Internship Report

Training Restricted Boltzmann Machines: Investigation of Parameters

Marco Roth

30.05.2014

Contents

1 Introduction

The goal of the internship was to investigate the influence of certain metaparamters on the quality of training restricted Boltzmann machines (RBM). RBMs are stochastic networks that can be used for unsupervised learning. The outcome of a training process is very sensitive to the correct adjustment of metaparameters such as learning rate and the usage of minibatches i.e. the division of training sets into subsets. It has been suggested that minibatches can improve learning by avoiding the creation of very deep modes [Hinton, 2010]. A recent master thesis by L. Leng has raised the question whether the usage of minibatches is really necessary in order to achieve good results [Leng, 2014].
In chapter 2 we will briefly discuss the theory of restricted Boltzmann machines and the methods that can be used for learning. Chapter 3 presents the results of experiments that were performed on the MNIST data set of handwritten digits in order to determine suitable suggestions for the learning rate and the size of minibatches. Subsequently, chapter 4 provides a discussion of the results and an outlook to further investigations that will be done in a bachelor's thesis.

2 Theoretical Background

In this section we provide a theoretical overview of the methods involved in the experiments. After a brief introduction of restricted Boltzmann machines and Gibbs sampling, we discuss contrastive divergence and persistent contrastive divergence, which are the two training methods that we used.

2.1 Restricted Boltzmann Machines

A Boltzmann Machine is an undirected graphical model and a special case of a Markov Random Field (MRF). For the purposes of this report, we will only consider a binary state space for the random variables. The probability of a state $\mathbf{x}$ is given by:

$$\begin{aligned} p(\mathbf{z}) &= \frac{1}{Z}\exp(-E(\mathbf{z})) \\ E(\mathbf{z}) &= -\frac{1}{2}\sum_i w_{ij} z_i z_j - \sum_i z_i b_i \end{aligned} \tag{1}$$

where $E(\mathbf{z})$ denotes the energy function of a state $\mathbf{x}$.
A Boltzmann Machine is called restricted if the associated graph is bipartite, i.e. the vertices can be divided into two subsets of nodes $\mathbf{V}$ and $\mathbf{H}$, such that every element in $\mathbf{V}$ is connected to every element in $\mathbf{H}$ and there are no edges connecting vertices within each set. We will refer to the data driven and, in this sense, observable units as visible units ($\mathbf{V}$) whereas the latent nodes, that will serve as feature detectors, are called hidden units ($\mathbf{H}$). Figure 1 shows the bipartite structure of a restricted Boltzmann Machine

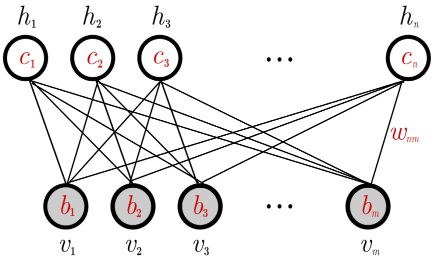

Figure 1: Structure of a RBM with weights w_{ij}, biases of the n hidden units c_j and biases of the m visible units b_i (Figure taken from [Fischer and Igel, 2014])

The energy function of a RBM is given by

$$E(\mathbf{v}, \mathbf{h}; \Theta) = -\sum_{i,j} w_{ij} v_i h_j - \sum_i b_i v_i - \sum_j c_j h_j \;, \tag{2}$$

with the model parameters $\Theta = \{\mathbf{w}, \mathbf{b}, \mathbf{c}\}$ (Figure 1). Due to the bipartite structure of the RBM, the conditional probability distribution factorizes:

$$p(\mathbf{v}|\mathbf{h}) = \prod_i p(v_i|\mathbf{h}) \tag{3}$$

$$p(\mathbf{h}|\mathbf{v}) = \prod_j p(h_j|\mathbf{v}) \tag{4}$$

2.2 Learning Algorithms

RBMs can be used for unsupervised learning i.e. adapting the model parameters in such a way that the likelihood given the training data is maximized. Due to the high dimensionality of the problem, an analytical solution is intractable for our purposes. An intuitive way to approximate the maximum of the likelihood function is gradient ascent where the model parameters are updated stepwise in order to follow the gradient of the likelihood function $\mathcal{L}$ to find the maximum

$$\Theta^{(t+1)} = \Theta^{(t)} + \eta \nabla \mathcal{L}(\Theta|\mathbf{S}) \tag{5}$$

where the step size is given by the learning rate η. For RBMs the following update rule can be derived:

$$\frac{\partial \ln \mathcal{L}(\Theta|\mathbf{v})}{\partial \Theta} = -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \Theta} + \sum_{\mathbf{v},\mathbf{h}} p(\mathbf{v}, \mathbf{h}) \frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \Theta} \tag{6}$$

The first term is the expectation of the energy function under the conditional probability of the hidden units driven by training data. The second term is the expected value of the energy function under the model joint probability distribution.

2.3 Gibbs Sampling

The first term in Equation 6 can usually be calculated directly from the model. The second term is computationally expensive and therefore has to be approximated. A common method is Gibbs sampling, a Markov Chain Monte Carlo (MCMC) algorithm. The basic idea of Gibbs sampling is to stochastically update the states of the units according to the states of all the other units in a rigid order, thus, taking samples from the desired distribution. In the special case of an RBM, the hidden and visible units are independent which allows us to update the states of all the hidden and visible units simultaneously with the probabilities:

$$p(v_i|\mathbf{h}) = \frac{1}{1 + \exp\left(\sum_j w_{ij} h_j + b_i\right)} \tag{7}$$

$$p(h_j|\mathbf{v}) = \frac{1}{1 + \exp\left(\sum_i w_{ij} v_i + c_j\right)} \tag{8}$$

2.4 Contrastive Divergence

Contrastive Divergence (CD-k) is a widely used algorithm that uses k steps of Gibbs sampling to approximate the model distribution in Equation 6. Hinton et al. suggested that often even one step of Gibbs sampling is sufficient [Hinton, 2002]. The parameter update rule then just becomes

$$\begin{aligned} \frac{\partial \ln \mathcal{L}}{\partial \Theta} &= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v}^{(0)}, \mathbf{h})}{\partial \Theta} + \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v}^{(k)}, \mathbf{h})}{\partial \Theta} \\ &= \left\langle v_i^{(0)} h_j^{(0)} \right\rangle - \left\langle v_i^{(k)} h_j^{(k)} \right\rangle \end{aligned} \tag{9}$$

The sampling of the states under the model distribution now only requires the conditional probability which can be computed easily with eq. 3 and 4.

2.5 Persistent Contrastive Divergence

CD-k, especially if k is small, has a few disadvantages and even causes the model parameters to diverge as the training progresses [Fischer and Igel, 2010]. Recently proposed algorithms like persistent contrastive divergence (PCD) can increase the mixing rate of the Markov chain, thus, achieve better learning results [Tieleman, 2008]. The idea is to refrain from resetting the Markov chain for every parameter update and instead use the state of the Boltzmann machine from the previous iteration. The obtained samples provide a better estimation for the model distribution as the chain stays closely to the stationary distribution. This is only the case if the model does not change too much between iterations which can be achieved by using a sufficiently small learning rate.

2.6 Conservative Sampling-based Likelihood Estimator

Since calculating the likelihood of an RBM is intractable, we use the Conservative Sampling-based Likeliood Estimator (CSL) proposed by Benigio et al. [Yoshua Bengio, 2013] to assess the quality of training.

$$\ln \hat{f}(x) = \ln mean_{\mathbf{h'}} p(\mathbf{v}|\mathbf{h'}) \tag{10}$$

In our case, for every CSL calculation, 1000 samples of hidden units are drawn with PCD.

3 Experimental Results

All experiments are performed by training on the MNIST dataset of handwritten digits. The trainingset can be divided into subsets that are called minibatches. Running several Markov chains in parallel and averaging over multiple gradients before parameter updates, rather than updating the parameters after each presentation of a training image, is supposed to prevent the model from creating modes of very high probability where the RBM can get "trapped". This causes a decrease in the mixing rate and therefore in the log-likelihood. We investigated the influence of the metaparameters *learning rate* η, the size of *minibatches* and the *training method* (i.e. CD-k and PCD) on the quality of learning. We used a subset of 30 binarized images with a threshold of 0.5. The number of hidden units was set to 50 and the parameters of the RBM were initialized randomly.

Figure 2 shows a comparison between a CD1 and a PCD trained model. The learning rate is constant and the minibatch size is 1. After about 200,000 iterations the CSL decreases with CD-1, which is a known problem [Fischer and Igel, 2010]. As the absolute values of the parameters increase, the stochasticity of the network decreases, hence, the mixing rate worsens. PCD can avoid this problem by drawing samples from a distribution that represents the model more accurately. As a result, PCD outperforms CD-1 after a few iterations.

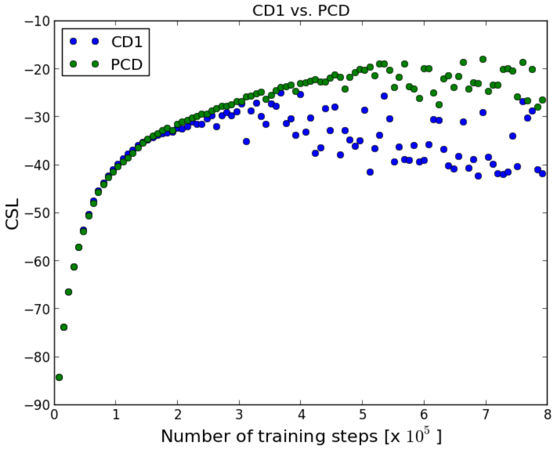

Figure 2: Comparison between CD-1 - and PCD based learning on the MNIST dataset with minibatch size 1. The number of hidden units is 50 and the model is trained with a constant learning rate of $\eta = 0.0001$. The CD-1 algorithm leads to a decreasing log-likelihood after about 300000 training steps due to increasing weights and biases (see also [Fischer and Igel, 2010]).

Finding a suitable learning rate for training is essential for obtaining good results. Figure 3 shows a PCD trained RBM with four different values for η. As PCD requires a small learning rate, the likelihood decreases after an initial increase if the learning rate is too high. However, a small η seems to achieve good results. We used an additional adaptive learning rate $\eta = \frac{1}{t+t_0}$ that decreases as the training progresses. The adaptive learning rate achieves good results quickly and seems to converge towards a limit as η approaches 0.

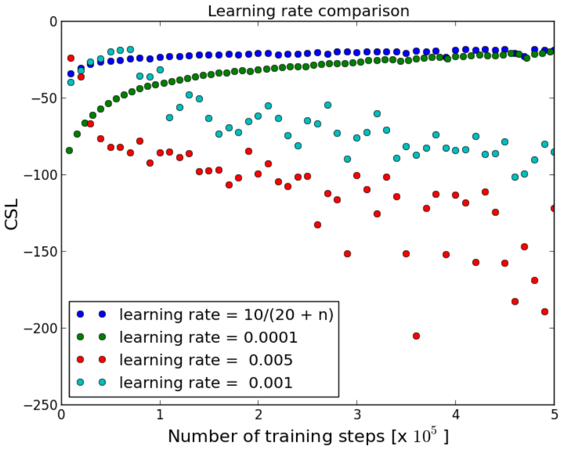

Figure 3: The figure shows the log-likelihood of an RBM which has been trained using the PCD algorithm with minibatch size 3 and varying learning rates. High learning rates lead to a decrease in the log-likelihood after a few training steps [Fischer and Igel, 2010]. It is the same effect that can be observed in Figure 2.

The influence of the minibatch size can be seen in figure 4. The training benefits from minibatch sizes > 1. In our case, a minibatch size 30, that is the whole training dataset, proofs to lead to the best results. We observe that both the overall likelihood increases and the fluctuations reduce with increasing an minibatch size. We compared different variants of clustering images to subsets. The purple data points show minibatches of size 3 with one representative of each digit in a subset. The green points show minibatches that contain only one class of digits per subset. We can not see a clear difference in both ways to create a minibatch from our data. This could be due to the small data set we used. We expect the results to differ significantly if larger data sets and larger minibatch sizes are used.

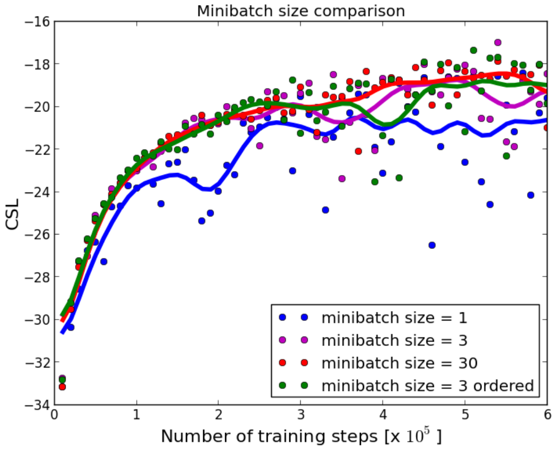

Figure 4: Performance of a PCD trained RBM with varying minibatch sizes. The colored lines show convolutions of the respective data with a gaussian. The training improves with increasing minibatch sizes.

We used our findings to again compare the training methods with the metaparameters that proofed to achieve the best results in the previous experiments. Figure 5 shows a comparison between CD-1, CD-10 and PCD with minibatch size 30 and an adaptive learning rate. CD-1 performs unsatisfactory which might be due to the large initialization value of η. The persisting strong fluctuations, although the learning rate monotonically decreases, hint, that no stable equilibrium could be found. The large variance of the data points most likely stems from the sample based nature of the CSL. CD-10 performs nearly as good as PCD but since it requires 10 steps of Gibbs sampling between each parameter update, it is roughly 10 times slower than PCD which requires the same amount of time as plain CD-1.

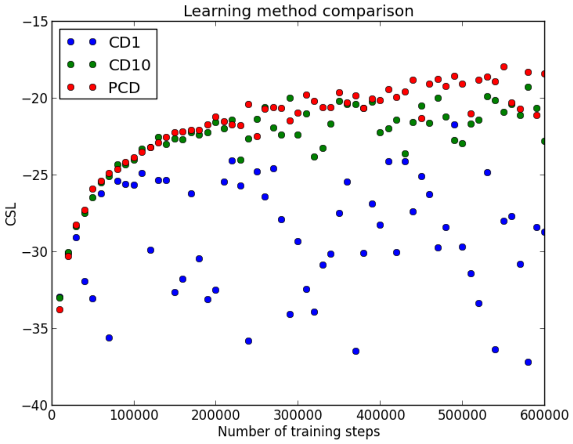

Figure 5: The figure shows a comparison of the different learning methods CD-1, CD-10 and PCD with $\eta = \frac{10}{20+n}$ and minibatch size 30. It can be seen that PCD outperforms both CD-1 and CD-10

4 Discussion

In the framework of this internship, we conducted experiments with restricted Boltzmann machines. In particular, the metaparameters involved in the training process of RBMs have been examined.
We compared CD-k with PCD and concluded that CD-1 might be an oversimplification when used with complex training patterns like the MNIST dataset. PCD provides an efficient improvement to the contrastive divergence algorithm that does not require more training time than CD-1. CD-10 leads to slightly lower log-likelihoods than PCD. Additionally, if we take the time needed to train the model into account, PCD clearly outperforms CD-10.
Furthermore, we conclude that the usage of minibatches can help to improve training. The adjustment of the learning rate is a crucial factor in learning that can, if set correctly, speed up the learning process and prevent poor results. Further investigation regarding the learning rate would need to be done.
Our measurements do not show any evidence that one method of clustering images to minibatches is superior to the other. Since we would expect that minibatches where the digits are mixed more should lead to better results, further analysis with a larger minibatch size or even a larger training dataset size is needed.
The presented findings will be used to train RBMs for a subsequent bachelor thesis in which we will try to perform completion of digits when they are in the process of being written. Furthermore, the presented learning methods will be implemented with neural sampling [Buesing, Bill, Nessler, and Maass, 2011] and LIF sampling [Petrovici, Bill, Bytschok, Schemmel, and Meier, 2013].

References

L. Buesing, J. Bill, B. Nessler, and W. Maass. Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons. *PLoS Computational Biology*, 7(11):e1002211, 2011.

A. Fischer and C. Igel. Empirical analyis of the divergence of gibbs sampling based learning algorithms for restricted boltzmann machines. *Proceedings of the International Conference on Artificial Neural Networks*, 208-217, 2010.

A. Fischer and C. Igel. Training restricted boltzmann machines: An introduction. *Pattern Recognition*, 47, 2014.

Geoffrey Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1), 2010.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Luziwei Leng. Deep learning architectures for neuromarphic hardware. Msc thesis, Ruprecht-Karls-Universität Heidelberg, 2014.

Mihai A Petrovici, Johannes Bill, Ilja Bytschok, Johannes Schemmel, and Karlheinz Meier. Stochastic inference with deterministic spiking neurons. *arXiv preprint arXiv:1311.3211*, 2013.

Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. ACM, 2008.

Kyunghyun Cho Yoshua Bengio, Li Yao. Bounding the test log-likelihood of generative models. *arXiv*, 2013.