JMLR: Workshop and Conference Proceedings 45:391–406, 2015 ACML 2015

Similarity-based Contrastive Divergence Methods for Energy-based Deep Learning Models

Adepu Ravi Sankar CS14RESCH11001@IITH.AC.IN
Vineeth N Balasubramanian VINEETHNB@IITH.AC.IN
Indian Institute of Technology Hyderabad, Sangareddy, Telangana, India 502285

Editor: Geoffrey Holmes and Tie-Yan Liu

Abstract

Energy-based deep learning models like Restricted Boltzmann Machines are increasingly used for real-world applications. However, all these models inherently depend on the Contrastive Divergence (CD) method for training and maximization of log likelihood of generating the given data distribution. CD, which internally uses Gibbs sampling, often does not perform well due to issues such as biased samples, poor mixing of Markov chains and high-mass probability modes. Variants of CD such as PCD, Fast PCD and Tempered MCMC have been proposed to address this issue. In this work, we propose a new approach to CD-based methods, called Diss-CD, which uses dissimilar data to allow the Markov chain to explore new modes in the probability space. This method can be used with all variants of CD (or PCD), and across all energy-based deep learning models. Our experiments on using this approach on standard datasets including MNIST, Caltech-101 Silhouette and Synthetic Transformations, demonstrate the promise of this approach, showing fast convergence of error in learning and also a better approximation of log likelihood of the data.

Keywords: Deep Learning, Boltzmann Machines, Sampling, Data Similarity

1. Introduction

Deep learning is a sub-area of machine learning which has provided methods to learn features at different layers in a hierarchical manner, inspired by models of the brain. In its early years, Deep Learning (DL) was not so successful due to the requirement of huge computational power to address the complexity of deep hierarchical layers. With the recent availability of large computation power with the advent of GPUs, DL has gained significant traction over the last few years. The applications of DL can be found in many research areas like computer vision, signal processing and natural language processing. DL techniques have achieved state-of-the-art performance on many real-world problems including ImageNet object classification (Krizhevsky et al., 2012), LFW face detection (Taigman et al., 2014), as well as natural language processing on the FrameNet and WordNet datasets (Bordes et al., 2012).

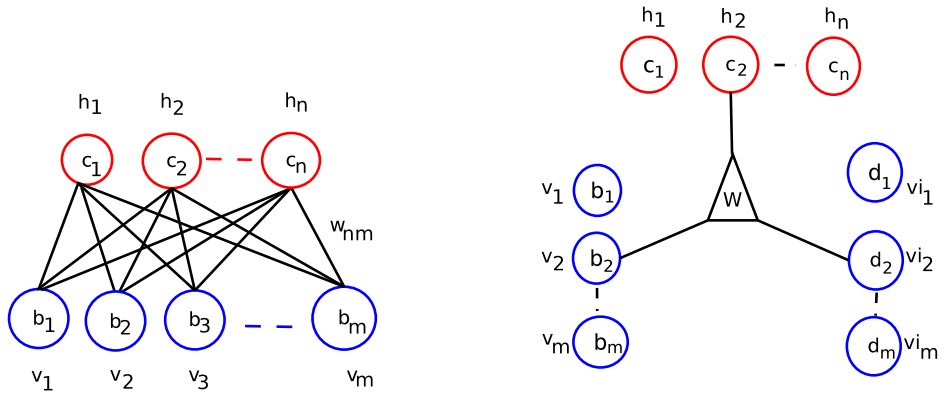
Deep learning models can be broadly classified into energy-based models and non-energy based models. Energy-based models, which form a subclass of Markov random fields, are generally used for unsupervised learning, or as parameter initializers for non-energy based models in supervised learning. For example, the weights of a Restricted Boltzmann Machine, which is an energy-based model, have been used to initialize weights for an Artificial Neural Network (ANN) (Hinton et al., 2006), which is an example of non-energy based deep learning

models. Boltzmann Machines (BMs), Restricted Boltzmann Machine (RBMs) (Fischer and Igel, 2014), Gated Factored Restricted Boltzmann Machines (GFRBMs) (Memisevic and Hinton, 2010) and Convolutional Restricted Boltzmann Machines (CRBMs) (Lee et al., 2009) are all examples of energy- based models. All such energy-based models have two layers: *visible* and *hidden*, with different number of neurons in each layer constituting its architecture. The visible layer is directly connected to the training data and the hidden layer is used to capture the dependencies among the visible layer neurons. The visible and hidden neurons are connected by undirected weights, and there are no intra-layer connections in either of the layers (except in case of BMs). The architecture of connections differentiates the variants of energy-based models.

The loss function, also called the energy, of any energy-based model is given as a linear combination of the activations at the visible and hidden units with their respective weights (e.g. Equation 2 in Section 2). The energy value corresponding to known data is low, and the energy value corresponding to unknown data is high. This concept of energy is motivated from statistical physics, where a material is said to be stable at a lower energy, and unstable at higher energy. Considering energy-based models are generative, the probability density of given training data is proportional to the negative exponential of the model energy. Hence, minimizing the energy of the network in the training phase leads to a model which has learned the probability distribution that originally generated the training data.

Unfortunately, due to the presence of an intractable normalization constant, it is not easy to maximize the probability (or equivalently, minimize the energy) using analytical approaches. To address this issue, Hinton proposed (Hinton, 2002) a method called Contrastive Divergence (CD), which uses Gibbs sampling as an approximation to the gradient of the loss function. The CD method was singularly responsible for making energy-based learning models tractable, and thus made energy-based deep learning models successful in real-world applications. Every iteration of the CD method generates samples from the joint probability distribution of the current model state; and the method unlearns the sample data generated after a pre-defined number of iterations (this sample data is also called *fantasy particle* in literature, and is described further in Section 2). This Gibbs sampling approach suffers from various problems related to exploration of the distribution space, due to which variants of CD have been proposed to sample from the model. These variants of CD (discussed in detail in Section 2.2) include Persistent Contrastive Divergence (PCD) (Tieleman and Hinton, 2009), Fast Persistent Contrastive Divergence (FPCD) (Tieleman and Hinton, 2009), and Tempered MCMC (T-MCMC) (Desjardins et al., 2010). All the proposed variants differ only in the way they generate the fantasy particle for unlearning, and thereby, computation of the gradient approximation.

In this paper, we propose a new approach to improve the performance of CD methods in energy-based deep learning models through better exploration of the distribution space, using dissimilar data. We note that the proposed approach is generic and can be used in conjunction with many variants of CD methods, and is thus relevant to almost all real-world applications that use CD methods to train deep learning models. Our proposed approach, which we call *Dissimilar CD (Diss-CD)* is motivated by the idea to use dissimilar data (dissimilar w.r.t a given training data point) for fantasy sample generation, and thus unlearn the artifacts of an antagonistic sample given a training data sample. The concept of dissimilarity is subjective, and can be varied with respect to a given training data sample.



(a) Restricted Boltzmann Machine

(b) Gated factored RBM

Figure 1: Sample architectures of energy-based models

Our proposition is that unlearning dissimilar data leads to better approximation of the distribution of the training data. We evaluated the proposed approach using different variants of CD on RBMs (as well as Gated Factored RBMs) on different datasets. The performance of the method was studied using standard evaluation criteria that have been used in earlier similar work (Tieleman and Hinton, 2009), (Tieleman and Hinton, 2009), (Desjardins et al., 2010), i.e. test log likelihood and convergence of reconstruction error. Our experiments showed high promise for the proposed *Diss-CD* approach.

2. Background

Energy-based learning models have historically been based on the Boltzmann Machine (BM) architecture. However, considering the challenges of training the BM architecture, which has intra-layer connections, Restricted Boltzmann Machines (RBMs) - which restrict intra-layer connections - have emerged as the fundamental building blocks for energy-based deep learning models. Hence, we present the details of the RBM model in this section, and show how this can be extended to secondary architectures such as Gated Factored RBMs.

2.1. Energy-based Deep Learning Models

An RBM (Fischer and Igel, 2014), shown in Figure 1(a), is a two-layer undirected bipartite Markov random field with m visible units v_j, $j \in \{1, .., m\}$ which are mapped to training data, and n hidden units h_i, $i \in \{1, .., n\}$ which capture the dependencies among the visible units. w_{ij} is the weight connecting v_j and h_i units, b_j is the bias associated with j^{th} visible unit, and c_i is the bias associated with the i^{th} hidden unit. The architecture of RBM makes hidden units conditionally independent given the visible layer inputs, and vice-versa.

Let us consider the case of a Binary-Binary RBM, where $\mathbf{v} \in \{0, 1\}^m$ and $\mathbf{h} \in \{0, 1\}^n$. The joint probability distribution of RBM is defined as:

$$p(\mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{Z} \tag{1}$$

Equation 1 is also called the Gibbs distribution, where E is the energy function of RBM, and Z is the normalization constant. The energy E is defined as:

$$E = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i \tag{2}$$

The objective function in an RBM seeks to maximize the log likelihood of Equation 1. Since units in each layer are conditionally independent given the other layer, the hidden unit activations can be calculated simultaneously using:

$$p(h_i|\mathbf{v}) = sigm(c_i + \sum_{j \in visible} w_{ij}v_j) \tag{3}$$

where *sigm* is the sigmoid activation function. Similarly, the visible layer activations can be calculated as:

$$p(v_j|\mathbf{h}) = sigm(b_j + \sum_{i \in hidden} w_{ij}h_j) \tag{4}$$

Finding the gradient of Eq 1 to maximize the log likelihood is intractable due to the presence of the normalization constant Z. A method called *Contrastive Divergence* (CD) (Tieleman, 2008) is used to approximate the gradient of log likelihood and is given as:

$$\triangle w_{ij} = \eta(\langle h_i v_j \rangle_{data} - \langle h_i v_j \rangle_{\infty}) \tag{5}$$

The term $\langle h_i v_j \rangle_{data}$ signifies the expectation of unit h_i and v_j being simultaneously active together. The term $\langle \rangle_{\infty}$ is obtained by using Gibbs sampling for infinite iterations. The given input is first clamped to the visible units; the hidden unit activations are obtained by using Equation 3 followed by stochastic binarization using a Bernoulli distribution; and the expected value of $h_i v_j$ for all pairs of visible and hidden units are subsequently calculated (called $\langle h_i v_j \rangle_{data}$ indicating that it is calculated with respect to data). The same steps are repeated to then compute the visible unit activations from the hidden unit activations (reverse direction). These visible unit activations are called one-step reconstructions, and this chain is continued by updating the visible and hidden unit activations iteratively. If this chain is repeated for a very long time, the visible and hidden unit activations converge to the term $\langle h_i v_j \rangle_{\infty}$ and the system is said to be in thermal equilibrium at this step (the v_j at this step is called the fantasy particle). The term $\langle \rangle_{data}$, also called the learning phase, specifies the expectation calculated using the given data, and $\langle \rangle_{\infty}$, also called the unlearning phase, denotes the expectation calculated when the network is allowed to run freely using Gibbs sampling.

Few extensions of RBMs have also been proposed over the years. Gated Factored RBM (GFRBM) (Memisevic and Hinton, 2010) is a two-layer energy-based model, where the input layer takes two data points as input as shown in Figure 1(b). The objective of the GFRBM is to learn the conditional probability distribution of the two inputs through the three-dimensional weights. The three-dimensional weights learn the transformations between two input data points. Please refer (Memisevic and Hinton, 2010) for more details. In addition to the aforementioned, other models like Convolutional RBMs (Lee et al., 2009), Auto-Encoders (Baldi, 2012), and Deep Belief Nets (Hinton et al., 2006) have been proposed, which use RBMs as basic building blocks.

2.2. Fantasy Sample Generation Methods

All the architectures discussed above use Gibbs sampling for generating the fantasy sample which is then used in the unlearning phase. The problem with Gibbs sampling is that it needs to be run for a long time. Hinton proposed (Hinton, 2002) a simplification and stated that even if we run the chain for 1 step (called CD-1), i.e. the fantasy sample is selected after one iteration of Gibbs sampling, the learning still works. The learning rule (what was Equation 5) now becomes:

$$\triangle w_{ij} = \eta(\langle h_i v_j \rangle_0 - \langle h_i v_j \rangle_1) \tag{6}$$

where $\langle h_i v_j \rangle_0$ is the same as $\langle h_i v_j \rangle_{data}$ in Equation 5, and is used hereafter for convenience of understanding.

As can be seen from Equation 6, the second term i.e $\langle h_i v_j \rangle_1$, or the fantasy sample, plays a crucial role for effective learning. The generation of fantasy samples is prone to problems such as biased samples (leading to biased estimates of gradient), getting stuck in high-mass probability modes, and poor mixing of the Markov chain. Persistent Contrastive Divergence (PCD) (Tieleman and Hinton, 2009) was proposed as an improvement of CD for better mixing of the modes by using a persistent Markov chain to generate negative samples from model distribution. Instead of starting a new Markov chain for every training data sample, PCD persists the model state from the previous iteration of gradient calculation, and hence the name PCD. Considering that PCD has replaced CD in practical use, we have used PCD as the basis for the rest of this paper.

Another variant of PCD called Fast PCD (FPCD) (Tieleman and Hinton, 2009) was later proposed as an improvement to PCD. FPCD uses two sets of weights: $\theta\_regular$ and θ_fast, where θ_fast is specified using large learning rates leading to fast mixing of the Markov chain. The FPCD is equal to PCD when θ_fast equals zero. A large weight decay is used so that the θ_fast converges to zero very soon. It was shown that such weight updates force the Markov chain to mix faster. A potential disadvantage with FPCD is that the negative samples that are generated could diverge from the invariant model distribution.

Tempered MCMC (Desjardins et al., 2010) was another method that was proposed for better sampling from RBMs. This method belongs to a class of methods called Extended Ensemble Monte Carlo methods which aim to overcome the inability of mixing in multi-modal distributions by Markov chains. Unlike a single persistent Markov chain (as in PCD or FPCD), multiple PCD chains are run in parallel, with each chain initialized at different parameters (or temperatures). Samples are drawn from distributions with different temperatures to promote mixing between multiple modes. Each Markov chain generates a fantasy particle. The swapping of the samples between consecutive chains are carried out using probabilities as computed using Equation 9 (details given in Algorithm 3). The visible-hidden sample pair obtained at the lowest temperature is used for weight update.

In this work, we propose a new fantasy sample generation method called *Dissimilar Contrastive Divergence*(Diss-CD). The proposed Diss-CD method uses dissimilar data for fantasy sample generation, with the premise that using fantasy samples from the true input distribution may be better than generating them using Gibbs sampling. We now describe our approach.

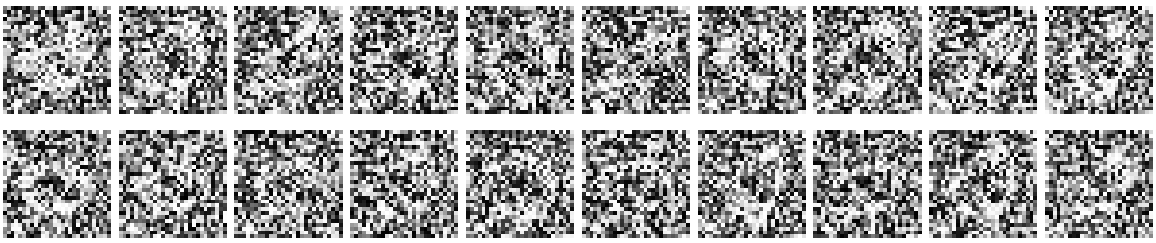


Figure 2: Fantasy particles generated from different training images using CD-1 in the first epoch of RBM training on digit 0 from the MNIST dataset. Almost all images just appear to be noisy versions of the "0" image, motivating us to use dissimilar data for fantasy sample generation.

3. Similarity-based Contrastive Divergence (Diss-CD)

The proposed Diss-CD method obtains its inspiration from the following problem with the CD method. When a Markov chain is initialized with the given training data, the chain eventually settles at an equilibrium, which is influenced by the randomly chosen initial weights of the model. As mentioned earlier, the sample generated at this equilibrium state is called the fantasy sample, and the CD method attempts to lower the probability of generating this fantasy sample (which is also called reconstructed data, since it is obtained from the original data after Gibbs sampling), and increase the probability of generating the original training data.

However, as proposed by Hinton in (Hinton, 2002), the ∞-step CD is generally replaced by a 1-step CD (also called CD-1), as mentioned earlier in Section 2. This implicitly causes the reconstructed data sample (fantasy) to lie in the immediate neighborhood of the training data as shown in Figure 2, where the reconstructions are generally noisy versions of training data. If we now carry out the weight updates, it is very likely that the probability distribution is not modelled effectively across the complete space due to poor mixing of the Markov chain. The mixing is poor because CD using Gibbs sampling will increase the unnormalized probability of generating the training data, and decrease or unlearn the probability of generating the data which, in turn, is also near the training data in the probability space. Thus, fantasy samples that actually exist far away from the current data sample are not captured by the model. Any variant of CD, including the ones discussed in Section 2.2, attempts to address this 'poor mixing' issue. While this issue is mitigated in case of CD-k to an extent, the issue still exists since k is often a small number in practice.

In this work, we propose a new idea to improve mixing in CD-based methods. Given a training data sample, our Diss-CD method breaks the Markov chain for fantasy sample generation by proposing dissimilar data samples as potential start points for fantasy sample generation, and continuing the Markov chain from these new start points. Figure 3 describes the proposed Diss-CD method, where the fantasy generation is initialized from dissimilar data. To illustrate further, Figure 2 shows that the fantasy particles generated using the standard CD, which are subsequently used for unlearning, are often just noisy versions of the data itself. Instead of unlearning the noisy samples generated by the invariant distribution of the corresponding Markov chain, we force the RBM to unlearn the data that is dissimilar to the current data sample. This process also ensures that the Markov chain explores other

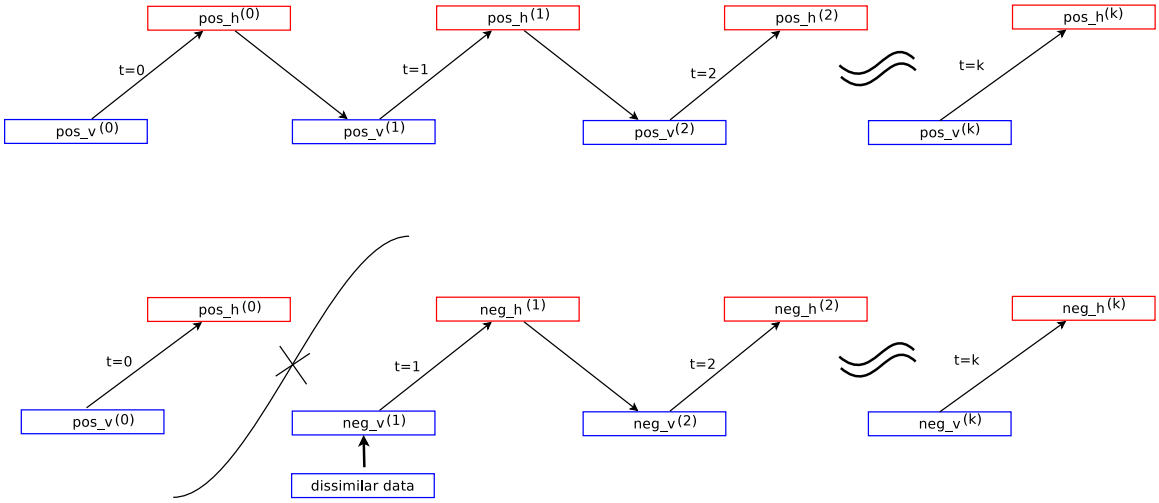


Figure 3: Proposed Diss-CD approach. Top image shows how CD-k is run using Gibbs sampling. In our proposed approach (bottom image), a new Markov chain is started from t=1 using dissimilar data as the starting point for fantasy sample generation

modes in the probability space, and thus builds a more complete model of the data p.d.f. Algorithm 1 describes the proposed method.

Algorithm 1: Dissimilar CD (Diss-CD) Algorithm

Input: $RBM(\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\})$, Training data $\mathbf{S}$, Dissimilar data $\overline{\mathbf{S}}$, Number of Gibbs cycles k, Number of hidden layer units n, Number of visible layer units m
Output: Gradient approximation $\triangle w_{ij}, \triangle b_j, \triangle c_i$ for $i = 1 \cdots n$ and $j = 1 \cdots m$

1. Initialize $\triangle w_{ij} = \triangle b_j = \triangle c_i = 0$ for $i = 1 \cdots n$ and $j = 1 \cdots m$
2. for all $\mathbf{pos\_v} \in \mathbf{S}$, $\mathbf{neg\_v} \in \overline{\mathbf{S}}$ do:
   (a) $\mathbf{pos\_v}^{(0)} \leftarrow \mathbf{S}$, $\mathbf{neg\_v}^{(1)} \leftarrow \overline{\mathbf{S}}$
   (b) for $i = 1 \cdots n$ do: sample $pos\_h_i^{(0)} \sim p(h_i | \mathbf{pos\_v}^{(0)})$
   (c) for $t = 1 \cdots k$ do:
      i. for $i = 1 \cdots n$ do: sample $neg\_h_i^{(t)} \sim p(neg\_h_i | \mathbf{neg\_v}^{(t)})$
      ii. for $j = 1 \cdots m$ do: sample $neg\_v_j^{(t+1)} \sim p(neg\_v_j | \mathbf{neg\_h}^{(t)})$
   (d) for $i = 1 \cdots n$, $j = 1 \cdots m$ do
      i. $\triangle w_{ij} \leftarrow$
      $\triangle w_{ij} + p(pos\_h_i = 1 | \mathbf{pos\_v}^{(0)}) pos\_v_j^{(0)} - p(neg\_h_i = 1 | \mathbf{neg\_v}^{(k)}) neg\_v_j^{(k)}$
      ii. $\triangle b_j \leftarrow \triangle b_j + pos\_v_j^{(0)} - neg\_v_j^{(k)}$
      iii. $\triangle c_i \leftarrow \triangle c_i + p(pos\_h_i = 1 | \mathbf{pos\_v}^{(0)}) - p(neg\_h_i = 1 | \mathbf{neg\_v}^{(k)})$

Algorithm 2: Dissimilar CD (Diss-CD) Algorithm for Fast PCD

Input: $RBM(\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}, \theta\_fast = \{\mathbf{W\_fast}, \mathbf{b\_fast}, \mathbf{c\_fast}\})$, Training data $\mathbf{S}$, Dissimilar data $\overline{\mathbf{S}}$, Number of Gibbs cycles k, Number of hidden layer units n, Number of visible layer units m

Output: Gradient approximation $\triangle\theta = \{\triangle w_{ij}, \triangle b_j, \triangle c_i\}, \triangle\theta\_fast = \{\triangle w_{ij}^{fast}, \triangle b_j^{fast}, \triangle c_i^{fast}\}$ for $i = 1 \cdots n$ and $j = 1 \cdots m$

1. init $\triangle w_{ij} = \triangle b_j = \triangle c_i = 0$ for $i = 1 \cdots n$ and $j = 1 \cdots m$
2. for all $\mathbf{pos\_v} \in \mathbf{S}, \mathbf{neg\_v} \in \overline{\mathbf{S}}$ do:
   (a) $\mathbf{pos\_v}^{(0)} \leftarrow \mathbf{S}$, $\mathbf{neg\_v}^{(1)} \leftarrow \overline{\mathbf{S}}$
   (b) for $i = 1 \cdots n$ do: sample $pos\_h_i^{(0)} \sim p(h_i | \mathbf{pos\_v}^{(0)})$
   (c) for $t = 1 \cdots k$ do
      i. for $i = 1 \cdots n$ do: sample $neg\_h_i^{(t)} \sim p(neg\_h_i | \mathbf{neg\_v}^{(t)}; \theta\_fast + \theta)$
      ii. for $j = 1 \cdots m$ do: sample $neg\_v_j^{(t+1)} \sim p(neg\_v_j | \mathbf{neg\_h}^{(t)}; \theta\_fast + \theta)$
   (d) for $i = 1 \cdots n, j = 1 \cdots m$ do
      i. $\triangle w_{ij} \leftarrow$ $\triangle w_{ij} + p(pos\_h_i = 1 | \mathbf{pos\_v}^{(0)}) pos\_v_j^{(0)} - p(neg\_h_i = 1 | \mathbf{neg\_v}^{(k)}) neg\_v_j^{(k)}$
      ii. $\triangle b_j \leftarrow \triangle b_j + pos\_v_j^{(0)} - neg\_v_j^{(k)}$
      iii. $\triangle c_i \leftarrow \triangle c_i + p(pos\_h_i = 1 | \mathbf{pos\_v}^{(0)}) - p(neg\_h_i = 1 | \mathbf{neg\_v}^{(k)})$
   (e) $\triangle\theta\_fast = \epsilon \cdot \theta\_fast + \triangle\theta$

To explain our idea further, maximizing the likelihood using CD-k is equivalent to minimizing the Kullback-Leibler divergence between data distribution P^0 and the model equilibrium distribution P_θ^k (where $\theta = \{\mathbf{W,b,c}\}$) (Hinton, 2002).

$$KL(P^0 \| P_\theta^k) = -H(P^0) - \langle log P_\theta^k \rangle_{P^0} \tag{7}$$

where the first term in Equation 7 denotes entropy over the data distribution and the second denotes the entropy over the model-generated distribution. The minimization of the KL divergence involves minimizing the second term w.r.t the model parameters as the first term is independent. Minimization of the second term results in:

$$\left\langle \frac{\partial log P_\theta^k(D)}{\partial \theta} \right\rangle_{P^0} = \left\langle \frac{\partial log f_\theta}{\partial \theta} \right\rangle_{P^0} - \left\langle \frac{\partial log f_\theta}{\partial \theta} \right\rangle_{P^k} \tag{8}$$

The term $log f_\theta$ is a random variable sampled from the distribution with parameters given by θ. As discussed in Section 2.2, Hinton proposed CD-1 to approximate the second term of Equation 8, which is commonly used in deep learning applications. CD-1 minimizes the difference between $KL(P^0 \| P_\theta^\infty)$ and $KL(P_\theta^1 \| P_\theta^\infty)$, where P_θ^1 is the reconstruction of data generated using one Gibbs sampling. This approximation suffers from the problem of P_θ^1 being very close to that of the data distribution itself rendering the Markov chain incapable of exploring other modes in the data distribution. As mentioned earlier, this

issue is mitigated to some extent when using CD-k, considering k Gibbs sampling steps are performed. In the proposed approach, we replace P^1_θ by a sample from the distribution that we know is dissimilar to P^0 which can coerce the Markov chain to visit other modes when using PCD for k steps. By thus modeling the distribution across a larger probability space, this approach provides an opportunity to reduce future generalization error. Please see Algorithm 1 for more details.

While Algorithm 1 shows how Diss-CD can be used in standard PCD (or CD-k), the proposed approach can also be viewed as a generic framework that can be applied along with other Gibbs sampling variants such as Fast PCD and Tempered MCMC (described in Section 2.2). Algorithm 2 describes how Diss-CD is adapted to the Fast PCD method. The fantasy sample generation in Fast PCD using fast weights is done using dissimilar data. The objective of fast weights in Fast PCD is to explore many modes with a high learning rate for fantasy sample generation. With the help of dissimilar data being explicitly supplied to fast weights, there will be an acceleration in the diversity of modes being visited leading to a better approximation of the probability distribution of the training data.

Algorithm 3: Dissimilar CD (Diss-CD) Algorithm for Tempered MCMC

Input: RBM($\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$), Training data $\mathbf{S}$, Dissimilar data $\overline{\mathbf{S}}$, Number of Gibbs cycles k, Number of hidden layer units n, Number of visible layer units m, list of temperatures T
Output: Gradient approximation $\triangle w_{ij}, \triangle b_j, \triangle c_i$ for $i = 1 \cdots n$ and $j = 1 \cdots m$

1. init $\triangle w_{ij} = \triangle b_j = \triangle c_i = 0$ for $i = 1 \cdots n$ and $j = 1 \cdots m$
2. for all the $\mathbf{pos\_v} \in \mathbf{S}$, $\mathbf{neg\_v} \in \overline{\mathbf{S}}$ do
   (a) for $r = 1 \cdots length(T)$ do
      i. $\mathbf{pos\_v}_r^{(0)} \leftarrow \mathbf{S}, \mathbf{neg\_v}_r^{(1)} \leftarrow \overline{\mathbf{S}}$
      ii. for $i = 1 \cdots n$ do sample $pos\_h_{r,i} \sim p(pos\_h_{r,i} | \mathbf{pos\_v}_r^{(0)})$
      iii. for $t = 1 \cdots k$ do
         A. for $i = 1 \cdots n$ do sample $neg\_h_{r,i}^{(t)} \sim p(neg\_h_{r,i} | \mathbf{neg\_v}_r^{(t)})$
         B. for $j = 1 \cdots m$ do sample $neg\_v_{r,j}^{(t+1)} \sim p(neg\_v_{r,j} | \mathbf{neg\_h}_r^{(t)})$
   (b) for $r = 1 \cdots length(T)$ do
      i. swap $(\mathbf{neg\_v}_r^{(k)}, \mathbf{neg\_h}_r^{(k)})$ and $(\mathbf{neg\_v}_{r-1}^{(k)}, \mathbf{neg\_h}_{r-1}^{(k)})$ using T_r and T_{r-1} with probability given by Equation 9
   (c) **for** $i = 1 \cdots n$, $j = 1 \cdots m$ **do**
      i. $\triangle w_{ij} \leftarrow$
      $\triangle w_{ij} + p(pos\_h_i = 1 | \mathbf{pos\_v}^{(0)}) pos\_v_j^{(0)} - p(neg\_h_i = 1 | \mathbf{neg\_v}^{(k)}) neg\_v_j^{(k)}$
      ii. $\triangle b_j \leftarrow \triangle b_j + pos\_v_j^{(0)} - neg\_v_j^{(k)}$
      iii. $\triangle c_i \leftarrow \triangle c_i + p(pop\_h_i = 1 | \mathbf{pos\_v}^{(0)}) - p(neg\_h_i = 1 | \mathbf{neg\_v}^{(k)})$

Similarly, Algorithm 3 describes the adaptation of Diss-CD to the Tempered MCMC method. Instead of running the parallel Markov chains with respect to training data, we propose to start running them by using dissimilar data. Starting from the Markov chain with highest temperature, the fantasy particles in consecutive chains are compared and swapped with a probability given by Equation 9, where E is the energy function and T_r is temperature at which r^{th} chain is operating. The fantasy particle at the lowest temperature chain after swaps is used for weight update.

$$\min\left(1, exp\left(\left(\frac{1}{T_r} - \frac{1}{T_{r-1}}\right).(E(v_r, h_r) - E(v_{r-1}, h_{r-1}))\right)\right) \tag{9}$$

4. Experiments

A detailed study of our proposed approach on various datasets using different energy-based models was carried out using PCD and its variants. The proposed Diss-CD approach was first evaluated on the basic building block of energy-based deep learning architectures, viz. RBMs. To study the change in performance on using Diss-CD against the standard PCD (or its variants), we used the convergence error (over epochs) as well as the log likelihood on test data. These metrics have been used in earlier work that proposed PCD variants (Tieleman and Hinton, 2009), (Tieleman and Hinton, 2009), (Salakhutdinov, 2010), (Desjardins et al., 2010). The convergence is observed as the rate of decrease in the squared reconstruction error of training data in each epoch. Instead of the true log likelihood (which can be calculated only for small networks), we use the approximated log likelihood on test data which is computed using Equation 2 (likelihood is simply the negative exponential of the energy without normalization constant). This approximation assumes that the normalization constant is fairly comparable across the networks for a given experiment (we empirically verified this on small networks with upto 15 hidden nodes by exhaustively enumerating the configurations). For PCD, the length of persistence k is taken as 3 as in (Hinton, 2002). In Tempered MCMC, the number of parallel Markov chains M that are run are 11 as in (Desjardins et al., 2010), and each Markov chain is run with a persistence of 3. All the experiments are repeated for 5 runs and the average performance across the runs are reported. Subsequently, we also validated the performance of Diss-CD on another energy-based deep learning architecture, the Gated Factored RBM (GFRBM), discussed in Section 2 (Memisevic and Hinton, 2010). The proposed Diss-CD method showed promising performance in all the models under all variants of sampling.

4.1. Datasets

The proposed method is evaluated on datasets that are commonly used to verify deep learning methods as in (Tieleman and Hinton, 2009), (Tieleman and Hinton, 2009), (Salakhutdinov, 2010), (Desjardins et al., 2010). The datasets include MNIST, Caltech-101 Silhouettes, and the Synthetic Transformation dataset[1]. The datasets are described in Table 1.

1. http://yann.lecun.com/exdb/mnist, https://people.cs.umass.edu/~marlin/data.shtml, http://www.cs.toronto.edu/~rfm/factored

Dataset Name	Total Size	Description	Size of each feature	Tested on
MNIST	60000	Images of digits(0 to 9)	28x28 Bin image	RBM
Caltech-101 Silhouttes	8671	Images of convex shapes (black on white background)	28x28 Binary image	RBM
Synthetic Transformations	10000 pairs	Each pair of image differs by a rotation constant	2x13x13 Binary image	GFRBM

Table 1: Description of datasets used to test Diss-CD

4.2. Choosing Dissimilar data

The choice of dissimilar data for Diss-CD is subjective. Much of the knowledge about choosing the type of dissimilar data comes from knowledge about the dataset. Although it is difficult to come up with concrete rules for choosing the dissimilarity, we lay down a few ground rules in Table 2 for choosing dissimilar data based on the type of dataset. The thumb rule is to choose the type of data that you want to unlearn as dissimilar data, to leverage the learning mechanism in energy-based deep models. In future work, we plan to explore the concept of adaptively choosing dissimilar data for every training example.

Type of Dataset	Potential dissimilar data
Classification problem	Data from other classes
Modelling shapes of images	Images with other shapes from/other datasets
Object detection	Dissimilar objects from other datasets
Anomaly detection in videos	Normal videos

Table 2: Heuristics to choose dissimilar data for various types of data

4.3. Experiments with RBM

A vanilla standard RBM with 500 nodes in the hidden layer is used. All the hyperparameters i.e number of epochs, learning rate, momentum, weight decay are kept constant for all experiments. The convergence of error and log likelihood are reported at the end of the 40^{th} epoch in all RBM experiments (and the evolution over these epochs is reported where possible). As the performance of the proposed method is dependent on the type of dissimilarity used, the performance is studied under different scenarios of dissimilarity: (i) Dissimilar data from same dataset; and (ii) Dissimilar data from another dataset.

4.3.1. RBM trained with dissimilar data from other classes of same dataset

We studied the performance of RBMs trained on MNIST and Caltech-101 silhouette datasets, where dissimilar data for both experiments are chosen from the respective dataset.

MNIST: Given data from all the digits in the MNIST dataset, an RBM is trained to learn the distribution of each digit individually, by choosing the remaining digits randomly as dissimilar data. For example, when the RBM is trained to learn digit 0, all the remaining digits 1 to 9 are considered as dissimilar data. Figure 4 presents the error to which the

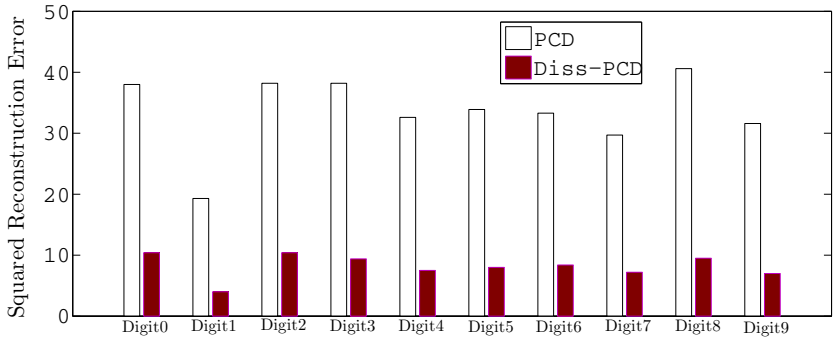


Figure 4: Squared reconstruction error (x10^3) at the end of training for each MNIST digit

RBM converged for learning each digit in MNIST using PCD and Diss-PCD. The results show that the convergence is better in case of Diss-PCD. The test log likelihood of each digit on the trained RBMs is shown in Table 3.

The results of the evaluation of the proposed approach against other variants of PCD i.e Fast-PCD and T-MCMC are shown in Table 4[2]. The error convergence for digit 0 is shown in Figure 5. A similar trend was observed for other digits, and the corresponding results are not presented here for space constraint reasons. Also, an inferential statistic t-test (independent samples test, as we are comparing log likelihood of PCD and Diss-PCD at the end of training) was conducted on the reported log likelihood values of PCD and Diss-PCD for the Digit 0 experiment across 20 trials, and we found the difference of the log likelihood values in the two methods to be statistically significant with a confidence level of 99%.

Digit	PCD LL	Diss-PCD LL
Digit 0	552	**2369**
Digit 1	240	**1237**
Digit 2	455	**1766**
Digit 3	457	**1665**
Digit 4	416	**1772**
Digit 5	384	**1541**
Digit 6	382	**1708**
Digit 7	399	**1873**
Digit 8	485	**1960**
Digit 9	442	**1537**

Table 3: Log likelihood (LL) (higher is better) of each digit in MNIST at the end of training

Caltech-101 Silhouettes: In this experiment, a standard RBM (with 500 hidden units) is trained on a subset (of about 1000 images with convex shapes) of images from the Caltech-101 Silhouettes dataset, and the experiment is run with non-convex shapes from the same dataset as dissimilar data. The squared reconstruction error at the end of training is shown in Table 5. Evidently, the proposed approach showed promise over all known variants of PCD, although it showed a slightly lower log likelihood in case of T-MCMC. This can be explained due to the presence of multiple Markov chains in T-MCMC, which may anyway be causing it to explore other modes of the distribution. We note, however, that T-MCMC is not used often in practice due to the computational overhead (Desjardins et al., 2010).

2. We note that since the test log likelihoods are unnormalized, i.e. the normalization constants are ignored, the log likelihood values should not be compared across variants of PCD as the normalization constant can differ in each case.

Digit	Fast PCD		Diss Fast PCD		T-MCMC		Diss TMCMC	
	Error	Likelihood	Error	Likelihood	Error	Likelihood	Error	Likelihood
Digit 0	37.7	601	**10.9**	**2012**	37	599	**28**	**1522**
Digit 1	21.7	513	**3.9**	**1410**	19.1	402	**18.1**	**1392**
Digit 2	38.2	480	**10.5**	**1907**	34.2	522	**31**	**1892**
Digit 3	37.8	475	**9.7**	**2001**	38.4	468	**35.6**	**1981**
Digit 4	32.5	430	**7.7**	**1788**	**32.5**	445	34	**1922**
Digit 5	33.9	422	**8.3**	**1585**	34	422	**21**	**1478**
Digit 6	40	700	**8.1**	**1933**	33	470	**24**	**1496**
Digit 7	35	610	**7.1**	**1635**	**29**	416	32	**1544**
Digit 8	53	781	**9.5**	**1924**	40	465	**32**	**2033**
Digit 9	38	596	**7.2**	**1811**	31	434	**26**	**1546**

Table 4: Squared reconstruction error(x10^3) and log likelihood (higher is better) for each digit in MNIST dataset using Fast PCD vs Diss-Fast PCD, and Tempered MCMC(T-MCMC) vs Diss-TMCMC

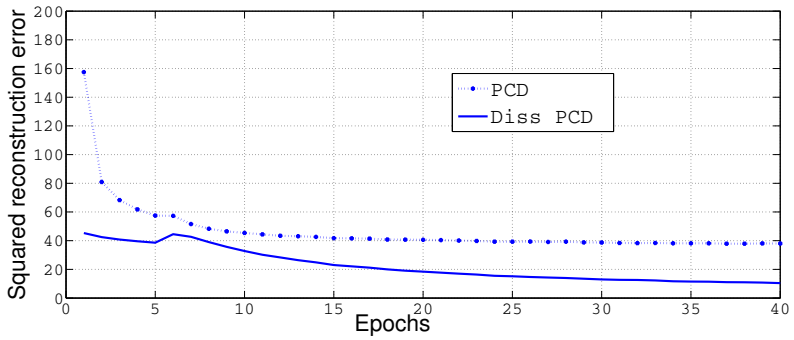


Figure 5: Convergence of error^2(x10^3) in RBM over epochs for digit 0 in MNIST dataset.

4.3.2. RBM trained with dissimilar data chosen from different dataset

The experiments in this section study the performance of the proposed approach when the dissimilar sample is chosen from a completely different dataset. We studied the performance of RBM when trained on the Caltech-101 Silhouettes dataset, where dissimilar data are

CD Variant	Without Diss-CD		With Diss-CD	
	Error	Likelihood	Error	Likelihood
PCD	6.8	3721	**5.3**	**3906**
Fast PCD	7.7	1271	**6.6**	**3801**
Tempered MCMC	6.5	**3778**	**6.4**	3712

Table 5: Squared reconstruction error(x10^3) and likelihood after convergence on a subset of Caltech 101 Silhouettes dataset with dissimilar data from Caltech-101

chosen from the MNIST dataset. As before, an RBM is trained on Caltech-101 Silhouettes on a subset (of about 1000 images with convex shapes) of images, with MNIST data as dissimilar. The results are reported in Table 6.

PCD Variant	Without Diss-CD		With Diss-CD	
	Error	Likelihood	Error	Likelihood
PCD	6.8	3721	**6.2**	**3812**
Fast PCD	7.7	1271	**6.6**	**2713**
Tempered MCMC	**6.5**	3778	**6.5**	**3798**

Table 6: Squared reconstruction error(x10^3) and log likelihood at the end of training on a subset of Caltech-101 Silhouettes data with dissimilar data from MNIST

4.4. Gated Factored RBM

The GFRBM is trained on the Synthetic Transformation (ST) dataset as in (Memisevic and Hinton, 2010). The input for GFRBM is a pair of 13x13 binary images, where one image is a left shift of the other. The dissimilar data for this model is a pair of images constituted by a different image and its randomly shifted version. Figure 6 shows the squared reconstruction error at the end of training in this experiment, and also the log likelihood modelled by GFRBM on the aforementioned left shift dataset (subset of the ST dataset). It can be seen that the likelihood of test data is higher in case of proposed Diss-PCD.

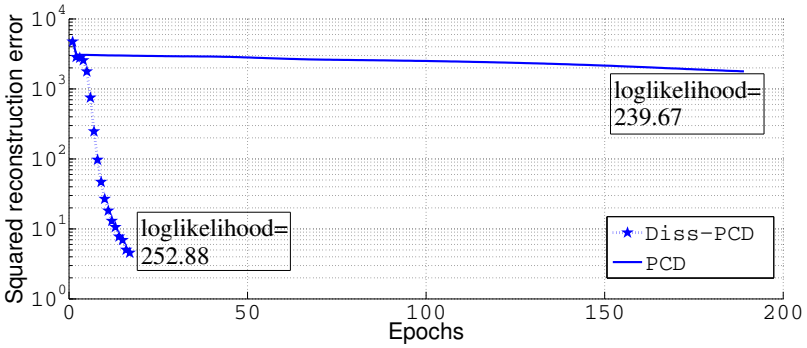


Figure 6: Squared reconstruction error over epochs and log likelihood at the end of training (value in box) in GFRBM for ST dataset (Diss-PCD converged after a few epochs, which is why the curve is truncated)

5. Discussion

We note here that in all the aforementioned experiments, the dissimilar data was used only for the first 5 epochs to ensure the Markov chain visits other modes of the probability distribution, after which it was allowed free exploration. In Figure 5, although the error increased slightly after 5 epochs of Diss-PCD, the error is still upper-bounded by PCD, and quickly decreases again after a couple of epochs. From a different viewpoint, our proposed

method can hence be viewed as a "supervised pre-training" step before letting the model run freely. To describe this further, Table 7 analyzes the effect of learning digit 2 of the MNIST dataset by training a standard RBM using Diss-PCD for varying number of initial epochs (after which the model runs freely). We find that when Diss-PCD is used for too many epochs, there is a deterioration in error performance, since there may be too many shifts in the modes explored by Gibbs sampling. We overcome this issue, by using Diss-CD only for the first few epochs (5, in our experiments in this work).

Dissimilarity epochs		
5 epochs	10 epochs	40 epochs
10.2	12.8	80.4

Table 7: Squared reconstruction error(x10^3) at the end of training (40th epoch) when using Diss-PCD at different levels

Also, while it is evident from the experimental results that the proposed Diss-CD approach has merit in being used in energy-based deep learning models, one of the key issues of the proposed Diss-CD approach is that the choice of dissimilar data can be subjective. In most classification problems, we believe that this choice is often quite intuitive. However, we conducted an experiment to study the significance of the effect of choice of dissimilarity, while training an RBM on digit 1. The RBM, when trained with dissimilar samples from other digits, reported a log likelihood of 1237 (Table 3). When the same experiment is conducted with dissimilar data consisting of only digit seven (which is similar to one visually), Diss-CD reported a lower log likelihood of 952, clearly implying that the choice of dissimilarity plays an important role. We also note that if the dissimilar data is reasonably similar to that of training data, our Diss-CD effectively becomes PCD. Hence, it is possible to ensure that the performance of the proposed approach is always bounded below by the performance of PCD itself (or any other variant it is used with).

6. Conclusions and Future Work

A novel training method for energy-based deep learning methods based on using dissimilar data to improve Contrastive Divergence performance is proposed. The proposed Diss-CD approach allows the system to model the p.d.f of the training data better than using CD alone, by allowing the Markov chain to visit different modes in the probability space of training data, thus addressing a common issue in traditional Gibbs sampling. The proposed method is evaluated on many datasets with different variants of PCD, and also on two kinds of energy-based deep learning models, viz. RBMs and GFRBMs. Our experimental studies show promise in the use of this approach. Our future work includes: (i) although it is not possible to prove theoretical bounds on CD performance considering it is an approximation itself (as in earlier work on CD variants, (Tieleman and Hinton, 2009), (Salakhutdinov, 2010), (Desjardins et al., 2010)), we plan to study the possibility of proving bounds on the performance of the proposed Diss-CD approach; (ii) considering similarity plays a big role in this idea, we plan to explore different similarity metrics (or even learning one adaptively for a given application) to improve Diss-CD performance; (iii) considering that T-MCMC occasionally performed better than our approach, we plan to investigate the use of parallel Markov chains with dissimilar data in our future work; and (iv) we plan to extend our study to include other energy-based models such as CRBMs.

References

Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *ICML Unsupervised and Transfer Learning*, volume 27 of *JMLR Proceedings*, pages 37–50. JMLR.org, 2012.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *In Proceedings of 15th International Conference on Artificial Intelligence and Statistics*, 2012.

Guillaume Desjardins, Aaron C. Courville, and Yoshua Bengio. Adaptive parallel tempering for stochastic maximum likelihood learning of rbms. *CoRR*, abs/1012.3476, 2010.

Asja Fischer and Christian Igel. Training restricted boltzmann machines: An introduction. *Pattern Recognition*, 47(1):25–39, 2014.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, August 2002. ISSN 0899-7667.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006. ISSN 0899-7667.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th International Conference on Machine Learning*, pages 609–616, 2009.

Roland Memisevic and Geoffrey E. Hinton. Learning to represent spatial transformations with factored higher-order boltzmann machines. *Neural Comput.*, 22(6):1473–1492, June 2010. ISSN 0899-7667.

Ruslan Salakhutdinov. Learning deep boltzmann machines using adaptive mcmc. In Johannes Frnkranz and Thorsten Joachims, editors, *ICML*, pages 943–950. Omnipress, 2010. ISBN 978-1-60558-907-7.

Y. Taigman, Ming Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1701–1708, June 2014.

Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1064–1071, New York, NY, USA, 2008. ACM.

Tijmen Tieleman and Geoffrey Hinton. Using fast weights to improve persistent contrastive divergence. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1033–1040, New York, NY, USA, 2009. ACM.