Justifying and Generalizing Contrastive Divergence

Yoshua Bengio and Olivier Delalleau

Dept. IRO, University of Montreal

first submitted to Neural Computation , Nov 9th, 2007 Final revision , submitted Aug 5th, 2008

Abstract

We study an expansion of the log-likelihood in undirected graphical models such as the RBM, where each term in the expansion is associated with a sample in a Gibbs chain alternating between two random variables (the visible vector and the hidden vector, in RBMs).

We are particularly interested in estimators of the gradient of the log-likelihood obtained through this expansion. We show that its residual term converges to zero, justifying the use of a truncation, i.e. running only a short Gibbs chain, which is the main idea behind the CD estimator of the log-likelihood gradient. By truncating even more, we obtain a stochastic reconstruction error, related through a mean-field approximation to the reconstruction error often used to train autoassociators and stacked auto-associators. The derivation is not specific to the particular parametric forms used in RBMs, and only requires convergence of the Gibbs chain.

We present theoretical and empirical evidence linking the number of Gibbs steps k and the magnitude of the RBM parameters to the bias in the CD estimator.

These experiments also suggest that the sign of the CD estimator is correct most of the time, even when the bias is large, so that CD-k is a good descent direction even for small k.

1 Introduction

Motivated by the theoretical limitations of a large class of non-parametric learning algorithms (Bengio & Le Cun, 2007), recent research has focussed on learning algorithms for so-called **deep architectures** (Hinton, Osindero, & Teh, 2006; Hinton & Salakhutdinov, 2006; Bengio, Lamblin, Popovici, & Larochelle, 2007; Salakhutdinov & Hinton, 2007; Ranzato, Poultney, Chopra, & LeCun, 2007; Larochelle, Erhan, Courville, Bergstra, & Bengio, 2007). These represent the learned function through many levels of composition of elements taken in a small or parametric set. The most common element type found in the above papers is the soft or hard linear threshold unit, or **artificial neuron**

$$\text{output}(\text{input}) = s(w'\text{input} + b) \tag{1}$$

Here, we are particularly interested in the RBM (Smolensky, 1986; Freund & Haussler, 1994; Hinton, 2002; Welling, Rosen-Zvi, & Hinton, 2005; Carreira-Perpinan & Hinton, 2005), a family of bipartite graphical models with hidden variables (the hidden layer) which are used as components in building Deep Belief Networks (Hinton et al., 2006; Bengio et al., 2007;Salakhutdinov & Hinton, 2007; Larochelle et al., 2007). DBNs have yielded impressive performance on several benchmarks, clearly beating the sota and other non-parametric learning algorithms in several cases. A very successful learning algorithm for training a RBM is the CD algorithm. An RBM represents the joint distribution between a **visible**

vector X which is the rv observed in the data, and a **hidden** rv H. There is no tractable representation of $P(X, H)$ but conditional distributions $P(H|X)$ and $P(X|H)$ can easily be computed and sampled from. CD-k is based on a Gibbs MCMC starting at an example $X = x_1$ from the empirical distribution and converging to the RBM's generative distribution $P(X)$. CD-k relies on a biased estimator obtained after a small number k of Gibbs steps (often only 1 step). Each Gibbs step is composed of two alternating sub-steps: sampling $h_t \sim P(H|X = x_t)$ and sampling $x_{t+1} \sim P(X|H = h_t)$, starting at $t = 1$.

The surprising empirical result is that even $k = 1$ (CD-1) often gives good results. An extensive numerical comparison of training with CD-k versus exact log-likelihood gradient has been presented in (Carreira-Perpiñan & Hinton, 2005). In these experiments, taking k larger than 1 gives more precise results, although very good approximations of the solution can be obtained even with $k = 1$. Here we present a follow-up to (Carreira-Perpiñan & Hinton, 2005) that brings further theoretical and empirical support to CD-k, even for small k.

CD-1 has originally been justified (Hinton, 2002) as an approximation of the gradient of $KL(P(X_2 = \cdot\,|x_1)||P(X = \cdot)) - KL(\hat{P}(X = \cdot)||P(X = \cdot))$, $\hat{P}$ is the empirical distribution of the training data, and $P(X_2 = \cdot\,|x_1)$ denotes the distribution of the chain after one step. The term left out in the approximation of the gradient of the KL difference is (Hinton, 2002)

$$\sum_x \frac{\partial KL(P(X_2 = \cdot\,|x_1)||P(X = \cdot)}{\partial P(X_2 = x|x_1)} \frac{\partial P(X_2 = x|x_1)}{\partial \theta} \qquad (2)$$

which was empirically found to be small. On the one hand it is not clear how aligned are the log-likelihood gradient and the gradient wrt the above KL difference. On the other hand it would be nice to prove that left-out terms are small in some sense. One of the motivations for this paper is to obtain the CD algorithm from a different route, by which we can prove that the term left-out wrt the log-likelihood gradient is small and converging to zero, as we take k larger.

We show that the log-likelihood and its gradient can be expanded by considering samples in a Gibbs chain. We show that when truncating the gradient expansion to k steps, the remainder converges to zero at a rate that depends on the mixing rate of the chain. The inspiration for this derivation comes from Hinton et al. (2006): first the idea that the Gibbs chain can be associated with an infinite directed graphical model (which here we associate to an expansion of the log-likelihood and of its gradient), and second that the convergence of the chain justifies CD (since the k-th sample from the Gibbs chain becomes equivalent to a model sample). However, our empirical results also show that the convergence of the chain alone cannot explain the good results obtained by CD,

because this convergence becomes too slow as weights increase during training. It turns out that even when k is not large enough for the chain to converge (e.g. the typical value $k = 1$), the CD-k rule remains a good update direction to increase the log-likelihood of the training data.

Finally, we show that when truncating the series to a single sub-step we obtain the gradient of a stochastic reconstruction error. A mean-field approximation of that error is the reconstruction error often used to train autoassociators (Rumelhart, Hinton, & Williams, 1986; Bourlard & Kamp, 1988; Hinton & Zemel, 1994; Schwenk & Milgram, 1995; Japkowicz, Hanson, & Gluck, 2000). Auto-associators can be stacked using the same principle used to stack RBMs into a Deep Belief Network in order to train deeep neural networks (Bengio et al., 2007; Ranzato et al., 2007; Larochelle et al., 2007). Reconstruction error has also been used to monitor progress in training RBMs by CD (Taylor, Hinton, & Roweis, 2006; Bengio et al., 2007), because it can be computed tractably and analytically, without sampling noise.

2 RBMs and CD

2.1 BMs

A BM (Hinton, Sejnowski, & Ackley, 1984; Hinton & Sejnowski, 1986) is a probabilistic model of the joint distribution between **visible units** x, **hidden units** h, associated with a *quadratic* **energy function**

$$\mathcal{E}(x,h) := -b'x - c'h - h'Wx - x'Ux - h'Vh \tag{4}$$

$$P(x,h) = \frac{e^{-\mathcal{E}(x,h)}}{Z} \tag{5}$$

where $Z = \sum_{x,h} e^{-\mathcal{E}(x,h)}$ is the partition function and (b, c, W, U, V) are parameters of the model.

b_j is called the bias of visible unit x_j, c_i is the bias of visible unit h_i, and the matrices W, U, and V represent **interaction terms** between units. Note that non-zero U and V mean that there are interactions between units belonging to the same layer . **free energy** :

$$\mathcal{F}(x) = -\log \sum_h e^{-\mathcal{E}(x,h)}. \tag{6}$$

==>

$$\begin{aligned} \frac{\partial \log P(x)}{\partial \theta} &= -\frac{\sum_h e^{-\mathcal{E}(x,h)} \frac{\partial \mathcal{E}(x,h)}{\partial \theta}}{\sum_h e^{-\mathcal{E}(x,h)}} + \frac{\sum_{\tilde{x},\tilde{h}} e^{-\mathcal{E}(\tilde{x},\tilde{h})} \frac{\partial \mathcal{E}(\tilde{x},\tilde{h})}{\partial \theta}}{\sum_{\tilde{x},\tilde{h}} e^{-\mathcal{E}(\tilde{x},\tilde{h})}} \\ &= -\sum_h P(h|x) \frac{\partial \mathcal{E}(x,h)}{\partial \theta} + \sum_{\tilde{x},\tilde{h}} P(\tilde{x},\tilde{h}) \frac{\partial \mathcal{E}(\tilde{x},\tilde{h})}{\partial \theta}. \end{aligned} \tag{8}$$

If sampling from the model was possible , one could obtain a *stochastic gradient* for use in training the model . Two samples are necessary : h given x for the first term , **positive phase** , and an $(\tilde{x},\tilde{h}) \sim P(\tilde{x},\tilde{h})$ **negative phase**. Note how the resulting *stochastic gradient estimator*

$$-\frac{\partial \mathcal{E}(x,h)}{\partial \theta} + \frac{\partial \mathcal{E}(\tilde{x},\tilde{h})}{\partial \theta} \tag{9}$$

has one term for each of the positive phase and negative phase, with the same form but opposite signs.

2.2 RBMs

In a RBM, $U = 0$ and $V = 0$ in (4),This form of model was first introduced under the name of **Harmonium** (Smolensky, 1986). Because of this restriction, $P(h|x)$ and $P(x|h)$ factorize and can be computed and sampled from easily. This enables the use of a 2-step Gibbs sampling alternating between $h \sim P(H|X = x)$ and $x \sim P(X|H = h)$. In addition, the positive phase gradient can be obtained exactly and efficiently because the free energy factorizes:

$$\begin{aligned} e^{-\mathcal{F}(x)} = \sum_h e^{b'x+c'h+h'Wx} &= e^{b'x} \sum_{h_1} \sum_{h_2} \cdots \sum_{h_{d_h}} \prod_{i=1}^{d_h} e^{c_i h_i + (Wx)_i h_i} \\ &= e^{b'x} \sum_{h_1} e^{h_1(c_1+W_1 x)} \cdots \sum_{h_{d_h}} e^{h_{d_h}(c_{d_h}+W_{d_h} x)} \\ &= e^{b'x} \prod_{i=1}^{d_h} \sum_{h_i} e^{h_i(c_i+W_i x)} \end{aligned}$$

where W_i is the i-th row of W and d_h the dimension of h. Using the same type of factorization, one obtains for example in the most common case where h_i is binary

$$-\sum_h P(h|x) \frac{\partial \mathcal{E}(x,h)}{\partial W_{ij}} = E[H_i|x] \cdot x_j, \tag{10}$$

where

$$E[H_i|x] = P(H_i = 1|X = x) = \text{sigm}(c_i + W_i x). \tag{11}$$

The log-likelihood gradient for W_{ij} thus has the form

$$\frac{\partial \log P(x)}{\partial W_{ij}} = P(H_i = 1|X = x) \cdot x_j - E_X[P(H_i = 1|X) \cdot X_j] \tag{12}$$

where E_X is an expectation over $P(X)$. Samples from $P(X)$ can be approximated by running an alternating Gibbs chain $x_1 \Rightarrow h_1 \Rightarrow x_2 \Rightarrow h_2 \Rightarrow \ldots$ Since the model P is trying to imitate the empirical distribution $\hat{P}$, it is a good idea to start the chain with a sample from $\hat{P}$, so that we start the chain from a distribution close to the asymptotic one.

2.3 CD

The CD-k (Hinton, 1999, 2002) involves a second approximation besides the use of MCMC to sample from P. This additional approximation introduces some bias in the gradient:

we run the MCMC chain for only k steps, starting from the observed example x.

$$\frac{\partial \log P(x)}{\partial \theta} = -\frac{\partial \mathcal{F}(x)}{\partial \theta} + \sum_{\tilde{x}} P(\tilde{x}) \frac{\partial \mathcal{F}(\tilde{x})}{\partial \theta}. \tag{13}$$

The CD-k update after seeing example x is taken proportional to

$$\Delta\theta = -\frac{\partial \mathcal{F}(x)}{\partial \theta} + \frac{\partial \mathcal{F}(\tilde{x})}{\partial \theta} \tag{14}$$

where ˜x is a sample from our Markov chain after k steps.

3 Log-Likelihood Expansion via Gibbs Chain

In the following we consider the case where both h and x can only take a finite number of values. We also assume that there is no pair (x, h) such that $P(x|h) = 0$ or $P(h|x) = 0$. This ensures the Markov chain associated with Gibbs sampling is **irreducible**, and there exists a unique stationary distribution $P(x, h)$ the chain converges to.

Note that $E_{X_t}[\log P(X_t)|x_1]$ is the negative entropy of the t-th visible sample of the chain, and it does not become smaller as $t \to \infty$. Therefore it does not seem reasonable to truncate this expansion.

Lemma 3.2. *For any model $P(Y)$ with parameters θ,* $E\left[\frac{\partial \log P(Y)}{\partial \theta}\right] = 0$

Theorem 3.3. *Consider the converging Gibbs chain* $x_1 \Rightarrow h_1 \Rightarrow x_2 \Rightarrow h_2 \ldots$. *The log-likelihood gradient can be written*

$$\frac{\partial \log P(x_1)}{\partial \theta} = \underbrace{-\frac{\partial \mathcal{F}(x_1)}{\partial \theta} + E_{X_t}\left[\frac{\partial \mathcal{F}(X_t)}{\partial \theta}\middle| x_1\right]}_{\text{expected CD-update}} + \underbrace{E_{X_t}\left[\frac{\partial \log P(X_t)}{\partial \theta}\middle| x_1\right]}_{\text{bias}} \tag{17}$$

and the final term (the bias of the CD estimator) $\to 0$ *as* $t \to \infty$.

Proof. take derivatives wrt a parameter θ in the log-likelihood expansion:

$$\begin{aligned}\frac{\partial \log P(x_1)}{\partial \theta} &= \frac{\partial}{\partial \theta}\log\frac{P(x_1)}{P(x_t)} + \frac{\partial \log P(x_t)}{\partial \theta} \\ &= -\frac{\partial \mathcal{F}(x_1)}{\partial \theta} + \frac{\partial \mathcal{F}(x_t)}{\partial \theta} + \frac{\partial \log P(x_t)}{\partial \theta}.\end{aligned}$$

Then we take expectations wrt the Markov chain conditional on x_1:

$$\frac{\partial \log P(x_1)}{\partial \theta} = -\frac{\partial \mathcal{F}(x_1)}{\partial \theta} + E_{X_t}\left[\frac{\partial \mathcal{F}(X_t)}{\partial \theta}\middle| x_1\right] + E_{X_t}\left[\frac{\partial \log P(X_t)}{\partial \theta}\middle| x_1\right].$$

In order to prove the CD bias $\to 0$, we will use the assumed convergence of the chain:

$$P(X_t = x | X_1 = x_1) = P(x) + \epsilon_t(x) \tag{18}$$

with $\sum_x \epsilon_t(x) = 0$ and $\lim_{t\to+\infty} \epsilon_t(x) = 0$ for all x. Since x is finite, $\epsilon_t \overset{def}{=} \max_x |\epsilon_t(x)| \to 0$

$$\begin{aligned}E_{X_t}\left[\frac{\partial \log P(X_t)}{\partial \theta}\middle| x_1\right] &= \sum_{x_t} P(x_t|x_1)\frac{\partial \log P(x_t)}{\partial \theta} \\ &= \sum_{x_t} P(x_t)\frac{\partial \log P(x_t)}{\partial \theta} + \sum_{x_t} \epsilon_t(x_t)\frac{\partial \log P(x_t)}{\partial \theta}.\end{aligned}$$

Using Lemma 3.2, the first sum =0. ==>

$$\begin{aligned}\left|E_{X_t}\left[\frac{\partial \log P(X_t)}{\partial \theta}\middle| x_1\right]\right| &\le \sum_{x_t} |\epsilon_t(x_t)|\left|\frac{\partial \log P(x_t)}{\partial \theta}\right| \\ &\le \left(N_x \max_x \left|\frac{\partial \log P(x)}{\partial \theta}\right|\right)\epsilon_t \end{aligned} \tag{19}$$

where N_x is the number of discrete configurations for the rv X. This proves the expectation $\to 0$ as $t \to +\infty$, since $\lim_{t\to+\infty} \epsilon_t = 0$.

□

4 Connection with CD

4.1 Theoretical Analysis

Theorem 3.3 justifies truncating the series after t steps, i.e. ignoring $E_{X_t}\left[\frac{\partial \log P(X_t)}{\partial \theta}\middle| x_1\right]$

$$\frac{\partial \log P(x_1)}{\partial \theta} \simeq -\frac{\partial \mathcal{F}(x_1)}{\partial \theta} + E_{X_t}\left[\frac{\partial \mathcal{F}(X_t)}{\partial \theta}\middle| x_1\right]. \tag{23}$$

Note how the expectation can be readily replaced by sampling $x_t \sim P(X_t|x_1)$, giving rise to the stochastic update (14)

$$\Delta\theta = -\frac{\partial \mathcal{F}(x_1)}{\partial \theta} + \frac{\partial \mathcal{F}(x_t)}{\partial \theta}$$

The idea that faster mixing yields to better approximation by CD-k was already introduced earlier (Carreira -Perpinan & Hinton , 2005 ; Hinton et al., 2006). (19) explicitly relates the convergence of the chain (through $\epsilon_t \to 0$, estimating $P(x)$ with $P(X_{k+1} = x|x_1)$) to the approximation error of the CD-k gradient estimator. When the RBM weights are large it is plausible that the chain will mix more slowly because there is less randomness in each sampling step. Hence it might be advisable to use larger values of k as the weights become larger during training. It is thus interesting to study how fast the bias $\to 0$ as t increases, depending on the magnitude of the weights in an RBM.

Markov chain theory (Schmidt, 2006) ensures that, in the discrete case,

$$\epsilon_t = \max_x |\epsilon_t(x)| \leq (1 - N_x a)^{t-1} \tag{24}$$

where Nx is the number of possible configurations for x, and a is the smallest element in the transition matrix of the Markov chain.

In order to obtain a meaningful bound on (19) we also need to bound the gradient of the log-likelihood. In the following we will thus consider the typical case of a binomial RBM, with θ being a weight W_{ij}.

$$\frac{\partial \log P(x)}{\partial W_{ij}} = P(H_i = 1|X = x) \cdot x_j - E_X[P(H_i = 1|X) \cdot X_j].$$

For any x, both $P(H_i = 1|X = x)$ and x_j are in $(0, 1)$. ==> $\left|\frac{\partial \log P(x)}{\partial W_{ij}}\right| \leq 1.$

Combining this inequality with (24), we obtain from (19) that

$$\left|E_{X_t}\left[\frac{\partial \log P(X_t)}{\partial \theta}\middle| x_1\right]\right| \leq N_x(1 - N_x a)^{t-1}. \quad (25)$$

the Markov chain transition matrix:

$$\begin{aligned} P(x_2|x_1) &= \sum_h P(x_2|h)P(h|x_1) \\ &= \sum_h \Pi_j P(x_{2,j}|h)\Pi_i P(h_i|x_1). \end{aligned}$$

$$==> \quad P(x_{2,j}|h) = \begin{cases} \text{sigm}(h'W_{\cdot j} + b_j) \text{ if } x_{2,j} = 1 \\ \text{sigm}(-h'W_{\cdot j} - b_j) \text{ if } x_{2,j} = 0 \end{cases}$$

$$\geq \text{sigm}\left(-\left|h'W_{\cdot j} + b_j\right|\right)$$

$$\geq \text{sigm}\left(-\left(\sum_i |W_{ij}| + |b_j|\right)\right).$$

Let us denote $\alpha_j = \sum_i |W_{ij}| + |b_j|$, and $\beta_i = \sum_j |W_{ij}| + |c_i|$. $P(h_i|x_1) \geq \text{sigm}(-\beta_i)$. ==>

$$a \geq \sum_h \Pi_j \text{sigm}(-\alpha_j)\Pi_i \text{sigm}(-\beta_i) = N_h \Pi_j \text{sigm}(-\alpha_j)\Pi_i \text{sigm}(-\beta_i). \quad (26)$$

let us denote $\alpha = \max_j \alpha_j \quad \beta = \max_i \beta_i.$ ==>

$$\left|E_{X_t}\left[\frac{\partial \log P(X_t)}{\partial \theta}\middle| x_1\right]\right| \leq N_x\left(1 - N_x N_h \text{sigm}(-\alpha)^{d_x}\text{sigm}(-\beta)^{d_h}\right)^{t-1} \quad (29)$$

where $N_x = 2\wedge_{dx}$ and $N_h = 2\wedge_{dh}$.

Note that although this bound is tight (=0) for any $t \geq 2$ when W and b =0 (since mixing is immediate), the bound is likely to be loose in practical cases . note that this bound on the bias decreases exponentially with the number of steps performed in the CD update, even though this decrease may become linear when the bound is loose. In such cases, it can be written $N_x(1-\gamma)_{t-1}$ with a small γ, close to $N_x(1-\gamma(t-1))$.

If the CD update is considered like a biased and noisy estimator of the true log- likelihood gradient, it can be shown that SGD converges (to a local minimum), provided that the bias is not too large (Yuille, 2005). On the other hand, one should keep in mind that for small k, there is no guarantee that CD converges near the ML solution (MacKay, 2001).

4.2 Experiments

The experiments below confirm the above theoretical results and suggest that even when the bias is large and the weights are large, the sign of the CD estimator may be generally correct.

$$\begin{aligned} \Delta_k(x_1) &= -\frac{\partial \mathcal{F}(x_1)}{\partial \theta} + E_{X_{k+1}}\left[\frac{\partial \mathcal{F}(X_{k+1})}{\partial \theta}\middle| x_1\right] \\ \Delta(x_1) &= -\frac{\partial \mathcal{F}(x_1)}{\partial \theta} + E_X\left[\frac{\partial \mathcal{F}(X)}{\partial \theta}\right] \end{aligned} \tag{30}$$

where $\Delta(x_1)$ is the gradient of the likelihood (13) and $\Delta_k(x_1)$ its average approximation by CD-k (23).

$$\delta_k(x_1) = \Delta(x_1) - \Delta_k(x_1) = E_{X_{k+1}}\left[\frac{\partial \log P(X_{k+1})}{\partial \theta}\middle| x_1\right]$$

as shown in section 4.1, we have

$$\lim_{k \to +\infty} \delta_k(x_1) = 0.$$

Note that our analysis is different from the one in (Carreira-Perpinan & Hinton, 2005), where the solutions found by CD-k and GD on the NLL were compared, while we focus on the updates themselves.

In these experiments, we use two manually generated binary datasets:

1. $Diag_d$ is a d-dimensional dataset containing $d + 1$ samples as follows:

$$\overbrace{000\ldots000}^{d \text{ bits}}$$

$$100\ldots000$$

$$110\ldots000$$

$$111\ldots000$$

$$\ldots$$

$$111\ldots110$$

$$111\ldots111$$

2. $1DBall_d$ is a d-dimensional dataset containing $2d\left\lfloor \frac{d-1}{2} \right\rfloor$ samples, representing "balls" on a one-dimensional discrete line with d pixels. Half of the data examples are generated by first picking the position b of the beginning of the ball (among d possibilities), then its width w. Pixels from b to $b + w - 1 \pmod d$ are then set to 1 while the rest of the pixels are set to 0. The second half of the dataset is generated by simply "reverting" its first half (switching 0s and 1s).

In order to be able to compute $\delta_k(x_1)$ exactly, only RBMs with a small (<=10) number of visible and hidden units are used. We compute quantities for all $\theta = W_{ij}$

The following statistics are then computed over all weights W_{ij} and all training examples x_1:

- the weight magnitude indicators α and β, as defined in (27) and (28),
- the mean of the gradient bias $|\delta_k(x_1)|$, denoted by δ_k and called the absolute bias,
- the median of $\left|\frac{\delta_k(x_1)}{\Delta(x_1)}\right|$, gradient[1] (we use the median to avoid numerical issues for small gradients), denoted by r_k and called the relative bias,
- the sign error s_k, i.e., the fraction of updates for which $\Delta_k(x_1)$ and $\Delta(x_1)$ have different signs.

The RBM is initialized with zero biases and small weights uniformly sampled in $\left[-\frac{1}{d}, \frac{1}{d}\right]$ where d is the number of visible units. Note that even with such small weights, the bound from eq. 29 is already close to its maximum value N_x, so that it is not interesting to plot it on the figures. The number of hidden units is also set to d for the sake of simplicity. The RBM weights and biases are trained by CD-1 with a learning rate set to 10^{-3}: keep in mind that we are not interested in comparing the learning process itself, but rather how the quantities above evolve for different kinds of RBMs, in particular as weights become larger during training. Training is stopped once the average negative log-likelihood over training samples has less than 5% relative difference compared to its lower bound, which here is $\log(N)$, where N is the number of training samples (which are all unique).

Figure 1 shows a typical example of how the quantities defined above evolve during training (β is not plotted as it exhibits the same behavior as α). As the weights increase (as shown by α), so does the

[1]This quantity is more interesting than the absolute bias because it tells us what proportion of the true gradient of the log-likelihood is "lost" by using the CD-k update.

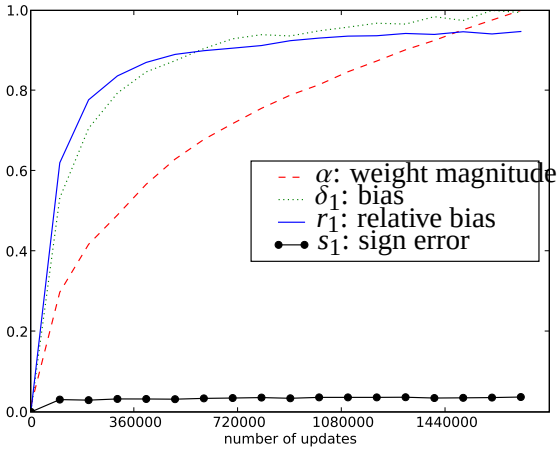

Figure 1: Typical evolution of weight magnitude α, gradient absolute bias δ_1, relative bias r_1 and sign error s_1 as the RBM is being trained by CD-1 on $1DBall_{10}$. The size of weights α and the absolute bias δ_1 are rescaled so that their maximum value is 1, while the relative bias r_1 and the sign disagreement s_1 naturally fall within $[0, 1]$.

absolute value of the left out term in CD-1 (δ_1), and its relative magnitude compared to the log-likelihood (r_1). In particular, we observe that most of the log-likelihood gradient is quickly lost in CD-1 (here after only 80000 updates), so that CD-1 is not anymore a good approximation of negative log-likelihood gradient descent. However, the RBM is still able to learn its input distribution, which can be explained by the fact that the "sign disagreement" s_1 between CD-1 and the log-likelihood gradient remains small (less than 5% for the whole training period).

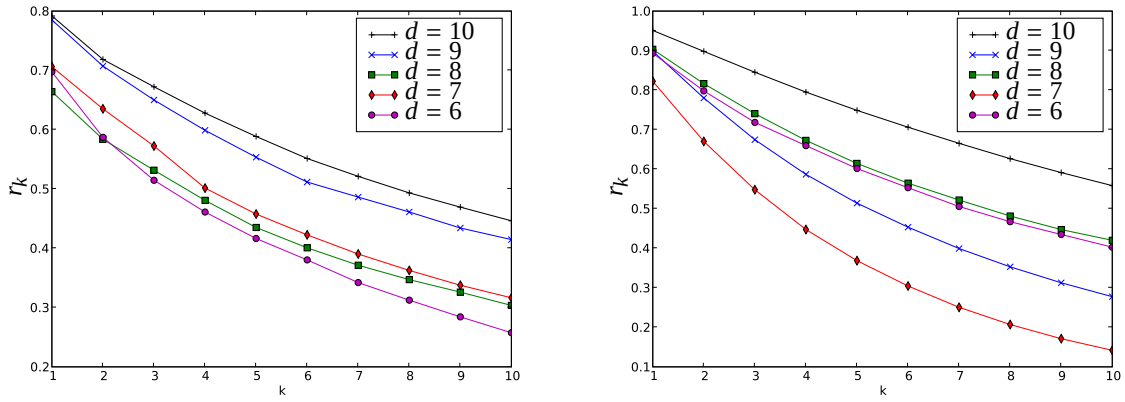

Figure 2: Median relative bias r_k between the CD-k update and the gradient of the log-likelihood, for k from 1 to 10, with input dimension $d \in \{6, 7, 8, 9, 10\}$, when the stopping criterion is reached. Left: on datasets $Diag_d$. Right: on datasets $1DBall_d$.

Figures 2 and 4 show how r_k and s_k respectively vary depending on the number of steps k performed in CD, on the $Diag_d$ (left) and $1DBall_d$ (right) datasets, for $d \in \{6, 7, 8, 9, 10\}$. All these values are taken when our stopping criterion is reached (i.e. we are close enough to the empirical distribution). It may seem surprising that r_k does not systematically increase with d, but remember that each RBM may be trained for a different number of iterations, leading to potentially very different weight magnitude. Figure 3 shows the corresponding values for α and β (which reflect the magnitude of weights): we can see for instance

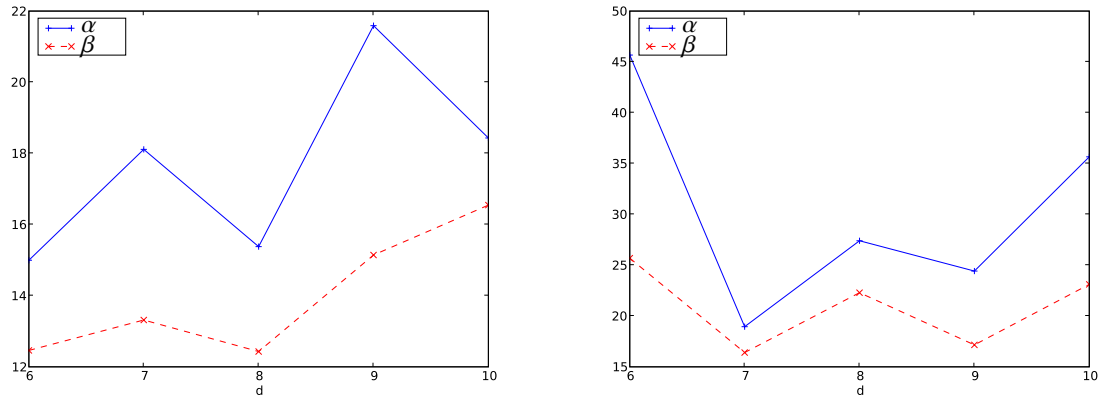

Figure 3: Measures of weight magnitude α and β as the input dimension d varies from 6 to 10, when the stopping criterion is reached. Left: on datasets $Diag_d$. Right: on datasets $1DBall_d$.

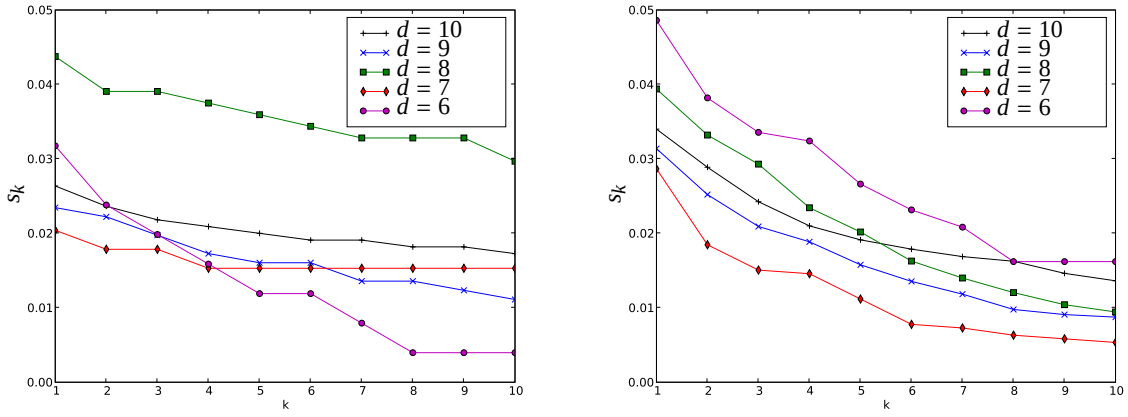

Figure 4: Average disagreement s_k between the CD-k update and negative log-likelihood gradient descent, for k from 1 to 10, with input dimension $d \in \{6, 7, 8, 9, 10\}$, when the stopping criterion is reached. Left: on datasets $Diag_d$. Right: on datasets $1DBall_d$.

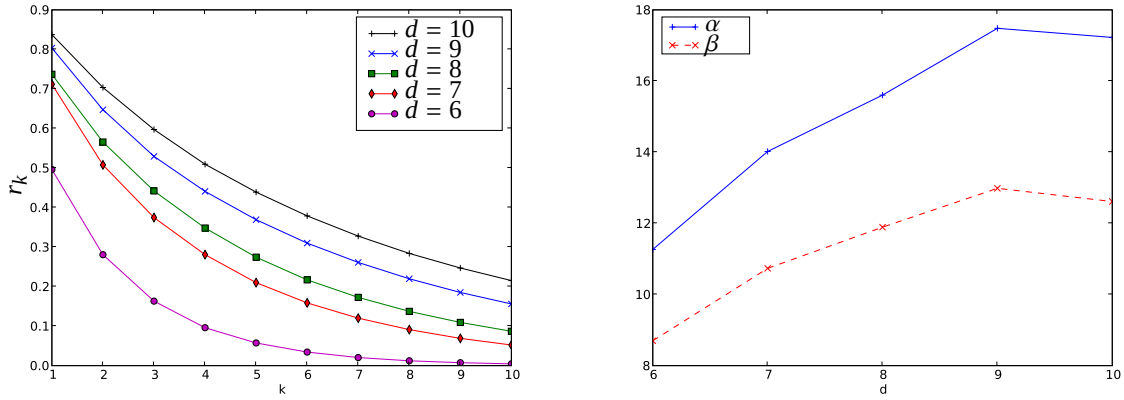

Figure 5: r_k (left) and α and β (right) on datasets $1DBall_d$, after only 300000 training iterations: r_k systematically increases with d when weights are small (compared to figures 2 and 3).

that α and β for dataset $1DBall_6$ are larger than for dataset $1DBall_7$, which explains why r_k is also larger, as shown in figure 2 (right). Figure 5 shows a "smoother" behavior of r_k w.r.t. d when all RBMs are trained for a fixed (small) number of iterations, illustrating how the quality of CD-k decreases in higher dimension (as an approximation to negative log-likelihood gradient descent).

We observe on figure 2 that the relative bias r_k becomes large not only for small k (which means the CD-k update is a poor approximation of the true log-likelihood gradient), but also for larger k in higher dimensions. As a result, increasing k moderately (from 1 to 10) still leaves a large approximation error (e.g. from 80% to 50% with $d = 10$ in Figure 2) in spite of a 10-fold increase in computation time. This suggests that when trying to obtain a more precise estimator of the gradient, alternatives to CD-k such as persistent CD (Tieleman, 2008) may be more appropriate. On another hand, we notice from figure 4 that the disagreement s_k between the two updates remains low even for small k in larger dimensions (in our experiments it always remains below 5%). This may explain why CD-1 can successfully train RMBs even when connection weights become larger and the Markov chain does not mix fast anymore. An intuitive explanation for this empirical observation is the popular view of CD-k as a process that, on one hand, decreases the energy of a training sample x_1 (first term in (30)), and on another hand increases the energy of other nearby input examples (second term), thus leading to an overall increase of $P(x_1)$.

5 Connection with Autoassociator Reconstruction Error

we relate the autoassociator reconstruction error criterion (an alternative to CD learning) to another similar truncation of the log-likelihood expansion.

$$\frac{\partial \log P(x_1)}{\partial \theta} = \frac{\partial}{\partial \theta} \log \frac{P(x_1|h_1)}{P(h_1|x_1)} + \frac{\partial \log P(h_1)}{\partial \theta}.$$

Taking the expectation ==>

$$\frac{\partial \log P(x_1)}{\partial \theta} = E_{H_1}\left[\frac{\partial \log P(x_1|H_1)}{\partial \theta}\middle| x_1\right] - \overbrace{E_{H_1}\left[\frac{\partial \log P(H_1|x_1)}{\partial \theta}\middle| x_1\right]}^{=0} + \overbrace{E_{H_1}\left[\frac{\partial \log P(H_1)}{\partial \theta}\middle| x_1\right]}^{\text{bias}} \quad (31)$$

removing the last term (as in CD) we obtain:

$$\sum_{h_1} P(h_1|x_1) \frac{\partial \log P(x_1|h_1)}{\partial \theta} \quad (32)$$

Note that this is not quite the negated gradient of the **stochastic reconstruction error**

$$\text{SRE} = -\sum_{h_1} P(h_1|x_1) \log P(x_1|h_1). \quad (33)$$

mean-field approximation : an average $E_X[f(X)]$ over configurations of X is approximated by $f(E[X])$, i.e., using the mean configuration.

Applying MFA to SRE (33) gives the **reconstruction error** typically used in training autoassociators (Rumelhart et al., 1986 ; Bourlard & Kamp , 1988 ; Hinton & Zemel , 1994 ; Schwenk & Milgram , 1995 ;

Japkowicz et al., 2000; Bengio et al., 2007; Ranzato et al., 2007; Larochelle et al., 2007),

$$\text{RE} = -\log P(x_1|\hat{h}_1) \quad (34)$$

where $\hat{h}_1 = E[H_1|x_1]$ is the MF output of the hidden units given the observed input x_1.

apply the MFA to (32) ==>

$$\frac{\partial \log P(x_1)}{\partial \theta} \simeq \frac{\partial \log P(x_1|\hat{h}_1)}{\partial \theta}.$$

It is arguable whether the mean-field approximation per se gives us license to include in $\frac{\partial \log P(x_1|\hat{h}_1)}{\partial \theta}$ the effect of θ on $\hat{h}_1$, but if we do so then we obtain the gradient of the RE(34).

As a result, whereas CD-1 truncates the chain expansion at x_2 (in section 2.3), ignoring

$$E_{X_2}\left[\frac{\partial \log P(X_2)}{\partial \theta}\middle| x_1\right],$$

we see (using the fact that the second term of (31)=0) that reconstruction update truncates the chain expansion one step earlier (at h_1), ignoring

$$E_{H_1}\left[\frac{\partial \log P(H_1)}{\partial \theta}\middle| x_1\right]$$

and working on a MFA instead of a stochastic approximation. The RE gradient can thus be seen as a more biased approximation of the log-likelihood gradient than CD-1. Comparative experiments between RE training and CD-1 training confirm this view (Bengio et al., 2007; Larochelle et al., 2007): CD-1 updating generally has a slight advantage over RE gradient.

RE can be computed deterministically and has been used as an easy method to monitor the progress of training RBMs with CD, whereas the CD-k is generally not the gradient of anything and is stochastic.

6 Conclusion

This paper provides a theoretical /empirical analysis of the log-likelihood gradient in graphical models involving a hidden variable h in addition to the observed variable x, and where $P(h|x)$ and $P(x|h)$ are easy to compute and sample from. That includes the case of CD-RBM. The analysis justifies the use of a short Gibbs chain of length k to obtain a biased estimator of the log-likelihood gradient. Even though our results do not guarantee that the bias decreases monotically with k, we prove a bound that does, and observe this decrease experimentally. Moreover, although this bias may be large when using only few steps in the Gibbs chain (as is usually done in practice), our empirical analysis indicates this estimator remains a good update direction compared to the true (but intractable) log-likelihood gradient.

The analysis also shows a connection between RE, log-likelihood and CD, which helps understand the better results generally obtained with CD and justify the use of RE as a monitoring device when training an RBM by CD. The generality of the analysis also opens the door to other learning algorithms in which $P(h|x)$ and $P(x|h)$ do not have the parametric forms.

References

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. In Schölkopf, B., Platt, J., & Hoffman, T. (Eds.), *Advances in Neural Information Processing Systems 19*, pp. 153–160. MIT Press.

Bengio, Y., & Le Cun, Y. (2007). Scaling learning algorithms towards AI. In Bottou, L., Chapelle, O., DeCoste, D., & Weston, J. (Eds.), *Large Scale Kernel Machines*. MIT Press.

Bourlard, H., & Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics, 59*, 291–294.

Carreira-Perpinan, M. A., & Hinton, G. E. (2005). On CD learning. In Cowell, R. G., & Ghahramani, Z. (Eds.), *Proceedings of the Tenth International Workshop on AI and Statistics, Jan 6-8, 2005, Savannah Hotel, Barbados*, pp. 33–40. Society for Artificial Intelligence and Statistics.

Freund, Y., & Haussler, D. (1994). Unsupervised learning of distributions on binary vectors using two layer networks. Tech. rep. UCSC-CRL-94-25, University of California, Santa Cruz.

Hernández-Lerma, O., & Lasserre, J. B. (2003). *Markov Chains and Invariant Probabilities*. Birkhäuser Verlag.

Hinton, G. E., Osindero, S., & Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*, 1527–1554.

Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in BMs. In Rumelhart, D. E., & McClelland, J. L. (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. MIT Press, Cambridge, MA.

Hinton, G. E., Sejnowski, T. J., & Ackley, D. H. (1984). BMs: Constraint satisfaction networks that learn. Tech. rep. TR-CMU-CS-84-119, Carnegie-Mellon University, Dept. of Computer Science.

Hinton, G. E. (1999). PoEs. In *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN)*, Vol. 1, pp. 1–6 Edinburgh, Scotland. IEE.

Hinton, G. E. (2002). Training PoEs by minimizing CD. *Neural Com- putation*, *14*, 1771–1800.

Hinton, G. E., & Salakhutdinov, R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, *313*, 504–507.

Hinton, G. E., & Zemel, R. S. (1994). AE, minimum description length, and Helmholtz free energy. In Cowan, D., Tesauro, G., & Alspector, J. (Eds.), *Advances in Neural Information Processing Systems 6*, pp. 3–10. Morgan Kaufmann Publishers, Inc.

Japkowicz, N., Hanson, S. J., & Gluck, M. A. (2000). Nonlinear autoassociation is not equivalent to PCA. *Neural Computation*, *12*(3), 531–545.

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., & Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. In Ghahramani, Z. (Ed.), *Twenty-fourth International Conference on Machine Learning (ICML 2007)*: 473–480. Omnipress.

MacKay, D. (2001). Failures of the one-step learning algorithm.. Unpublished report.

Ranzato, M., Poultney, C., Chopra, S., & LeCun, Y. (2007). Efficient learning of sparse representations with an energy-based model. In Schölkopf, B., Platt, J., & Hoffman, T. (Eds.), *Advances in Neural Information Processing Systems 19*. MIT Press.