

$$\mathbb{P}_\infty^\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{\sum_{\tilde{\mathbf{x}}} \exp(-E_\theta(\tilde{\mathbf{x}}))} \quad (151)$$

and its gradient (recall (18)) as

$$\nabla_\theta \text{KL}(\mathbb{P}_0 \parallel \mathbb{P}_\infty^\theta) = \sum_{\tilde{\mathbf{x}}} \mathbb{P}_0(\tilde{\mathbf{x}}) \nabla_\theta E_\theta(\tilde{\mathbf{x}}) - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_\infty^\theta(\tilde{\mathbf{x}}) \nabla_\theta E_\theta(\tilde{\mathbf{x}}). \quad (152)$$

The first term of the right-hand side of (152) is the expectation with respect to the target distribution \mathbb{P}_0 , which is readily computable. The second term is the expectation with respect to the model (with parameter θ), which is in general computationally intractable for a Boltzmann machine.

To cancel out this computationally intractable second term, consider the following contrastive divergence:

$$\text{CD}_1(\theta) \equiv \text{KL}(\mathbb{P}_0 \parallel \mathbb{P}_\infty^\theta) - \text{KL}(\mathbb{P}_1^\theta \parallel \mathbb{P}_\infty^\theta). \quad (153)$$

Here, we quote an intuitive motivation of the contrastive divergence from [18], using our notations (shown within [·]):

The intuitive motivation for using this “contrastive divergence” is that we would like the Markov chain that is implemented by Gibbs sampling to leave the initial distribution $[\mathbb{P}_0]$ over the visible variables unaltered. Instead of running the chain to equilibrium and comparing the initial and final derivatives we can simply run the chain for one full step and then update the parameters to reduce the tendency of the chain to wander away from the initial distribution on the first step. Because $[\mathbb{P}_1^\theta]$ is one step closer to the equilibrium distribution than $[\mathbb{P}_0]$, we are guaranteed that $[\text{KL}(\mathbb{P}_0 \parallel \mathbb{P}_\infty^\theta)]$ exceeds $[\text{KL}(\mathbb{P}_1^\theta \parallel \mathbb{P}_\infty^\theta)]$ unless $[\mathbb{P}_0]$ equals $[\mathbb{P}_1^\theta]$, so the contrastive divergence can never be negative. Also, for Markov chains in which all transitions have non-zero probability, $[\mathbb{P}_0 = \mathbb{P}_1^\theta]$ implies $[\mathbb{P}_0 = \mathbb{P}_1^\theta]$ so the contrastive divergence can only be zero if the model is perfect.

The gradient of the second term of (153) can be derived as follows:

$$\text{KL}(\mathbb{P}_1^\theta \parallel \mathbb{P}_\infty^\theta) = \sum_{\tilde{\mathbf{x}}} \mathbb{P}_1^\theta(\tilde{\mathbf{x}}) \log \frac{\mathbb{P}_1^\theta(\tilde{\mathbf{x}})}{\mathbb{P}_\infty^\theta(\tilde{\mathbf{x}})} \quad (154)$$

$$\nabla_\theta \text{KL}(\mathbb{P}_1^\theta \parallel \mathbb{P}_\infty^\theta) = \sum_{\tilde{\mathbf{x}}} \nabla_\theta \mathbb{P}_1^\theta(\tilde{\mathbf{x}}) \log \frac{\mathbb{P}_1^\theta(\tilde{\mathbf{x}})}{\mathbb{P}_\infty^\theta(\tilde{\mathbf{x}})} + \sum_{\tilde{\mathbf{x}}} \mathbb{P}_1^\theta(\tilde{\mathbf{x}}) \nabla_\theta \log \mathbb{P}_1^\theta(\tilde{\mathbf{x}}) - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_1^\theta(\tilde{\mathbf{x}}) \nabla_\theta \log \mathbb{P}_\infty^\theta(\tilde{\mathbf{x}}) \quad (155)$$

$$= \sum_{\tilde{\mathbf{x}}} \nabla_\theta \mathbb{P}_1^\theta(\tilde{\mathbf{x}}) \log \frac{\mathbb{P}_1^\theta(\tilde{\mathbf{x}})}{\mathbb{P}_\infty^\theta(\tilde{\mathbf{x}})} + \sum_{\tilde{\mathbf{x}}} \nabla_\theta \mathbb{P}_1^\theta(\tilde{\mathbf{x}}) - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_1^\theta(\tilde{\mathbf{x}}) \nabla_\theta \log \mathbb{P}_\infty^\theta(\tilde{\mathbf{x}}) \quad (156)$$

$$= \sum_{\tilde{\mathbf{x}}} \nabla_\theta \mathbb{P}_1^\theta(\tilde{\mathbf{x}}) \left(1 + \log \frac{\mathbb{P}_1^\theta(\tilde{\mathbf{x}})}{\mathbb{P}_\infty^\theta(\tilde{\mathbf{x}})} \right) + \sum_{\tilde{\mathbf{x}}} \mathbb{P}_1^\theta(\tilde{\mathbf{x}}) \nabla_\theta E_\theta(\tilde{\mathbf{x}}) - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_\infty^\theta(\tilde{\mathbf{x}}) \nabla_\theta E_\theta(\tilde{\mathbf{x}}), \quad (157)$$

where the last equality follows analogously to (18).

From (152)-(153) and (157), we obtain

$$\nabla_\theta \text{CD}_1(\theta) = \sum_{\tilde{\mathbf{x}}} \mathbb{P}_0(\tilde{\mathbf{x}}) \nabla_\theta E_\theta(\tilde{\mathbf{x}}) - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_1(\tilde{\mathbf{x}}) \nabla_\theta E_\theta(\tilde{\mathbf{x}}) + \varepsilon(\theta), \quad (158)$$

where

$$\varepsilon(\theta) \equiv \sum_{\tilde{\mathbf{x}}} \nabla_\theta \mathbb{P}_1^\theta(\tilde{\mathbf{x}}) \left(1 + \log \frac{\mathbb{P}_1^\theta(\tilde{\mathbf{x}})}{\mathbb{P}_\infty^\theta(\tilde{\mathbf{x}})} \right). \quad (159)$$

Hinton has empirically shown that $\varepsilon(\theta)$ is small and recommended the approximation of $\varepsilon(\theta) \approx 0$ in [18]. The first term of the right-hand side of (158) is an expectation with respect to \mathbb{P}_0 and can be readily evaluated. The second term is an expectation with respect to \mathbb{P}_1^θ and can be estimated with the samples from the Gibbs sampler in one step.

In [18], $\varepsilon(\theta)$ is represented in the following equivalent form:

$$\varepsilon(\theta) = \sum_{\tilde{\mathbf{x}}} \nabla_{\theta} \mathbb{P}_1^\theta(\tilde{\mathbf{x}}) \frac{\partial}{\partial \mathbb{P}_1(\tilde{\mathbf{x}})} \text{KL}(\mathbb{P}_1 \parallel \mathbb{P}_\infty^\theta). \quad (160)$$

5.2.1 Score matching (Fisher divergence)

A particularly interesting objective function for energy-based models is score matching [22], which has been subsequently used for example in [66, 29, 24, 30, 60, 62]. Similar to contrastive divergence, score matching is an objective function, which we can avoid computationally intractable evaluation of expectation with.

Specifically, Hyvärinen [22] defines score matching as

$$\text{FD}(\mathbb{P}_{\text{target}} \parallel \mathbb{P}_\theta) = \int_{\mathbf{x}} \mathbb{P}_{\text{target}}(\mathbf{x}) |\nabla_{\mathbf{x}} \log \mathbb{P}_{\text{target}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log \mathbb{P}_\theta(\mathbf{x})|^2 d\mathbf{x}, \quad (161)$$

which is also referred to as Fisher divergence [38]. Note that the gradient is with respect to \mathbf{x} and not θ . Hyvärinen shows that

$$\theta^* \equiv \underset{\theta}{\text{argmin}} \text{FD}(\mathbb{P}_{\text{target}} \parallel \mathbb{P}_\theta) \quad (162)$$

is a consistent estimator [22].

A key property of score matching is that there is no need for calculating the partition function. In particular,

$$\nabla_{\mathbf{x}} \log \mathbb{P}_\theta(\mathbf{x}) = -\nabla_{\mathbf{x}} E_\theta(\mathbf{x}) - \nabla_{\mathbf{x}} \log \int_{\tilde{\mathbf{x}}} \exp(-E_\theta(\tilde{\mathbf{x}})) d\tilde{\mathbf{x}} \quad (163)$$

$$= -\nabla_{\mathbf{x}} E_\theta(\mathbf{x}). \quad (164)$$

Also, although $\nabla_{\tilde{\mathbf{x}}} \log \mathbb{P}_{\text{target}}(\mathbf{x})$ might appear to be intractable, it has been shown in [22] that

$$\frac{1}{D} \sum_{\mathbf{x} \in \mathcal{D}} \sum_{i=1}^N \left(\frac{\partial \log \mathbb{P}_\theta(\mathbf{x})}{\partial x_i^2} + \frac{1}{2} \left(\frac{\partial \log \mathbb{P}_\theta(\mathbf{x})}{\partial x_i} \right)^2 \right) + \text{const} \quad (165)$$

is asymptotically equivalent to $\text{FD}(\mathbb{P}_{\text{target}} \parallel \mathbb{P}_\theta)$ as $D \equiv |\mathcal{D}| \rightarrow \infty$, where \mathcal{D} is the set of data (samples from $\mathbb{P}_{\text{target}}$), N is the dimension of $\mathbf{x} \in \mathcal{D}$, and const is the term independent of θ .

A limitation of the estimator in (165) is that it assumes among others that the variables \mathbf{x} are continuous and $\mathbb{P}_\theta(\cdot)$ is differentiable. There has been prior work for relaxing these assumptions. For example, Hyvärinen studies an extension for binary or non-negative data [23], and Kingma and LaCun study an approach of adding Gaussian noise to data samples for smoothness condition [29].

5.3 A separate generator

The learning rule that follows from maximization of loglikelihood (minimization of KL divergence) via a gradient ascent method may be considered as decreasing the energy of “positive” or “real” samples that are generated according to a target distribution $\mathbb{P}_{\text{target}}$ and increasing the energy of “negative” or “fake” samples that are generated according to the current model (recall (18)):

$$\nabla_{\theta} f(\theta) = - \sum_{\tilde{\mathbf{x}}} \mathbb{P}_{\text{target}}(\tilde{\mathbf{x}}) \nabla_{\theta} E_\theta(\tilde{\mathbf{x}}) + \sum_{\tilde{\mathbf{x}}} \mathbb{P}_\theta(\tilde{\mathbf{x}}) \nabla_{\theta} E_\theta(\tilde{\mathbf{x}}). \quad (166)$$

With this learning rule, one can let the model to be able to better “discriminate between the positive examples from the original data and the negative examples generated by sampling from the current density estimate” [65].

Then an energy-based model may be considered as taking both the role of a generator and the role of a discriminator in a generative adversarial network (GAN) [53, 14]. There is a line of work [15, 8, 67, 26] that prepares a separate generator with an energy-based model. If the separate generator allows more efficient sampling than the energy-based model, the expectation with respect to the distribution of the current generator (*i.e.*, $\sum_{\tilde{\mathbf{x}}} \mathbb{P}_\theta(\tilde{\mathbf{x}}) \nabla_\theta E_\theta(\tilde{\mathbf{x}})$ in (166)) can be more efficiently evaluated [26, 15].

5.4 Mean-field Boltzmann machines

There are also approaches that avoid computationally expensive evaluation of expectation. An example is a mean-field Boltzmann machine [49], which can be used to approximate a Boltzmann machine. A mean-field Boltzmann machine ignores connections between units and chooses the real value m_i for each unit i in a way that the distribution defined through

$$\mathbb{Q}_{\mathbf{m}}(\mathbf{x}) \equiv \prod_i m_i^{x_i} (1 - m_i)^{1-x_i} \quad (167)$$

well approximates the distribution $\mathbb{P}_\theta(\mathbf{x})$ of a Boltzmann machine in the sense of the KL divergence [63].

6 Other energy-based models

Here we review stochastic models that are related to the Boltzmann machine.

6.1 Markov random fields

A Boltzmann machine is a Markov random field [27] having a particular structure. A Markov random field consists of a finite set of units similar to a Boltzmann machine. Each unit of a Markov random field takes a value in a finite set. The probability distribution of the values (configurations) of the Markov random field can be represented as

$$\mathbb{P}(\mathbf{x}) = \frac{\exp(-E(\mathbf{x}))}{\sum_{\tilde{\mathbf{x}}} \exp(-E(\tilde{\mathbf{x}}))}, \quad (168)$$

where the summation with respect to $\tilde{\mathbf{x}}$ is over all of the possible configurations of the values in the finite set, for which the Markov random field is defined. The energy of a configuration is defined as follows:

$$E(\mathbf{x}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}), \quad (169)$$

where $\boldsymbol{\phi}(\mathbf{x})$ is a feature vector of \mathbf{x} . A Markov random field is also called an undirected graphical model.

6.1.1 Boltzmann machine and Ising model

A Markov random field is reduced to a Boltzmann machine when it has the following two properties. First, $\boldsymbol{\phi}(\mathbf{x})$ is a vector of monomials of degrees up to 2. Second, each unit takes a binary value.

An Ising model [25] is essentially equivalent to a Boltzmann machine but the binary variable takes values in $\{-1, +1\}$.

6.1.2 Higher-order Boltzmann machine

A higher-order Boltzmann machine extends a Boltzmann machine by allowing $\phi(\mathbf{x})$ to include monomials of degree greater than 2 [55]. Each unit of a higher-order Boltzmann machine takes a binary value.

6.2 Determinantal point process

A determinantal point process (DPP) defines a probability distribution over the subsets of a given ground set [39, 32, 56, 57]. In our context, the ground set \mathcal{Y} can be considered as the set of all units:

$$\mathcal{Y} = \{1, 2, \dots, N\}. \quad (170)$$

A subset \mathcal{X} can be considered as a set of units that take the value 1:

$$\mathcal{X} = \{i \mid x_i = 1, i \in \mathcal{Y}\}. \quad (171)$$

A DPP can be characterized by a kernel \mathbf{L} , which is an $N \times N$ positive semi-definite matrix. The probability that the subset \mathcal{X} is selected (*i.e.*, the units in \mathcal{X} take the value 1) is then given by

$$\mathbb{P}(\mathcal{X}) = \frac{\det(\mathbf{L}_{\mathcal{X}})}{\det(\mathbf{L} + \mathbf{I})}, \quad (172)$$

where $\mathbf{L}_{\mathcal{X}}$ denotes the principal submatrix of \mathbf{L} indexed by \mathcal{X} , and \mathbf{I} is the $N \times N$ identity matrix. A DPP can be seen as an energy-based model, whose energy is given by $E(\mathcal{X}) = -\log \det(\mathbf{L}_{\mathcal{X}})$.

In general, the kernel \mathbf{L} can be represented as

$$\mathbf{L} = \mathbf{B}^{\top} \mathbf{B} \quad (173)$$

by the use of a $K \times N$ matrix \mathbf{B} , where K is the rank of \mathbf{L} . For $i \in [1, N]$, let \mathbf{b}_i be the i -th column of \mathbf{B} , which may be understood as a feature vector of the i -th item (unit). One can further decompose \mathbf{b}_i into $\mathbf{b}_i = q_i \phi_i$, where $\|\phi_i\| = 1$ and $q_i \geq 0$. Then we can write the (i, j) -th element of \mathbf{L} as follows:

$$\ell_{i,j} = (\mathbf{b}_i)^{\top} \mathbf{b}_j = q_i (\phi_i)^{\top} \phi_j q_j. \quad (174)$$

In particular,

$$\mathbb{P}(\{i\}) \sim \det(\mathbf{L}_{\{i\}}) = \ell_{i,i} = q_i^2, \quad (175)$$

so that q_i can be understood as the "quality" of the i -th item. Specifically, given that exactly one item is selected, the conditional probability that the i -th item is selected is proportional to q_i^2 . Likewise,

$$\mathbb{P}(\{i, j\}) \sim \det(\mathbf{L}_{\{i,j\}}) = \det \begin{pmatrix} q_i (\phi_i)^{\top} \phi_i q_i & q_i (\phi_i)^{\top} \phi_j q_j \\ q_j (\phi_j)^{\top} \phi_i q_i & q_j (\phi_j)^{\top} \phi_j q_j \end{pmatrix} = q_i^2 q_j^2 (1 - S_{i,j}^2), \quad (176)$$

where

$$S_{i,j} \equiv (\phi_i)^{\top} \phi_j \quad (177)$$

may be understood as the similarity between item i and item j . Equation (176) implies that the similar items are unlikely to be selected together. In general, a DPP tends to give high probability to diverse subsets of items having high quality.

Finally, we discuss a computational aspect of a DPP. Let's compare the denominator of the right-hand side of (172) against the corresponding denominator (partition function) of the Boltzmann machine in (4). The partition function of the Boltzmann machine is a summation over 2^N terms, which

suggest that we need $O(2^N)$ time to evaluate it. On the other hand, the determinant of an $N \times N$ matrix can be computed in $O(N^3)$ time.

When \mathbf{L} has a low rank ($K < N$), one can further reduce the computational complexity. Let

$$\mathbf{C} = \mathbf{B}\mathbf{B}^\top \quad (178)$$

be the $K \times K$ positive semi-definite matrix, defined from (173). Because the eigenvalues $\{\lambda_1, \dots, \lambda_K\}$ of \mathbf{L} are eigenvalues of \mathbf{C} and vice versa, we have

$$\det(\mathbf{L} + \mathbf{I}) = \prod_{k=1}^K (\lambda_k + 1) = \det(\mathbf{C} + \mathbf{I}), \quad (179)$$

where the first identity matrix \mathbf{I} is $N \times N$, and the second is $K \times K$. Therefore, (172) can be represented as follows:

$$\mathcal{P}(\mathcal{X}) = \frac{\det(\mathbf{L}\mathcal{X})}{\det(\mathbf{C} + \mathbf{I})}. \quad (180)$$

The denominator of this dual representation of a DPP can be evaluated in $O(K^3)$ time.

The DPP has been receiving increasing attention in machine learning, and various learning algorithms have been proposed in the literature [12, 11, 40, 46].

6.3 Gaussian Boltzmann machines

Here we review energy based models that deal with real values with a particular focus on those models that are extended from Boltzmann machines. A standard approach to extend the Boltzmann machine to deal with real values is the use of a Gaussian unit [41, 64, 31].

6.3.1 Gaussian Bernoulli restricted Boltzmann machines

For real values $\mathbf{x} \in \mathbb{R}^N$ and binary values $\mathbf{h} \in \{0, 1\}^M$, Krizhevsky studies the following energy [31]:

$$E_\theta(\mathbf{x}, \mathbf{h}) = \sum_{i=1}^N \frac{(x_i - b_i^V)^2}{2\sigma_i^2} - \sum_{j=1}^M b_j^H h_j - \sum_{i=1}^N \sum_{j=1}^M x_i \frac{w_{i,j}}{\sigma_i} h_j, \quad (181)$$

where $\theta \equiv (\mathbf{b}^V, \mathbf{b}^H, \mathbf{W}, \boldsymbol{\sigma})$ is the set of parameters.

Krizhevsky shows that the conditional probability of \mathbf{x} given \mathbf{h} has a normal distribution, and the conditional probability of \mathbf{h} given \mathbf{x} has a Bernoulli distribution [31]. Here, we re-derive them with our notations and in a simpler manner.

Theorem 1. *Consider the energy given by (181). Then the elements of \mathbf{x} are conditionally independent of each other given \mathbf{h} , and the elements of \mathbf{h} are conditionally independent of each other given \mathbf{x} . Also, the conditional probability density of x_i given \mathbf{h} is given by*

$$p_\theta^{(i)}(x_i | \mathbf{h}) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{\left(x_i - (b_i^V + \sigma_i \sum_{j=1}^M w_{i,j} h_j)\right)^2}{2\sigma_i^2}\right) \quad (182)$$

for $x_i \in \mathbb{R}$, and the conditional mass probability of h_j for $h_j \in \{0, 1\}$ given \mathbf{x} is given by

$$p_\theta^{(j)}(h_j | \mathbf{x}) = \frac{\exp\left(\left(b_j^H + \sum_{i=1}^N x_i \frac{w_{i,j}}{\sigma_i}\right)h_j\right)}{1 + \exp\left(b_j^H + \sum_{i=1}^N x_i \frac{w_{i,j}}{\sigma_i}\right)}. \quad (183)$$

Proof. First observe that

$$\exp(-E_\theta(\mathbf{x}, \mathbf{h})) = \exp((\mathbf{b}^H)^\top \mathbf{h}) \prod_{i=1}^N \exp(-E_\theta^{(i)}(x_i, \mathbf{h})), \quad (184)$$

where

$$E_\theta^{(i)}(x_i, \mathbf{h}) = \frac{(x_i - b_i^V)^2}{2\sigma_i^2} - \sum_{j=1}^M x_i \frac{w_{i,j}}{\sigma_i} h_j. \quad (185)$$

This means that the elements of \mathbf{x} are conditionally independent of each other given \mathbf{h} . The conditional independence of the elements of \mathbf{h} given \mathbf{x} can be shown analogously.

Then we have

$$p_\theta(\mathbf{x} | \mathbf{h}) = \prod_{i=1}^N p_\theta^{(i)}(x_i | \mathbf{h}) \quad (186)$$

$$p_\theta(\mathbf{h} | \mathbf{x}) = \prod_{j=1}^M p_\theta^{(j)}(h_j | \mathbf{x}), \quad (187)$$

where

$$p_\theta^{(i)}(x_i | \mathbf{h}) \sim \exp(-E_\theta^{(i)}(x_i, \mathbf{h})) \quad (188)$$

$$= \exp\left(-\frac{x_i^2 - 2\left(b_i^V + \sigma_i \sum_{j=1}^M w_{i,j} h_j + (b_i^V)^2\right)}{2\sigma_i^2}\right) \quad (189)$$

$$\sim \exp\left(-\frac{\left(x_i - (b_i^V + \sigma_i \sum_{j=1}^M w_{i,j} h_j)\right)^2}{2\sigma_i^2}\right) \quad (190)$$

$$p_\theta^{(j)}(h_j | \mathbf{x}) \sim \exp\left(\left(b_j^H + \sum_{i=1}^N x_i \frac{w_{i,j}}{\sigma_i}\right) h_j\right) \quad (191)$$

for $x_i \in \mathbb{R}$ and $h_j \in \{0, 1\}$. By taking into account the normalization for the total probability to become 1, we obtain (182) and (183). \square

One might wonder why the product $\sigma_i w_{i,j}$ appears in (182), and the quotient $w_{i,j}/\sigma_i$ appears in (183). We can shed light on these expressions by studying natural parameters as in [64]. The natural parameters of the normal distribution with mean μ and standard deviation σ are μ/σ^2 and $-1/(2\sigma^2)$. Hence, the natural parameters in (182) are

$$\frac{b_i^V}{\sigma_i^2} + \sum_{j=1}^M \frac{w_{i,j}}{\sigma_i} h_j \quad \text{and} \quad -\frac{1}{2\sigma_i^2}. \quad (192)$$

Likewise, the natural parameter in (183) is

$$b_j^H + \sum_{i=1}^N x_i \frac{w_{i,j}}{\sigma_i}. \quad (193)$$

Therefore, only the quotient $w_{i,j}/\sigma_i$ appears in natural parameters.

6.3.2 Spike and slab restricted Boltzmann machines

Courville et al. study a class of particularly structured higher-order Boltzmann machines with Gaussian units, which they refer to as spike and slab RBMs [6, 7]. For example, the energy may be represented as follows:

$$E_{\theta}(\mathbf{x}, \mathbf{h}, \mathbf{S}) = \sum_{i=1}^N \frac{\lambda_i}{2} x_i^2 + \sum_{j=1}^M \sum_{k=1}^K \frac{\alpha_{j,k}}{2} s_{j,k}^2 - \sum_{j=1}^M b_j h_j - \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^K w_{i,j,k} x_i h_j s_{j,k}, \quad (194)$$

where \mathbf{x} denotes real-valued visible values, \mathbf{h} denotes binary hidden values (“spikes”), \mathbf{S} denotes real-valued “slabs,” and $\theta \equiv (\boldsymbol{\lambda}, \boldsymbol{\alpha}, \mathbf{W}, \mathbf{b})$ denotes the parameters. The term $w_{i,j,k} x_i h_j s_{j,k}$ represents a three way interactions.

Similar to Theorem 1, one can show that the elements of \mathbf{x} are conditionally independent of each other and have normal distributions given \mathbf{h} and \mathbf{S} :

$$p(x_i | \mathbf{h}, \mathbf{S}) \sim \exp\left(-\frac{\lambda_i}{2} \left(x_i - \frac{1}{\lambda_i} \sum_{j=1}^M \sum_{k=1}^K w_{i,j,k} h_j s_{j,k}\right)^2\right) \quad (195)$$

for $x_i \in \mathbb{R}$. Likewise, the elements of \mathbf{S} are conditionally independent of each other and have normal distributions given \mathbf{x} and \mathbf{h} :

$$p(s_{j,k} | \mathbf{x}, \mathbf{h}) \sim \exp\left(-\frac{\alpha_{j,k}}{2} \left(s_{j,k} - \frac{1}{\alpha_{j,k}} \sum_{i=1}^N w_{i,j,k} x_i h_j\right)^2\right) \quad (196)$$

for $s_{j,k} \in \mathbb{R}$. Given \mathbf{x} and \mathbf{S} , the elements of \mathbf{h} are conditionally independent of each other and have Bernoulli distributions:

$$p(h_j | \mathbf{x}, \mathbf{S}) \sim \exp\left(\left(b_j + \sum_{i=1}^N \sum_{k=1}^K w_{i,j,k} x_i s_{j,k}\right) h_j\right) \quad (197)$$

for $h_j \in \{0, 1\}$.

It has been argued and experimentally confirmed that spike and slab RBMs can generate images with sharper boundaries than those generated by models with binary hidden units [6, 7].

6.4 Using expected values to represent real values

Here, we discuss the approach of using expected values given by the probability distribution defined by a Boltzmann machine. Because a unit of a Boltzmann machine takes a binary value, 0 or 1, the expected value is in $[0, 1]$. With appropriate scaling, any closed interval can be mapped to $[0, 1]$.

6.4.1 Expected values in visible units

Recall, from Section 3, that a Boltzmann machine with parameter $\theta \equiv (\mathbf{b}, \mathbf{W})$ defines a probability distribution over binary values:

$$\mathbb{P}_{\theta}(\mathbf{x}) = \frac{\exp(-E_{\theta}(\mathbf{x}))}{\sum_{\tilde{\mathbf{x}}} \exp(-E_{\theta}(\tilde{\mathbf{x}}))}, \quad (198)$$

where

$$E_{\theta}(\mathbf{x}) \equiv -\mathbf{b}^{\top} \mathbf{x} - \mathbf{x}^{\top} \mathbf{W} \mathbf{x} \quad (199)$$

for $\mathbf{x} \in \{0, 1\}^N$. The expected values can then be given as

$$\mathbb{E}_\theta[\mathbf{X}] = \sum_{\mathbf{x}} \mathbf{x} \mathbb{P}_\theta(\mathbf{x}), \quad (200)$$

which take values in $[0, 1]^N$.

A question is whether the values given by the expectation are suitable for a particular purpose. In other words, how should we set θ so that the expectation gives suitable values for the purpose under consideration?

Here, we discuss a particular learning rule of simply using the values in $[0, 1]^N$ in the learning rules for binary values. In particular, a gradient ascent method for generative learning without hidden units is given by (28) and (29). A stochastic gradient method with mini-batches is then given by

$$b_i \leftarrow b_i + \eta \left(\frac{1}{K} \sum_{k=1}^K X_i(\omega_k) - \mathbb{E}_\theta[X_i] \right) \quad (201)$$

$$w_{i,j} \leftarrow w_{i,j} + \eta \left(\frac{1}{K} \sum_{k=1}^K X_i(\omega_k) X_j(\omega_k) - \mathbb{E}_\theta[X_i X_j] \right), \quad (202)$$

where we take K samples, $\mathbf{X}(\omega_1), \dots, \mathbf{X}(\omega_K)$, at each step of the mini-batch stochastic gradient method. As $K \rightarrow \infty$, these learning rules converge to the gradient ascent method given by (28) and (29). If the real values under consideration are actually expectation of some binary random variables, we can justify a stochastic gradient method of updating b_i based on a sampled real value $R_i(\omega)$ according to

$$b_i \leftarrow b_i + \eta (R_i(\omega) - \mathbb{E}_\theta[X_i]). \quad (203)$$

However, the corresponding update of $w_{i,j}$ cannot be justified, because $\mathbb{E}_\theta[X_i X_j]$ cannot be represented solely by $\mathbb{E}_\theta[X_i]$ and $\mathbb{E}_\theta[X_j]$ unless X_i and X_j are independent. If X_i and X_j are independent, we should have $w_{i,j} = 0$.

In general, the use of expected values can be justified only for the special case where the corresponding random variables are conditionally independent of each other given input values. The simple discriminative model in Figure 5 is an example of such a special case. In this case, one can apply a stochastic gradient method of updating the parameters as follows:

$$b_j \leftarrow b_j + \eta (R_j(\omega) - \mathbb{E}_\theta[Y_j | \mathbf{X}(\omega)]) \quad (204)$$

$$w_{i,j} \leftarrow b_{i,j} + \eta (X_i(\omega) R_j(\omega) - X_i(\omega) \mathbb{E}_\theta[Y_j | \mathbf{X}(\omega)]) \quad (205)$$

for a sampled pair $(X_i(\omega), R_j(\omega))$, where $X_i(\omega)$ is an input binary value for $i \in [1, N_{\text{in}}]$, and $R_j(\omega)$ is an output real value in $[0, 1]$ for $j \in [1, N_{\text{out}}]$.

6.4.2 Expected values in hidden units

Expected values are more often used for hidden units [58, 59, 10, 44] than for visible units.

Consider a Boltzmann machine with visible units and hidden units, which have no connections between visible units or between hidden units (namely, a restricted Boltzmann machine or RBM; see Figure 6). Let \mathbf{b}^H be the bias associated with hidden units, \mathbf{b}^V be the bias associated with visible units, and \mathbf{W} be the weight between visible units and hidden units. Then, given the visible values \mathbf{x} , the hidden values \mathbf{h} are conditionally independent of each other, and we can represent the conditional expected value of the j -th hidden unit as follows:

$$m_j(\mathbf{x}) = \frac{1}{1 + \exp(-b_j(\mathbf{x}))}, \quad (206)$$

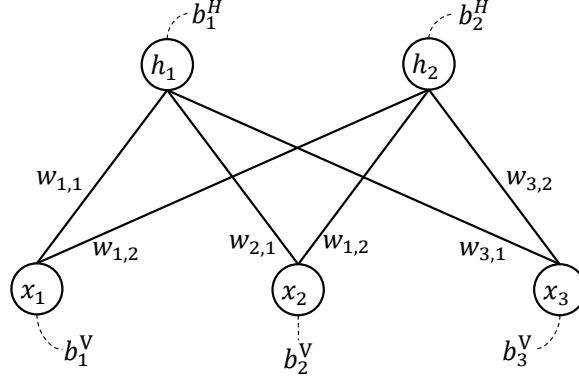


Figure 6: A restricted Boltzmann machine

where

$$b_j(\mathbf{x}) \equiv b_j^H + \mathbf{x}^\top \mathbf{W}_{:,j}. \quad (207)$$

Because the energy is a linear function of \mathbf{h} , we can represent the expected energy, where the expectation is with respect to the conditional distribution of the hidden values \mathbf{H} given the visible values \mathbf{x} , using the conditional expected values of hidden units:

$$\mathbb{E}[E_\theta(\mathbf{x}, \mathbf{H})] = -(\mathbf{b}^V)^\top \mathbf{x} - \mathbf{x}^\top \mathbf{W} \mathbf{m} - (\mathbf{b}^H)^\top \mathbf{m} \quad (208)$$

$$= -(\mathbf{b}^V)^\top \mathbf{x} - \mathbf{b}(\mathbf{x})^\top \mathbf{m}(\mathbf{x}). \quad (209)$$

Notice that the distribution of visible values is then given by

$$\mathbb{P}_\theta(\mathbf{x}) = \frac{1}{1 + \exp\left(-\mathbb{E}[E_\theta(\mathbf{x}, \mathbf{H})]\right)}. \quad (210)$$

This expected energy may be compared against the corresponding free energy:

$$F_\theta(\mathbf{x}) = -\log \sum_{\tilde{\mathbf{h}}} \exp(-E_\theta(\mathbf{x}, \tilde{\mathbf{h}})), \quad (211)$$

where

$$E_\theta(\mathbf{x}, \tilde{\mathbf{h}}) = -(\mathbf{b}^V)^\top \mathbf{x} - (\mathbf{b}^H)^\top \tilde{\mathbf{h}} - \mathbf{x}^\top \mathbf{W} \tilde{\mathbf{h}}. \quad (212)$$

The free energy can be represented as follows:

$$F_\theta(\mathbf{x}) = -\log \exp\left((\mathbf{b}^V)^\top \mathbf{x}\right) \sum_{\tilde{\mathbf{h}}} \exp(\mathbf{b}(\mathbf{x})^\top \tilde{\mathbf{h}}) \quad (213)$$

$$= -(\mathbf{b}^V)^\top \mathbf{x} - \log \prod_{j=1}^M (1 + \exp(b_j(\mathbf{x}))) \quad (214)$$

$$= -(\mathbf{b}^V)^\top \mathbf{x} - \sum_{j=1}^M \log(1 + \exp(b_j(\mathbf{x}))). \quad (215)$$

Theorem 2.

$$\mathbb{E}_\theta[E_\theta(\mathbf{x}, \mathbf{H})] = F_\theta(\mathbf{x}) - \mathbb{E}_\theta[\log \mathbb{P}_\theta(\mathbf{H} | \mathbf{x})], \quad (216)$$

where $-\mathbb{E}_\theta[\log \mathbb{P}_\theta(\mathbf{H} | \mathbf{x})]$ is the entropy of the conditional distribution of hidden values \mathbf{H} given visible values \mathbf{x} .

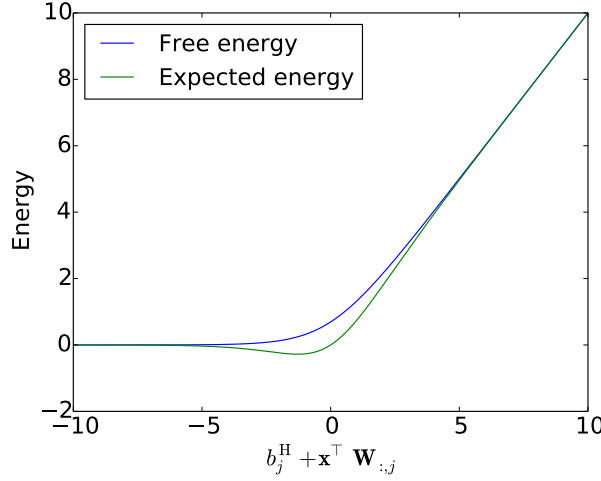


Figure 7: Comparison between expected energy (209) and free energy (215) associated with the j -th hidden unit of a restricted Boltzmann machine.

Proof. By Corollary 1, hidden values are conditionally independent of each other given visible values:

$$\log \mathbb{P}_\theta(\mathbf{H} \mid \mathbf{x}) = \sum_{j=1}^M \log \mathbb{P}_\theta(H_j \mid \mathbf{x}). \quad (217)$$

Then, using the notation in (206), we obtain

$$E_\theta[\log \mathbb{P}_\theta(\mathbf{H} \mid \mathbf{x})] = \sum_{j=1}^M \left(m_j(\mathbf{x}) \log m_j(\mathbf{x}) + (1 - m_j(\mathbf{x})) \log(1 - m_j(\mathbf{x})) \right) \quad (218)$$

$$= \sum_{j=1}^M \left(m_j \log \frac{\exp(b_j(\mathbf{x}))}{1 + \exp(b_j(\mathbf{x}))} + (1 - m_j) \log \frac{1}{1 + \exp(b_j(\mathbf{x}))} \right) \quad (219)$$

$$= \mathbf{b}(\mathbf{x})^\top \mathbf{m}(\mathbf{x}) - \sum_{j=1}^M \log(1 + \exp(b_j(\mathbf{x}))). \quad (220)$$

The theorem now follows by adding (209) and (220), comparing it against (215). \square

Figure 7 compares expected energy and free energy. Specifically, the blue curve shows the value of

$$\frac{b_j^H + \mathbf{x}^\top \mathbf{W}_{:,j}}{1 + \exp(-b_j^H - \mathbf{x}^\top \mathbf{W}_{:,j})}, \quad (221)$$

which appears in the expression of expected energy (209), and the green curve shows the value of

$$\log(1 + \exp(b_j^H + \mathbf{x}^\top \mathbf{W}_{:,j})), \quad (222)$$

which appears in the expression of free energy (215). The difference between the two curves is largest ($\log 2 \approx 0.30$) when $b_j^H + \mathbf{x}^\top \mathbf{W}_{:,j} = 0$. The two curves are essentially indistinguishable when $|b_j^H + \mathbf{x}^\top \mathbf{W}_{:,j}|$ is sufficiently large. This suggests that the Boltzmann machine with expected energy is different from the corresponding Boltzmann machine (with free energy), but the two models have some similarity.

7 Non-probabilistic energy-based models

So far, we have studied probabilistic models. A difficulty that we often face with probabilistic models is in high computational complexity for evaluating the partition function, or the normalization for the total probability to become 1. The probabilistic models that we have studied can be turned into (non-probabilistic) energy-based models. Such energy-based models do not require normalization, which is attractive from computational point of view. A difficulty in energy-based models is in designing appropriate objective functions. In this section, we will study (non-probabilistic) energy-based models through an example [42]. For more details, see a tutorial by LeCun [34].

7.1 Objective functions for energy-based models

In learning an energy-based model, one needs to carefully design an objective (loss) function [34, 35] in a way that minimizing the objective function leads to desired results. We can then optimize the parameters θ of the energy-based model by minimizing the objective function. The energy-based model with the optimized parameters θ should give low energy to desirable values of the variables of the energy-based model and high energy to other values.

Here, we consider energy-based models with input and output. Let $E_\theta(\mathbf{x}, y)$ be the energy for a pair of an input \mathbf{x} and an output y . An energy-based model with parameter θ gives

$$y^* = \underset{y}{\operatorname{argmin}} E_\theta(\mathbf{x}, y) \quad (223)$$

as the output for input \mathbf{x} . A desirable pair of input and output should give lower energy than undesirable ones.

When we optimize an energy-based model, minimizing the energy of a given data is usually inappropriate. In particular, such an objective function may be unbounded. It may not distinguish two patterns, one is good and the other is very good, as both of the two patterns have the minimum energy.

An objective function of an energy-based model should have a contrastive term, which naturally appear in the objective function of a probabilistic model (*i.e.*, the KL divergence or the log likelihood). For example, recall from (4) that the average negative log likelihood of a set of patterns \mathcal{D} with respect to a Boltzmann machine is given by

$$-\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \mathbb{P}_\theta(\mathbf{x}) = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} E_\theta(\mathbf{x}) - \log \sum_{\tilde{\mathbf{x}}} \exp(-E_\theta(\tilde{\mathbf{x}})). \quad (224)$$

The second term of the right-hand side of (224) is a contrastive term. In particular, to minimize this objective function, we should not only reduce the energy of the patterns in \mathbf{D} but also increase the energy of the patterns not in \mathbf{D} . Recall also Figure 3. However, the contrastive term involves the summation over exponentially many (2^N) patterns, and is the source of computational intractability. In designing an objective function for an energy-based model, we want to design a contrastive term that can be more efficiently evaluated. We will see an example in the following.

7.2 An example of face detection with pose estimation

Osadchy et al. study an energy-based approach for classifying images into “face” or “non-face” and estimating the facial pose at the same time [42]. Let \mathbf{x} be a vector representing an image, y be a variable indicating “face” or “non-face,” and \mathbf{z} be a vector representing a facial pose. They consider an energy function of the following form:

$$E_\theta(y, \mathbf{z}, \mathbf{x}) = y \|G_\theta(\mathbf{x}) - F(\mathbf{z})\| + (1 - y) T, \quad (225)$$

where $G_\theta(\cdot)$ is a convolutional neural network (CNN), having parameter θ , that maps an \mathbf{x} into a lower dimensional vector, $F(\cdot)$ is an arbitrarily defined function that maps a \mathbf{z} on to a manifold embedded in the low dimensional space of the output of the CNN, and T is a constant (see Figure 8).

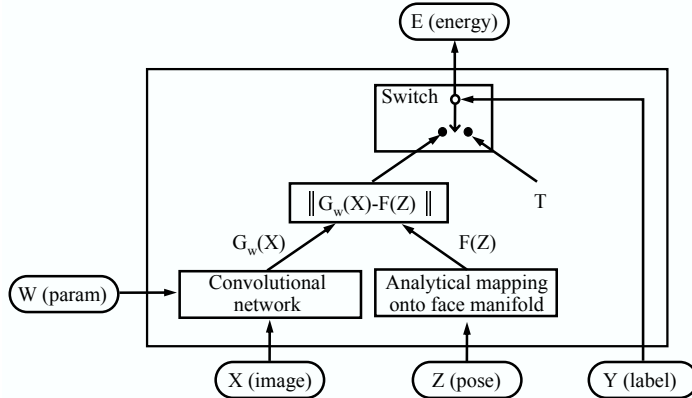


Figure 8: An architecture of the energy (225): Figure 4 from [42], where W is used to denote parameters θ .

An image \mathbf{x} is classified as “face” ($y = 1$) if

$$\min_{\tilde{\mathbf{z}}} E_{\theta}(1, \tilde{\mathbf{z}}, \mathbf{x}) < \min_{\tilde{\mathbf{z}}} E_{\theta}(0, \tilde{\mathbf{z}}, \mathbf{x}), \quad (226)$$

which is equivalent to

$$\min_{\tilde{\mathbf{z}}} \|G_{\theta}(\mathbf{x}) - F(\tilde{\mathbf{z}})\| < T. \quad (227)$$

We want to learn the parameters θ from a given training dataset. A training dataset \mathcal{D} consists of two subsets. The first subset \mathbf{D}_1 includes facial images with their facial poses (\mathbf{z}, \mathbf{x}) . The second subset \mathbf{D}_2 includes non-facial images \mathbf{x} .

It would be computationally intractable to maximize the log-likelihood of the training dataset with respect to the probability distribution that could be defined with the energy (225) through

$$\mathbb{P}_{\theta}(y, \mathbf{z}, \mathbf{x}) \sim \exp(-E_{\theta}(y, \mathbf{z}, \mathbf{x})). \quad (228)$$

Osadchy et al. instead minimizes the following objective function [42]:

$$L_{\theta}(\mathcal{D}) = \frac{1}{|\mathcal{D}_1|} \sum_{(\mathbf{z}, \mathbf{x}) \in \mathcal{D}_1} E_{\theta}(1, \mathbf{z}, \mathbf{x})^2 + \frac{\kappa}{|\mathcal{D}_2|} \sum_{\mathbf{x} \in \mathcal{D}_2} \exp\left(-\min_{\tilde{\mathbf{z}}} E_{\theta}(1, \tilde{\mathbf{z}}, \mathbf{x})\right), \quad (229)$$

where κ is a positive constant (hyper-parameter). The first term of the right-hand side of (229) is the average squared energy of the “face” samples. Hence, minimizing the energy of “face” samples leads to minimizing the objective function. The second (contrastive) term involves the energy of “non-face” samples when they are classified as “face” ($y = 1$), where the face pose is set optimally so that the energy is minimized. By minimizing the second term, one can expect to make the energy of those “non-face” samples with the face label ($y = 1$) greater than the corresponding energy with the non-face label ($y = 0$).

8 Conclusion

We have reviewed basic properties of Boltzmann machines with a particular focus on those that are relevant for gradient-based approaches to learning their parameters. As it turns out that exact learning

is in general intractable, we have discussed general approaches to approximation as well as tractable alternative, namely energy-based models.

This paper has covered limited perspectives of Boltzmann machines and energy-based models. For example, we have only briefly discussed restricted Boltzmann machines (RBMs), which deserves intensive review in its own. In fact, RBMs and deep Boltzmann machines [51] are the topics that are covered in the second part of an IJCAI-17 tutorial². This paper does not cover Boltzmann machines for time-series, which are reviewed in a companion paper [43].

The use of energy is also restrictive in this paper and in line with the tutorial by LeCun [34]. Our perspective on energy-based learning in the tutorial is, however, broader than what is suggested by LeCun. We may use energy for other purposes. An example is the use of free energy [52, 17, 48] or expected energy [10] to approximate the Q-function in reinforcement learning. This energy-based reinforcement learning is the topic covered in the fourth part of the IJCAI-17 tutorial. A key aspect of energy-based reinforcement learning is that the energy used to approximate the Q-function naturally defines a probability distribution, which is used for exploration. Thus, energy-based models that allow efficient sampling have the advantage in sampling the actions to explore in reinforcement learning. There is recent work on the use of deep neural networks to approximate the Q-function in the framework of energy-based reinforcement learning [15], where they use a separate sampling network for efficiency.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR1304, Japan.

References

- [1] Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning (ICML 2017)*, pages 791–798, June 2007.
- [2] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9:147–169, 1985.
- [3] L. Bottou. Online learning and stochastic approximations. In D. Saad, editor, *On-Line Learning in Neural Networks*, chapter 2, pages 9–42. Cambridge University Press, 2009.
- [4] M. Á. Carreira-Perpiñán and G. E. Hinton. On contrastive divergence learning. In *Proceedings of the tenth International Workshop on Artificial Intelligence and Statistics*, pages 33–40, January 2005.
- [5] G. Casella and E. I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.
- [6] A. Courville, J. Bergstra, and Y. Bengio. A spike and slab restricted Boltzmann machine. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 233–241, 2011.
- [7] A. Courville, J. Bergstra, and Y. Bengio. Unsupervised models of images by spike-and-slab RBMs. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1145–1152, 2011.
- [8] Z. Dai, A. Almahairi, P. Bachman, E. Hovy, and A. Courville. Calibrating energy-based generative adversarial networks. *CoRR*, abs/1702.01691, 2017.
- [9] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

²https://researcher.watson.ibm.com/researcher/view_group_up.php?id=7834

- [10] S. Elfving, E. Uchibe, and K. Doya. From free energy to expected energy: Improving energy-based value function approximation in reinforcement learning. *Neural Networks*, 84:17–27, 2016.
- [11] M. Gartrell, U. Paquet, and N. Koenigstein. Low-rank factorization of determinantal point processes. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 1912–1918, February 2017.
- [12] J. A. Gillenwater, A. Kulesza, E. Fox, and B. Taskar. Expectation-maximization for learning determinantal point processes. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3149–3157. Curran Associates, Inc., 2014.
- [13] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.
- [14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.
- [15] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1352–1361, August 2017.
- [16] D. O. Hebb. *The organization of behavior: A neuropsychological approach*. Wiley, 1949.
- [17] N. Heess, D. Silver, and Y. W. Teh. Actor-critic reinforcement learning with energy-based policies. In M. P. Deisenroth, C. Szepesvári, and J. Peters, editors, *Proceedings of the Tenth European Workshop on Reinforcement Learning*, volume 24 of *Proceedings of Machine Learning Research*, pages 45–58, Edinburgh, Scotland, 2013. PMLR.
- [18] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, August 2002.
- [19] G. E. Hinton. Boltzmann machine. *Scholarpedia*, 2(5):1668, 2007.
- [20] G. E. Hinton. Boltzmann machines. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning*. Springer, 2010.
- [21] G. E. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504–507, 2006.
- [22] A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.
- [23] A. Hyvärinen. Some extensions of score matching. *Computational Statistics & Data Analysis*, 51(5):2499–2512, 2007.
- [24] A. Hyvärinen. Optimal approximation of signal priors. *Neural Computation*, 20:3087–3110, 2008.
- [25] E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253?258, 1925.
- [26] T. Kim and Y. Bengio. Deep directed generative models with energy-based probability estimation. *CoRR*, abs/1606.03439, 2016.
- [27] R. Kindermann and J. L. Snell. *Markov random field and their applications*. American Mathematical Society, 1980.
- [28] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, arXiv:1412.6980, 2015.

- [29] D. P. Kingma and Y. LeCun. Regularized estimation of image statistics by score matching. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1126–1134. Curran Associates, Inc., 2010.
- [30] U. Köster, J. T. Lindgren, and A. Hyvärinen. Estimating Markov random field potentials for natural images. In T. Adali T, C. Jutten C, J. M. T. Romano, and A. K. Barros, editors, *Independent Component Analysis and Signal Separation. ICA 2009. Lecture Notes in Computer Science, vol. 5441*. Springer, 2009.
- [31] A. Krizhevsky. Learning multiple layers of features from tiny images. Master’s thesis, Computer Science Department, University of Toronto, Toronto, Canada, 2009.
- [32] A. Kulesza and B. Taskar. *Determinantal Point Processes for Machine Learning*. Now Publishers Inc., Hanover, MA, USA, 2012.
- [33] H. Larochelle and Y. Bengio. Classification using discriminative restricted Boltzmann machines. In *Proceedings of the 25th international conference on Machine learning (ICML 2008)*, pages 536–543, 2008.
- [34] Y. LeCun, S. Chopra, R. Hadsell, M. A. Ranzato, and F. J. Huang. *A Tutorial on Energy-Based Learning*. MIT Press, 2006.
- [35] Y. LeCun and F. J. Huang. Loss functions for discriminative training of energy-based models. In *Proceedings of the 10-th International Workshop on Artificial Intelligence and Statistics*, pages 206–213, 2005.
- [36] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, first edition, 2008.
- [37] S. Löwel and W. Singer. Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity. *Science*, 255:209–212, 1992.
- [38] S. Lyu. Interpretation and generalization of score matching. In *Proceedings of the 25th Conference in Uncertainty in Artificial Intelligence (UAI09)*, pages 359–366, June 2009.
- [39] O. Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122, 1975.
- [40] Z. Mariet and S. Sra. Kronecker determinantal point processes. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2694–2702. Curran Associates, Inc., 2016.
- [41] T. Marks and J. Movellan. Diffusion networks, products of experts, and factor analysis. In *Proceedings of the Third International Conference on Independent Component Analysis and Blind Source Separation*, 2001.
- [42] M. Osadchy, Y. LeCun, and M. Miller. Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research*, 8(1197–1215), 2007.
- [43] T. Osogami. Boltzmann machines for time-series. Technical Report RT0980, IBM Research - Tokyo, 2017.
- [44] T. Osogami, H. Kajino, and T. Sekiyama. Bidirectional learning for time-series models with hidden units. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, pages 2711–2720, August 2017.
- [45] T. Osogami and M. Otsuka. Restricted Boltzmann machines modeling human choice. In *Advances in Neural Information Processing Systems 27*, pages 73–81. 2014.

- [46] T. Osogami, R. Raymond, A. Goel, T. Shirai, and T. Maehara. Dynamic determinantal point processes. 2017.
- [47] M. Otsuka and T. Osogami. A deep choice model. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 850–856, January 2016.
- [48] M. Otsuka, J. Yoshimoto, and K. Doya. Free-energy-based reinforcement learning in a partially observable environment. In *Proceedings of the European Symposium on Artificial Neural Networks — Computational Intelligence and Machine Learning*, pages 28–30, April 2010.
- [49] C. Peterson and J. Anderson. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1(5):995–1019, 1987.
- [50] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks: The Official Journal of the International Neural Network Society*, 12(1):145–151, 1999.
- [51] R. R. Salakhutdinov and G. E. Hinton. Deep Boltzmann machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS-09)*, pages 448–455, April 2009.
- [52] B. Sallans and G. E. Hinton. Using free energies to represent Q-values in a multiagent reinforcement learning task. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 1075–1081. MIT Press, 2001.
- [53] J. Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.
- [54] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.
- [55] T. J. Sejnowski. Higher-order Boltzmann machines. *American Institute of Physics, Conference Proceedings*, 151(1):398–403, 1986.
- [56] Tomoyuki Shirai and Yoichiro Takahashi. Random point fields associated with certain Fredholm determinants. I. Fermion, Poisson and boson point processes. *Journal of Functional Analysis*, 205(2):414–463, 2003.
- [57] Tomoyuki Shirai and Yoichiro Takahashi. Random point fields associated with certain Fredholm determinants. II. Fermion shifts and their ergodic and Gibbs properties. *The Annals of Probability*, 31(3):1533–1564, 2003.
- [58] I. Sutskever and G. E. Hinton. Learning multilevel distributed representations for high-dimensional sequences. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07)*, volume 2, pages 548–555. Journal of Machine Learning Research - Proceedings Track, 2007.
- [59] I. Sutskever, G. E. Hinton, and G. W. Taylor. The recurrent temporal restricted Boltzmann machine. In *Advances in Neural Information Processing Systems 21*, pages 1601–1608. December 2008.
- [60] K. Swersky, M. Ranzato, D. Buchman, B. M. Marlin, and N. Freitas. On autoencoders and score matching for energy based models. In *Proceedings of the 28th International Conference on Machine Learning (ICML 2011)*, pages 1201–1208, June 2011.
- [61] T. Tieleman and G. E. Hinton. Lecture 6.5—Rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

- [62] P. Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.
- [63] M. Welling and G. E. Hinton. A new learning algorithm for mean field Boltzmann machines. In *Proceedings of the 12th International Conference on Artificial Neural Networks*, pages 351–357, August 2002.
- [64] M. Welling, M. Rosen-Zvi, and G. E. Hinton. Exponential family harmoniums with an application to information retrieval. In *Advances in Neural Information Processing Systems 17*, pages 1481–1488. 2004.
- [65] M. Welling, R. S. Zemel, and G. E. Hinton. Self supervised boosting. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 681–688. MIT Press, 2003.
- [66] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang. Deep structured energy based models for anomaly detection. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, pages 1100–1109, June 2016.
- [67] J. J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *CoRR*, abs/1609.03126, 2016.