Wasserstein Training of Deep Boltzmann Machines

Chaoyang Wang [1] Tian Tong [2] Yang Zou [2]

Abstract

Training various probabilistic graphical models, such as restricted Boltzmann machines (RBMs), deep Boltzmann machines (DBMs), is usually done by maximum likelihood estimation, or equivalently minimizing the Kullback-Leibler (KL) divergence between the data distribution and the model distribution. On the other hand, Wasserstein distance, also known as optimal transport distance and earth mover's distance, is a fundamental distance to quantify the difference between probability distributions. Due to its good properties like smoothness and symmetry, Wasserstein distance aroused numerous researchers' interests in machine learning and computer vision. In this project, we proposed a general Wasserstein training method for graphical models by replacing the standard KL-divergence with Wasserstein distance as novel loss functions. And we developed Wasserstein training of RBMs and DBMs as specific examples. Finally, we experimentally explored Wasserstein training of RBMs and DBMs for digit generation with MNIST dataset, and showed the superiority of Wasserstein training compared to traditional KL-divergence training.

1. Introduction

Boltzmann machine is a family of stochastic recurrent neural networks and Markov Random Fields. It has applications in dimensionality reduction, classification, collaborative filtering, feature learning and topic modelling (Hinton & Salakhutdinov, 2006; Larochelle & Bengio, 2008; Salakhutdinov et al., 2007; Coates et al., 2010; Hinton & Salakhutdinov, 2009). Boltzmann machines have many variants, such as Deep Boltzmann Machines (DBMs) (Salakhutdinov & Hinton, 2009), Restricted Boltzmann Machines (RBMs) (Salakhutdinov et al., 2007), Conditional Restricted Boltzmann Machines (CRBMs) (Taylor & Hinton, 2009), etc.

Learning in Boltzmann machines is usually done by maximum likelihood estimation, which is equivalent to minimizing the Kullback-Leibler Divergence (KL-divergence) between the empirical distribution and the model distribution.

In the world of high dimensions, KL-divergence will fail to work when the two distributions do not have non-negligible common supports, which happens commonly when dealing with distributions supported by lower dimensional manifolds. In such cases, another kind of probability distance, Wasserstein distance, has gained lots of attention in machine learning and computer vision (Cuturi, 2013), (Doucet, 2014) (Montavon et al., 2016), (Arjovsky et al., 2017) developed a training method for a restricted Boltzmann machine with the Wasserstein distance as the loss function, and showed the power of Wasserstein training in RBM through its application in digit generation, data completion and denoising. The Wasserstein training of RBM leads to better generative properties compared to traditional RBMs. (Arjovsky et al., 2017) introduced the Wasserstein distance for Generative Adversarial Networks (GAN) to improve learning stability. (Frogner et al., 2015) defined Wassserstein distance as the loss function for supervised learning in the label space. Theoretically, the Wasserstein distance is a well-defined distance metric even for non-overlapping distributions, and has the advantages of continuity with respect to the parameters (Arjovsky et al., 2017).

In this project, we propose a general training method to minimize the Wasserstein distance, and discuss specific examples on training the RBMs and DBMs. In experiments, we evaluated the validity of Wasserstein training of the RBMs and DBMs in digit generation with MNIST dataset. The organization of the report is as follows. Section 2 gives a brief introduction to the definition of Wasserstein distance, its duality and comparison with KL-divergence. Section 3 presents the theory of Wasserstein training. In section 4 we focus on Wasserstein RBM while section 5 focuses on Wasserstein DBM. Section 6 considers the practical implementation of Wasserstein training with the KL regularization. In section 7, we experimentally demonstrate Wasserstein Boltzmann machines' validity in digit

[*] Equal contribution [1] Robotics Institute [2] Department of Electrical & Computer Engineering. Correspondence to: Tian Tong <ttong1>, Chaoyang Wang <chaoyanw>, Yang Zou <yzou2>.

generation with MNIST dataset. Section 8 gives our final conclusion.

2. Background

2.1. Wasserstein Distance

Given a complete separable metric space (M, d), let p and q be absolutely continuous Borel probability measures on M. Consider π as a joint probability measure on $M \times M$ with marginals p and q, also called a transport plan:

$$\begin{aligned} &\pi(A \times M) = p(A) \quad \text{for any Borel subset } A \subseteq M; \\ &\pi(M \times C) = q(B) \quad \text{for any Borel subset } B \subseteq M. \end{aligned} \tag{1}$$

For $d(x, x')$ as a distance metric on $M \times M$, the Wasserstein distance between p and q is defined as

$$W(p, q) = \min_{\pi \in \Pi(p,q)} E_\pi d(x, x'), \tag{2}$$

where $\Pi(p, q)$ denotes the marginal polytope of p and q. A transport plan π is optimal if (2) achieves its infimum. Optimal transport plans on the Euclidean spaces are characterized by the push forward measures. Practically, a linear program is usually adopted to solve the Wasserstein distance. However, when the dimension of probability distribution domain is high, solving the linear programming becomes intractable. Recently, (Cuturi, 2013) and (Doucet, 2014) proposed an efficient approximation of Wasserstein distances along with their derivatives. The approximation plays an important role in the practical implementation of these computations.

2.2. Kantorovich Duality

Under the case that p and q are discrete distributions, the Wasserstein distance can be written as a linear programming as

$$\begin{aligned} W(p, q) &= \min_\pi \sum_{x,x'} \pi(x, x') d(x, x') \\ &\text{s.t. } \sum_{x'} \pi(x, x') = p(x), \sum_x \pi(x, x') = q(x'), \\ &\quad \pi(x, x') \geq 0. \end{aligned} \tag{3}$$

Its dual form can be written as

$$\begin{aligned} W(p, q) &= \max_{\alpha,\beta} \sum_x \alpha(x) p(x) + \sum_{x'} \beta(x') q(x') \\ &\text{s.t. } \alpha(x) + \beta(x') \leq d(x, x'). \end{aligned} \tag{4}$$

The primal and dual forms have an interesting intuitive interpretation (Villani, 2008). Consider $p(x)$ as the amount of bread produced in bakery located at x, $q(x')$ as the amount of bread sold in café located x', $d(x, x')$ as the cost to transport unit bread from x to x', and $\pi(x, x')$ as the transport plan, i.e., the amount of bread to transport from x to x'. The primal problem can be interpreted as the minimum cost to transport all bread from bakeries to cafés. Furthermore, assume that there is an agent who wants to compete for business with the transportation department. He buys the bread from bakery located at x with unit price $-\alpha(x)$, and sells the bread to café located at x' with unit price $\beta(x')$. The agent hopes to keep competitive, meaning that his net price is lower than the transportation cost, i.e., $\alpha(x) + \beta(x') \leq d(x, x')$. Therefore, the dual problem can be interpreted as the maximum profit of the agent while keeping competitive. If the agent chooses the optimal pricing scheme, he will gain the same as the transportation department, i.e., the dual achieves the primal.

2.3. Comparison between Wasserstein Distance and KL-divergence

The KL-divergence is defined as

$$KL(p, q) := \int_M \log\left(\frac{p(x)}{q(x)}\right) p(x) dx. \tag{5}$$

KL-divergence is asymmetric and thus not a valid distance. From the definition of KL-divergence, we know that if the supports of p and q do not have non-negligible intersections, which means that the probabilities in the two distributions cannot be both non-zero except for some negligible regions, the KL-divergence will become undefined or infinite. The following example could illustrates the idea.

Let $Y \sim U[0, 1]$ be the uniform distribution on the unit interval. Let p_0 be the distribution of $(0, Y) \in R^2$, uniform on a straight horizontal line segment passing through the origin; p_θ be the distribution of (θ, Y) with θ as a parameter. Fig. 1 illustrates both distributions.

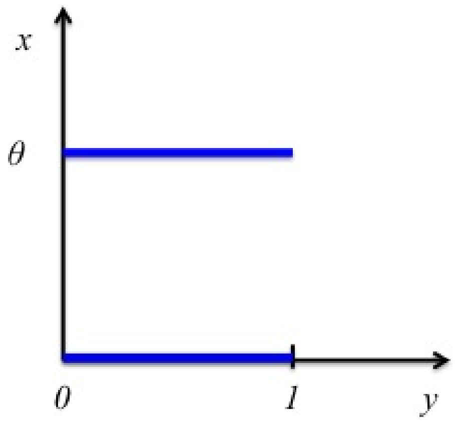


Figure 1. Example for Wasserstein distance vs. KL-divergence.

The Wasserstein distance between p_0 and p_θ is

$$W(p_0, p_\theta) = |\theta|, \tag{6}$$

while the KL-divergence is

$$KL(p_0, p_\theta) = \begin{cases} \infty & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases}. \tag{7}$$

As we could see, the Wasserstein distance is continuous with respect to θ, while KL-divergence is not.

The above simple example illustrates the idea that if two distributions are non-overlapping, the Wasserstein distance could provide discriminant information, while KL-divergence fails to do so.

3. Theory of Wasserstein Training

In this section, we introduce the theory of Wasserstein training. First, we tell about computing the Wasserstein distance. Since solving the vanilla Wasserstein distance involves a linear programming and is quite time consuming, we turn to a γ-smoothed Wasserstein distance (Cuturi, 2013). Given the marginal distributions $p(x)$ and $q(x')$, the γ-smoothed Wasserstein distance is defined as:

$$W_\gamma(p, q) = \min_{\pi \in \Pi(p,q)} E_\pi d(x, x') - \gamma H(\pi), \tag{8}$$

where $\Pi(p, q)$ denotes the set of joint distributions with marginals as $p(x)$ and $q(x')$, $H(\pi)$ denotes the Shannon entropy of π. This is equivalent to considering the maximum-entropy principle. The γ-smoothed Wasserstein distance is differentiable with respect to p and q, which considerably facilitates computations. We will show that it can be computed for a much lower complexity than the vanilla Wasserstein distance.

For simplicity, assume that p and q take discrete values. The γ-smoothed Wasserstein distance can be written as an optimization problem:

$$\begin{aligned} W_\gamma(p, q) = \min_\pi \sum_{x,x'} \pi(x, x')(d(x, x') + \gamma \log \pi(x, x')) \\ \text{s.t.} \sum_{x'} \pi(x, x') = p(x), \sum_x \pi(x, x') = q(x'). \end{aligned}$$

To solve the problem, the Lagrangian is introduced as

$$\begin{aligned} L = &\sum_{x,x'} \pi(x, x')(d(x, x') + \gamma \log \pi(x, x')) \\ &+ \sum_x \alpha(x)(p(x) - \sum_{x'} \pi(x, x')) \\ &+ \sum_{x'} \beta(x')(q(x') - \sum_x \pi(x, x')), \end{aligned}$$

where $\alpha(x)$, $\beta(x')$ are Lagrange multipliers, also as dual variables. Set the derivative of L with respect to $\pi(x, x')$ as 0:

$$\begin{aligned} \frac{\partial L}{\partial \pi(x, x')} = d(x, x') + \gamma(1 + \log \pi(x, x')) \\ - \alpha(x) - \beta(x') = 0. \end{aligned}$$

The solution is

$$\pi(x, x') = \exp\left(\frac{1}{\gamma}(\alpha(x) + \beta(x') - d(x, x')) - 1\right). \tag{9}$$

Define vectors $u(x) = \exp(\frac{\alpha(x)}{\gamma})$, $v(x') = \exp(\frac{\beta(x')}{\gamma})$, matrix $K(x, x') = \exp(-\frac{d(x,x')}{\gamma} - 1)$. The constraints require that:

$$\begin{aligned} u(x) \sum_{x'} K(x, x')v(x') = p(x) \\ \sum_x u(x)K(x, x')v(x') = q(x'). \end{aligned} \tag{10}$$

Under the case where x, x' take finite values, (10) becomes a matrix equation:

$$Kv = p./u, K^T u = q./v, \tag{11}$$

which can be solved by iteratively updating u, v until fixed points. After that, we can recover $\alpha^\star(x), \beta^\star(x')$ as

$$\alpha^\star(x) = \gamma \log u(x), \beta^\star(x') = \gamma \log v(x'), \tag{12}$$

and the final result for γ-smoothed Wasserstein distance is

$$\begin{aligned} W_\gamma(p, q) = \sum_x \alpha^\star(x)p(x) + \sum_{x'} \beta^\star(x')q(x') - \gamma \sum_{x,x'} \\ \exp\left(\frac{1}{\gamma}(\alpha^\star(x) + \beta^\star(x') - d(x, x')) - 1\right). \end{aligned} \tag{13}$$

Notice that given $(\alpha^\star(x), \beta^\star(x'))$ as optimal dual variables, for any constant c, $(\alpha^\star(x) - c, \beta^\star(x') + c)$ are also optimal. The optimal dual variables have one extra degree of freedom, due to the existence of one redundant constraint. Therefore, we can set $c = \sum_x \alpha^\star(x)p(x)$, i.e., require that $\alpha^\star(x)$ as the centered optimal dual variable, satisfying $\sum_x \alpha^\star(x)p(x) = 0$.

After that, we consider the parameter estimation using the γ-smoothed Wasserstein distance. Here we choose p as the model distribution p_θ, while q as the data distribution $\hat{p} = \sum_{i=1}^N \frac{1}{N}\delta_{x_i}$, where $\{x_i\}_{i=1}^n$ are data points. The sensitive analysis theorem relates the gradient with respect to $p_\theta(x)$ to the optimal dual variable $\alpha^\star(x)$, as

$$\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial p_\theta(x)} = \alpha^\star(x),$$

Algorithm 1 Compute Wasserstein distance and optimal dual variables

```
Input: data $x_i$, samples $\tilde{x_j}$, distance metric $d$, smoothing parameter $\gamma$.
Output: Wasserstein distance $W(\tilde{p}_\theta, \hat{p})$, optimal dual variables $\alpha^\star(\tilde{x}_j), \beta^\star(x_i)$.
$D = [d(\tilde{x_j}, x_i)]$; $K = \exp(D/\gamma - 1)$; $u = \mathbf{1}/\tilde{N}$
while $u$ changes or other relevant stopping criterion do
    $v = \mathbf{1}/N./(K^T u)$
    $u = \mathbf{1}/\tilde{N}./(Kv)$
end while
$W(\tilde{p}_\theta, \hat{p}) = \mathrm{dot}(u, (D. * K)v)$
$\alpha^\star = \gamma \log u$
$\beta^\star = \gamma \log v + \mathrm{mean}(\alpha^\star)$
$\alpha^\star = \alpha^\star - \mathrm{mean}(\alpha^\star)$ % centered
```

therefore, by chain rule, the gradient with respect to θ is:

$$\begin{aligned}\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial \theta} &= \sum_x \alpha^\star(x) \frac{\partial p_\theta(x)}{\partial \theta} \\ &= \sum_x \alpha^\star(x) \frac{\partial \log p_\theta(x)}{\partial \theta} p_\theta(x) \\ &= E_{x \sim p_\theta(x)} \left[\alpha^\star(x) \frac{\partial \log p_\theta(x)}{\partial \theta} \right]. \end{aligned} \tag{14}$$

Since $\alpha^\star(x)$ are centered, $p_\theta(x)$ in (14) actually can be an unnormalized distribution.

Next, we point out that solving $\alpha^\star(x)$ corresponding to $W_\gamma(p_\theta, \hat{p})$, and computing the expectation in (14), are usually intractable, since x has exponentially many configurations under high dimensional cases. In practice, we adopt the *sampling approximation*, i.e., draw samples $\{\tilde{x}_j\}_{j=1}^{\tilde{N}}$ from the distribution p_θ, and replace p_θ by an empirical distribution $\tilde{p}_\theta = \sum_{j=1}^{\tilde{N}} \frac{1}{\tilde{N}} \delta_{\tilde{x}_j}$. The optimal dual variable $\alpha^\star(\tilde{x}_j)$, corresponding to $W_\gamma(\tilde{p}_\theta, \hat{p})$, can be computed with a complexity of order $\tilde{N}N$ through iteratively updates in (11), whose details are described in Algorithm 1. The gradient is estimated as:

$$\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial \theta} \approx \frac{1}{\tilde{N}} \sum_{j=1}^{\tilde{N}} \alpha^\star(\tilde{x}_j) \frac{\partial \log p_\theta(\tilde{x}_j)}{\partial \theta}. \tag{15}$$

Notice that in the gradient, the real data involves only indirectly through optimal dual variables $\alpha^\star(\tilde{x}_j)$.

Finally, in some probabilistic graphical models like DBMs, it is simpler to consider a joint distribution including hidden variables, where the marginal distribution is $p_\theta(x) = \sum_h p_\theta(x, h)$. In such cases, the gradient with respect to θ is:

$$\begin{aligned}\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial \theta} &= \sum_x \alpha^\star(x) \frac{\partial p_\theta(x)}{\partial \theta} \\ &= \sum_x \alpha^\star(x) \sum_h \frac{\partial p_\theta(x,h)}{\partial \theta} \\ &= \sum_x \sum_h \alpha^\star(x) \frac{\partial \log p_\theta(x,h)}{\partial \theta} p_\theta(x,h) \\ &= E_{(x,h) \sim p_\theta(x,h)} \left[\alpha^\star(x) \frac{\partial \log p_\theta(x,h)}{\partial \theta} \right]. \end{aligned} \tag{16}$$

By adopting the sampling approximation, we can draw samples $(\tilde{x}_j, \tilde{h}_j)$ from the joint distribution $p_\theta(x, h)$, and estimate the gradient in (16) as

$$\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial \theta} \approx \frac{1}{\tilde{N}} \sum_{j=1}^{\tilde{N}} \alpha^\star(\tilde{x}_j) \frac{\partial \log p_\theta(\tilde{x}_j, \tilde{h}_j)}{\partial \theta}. \tag{17}$$

4. Wasserstein Training of RBM

In this section, we implement the Wasserstein training algorithm for a restricted Boltzmann machine (RBM). In a RBM, the joint distribution of the observable variable x and the hidden variables h is

$$p_\theta(x, h) = \frac{1}{Z_\theta} \exp(c^T x + h^T W x + b^T h).$$

The gradients of $\log p_\theta(x)$ with respect to parameters $\theta = (W, b, c)$ are:

$$\begin{aligned}\frac{\partial \log p_\theta(x)}{\partial W} &= \sigma(Wx + b)x^T - E_{x \sim p_\theta(x)}[\sigma(Wx + b)x^T] \\ \frac{\partial \log p_\theta(x)}{\partial b} &= \sigma(Wx + b) - E_{x \sim p_\theta(x)}[\sigma(Wx + b)] \\ \frac{\partial \log p_\theta(x)}{\partial c} &= x - E_{x \sim p_\theta(x)}[x]. \end{aligned}$$

Plug these results into (15). The second terms vanish due to the requirement that $\alpha^\star(\tilde{x}_i)$ are centered. The gradients are estimated as:

$$\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial W} \approx \frac{1}{\tilde{N}} \sum_{j=1}^{\tilde{N}} \alpha^\star(\tilde{x}_j) \sigma(W\tilde{x}_j + b)\tilde{x}_j^T \tag{18}$$

$$\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial b} \approx \frac{1}{\tilde{N}} \sum_{j=1}^{\tilde{N}} \alpha^\star(\tilde{x}_j) \sigma(W\tilde{x}_j + b) \tag{19}$$

$$\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial c} \approx \frac{1}{\tilde{N}} \sum_{j=1}^{\tilde{N}} \alpha^\star(\tilde{x}_j)\tilde{x}_j. \tag{20}$$

The samples $\{\tilde{x}_j\}_{j=1}^{\tilde{N}}$ are updated using persistent contrastive divergence (PCD).

5. Wasserstein Training of DBM

In this section, we implement the Wasserstein training algorithm for a deep Boltzmann machine (DBM). Take a two layer DBM as example. The joint distribution is

$$
\begin{aligned}
p_\theta(x,h) = \frac{1}{Z_\theta}\exp(&c^T x + h^{(1)T}W^{(1)}x + b^{(1)T}h^{(1)} \\
&+ h^{(2)T}W^{(2)}h^{(1)} + b^{(2)T}h^{(2)}).
\end{aligned}
$$

The gradients of $\log p_\theta(x,h)$ with respect to parameters $\theta = (W^{(1)}, W^{(2)}, b^{(1)}, b^{(2)}, c)$ are:

$$
\begin{aligned}
\frac{\partial \log p_\theta(x,h)}{\partial W^{(1)}} &= h^{(1)}x^T - E_{(x,h)\sim p_\theta(x,h)}[h^{(1)}x^T] \\
\frac{\partial \log p_\theta(x,h)}{\partial W^{(2)}} &= h^{(2)}h^{(1)T} - E_{(x,h)\sim p_\theta(x,h)}[h^{(2)}h^{(1)T}] \\
\frac{\partial \log p_\theta(x,h)}{\partial b^{(1)}} &= h^{(1)} - E_{(x,h)\sim p_\theta(x,h)}[h^{(1)}] \\
\frac{\partial \log p_\theta(x,h)}{\partial b^{(2)}} &= h^{(2)} - E_{(x,h)\sim p_\theta(x,h)}[h^{(2)}] \\
\frac{\partial \log p_\theta(x,h)}{\partial c} &= x - E_{(x,h)\sim p_\theta(x,h)}[x].
\end{aligned}
$$

Plug the results into (17). The second terms vanish. The gradients are estimated as:

$$\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial W^{(1)}} \approx \frac{1}{\tilde{N}}\sum_{j=1}^{\tilde{N}} \alpha^\star(\tilde{x}_j)\tilde{h}_j^{(1)}\tilde{x}_j^T \tag{21}$$

$$\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial W^{(2)}} \approx \frac{1}{\tilde{N}}\sum_{j=1}^{\tilde{N}} \alpha^\star(\tilde{x}_j)\tilde{h}_j^{(2)}\tilde{h}_j^{(1)T} \tag{22}$$

$$\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial b^{(1)}} \approx \frac{1}{\tilde{N}}\sum_{j=1}^{\tilde{N}} \alpha^\star(\tilde{x}_j)\tilde{h}_j^{(1)} \tag{23}$$

$$\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial b^{(2)}} \approx \frac{1}{\tilde{N}}\sum_{j=1}^{\tilde{N}} \alpha^\star(\tilde{x}_j)\tilde{h}_j^{(2)} \tag{24}$$

$$\frac{\partial W_\gamma(p_\theta, \hat{p})}{\partial c} \approx \frac{1}{\tilde{N}}\sum_{j=1}^{\tilde{N}} \alpha^\star(\tilde{x}_j)\tilde{x}_j. \tag{25}$$

The samples $\{\tilde{x}_j, \tilde{h}_j^{(1)}, \tilde{h}_j^{(2)}\}_{j=1}^{\tilde{N}}$ are updated using PCD.

Compared to that the gradients of KL-divergence are composed of model expectation terms minus data expectation terms, the gradients of Wasserstein distance does not have the data expectation terms. We should know in mind that real data only influences through the optimal dual variable $\alpha^\star(\tilde{x}_j)$.

6. Stabilization with KL regularization

For the training based on the pure Wasserstein distance, we notice that it is very likely to be entrapped into local minimum. Based on the observed data x, we generate x' using one Gibbs step, and calculate the cross-entropy loss between x and x' as a monitor of the training. As shown in Fig.2, the loss stops decreasing at a level around 300.

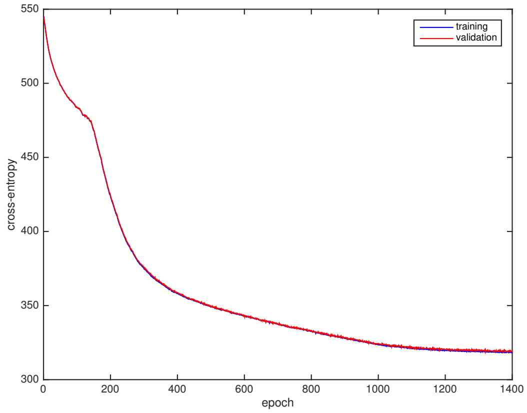


Figure 2. loss v.s. epoch numbers for pure Wasserstein RBM

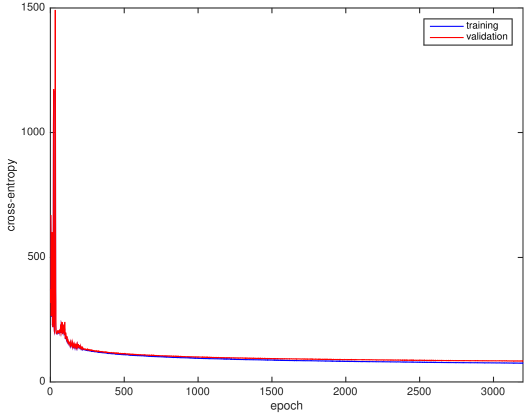


Figure 3. loss v.s. epoch numbers for KL regularized Wasserstein RBM

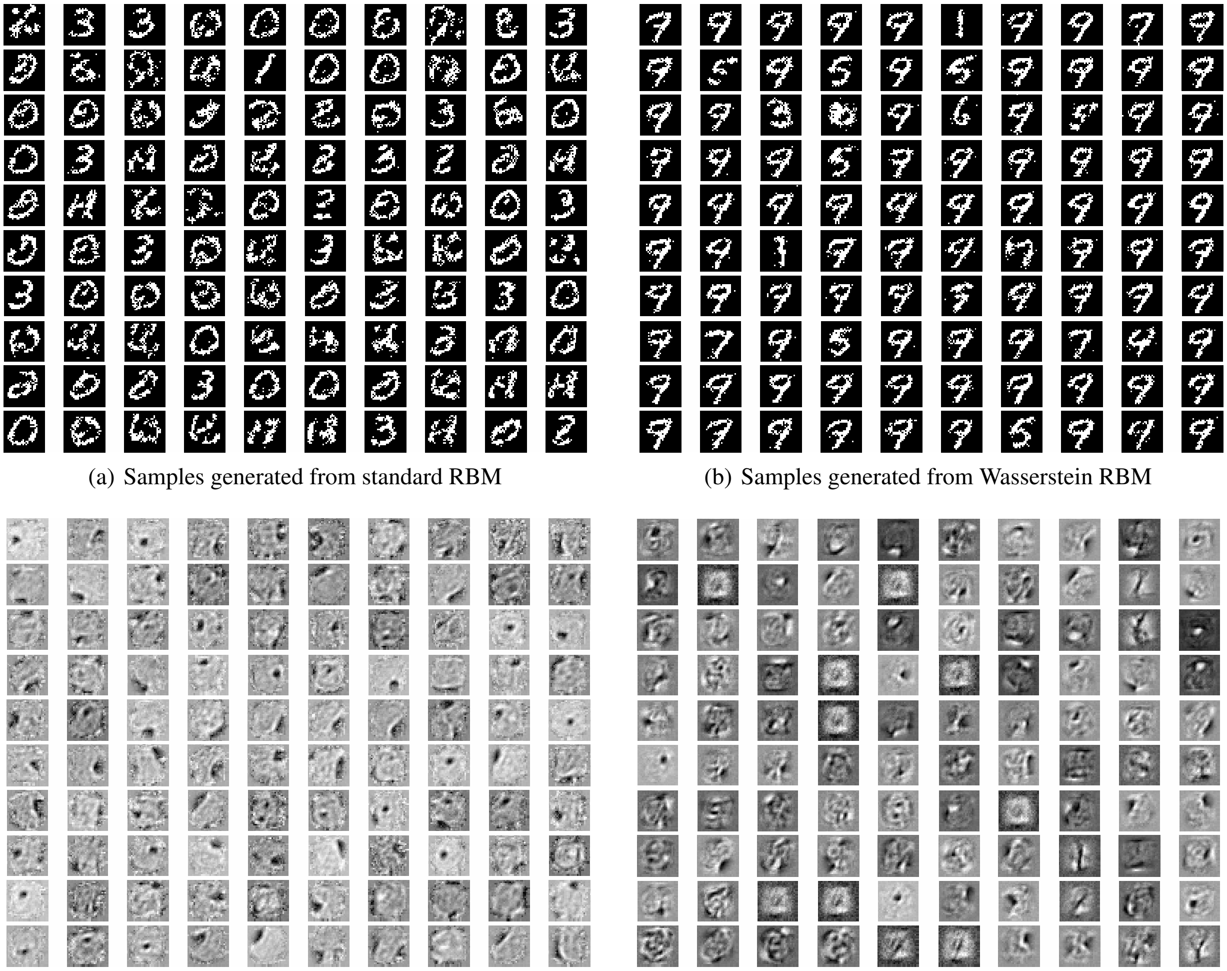
We hypothesize the poor training is due to the fact that the Wasserstein gradient mainly depends on the generated samples from the model distribution p_θ, only indirectly on the data distribution $\hat{p}$. Suppose the samples generated by the model strongly differs from data, this becomes a problem because there is no weighting $\alpha^\star(\tilde{x}_j)$ of the generated samples that can represent the desired direction to a better minimum. In that case, the Wasserstein gradient will lead to a bad local minimum.

To alleviate this issue, we use a hybrid learning of Wasserstein distance and KL-divergence, which is similar to that proposed in (Montavon et al., 2016). The learning objective now becomes minimizing:

$$(1-\eta)W(p_\theta, \hat{p}) + \eta KL(\hat{p}, p_\theta), \tag{26}$$

where η is the mixing ratio.

Fig.3 shows that by using the proposed KL regularized



(a) Samples generated from standard RBM

(b) Samples generated from Wasserstein RBM

(c) Illustration of W for standard RBM

(d) Illustration of W for Wasserstein RBM

Figure 4. Visualization of generated samples and model weights for Wasserstein RBM($\eta = 0.1$) and standard RBM.

Wasserstein learning objective with $\eta = 0.1$, the validation cross-entropy loss now decreases to < 200, which is a big improvement over the pure Wasserstein distance training.

7. Experiments

We evaluate the proposed Wasserstein training for RBM and DBM on MNIST dataset. We strictly follow the MNIST protocol which divides the dataset into a training set with 60,000 digits, and a testing set with 10,000 digits. Throughout the experiments, we set the smoothing parameter $\gamma = 0.1$ for the γ-smoothed Wasserstein distance.

7.1. RBM

The results of training with standard KL divergence and with the Wasserstein distance are compared as following.

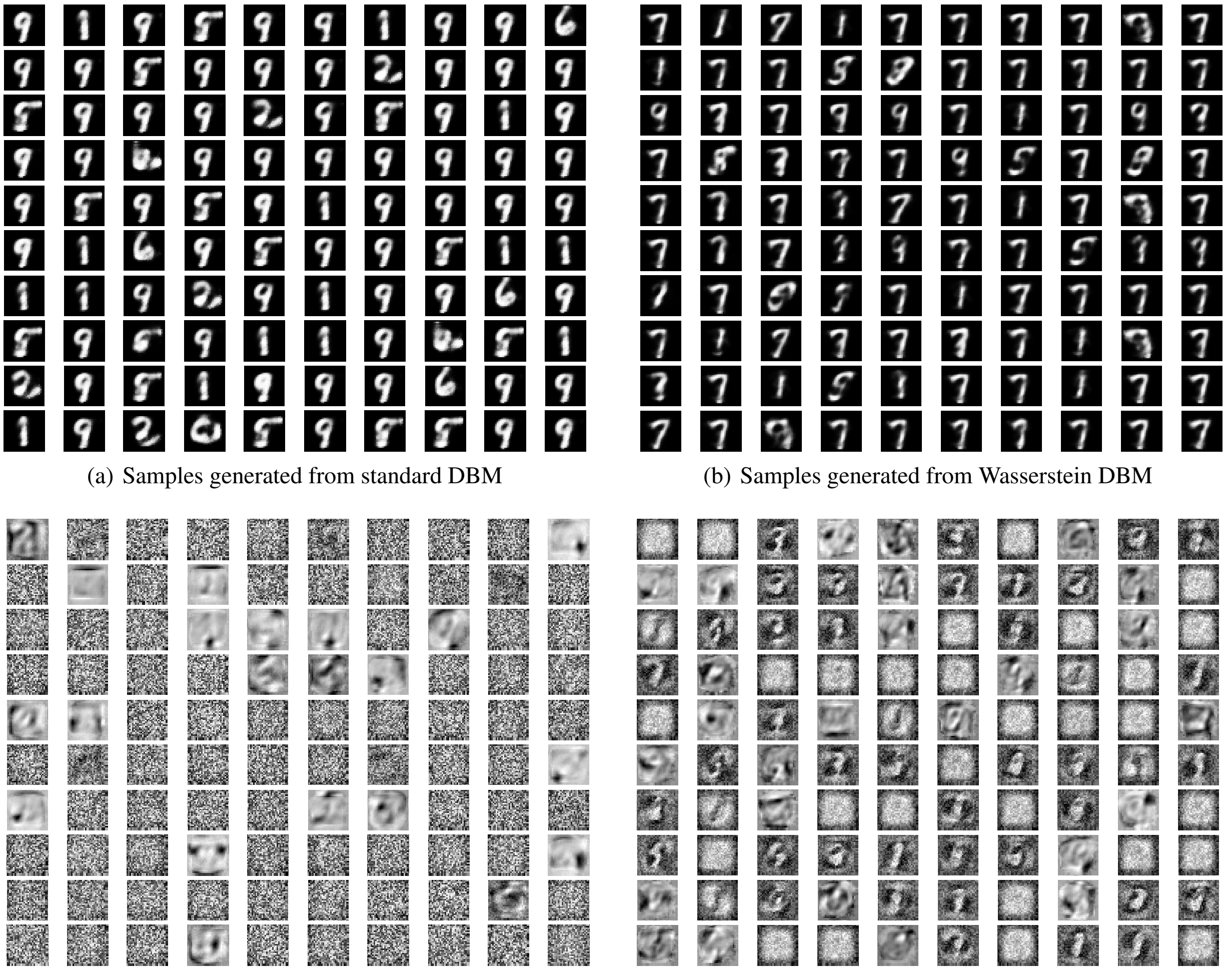
In the training of RBM, we use 100 hidden nodes, adopt the gradient descent using batch size 100, PCD chain size 100, start with learning rate 0.01 with adaptive adjustments (RMSprop), and train for at most 5000 epochs with early stopping.

The learned feature W is shown in Fig. 4(c,d). It reveals some structures such as edges.

Samples drawn from the learned RBMs are shown in Fig. 4(a,b). We draw these samples by first randomly initialize the nodes and then perform 1000 Gibbs sampling steps.

7.2. DBM

The DBM model we used in this experiment consists of two hidden layers: the first hidden layer consists of 1000 hidden nodes; and the second layer has 500 nodes. In the training of DBM, we adopt the gradient descent using batch size 100, PCD chain size 100, RMSprop to adaptively adjust



(a) Samples generated from standard DBM

(b) Samples generated from Wasserstein DBM

(c) Illustration of $W^{(1)}$ for standard DBM

(d) Illustration of $W^{(1)}$ for Wasserstein DBM

Figure 5. Visualization of generated samples and model weights for W-DBM($\eta = 0.3$) and standard DBM.

learning rate, and train for 2000 epochs.

To investigate the effect of mixing ratio parameter η, we train DBMs with different η ranging from 0 to 1, and report their performance in terms of Wasserstein distance between 10,000 digits sampled form the model and 10,000 digits from the test set. Moreover, to faithfully reflect the performance of Wasserstein training, we do not perform discriminative fine-tuning which is usually done in DBM training. The final values of Wasserstein distances after 2000 epochs of training are shown in Fig. 6.

From Fig. 6, we can see that a mixture of Wassertein-distance and KL-divergence outperforms either pure Wasserstein-distance ($\eta = 0$) or KL-divergence ($\eta = 1$). This result empirically shows that though using Wassertein-distance alone is problematic, it is complementary to KL-divergence and a combination of those two achieves better result for DBM training.

To give a more subjective comparison between the proposed Wasserstein-trained DBM (W-DBM, $\eta = 0.3$) to DBM trained with KL-divergence, we show the learned model weights $W^{(1)}$ and digits randomly sampled from W-DBM (Fig. 5, right) and standard DBM (Fig. 5, left).

Finally, in Fig. 7, we visualize the learned W-DBM model distribution by projecting 10,000 randomly drawn samples from the model (red dots) into a 2D plane. The projection is acquired by performing PCA over the data points of the training set. As a reference, we also plot the samples from test set (blue dots). From the figure, we can see that though not perfectly assembles the whole empirical data distribution, W-DBM does loosely align with the data distribution in many local areas.

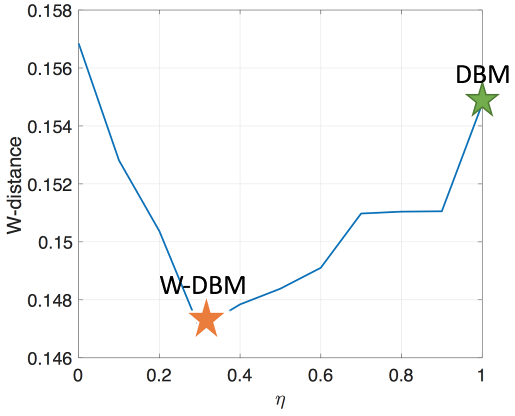


Figure 6. Wasserstein training using different mixing ratio η. When $\eta = 1$, it's equivalent to standard DBM (green star) trained by KL divergence; When $\eta = 0.3$ (orange star), it achieves the lowest W-distance on test set.

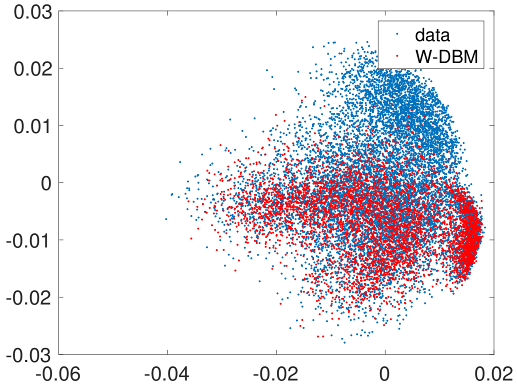


Figure 7. 2-D visualization of samples from W-DBM (red dots) and real data (blue dots).

8. Conclusion

In this report, we introduced Wasserstein distance as a novel loss function instead of KL-divergence, and proposed a general training method to minimize the Wasserstein distance between the sample distribution and the model distribution for various probabilistic models. By adding a regularized entropy term, we found an efficient method to approximate the gradient of γ-smoothed Wasserstein distance with respect to parameters. We also discussed how to train for models containing hidden variables. After that, we developed the Wasserstein training of RBMs and DBMs as specific examples. Experiments and analysis have shown the superiority of W-RBMs/DBMs compared to their traditional counterparts.

References

Arjovsky, Martin, Chintala, Soumith, and Bottou, Leon. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Coates, Adam, Lee, Honglak, and Ng, Andrew Y. An analysis of single-layer networks in unsupervised feature learning. *Ann Arbor*, 1001(48109):2, 2010.

Cuturi, Marco. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pp. 2292–2300, 2013.

Doucet, Arnaud. Fast computation of wasserstein barycenters. 2014.

Frogner, Charlie, Zhang, Chiyuan, Mobahi, Hossein, Araya, Mauricio, and Poggio, Tomaso A. Learning with a wasserstein loss. In *Advances in Neural Information Processing Systems*, pp. 2053–2061, 2015.

Hinton, Geoffrey E and Salakhutdinov, Ruslan R. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Hinton, Geoffrey E and Salakhutdinov, Ruslan R. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, pp. 1607–1614, 2009.

Larochelle, Hugo and Bengio, Yoshua. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pp. 536–543. ACM, 2008.

Montavon, Griegoire, Muller, Klaus-Robert, and Cuturi, Marco. Waterstone training of restricted boltzmann machines. In *Advances in Neural Information Processing Systems*, pp. 3711–3719, 2016.

Salakhutdinov, Ruslan and Hinton, Geoffrey. Deep boltzmann machines. In *Artificial Intelligence and Statistics*, pp. 448–455, 2009.

Salakhutdinov, Ruslan, Mnih, Andriy, and Hinton, Geoffrey. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pp. 791–798. ACM, 2007.

Taylor, Graham W and Hinton, Geoffrey E. Factored conditional restricted boltzmann machines for modeling motion style. In *Proceedings of the 26th annual international conference on machine learning*, pp. 1025–1032. ACM, 2009.

Villani, Cédric. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.