Empirical Analysis of the Divergence of Gibbs Sampling Based Learning Algorithms for Restricted Boltzmann Machines

Asja Fischer and Christian Igel

Institut für Neuroinformatik
Ruhr-Universität Bochum, 44780 Bochum, Germany
asja.fischer@ini.rub.de,christian.igel@ini.rub.de

Abstract. Learning algorithms relying on Gibbs sampling based stochastic approximations of the log-likelihood gradient have become a common way to train Restricted Boltzmann Machines (RBMs). We study three of these methods, Contrastive Divergence (CD) and its refined variants Persistent CD (PCD) and Fast PCD (FPCD). As the approximations are biased, the maximum of the log-likelihood is not necessarily obtained. Recently, it has been shown that CD, PCD, and FPCD can even lead to a steady decrease of the log-likelihood during learning. Taking artificial data sets from the literature we study these divergence effects in more detail. Our results indicate that the log-likelihood seems to diverge especially if the target distribution is difficult to learn for the RBM. The decrease of the likelihood can not be detected by an increase of the reconstruction error, which has been proposed as a stopping criterion for CD learning. Weight-decay with a carefully chosen weight-decay-parameter can prevent divergence.

Key words: Unsupervised Learning, Restricted Boltzmann Machines, Contrastive Divergence, Gibbs Sampling

1 Introduction

Training large undirected graphical models by vanilla likelihood maximization is in general computationally intractable because it involves averages over an exponential number of terms. Obtaining unbiased estimates of these averages by Markov chain Monte Carlo methods typically requires many sampling steps. However, biased estimates obtained after running a Gibbs chain for just a few steps can be sufficient for model training [1]. This is exploited by *Contrastive Divergence* (CD, [1]) learning and its variants *Persistent CD* (PCD, [2]) and *Fast PCD* (FPCD, [3]), which have been, for example, successfully applied to training of *Restricted Boltzmann Machines* (RBMs), the building blocks of *Deep Belief Networks* (DBNs) [4, 5].

Contrastive Divergence learning is a biased approximation of gradient-ascent on the log-likelihood of the model parameters and thus does not necessarily reach the maximum likelihood estimate of the parameters. The bias depends on

the mixing rate of the Markov chain, and mixing slows down with increasing model parameters [1, 6, 7].[1] Recently it has been shown that the bias can lead to a divergence of the log-likelihood when training RBMs [8, 9]. In this study, we further investigate this divergence behavior and its dependence on the number of sampling steps used for the approximation, the number of hidden neurons of the RBM, and the choice of the weight decay parameter. After a brief description of CD, PCD, and FPCD, we describe our experiments, discuss the results and finally draw our conclusions.

2 Training RBMs

An RBM is an undirected graphical model [1, 10]. Its structure is a bipartite graph consisting of one layer of visible units $\boldsymbol{V} = (V_1, \ldots, V_m)$ to represent observable data and one layer of hidden units $\boldsymbol{H} = (H_1, \ldots, H_n)$ to capture dependencies between observed variables. It is parametrized by the connection weights w_{ij} as well as the biases b_j and c_i of visible and hidden units, respectively ($i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$). Given these parameters, jointly denoted as $\boldsymbol{\theta}$, the modeled joint distribution of $\boldsymbol{V}$ and $\boldsymbol{H}$ is $p(\boldsymbol{v}, \boldsymbol{h}) = e^{-\mathcal{E}(\boldsymbol{v},\boldsymbol{h})}/Z$, where $Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-\mathcal{E}(\boldsymbol{v},\boldsymbol{h})}$ and the energy $\mathcal{E}$ is given by

$$\mathcal{E}(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i \ .$$

Differentiating the log-likelihood $\ell(\boldsymbol{\theta}|\boldsymbol{v}_l)$ of the model parameters $\boldsymbol{\theta}$ given one training example $\boldsymbol{v}_l$ with respect to $\boldsymbol{\theta}$ yields

$$\frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\theta}|\boldsymbol{v}_l) = -\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}_l) \frac{\partial \mathcal{E}(\boldsymbol{v}_l, \boldsymbol{h})}{\partial \boldsymbol{\theta}} + \sum_{\boldsymbol{v}} p(\boldsymbol{v}) \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}) \frac{\partial \mathcal{E}(\boldsymbol{v}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \ . \qquad (1)$$

Computing the first term on the right side of the equation is straightforward because it factorizes. The computation of the second term is intractable for regular sized RBMs because its complexity is exponential in the size of the smallest layer. However, the expectation over $p(\boldsymbol{v})$ can be approximated by alternating Gibbs sampling [11, 12]. But since the sampling chain needs to be long to get almost unbiased samples of the distribution modeled by the RBM, the computational effort is still too large.

Contrastive divergence. Instead of running the Gibbs chain until a near-to-equilibrium distribution is reached, in the k-step Contrastive Divergence (CD_k) algorithm [1] the chain is run for only k steps, starting from an example $\boldsymbol{v}^{(0)}$ of the training set and yielding the sample $\boldsymbol{v}^{(k)}$. Each step t consists of sampling $\boldsymbol{h}^{(t)}$ from $p(\boldsymbol{h}|\boldsymbol{v}^{(t)})$ and sampling $\boldsymbol{v}^{(t+1)}$ from $p(\boldsymbol{v}|\boldsymbol{h}^{(t)})$ subsequently. The gradient (1) with respect to $\boldsymbol{\theta}$ of the log-likelihood for one training pattern $\boldsymbol{v}^{(0)}$ is

[1] When referring to sizes of model parameters, we refer to their absolute values.

then approximated by

$$\mathrm{CD}_k(\boldsymbol{\theta}, \boldsymbol{v}^{(0)}) = -\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^{(0)}) \frac{\partial \mathcal{E}(\boldsymbol{v}^{(0)}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} + \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^{(k)}) \frac{\partial \mathcal{E}(\boldsymbol{v}^{(k)}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \ . \quad (2)$$

In the following, we restrict our considerations to RBMs with binary units for which $E_{p(h_i|\boldsymbol{v})}[h_i] = \mathrm{sigmoid}\left(c_i + \sum_{j=1}^{m} w_{ij} v_j\right)$ with $\mathrm{sigmoid}(x) = (1 + \exp(-x))^{-1}$.

The expectation $E_{p(\boldsymbol{v}^{(k)}|\boldsymbol{v}^{(0)})}\left[\mathrm{CD}_k(\boldsymbol{\theta}, \boldsymbol{v}^{(0)})\right]$ is denoted by $\mathrm{CD}_k^*(\boldsymbol{\theta}, \boldsymbol{v}^{(0)})$. Further, we denote the average of $\mathrm{CD}_k(\boldsymbol{\theta}, \boldsymbol{v}^{(0)})$ over a training set by $\overline{\mathrm{CD}}_k(\boldsymbol{\theta})$ and its expectation by $\overline{\mathrm{CD}}_k^*(\boldsymbol{\theta})$. The expectations are considered for theoretical reasons. They lead to deterministic updates, but are computable only for small models.

Training RBMs using CD need not lead to a maximum likelihood estimate of the model parameters. Examples of energy functions and Markov chains for which CD_1 learning does not converge are given in [13]. Yuille [14] specifies conditions under which CD learning is guaranteed to converge to the maximum likelihood solution, which need not hold for RBM training in general. Experiments comparing the quality of small RBMs trained based on CD_k^* and true likelihood maximization are presented in [6] and [7].

Refined learning algorithms. More recently, refined algorithms also based on approximating the log-likelihood via Gibbs sampling have been proposed [2, 3]. In *Persistent Contrastive Divergence* (PCD, [2]) the sample $\boldsymbol{v}^{(k)}$ in the CD approximation (2) is sampled from a Markov chain defined by the RBM parameters that is independent of $\boldsymbol{v}^{(0)}$. This corresponds to standard CD learning without reinitializing the visible units of the Markov chain with the current training sample. It is assumed that the chain stays close to the stationary distribution if the learning rate is sufficiently small and thus the model changes only slightly between parameter updates [15, 2]. The PCD algorithm was further refined leading to a variant called *Fast Persistent Contrastive Divergence* (FPCD, [3]). A set of additional parameters is introduced, which are only used for Gibbs sampling. The new parameters are referred to as *fast* parameters and should lead to higher mixing rates. When calculating the conditional distributions for Gibbs sampling, the regular parameters are replaced by the sum of the regular and the fast parameters. The update rule for the fast parameters is equal to that of the regular parameters, but with an independent, large learning rate and a large weight-decay parameter. Weight decay can also be used for the regular parameters, but it was suggested that regularizing just the fast weights is sufficient [3]. For details about (F)PCD we refer to the original publications [2, 3].

Limitations of the proposed learning algorithms. Bengio and Delalleau [7] show that CD_k is an approximation of the true log-likelihood gradient by finding an expansion of the gradient that considers the k-th sample in the Gibbs chain and showing that CD_k is equal to a truncation of this expansion. Furthermore, they prove that the residual term (i.e., the bias of CD) converges to zero as k goes to

infinity, and show empirically (by comparing the log-likelihood gradient and the expectation CD_k^* in small RBMs) that the quality of CD_k as an approximation of the log-likelihood gradient decreases as the norm of the parameters increases. Anyhow, the RBMs are still able to model the considered simple target distributions. Additionally, they find that the bias of CD_k also increases with increasing number of visible units.

Fischer and Igel [8] show that CD can even lead to a steady decrease of the log-likelihood during learning. This is confirmed in [9] also for PCD and FPCD. Desjardins et al. as well as Salakhutdinov [9, 16] further show that using algorithms based on tempered Markov Chain Monte Carlo techniques yields better training results than Gibbs sampling.

3 Experiments

In our experiments, we study the evolution of the log-likelihood during gradient-based training of RBMs using CD_k, PCD, or FPCD. We first briefly describe our benchmark problems and then give details of the experimental setup.

Benchmark problems. We consider two artificial benchmark problems taken from the literature [12, 17]. The *Labeled Shifter Ensemble* is a 19 dimensional data set containing 768 samples. The samples are generated in the following way: The states of the first 8 visible units are set uniformly at random. The states of the following 8 units are cyclically shifted copies of the first 8. The shift can be zero, one unit to the left, or one to the right and is indicated by the last three units. The log-likelihood is $768 \log \frac{1}{768} \approx -5102.43$ if the distribution of the data set is modeled perfectly.

Further, we consider a smaller variant of the *Bars-and-Stripes* problem described in [17] with 16 instead of 25 visible units. Each pattern corresponds to a square of 4×4 units (if not stated otherwise) and is generated by first randomly choosing an orientation, vertical or horizontal with equal probability, and then picking the state for all units of every row or column uniformly at random. Since each of the two completely uniform patterns can be generated in two ways, the lower bound of the log-likelihood is -102.59.

Experimental setup. The RBMs were initialized with weights drawn uniformly from $[-0.5, 0.5]$ and zero biases. The number of hidden units was chosen to be equal to, twice, or half the number of the visible units.

The models were trained on both benchmark problems using gradient ascent on CD_k with $k \in \{1, 2, 4, 10, 100\}$, PCD, or FPCD. In the experiments presented here, we only discuss batch learning, but it was verified for CD that online learning leads to similar results. The batch update rule was augmented with optional weight-decay, that is, for CD_k we have

$$\boldsymbol{\theta}^{(g+1)} = \boldsymbol{\theta}^{(g)} + \eta \overline{\mathrm{CD}}_k(\boldsymbol{\theta}^{(g)}) - \lambda \boldsymbol{\theta}^{(g)} \quad . \tag{3}$$

We tested different learning rates η and values of the weight-decay parameter λ (which is set to zero if not stated otherwise). Using a momentum term did not

improve the results in our experiments (not shown). For the fast parameters in FPCD the learning rate was set to 0.1 and the weight-decay-parameter was set to $\lambda_{\text{fast}} = \frac{19}{20}$ as suggested in [3].

In order to analyze stochastic effects of the Gibbs sampling, we also did experiments using the computationally expensive expectation $\overline{\text{CD}}_1^*(\boldsymbol{\theta}^{(g)})$ of the CD update on a further reduced Bars-and-Stripes problem with 9 visible units.

To save computation time, the exact likelihood was calculated only every 10 iterations of the learning algorithm. We additionally computed the mean reconstruction error, which has been proposed as an early stopping criterion for CD training [18, 19] and is typically used to train autoassociators [18, 20]. The reconstruction error for one training example $\boldsymbol{v}$ is given by $-\log P\left(\boldsymbol{v}\middle|E\left[\boldsymbol{H}|\boldsymbol{v}\right]\right)$. All experiments were repeated 25 times and the medians of the results are presented if not stated otherwise.

Results. The evolution of the median log-likelihood for CD_1 with different learning rates is shown in Fig. 1. After an initial increase, the log-likelihood steadily decreases and the model gets worse. This happens systematically in every trial as indicated by the quartiles. The higher the learning rate (i.e., the faster the learning) the more pronounced the divergence.

The increase of the negative log-likelihood was also observed if the expectation $\overline{\text{CD}}_1^*$ of the 1-step sample was used instead of a real sample. This shows that the observed effects are not caused by sampling noise. Because of computational complexity, we performed only three single trials with different learning rates on a smaller variant of the Bars-and-Stripes problem (3×3 pixel) in the $\overline{\text{CD}}_1^*$ experiments, see right plot in Fig. 2.

Without weight decay, the norm (we looked at both ∞- and 2-norm) of the RBM parameters steadily increased (this is no surprise and therefore the results are not shown).

Comparing the course of the log-likelihood with the corresponding evolution of the reconstruction error, also shown in Fig. 1, reveals that the reconstruction error constantly decreased in our experiments. Thus, an *increase* of the reconstruction error could not be used as a criterion for early-stopping.

As shown in the left plot of Fig. 2, using a decaying learning rate $\eta^{(g)}$ (with decay schedule $\eta^{(g)} = \frac{c_1}{c_2+g}$, where $c_1, c_2 \in \mathbb{R}^+$) can prevent divergence. However, if the learning rate decays too fast (c_2 is small), learning will become too slow and if $\eta^{(g)}$ decays too slowly, the divergence is still observed.

The plots in Fig. 3 show the dependence on the number of sampling steps k. As expected, the larger k the less severe the divergence. However, in our experiments we observed a clear decrease of the likelihood even for $k = 10$. For $k = 100$, there was only a slight final decrease of the likelihood in the Shifter task and a negligible decrease in the Bars-and-Stripes benchmark.

The effect of L_2-norm weight-decay with different weight-decay parameters λ is shown in Fig. 4. The results indicate that the choice of the hyperparameter λ is crucial. If it was chosen correctly, the divergence problem was solved. If λ was too large, the RBMs did not model the target distribution accurately. If the weight-decay parameter was too small, it could not prevent divergence.

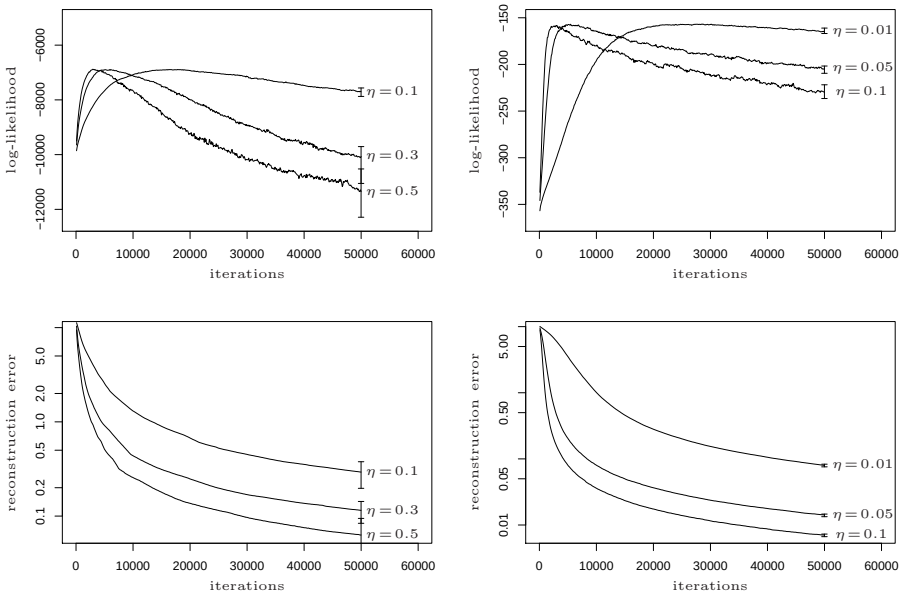

Fig. 1. Top: Evolution of log-likelihood for CD_1 using steepest-descent with different learning rates. Shown are the medians over 25 trials for the Shifter (left) and Bars-and-Stripes (right) problem, error bars indicate quartiles. Bottom: Corresponding reconstruction error.

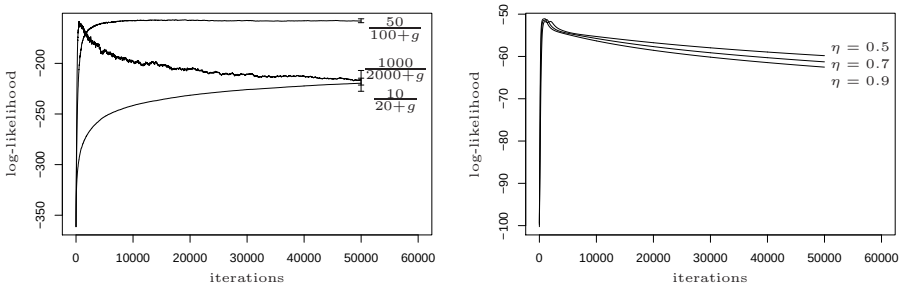

Fig. 2. Left: Log-likelihood for CD_1 with different adaptive learning rates for Bars-and-Stripes. Right: Log-likelihood for CD_1^* for the smaller Bars-and-Stripes problem with 3×3 units. In contrast to all other plots, here only single trials and not medians are shown.

The influence of the number of hidden units is demonstrated in Fig. 5. The more hidden units the more expressive power the RBM has [21]. Thus, the more hidden units the "easier" the problem for the RBM. Therefore, the results in Fig. 5 suggest that the easier the problem the lower the risk of divergence.

The learning curves for PCD and FPCD were very similar to each other. We observed the same divergence effects (e.g., see Fig. 6) that could be tamed by weight-decay (weight-decay results are not shown).

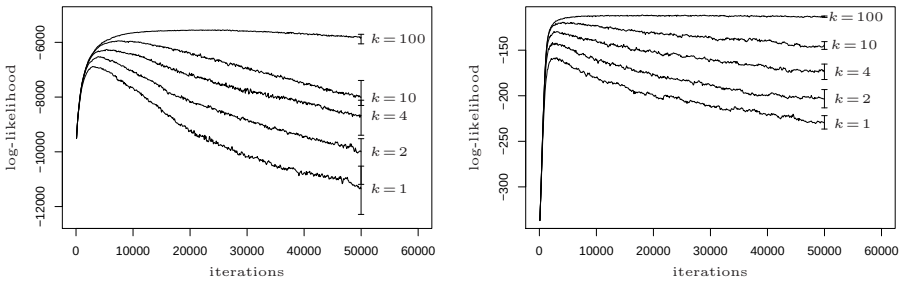

Fig. 3. Log-likelihood for CD_k with different choices of k. The learning rates were $\eta = 0.5$ and $\eta = 0.1$ for the Shifter (left) and Bars-and-Stripes (right) problem.

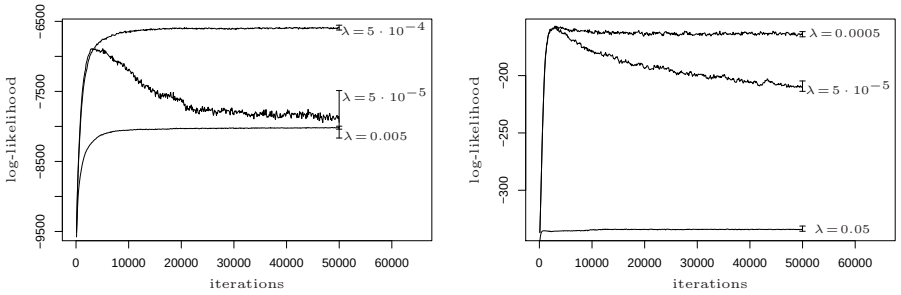

Fig. 4. Evolution of the log-likelihood for CD_1 and weight-decay with different weight-decay parameters λ and learning rates as in Fig. 3 for the Shifter (left) and Bars-and-Stripes (right) problem.

4 Discussion

We have shown on benchmark problems from the literature that vanilla gradient-based optimization of RBMs via k-step CD, PCD, and FPCD can lead to a systematic decrease of the likelihood after an initial increase. The reason for this is that an increase of the model parameters increases the difference between the CD update and a gradient step on the log-likelihood. The weight increase steadily slows down the mixing rate of the Gibbs chain associated with the CD approximation (the fact that Gibbs chains in general converge faster if the start distribution is close to the target distribution does not compensate for this). With increasing weights the mixing rate goes down and makes sampling in the Gibbs chain more and more deterministic, thus inducing a strong bias. If (for some components) this bias further increases the respective weights and decreases the likelihood, CD learning diverges. However, is this really a problem in practice? One may argue that there are several well-known and simple ways to address this problem, namely (i) early stopping of the learning process [18, 19], (ii) regularization using weight-decay [1, 5, 4], and (iii) increasing k dynamically when the weights increase [7].

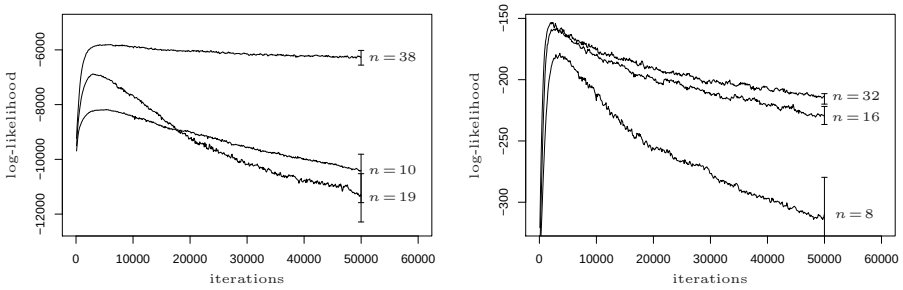

Fig. 5. Log-likelihood for CD_1 applied to RBMs with different numbers of hidden variables n for the Shifter (left) and Bars-and-Stripes (right) problem.

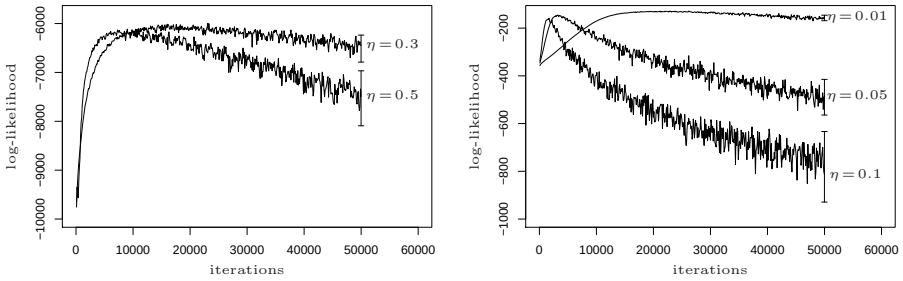

Fig. 6. Evolution of log-likelihood for PCD (Shifter, left plot) and FPCD (Bars-and-Stripes, right plot) depending on learning rate η.

Early stopping requires some reliable indicator that tells us when to stop and that can be computed efficiently. However, monitoring the true likelihood periodically is only possible for small problems, and the reconstruction error as discussed by [18] and [19] may gradually decrease in spite of decreasing likelihood as, for example, shown in our experiments. An adaptive learning rate can have a similar effect as early-stopping. A decaying learning rate $\eta^{(g)}$ with an appropriate schedule can prevent divergence. However, choosing the right schedule is crucial and difficult. If the learning rate decays too fast, learning will become too slow and we will not reach sufficiently good solutions in time. If $\eta^{(g)}$ decays too slowly, the learning rate adaptation has little effect and divergence is still observed.

Weight decay offers a solution – if the regularization parameter is chosen correctly. If chosen too large, the model may not represent the target density accurately enough. If chosen too small, the decay term does not prevent divergence.

As suggested in [7], increasing k can be a good strategy to find models with higher likelihood and it can also prevent divergence. However, divergence occurs even for values of k too large to be computationally tractable for large models. Thus, a dynamic schedule that enlarges k as the weights increase is needed [7]. Finding such a schedule is an open problem.

It seems that the more difficult the problem (i.e., the more difficult it is for the RBM to represent the target density) the more pronounced the divergence effect.

The low-dimensional problems investigated in [7] are all rather easy to learn for the considered RBMs and therefore the divergence is not apparent in that study. The dependence on difficulty makes the observed phenomenon relevant for DBNs. In these multi-layer architectures, simple models such as RBMs are used in each layer and the complexity of the target distribution is reached by stacking these simple models. The lower layer(s) cannot (or should not) represent the target density alone – and thus RBMs in DBNs face distributions that are difficult to learn.

5 Conclusion

Optimization based on k-step Contrastive Divergence (CD) or (Fast) Persistent CD is a promising approach to train undirected graphical models. It has proven to be successful in challenging applications and has contributed significantly to the current revival of Restricted Boltzmann Machines (RBMs) and deep architectures. The CD is a biased estimate of the desired update direction. This bias is reduced with increasing k and gets worse with increasing norm of the model parameters (i.e., slower mixing rates of the involved Markov chain). While it is common knowledge that CD learning may only approximate the maximum likelihood solution, we showed that the bias can lead to divergence of the learning process in the sense that the model systematically and drastically gets worse if k is not large. Thus, for training algorithms relying on Gibbs sampling based stochastic approximations of the log-likelihood gradient, there is a need for robust mechanisms that control the weight growth in CD and related learning algorithms, for example, reliable heuristics for choosing the weight decay parameters or suitable criteria for early-stopping. New learning methods for RBMs using Markov Chain Monte Carlo algorithms based on tempered transitions are promising [16, 9], but their learning and scaling behavior needs to be further explored.

Acknowledgements

We acknowledge support from the German Federal Ministry of Education and Research within the National Network Computational Neuroscience under grant number 01GQ0951.

References

1. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation **14** (2002) 1771–1800
2. Tieleman, T.: Training restricted Boltzmann machines using approximations to the likelihood gradient. In Cohen, W.W., McCallum, A., Roweis, S.T., eds.: International Conference on Machine learning (ICML), ACM (2008) 1064–1071
3. Tieleman, T., Hinton, G.E.: Using fast weights to improve persistent contrastive divergence. In Pohoreckyj Danyluk, A., Bottou, L., Littman, M.L., eds.: International Conference on Machine Learning (ICML), ACM (2009) 1033–1040

4. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation **18**(7) (2006) 1527–1554
5. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786) (2006) 504–507
6. Carreira-Perpiñán, M.Á.., Hinton, G.E.: On contrastive divergence learning. In: 10th International Workshop on Artificial Intelligence and Statistics (AISTATS 2005). (2005) 59–66
7. Bengio, Y., Delalleau, O.: Justifying and generalizing contrastive divergence. Neural Computation **21**(6) (2009) 1601–1621
8. Fischer, A., Igel, C.: Contrastive divergence learning may diverge when training restricted Boltzmann machines. Frontiers in Computational Neuroscience. Bernstein Conference on Computational Neuroscience (BCCN 2009) (2009)
9. Desjardins, G., Courville, A., Bengio, Y., Vincent, P., Dellaleau, O.: Parallel tempering for training of restricted Boltzmann machines. Journal of Machine Learning Research Workshop and Conference Proceedings **9**(AISTATS 2010) (2010) 145–152
10. Smolensky, P.: Information processing in dynamical systems: Foundations of harmony theory. In Rumelhart, D.E., McClelland, J.L., eds.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1: Foundations. MIT Press (1986) 194–281
11. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for Boltzmann machines. Cognitive Science **9** (1985) 147–169
12. Hinton, G.E., Sejnowski, T.J.: Learning and relearning in Boltzmann machines. In Rumelhart, D.E., McClelland, J.L., eds.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1: Foundations. MIT Press (1986) 282–317
13. MacKay, D.J.C.: Failures of the one-step learning algorithm. Cavendish Laboratory, Madingley Road, Cambridge CB3 0HE, UK. http://www.cs.toronto.edu/ mackay/gbm.pdf (2001)
14. Yuille, A.: The convergence of contrastive divergence. In Saul, L., Weiss, Y., Bottou, L., eds.: Advances in Neural Processing Systems (NIPS 17). (2004) 1593–1600
15. Younes, L.: Maximum likelihood estimation of gibbs fields. In Possolo, A., ed.: Proceedings of an AMS-IMS-SIAM Joint Conference on Spacial Statistics and Imaging. Lecture Notes Monograph Series. Institute of Mathematical Statistics, Hayward, California (1991)
16. Salakhutdinov, R.: Learning in markov random fields using tempered transitions. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C.K.I., Culotta, A., eds.: Advances in Neural Information Processing Systems 22. (2009) 1598–1606
17. MacKay, D.J.C.: Information Theory, Inference & Learning Algorithms. Cambridge University Press (2002)
18. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., Montreal, U.: Greedy layer-wise training of deep networks. In Schölkopf, B., Platt, J., Hoffman, T., eds.: Advances in Neural Information Processing (NIPS 19), MIT Press (2007) 153–160
19. Taylor, G.W., Hinton, G.E., Roweis, S.T.: Modeling human motion using binary latent variables. In Schölkopf, B., Platt, J., Hoffman, T., eds.: Advances in Neural Information Processing Systems (NIPS 19), MIT Press (2007) 1345–1352
20. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088) (1986) 533–536
21. Le Roux, N., Bengio, Y.: Representational power of restricted Boltzmann machines and deep belief networks. Neural Computation **20**(6) (2008) 1631–1649