Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Denoising Criterion for Variational Auto-Encoding Framework

Daniel Jiwoong Im, Sungjin Ahn, Roland Memisevic, Yoshua Bengio*
Montreal Institute for Learning Algorithms
University of Montreal
Montreal, QC, H3C 3J7
{imdaniel,ahnsungj,memisevr,¡findme¿}@iro.umontreal.ca

Abstract

Denoising autoencoders (DAE) are trained to reconstruct their clean inputs with noise injected at the input level, while variational autoencoders (VAE) are trained with noise injected in their stochastic hidden layer, with a regularizer that encourages this noise injection. In this paper, we show that injecting noise both in input and in the stochastic hidden layer can be advantageous and we propose a modified variational lower bound as an improved objective function in this setup. When input is corrupted, then the standard VAE lower bound involves marginalizing the encoder conditional distribution over the input noise, which makes the training criterion intractable. Instead, we propose a modified training criterion which corresponds to a tractable bound when input is corrupted. Experimentally, we find that the proposed denoising variational autoencoder (DVAE) yields better average log-likelihood than the VAE and the importance weighted autoencoder on the MNIST and Frey Face datasets.

Introduction

Variational inference (Jordan et al. 1999) has been a core component of approximate Bayesian inference along with the Markov chain Monte Carlo (MCMC) method (Neal 1993). It has been popular to many researchers and practitioners because the problem of learning an intractable posterior distribution is formulated as an optimization problem which has many advantages compared to MCMC; (i) we can easily take advantage of many advanced optimization tools (Kingma and Ba 2014a; Duchi, Hazan, and Singer 2011; Zeiler 2012), (ii) the training by optimization is usually faster than the MCMC sampling, and (iii) unlike MCMC, where it is difficult to decide when to finish the sampling, the stopping criterion of variational inference is more clear.

One remarkable recent advance in variational inference is to use the inference network (also known as the recognition network) as the approximate posterior distribution (Kingma and Welling 2014; Rezende and Mohamed 2014; Dayan et al. 1995; Bornschein and Bengio 2014). Unlike the traditional variational inference where different variational parameters are required for each latent variable, in the inference network, the approximate posterior distribution for each latent variable is conditioned on an observation and the parameters are shared among the latent variables. Combined with advances in training techniques such as the re-parameterization trick and the REINFORCE (Williams 1992; Mnih and Gregor 2014), it became possible to train variational inference models efficiently for large-scale datasets.

Despite these advances, it is still a major challenge to obtain a class of variational distributions which is flexible enough to accurately model the true posterior distribution. For instance, in the variational autoencoder (VAE), in order to achieve efficient training, each dimension of the latent variable is assumed to be independent each other and modeled by a univariate Gaussian distribution whose parameters (i.e., the mean and the variance) are obtained by a nonlinear projection of the input using a neural network. Although VAE performs well in practice for a rather simple problems such as generating small and simple images (e.g., MNIST), it is desired to relax this strong restriction on the variational distributions in order to apply it to more complex real-world problems. Recently, there have been efforts in this direction. (Salimans, Kingma, and Welling 2015) integrated MCMC steps into the variational inference such that the variational distribution becomes closer to the target distribution as it takes more MCMC steps inside each iteration of the variational inference. Similar ideas but applying a sequence of invertible and deterministic non-linear transformations rather than MCMC are also proposed by (Dinh, Krueger, and Bengio 2015) and (Rezende and Mohamed 2015).

On the other hand, the denoising criterion, where the input is corrupted by adding some noise and the model is asked to recover the original input, has been studied extensively for deterministic generative models (Seung 1998; Vincent et al. 2008; Bengio et al. 2013). These studies showed that the denoising criterion plays an important role in achieving good generalization performance (Vincent et al. 2008) because it makes the nearby data points in the low dimensional manifold to be robust against the presence of small noise in the high dimensional observation space (Seung 1998; Vincent et al. 2008; Rifai 2011; Alain and Bengio 2014; Im, Belghazi, and Memisevic 2016). Therefore, it seems a legitimate question to ask if the denoising criterion (where we add the noise to the inputs) can also be advantageous for

*CIFAR Senior Fellow

the variational auto-encoding framework where the noise is added to the latent variables, but not to the inputs, and if so, how can we formulate the problem for efficient training. Although it has not been considerably studied how to combine these, there has been some evidences of its usefulness[1]. For example, (Rezende and Mohamed 2014) pointed out that injecting additional noise to the recognition model is crucial to achieve the reported accuracy for unseen data, advocating that in practice denoising can help the regularization of probabilistic generative models as well.

In this paper, motivated by the DAE and the VAE, we study the denoising criterion for variational inference based on recognition networks, which we call the *variational auto-encoding framework* throughoutthe paper. Our main contributions are as follows. We introduce a new class of approximate distributions where the recognition network is obtained by marginalizing the input noise over a corruption distribution, and thus provides capacity to obtain a more flexible approximate distribution class such as the mixture of Gaussian. Because applying this approximate distribution to the standard VAE objective makes the training intractable, we propose a new objective, called the denoising variational lower bound, and show that, given a sensible corruption function, this is (i) tractable and efficient to train , and (ii) easily applicable to many existing models such as the variational autoencoder, the importance reweighted autoencoder (IWAE) (Burda, Grosse, and Salakhutdinov 2015), and the neural variational inference and learning (NVIL) (Mnih and Gregor 2014). In the experiments, we empirically demonstrate that the proposed denoising criterion for variational auto-encoding framework helps to improve the performance in both the variational autoencoders and the importance weighted autoencoders (IWAE) on the binarized MNIST dataset and the Frey Face dataset.

Variational Autoencoders

The variational autoencoder (Kingma and Welling 2014; Rezende and Mohamed 2014) is a particular type of variational inference framework which is closely related to our focus in this work (see Appendix for background on variational inference). With the VAE, the posterior distribution is defined as $p_\theta(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$. Specifically, we define a prior $p(\mathbf{z})$ on the latent variable $\mathbf{z} \in \mathbb{R}^D$, which is usually set to an isotropic Gaussian distribution $\mathcal{N}(0, \sigma\mathbb{I}_D)$. Then, we use a parameterized distribution to define the observation model $p_\theta(\mathbf{x}|\mathbf{z})$. A typical choice for the parameterized distribution is to use a neural network where the input is $\mathbf{z}$ and the output a parametric distribution over $\mathbf{x}$, such as the Gaussian or Bernoulli distributions, depending on the type of the output observation. Then, θ becomes the weights of the neural network. We call this network $p_\theta(\mathbf{x}|\mathbf{z})$ the *generative network*. Due to the complex nonlinearity of the neural network, the posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$ is intractable.

[1]In practice, it turned out to be useful to *augment* the dataset by adding some random noise to the inputs. However, in denoising criterion, unlike the augmenting, the model tries to recover the original data, not the corrupted one.

One interesting aspect of VAE is that the approximate distribution q is conditioned on the observation $\mathbf{x}$, resulting in a form $q_\phi(\mathbf{z}|\mathbf{x})$. Similar to the generative network, we use a neural network for $q_\phi(\mathbf{z}|\mathbf{x})$ with $\mathbf{x}$ and $\mathbf{z}$ as its input and output, respectively. The variational parameter ϕ, which is also the weights of the neural network, is shared among all observations. We call this network $q_\phi(\mathbf{z}|\mathbf{x})$ the *inference network*, *recognition network*.

The objective of VAE is to maximize the following variational lower bound with respect to the parameters θ and ϕ.

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \tag{1}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathbb{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \tag{2}$$

Note that in Eqn. (2), we can interpret the first term as a reconstruction accuracy through an autoencoder with noise injected in the hidden layer that is the output of the inference network, and the second term as a regularizer which enforces the approximate posterior to be close to the prior and maximizes the entropy of the injected noise.

The earlier approaches to train this type of models were based on the variational EM algorithm: in the E-step, fixing θ, we update ϕ such that the approximate distribution $q_\phi(\mathbf{z}|\mathbf{x})$ close to the true posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$, and then in the M-step, fixing ϕ, we update θ to increase the marginal log-likelihood. However, with the VAE it is possible to apply the backpropagation on the variational parameter ϕ by using the re-parameterization trick (Kingma and Welling 2014), considering $\mathbf{z}$ as a function of i.i.d. noise and of the output of the encoder (such as the mean and variance of the Gaussian). Armed with the gradient on these parameters, the gradient on the generative network parameters θ can readily be computed by back-propagation, and thus we can jointly update both ϕ and θ using efficient optimization algorithms such as the stochastic gradient descent.

Although our exposition in the following proceeds mainly with the VAE for simplicity, the proposed method can be applied to a more general class of variational inference methods which use the inference network $q_\phi(\mathbf{z}|\mathbf{x})$. This includes other recent models such as the importance weighted autoencoders (IWAE), the neural variational inference and learning (NVIL), and DRAW (Gregor et al. 2015).

Denoising Criterion in Variational Framework

With the denoising autoencoder criterion (Seung 1998; Vincent et al. 2008), the input is corrupted according to some noise distribution, and the model needs to learn to reconstruct the original input (e.g., by maximize the log-probability of the clean input $\mathbf{x}$, given the corrupted input $\tilde{\mathbf{x}}$). Before applying the denoising criterion to the variational autoencoder, we shall investigate a synthesized inference formulation of the VAE in order to comprehend the consequences of the denoising criterion.

Proposition 1. *Let $q_\phi(\mathbf{z}|\tilde{\mathbf{x}})$ be a conditional Gaussian distribution such that $q_\phi(\mathbf{z}|\tilde{\mathbf{x}}) = \mathcal{N}(\mathbf{z}|\mu_\phi(\tilde{\mathbf{x}}), \sigma_\phi(\tilde{\mathbf{x}}))$ where $\mu_\phi(\tilde{\mathbf{x}})$ and $\sigma_\phi(\tilde{\mathbf{x}})$ are non-linear functions of $\tilde{\mathbf{x}}$. Let $p(\tilde{\mathbf{x}}|\mathbf{x})$*

be a known corruption distribution around $\mathbf{x}$*. Then,*

$$\tilde{q}_\phi(\mathbf{z}|\mathbf{x}) = \mathbb{E}_{p(\tilde{\mathbf{x}}|\mathbf{x})}\left[q_\phi(\mathbf{z}|\tilde{\mathbf{x}})\right] = \int_{\tilde{\mathbf{x}}} q_\phi(\mathbf{z}|\tilde{\mathbf{x}})p(\tilde{\mathbf{x}}|\mathbf{x})d\tilde{\mathbf{x}} \quad (3)$$

is a mixture of Gaussian.

Depending on whether the distribution is over a continuous or discrete variables, the integral in Equation 3 can be replaced by a summation. It is instructive to consider the distribution over discrete domain to see that Equation 3 has a form of mixture of Gaussian - that is, each time we sample $\tilde{\mathbf{x}} \sim p(\tilde{\mathbf{x}}|\mathbf{x})$ and substitute into $q(\mathbf{z}|\tilde{\mathbf{x}})$, we get different Gaussian distributions.

Example 1. *Let* $\mathbf{x} \in \{0,1\}^D$ *be a* D*-dimension observation, and consider a Bernoulli corruption distribution* $p_{\boldsymbol{\pi}}(\tilde{\mathbf{x}}|\mathbf{x}) = Ber(\boldsymbol{\pi})$ *around the input* $\mathbf{x}$*. Then,*

$$\mathbb{E}_{p_{\boldsymbol{\pi}}(\tilde{\mathbf{x}}|\mathbf{x})}\left[q_\phi(\mathbf{z}|\tilde{\mathbf{x}})\right] = \sum_{i=1}^{K} q_\phi(\mathbf{z}|\tilde{\mathbf{x}}_i)p_{\boldsymbol{\pi}}(\tilde{\mathbf{x}}_i|\mathbf{x}) \quad (4)$$

is a finite mixture of Gaussian with $K = 2^D$ *mixture components.*

As mentioned in the previous section, usually a feedforward neural network is used for the inference network. In the case of the Bernoulli distribution as a corrupting distribution and $q_\phi(\mathbf{z}|\tilde{\mathbf{x}})$ is a Gaussian distribution, we will have 2^D Gaussian mixture components and all of them share the parameter ϕ.

Example 2. *Consider a Gaussian corruption model* $p(\tilde{\mathbf{x}}|\mathbf{x}) = N(\mathbf{x}|\mathbf{0}, \sigma I)$*. Let* $q_\phi(\mathbf{z}|\tilde{\mathbf{x}})$ *be a Gaussian inference network. Then,*

$$\mathbb{E}_{p(\tilde{\mathbf{x}}|\mathbf{x})}\left[q_\phi(\mathbf{z}|\tilde{\mathbf{x}})\right] = \int_{\tilde{\mathbf{x}}} q_\phi(\mathbf{z}|\tilde{\mathbf{x}})p(\tilde{\mathbf{x}}|\mathbf{x})d\tilde{\mathbf{x}}. \quad (5)$$

1. *If* $q_\phi(\mathbf{z}|\phi^\top\tilde{\mathbf{x}}) = \mathcal{N}(\mathbf{z}|\mu = \phi^\top\tilde{\mathbf{x}}, \sigma = \sigma^2 I)$ *such that the mean parameter is a linear model of weight vector* ϕ *and input* $\tilde{\mathbf{x}}$*, then the Equation 5 is a Gaussian distribution.*

2. *If* $q_\phi(\mathbf{z}|\tilde{\mathbf{x}}) = \mathcal{N}(\mathbf{z}|\mu(\tilde{\mathbf{x}}), \sigma(\tilde{\mathbf{x}}))$ *where* $\mu(\tilde{\mathbf{x}})$ *and* $\sigma(\tilde{\mathbf{x}})$ *are non-linear functions of* $\tilde{\mathbf{x}}$*, then the Equation 5 is an infinite mixture of Gaussian.*

In practice, there will be infinitely many number of Gaussian mixture components as in the second case, all of whose parameters are predicted by a single neural network. In other words, the inference neural network will learn which Gaussian distribution is needed for the given input $\tilde{\mathbf{x}}$[2].

We can see this corruption procedure as adding a stochastic layer to the bottom of the inference network. For example, we can define a corruption network $p_{\boldsymbol{\pi}}(\tilde{\mathbf{x}}|\mathbf{x})$ which is a neural network where the input is $\mathbf{x}$ and the output is stochastic units (e.g., Gaussian or Bernoulli distributions). Then, it is also possible to learn the parameter $\boldsymbol{\pi}$ of the corruption network by backpropagation using the reparameterization trick. Note that a similar idea is explored in IWAE (Burda, Grosse, and Salakhutdinov 2015). However, our method is different in the sense that we use the denoising variational lower bound as described below.

[2]The mixture components are encoded in a vector form.

The Denoising Variational Lower Bound

Previously, we described that integrating the denoising criterion into the variational auto-encoding framework is equivalent to having a stochastic layer at the bottom of the inference network, and then estimating the variational lower bound becomes intractable because $\mathbb{E}_{p(\tilde{\mathbf{x}}|\mathbf{x})}[q_\phi(\mathbf{z}|\tilde{\mathbf{x}})]$ requires integrating out the noise $\tilde{\mathbf{x}}$ of the corruption distribution. Before introducing the denoising variational lower bound, let us examine the variational lower bound when an additional stochastic layer is added to the inference network and then it is integrated out over the stochastic variables.

Lemma 1. *Consider an approximate posterior distribution of the following form:*

$$q_\Phi(\mathbf{z}|\mathbf{x}) = \int_{\mathbf{z}'} q_\varphi(\mathbf{z}|\mathbf{z}')q_\psi(\mathbf{z}'|\mathbf{x})d\mathbf{z}',$$

here, we use $\Phi = \{\varphi, \psi\}$*. Then, given* $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$*, we obtain the following inequality:*

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\Phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\varphi(\mathbf{z}|\mathbf{z}')}\right] \geq \mathbb{E}_{q_\Phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\Phi(\mathbf{z}|\mathbf{x})}\right].$$

Refer to the Appendix for the proof. Note that $q_\psi(\mathbf{z}'|\mathbf{x})$ can be either parametric or non-parametric distribution. We can further show that this generalizes to multiple stochastic layers in the inference network.

Theorem 1. *Consider an approximate posterior distribution of the following form*

$$q_\Phi(\mathbf{z}|\mathbf{x}) = \int_{\mathbf{z}^1 \cdots \mathbf{z}^{L-1}} q_{\phi^L}(\mathbf{z}|\mathbf{z}^{L-1}) \cdots q_{\phi^1}(\mathbf{z}^1|\mathbf{x})d\mathbf{z}^1 \cdots d\mathbf{z}^{L-1}.$$

Then, given $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$*, we obtain the following inequality:*

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\Phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{\prod_{i=1}^{L-1} q_{\phi^i}(\mathbf{z}^{i+1}|\mathbf{z}^i)}\right] \quad (6)$$

$$\geq \mathbb{E}_{q_\Phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\Phi(\mathbf{z}|\mathbf{x})}\right], \quad (7)$$

where $\mathbf{z} = \mathbf{z}^L$ *and* $\mathbf{x} = \mathbf{z}^1$*.*

The proof is presented in the Appendix. Theorem 1 implies that adding more stochastic layers gives tighter lower bound.

We now use Lemma 1 to derive the *denoising variational lower bound*. When the approximate distribution has the following form $\tilde{q}_\phi(\mathbf{z}|\mathbf{x}) = \int q_\phi(\mathbf{z}|\tilde{\mathbf{x}})p(\tilde{\mathbf{x}}|\mathbf{x})d\tilde{\mathbf{x}}$, we can write the variational lower bound as follows:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{\tilde{q}_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\tilde{\mathbf{x}})}\right] \quad (8)$$

$$= \mathbb{E}_{\tilde{q}_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{\tilde{q}_\phi(\mathbf{z}|\mathbf{x})}\right] \stackrel{\text{def}}{=} \mathcal{L}_{cvae}. \quad (9)$$

Applying Lemma 1 to Equation 9, we can pull out the expectation in the denominator outside of the log and obtain the denoising variational lower bound:

$$\mathcal{L}_{dvae} \stackrel{\text{def}}{=} \mathbb{E}_{\tilde{q}_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\tilde{\mathbf{x}})}\right]. \quad (10)$$

Note that the $p_\theta(\mathbf{x}, \mathbf{z})$ in the numerator of the above equation is a function of $\mathbf{x}$ not $\tilde{\mathbf{x}}$. That is, given corrupted input $\tilde{\mathbf{x}}$ (in the denominator), the $\mathcal{L}_{dvae}$ objective tries to reconstruct the original input $\mathbf{x}$ not the corrupted input $\tilde{\mathbf{x}}$. This *denoising* criterion is different from the popular *data augmentation* approach where the model tries to reconstruct the corrupted input.

By the Lemma 1, we finally have the following:

$$\log p_\theta(\mathbf{x}) \geq \mathcal{L}_{dvae} \geq \mathcal{L}_{cvae}. \tag{11}$$

It is important to note that the above does not necessarily mean that $\mathcal{L}_{dvae} \geq \mathcal{L}_{vae}$ where

$$\mathcal{L}_{vae} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]. \tag{12}$$

This is because $\tilde{q}_\phi(\mathbf{z}|\mathbf{x})$ in $\mathcal{L}_{cvae}$ depends on the choice of a corruption distribution while $q_\phi(\mathbf{z}|\mathbf{x})$ in $\mathcal{L}_{vae}$ does not.

Note also that $\tilde{q}_\phi(\mathbf{z}|\mathbf{x})$ has the capacity to cover a much broader class of distributions than $q_\phi(\mathbf{z}|\mathbf{x})$. This makes it possible for $\mathcal{L}_{dvae}$ to be a tighter lower bound of $\log p_\theta(\mathbf{x})$ than $\mathcal{L}_{vae}$. For example, suppose that the true posterior $p(\mathbf{z}|\mathbf{x})$ consists of multiple modes. Then, $\tilde{q}_\phi(\mathbf{z}|\mathbf{x})$ has the potential of modeling more than a single mode, whereas it is impossible to model multiple modes of $p(\mathbf{z}|\mathbf{x})$ using $q_\phi(\mathbf{z}|\mathbf{x})$ regardless of which lower bound of $\log p_\theta(\mathbf{x})$ is used as the objective function. However, we also note that it is also possible to make the $\mathcal{L}_{cvae}$ a looser lower bound than $\mathcal{L}_{vae}$ by choosing a very inefficient corruption distribution $p(\tilde{\mathbf{x}}|\mathbf{x})$ such that it completely distorts the input $\mathbf{x}$ in such a way to lose all useful information required for the reconstruction, resulting in $\mathcal{L}_{vae} > \mathcal{L}_{cvae}$. Therefore, for $\mathcal{L}_{dvae}$, it is important to choose a sensible corruption distribution.

A question that arises when we consider $\mathcal{L}_{dvae}$ is what is the underlying meaning of maximizing $\mathcal{L}_{dvae}$. As mentioned earlier, the aim of variational objective is to minimize the distributions between approximate posterior and true posterior distribution, i.e. $\mathbb{KL}_{dvae} = \mathbb{KL}\left(\tilde{q}_\phi(\mathbf{z}||\mathbf{x})||p(\mathbf{z}|\mathbf{x})\right) = \log p_\theta(\mathbf{x}) - \mathcal{L}_{cvae}$. However, $\mathcal{L}_{dvae}$ definitely does not minimizes only the KL between approximate posterior and true posterior distribution as we can observe that $\mathbb{KL}_{cvae} = \mathbb{KL}\left(\tilde{q}_\phi(\mathbf{z}||\mathbf{x})||p(\mathbf{z}|\mathbf{x})\right) \geq \log p_\theta(\mathbf{x}) - \mathcal{L}_{dvae}$. This illustrates that $\mathbb{KL}_{dvae} \leq \mathbb{KL}_{cvae}$. Nonetheless, $\mathbb{KL}_{dvae}$ provides a tractable way to optimize from the approximate posterior distribution $q(\mathbf{z}|\mathbf{x})$. Thus, it is interesting to see the following proposition.

Proposition 2. *Maximizing $\mathcal{L}_{dvae}$ is equivalent to minimizing the following objective*

$$\mathbb{E}_{p(\tilde{\mathbf{x}}|\mathbf{x})}[\mathbb{KL}(\tilde{q}_\phi(\mathbf{z}|\tilde{\mathbf{x}})||p(\mathbf{z}|\mathbf{x}))]. \tag{13}$$

That is,

$$\log p_\theta(\mathbf{x}) = \mathcal{L}_{dvae} + \mathbb{E}_{p(\tilde{\mathbf{x}}|\mathbf{x})}[\mathbb{KL}(\tilde{q}_\phi(\mathbf{z}|\tilde{\mathbf{x}})||p(\mathbf{z}|\mathbf{x}))].$$

The proof is presented in the Appendix. Proposition 2 illustrates that maximizing $\mathcal{L}_{dvae}$ is equivalent to minimizing the expectation of the KL between the true posterior distribution and approximate posterior distribution *over all noised inputs* from $p(\tilde{\mathbf{x}}|\mathbf{x})$. We believe that this is indeed an effective objective because the inference network tries to learn to map not only the training data point but also its corrupted variations to the true posterior distribution, resulting in a more robust training of the inference network to unseen data points. As shown in Theorem 1, this argument also applies for multiple stochastic layers of inference network.

Training Procedure

One may consider a simple way of training VAE with the denoising criterion, which is similar to how the vanilla denoising autoencoder is trained: (i) sample a corrupted input $\tilde{\mathbf{x}}^{(m)} \sim p(\tilde{\mathbf{x}}|\mathbf{x})$, (ii) sample $\mathbf{z}^{(l)} \sim q(\mathbf{z}|\tilde{\mathbf{x}}^{(m)})$, and (iii) sample reconstructed images from the generative network $p_\theta(\mathbf{x}|\mathbf{z}^{(l)})$. This procedure is akin to the regular VAE except that the input is corrupted by a noise distribution at every update.

The above procedure can be seen as a special case of optimizing the following objective which can be easily approximated by Monte Carlo sampling.

$$\mathcal{L}_{dvae} = \mathbb{E}_{p(\tilde{\mathbf{x}}|\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\tilde{\mathbf{x}})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\tilde{\mathbf{x}})}\right] \tag{14}$$

$$\simeq \frac{1}{MK}\sum_{m=1}^{M}\sum_{k=1}^{K} \log \frac{p_\phi(\mathbf{x}, \mathbf{z}^{(k|m)})}{q_\phi(\mathbf{z}^{(k|m)}|\tilde{\mathbf{x}}^{(m)})} \tag{15}$$

where $\tilde{\mathbf{x}}^{(m)} \sim p(\tilde{\mathbf{x}}|\mathbf{x})$ and $\mathbf{z}^{(k|m)} \sim q_\phi(\mathbf{z}|\tilde{\mathbf{x}}^{(m)})$. In the experiment section, we call the estimator of Equation 15 DVAE. Although in the above we applied the denoising criterion for VAE (resulting in DVAE) as a demonstration, the proposed procedure is applicable more generally to other variational methods with inference networks. For example, the training procedure for IWAE with denoising criterion can be formulated with Monte Carlo approximation:

$$\mathcal{L}_{diwae} = \mathbb{E}_{p(\tilde{\mathbf{x}}|\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\tilde{\mathbf{x}})}\left[\log \sum_{m=1}^{M}\sum_{k=1}^{K} \frac{p_\theta(\mathbf{x}, \mathbf{z}^{(k|m)})}{q_\phi(\mathbf{z}^{(k|m)}|\tilde{\mathbf{x}}^{(m)})}\right] \tag{16}$$

$$\simeq \log \frac{1}{MK}\sum_{m=1}^{M}\sum_{k=1}^{K} \frac{p_\phi(\mathbf{x}, \mathbf{z}^{(k|m)})}{q_\phi(\mathbf{z}^{(k|m)}|\tilde{\mathbf{x}}^{(m)})}. \tag{17}$$

where $\tilde{\mathbf{x}}^{(m)} \sim p(\tilde{\mathbf{x}}|\mathbf{x})$, $\mathbf{z}^{(k|m)} \sim q_\phi(\mathbf{z}|\tilde{\mathbf{x}}^{(m)})$, and Monte Carlo sample size is set to 1. We named the following estimator of Equation 17 as DIWAE.

Experiments

We conducted empirical studies of DVAE under the denoising variational lower bound as discussed in Section . To assess whether adding a denoising criterion to the variational auto-encoding models enhance the performance or not, we tested on the denoising criterion on VAE and IWAE throughout the experiments. As mentioned in Section , since the choice of the corruption distribution is crucial, we compare on different corruption distributions of various noise levels.

We consider two datasets, the binarized MNIST dataset and the Frey face dataset. The MNIST dataset contains 60,000 images for training and 10,000 images for test and each of the images is 28 $\times$ 28 pixels for handwritten digits

from 0 to 9 (LeCun et al. 1998). Out of the 60,000 training examples, we used 10,000 examples as validation set to tune the hyper-parameters of our model. We use the binarized version of MNIST, where each pixel of an image is sampled from $\{0, 1\}$ according to its pixel intensity value. The Frey Face[3] dataset consists of 2000 images of Brendan Frey's face. We split the images into 1572 training data, 295 validation data, and 200 test data. We normalized the images such that each pixel value ranges between $[0, 1]$.

Throughout the experiments, we used the same neural network architectures for VAE and IWAE. Also, a single stochastic layer with 50 latent variables is used for both VAE and IWAE. For the generation network, we used a neural network of two hidden layers each of which has 200 units. For the inference network, we tested two architectures, one with a single hidden layer and the other with two hidden layers. We then used 200 hidden units for both of them. We used softplus activations for VAE and tanh activations for IWAE following the same configuration of the original papers of (Kingma and Welling 2014) and (Burda, Grosse, and Salakhutdinov 2015). For binarized MNIST, the last layer of the generative network was sigmoid and the usual cross-entropy term was used. For the Frey Face dataset where the input value is real numbers, we used Gaussian stochastic units for the output layer of the generation network.

For all our results, we ran 10-fold experiments. We optimized all our models with ADAM (Kingma and Ba 2014b). We set the batch size to 100 and the learning rate was selected from a discrete range chosen based on the validation set. We used 1 and 5 samples of $\mathbf{z}$ per update for VAE and 5 samples for IWAE. Note that using 1 sample for IWAE is equivalent to VAE. The reported results were only trained with training set, not including the validation set.

Following common practices of choosing a noise distribution, we deployed the *salt and pepper* noise to the binary MNIST and Gaussian noise to the real-valued Frey Face dataset. Table 1 presents the negative variational lower bounds with respect to different corruption levels on the MNIST. Table 2 presents the negative variational lower bound using unnormalized generation networks, with respect to different corruption levels on the Frey Face dataset. Note that when the corruption level is set to zero, DVAE and DIWAE are identical to VAE and IWAE, respectively.

In the following, we analyze the results by answering questions on the experiments.

***Q**: Does adding the denoising criterion improve the performance of variational autoencoders?*

Yes. All of the methods with denoising criterion surpassed the performance of vanilla VAE and vanilla IWAE as shown in Table 1 and Table 2. But, it is dependent on the choice of proper corruption level; for a large amount of noise, as we expected, it tends to perform worse than the vanilla VAE and IWAE.

***Q**: How sensitive is the model for the type and the level of the noise?*

It seems that both of the models are not very sensitive with respect to the two types of noises: Gaussian and salt and pepper. They are more sensitive to the *level* of the noise rather than the *type*. Based on the experiments, the optimal corruption level lies in between $(0, 5]$ since all of the results in that range are better than the one with 0% noise. It is natural to see this result considering that, when the noise level is excessive, (i) the model will lose information required to reconstruct the original input and that (ii) there will be large gap between the distributions of the (corrupted) training dataset and the test dataset.

[3]Available at http://www.cs.nyu.edu/∼roweis/data.html.

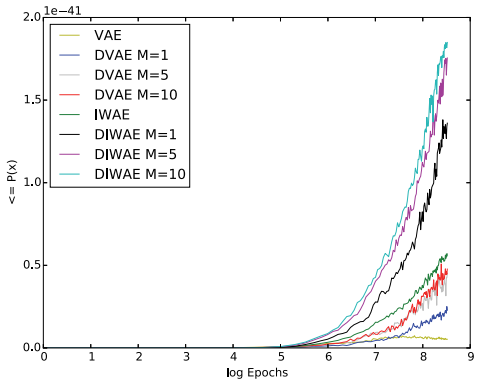

Figure 1: Denoising Variational Lower Bound for DVAE and DIWAE

***Q**: How do the sample sizes M affect to the result?*

In Figure 1, we show the results on different configurations of M. As shown, increasing the sample size helps to converge faster in terms of the number of epochs and converge to better log-likelihood. The converged values of VAE are 94.97, 94.44, and $\mathbf{94.39}$ for $M = 1, 5$, and 10 respectively, and 93.17, 92.89, and $\mathbf{92.85}$ for IWAE. Note, however, that increasing sample size requires more computation. Thus, in practice using $M = 1$ seems a reasonable choice.

***Q**: What happens when we replace the neural network in the inference network with some other types of model?*

Several applications have demonstrated that recurrent neural network can be more powerful than neural netework. Here, we tried replacing neural network in the inference network with gated recurrent neural network that consist of single recurrent hidden layers with five time steps (Chung et al. 2014). We denote these models DVAE (GRU) and DIWAE (GRU) where GRU stands for gated recurrent units.

Table 3 demonstrates the results with different noise level on MNIST dataset. We notice that when VAE combined with GRU tend to severely overfit on the training data and it actually performed worse than having a neural network at the inference network. However, denoising criterion redeems the overfitting behaviour and produce much better results comparing with both VAE (GRU) and DVAE with regular neural networks. Similarly, IWAE combined with GRU showed overfitting behaviour although it gave better results than DIWAE with neural networks. Also, DIWAE (GRU) gave the best performance among all models we experimented with.

***Q**: Data augmentation v.s. data corruption?*

We consider specific data augmentation where our data

Table 1: Negative variational lower bounds using different corruption levels on MNIST (the lower, the better). The salt-and-pepper noises are injected to data $\mathbf{x}$ during the training.

Model	# Hidden Layers	Noise Level			
		0	5	10	15
DVAE (K=1)	1	96.14 ± 0.09	**95.52 ± 0.12***	**96.12 ± 0.06**	96.83 ± 0.17
DVAE (K=1)	2	95.90 ± 0.23	**95.34 ± 0.17***	**95.65 ± 0.14**	96.17 ± 0.17
DVAE (K=5)	1	95.20 ± 0.07	**95.01 ± 0.04***	95.55 ± 0.07	96.41 ± 0.11
DVAE (K=5)	2	95.01 ± 0.07	**94.71 ± 0.13***	**94.90 ± 0.22**	96.41 ± 0.11
DIWAE (K=5)	1	94.36 ± 0.07	**93.67 ± 0.10***	**93.97 ± 0.07**	**94.35 ± 0.08**
DIWAE (K=5)	2	94.31 ± 0.07	**93.08 ± 0.08***	**93.35 ± 0.13**	**93.71 ± 0.07**

Table 2: Negative variational lower bound using different corruption levels on the Frey Face dataset. Gaussian noises are injected to data $\mathbf{x}$ during the training.

Model	# Hid. Layers	Noise Level			
		0	2.5	5	7.5
DVAE (K=1)	1	1304.79 ± 5.71	**1313.74 ± 3.64***	**1314.48 ± 5.85**	1293.07 ± 5.03
DVAE (K=1)	2	1317.53 ± 3.93	**1322.40 ± 3.11***	**1319.60 ± 3.30**	1306.07 ± 3.35
DVAE (K=5)	1	1306.45 ± 6.13	**1320.39 ± 4.17***	**1313.14 ± 5.80**	1298.40 ± 4.74
DVAE (K=5)	2	1317.51 ± 3.81	**1324.13 ± 2.62***	**1320.99 ± 3.49**	**1317.56 ± 3.94**
DIWAE (K=5)	1	1318.04 ± 2.83	**1320.18 ± 3.43**	**1333.44 ± 2.74***	1305.38 ± 2.97
DIWAE (K=5)	2	1320.03 ± 1.67	**1334.77 ± 2.69***	**1323.97 ± 4.15**	1309.30 ± 2.95

Table 3: Negative variational lower bounds using different corruption levels on MNIST (the lower, the better) with recurrent neural network as a inference network. The salt-and-pepper noises are injected to data $\mathbf{x}$ during the training.

Model	# Hidden Layers	Noise Level			
		0	5	10	15
DVAE (GRU)	1	96.07 ± 0.17	**94.30 ± 0.09***	**94.32 ± 0.12**	**94.88 ± 0.11**
DIWAE (GRU)	1	93.94 ± 0.06	**93.13 ± 0.11**	**92.84 ± 0.07***	**93.03 ± 0.04**

lies in between 0 and 1, $\mathbf{x} \in (0,1)^D$ like MNIST. We consider a new binary data point $\mathbf{x}' \in \{0,1\}^D$ where the previous data is treated as a probability of each pixel value turning on, i.e. $p(\mathbf{x}') = \mathbf{x}$. Then, we augment the data by sampling the data from $\mathbf{x}$ at every iteration. Although, this setting is not realistic, we were curious whether the performance of this data augmentation compare to denoising criterion. The performance of such data augmentation on MNIST gave 93.88 ± 0.08 and 92.51 ± 0.07 for VAE and IWAE. Comparing these negative log-likelihood with the performance of DVAE and DIWAE, which were 94.32 ± 0.37 and 93.83 ± 0.06, data augmentation VAE outperformed DVAE but data augmentation IWAE was worse than DIWAE.

***Q**: Can we propose a more sensible noise distribution?*

For all the experiment results, we have used a simple corruption distribution using a global corruption rate (the parameter of the Bernoulli distribution or the variance of the Gaussian distribution) to all pixels in the images. To see if a more sensible corruption can lead to an improvement, we also tested another corruption distribution by obtaining a *mean image*. Here, we obtained the mean image by averaging all training images and then used the pixel intensity of the mean image as the corruption rate so that each pixel has different corruption rate which statistically encodes at some extent the pixel-wise noise from the entire dataset. However, we could not observe a noticeable improvement from this compared to the version with the global corruption rate, although we believed that this is a better way of designing the corrupting distribution. One interesting direction is to use a parameterized corruption distribution and learn the parameter. This will be advantageous because we can use our denoising variational lower bound which it is tighter than the classical variational lower bound on noisy inputs. We leave this for the future work.

Conclusions

In this paper, we studied the denoising criterion for a general class of variational inference models where the approximate posterior distribution is conditioned on the input $\mathbf{x}$. The main result of our paper was to introduce the denoising variational lower bound which, provided a sensible corruption function, can be tighter than the standard variational lower bound on noisy inputs. We claimed that this training criterion makes it possible to learn more flexible and robust approximate posterior distributions such as the mixture of Gaussian than the standard training method without corruption. In the experiments, we empirically observed that the proposed method can consistently help to improve the performance for the variational autoencoder and the importance weighted autoencoder. Although we observed considerable improvements for our experiments with simple corruption distributions, how to obtain the sensible corruption distribution is still an important open question. We think that learning with a parametrized corruption distribution or obtaining a better heuristic procedure will be important for the method to be applied more broadly.

References

Alain, G., and Bengio, Y. 2014. What regularized auto-encoders learn from the data generating distribution. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Bengio, Y.; Yao, L.; Alain, G.; and Vincent, P. 2013. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, 899–907.

Bornschein, J., and Bengio, Y. 2014. Reweighted wake-sleep. *arXiv preprint arXiv:1406.2751*.

Burda, Y.; Grosse, R.; and Salakhutdinov, R. 2015. Importance weighted auto-encoder. In *http://arxiv.org/pdf/1509.00519v1.pdf*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Gated feedback recurrent neural networks. In *Proceedings of the International Conference of Machine Learning (ICML)*.

Dayan, P.; Hinton, G. E.; Neal, R. M.; and Zemel, R. S. 1995. The helmholtz machine. *Neural computation* 7(5):889–904.

Dinh, L.; Krueger, D.; and Bengio, Y. 2015. Nice: Nonlinear independent component estimation. In *International Conference on Learning Representation Workshop*.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research* 12:2121–2159.

Gregor, K.; Danihelka, I.; Graves, A.; Rezende, D. J.; and Wierstra, D. 2015. Draw: A recurrent neural network for image generation. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Im, D. J.; Belghazi, M. I. D.; and Memisevic, R. 2016. Conservativeness of untied auto-encoders. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Jordan, M.; Ghahramani, Z.; Jaakkola, T. S.; and Saul, L. K. 1999. An introduction to variational methods for graphical models. *Machine Learning* 37:183–233.

Kingma, D., and Ba, J. 2014a. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D., and Ba, J. 2014b. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. In *Proceedings of the Neural Information Processing Systems (NIPS)*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 87:2278–2324.

Mnih, A., and Gregor, K. 2014. Neural variational inference and learning in belief networks. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Neal, R. 1993. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, University of Toronto, Computer Science.

Rezende, D. J., and Mohamed, S. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Rezende, D. J., and Mohamed, S. 2015. Variational inference with normalizing flows. In *Proceedings of the International Conference of Machine Learning (ICML)*.

Rifai, S. 2011. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*.

Salimans, T.; Kingma, D. P.; and Welling, M. 2015. Markov chain monte carlo and variational inference: Bridging the gap. In *Proceedings of the International Conference of Machine Learning (ICML)*.

Seung, H. 1998. Learning continuous attractors in recurrent networks. In *Proceedings of the Neural Information Processing Systems (NIPS)*, 654–660.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 58:229–256.

Zeiler, M. D. 2012. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.