Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks

Lars Mescheder LARS.MESCHEDER@TUEBINGEN.MPG.DE
MPI Tübingen, Autonomous Vision Group

Sebastian Nowozin SEBASTIAN.NOWOZIN@MICROSOFT.COM
Microsoft Research Cambridge, UK

Andreas Geiger ANDREAS.GEIGER@TUEBINGEN.MPG.DE
MPI Tübingen, Autonomous Vision Group

Abstract

Variational Autoencoders (VAEs) are expressive latent variable models that can be used to learn complex probability distributions from training data. However, the quality of the resulting model crucially relies on the expressiveness of the inference model used during training. We introduce Adversarial Variational Bayes (AVB), a technique for training Variational Autoencoders with arbitrarily expressive inference models. We achieve this by introducing an auxiliary discriminative network that allows to rephrase the maximum-likelihood-problem as a two-player game, hence establishing a principled connection between VAEs and Generative Adversarial Networks (GANs). We show that in the nonparametric limit our method yields an *exact* maximum-likelihood assignment for the parameters of the generative model, as well as the *exact* posterior distribution over the latent variables given an observation. Contrary to competing approaches which combine VAEs with GANs, our approach has a clear theoretical justification, retains most advantages of standard Variational Autoencoders and is easy to implement.

1. Introduction

Generative models in machine learning are models that can be trained on an unlabeled dataset and are capable of generating new data points after training is completed. As generating new content requires a good understanding of the training data at hand, such models are often regarded as a key ingredient to unsupervised learning. In fact, it can be argued that generating new content is also extremely important to the human learning process: in order to fully grasp a subject, students usually have to work through numerous exercises on their own and thereby learn how to "generate" new content.

In recent years, generative models have become more and more powerful. While many model classes such as PixelRNNs (van den Oord et al., 2016b), PixelCNNs (van den Oord et al., 2016a), real NVP (Dinh et al., 2016) and Plug & Play generative networks (Nguyen et al., 2016) have been introduced and studied, the two most prominent ones are Variational Autoencoders (VAEs) (Kingma & Welling, 2013; Rezende et al., 2014) and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014).

Both VAEs and GANs have their own advantages and disadvantages: while GANs generally yield visually sharper results when applied to images, VAEs are attractive because they naturally yield both a generative model and an inference model. Moreover, it was reported, that VAEs often lead to better log-likelihoods (Wu et al., 2016), making them interesting for scenarios where visual realism is not the main goal. The recently introduced BiGANs (Donahue et al., 2016; Dumoulin et al., 2016) add an inference model to GANs. However, it was observed that the reconstruction results often only vaguely resemble the input and often do so only semantically and not in a quantitative way.

The failure of VAEs to generate sharp images is often attributed to the fact that the inference models used during training are usually not expressive enough to capture the true posterior distribution. Indeed, recent work shows that using more expressive model classes can lead to substantially better results (Kingma et al., 2016), both visually and in terms of log-likelihood-bounds.

In this paper, we present Adversarial Variational Bayes (AVB), a technique for training Variational Autoencoders with arbitrarily flexible inference models parameterized by neural networks. We can show that in the nonparametric limit we obtain a maximum-likelihood assignment for the generative model together with the correct posterior distribution.

While there were some attempts at combining VAEs and GANs (Makhzani et al., 2015; Larsen et al., 2015), most of these attempts are not motivated from a maximum-likelihood point of view and therefore usually do not lead to maximum-likelihood assignments. For example, in Adversarial Autoencoders (AAEs) (Makhzani et al., 2015) the Kullback-Leibler-regularization term that appears in the training objective for VAEs is replaced with an adversarial loss that encourages the aggregated posterior to be close to the prior over the latent variables. Even though AAEs do not maximize a lower bound to the maximum-likelihood-objective, we show in Section 5.2 that AAEs can be interpreted as an approximation to our approach, thereby establishing a connection of AAEs to maximum-likelihood learning.

Our contributions are as follows:

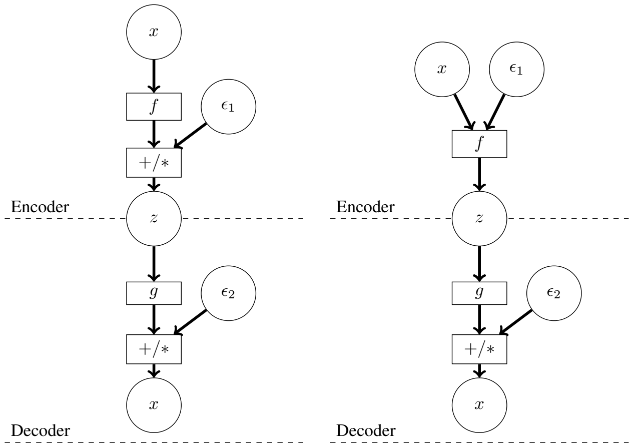
- we enable the usage of arbitrarily complex inference models for Variational Autoencoders using adversarial training
- we give theoretical insight into our method, showing that in the nonparametric limit our method recovers the true posterior distribution as well as a true maximum-likelihood assignment for the parameters of the generative model
- we demonstrate experimentally that our model is able to learn rich posterior distributions and show that the model is able to generate compelling samples for complex data sets

2. Background

As our model is an extension of Variational Autoencoders (VAEs) (Kingma & Welling, 2013; Rezende et al., 2014), we start with a brief review of VAEs.

VAEs are specified by a parametric generative model $p_\theta(x \mid z)$ of the visible variables given the latent variables, a prior $p(z)$ over the latent variables and an approximate inference model $q_\phi(z \mid x)$ over the latent variables given the visible variables. It can be shown that

$$\log p(x) \geq -\mathrm{KL}(q_\phi(z \mid x), p(z)) + \mathrm{E}_{q_\phi(z|x)} \log p_\theta(x \mid z). \quad (2.1)$$



(a) Standard VAE

(b) VAE with black box inference model

Figure 1. Schematic comparison of standard VAE and VAE with black-box inference model, where ϵ_1 and ϵ_2 denote samples from some noise distribution. While more complicated inference models for Variational Autoencoders are possible, they are usually not as flexible as our black-box approach.

The right hand side of (2.1) is called the variational lower bound or evidence lower bound (ELBO). If there is ϕ such that $q_\phi(z \mid x) = p_\theta(z \mid x)$, we would have

$$\log p(x) = \max_\phi -\mathrm{KL}(q_\phi(z \mid x), p(z)) + \mathrm{E}_{q_\phi(z|x)} \log p_\theta(x \mid z). \quad (2.2)$$

However, in general this is not true, so that we only have an inequality in (2.2).

When performing maximum-likelihood training, our goal is to optimize the marginal log-likelihood

$$\mathrm{E}_{p_\mathcal{D}(x)} \log p_\theta(x), \quad (2.3)$$

where $p_\mathcal{D}$ is the data distribution. Unfortunately, computing $\log p_\theta(x)$ requires marginalizing out z in $p_\theta(x, z)$ which is usually intractable. Variational Bayes uses inequality (2.1) to rephrase the intractable problem of optimizing (2.3) into

$$\max_\theta \max_\phi \mathrm{E}_{p_\mathcal{D}(x)} \Big[-\mathrm{KL}(q_\phi(z \mid x), p(z)) + \mathrm{E}_{q_\phi(z|x)} \log p_\theta(x \mid z) \Big]. \quad (2.4)$$

Due to inequality (2.1), we still optimize a lower bound to the true maximum-likelihood-objective (2.3).

Naturally, the quality of this lower bound depends on the expressiveness of the inference model $q_\phi(z \mid x)$. Usually, $q_\phi(z \mid x)$ is taken to be a Gaussian distribution with diagonal covariance matrix whose mean and variance vectors are parameterized by neural networks with x as input (Kingma & Welling, 2013; Rezende et al., 2014). While this model is very flexible in its dependence on x, its dependence on z is very restrictive, potentially limiting the quality of the resulting generative model. Indeed, it was observed that applying standard Variational Autoencoders to natural images often results in blurry images (Larsen et al., 2015).

3. Method

In this work we show how we can instead use a black-box inference model $q_\phi(z \mid x)$ and use adversarial training to obtain an approximate maximum likelihood assignment θ^* to θ and a close approximation $q_{\phi^*}(z \mid x)$ to the true posterior $p_{\theta^*}(z \mid x)$. This is visualized in Figure 1: on the left hand side the structure of a typical Variational Autoencoder is shown. The right hand side shows our flexible black-box inference model.

3.1. Derivation

To derive our method, we rewrite the optimization problem in (2.4) as

$$\max_\theta \max_\phi \mathrm{E}_{p_\mathcal{D}(x)} \mathrm{E}_{q_\phi(z|x)} \big(\log p(z) \\ - \log q_\phi(z \mid x) + \log p_\theta(x \mid z) \big). \quad (3.1)$$

When we have an explicit representation of $q_\phi(z \mid x)$, (3.1) can be optimized using the reparameterization trick (Kingma & Welling, 2013) and stochastic gradient descent. Unfortunately, this is not the case when we define $q_\phi(z \mid x)$ by a black-box procedure like in Figure 1b.

The idea of our approach is to circumvent this problem by representing the term

$$\log p(z) - \log q_\phi(z \mid x) \quad (3.2)$$

indirectly as the optimal value of an additional real-valued discriminative network $T(x, z)$ that we introduce to the problem.

More specifically, consider the following objective for the discriminator $T(x, z)$:

$$\max_T \mathrm{E}_{p_\mathcal{D}(x)} \mathrm{E}_{q_\phi(z|x)} \log \sigma(T(x,z)) \\ + \mathrm{E}_{p_\mathcal{D}(x)} \mathrm{E}_{p(z)} \log \left(1 - \sigma(T(x,z))\right). \quad (3.3)$$

Here, $\sigma(t) := (1 + \mathrm{e}^{-t})^{-1}$ denotes the sigmoid-function. Intuitively, $T(x, z)$ tries to distinguish pairs (x, z) that were sampled independently using the distribution $p_\mathcal{D}(x)p(z)$ from those that were sampled using the current inference model, i.e., using $p_\mathcal{D}(x)q_\phi(z \mid x)$.

To simplify the theoretical analysis, we assume that the model $T(x, z)$ is flexible enough to represent any function of the two variables x and z. This assumption is often referred to as the nonparametric limit (Goodfellow et al., 2014) and is justified by the fact that deep neural networks are universal function approximators (Hornik et al., 1989).

As it turns out, the optimal discriminator $T^*(x, z)$ according to the objective in (3.3) is given by the negative of (3.2).

Proposition 1. *For $p_\theta(x \mid z)$ and $q_\phi(z \mid x)$ fixed, the optimal discriminator T^* according to the objective in* (3.3) *is given by*

$$T^*(x, z) = \log q_\phi(z \mid x) - \log p(z). \quad (3.4)$$

Proof. The proof is analogous to the proof of Proposition 1 in Goodfellow et al. (2014). See the Appendix for details. □

Together with (3.1), Proposition 1 allows us to write the optimization objective in (2.4) as

$$\max_{\theta,\phi} \mathrm{E}_{p_\mathcal{D}(x)} \mathrm{E}_{q_\phi(z|x)} \big(-T^*(x,z) + \log p_\theta(x \mid z) \big), \quad (3.5)$$

where $T^*(x, z)$ is defined as the function that maximizes (3.3).

To optimize (3.5), we need the gradients of (3.5) with respect to θ and ϕ. While taking the gradient with respect to θ is straightforward, taking the gradient with respect to ϕ is complicated by the fact that we have defined $T^*(x, z)$ indirectly as the solution of an auxiliary optimization problem which itself depends on ϕ. The following Proposition shows that taking the gradient with respect to the explicit occurrence of ϕ in $T^*(x, z)$ is not necessary:

Proposition 2. *We have*

$$\mathrm{E}_{q_\phi(z|x)} \left(\nabla_\phi T^*(x, z) \right) = 0. \quad (3.6)$$

Proof. By Proposition 1,

$$\mathrm{E}_{q_\phi(z|x)} \left(\nabla_\phi T^*(x, z) \right) \\ = \mathrm{E}_{q_\phi(z|x)} \left(\nabla_\phi \log q_\phi(z \mid x) \right). \quad (3.7)$$

For an arbitrary family of probability densities q_ϕ we have

$$E_{q_\phi} \left(\nabla_\phi \log q_\phi \right) = \int q_\phi(z) \frac{\nabla_\phi q_\phi(z)}{q_\phi(z)} \mathrm{d}z \\ = \nabla_\phi \int q_\phi(z) \mathrm{d}z = \nabla_\phi 1 = 0. \quad (3.8)$$

Together with (3.7), this implies (3.6). □

Using the reparameterization trick (Kingma & Welling, 2013; Rezende et al., 2014), (3.5) can be rewritten in the form

$$\max_{\theta,\phi} \mathrm{E}_{p_{\mathcal{D}}(x)} \mathrm{E}_{\epsilon} \big(-T^*(x, z_\phi(x,\epsilon)) + \log p_\theta(x \mid z_\phi(x,\epsilon)) \big) \tag{3.9}$$

for a suitable function $z_\phi(x,\epsilon)$. Together with Proposition 1, (3.9) allows us to take unbiased estimates of the gradients of (3.5) with respect to ϕ and θ.

3.2. Algorithm

Algorithm 1 Adversarial Variational Bayes (AVB)

```
$i = 0$
while not converged do
    Sample $m$ examples $\{x^{(1)}, \dots, x^{(m)}\}$ from data distribution $p_{\mathcal{D}}(x)$.
    Sample $m$ examples $\{z^{(1)}, \dots, z^{(m)}\}$ from prior distribution $p(z)$.
    Sample $m$ noise examples $\{\epsilon^{(1)}, \dots, \epsilon^{(m)}\}$ from $\mathcal{N}(0,1)$.
    Compute $\theta$-gradient (eq. 3.9):
        $g_\theta = \nabla_\theta \frac{1}{m} \sum_{i=1}^{m} \log p_\theta\left(x^{(i)} \mid z_\phi\left(x^{(i)}, \epsilon^{(i)}\right)\right)$
    Compute $\phi$-gradient (eq. 3.9):
        $g_\phi = \nabla_\phi \frac{1}{m} \sum_{i=1}^{m} \big[ -T_\psi\left(x^{(i)}, z_\phi(x^{(i)}, \epsilon^{(i)})\right) + \log p_\theta\left(x^{(i)} \mid z_\phi(x^{(i)}, \epsilon^{(i)})\right) \big]$.
    Compute $\psi$-gradient (eq. 3.3) :
        $g_\psi = \nabla_\psi \frac{1}{m} \sum_{i=1}^{m} \Big[ \log\left(\sigma(T_\psi(x^{(i)}, z_\phi(x^{(i)}, \epsilon^{(i)})))\right) + \log\left(1 - \sigma(T_\psi(x^{(i)}, z^{(i)}))\right) \Big]$.
    Perform SGD-updates for $\theta$, $\phi$ and $\psi$:
        $\theta = \theta + h_i\, g_\theta, \quad \phi = \phi + h_i\, g_\phi, \quad \psi = \psi - h_i\, g_\psi$.
    $i = i + 1$
end while
```

In theory, Propositions 1 and 2 allow us apply stochastic gradient descent directly to the objective in (2.4). However, this requires keeping $T^*(x,z)$ optimal which is computationally challenging. We therefore regard the optimization problems in (3.3) and (3.9) as a two-player game. Propositions 1 and 2 show that any Nash-equilibrium of this game yields a stationary point of the objective in (2.4).

In practice, we try to find a Nash-equilibrium by applying stochastic-gradient-descent (SGD) jointly to (3.3) and (3.9), see Algorithm 1. Here, we parameterize the neural network T with a vector ψ. Even though we have no guarantees that this algorithm converges, any fix point of this algorithm yields a stationary point of the objective in (2.4).

Note that optimizing (3.5) with respect to ϕ while keeping θ and T fixed makes the encoder network collapse to a deterministic function. This is also a common problem for regular GANs (Radford et al., 2015). It is therefore crucial to keep the discriminative T network close to optimality while optimizing (3.5).

3.3. Theoretical results

In Sections 3.1 we derived AVB as a way of performing stochastic gradient descent on the variational lower bound in (2.4). In this section, we analyze the properties of Algorithm 1 from a game theoretical point of view.

As the next proposition shows, global Nash-equilibria of Algorithm 1 yield global optima of the objective in (2.4):

Proposition 3. *If (θ^*, ϕ^*, T^*) defines a Nash-equilibrium of the two-player game defined by (3.3) and (3.9), then*

$$T^*(x,z) = \log q_{\phi^*}(z \mid x) - \log p(z) \tag{3.10}$$

and (θ^*, ϕ^*) is a global optimum of the variational lower bound in (2.4).

Proof. The proof can be found in the Appendix. □

Consider now the case where T and $q_\phi(z \mid x)$ are both flexible enough, so that T can represent any function of two variables and $q_\phi(z \mid x)$ can represent any family of positive probability densities on the latent space. In this case we have

Corollary 4. *Assume T can represent any function of two variables and $q_\phi(z \mid x)$ can represent any family of positive probability densities on the latent space. If (θ^*, ϕ^*, T^*) defines a Nash-equilibrium for the game defined by (3.3) and (3.9), then*

1. *θ^* is a maximum-likelihood assignment*
2. *$q_{\phi^*}(z \mid x)$ is equal to the true posterior $p_{\theta^*}(z \mid x)$*
3. *T^* is the pointwise mutual information between x and z, i.e.*

$$T^*(x,z) = \log \frac{p_{\theta^*}(x,z)}{p_{\theta^*}(x)p(z)}. \tag{3.11}$$

Proof. This is a straightforward consequence of Proposition 3, as in this case (θ^*, ϕ^*) optimizes the variational lower bound in (2.4) if and only if 1 and 2 hold. Inserting the result from 2 into (3.10) yields 3. □

4. Experiments

We tested our method both on shallow fully connected and a deep convolutional neural networks.

The CNN implementation is based on Radford et al. (2015) who show that deep convolutional networks can be used to parameterize the generator and discriminator for GANs. In

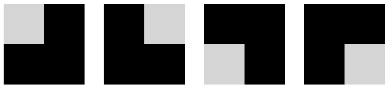


Figure 2. Training examples in the synthetic dataset.

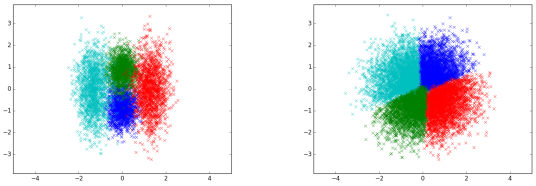
	VAE	AVB
log-likelihood	-1.593	**-1.462**
reconstruction error	0.091	**0.021**
ELBO	-1.700	≈ -1.483
$\mathrm{KL}(q_\phi(z), p(z))$	≈ 0.175	≈ **0.028**

Table 1. Comparison of VAE and AVB on synthetic dataset. The optimal log-likelihood score on this dataset is $-\log(4) \approx -1.386$.

particular, Radford et al. demonstrate that their network architecture is capable of generating sharp images that are visually close to the training data. Our method also leads to sharp images, but in addition also yields an approximation to the posterior distribution that can be used for inference.

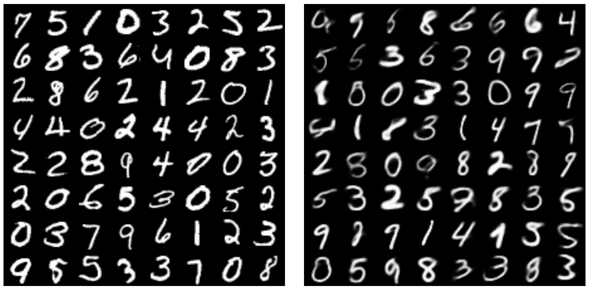
Synthetic example As a first test, we trained a fully connected neural network on a simple synthetic dataset containing only the 4 data points shown in Figure 2 and a 2-dimensional latent space. Both the encoder and decoder are parameterized by 2-layer fully connected neural networks with 512 hidden units each. The encoder network takes as input a data point x and a vector of Gaussian random noise ϵ and produces a latent code z. The decoder network takes as input a latent code z and produces the parameters for four independent Bernoulli-distributions, one for each pixel of the output image. The adversary is parameterized by two neural networks with two 512-dimensional hidden layers each, acting on x and z respectively, whose 512-dimensional outputs are combined using an inner product.

We compare our method to a Variational Autoencoder with a diagonal Gaussian posterior distribution. The encoder and decoder networks are parameterized similarly as above, but the encoder network does not take the noise ϵ as input and produces a mean and variance vector instead of a single sample.



(a) VAE (b) AVB

Figure 3. Distribution of latent code for VAE and AVAE trained on artificial dataset consisting of 4 different states.



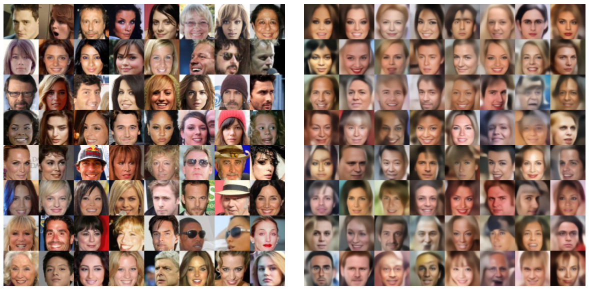
(a) Training data (b) Random samples

Figure 4. Independent samples for a model trained on MNIST

We visualize the resulting division of the latent space in Figure 3, where each color corresponds to one state in the x-space. Whereas the Variational Autoencoder divides the space into a mixture of 4 Gaussians, the Adversarial Variational Autoencoder learns a complex posterior distribution. Quantitatively this can be verified by computing the KL-divergence between the prior $p(z)$ and the aggregated posterior $q_\phi(z) := \int q_\phi(z \mid x)p_\mathcal{D}(x)\mathrm{d}x$, which we estimate using the ITE-package (Szabo, 2013), see Table 1. Note that the variations for different colors in Figure 3 are solely due to noise ϵ used in the inference model.

The ability of AVB to learn more complex posterior models leads to improved performance as Table 1 shows. In particular, AVB leads to a much higher likelihood score that is close to the optimal value of $-\log(4)$ compared to a standard VAE that struggles with the fact that it cannot divide the latent space appropriately. Moreover, we see that the reconstruction error given by the mean cross-entropy between an input x and its reconstruction using the encoder and decoder networks is much lower when using AVB instead of a VAE with diagonal Gaussian inference model. We also observe that the estimated variational lower bound is close to the true log-likelihood, indicating that the adversary has learned the correct function.

MNIST In addition, we trained deep convolutional networks based on the DC-GAN-architecture (Radford et al., 2015) on the MNIST-dataset (LeCun et al., 1998). We use a similar architecture as for the synthetic example, but use 5-layer deep convolutional neural networks instead of fully connected neural networks acting on x. In the encoder network, we add the noise ϵ via a learned projection matrix to each hidden layer of the neural network. For the adversary, we keep the fully connected neural network with one hidden layer acting on z, but replace the network acting on x with a deep convolutional neural network. Moreover, we use batch normalization both in the encoder and adversary but not in the decoder, as we found that this decreases re-



(a) Training data (b) Random samples

Figure 5. Independent samples for a model trained on celebA

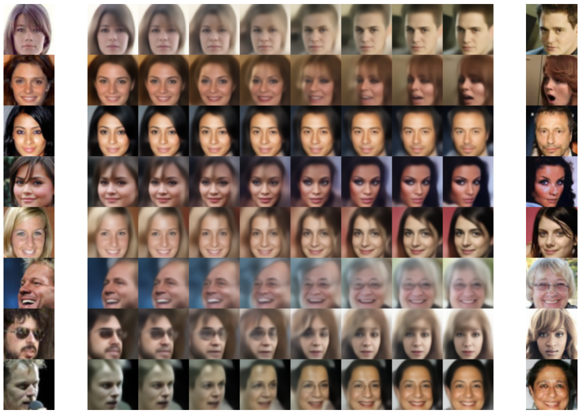


Figure 6. Interpolation experiments for celebA

construction quality.

Some random samples for MNIST are shown in Figure 4. We see that our model produces random samples that are visually close the training set.

celebA We also trained a deep convolutional network on the celebA-dataset (Liu et al., 2015). We use the same architecture as for MNIST except that we replace the adversary with a single deep convolutional neural network acting on x. We add the latent code z to each hidden layer via a learned projection matrix. Moreover, in the encoder and decoder we replace every hidden layer by three RESNET-units (He et al., 2015).

The samples for celebA are shown in Figure 5. We see that our model produces visually sharp images of faces. To demonstrate that the model has indeed learned an abstract representation of the data, we show reconstruction results and the result of linearly interpolating the z-vector in the latent space in Figure 6. We see that the reconstructions are reasonably sharp and the model produces realistic images for all interpolated z-values.

5. Related Work

5.1. Connection to Variational Autoencoders

Adversarial Variational Bayes strives to optimize the same objective as a standard Variational Autoencoder (Kingma & Welling, 2013; Rezende et al., 2014), but approximates the Kullback-Leibler divergence using an adversary instead of relying on a closed-form formula.

Substantial work has focused on making the the class of approximate inference models more expressive. Normalizing flows (Rezende & Mohamed, 2015; Kingma et al., 2016) make the posterior more complex by composing a simple Gaussian posterior with an invertible smooth mapping for which the determinant of the Jacobian is tractable. Auxiliary Variable VAEs (Maaløe et al., 2016) add auxiliary variables to the posterior to make it more flexible. However, no other approach we are aware of allows to use black-box inference models parameterized by a general neural network that takes as input a data point and a noise vector and produces a sample from the approximate posterior.

As neural network are universal function approximators (Hornik et al., 1989), our approach recovers the true posterior in the nonparametric limit. In the nonparametric limit, this implies that we obtain a true maximum-likelihood assignment for the parameters of the generative model. Of course, in practice this is only true if the adversary is sufficiently expressive to represent the optimal discriminator. However, in practice we found that our method is able to generate high quality images also if the discriminative network is of limited capacity.

5.2. Connection to Adversarial Autoencoders

Makhzani et al. (Makhzani et al., 2015) introduced the concept of Adversarial Autoencoders. The idea is to replace the term

$$\mathrm{KL}(q_\phi(z \mid x), p(z)) \tag{5.1}$$

in (2.4) with an adversarial loss that tries to enforce that upon convergence

$$\int q_\phi(z \mid x) p_{\mathcal{D}}(x) \mathrm{d}x \approx p(z). \tag{5.2}$$

While related to our approach, the approach by Makhzani et al. modifies the variational objective while our approach retains the objective.

Interestingly, the approach by Makhzani et al. can be regarded as an approximation to our approach, where $T(x, z)$ is restricted to the class of functions that do not depend on x. Indeed, an ideal discriminator that only depends on z

maximizes

$$\int\int p_{\mathcal{D}}(x)q(z \mid x)\log\sigma(T(z))\mathrm{d}x\mathrm{d}z + \int\int p_{\mathcal{D}}(x)p(z)\log\left(1-\sigma(T(z))\right)\mathrm{d}x\mathrm{d}z \quad (5.3)$$

which is the case if and only if

$$T(z) = \log\int q(z \mid x)p_{\mathcal{D}}(x)\mathrm{d}x - \log p(z). \quad (5.4)$$

Clearly, this simplification is a crude approximation to our formulation from Section 3, but Makhzani et al. (2015) show that this method can still lead to good sampling results. In theory, restricting $T(x,z)$ in this way ensures that upon convergence we approximately have

$$\int q_\phi(z \mid x)p_{\mathcal{D}}(x)\mathrm{d}x = p(z), \quad (5.5)$$

but $q_\phi(z \mid x)$ need not be close to the true posterior $p_\theta(z \mid x)$. Intuitively, while mapping $p_{\mathcal{D}}(x)$ through $q_\phi(z \mid x)$ results in the correct marginal distribution, the contribution of each x to this distribution can be very inaccurate. When using this model as a generative model, the resulting sampler therefore has no guarantee to generate samples according to the true probability of the data point in the data distribution, even if the encoder and decoder networks are allowed to be arbitrarily complex.

5.3. Connection to f-GANs

Nowozin et al. (Nowozin et al., 2016) proposed to generalize Generative Adversarial Networks (Goodfellow et al., 2014) to f-divergences (Ali & Silvey, 1966) based on results by Nguyen et al. (Nguyen et al., 2010). In this paragraph we show that f-divergences allow to represent AVB as a zero-sum two-player game.

The family of f-divergences is given by

$$D_f(p\|q) = \mathrm{E}_p f\left(\frac{q(x)}{p(x)}\right). \quad (5.6)$$

for some convex functional $f: \mathbb{R} \to \mathbb{R}_\infty$ with $f(1) = 0$.

Nguyen et al. (2010) show that using the convex conjugate f^* of f (Hiriart-Urruty & Lemaréchal, 2013)

$$D_f(p\|q) = \sup_T \mathrm{E}_{q(x)}\left[T(x)\right] - \mathrm{E}_{p(x)}\left[f^*(T(x))\right] \quad (5.7)$$

where T is a real-valued function. In particular, this is true for the reverse Kullback-Leibler divergence with $f(t) = t\log t$. We therefore obtain

$$\begin{aligned}\mathrm{KL}(q(z \mid x), p(z)) &= D_f(p(z), q(z \mid x)) \\ &= \sup_T \mathrm{E}_{q(z|x)}T(x,z) - \mathrm{E}_{p(z)}f^*(T(x,z)) \quad (5.8)\end{aligned}$$

with $f^*(\xi) = \exp(\xi - 1)$ the convex conjugate of $f(t) = t\log t$.

All in all, this yields

$$\begin{aligned}&\max_\theta \mathrm{E}_{p_{\mathcal{D}}(x)}\log p_\theta(x) \\ &\quad= \max_{\theta,q}\min_T \mathrm{E}_{p_D(x)}\mathrm{E}_{p(z)}f^*(T(x,z)) \\ &\quad+ \mathrm{E}_{p_D(x)}\mathrm{E}_{q(z|x)}(\log p_\theta(x \mid z) - T(x,z)) \quad (5.9)\end{aligned}$$

By replacing the objective (3.3) for the discriminator with

$$\min_T \mathrm{E}_{p_D(x)}\left[\mathrm{E}_{p(z)}\mathrm{e}^{T(x,z)-1} - \mathrm{E}_{q(z|x)}T(x,z)\right], \quad (5.10)$$

we can reformulate the maximum-likelihood-problem as a min-max zero-sum game. In fact, the derivations from Section 3 remain valid for any f -divergence that we use to train the discriminator. This is similar to the approach taken by Poole et al. (Poole et al., 2016) to improve the GAN-objective. In practice, we observed that the objective (5.10) results in unstable training. We therefore used the standard GAN-objective (3.3), which corresponds to the Jensen-Shannon-divergence.

5.4. Connection to BiGANs

BiGANs (Donahue et al., 2016; Dumoulin et al., 2016) are a recent extension of generative adversarial networks with the goal to add an inference network to the generative model. Similarly to our approach, the authors introduce an adversary that acts on pairs (x,z) of data points and latent codes. However, whereas in BiGANs the adversary is used to optimize the generative and inference networks separately, our approach optimizes the generative and inference model jointly. As a result, our approach obtains good reconstructions of the input data, whereas for BiGANs we obtain these reconstruction only indirectly.

6. Conclusion

We presented a new training procedure for Variational Autoencoders based on adversarial training. This allows us to make the inference model much more flexible, effectively allowing it to represent almost any family of conditional distributions over the latent variables.

We believe that further progress can be made by investigating the class of neural network architectures used for the adversary and the encoder and decoder networks.

Acknowledgements

This work was supported by Microsoft Research through its PhD Scholarship Programme.

References

Ali, Syed Mumtaz and Silvey, Samuel D. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 131–142, 1966.

Dinh, Laurent, Sohl-Dickstein, Jascha, and Bengio, Samy. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Donahue, Jeff, Krähenbühl, Philipp, and Darrell, Trevor. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

Dumoulin, Vincent, Belghazi, Ishmael, Poole, Ben, Lamb, Alex, Arjovsky, Martin, Mastropietro, Olivier, and Courville, Aaron. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Hiriart-Urruty, Jean-Baptiste and Lemaréchal, Claude. *Convex analysis and minimization algorithms I: fundamentals*, volume 305. Springer science & business media, 2013.

Hornik, Kurt, Stinchcombe, Maxwell, and White, Halbert. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, Diederik P, Salimans, Tim, and Welling, Max. Improving variational inference with inverse autoregressive flow. *arXiv preprint arXiv:1606.04934*, 2016.

Larsen, Anders Boesen Lindbo, Sønderby, Søren Kaae, and Winther, Ole. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Liu, Ziwei, Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

Maaløe, Lars, Sønderby, Casper Kaae, Sønderby, Søren Kaae, and Winther, Ole. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

Makhzani, Alireza, Shlens, Jonathon, Jaitly, Navdeep, and Goodfellow, Ian. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Nguyen, Anh, Yosinski, Jason, Bengio, Yoshua, Dosovitskiy, Alexey, and Clune, Jeff. Plug & play generative networks: Conditional iterative generation of images in latent space. *arXiv preprint arXiv:1612.00005*, 2016.

Nguyen, XuanLong, Wainwright, Martin J, and Jordan, Michael I. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

Nowozin, Sebastian, Cseke, Botond, and Tomioka, Ryota. f-gan: Training generative neural samplers using variational divergence minimization. *arXiv preprint arXiv:1606.00709*, 2016.

Poole, Ben, Alemi, Alexander A, Sohl-Dickstein, Jascha, and Angelova, Anelia. Improved generator objectives for gans. *arXiv preprint arXiv:1612.02780*, 2016.

Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Rezende, Danilo Jimenez and Mohamed, Shakir. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Szabo, Zoltán. Information theoretical estimators (ite) toolbox. 2013.

van den Oord, Aaron, Kalchbrenner, Nal, Espeholt, Lasse, Vinyals, Oriol, Graves, Alex, et al. Conditional image generation with pixelcnn decoders. In *Advances In Neural Information Processing Systems*, pp. 4790–4798, 2016a.

van den Oord, Aaron van den, Kalchbrenner, Nal, and Kavukcuoglu, Koray. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016b.

Wu, Yuhuai, Burda, Yuri, Salakhutdinov, Ruslan, and Grosse, Roger. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016.

Appendix

Proofs

This section contains proofs that were ommitted in the main text.

Proposition 1. *For $p_\theta(x \mid z)$ and $q_\phi(z \mid x)$ fixed, the optimal discriminator T^* according to the objective in (3.3) is given by*

$$T^*(x, z) = \log q_\phi(z \mid x) - \log p(z). \tag{3.4}$$

Proof. As in the proof of Proposition 1 in Goodfellow et al. (2014), we rewrite the objective in (3.3) as

$$\int \big(p_{\mathcal{D}}(x) q_\phi(z \mid x) \log \sigma(T(x, z)) + p_{\mathcal{D}}(x) p(z) \log(1 - \sigma(T(x, z)))\big) \mathrm{d}x \mathrm{d}z. \tag{6.1}$$

This integral is maximal as a function of $T(x, z)$ if and only if the integrand is maximal for every (x, z). However, the function

$$t \mapsto a \log(t) + b \log(1 - t) \tag{6.2}$$

attains its maximum at $t = \frac{a}{a+b}$, showing that

$$\sigma(T^*(x, z)) = \frac{q_\phi(z \mid x)}{q_\phi(z \mid x) + p(z)} \tag{6.3}$$

or, equivalently,

$$T^*(x, z) = \log q_\phi(z \mid x) - \log p(z). \tag{6.4}$$

□

Proposition 3. *If (θ^*, ϕ^*, T^*) defines a Nash-equilibrium of the two-player game defined by (3.3) and (3.9), then*

$$T^*(x, z) = \log q_{\phi^*}(z \mid x) - \log p(z) \tag{3.10}$$

and (θ^*, ϕ^*) is a global optimum of the variational lower bound in (2.4).

Proof. If (θ^*, ϕ^*, T^*) defines a Nash-equilibrium, Proposition 1 shows (3.10). Inserting (3.10) into (3.5) shows that (ϕ^*, θ^*) maximizes

$$\mathrm{E}_{p_{\mathcal{D}}(x)} \mathrm{E}_{q_\phi(z|x)} \big(- \log q_{\phi^*}(z \mid x) + \log p(z) + \log p_\theta(x \mid z)\big) \tag{6.5}$$

as a function of ϕ and θ. A straightforward calculation shows that (6.5) is equal to

$$\mathcal{L}(\theta, \phi) + \mathrm{E}_{p_{\mathcal{D}}(x)} \mathrm{KL}(q_\phi(z \mid x), q_{\phi^*}(z \mid x)) \tag{6.6}$$

where

$$\mathcal{L}(\theta, \phi) := \mathrm{E}_{p_{\mathcal{D}}(x)} \Big[- \mathrm{KL}(q_\phi(z \mid x), p(z)) + \mathrm{E}_{q_\phi(z|x)} \log p_\theta(x \mid z) \Big] \tag{6.7}$$

is the variational lower bound in (2.4).

Notice that (6.6) evaluates to $\mathcal{L}(\theta^*, \phi^*)$ when we insert (θ^*, ϕ^*) for (θ, ϕ).

Assume now, that (θ^*, ϕ^*) does not maximize the variational lower bound $\mathcal{L}(\theta, \phi)$. Then there is (θ', ϕ') with

$$\mathcal{L}(\theta', \phi') > \mathcal{L}(\theta^*, \phi^*). \tag{6.8}$$

Inserting (θ', ϕ') for (θ, ϕ) in (6.6) we obtain

$$\mathcal{L}(\theta', \phi') + \mathrm{E}_{p_{\mathcal{D}}(x)} \mathrm{KL}(q_{\phi'}(z \mid x), q_{\phi^*}(z \mid x)), \tag{6.9}$$

which is strictly bigger than $\mathcal{L}(\theta^*, \phi^*)$, contradicting the fact that (θ^*, ϕ^*) maximizes (6.6). Together with (3.10), this proves the theorem. □