WORKING TITLE: Maximum Entropy models, Restricted Boltzmann Machines, and the whole shebag

Tor Erlend Fjelde

July 5, 2018

Contents

1 Introduction

Maximum Entropy (MaxEnt) models are parametrizable probability distributions which are constructed from application of the Principle of Maximum Entropy. In general, a MaxEnt model of a discrete random variable is the distribution given by

$$p(x) = \frac{1}{Z}\exp(-E(x))$$

where

$$E(x) = -\sum_{i=1}^{n} \lambda_i f_i(x)$$

$$\lambda_i \in \mathbb{R} \quad \text{such that} \quad \frac{\partial Z}{\partial \lambda_i} = F_i, \quad i = 1, \ldots, n$$

with F_i corresponding to the constraints enforced when applying the Principle of Maximum Entropy, whose values are the parameters of the model.

These kind of models, being closely related to the concept of *information entropy*, sees frequent use in multiple areas (e.g. statistcal physics, natural language processing), sometimes under names such as Boltzmann distributions in the physics literature or Markov Random Fields in the graphical modelling literature.

One class of such models are called **Boltzmann machines**. These were first introduced as a method for learning unknown "soft" constraints within systems, that is, learning the F_i in the expression above of a distribution from the data. [1] Due to the computational complexity of the Boltzmann machines, **Restricted Boltzmann machines (RBMs)** were introduced, for which tractable approximate learning schemes are more easily obtainable. In this paper we have a closer look at RBMs, theoretical justifications and the different methods used for training.

2 Background

2.1 Maximum Entropy Models

2.2 Boltzmann machines

2.3 Sampling methods

2.3.1 Metropolis-Hastings

2.3.2 Gibbs sampling

3 Restricted Boltzmann Machines

3.1 Definition

Definition 3.1. A **Restricted Boltzmann Machine (RBM)** is an *energy-based model* consisting of a set of *hidden* units $\mathcal{H} = \{H_\mu\}$ and a set of *visible* units $\mathcal{V} = \{V_j\}$, whereby "units" we mean random variables, taking on the values **h** and **v**, respectively. The *restricted* part of the name comes from the fact that we assume independence between the hidden units and the visible units, i.e.

$$p(h_\mu \mid h_1, \ldots, h_{\mu-1}, h_{\mu+1}, \ldots, h_{|\mathcal{H}|}) = p(h_\mu)$$
$$p(v_j \mid v_1, \ldots, v_{j-1}, v_{j+1}, \ldots, v_{|\mathcal{V}|}) = p(v_j)$$

An **RBM** therefore assumes the following joint probability distribution of the visible and hidden units:

$$p(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\tilde{p}(\mathbf{v},\mathbf{h}) = \exp\Big(-E(\mathbf{v},\mathbf{h})\Big)$$

with Z being the partition function (normalization factor), $\tilde{p}$ denoting the unnormalized density, and the energy function is given by

$$\begin{aligned} E(\mathbf{v},\mathbf{h}) &= -\mathbf{c}^T\mathbf{v} - \mathbf{b}^T\mathbf{h} - \mathbf{v}^T\mathbf{W}\mathbf{h} \\ &= -c_j v_j - b_\mu h_\mu - v_j W_{j\mu} h_\mu \end{aligned}$$

implicitly summing over repeating indices.

3.2 Log-likelihood

From Definition 3.1, we have

$$\log p(\mathbf{v}) = \log\Big(\sum_{\mathbf{h}} \exp\Big(-E(\mathbf{v},\mathbf{h})\Big)\Big) - \log Z$$

Now suppose we're given a set of samples $\{\mathbf{v}^{(n)}, n = 1,\dots,N\}$, then the likelihood (assuming i.i.d. of the $\mathbf{v}^{(n)}$) is given by

$$p\Big(\{c_j, b_\mu, W_{j\mu}\} \mid \{\mathbf{v}^{(n)}\}\Big) = \prod_{n=1}^{N} p\Big(\mathbf{v}^{(n)} \mid \{c_j, b_\mu, W_{j\mu}\}\Big)$$

Taking the log of this expression, and substituing in the above expression for $\log p(\mathbf{v})$, we have

$$\begin{aligned} \mathcal{L}(\{c_j, b_\mu, W_{j\mu}\}) &= \sum_{n=1}^{N}\left[\log\Big(\sum_{\mathbf{h}}\tilde{p}\Big(\mathbf{v}^{(n)},\mathbf{h}\Big)\Big) - \log Z\right] \\ &= \sum_{n=1}^{N}\left[\log\Big(\sum_{\mathbf{h}}\tilde{p}\Big(\mathbf{v}^{(n)},\mathbf{h}\Big)\Big)\right] - N\log Z \end{aligned}$$

Let $\theta \in \{c_j, b_\mu, W_{j\mu}\}$, taking the partial derivative wrt. θ for the n-th term we have

$$\begin{aligned} \frac{\partial}{\partial\theta}\left[\log\Big(\sum_{\mathbf{h}}\tilde{p}\Big(\mathbf{v}^{(n)},\mathbf{h}\Big)\Big) - \log Z\right] &= -\frac{\sum_{\mathbf{h}}\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta}\tilde{p}\Big(\mathbf{v}^{(n)},\mathbf{h}\Big)}{\sum_{\mathbf{h}}\tilde{p}\Big(\mathbf{v}^{(n)},\mathbf{h}\Big)} \\ &\quad - \frac{1}{Z}\frac{\partial Z}{\partial\theta} \end{aligned}$$

The first term can be written as an expectation

$$\frac{\sum_{\mathbf{h}}\frac{\partial E(\mathbf{v}^{(n)},\mathbf{h})}{\partial\theta}\tilde{p}\Big(\mathbf{v}^{(n)},\mathbf{h}\Big)}{\sum_{\mathbf{h}}\tilde{p}\Big(\mathbf{v}^{(n)},\mathbf{h}\Big)} = \mathbb{E}\left[\frac{\partial E(\mathbf{v}^{(n)},\mathbf{h})}{\partial\theta} \;\middle|\; \mathbf{v}^{(n)}\right]$$

since on the left-hand side (LHS) we're marginalizing over all $\mathbf{h}$ and then normalizing wrt. the same distribution we just summed over. For the second term recall that $Z = \sum_{\mathbf{v},\mathbf{h}}\exp\Big(-E(\mathbf{v},\mathbf{h})\Big)$, therefore

$$\frac{1}{Z}\frac{\partial Z}{\partial\theta} = -\frac{1}{Z}\sum_{\mathbf{v},\mathbf{h}}\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta}\exp\Big(-E(\mathbf{v},\mathbf{h})\Big) = -\mathbb{E}\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta}\right]$$

Substituting it all back into the partial derivative of the log-likelihood, we end get

$$\frac{\partial\mathcal{L}}{\partial\theta} = -\sum_{n=1}^{N}\mathbb{E}\left[\frac{\partial E(\mathbf{v}^{(n)},\mathbf{h})}{\partial\theta} \;\middle|\; \mathbf{v}^{(n)}\right] + N\,\mathbb{E}\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta}\right]$$

Where the expectations are all over the probability distribution defined by the *model*. Since maximizing $\mathcal{L}$ is equivalent to maximizing $\mathcal{L}/N$ we instead consider the expression in Lemma 3.1.

Lemma 3.1. *Given a set of i.i.d. drawn samples $\{\mathbf{v}^{(n)}, n = 1,\dots,N\}$, the gradient of the log-likelihood is given by*

$$\frac{1}{N}\frac{\partial\mathcal{L}}{\partial\theta} = -\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[\frac{\partial E(\mathbf{v}^{(n)},\mathbf{h})}{\partial\theta} \;\middle|\; \mathbf{v}^{(n)}\right] + \mathbb{E}\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta}\right]$$

where the expectations are taken wrt. the RBM, as defined in Definition 3.1

Observe that the first term in Lemma 3.1 can be written

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[\frac{\partial E(\mathbf{v}^{(n)},\mathbf{h})}{\partial\theta} \;\middle|\; \mathbf{v}^{(n)}\right] = \left\langle\mathbb{E}\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} \;\middle|\; \mathbf{v}\right]\right\rangle_{\{\mathbf{v}^{(n)}\}}$$

where we use angular brackets $\langle\cdot\rangle$ to denote the *empirical* expectation over the data $\{\mathbf{v}^{(n)}\}$. Then,

$$\frac{1}{N}\frac{\partial\mathcal{L}}{\partial\theta} = -\left\langle\mathbb{E}\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} \;\middle|\; \mathbf{v}\right]\right\rangle_{\{\mathbf{v}^{(n)}\}} + \mathbb{E}\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta}\right] \tag{1}$$

3.2.1 Approximating the log-likelihood gradient

In general, the second term in Eq. 1 is clearly intractable, as we would have to sum over all possible $\mathbf{v}$ and $\mathbf{h}$. The first term should be less computationally expensive, due to only having to sum over all *observed* $\mathbf{v}$, rather than all possible $\mathbf{v}$. Nonetheless this might also be intractable as we would still have to marginalize over all the hidden states $\mathbf{h}$ with $\mathbf{v}$ fixed to $\mathbf{v}^{(n)}$. In general we need to also sample $\mathbf{h}$ to obtain the conditional expectation. For example we could sample M hidden states $\mathbf{h}$ for each observed $\mathbf{v}^{(n)}$, i.e.

$$\begin{aligned} \left\langle\mathbb{E}\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} \;\middle|\; \mathbf{v}\right]\right\rangle_{\{\mathbf{v}^{(n)}\}} &= \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[\frac{\partial E(\mathbf{v}^{(n)},\mathbf{h})}{\partial\theta} \;\middle|\; \mathbf{v}^{(n)}\right] \\ &\approx \frac{1}{N}\sum_{n=1}^{N}\frac{1}{M}\sum_{m=1}^{M}\frac{\partial E(\mathbf{v}^{(n)},\mathbf{h}^{(m)})}{\partial\theta}, \\ &\qquad \text{with } \mathbf{h}^{(m)} \sim p(\mathbf{h}) \end{aligned}$$

This still leaves the problem of sampling from $p(\mathbf{h})$, which is not necessarily known. Most often $M = 1$ is used, corresponding to Gibbs sampling.[2] As we will see later (see Remark **??**), in certain cases, the first term can in fact be computed analytically.

Going forward we will use the following notation

$$\left\langle \mathbb{E}\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \;\middle|\; \mathbf{v} \right] \right\rangle_{\{\mathbf{v}^{(n)}\}} = \left\langle \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right\rangle_{\text{data}}$$

which is often used in the literature.[2, 3] The $\{\mathbf{v}^{(n)}\}$ was used to make explicit that we're computing the empirical expectation conditioned on visible units over the *visible* units, *not* the joint expectation over $p(\mathbf{v},\mathbf{h})$.

It might also be worth noting that the second expectation in Eq. 1 can also be written as an expectation of the *conditional* expectation over $\mathbf{h} \mid \mathbf{v}$, i.e.

$$\begin{aligned}
\mathbb{E}\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right] &= \sum_{\mathbf{v},\mathbf{h}} p(\mathbf{v},\mathbf{h}) \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \\
&= \sum_{\mathbf{v},\mathbf{h}} p(\mathbf{h} \mid \mathbf{v}) p(\mathbf{v}) \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \\
&= \sum_{\mathbf{v}} p(\mathbf{v}) \sum_{\mathbf{h}} p(\mathbf{h} \mid \mathbf{v}) \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \;\middle|\; \mathbf{v} \right] \right]
\end{aligned}$$

This is useful to know later on when we want to approximate this expectation.

The approximation to the gradient of the log-likelihood can then be written

$$\frac{1}{N} \frac{\partial \mathcal{L}}{\partial \theta} \approx -\left\langle \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right\rangle_{\text{data}} + \left\langle \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right\rangle_{\text{model}} \tag{2}$$

Making θ explicit in Eq. 2 with an RBM from Definition 3.1, and approximating these expectations using *empirical* estimates, we have

$$\begin{aligned}
\frac{1}{N} \frac{\partial \mathcal{L}}{\partial c_j} &= \langle v_j \rangle_{\text{data}} - \langle v_j \rangle_{\text{model}} \\
\frac{1}{N} \frac{\partial \mathcal{L}}{\partial b_\mu} &= \langle h_\mu \rangle_{\text{data}} - \langle h_\mu \rangle_{\text{model}} \\
\frac{1}{N} \frac{\partial \mathcal{L}}{\partial W_{j\mu}} &= \langle v_j h_\mu \rangle_{\text{data}} - \langle v_j h_\mu \rangle_{\text{model}}
\end{aligned} \tag{3}$$

Observe that the signs have switched compared to Eq. 2, which is simply due to the fact that $E(\mathbf{v},\mathbf{h})$ depends negatively on all the variables $\{v_j, h_\mu, W_{j\mu}\}$.

We will now see how we can in fact produce these empirical estimates for the different quantities used in computation of the gradients as seen above.

4 Training

In this section we will only consider the standard Bernoulli RBMs, which assume both the hidden and visible variables are Bernoulli random variables, only taking on values in $\{0,1\}$. As we will see in Section 6, we can extend the methods used for Bernoulli RBMs quite easily to different types of RBMs.

4.1 Approximating the log-likelihood gradient

Suppose now that both the visible and hidden units are Bernoulli random variables, i.e. only taking on values $\{0,1\}$. We then have

$$\mathbb{E}\big[\mathbf{h} \mid \mathbf{v}\big] = \prod_{\mu=1}^{|\mathcal{H}|} p\big(H_\mu = 1 \mid \mathbf{v}\big)$$

Substituing this into Eq. 2 gives us a much simpler expression for the gradients:

$$\begin{aligned}
\frac{1}{N} \frac{\partial \mathcal{L}}{\partial c_j} &= \frac{1}{N} \sum_{n=1}^{N} v_j^{(n)} - \mathbb{E}[v_j] \\
\frac{1}{N} \frac{\partial \mathcal{L}}{\partial b_\mu} &= \frac{1}{N} \sum_{n=1}^{N} p(H_\mu = 1 \mid \mathbf{v}^{(n)}) \\
&\quad - \mathbb{E}\big[p(H_\mu = 1 \mid \mathbf{v})\big] \\
\frac{1}{N} \frac{\partial \mathcal{L}}{\partial W_{j\mu}} &= \frac{1}{N} \sum_{n=1}^{N} v_j p\big(H_\mu = 1 \mid \mathbf{v}^{(n)}\big) \\
&\quad - \mathbb{E}\Big[v_j p(H_\mu = 1 \mid \mathbf{v})\Big]
\end{aligned} \tag{4}$$

since

$$\mathbb{E}\Big[\mathbb{E}[h_\mu \mid \mathbf{v}]\Big] = \mathbb{E}[p(H_\mu = 1 \mid \mathbf{v})]$$

and similarily for the other terms. In these equations, the first terms corresponding to the empirical expectations over the data are now tractable, but we still have terms involving expectations over all possible $\mathbf{v}$, which remain intractable.[1]

To address this, we decide to approximate these expectations by sampling $\mathbf{v}$. By the Law of Large numbers: [5]

$$\mathbb{E}[f(x)] \overset{p}{=} \lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^{M} f\big(x^{(m)}\big)$$

Hence, we can approximate the gradients in Eq. 4 by

$$\begin{aligned}
\frac{1}{N} \frac{\partial \mathcal{L}}{\partial c_j} &= \frac{1}{N} \sum_{n=1}^{N} v_j^{(n)} - \frac{1}{M} \sum_{m=1}^{M} \hat{\mathbf{v}}^{(m)} \\
\frac{1}{N} \frac{\partial \mathcal{L}}{\partial b_\mu} &= \frac{1}{N} \sum_{n=1}^{N} p(H_\mu = 1 \mid \mathbf{v}^{(n)}) \\
&\quad - \frac{1}{M} \sum_{m=1}^{M} p\big(H_\mu = 1 \mid \hat{\mathbf{v}}^{(m)}\big) \\
\frac{1}{N} \frac{\partial \mathcal{L}}{\partial W_{j\mu}} &= \frac{1}{N} \sum_{n=1}^{N} v_j p\big(H_\mu = 1 \mid \mathbf{v}^{(n)}\big) \\
&\quad - \frac{1}{M} \sum_{m=1}^{M} \hat{v}_j^{(m)} p\big(H_\mu = 1 \mid \hat{\mathbf{v}}^{(m)}\big)\Big]
\end{aligned} \tag{5}$$

[1] It is worth noting that in some cases it is advised to also sample $\mathbf{h}$ to approximate the expectation, rather than using the $\mathbb{E}[h_\mu \mid \mathbf{v}] = p(H_\mu = 1 \mid \mathbf{v})$.[4] But in general using $p(H_\mu = 1 \mid \mathbf{v})$ and only sample $\mathbf{v}$ is empirically shown to reduce variance of the approximation.[2]

where we assume $\hat{\mathbf{v}}^{(m)}$ to be M i.i.d. samples drawn from the *model*. It is mostly common to let $M = N$, i.e. draw one sample from the model for each sample of data.[2]

The gradients in Eq. 5 are tractable to compute, as long as drawing i.i.d. samples of $\mathbf{v}$ is also tractable.

For the purpose of readability, let us consider each term of the gradient of $\mathcal{L}$ separately and $M = N$, i.e.

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial c_j}\Big|_n &= v_j^{(n)} - \hat{v}_j^{(k)} \\
\frac{\partial \mathcal{L}}{\partial b_\mu}\Big|_n &= p(H_\mu = 1 \mid \mathbf{v}^{(n)}) - p\big(H_\mu = 1 \mid \hat{\mathbf{v}}^{(k)}\big) \\
\frac{\partial \mathcal{L}}{\partial W_{j\mu}}\Big|_n &= v_j p\big(H_\mu = 1 \mid \mathbf{v}^{(n)}\big) - \hat{v}_j^{(k)} p\big(H_\mu = 1 \mid \hat{\mathbf{v}}^{(k)}\big)
\end{aligned} \tag{6}$$

One way to obtain these samples is to run a Gibbs sampler on the machine for some k number of steps. This results in a *sequence* $\big(\tilde{\mathbf{v}}^{(t)}, t = 1, \ldots, k\big)$ of correlated samples from the RBM. With sufficient large k, the empirical distribution of these $\tilde{\mathbf{v}}^{(t)}$ will converge to the distribution $p(\mathbf{v})$ as wanted. It therefore makes sense to take the k-th (last sample in the sequence) as our sample, and use this as our $\hat{\mathbf{v}}^{(n)}$. Then Eq. 6 becomes

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial c_j}\Big|_n &= v_j^{(n)} - \tilde{v}_j^{(k)} \\
\frac{\partial \mathcal{L}}{\partial b_\mu}\Big|_n &= p(H_\mu = 1 \mid \mathbf{v}^{(n)}) - p\big(H_\mu = 1 \mid \tilde{\mathbf{v}}^{(k)}\big) \\
\frac{\partial \mathcal{L}}{\partial W_{j\mu}}\Big|_n &= v_j p\big(H_\mu = 1 \mid \mathbf{v}^{(n)}\big) - \tilde{v}_j^{(k)} p\big(H_\mu = 1 \mid \tilde{\mathbf{v}}^{(k)}\big)
\end{aligned} \tag{7}$$

where $\tilde{\mathbf{v}}^{(k)}$ is the last sample in the chain.

Eq. 7 is the foundation of the most of the algorithms being used to train RBMs. On its own, Eq. 7 is of course a possible learning scheme, but as we will see one can improve this further my making some changes to the sampling process.

4.2 Contrastive Divergence (CD)

From Eq. 7, understanding **Contrastive Divergence (CD)**, the most commonly used algorithm for training RBMs, becomes trivial.[1] Normally, one would initialize the sampler randomly and then sample a chain of k steps. The only difference with CD, is that we instead initialize the sampler using *one of the data samples* $\mathbf{v}^{(n)}$. We will refer to Contrastive Divergence with k steps to draw each sample as **CD-k**. The full procedure can be seen in Algorithm 3.

Data: Observations of the visible units: $\{\mathbf{v}^{(n)}, n = 1, \ldots, N\}$
Result: Estimated parameters for an RBM: $(\mathbf{b}, \mathbf{c}, \mathbf{W})$
$b_j := 0$ for $j = 1, \ldots, |\mathcal{V}|$
$c_\mu := 0$ for $\mu = 1, \ldots, |\mathcal{H}|$
$W_{j\mu} \sim \mathcal{N}(0, 0.01)$ for $(j, \mu) \in \{1, \ldots, |\mathcal{V}|\} \times \{1, \ldots, |\mathcal{H}|\}$
while *not converged* **do**
  $\Delta b_j := 0$
  $\Delta c_\mu := 0$
  $\Delta W_{j\mu} := 0$
  for $n = 1, \ldots, N$ **do**
    `// initialize sampling procedure`
    $\hat{\mathbf{v}} := \mathbf{v}^{(n)}$
    `// sample using Gibbs sampling`
    for $t = 1, \ldots, k$ **do**
      $\hat{\mathbf{h}} \sim p(\mathbf{h} \mid \hat{\mathbf{v}})$
      $\hat{\mathbf{v}} \sim p(\mathbf{v} \mid \hat{\mathbf{h}})$
    end
    `// accumulate changes`
    $\Delta b_j \leftarrow \Delta b_j + v_j^{(n)} - \hat{v}_j$
    $\Delta c_\mu \leftarrow \Delta c_\mu + p\big(H_\mu = 1 \mid \mathbf{v}^{(n)}\big) - p\big(H_\mu = 1 \mid \hat{\mathbf{v}}\big)$
    $\Delta W_{j\mu} \leftarrow \Delta W_{j\mu} + v_j^{(n)} p\big(H_\mu = 1 \mid \mathbf{v}^{(n)}\big) - \hat{v}_j p\big(H_\mu = 1 \mid \hat{\mathbf{v}}\big)$
  end
  `// update the parameters of the RBM using average gradient`
  $b_j \leftarrow b_j + \frac{\Delta b_j}{N}$
  $h_\mu \leftarrow h_\mu + \frac{\Delta h_\mu}{N}$
  $W_{j\mu} \leftarrow W_{j\mu} + \frac{\Delta W_{j\mu}}{N}$
end

Algorithm 1: Contrastive Divergence (CD-k) with k sampling steps.

4.2.1 Persistent Contrastive Divergence (PCD-k)

A slightly different version of CD-k is **Persistent Contrastive Divergence**, where instead of initializing the sampler by $\mathbf{v}^{(n)}$ for each n, we don't re-initialize the sampler at all between data samples. Instead we use the final state of the sampling chain from when we sampled for, say $\mathbf{v}^{(n')}$, to initialize the sampler when sampling for the next data point, say $\mathbf{v}^{(n)}$. The full procedure can be seen

[2] In these cases, with $M = N$, we usually also make use of batch training with Stochastic Gradient Ascent over multiple epochs. This way we end up performing this sampling procedure once for each data sample *per epoch*, thus ending up with more than a single sample drawn from the model per data sample. In addition these samples are spread out over the training procedure, and so one would expect the samples in the later epochs are more accurate than those in the earlier epochs.

in Algorithm 2.

Data: Observations of the visible units: $\{\mathbf{v}^{(n)}, n = 1, \dots, N\}$
Result: Estimated parameters for an RBM: $(\mathbf{b}, \mathbf{c}, \mathbf{W})$
$b_j := 0$ for $j = 1, \dots, |\mathcal{V}|$
$c_\mu := 0$ for $\mu = 1, \dots, |\mathcal{H}|$
$W_{j\mu} \sim \mathcal{N}(0, 0.01)$ for $(j, \mu) \in \{1, \dots, |\mathcal{V}|\} \times \{1, \dots, |\mathcal{H}|\}$
while *not converged* **do**
  $\Delta b_j := 0$
  $\Delta c_\mu := 0$
  $\Delta W_{j\mu} := 0$
  `// initialize sampling procedure BEFORE training loop`
  $\hat{\mathbf{v}} := \mathbf{v}^{(n)}$ for some $n \in \{1, \dots, N\}$
  for $n = 1, \dots, N$ **do**
    `// sample using Gibbs sampling`
    `// using final sample from previous chain as starting point`
    for $t = 1, \dots, k$ **do**
      $\hat{\mathbf{h}} \sim p(\mathbf{h} \mid \hat{\mathbf{v}})$
      $\hat{\mathbf{v}} \sim p(\mathbf{v} \mid \hat{\mathbf{h}})$
    end
    `// accumulate changes`
    $\Delta b_j \leftarrow \Delta b_j + v_j^{(n)} - \hat{v}_j$
    $\Delta c_\mu \leftarrow \Delta c_\mu + p\big(H_\mu = 1 \mid \mathbf{v}^{(n)}\big) - p\big(H_\mu = 1 \mid \hat{\mathbf{v}}\big)$
    $\Delta W_{j\mu} \leftarrow \Delta W_{j\mu} + v_j^{(n)} p\big(H_\mu = 1 \mid \mathbf{v}^{(n)}\big) - \hat{v}_j p\big(H_\mu = 1 \mid \hat{\mathbf{v}}\big)$
  end
  `// update the parameters of the RBM using average gradient`
  $b_j \leftarrow b_j + \frac{\Delta b_j}{N}$
  $h_\mu \leftarrow h_\mu + \frac{\Delta h_\mu}{N}$
  $W_{j\mu} \leftarrow W_{j\mu} + \frac{\Delta W_{j\mu}}{N}$
end

Algorithm 2: Persistent Contrastive Divergence (PCD-k) with k steps. Initialize next chain from final state of previous chain.

4.3 Parallel Tempering (PT)

As seen in Section 3.2.1 we can approximate the expectations by sampling from the model using MCMC-based methods. As mentioned, these sampling methods does not necessarily produce i.i.d. samples, and thus worse gradient-approximations. In Section 4.2 therefore we considered different schemes for initializing the sampler, in attempt to improve the quality of the samples.

Parallel Tempering (PT) is a method which attempts to improve the sample-quality, not by changing the initialization scheme, but by altering the sampling procedure itself.[2] It's a method heavily inspired by *Simulated Annealing (SA)*, which is heuristic used with MCMC samplers to deal with isolated modes in the target distribution.[6] In PT we introduce supplementary Gibbs chains from a *sequence of distributions* $(p_1, p_2, \dots, p_R)$, where $R \in \mathbb{N}$, such that

$$p_1 = p \quad \text{and} \quad p_r = f(p_r, p_{r-1})p_{r-1}, \quad r = 2, \dots, R$$

where p is the target distribution and f is a *transition probabilitiy* between p_r and p_{r-1}. This is often viewed as two different basins of temperatures T_r and T_{r-1}, which exchange particles with probability $f(p_r, p_{r-1})$.

In the case of RBMs we can easily construct these *tempered distributions* by letting $1 = T_1 < T_2 < \cdots < T_R$ and

$$p_r(\mathbf{v}, \mathbf{h}) = \frac{1}{Z_r} \exp\left(\frac{1}{T_r} E(\mathbf{v}, \mathbf{h})\right)$$

where Z_r is the partition function corresponding to p_r. Clearly

$$p(\mathbf{v}, \mathbf{h}) = p_1(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp\Big(E(\mathbf{v}, \mathbf{h})\Big)$$

as wanted. First we sample form each of these tempered distributions using Gibbs sampling, producing the sequence of samples $\Big((\mathbf{v}_1, \mathbf{h}_1), \dots, (\mathbf{v}_R, \mathbf{h}_R)\Big)$. We then *swap* the samples $(\mathbf{v}_{r-1}, \mathbf{h}_{r-1})$ and $(\mathbf{v}_r, \mathbf{h}_r)$ with probability

$$A\Big((\mathbf{v}_{r-1}, \mathbf{h}_{r-1}), (\mathbf{v}_r, \mathbf{h}_r)\Big) = \min\left\{1, \frac{p_r(\mathbf{v}_{r-1}, \mathbf{h}_{r-1})p_{r-1}(\mathbf{v}_r, \mathbf{h}_r)}{p_r(\mathbf{v}_r, \mathbf{h}_r)p_{r-1}(\mathbf{v}_{r-1}, \mathbf{h}_{r-1})}\right\}$$

which in the case of RBMs is,

$$\begin{aligned}&\frac{p_r(\mathbf{v}_{r-1}, \mathbf{h}_{r-1})p_{r-1}(\mathbf{v}_r, \mathbf{h}_r)}{p_r(\mathbf{v}_r, \mathbf{h}_r)p_{r-1}(\mathbf{v}_{r-1}, \mathbf{h}_{r-1})} \\ &= \exp\left[\left(\frac{1}{T_r} - \frac{1}{T_{r-1}}\right)\Big(E(\mathbf{v}_r, \mathbf{h}_r) - E(\mathbf{v}_{r-1}, \mathbf{h}_{r-1})\Big)\right]\end{aligned}$$

potentially producing, with probability $A\Big((\mathbf{v}_{r-1}, \mathbf{h}_{r-1}), (\mathbf{v}_r, \mathbf{h}_r)\Big)$, the new sequence

$$\Big((\mathbf{v}_1, \mathbf{h}_1), \dots, (\mathbf{v}_r, \mathbf{h}_r), (\mathbf{v}_{r-1}, \mathbf{h}_{r-1}), \dots, (\mathbf{v}_R, \mathbf{h}_R)\Big)$$

We perform these swaps $R - 1$ times, starting with p_{R-1}, p_R, finishing with p, p_2. Finally, we use the samples which are now in the first position as the samples from p.

From Section 2.3.1 we can recognize this expression as the Metropolis-Hastings acceptance ratio with transition probabilities f_r and f_{r-1}, and this produces a chain which guarantees the detailed balance.[6, 2]

Even though there is no rigorous theoretical founding for why one would want to do this, one can intuitively see that as T_r becomes larger, p_r becomes more similar to the uniform distribution. This then leads to a larger sample-variance, thus increasing the *mixing rate* of the Gibbs chain. This is also seen empirical by Neal.[6] Hence, we're left with less biased samples, and thus the

gradient approximations from Eq. 7 become less biased. Of course this comes at a cost of more computation.

In Section 7.1 we compare CD-k and single-step PT with k tempered distributions on some toy-problems. Both these method instances ought to have approximately the same computational cost, with a slightly higher memory usage by PT.

Data: Temperatures: $\{1, T_2, \ldots, T_R\}$
Result: Samples: $\{(\hat{\mathbf{v}}^{(1)}, \hat{\mathbf{h}}^{(1)}), \ldots, (\hat{\mathbf{v}}^{(M)}, \hat{\mathbf{h}}^{(M)})\}$
for $m = 1, \ldots, M$ **do**
  for $r = 1, \ldots, R$ **do**
    `// sample using Gibbs sampling`
    for $t = 1, \ldots, k$ **do**
      $\hat{\mathbf{h}} \sim p_r(\mathbf{h} \mid \hat{\mathbf{v}})$
      $\hat{\mathbf{v}} \sim p_r(\mathbf{v} \mid \hat{\mathbf{h}})$
    end
    $\mathbf{v}_r := \hat{\mathbf{v}}$
    $\mathbf{h}_r := \hat{\mathbf{h}}$
  end
  `// swap ("exchange particles")`
  for $r = R, R - 1, \ldots, 2$ **do**
    $A := \exp\left[\left(\frac{1}{T_r} - \frac{1}{T_{r-1}}\right)\left(E(\mathbf{v}_r, \mathbf{h}_r) - E(\mathbf{v}_{r-1}, \mathbf{h}_{r-1})\right)\right]$
    $u \sim \text{Uniform}(0, 1)$
    if $u < A$ **then**
      $(\mathbf{v}_{r-1}, \mathbf{h}_{r-1}), (\mathbf{v}_r, \mathbf{h}_r) := (\mathbf{v}_r, \mathbf{h}_r), (\mathbf{v}_{r-1}, \mathbf{h}_{r-1})$
    end
  end
  $\hat{\mathbf{v}}^{(m)} := \mathbf{v}_1$
  $\hat{\mathbf{h}}^{(m)} := \mathbf{h}_1$
end

Algorithm 3: Simulated Annealing of an RBM with k sampling steps. This process is used to generate samples in Parallel Tempering.

5 Estimating the partition function

5.1 Annealed Importance Sampling

6 Extending to other distributions

7 Experiments

All experiments in this section are carried out by the *insanely* popular machine-learning package `ml` by Tor Erlend Fjelde, found at `https://github.com/torfjelde/ml`.

7.1 CD-k vs. PT with k tempered distribution

8 Discussion

9 Appendix

9.1 Gaussian RBMs

References

[1] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines*. *Cognitive Science*, 9(1):147–169, Jan 1985.

[2] Asja Fischer. Training restricted boltzmann machines. *KI - Künstliche Intelligenz*, 29(4):441–444, May 2015.

[3] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G. R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. A high-bias, low-variance introduction to machine learning for physicists. *CoRR*, 2018.

[4] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer, 2012.

[5] Sheldon M Ross. *A first course in probability*. Ninth edition; pearson new international edition.. edition, 2014.

[6] Radford M. Neal. Annealed importance sampling. *CoRR*, 1998.