Importance Weighted Adversarial Variational Bayes

Marta Gómez-Sancho and Daniel Hernández-Lobato

Computer Science Department, Universidad Autónoma de Madrid,
Francisco Tomás y Valiente 11, 28049, Madrid, Spain
marta.gomez@uam.es, daniel.hernandez@uam.es

Abstract. Adversarial variational Bayes (AVB) can infer the parameters of a generative model from the data using approximate maximum likelihood. The likelihood of deep generative models model is intractable. However, it can be approximated by a lower bound obtained in terms of an approximate posterior distribution of the latent variables of the data q. The closer q is to the actual posterior, the tighter the lower bound is. Therefore, by maximizing the lower bound one should expect to also maximize the likelihood. Traditionally, the approximate distribution q is Gaussian. AVB relaxes this limitation and allows for flexible distributions that may lack a closed-form probability density function. Implicit distributions obtained by letting a source of Gaussian noise go through a deep neural network are examples of these distributions. Here, we combine AVB with the importance weighted autoencoder, a technique that has been shown to provide a tighter lower bound on the marginal likelihood. This is expected to lead to a more accurate parameter estimation of the generative model via approximate maximum likelihood. We have evaluated the proposed method on three datasets, MNIST, Fashion MNIST, and Omniglot. The experiments show that the proposed method improves the test log-likelihood of a generative model trained using AVB.

Keywords: Variational Autoencoder · Importance Weighted Autoencoder · Adversarial Variational Bayes · Generative Models

1 Introduction

Generative models can generate new data very similar to the observed data. A popular generative model assumes that the observed data has been generated by letting a random source of noise (*e.g.*, Gaussian distributed noise) go through a strong non-linear function such as the one corresponding to a deep neural network. This is how variational autoencoders (VAEs) and generative adversarial networks (GANs) work [10,7]. The task of interest is how to infer the parameters of the non-linear function that better explain the data. In general, however, this is a difficult task. Simple approaches such as maximum likelihood estimation fail because the likelihood of the model has no analytic form. Its computation involves marginalizing the input noise. The resulting integral is too complicated as a consequence of the strong non-linearities of the deep neural network.

Two approaches can be used to overcome the problems described. The first one, employed in GANs, consists in using a discriminator to evaluate the quality of the generative model inferred so far [7]. The discriminator is trained to learn to correctly classify data points as coming from the training data set or as coming from the generator. Then, the generator is trained to make things *difficult* for the discriminator. If the training process is carried out correctly, the result is a generator that outputs data very similar to the observed data [16]. Learning, however, becomes difficult as the optimization problem that infers the parameters of the generator is a max-min problem [7]. Thus, GAN training is sometimes unstable and fails to produce meaningful results [2]. In spite of this, the images generated by GANs are often described are very realistic [16].

A second approach for learning generative models approximates the likelihood using approximate inference. In the variational autoencoder (VAE) a lower bound on the log-likelihood of the model is provided [10]. This lower bound is obtained in terms of an approximate distribution q that targets the posterior distribution of the latent variables of the data (*i.e.*, the noise variables used to generate the data). The missing part of the lower bound is the Kullback-Leibler (KL) divergence between q and the exact posterior. Therefore, maximizing the lower bound is equivalent to minimizing the KL between q and the actual posterior. Often, q is set to be a Gaussian, which guarantees that the lower bound can be easily approximated and optimized using stochastic optimization. Furthermore, amortized variational inference relates the parameters of q to the actual observed data point $\mathbf{x}$ [19]. This non-linear relation is specified in terms of a deep neural network. Thus, in the VAE one obtains for free a recognition model (*i.e.*, q) that infers the latent variables used to generate each instance.

The VAE can be improved by considering an average over several samples from q to evaluate the lower bound. This method is known in the literature as the importance weighted autoencoder (IWAE) [4]. The IWAE relies on importance variational inference [5] and the lower-bound obtained can be proved to be tighter than the one of the VAE. Therefore, the IWAE carries out a process that is closer to maximum likelihood estimation. This is translated into better parameter learning of the generative model and better log-likelihood results on validation data. The drawback is that K samples are needed to approximate the lower bound, which is K times more expensive.

A limitation of the VAE is that q is often restricted to be Gaussian. The reason for this is that it simplifies the evaluation of the objective to be optimized. A Gaussian distribution, however, may be far from the actual posterior. This introduces some bias in the objective that is optimized. Namely, the lower bound. Specifically, the difference between the log-likelihood of the training data and the lower bound is the KL divergence between q and the actual posterior. A Gaussian approximate distribution q is expected to suffer from approximation bias. Therefore, by increasing the flexibility of q it is possible to make the lower bound tighter, which should lead to better parameters estimation.

The approximation bias described is alleviated by using more flexible distributions q. This is how adversarial variational Bayes (AVB) works [15]. A flexible model for q is one similar to the generative model considered in GANs or VAEs [7,10]. Namely, a non- linear function that receives as input a Gaussian noise. These are known as wild-variational approximations and also as implicit distributions [13,8]. If the non-linear function is complex enough, *e.g.*, it is a deep neural network, and the number of dimensions of the noise is big enough, such a distribution can generate almost anything, as illustrated by the expressive power of GANs [16]. These distributions are easy to sample from. For this, one only has to generate random noise and let it go through the non-linearity. Nevertheless, evaluating the probability density function (p.d.f.) is complicated since it requires the marginalization of the noise, which is intractable. The log-ratio between q and the prior distribution for the latent variables of the model appears in the lower bound of the VAE. Therefore, using an implicit distribution q makes things difficult. AVB solves this problem by approximating the log-ratio using a flexible classifier that discriminates between samples from q and the prior. It is possible to show that the optimal classifier is precisely given by such log-ratio. This classifier can be trained simultaneously as the generative model. Therefore, AVB allows to carry out approximate inference using more flexible distributions q that may lead to tighter lower bounds on the log-likelihood of the model. This has been shown to lead to better parameter estimation [15].

We improve AVB and the IWAE by combining both approaches. That is, we consider the lower bound that is optimized in the IWAE and we employ a flexible approximate distribution q that is specified as an implicit model. Namely, a distribution that is easy to sample from but that has no closed-form p.d.f. The lower bound of the IWAE also requires the estimation of the log-ratio between the approximate distribution q and the prior. To address this problem we use the trick employed in AVB. That is, we use a flexible classifier, specified by a deep neural network, to discriminate between samples from q and the prior. We refer to such an approach as importance weighted adversarial variational Bayes (IWAVB). We evaluate the performance of the this method on the MNIST, the Fashion MNIST, and the Omniglot datasets. The results obtained show that IWAVB improves the results of AVB. More precisely, it infers the parameters of the generative model so that the log-likelihood on test data is higher.

The rest of the paper is organized as follows: Section 2 introduces the VAE. We also review here the IWAE and AVB as two improvements of standard VAEs. Then, in Section 3 we described the proposed approach for learning deep generative models. This approach combines all the advantages of the IWAE and AVB for approximate inference. Section 4 describes important related work. Section 5 shows the experiments of this paper in which have evaluated the proposed approach, IWAVB. Finally, Section 6 gives the conclusions of this work.

2 Variational Autoencoders

We describe the problem of learning a generative model of the observed data and how this task can be carried out by using the variational autoencoder (VAE). We also describe here how this model can be improved by considering the importance weighted autoencoder (IWAE) and adversarial variational Bayes (AVB). Consider some observed data $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, with d the dimensionality of the data. We assume that these data have been generated by the following model:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}\,, \tag{2.1}$$

where $\mathbf{z} \in \mathbb{R}^l$ are latent variables associated to $\mathbf{x}$ so that $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Therefore, l is the dimensionality of the latent space. Furthermore, we assume that $p_\theta(\mathbf{x}|\mathbf{z})$ is a conditional distribution parameterized by θ. In the case that $\mathbf{x} \in \mathbb{R}^d$, an example of this distribution is a deep neural network that will output the means and diagonal variances of a multi-variate Gaussian distribution for $\mathbf{x}$. In the case of binary data. That is, when $\mathbf{x} \in \{0,1\}^d$, the conditional distribution can be a deep neural network that will output the activation probabilities of a product of Bernoulli distributions. Namely, in each case we have that:

$$p_\theta(\mathbf{x}|\mathbf{z}) = \prod_{j=1}^d \mathcal{N}(x_j|\mu_j^\theta(\mathbf{z}), \nu_j^\theta(\mathbf{z}))\,, \qquad p_\theta(\mathbf{x}|\mathbf{z}) = \prod_{j=1}^d \text{Bern}(x_j|\mu_j^\theta(\mathbf{z}))\,, \tag{2.2}$$

where $\mathcal{N}(\cdot|m, v)$ denotes a Gaussian density with mean m and variance v, and $\text{Bern}(\cdot|\mu_j^\theta(\mathbf{z}))$ is the probability mass function of a Bernoulli random variable with activation probability $\mu_j^\theta(\mathbf{z})$.

The task of interest is how to infer θ, *i.e.*, the parameters of $p_\theta(\mathbf{x}|\mathbf{z})$ given $\mathbf{X}$. The maximum likelihood principle can be used for this task [3]. The problem is that the marginalization of $\mathbf{z}$ in (2.1) is intractable. To overcome this, the VAE introduces an approximate distribution $q_\phi(\mathbf{z}|\mathbf{x})$ targeting $p(\mathbf{z}|\mathbf{x})$ and considers the following decomposition of the log-likelihood of the observed data [10]:

$$\log p_\theta(\mathbf{x}) = \mathcal{L}_{\phi,\theta}(\mathbf{x}) + \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}))|p(\mathbf{z}|\mathbf{x}))\,, \tag{2.3}$$

where

$$\mathcal{L}_{\phi,\theta}(\mathbf{x}) = \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} d\mathbf{z}\,, \tag{2.4}$$

$$\text{KL}(q_\phi(\mathbf{z}|\mathbf{x}))|p(\mathbf{z}|\mathbf{x})) = -\int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} d\mathbf{z} \geq 0\,. \tag{2.5}$$

Furthermore, the approximate distribution is constrained to be Gaussian with parameters specified non-linearly in terms of $\mathbf{x}$ by a deep neural network:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \prod_{j=1}^l \mathcal{N}(z_j|m_j^\phi(\mathbf{x}), v_j^\phi(\mathbf{x}))\,. \tag{2.6}$$

The second term in the r.h.s. of (2.3) is the Kullback-Leibler divergence between q and the actual posterior. The first term, *i.e.*, $\mathcal{L}(\mathbf{x})$, is hence a lower bound on the log marginal likelihood. Namely, $\log p(\mathbf{x}) \geq \mathcal{L}_{\phi,\theta}(\mathbf{x})$. Thus, a maximization of $\mathcal{L}_{\phi,\theta}(\mathbf{x})$ with respect to ϕ is equivalent to minimizing $\text{KL}(q_\phi(\mathbf{z}|\mathbf{x}))|p(\mathbf{z}|\mathbf{x}))$. Furthermore, at the maximum, $q_\phi(\mathbf{z}|\mathbf{x})$ is expected to be a good approximation to $p(\mathbf{z}|\mathbf{x})$ and hence $\text{KL}(q_\phi(\mathbf{z}|\mathbf{x}))|p(\mathbf{z}|\mathbf{x}))$ should be small. In that case, a maximization of $\mathcal{L}_{\phi,\theta}(\mathbf{x})$ with respect to θ, the parameters of the generative model, is expected to also maximize $\log p_\theta(\mathbf{x})$. Specifically, the VAE objective is:

$$\sum_{i=1}^{N} \mathcal{L}_{\theta,\phi}(\mathbf{x}_i) = \sum_{i=1}^{N} \mathbb{E}_{\mathbf{z}_i \sim q_\phi(\mathbf{z}_i|\mathbf{x}_i)}[\log p_\theta(\mathbf{x}_i|\mathbf{z}_i)] - \text{KL}(q_\phi(\mathbf{z}_i|\mathbf{x}_i)|p(\mathbf{z}_i))\,, \tag{2.7}$$

which is maximized simultaneously with respect to ϕ and θ.

The second term in (2.7) is the KL divergence between two Gaussian distributions, which can be computed analytically. The first term in (2.7), however, has no closed-form expression. This term (and its gradients) can be approximated by Monte Carlo methods. In particular, one can resort to the reparametrization trick [10]. For this, one generates a sample from $q_\phi(\mathbf{z}_i|\mathbf{x}_i)$, $\tilde{\mathbf{z}}_i$, by first generating a standard Gaussian random variable $\boldsymbol{\epsilon}_i$ to then apply a transformation so that $\tilde{\mathbf{z}}_i = g_\phi(\mathbf{x}_i, \boldsymbol{\epsilon}_i)$. In the case of the Gaussian distribution, this transformation is always possible. One just has to multiply each component $j = 1, \ldots, l$ of $\boldsymbol{\epsilon}_i$ by $\sqrt{v_j^\phi(\mathbf{x})}$ to then add $m_j^\phi(\mathbf{x})$. The noisy estimate of (2.7) is then,

$$\frac{N}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \log p_\theta(\mathbf{x}_i|\tilde{\mathbf{z}}_i) - \text{KL}(q_\phi(\mathbf{z}_i|\mathbf{x}_i)|p(\mathbf{z}_i))\,, \tag{2.8}$$

where we have considered a mini-batch $\mathcal{B}$ of data points to scale to large datasets. A unbiased gradient estimate can be obtained from (2.8) using the chain rule. Thus, (2.7) can be easily optimized using stochastic optimization [10].

2.1 Importance Weighted Variational Autoencoders

The VAE can be improved by considering the importance weighted autoencoder (IWAE) [4]. In this method for learning the parameters θ of the generative model it is considered a tighter lower bound that the one of the VAE. Namely,

$$\hat{\mathcal{L}}_{\theta,\phi}^{K}(\mathbf{x}) = \mathbb{E}_{\mathbf{z}^1,\mathbf{z}^2,\ldots,\mathbf{z}^K \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(\mathbf{x}|\mathbf{z}^k)p(\mathbf{z}^k)}{q_\phi(\mathbf{z}^k|\mathbf{x})} \right]. \tag{2.9}$$

This is a lower bound on $\log p_\theta(\mathbf{x})$ as follows from Jensen's inequality and the fact that the average is an unbiased estimator:

$$\mathbb{E}\left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(\mathbf{x}|\mathbf{z}^k)p(\mathbf{z}^k)}{q_\phi(\mathbf{z}^k|\mathbf{x})} \right] \leq \log \mathbb{E}\left[\frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(\mathbf{x}|\mathbf{z}^k)p(\mathbf{z}^k)}{q_\phi(\mathbf{z}^k|\mathbf{x})} \right] = \log p_\theta(\mathbf{x})\,, \tag{2.10}$$

where the expectations are the same as the one in (2.9).

When the number of samples K equals 1, (2.9) coincides with (2.4). When $K > 1$, it is expected that the variance inside of the log in (2.9) is reduced. In particular, $\hat{\mathcal{L}}^{K+1}_{\theta,\phi}(\mathbf{x}) \geq \hat{\mathcal{L}}^{K}_{\theta,\phi}(\mathbf{x})$, as proved in [4]. Therefore, one can obtain a tighter lower bound simply by increasing K. The IWAE objective can also be approximated stochastically as in (2.7). In this case, however, the KL divergence is not contained in the objective. One must include the ratio between $q_\phi(\mathbf{z}|\mathbf{x})$ and the prior $p(\mathbf{z})$ in the stochastic estimate. Namely,

$$\frac{N}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \log \frac{1}{K}\sum_{k=1}^{K} \frac{p_\theta(\mathbf{x}_i|\tilde{\mathbf{z}}_i^k)p(\tilde{\mathbf{z}}_i^k)}{q_\phi(\tilde{\mathbf{z}}_i^k|\mathbf{x}_i)}\,, \tag{2.11}$$

where we consider K samples from $q_\phi(\mathbf{z}_i|\mathbf{x}_i)$ instead of just one, as in the VAE. This objective leads to better estimation of the parameters of the generative model θ when $K > 1$. More precisely, by considering $K = 5$ and $K = 50$ samples, better test log-likelihood results are obtained [4].

2.2 Adversarial Variational Bayes

A limitation of the VAE (and also the IWAE) is that the approximate distribution $q_\phi(\mathbf{z}|\mathbf{x})$ is Gaussian. This introduces some bias in the estimation of the model parameters. In particular, in the VAE the difference between the $\mathcal{L}_{\phi,\theta}(\mathbf{x})$ and the actual log-likelihood of the data $\log p_\theta(\mathbf{x}))$ is the KL divergence between $q_\phi(\mathbf{z}|\mathbf{x})$ and the exact posterior distribution $p(\mathbf{z}|\mathbf{x})$. This means that the quantity that is optimized by the VAE need not be equal to the expected optimal one. Namely, the log-likelihood of the model parameters θ.

Adversarial variational Bayes (AVB) is a technique that can be used to consider flexible distributions q_ϕ [15]. Examples of these distributions include implicit distributions also known as wild-approximations [13]. These are distributions that are easy to sample from but that may lack an analytical expression for the probability density. For example, consider the approximate distribution:

$$\mathbf{z} = f_\phi(\mathbf{x},\mathbf{e})\,, \quad \mathbf{e} \sim \mathcal{N}(\mathbf{0},\mathbf{I})\,, \quad q_\phi(\mathbf{z}|\mathbf{x}) = \int \delta(\mathbf{z} - f_\phi(\mathbf{x},\mathbf{e}))\mathcal{N}(\mathbf{e}|\mathbf{0},\mathbf{I})d\mathbf{e}\,, \tag{2.12}$$

where $\delta(\cdot)$ is a point of probability mass and $f_\phi(\cdot,\cdot)$ is a non-linear function, *e.g.*, a deep neural network. If the dimensionality of $\mathbf{e}$ is large enough and the complexity of f_ϕ is big enough, one can generate almost anything. This is precisely the approach used in GANs or the VAE to generate data [10,7], which can generate very complex data [16].

A problem, however, is that distributions such as (2.12) lack closed-form probability densities. This makes difficult optimizing the objective of the VAE in (2.8). Specifically, it is required to estimate the log-ratio between q and the prior, in order to evaluate (2.8), as the KL divergence between q and the prior depends on this log-ratio. See (2.4) for further details. AVB provides an elegant solution to this problem. For this, it uses the fact that the log-ratio between q and the prior is given by the output of an optimal classifier that solves the problem of discriminating samples from the prior and from q. More precisely,

$$\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})|p(\mathbf{z})) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right] = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[T_{\omega^\star}(\mathbf{x},\mathbf{z})\right]\,, \tag{2.13}$$

where $T_{\omega^\star}(\mathbf{x}, \mathbf{z})$ is the output of the the optimal classifier. This classifier can be implemented as deep neural network with parameters ω. If $T_\omega(\mathbf{x}, \mathbf{z})$ is flexible enough it should approximate the log-ratio very accurately [15]. The objective that is considered for training the discriminator, assuming q is fixed, is:

$$\max_{\omega} \quad \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \sigma(T_\omega(\mathbf{x}, \mathbf{z}))\right)] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} \left[\log(1 - \sigma(T_\omega(\mathbf{x}, \mathbf{z})))\right)] \,. \tag{2.14}$$

where $\sigma(\cdot)$ is the sigmoid activation function. It is possible to show that the optimal $T_{\omega^\star}(\mathbf{z}, \mathbf{x})$ that maximizes (2.14) is given precisely by $\log q_\phi(\mathbf{z}|\mathbf{x}) - \log p(\mathbf{z})$ [15]. Note that (2.14) can be optimized using stochastic optimization. It is equivalent to training a deep neural network to solve a binary classification task.

Given $T_{\omega^\star}(\mathbf{z}, \mathbf{x})$, the objective of AVB for learning the parameters of the generative model is obtained by introducing in (2.8) the output of such a classifier:

$$\frac{N}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \log p_\theta(\mathbf{x}_i|\tilde{\mathbf{z}}_i)] - T_{\omega^\star}(\tilde{\mathbf{z}}_i, \mathbf{x}_i) \,. \tag{2.15}$$

To optimize this objective we need to differentiate with respect to ϕ. This may be complicated since $T_{\omega^\star}(\mathbf{z}, \mathbf{x})$ depends on ϕ. However, due to the expression for the optimal discriminator, it can be showed that $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left(\nabla_\phi T_{\omega^\star}(\mathbf{z}, \mathbf{x})\right) = 0$. Therefore the dependence of $T_{\omega^\star}(\mathbf{z}, \mathbf{x})$ w.r.t ϕ can be ignored [15]. In practice, both q_ϕ and the discriminator $T_\omega(\mathbf{z}, \mathbf{x})$ are trained simultaneously. However, q_ϕ is updated by maximizing (2.15) using a smaller learning rate than the one used to update the discriminator T_ω, which considers (2.14). Several experiments show that an implicit distribution for q improves the test log-likelihood results [15].

3 Importance Weighted Adversarial Variational Bayes

We propose to combine both the IWAE, which is able to optimize a tighter lower bound than the one considered by the VAE, and AVB to allow for approximate inference using an implicit approximate distribution q_ϕ. This is expected to lead to better optimization results of the parameters of the generative model θ. In the case of IWAE, however, the missing term (*i.e.*, the difference between the log-likelihood of the model parameters θ and the lower bound) is not the KL divergence between q and the actual posterior. Nevertheless, this method can also benefit from a more flexible distribution q. Specifically, the IWAE objective is an importance sampling estimate [4]. The optimal sampling distribution in such an estimate is the actual posterior distribution $p(\mathbf{z}|\mathbf{x})$. If that distribution is employed, it is possible to show that the objective in (2.9) coincides with the marginal likelihood of the data $\log p_\theta(\mathbf{x})$.

Even though the IWAE can benefit from using an implicit distribution as the approximate distribution q_ϕ, it is not trivial how to employ this distribution in practice. More precisely, the objective in (2.9) requires the computation of the ratio between the prior $p(\mathbf{z})$ and approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$. We propose to estimate this ratio using the approach of AVB. Namely, by using the output of a near-optimal classifier that discriminates between samples from these two distributions. The lower bound that we consider is hence

$$\begin{aligned}\tilde{\mathcal{L}}^K_{\theta,\phi}(\mathbf{x}) &= \mathbb{E}_{\mathbf{z}^1,\mathbf{z}^2,\ldots,\mathbf{z}^K\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{1}{K}\sum_{k=1}^K \frac{p_\theta(\mathbf{x}|\mathbf{z}^k)p(\mathbf{z}^k)}{q_\phi(\mathbf{z}^k|\mathbf{x})}\right] \\ &= \mathbb{E}_{\mathbf{z}^1,\mathbf{z}^2,\ldots,\mathbf{z}^K\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{1}{K}\sum_{k=1}^K \exp\left\{\log p_\theta(\mathbf{x}|\mathbf{z}^k) - T_{\omega^\star}(\mathbf{z}^k,\mathbf{x})\right\}\right] . \quad (3.1)\end{aligned}$$

and the objective is given by

$$\frac{N}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \log \frac{1}{K}\sum_{k=1}^K \exp\left\{\log p_\theta(\mathbf{x}|\mathbf{z}^k) - T_{\omega^\star}(\mathbf{z}^k,\mathbf{x})\right\} . \quad (3.2)$$

The optimal discriminator $T_{\omega^\star}(\mathbf{z}^k,\mathbf{x})$ can be trained as in AVB by optimizing the objective in (2.14). This can be carried out using the specific details employed in AVB, *e.g.*, training all the networks at the same time, and using a bigger learning rate to train the discriminator. We expect that the optimization of this tighter lower bound results in a better parameter estimation.

4 Related Work

There are other techniques that have been proposed to allow for implicit models besides adversarial variational Bayes. In [20] is it described a method to obtain an unbiased estimate of the gradients of the lower bound when an implicit model is used to approximate the posterior distribution. This estimate relies on Markov chain Monte Carlo techniques to approximate the posterior distribution of the noise $\mathbf{e}$ that was used to generate each $\mathbf{z}$. Even though this approach works in practice its implementation is difficult as simulating a Markov chain in frameworks such as Tensorflow require coding loops which are often computationally expensive and cannot be accelerated on GPUs [1].

Another approach to obtain flexible approximate distributions q is normalizing flows (NF) [17]. NF starts with a simple distribution q, *e.g.*, Gaussian, whose samples are modified using non-linear invertible transformations. If these transformations are chosen carefully, the p.d.f. of the resulting distribution can be evaluated in closed-form, avoiding the problems of implicit models for q that lack a closed-form p.d.f. The problem of NF is that the family of transformations is limited, since it has to be invertible, which may constrain the flexibility of q. The implicit model considered in our work does not have these restrictions and is hence expected to be more flexible.

Stein Variational Gradient Descent transforms a set of *particles* to match the posterior distribution [14]. This technique is competitive with state-of-the-art methods, but the main drawback is that many particles (points) need to be stored in memory to accurately represent the posterior. This can be a computational bottle-neck. The number of samples is fixed initially, and these samples or particles are optimized during training. In problems with a high dimensional latent space this can be problematic. Our approach only needs to generate a few samples to obtained better results, *i.e.*, 5 or 10 samples.

In [18] variational inference and MCMC methods are combined to obtain flexible approximate distributions. The key is to use a Markov chain as the approximate distribution q. The parameters of the Markov chain can be adjusted to match as close as possible the target distribution in terms of the KL divergence. This is an interesting idea, but is also limited by the difficulty of evaluating the p.d.f. of q, as in the case of AVB. In [18] this problem is addressed by learning a backward model, that infers the initial state of the Markov chain given the generated samples. Learning this backward model accurately is expensive and parametric models have to be used in practice, which may introduce some bias.

Another approach derives a lower bound that is looser than the one considered by the VAE [23]. However, this lower bound can be evaluated in closed-form when an implicit distribution is used as the approximate distribution. This work does not have the problems of AVB, in which extra models have to be used to approximate the log-ratio between the approximate distribution q and the prior. A problem, however, is that the looser lower bound can lead to sub-optimal approximation results. Nevertheless, the experiments carried out in that work indicate that some gains are obtained.

Finally, approximate inference by using importance weights is analyzed in detail in [6]. That paper shows that such a method optimizes a KL divergence between the approximate distribution q an implicit posterior distribution. That paper also extends the ideas of importance weighted approximate inference to general probabilistic graphical models, not only generative models as in [4].

While the proposed method to account for an implicit approximate distribution q is simple and can be easily codified in modern frameworks for machine learning such as Tensorflow [1], it suffers from the limitation of having to train a discriminator, in parallel, in order to optimize the lower bound in (3.2).

5 Experiments

We have carried out experiments on several datasets to evaluate the performance of the proposed method. The datasets considered are MNIST [12], Fashion MNIST [22], and Omniglot [11]. The number of instances in each of these datasets are 70,000, 70,000 and 32,640, respectively. Each instance is a 28×28 gray-scale image (in Omniglot we down-sample the images from an initial resolution of 105×105). We use 55,000 images for training in MNIST and Fashion MNIST, 5,000 for validation, and the rest for testing. In the case of the Omniglot dataset we use 5,000 instances for validation, 5,000 instances for testing and the rest for training. Fig. 1 shows 50 images extracted from each datasets.

In each dataset we train the proposed method, IWAVB, considering a different number of samples. Namely, $K = 1, 5$ and 10 samples. Importantly, for $K = 1$ samples IWAVB reduces to AVB, which allows to compare results with that method. We considered two potential values for the dimensionality of the latent space $l = 8$ and $l = 32$ and report results for both of them. In each exper-

Fig. 1. Sample images extracted from the MNIST dataset (left), Fashion MNIST (middle) and Omniglot (right). All images are of size 28 × 28 pixels in gray scale.

iment the generator, the non-linear function of the approximate distribution q and the classifier used to approximate the log-ratio use the same architecture as in [15]. That is, they are convolutional neural networks. The dimensionality of the noise injected in the implicit model is set to 32. We use ADAM for training the models with a learning rate of $5 \cdot 10^{-5}$ for the generative model and 10^{-4} for the classifier [9]. The number of steps is set to 150000. All the computations have been carried out using Tensorflow [1] and a Tesla P100 GPU. We use the adaptive contrast technique, as described in [15], to improve the results of the log-ratio estimation. The test log-likelihood is estimated using annealed-importance sampling with 8 parallel chains run for 1000 steps [21]. Validation data are used to track the overall progress of the training process.

The results obtained are displayed in Table 1. This table shows the average negative test log-likelihood (the lower the better) of IWAVB on each dataset in terms of the number of samples considered K and the dimensionality of the latent space, *i.e.*, l. We observe that IWAVB always improves results with K and that most of the times the best results correspond to the larger number of samples, as expected, since this results in a tighter lower bound of the log-likelihood associated to the training data. We also observe that increasing the dimensionality of the latent space improves results in general. The datasets considered in our experiments have a large number of samples. Therefore, we have only considered a single train / test partition of the data. It is hence difficult to obtain error bars on the estimated quantities. Nevertheless, the fact that always the best results correspond to $K > 1$ gives evidence supporting that the proposed approach performs better.

From these experiments we can conclude that the proposed method is able to improve the results of estimating the parameters of the generative model θ. In particular, it seems to always improve the results of AVB in terms of the log-likelihood of test data. Recall that AVB correspond to IWAVB with $K = 1$.

Table 1. Neg. test log-likelihood (NLL) on each dataset in terms of the latent space dimensionality (l) and the number of samples (K) used to compute the lower bound.

Dataset	l	K	**NLL**	**Dataset**	l	K	**NLL**	**Dataset**	l	K	**NLL**
MNIST	8	1	90.23	Fashion	8	1	231.25	Omniglot	8	1	156.62
MNIST	8	5	**90.16**	Fashion	8	5	230.34	Omniglot	8	5	**142.99**
MNIST	8	10	90.19	Fashion	8	10	**229.43**	Omniglot	8	10	152.70
MNIST	32	1	80.74	Fashion	32	1	226.96	Omniglot	32	1	91.54
MNIST	32	5	80.49	Fashion	32	5	227.16	Omniglot	32	5	**91.25**
MNIST	32	10	**79.77**	Fashion	32	10	**225.14**	Omniglot	32	10	91.43

Importantly, the results obtained are also similar and even better than the ones reported in [15]. We conclude that a combination of the IWAVB with implicit models results in better generative models in terms of the test log-likelihood.

6 Conclusions

We have proposed a novel method for training deep generative models. Training these models is challenging because the likelihood lacks a closed-form expression. A method that overcomes this problem is the variational autoencoder (VAE), which maximizes a lower bound on the log-likelihood of the model [10]. This lower bound can be made tighter by considering extra samples from an approximate distribution q that targets the posterior distribution of the latent variables of the data $\mathbf{z}$. This is how the importance weighted autoencoder (IWAE) works. An orthogonal approach to improve results considers an implicit model for the approximate distribution q, which is constrained to be Gaussian in the VAE and the IWAE. This technique is known as adversarial variational Bayes (AVB) [15]. A difficulty, however, is that evaluating the lower bound when q is implicit is no longer tractable, since the log-ratio between q and the prior distribution for the latent variables is required. AVB uses the output of a classifier that discriminates between samples from q and the prior to estimate the log-ratio.

In this paper we have combined the the IWAE to obtain a tighter lower bound on the log-likelihood of the generative model and AVB, which allows to consider an implicit distribution q that need not be Gaussian. Our hypothesis was that the tighter lower bound of the IWAE combined with the extra flexibility of an implicit model for q should lead to better generative models. We have validated the proposed approach on several experiments involving gray-scale images of size 28 × 28 pixels extracted from the MNIST, Fashion MNIST, and Omniglot datasets. In these experiments we have observed that the proposed approach always improves the results of AVB. Furthermore, using a bigger number of samples in the approximation of the log-likelihood of the generative model seems to improve results most of the times, which is the expected behavior, since more samples imply a tighter lower bound that is closer to the actual log-likelihood.

Acknowledgments: We acknowledge the use of the facilities of Centro de Computación Científica at UAM and support from the Spanish Plan Nacional I+D+i (grants TIN2016-76406-P, TEC2016-81900-REDT and PID2019-106827GB-I00) and from Comunidad de Madrid (grant PEJ-2017-AI_TIC-6464).

References

1. Abadi et al., M.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `https://www.tensorflow.org/`, available from tensorflow.org
2. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. In: ICLR (2017)
3. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer (2006)
4. Burda, Y., Grosse, R.B., Salakhutdinov, R.: Importance weighted autoencoders. In: ICLR (2016)
5. Domke, J., Sheldon, D.R.: Importance weighting and variational inference. In: NIPS. pp. 4470–4479 (2018)
6. Domke, J., Sheldon, D.R.: Importance weighting and variational inference. In: NIPS, pp. 4470–4479 (2018)
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS, pp. 2672–2680 (2014)
8. Huszár, F.: Variational inference using implicit distributions. arXiv preprint arXiv:1702.08235 (2017)
9. Kingma, D.P., Ba, J.: ADAM: a method for stochastic optimization. In: ICLR. pp. 1–15 (2015)
10. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: ICLR (2014)
11. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. Science **350**, 1332–1338 (2015)
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**, 2278–2324 (1998)
13. Li, Y., Liu, Q.: Wild variational approximations. In: NIPS workshop on advances in approximate Bayesian inference (2016)
14. Liu, Q., Wang, D.: Stein variational gradient descent: A general purpose Bayesian inference algorithm. In: NIPS. pp. 2378–2386 (2016)
15. Mescheder, L.M., Nowozin, S., Geiger, A.: Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In: ICML. pp. 2391–2400 (2017)
16. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR (2016)
17. Rezende, D.J., Mohamed, S.: Variational inference with normalizing flows. In: ICML. pp. 1530–1538 (2016)
18. Salimans, T., Kingma, D., Welling, M.: Markov chain Monte Carlo and variational inference: Bridging the gap. In: ICML. pp. 1218–1226 (2015)
19. Shu, R., Bui, H.H., Zhao, S., Kochenderfer, M.J., Ermon, S.: Amortized inference regularization. In: NIPS. pp. 4393–4402 (2018)
20. Titsias, M.K., Ruiz, F.J.R.: Unbiased implicit variational inference. In: Artificial Intelligence and Statistics. pp. 167–176 (2019)
21. Wu, Y., Burda, Y., Salakhutdinov, R., Grosse, R.: On the quantitative analysis of decoder-based generative models. arXiv preprint arXiv:1611.04273 (2016)
22. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017)
23. Yin, M., Zhou, M.: Semi-implicit variational inference. In: ICML. pp. 5660–5669 (2018)