On Variational Auto-Encoders

Recently, Variational Auto-Encoders (VAEs) have emerged as an influential model in the domain of unsupervised learning. VAEs are appealing because they can be derived beautifully in statistical terms, they are based on powerful function approximators, and can be trained by stochastic gradient descent. In this paper, we (1) derive the formulation of VAEs and explain the intuition behind the model, (2) summarize recent results on bounds on the reconstruction error.

1 Introduction

Unsupervised Learning is a long-standing problem in the field of high-dimensional Statistics and Machine Learning. Given a training set of high-dimensional oberservations $X_n = \{x_1, x_2, \ldots, x_n\} \subset \mathcal{X}$ from an unknown distribution we would like to draw samples $x \sim P_\theta(x)$ such that it resembles the observations. We assume an underyling generating process draws the observations X_n from a lower-dimensional manifold. That is, once we have learned the manifold we can draw previously unseen samples. Since we are only given the observations X_n in this setting, we shall adopt an *Auto-Encoder* regime to capture the manifold. This model consists of a composition of (1) an *encoder* function, which aims to embed or encode $X_n \subset \mathcal{X}$ into a latent lower-dimensional space $\mathcal{Z}$ such that each observation $x \in X_n$ will be associated with *code* $z \in \mathcal{Z}$, (2) an *decoder* function which maps a code z to a reconstruction in the data space $\tilde{x} \in \mathcal{X}$, and (3) as an objective function we aim to minimize a measure of *reconstruction loss* $\mathcal{L}(x, \tilde{x}) \propto \|x - \tilde{x}\|_\alpha$.
Earlier methods suffer from three significant drawbacks: (1) Strong assumptions on the structure of the data, (2) severe approximations leading to suboptimal models, (3) inference methods with high computational complexity such as Markov Chain Monte Carlo (MCMC). Recent advances in the field of Neural Networks as function approximators open the door to new paradigms which overcome these hurdles [7]:

1. *Generative Adversarial Networks (GANs)* [3]: Pose the training process as a zero-sum game between two separate networks: a generator network and a discriminative network that try to classify samples as either coming from the true distribution $p(x)$ or the model distribution $p_\theta(x)$. If discriminator notices a difference between the two distributions the generator adjusts its parameters slightly, until the generator exactly reproduces the true data distribution and the discriminator is guessing at random, unable to find a difference.

2. *Auto-Regressive Models* (PixelRNN) [11]: Instead train a network that models the conditional distribution of every individual pixel given previous pixels (to the left and to the top). This is similar to plugging the pixels of the image into a char-rnn,

but the RNNs run both horizontally and vertically over the image instead of just a 1D sequence of characters.

3. *Variational Auto-Encoders (VAEs)* [7]: Allow us to formalize this problem in the framework of probabilistic graphical models where we are maximizing a lower bound on the log likelihood of the data. We cast inference as an optimization problem and learn neural networks to perform as inference (encoding) and generator (decoding) mechanisms.

Specifically, Variational Auto-Encoders (VAEs) are a family of generative models for learning latent representations. Our goal is to learn P_θ given a sample in an unsupervised manner. Since VAEs seem particularly appealing from a theoretical perspective, in the following we shall focus our attention on the derivation and interpretation of VAEs to gain a throughout understanding.

2 Prior Art

In [7], D. Kingma and M. Welling originally introduce VAEs as an instantiation of an Auto-Encoding Variational Bayes framework. In [6], D. Kingma and T. Salimans introduce a flexible and computationally scalable method for improving the accuracy of variational inference. In particular, most VAEs have so far been trained using crude approximate posteriors, where every latent variable is independent. Recent extensions [9] have addressed this problem by conditioning each latent variable on the others before it in a chain, but this is computationally inefficient due to the introduced sequential dependencies.

Recent applications of VAEs include (1) spatial and temporal attention mechanisms which can be used to draw images (Google DeepMind DRAW [4]), (2) RNNs as encoder and decoder networks for sketch synthesis (Google Brain sketch-rnn) [5], (3) interpolation between text sentences [1].

Our main contribution is a throughout derivation of VAEs following [7] and a summary of results on bounding the reconstruction error based on [2].

3 Learning Latent Variable Models

Learning and sampling from Latent Variable Models in high-dimensional space is a non-trivial task. We shall define a problem statement which VAEs aim to solve and explore why traditional methods might be insufficient.

3.1 Latent Variable Models

Let $z \in \mathcal{Z}$ denote a latent variable in high-dimensional space $\mathcal{Z}$ where sampling $z \sim p(z)$ is tractable. Say we have a family of functions $f(z; \theta)$ parameterized by $\theta \in \Theta$ where

$f : \mathcal{Z} \times \Theta \to \mathcal{X}$ is deterministic. Note, $f(z; \theta)$ is a random variable in space $\mathcal{X}$ since θ is fixed, but z is random. We use the law ot total probability to denote the dependence of X on z in $f(z; \theta)$ explicitly as $p_\theta(x \mid z)$.

Definition 1 (Latent Variable Model)**.** Let $p_\theta(x, z)$ denote directed latent-variabel model of the form

$$p_\theta(x, z) = p_\theta(x \mid z) p_\theta(z) \tag{1}$$

with observed $x \in \mathcal{X}$ and latent $z \in \mathcal{Z}$.

We can generalize the models to many layers by the chain rule $p_\theta(x \mid z_1) p_\theta(z_2 \mid z_3) \cdots p_\theta(z_{m-1} \mid z_m) p_\theta(z_m)$. These are called deep generative models and can learn a hierchical latent represenent. In this paper, we will assue for simplicity that there is only one such layer. Suppose we are given a sample $X_n = \{x_1, x_2, \ldots x_n\}$. We are interested in three tasks (1) learning the parameters θ of $p_\theta(\cdot)$, (2) approximative posterior inference over z and (3) sampling $x \sim p_\theta(x \mid z)$ given z. Further, we will make the following additional assumptions: (A1) computing the posterior probability $p_\theta(z \mid x)$ is intractable, (A2) the cardinality n of sample X_n is exceeding the memory capacity.

3.2 Objective Function

In a maximum likelihood framework we choose parameters θ such that the model is likely to generate the training set and assume it will likely generate similar samples while it is unlikely to generate dissimlar ones.

Definition 2 (Objective function)**.** Given a training set $X_n = \{x_1, x_2, \ldots x_n\} \subset \mathcal{X}$ the objective is to maximize the log-likelihood of each $x \in X_n$

$$\begin{aligned} \theta^* &= \underset{\theta}{argmax} \log \prod_{i=1}^{n} p_\theta(x_i) \\ &= \underset{\theta}{argmax} \sum_{i=1}^{n} \log p_\theta(x_i) \\ &= \underset{\theta}{argmax} \sum_{i=1}^{n} \log \int p_\theta(x_i \mid z) p_\theta(z) dz. \end{aligned} \tag{2}$$

Note, both computing probability $p_\theta(x \mid z)$ and integration over z are intractable and therefore evidence $p_\theta(x)$ is intractable. We shall later see that in the case of VAEs objective functions is equivalent to minimizing (1) the mean squared reconstruction loss $\mathcal{L}(x, \tilde{x}) = \frac{1}{n} \|x - \tilde{x}\|_2^2$ as stated in the introduction and (2) an additional divergence term from a prior distribution $p(z)$.

3.3 Traditional Methods

There are several traditional techiques that could be used to learn this model which do not rely on Neural Networks. We will briefly describe why these methods might not be suitable:

(1) The *EM-algorithm* can learn latent-variable models. Note that performing the E-step requires the computation of the posterior $p_\theta(x \mid z)$ which by assumption (A1) is intractable. Further, in the M-step we want to determine θ by maximizing the expected value of the log likelihood function which requires to store X_n in memory violating assumption (A2).

(2) The *mean-field* method can be used to perform approximative inference. However, mean field requires the computation of an expectation whose time complexity scales exponentially with the size of the Markov blanket of the target variable. Observe that p will contain a factor $p(x \mid z_1, \ldots, z_k)$ in which all the z variables are tied. This will make mean-field intractable.

(3) The *sampling-based* techniques such as Metropolis-Hastings not only do not scale well in general to large datasets X_n, but also require a hand-crafted proposal distribution, which can be an art in itself.

4 Variational Auto-Encoders

As an introduction to the concept of Auto-Encoders, we will argue that VAEs can be seen as a non-linear extension of PCA. Then we will derive the formulation of VAEs as an instantiation of Auto-Encoding Variational Bayes (AEVB) which turn our three inference and learning tasks into tractable problems [7]. Since the log-likelihood in (2) is not tractable, we will (1) derive a tractable *evidence lower bound* (ELBO) by variational inference which we can maximize to increase $\log p_\theta(x)$, (2) derive a low-variance estimator of the gradient of the ELBO using the *re-parameterization trick*, and (3) parameterize the distributions p_θ and q_ϕ by the means of *neural networks*.

4.1 Auto-Encoder

The PCA algorithm is a old and trusted standard method in Statistics. The PCA method consists of three disjoint steps: (1) Computing the principal components, (2) projecting the data to obtain a low-dimensional representation, and (3) reconstructing the original data set. We show how steps (2) and (3) may fit the definition of a linear auto-encoder.

The term auto-encoder orginates from the domain of neural networks. In this paper, we will treat auto-encoders as a simple composition of an *encoder* $g(\cdot)$ and *decoder* $f(\cdot)$ function such that a reconstruction error is minimized.

Definition 3 (Auto-Encoder)**.** Let $g : \mathcal{X} \to \mathcal{Z}$ and $f : \mathcal{Z} \to \mathcal{X}$ be to functions such that their composition $\tilde{x} = f(g(x))$ with $x \in \mathcal{X}$ minimizes the reconstruction error $\mathcal{L}(x, \tilde{x}) \propto$

$\|\tilde{x} - x\|_\alpha$ over some norm. Then

$$(g, f) = \underset{g,f}{argmin}\, \mathcal{L}(x, (f \circ g)(x)). \tag{3}$$

Specifically in the case of PCA, we can define the reconstruction error $\mathcal{L}(x, \tilde{x}) = \|\tilde{x} - x\|_2^2$ which for the first principal component leads to the optimization problem

$$\omega_1 = \underset{\|w\|_2=1}{argmin}\, \|X - X\omega\omega^T\|_2^2. \tag{4}$$

Assume we keep the full basis of eigenvectos W with full-rank and interpret W as an orthogonal basis rotation. Then we define the *encoder* as $g(X) = XW$ and *decoder* as $f(Z) = ZW^T$. Further, observe that we can now sample from the latent space $\mathcal{Z}$ by probabilistic PCA.

Definition 4 (Probabilistic PCA)**.** Let W be eigenvectors with latent space $z \in \mathcal{Z}$ and variance σ^2

$$\begin{gathered} x = Wz + \epsilon, \\ z \sim \mathcal{N}(0, I),\, \epsilon \sim \mathcal{N}(0, \sigma^2 I), \\ x \mid z \sim \mathcal{N}(Wz, \sigma^2 I). \end{gathered} \tag{5}$$

In the following, we will see that VAEs can be understood as a non-linear extension of PCA.

4.2 Evidence Lower Bound

In the variational family of algorithms, inference is to cast as an optimization problem using variational calculus. Suppose we are given an intractable posterior probability distribution $P_\theta(Z \mid X)$, variational techniques will try to solve an optimization problem over a family of tractable distributions $Q_\phi(Z \mid X)$.

Assume we choose $Q_\phi(Z \mid X)$, say Gaussian family, then we want to adjust the parameters ϕ such that some measure of divergence between $Q_\phi(Z \mid X)$ and $P_\theta(Z \mid X)$ is minimized. Mean-field variational Bayes employs the reverse Kullback-Leibler divergence as such divergence metric between two distributions.

Definition 5 (Reverse Kullback-Leibler Divergence)**.** Let Q and P denote two probability distributions. Define divergence $D_{KL}(Q\|P)$ as

$$D_{KL}(Q\|P) = \int q(z \mid x) \log \frac{q(z \mid x)}{p(z \mid x)} dx. \tag{6}$$

This measure of divergence allows us to formulate variational inference as optimization problem.

Definition 6 (Variational-Inference Optimization Problem)**.** Let $Q_\phi \in \mathcal{F}$ denote a distribution in family $\mathcal{F}$ with variational parameters ϕ and P_θ a target distribution with fixed θ. Then

$$Q_\phi = \underset{\phi}{argmin}\, D_{KL}(Q_\phi \| P_\theta). \tag{7}$$

We substitute the definition of the conditional distribution and distribute

$$\begin{aligned} D_{KL}(Q_\phi \| P_\theta) &= \int q_\phi(z \mid x) \log \frac{q_\phi(z \mid x)}{p_\theta(z \mid x)} dz \\ &= \int q_\phi(z \mid x) \log \frac{q_\phi(z \mid x) p_\theta(x)}{p_\theta(z, x)} dz \\ &= \int q_\phi(z \mid x) \left(\log \frac{q_\phi(z \mid x)}{p_\theta(z, x)} + \log p_\theta(x) \right) dz \\ &= \int q_\phi(z \mid x) \log \frac{q_\phi(z \mid x)}{p_\theta(z, x)} dz \\ &\quad + \log p_\theta(x) \int q_\phi(z \mid x) dz \\ &= \int q_\phi(z \mid x) \log \frac{q_\phi(z \mid x)}{p_\theta(z, x)} dz + \log p_\theta(x). \end{aligned} \tag{8}$$

Observe in order to minimize $D_{KL}(Q_\phi \| P_\theta)$ with respect to ϕ , we only need to minimize $\int q_\phi(z \mid x) \log \frac{q_\phi(z|x)}{p_\theta(z,x)} dz$ since $\log p_\theta(x)$ is fixed under ϕ. We rewrite this quantity as expectation

$$\begin{aligned} &\int q_\phi(z \mid x) \log \frac{q_\phi(z \mid x)}{p_\theta(z, x)} dz \\ &= \mathbb{E}_{z \sim Q_{\phi(Z|X)}} \left[\log \frac{q_\phi(z \mid x)}{p_\theta(z, x)} \right] \\ &= \mathbb{E}_{z \sim Q_{\phi(Z|X)}} \left[\log \frac{q_\phi(z \mid x)}{p_\theta(x \mid z) p_\theta(z)} \right] \\ &= \mathbb{E}_Q \left[\log q_\phi(z \mid x) - \log p_\theta(x \mid z) - \log p_\theta(z) \right]. \end{aligned} \tag{9}$$

We rewrite (8) and substitute (9)

$$\begin{aligned} Q_\phi &= \underset{\phi}{argmin}\, D_{KL}(Q_\phi \| P_\theta) \\ &= \underset{\phi}{argmin} \int q_\phi(z \mid x) \log \frac{q_\phi(z \mid x)}{p_\theta(z, x)} dz \\ &= \underset{\phi}{argmin}\, \mathbb{E}_Q \left[\log q_\phi(z \mid x) - \log p_\theta(x \mid z) - \log p_\theta(z)\right] \\ &= \underset{\phi}{argmax}\, \mathbb{E}_Q \left[-\log q_\phi(z \mid x) + \log p_\theta(x \mid z) + \log p_\theta(z)\right] \\ &= \underset{\phi}{argmax}\, \mathbb{E}_Q \left[\log \frac{p_\theta(z)}{q_\phi(z \mid x)} + \log p_\theta(x \mid z)\right] \\ &= \underset{\phi}{argmax}\, \mathcal{L}(\phi). \end{aligned} \tag{10}$$

Note, if the terms $p_\theta(z)$, $p_\theta(x \mid z)$ and $q_\phi(z \mid x)$ are tractable, then $\mathcal{L}(\phi)$ is comutationally tractable. We can further rewite $\mathcal{L}(\phi)$ into it's commonly used form

$$\begin{aligned} \mathcal{L}(\phi) &= \mathbb{E}_Q \left[\log \frac{p_\theta(z)}{q_\phi(z \mid x)} + \log p_\theta(x \mid z)\right] \\ &= \mathbb{E}_Q \left[\log \frac{p_\theta(z)}{q_\phi(z \mid x)}\right] + \mathbb{E}_Q \left[\log p_\theta(x \mid z)\right] \\ &= \mathbb{E}_Q \left[\log p_\theta(x \mid z)\right] - \mathbb{E}_Q \left[\log \frac{q_\phi(z \mid x)}{p_\theta(z)}\right] \\ &= \mathbb{E}_Q \left[\log p_\theta(x \mid z)\right] - D_{KL}\left(q_\phi(z \mid x) \| p_\theta(z)\right). \end{aligned} \tag{11}$$

In the literature, $\mathcal{L}(\phi)$ is referred to as *variational lower bound*. In fact, the derived expression is a lower bound on evidence $\log p_\theta(x)$ by combining (11) and (8)

$$\log p_\theta(x) - \mathcal{L}(\phi) = D_{KL}(Q_\phi \| P_\theta) \tag{12}$$

which by non-negativity of $D_{KL}(Q_\phi \| P_\theta)$ yields

$$\log p_\theta(x) \leq \mathcal{L}(\phi). \tag{13}$$

Therefore, $\mathcal{L}(\phi)$ is also known as *evidence lower bound* (ELBO).

Lemma 7 (Evidence Lower Bound (ELBO))**.** *The variational lower bound for family $Q_\phi \in \mathcal{F}$ and target P_θ is*

$$\mathcal{L}(\phi) = \mathbb{E}_Q \left[\log p_\theta(x \mid z)\right] - D_{KL}\left(q_\phi(z \mid x) \| p_\theta(z)\right) \geq \log p_\theta(x). \tag{14}$$

Proof.

$$
\begin{aligned}
\log p_\theta(x) &= \log \int p_\theta(x \mid z) p_\theta(z) dz \\
&= \log \int p_\theta(x \mid z) p_\theta(z) \frac{q_\phi(z \mid x)}{q_\phi(z \mid x)} dz \\
&\geq \int q_\phi(z \mid x) \log \left(p_\theta(x \mid z) \frac{p_\theta(z)}{q_\phi(z \mid x)} \right) dz \\
&= \int q_\phi(z \mid x) \log \left(p_\theta(x \mid z) \right) dz \\
&- \int q_\phi(z \mid x) \log \left(\frac{p_\theta(z)}{q_\phi(z \mid x)} \right) dz \\
&= \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log \left(p_\theta(x \mid z) \right) \right] \\
&- \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log \left(\frac{p_\theta(z)}{q_\phi(z \mid x)} \right) \right] \\
&= \mathbb{E}_Q \left[\log \left(p_\theta(x \mid z) \right) \right] - D_{KL} \left(q_\phi(z \mid x) \| p_\theta(z) \right).
\end{aligned}
\tag{15}
$$

In the first equality we use marginalization on z and the fact that z does not depend on $p_\theta(x)$. The second equaility holds trivially. The inequality is obtained by the Jensen inequality. Finally, we rewrite the terms by the definition of the Kullback-Leiber divergence $D_{KL}(\cdot \| \cdot) \geq 0$. ■

Note the formulation of $\mathcal{L}(\phi)$ in (14) gives rise to an intuitive interpretation of the terms where $\mathbb{E}_Q \left[\log p_\theta(x \mid z) \right]$ is the log-likelihood of a data-point x under the true distribution (reconstruction term) and $D_{KL} \left(q_\phi(z \mid x) \| p_\theta(z) \right)$ captures the distance between q_ϕ and p_θ at a specific data-point x (regularization term).

4.3 Black-Box Variational Inference

Recall that in the Variational Inference we aim to maximize the ELBO

$$
\mathcal{L}(\phi, \theta) = \mathbb{E}_Q \left[\log p_\theta(x \mid z) \right] - D_{KL} \left(q_\phi(z \mid x) \| p_\theta(z) \right) \geq \log p_\theta(x) \tag{16}
$$

over the space of all q_ϕ. The ELBO satisfies

$$
\log p_\theta(x) = D_{KL}(q_\phi(z \mid x) \| p_\theta(z \mid x)) + \mathcal{L}(\phi, \theta). \tag{17}
$$

In order to optimize over $q_\phi(z \mid x)$ we maximize the ELBO over ϕ by gradient descent. Therefore, $q_\phi(z \mid x)$ must be differentiable with respect to ϕ. In contrast to traditional Variational Inference, we perform gradient descent jointly over both ϕ and θ. This will have two effects: (1) Optimization over ϕ will ensure that the ELBO stays close to $\log p(x)$, (2) Optimization over θ will increase the lower bound and thus $\log p(x)$.

To perform black-box variational inference, we need to compute the gradient on the ELBO

$$\nabla_{\theta,\phi}\mathbb{E}_Q\left[\log p_\theta(x,z) - \log q_\phi(z)\right]. \tag{18}$$

In most cases we cannot express the expectation with respect to q_ϕ in closed form. Instead, we may draw Monte-Carlo samples from q_ϕ to estimate the gradient by Ergodicity Theorem. For the gradient with respect to p_θ we change the order of gradient and expectation and estimate by Monte-Carlo

$$\mathbb{E}_Q\left[\nabla_\theta \log p_\theta(x,z)\right] \overset{MC}{\approx} \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \log p_\theta(x,z_i). \tag{19}$$

However, for the gradient with respect to q_ϕ we cannot change expectation and gradient, since we are differentiating over the distribution of which we take the expectation

$$\nabla_\phi\mathbb{E}_Q\left[\log q_\phi(z)\right]. \tag{20}$$

Therefore, we will introduce the so-called score function estimator [8].

Lemma 8 (Score function estimator)**.** *Let $p_\theta(x,z)$ and $q_\phi(z)$ denote densities. The score function estimator is*

$$\begin{aligned}&\nabla_\phi\mathbb{E}_Q\left[\log p_\theta(x,z) - \log q_\phi(z)\right]\\ &= \mathbb{E}_Q\left[\left(\log p_\theta(x,z) - \log q_\phi(z)\right)\nabla_\phi \log q_\phi(z)\right].\end{aligned} \tag{21}$$

Proof.

$$\begin{aligned}&\nabla_\phi\mathcal{L}(x) = \nabla_\phi\mathbb{E}_Q\left[\log p_\theta(x,z) - \log q_\phi(z\mid x)\right]\\ &= \nabla_\phi \int q_\phi(z\mid x)\log p_\theta(x,z)dz\\ &- \nabla_\phi \int q_\phi(z\mid x)\log q_\phi(z\mid x)dz\\ &= \int \log p_\theta(x,z)\nabla_\phi q_\phi(z\mid x)dz\\ &- \int (\log q_\phi(z\mid x)+1)\nabla_\phi q_\phi(z\mid x)dz\\ &= \int \left(\log p_\theta(x,z) - \log q_\phi(z\mid x)\right)\nabla_\phi q_\phi(z\mid x)dz\\ &= \int \left(\log p_\theta(x,z) - \log q_\phi(z\mid x)\right) q_\phi(z\mid x)\nabla_\phi(z\mid x)dz\\ &= \mathbb{E}_{\mathbb{Q}}\left[\left(\log p_\theta(x,z) - \log q_\phi(z\mid x)\right)\nabla_\phi(z\mid x)\right].\end{aligned} \tag{22}$$

■

Then identity (21) places the gradient inside the expectation with respect to q_ϕ. We may approximate (18) by Monte-Carlo integration. Unfortunately, the score function estimator suffers from high variance which causes slow convergence to the true expectation. As a remedy [7] proposes an estimator with lower variance which arguable is the main contribution. The estimator is derived in two steps: (1) reformulate the ELBO such that terms of it can be derived as a closed-form solution (without Monte-Carlo integration), (2) the so-called reparameterization trick as an alternative gradient estimator.

4.4 Stochastic-Gradient Variational Bayes

Recall by Lemma 1 the ELBO can be formulated as

$$\log p_\theta(x) \geq \underbrace{\mathbb{E}_Q \left[\log p_\theta(x \mid z)\right]}_{(i)} - \underbrace{D_{KL}\left(q_\phi(z \mid x) \| p_\theta(z)\right)}_{(ii)}. \tag{23}$$

Let $z \sim q_\phi(z \mid x)$ be a sample $z \in \mathcal{Z}$ given a observation $x \in \mathcal{X}$. Then z can be interpreted as a *code* describing x. Hence we may call $q_\phi(z \mid x)$ an *encoder*. Under this regime we inspect the two terms in (23):

In term (i), we want to maximize the log-likelihood $\log p_\theta(x \mid z)$ of observation x given code z. If $p_\theta(x \mid z)$ assigns high probability to observations $x \in X_n$, then the log-likelihood is maximized. We can say $p_\theta(x \mid z)$ aims to reconstruct x given code z and we may call $p_\theta(x \mid z)$ the *decoder*. Then term (i) is referred to as the *reconstruction error*.

In term (ii), we want to minimize the Kullback-Leiber divergence between $q_\phi(z \mid x)$ and prior $p_\theta(z)$. It encourages the codes z to follow a certain distribution (e.g. unit Gaussian). This term is referred to as *regularization term*.

Finally, we can interpret our optimization objective as finding a $q_\phi(z \mid x)$ such that x is encoded into a meaningful latent representation z from which we are able to decode x via $p_\theta(x \mid z)$. In a sense, this resembles traditional *Auto-Encoders*. Therefore, we shall summarize the above derivations as a Bayesian formulation of an auto-encoding process which gives rise to it's name "Auto-Encoding Variational Bayes".

4.5 Reparametrization Trick

As we have seen earlier, maximizing the ELBO requires a good estimate of the gradient. [7] provides a low-variance gradient estimator based on the so-called *reparametrization trick*. Assume we can express $q_\phi(z \mid x)$ as a two-step generative process: (1) Sample noise $\epsilon \sim p(\epsilon)$ from a simple distribution e.g. $\mathcal{N}(0, I)$, (2) apply a deterministic transformation $g_\phi(\epsilon, x)$ that shapes the random noise into an arbitrary distribution $z = g_\phi(\epsilon, x)$.

As an example, we can apply this formulation to Gaussian random variables:

$$\begin{aligned} z &\sim q_{\mu,\sigma}(z) = \mathcal{N}(\mu, \sigma), \\ z &= g_{\mu,\sigma}(\epsilon) = \mu + \epsilon \odot \sigma \end{aligned} \tag{24}$$

where $\odot$ denotes the Hadamard product.

Then we may change the order of expectation and gradient in (20) by applying this reformulation. More generally, for any f it holds

$$\begin{aligned}\nabla_\phi \mathbb{E}_{z\sim q_\phi(z|x)}\left[f(x,z)\right] &= \nabla_\phi \mathbb{E}_{\epsilon\sim p(\epsilon)}\left[f(x, g_\phi(z,\epsilon))\right] \\ &= \mathbb{E}_{\epsilon\sim p(\epsilon)}\left[\nabla_\phi f(x, g_\phi(z,\epsilon))\right] \\ &\overset{MC}{\approx} \frac{1}{N}\sum_{i=1}^{N} \nabla_\phi f(x, g_\phi(z,\epsilon)).\end{aligned} \tag{25}$$

Finally, we may use Monte-Carlo integration to estimate the expectation of the gradient with low variance [10].

4.6 Neural Networks

The AEVB algorithm combines (1) the auto-encoding ELBO reformulation, (2) the black-box variational inference, and (3) the low-variance gradient estimator based on the reparametrization-trick. As discussed earlier, it maximizes the auto-encoding ELBO using black-box variational inference with a reparameterized low-variance gradient estimator. As a constraint of this technique we must choose a model p_θ that is differentiable in θ.

We take both p and q from the Gaussian family and parameterize them by Neural Networks

$$\begin{aligned}p_\theta(z) &= \mathcal{N}(z; 0, I), \\ p_\theta(x \mid z) &= \mathcal{N}(z; \mu(z), \sigma(z) \odot I).\end{aligned} \tag{26}$$

Since we chose p and q as Normal distributions, the Kullback-Leibler divergence of the regularization term in the auto-encoding ELBO can be computed in closed-form. We may estimate the remaining reconstruction term by Monte-Carlo integration.

The structure of the VAE as a end-to-end Neural Network consists of (1) an Inference network $g_\phi : x \to \lambda$ for approximate posterior $q_\phi(z \mid x, \lambda)$ and (2) a generator network $f_\theta : z \to \omega$ for data distribution $p_\theta(x \mid z, \omega)$. Recall, sampling can be performed in two steps: (1) draw samples from prior $z \sim p(z)$ and (2) apply the generator network as deterministic transformation. Note that we can omit the Inference network for this task.

5 Bounds

We will summarize recent theoretical results with respect to VAEs. These consists of (1) a proof of zero reconstruction error with arbitrarily powerful estimators in the one-dimensional case [2].

5.1 Approximation Error in 1D

As in [2], we claim that VAEs obtain a zero approximation error in 1D given arbitrarily powerful learners. Let $P(X)$ denote a 1D distribution which we wish to approximate by a VAE. We assume $P(X) \geq 0$, $\forall X$ and $P(X)$ is infinitely differentiable and all derivatives are bounded. Recall a VAE maximized the ELBO (11) and therefore optimizes

$$\log P_\sigma(X) - D_{KL}\left(Q_\sigma(z \mid X) \| P_\sigma(z \mid X)\right) \tag{27}$$

where $P_\sigma(X \mid z) = \mathcal{N}(X \mid f(z), \sigma^2)$ with $z \sim \mathcal{N}(0,1)$, $P_\sigma(X) = \int P_\sigma(X \mid z)P(z)dz$ and $Q_\sigma(z \mid X) = \mathcal{N}(z \mid \mu_\sigma(X), \Sigma_\sigma(X))$.

Note the theoretical optimal solution is $P_\sigma = P$ and $D_{KL}\left(Q_\sigma(z \mid X) \| P_\sigma(z \mid X)\right) = 0$. The term "arbitrarily powerful learners" refers to the case, that if there exists (f, μ, Σ) which achieve the best possible solution, then the learning algorithm will identify them. Therefore, we only have to show the existence of such solution.

First, claim for $\sigma \to 0$ we can describe $P(X) = \int \mathcal{N}(X \mid f(z), \sigma^2)P(z)dz$ arbitrarily well. Let F denote the CDF of P and let G be the CDF of $\mathcal{N}(0,1)$. Then $G(z) \sim Unif(0,1)$ and thus $f(z) = F^{-1}(G(z)) \sim P(X)$. As $\sigma \to 0$, then distribution $P(X)$ converges to P, which is our desired result.

Second, we must show that $D_{KL}\left(Q_\sigma(z \mid X) \| P_\sigma(z \mid X)\right) \to 0$ as $\sigma \to 0$. Let $g(X) = G^{-1}(F(X))$ and let $Q_\sigma(z \mid X) = \mathcal{N}(z \mid g(X), (g'(X) \cdot \sigma)^2)$. Observe invariance of $D_{KL}\left(Q_\sigma(z \mid X) \| P_\sigma(z \mid X)\right)$ to affine transformations of the sample space. Hence, denote $Q^0(z^0 \mid X) = \mathcal{N}(z^0 \mid g(X), g'(X)^2)$. Then

$$D_{KL}\left(Q_\sigma(z \mid X) \| P_\sigma(z \mid X) = D_{KL}\left(Q^0(z^0 \mid X) \| P^0_\sigma(z^0 \mid X)\right)\right. \tag{28}$$

where $Q^0(z^0 \mid X)$ does not depend on σ. Hence, it is sifficient to show that $P^0_\sigma(z^0 \mid X) \to Q^0(z^0 \mid X)$, $\forall z$. Let $r = g(X) + (z^0 - g(X)) \cdot \sigma$. Then

$$\begin{aligned} P^0_\sigma(z^0 \mid X) &= P_\sigma(z = r \mid X = X) \cdot \sigma \\ &= \frac{P_\sigma(X = X \mid z = r) \cdot P(z = r) \cdot \sigma}{P_\sigma(X = X)}. \end{aligned} \tag{29}$$

Observe, $P_\sigma(X) \to P(X)$ as $\sigma \to 0$, which is a constant. And, $r \to g(X)$ as $\sigma \to 0$ also tends to be a constant. Combine both in constant C. Together (C, σ) ensure that the distribution is normalized.

$$= C \cdot \mathcal{N}(X \mid f(g(X)) + (z^0 - g(X)) \cdot \sigma), \sigma^2). \tag{30}$$

Then apply Taylor expansion of f around $g(X)$ on (30)

$$\begin{aligned} =& C \cdot \mathcal{N}(X \mid X + f'(g(X)) \cdot (z^0 - g(X)) \cdot \sigma \\ &+ \sum_{n=2}^{\infty} \frac{1}{n!} \left(f^{(n)}(g(X))((z^0 - g(X)) \cdot \sigma\right)^n, \sigma^2) \end{aligned} \tag{31}$$

Recall $\mathcal{N}(X \mid \mu, \sigma) = (\sqrt{2\pi\sigma})^{-1} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$. Rewrite (31) as Gaussian

$$\begin{aligned} =&\frac{C \cdot f'(g(X))}{\sigma} \cdot \mathcal{N}(z^0 \mid g(X) \\ &- \sum_{n=2}^{\infty} \frac{\left(f^{(n)}(g(X))((z^0 - g(X)) \cdot \sigma\right)^n}{n! \cdot f'(g(X)) \cdot \sigma}, \\ &f'(g(X))^{-2}). \end{aligned} \tag{32}$$

Note $f = g^{-1}$, that is $f'(g(X))^{-1} = g'(X)$. Further, as stated earlier, since $f^{(n)}$ is bounded for all n, all terms in the sum tend to 0 as $\sigma \to 0$. Recall, C must make the distribution normalize, so we have

$$(32) \to \mathcal{N}(z^0 \mid g(X), g'(X)^2) = Q^0(z^0 \mid X). \tag{33}$$

■

6 Conclusion

We introduced VAEs as a means to train latent variable models in high-dimensional spaces. Following [7], we treat VAEs as an instantiation of the Auto-Encoding Variational Bayes (AEVB) algorithm with neural networks as reparameterization. We may interpret VAEs as a directed latent-variable probabilistic graphical models. We may also view it as a particular objective for training an auto-encoder neural network. Unlike previous techniques, this objective derives reconstruction and regularization terms from a more principled, Bayesian perspective. As in prior work, we can still interpret the Kullback-Leibler divergence term as a regularizer, and the expected likelihood term as a reconstruction loss. But the probabilistic model approach emphasis why these terms exist: to minimize the Kullback-Leibler divergence between the approximate variational posterior $q_\phi(z|x)$ and model posterior $p_\theta(z|x)$.

References

[1] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. *CoRR*, abs/1511.06349, 2015.

[2] Carl Doersch. Tutorial on variational autoencoders, 2016. cite arxiv:1606.05908.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In

Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[4] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1462–1471. JMLR Workshop and Conference Proceedings, 2015.

[5] David Ha and Douglas Eck. A neural representation of sketch drawings. *CoRR*, abs/1704.03477, 2017.

[6] Diederik P. Kingma, Tim Salimans, and Max Welling. Improving variational inference with inverse autoregressive flow. *CoRR*, abs/1606.04934, 2016.

[7] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[8] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. *CoRR*, abs/1402.0030, 2014.

[9] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1530–1538. JMLR Workshop and Conference Proceedings, 2015.

[10] Danilo J. Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1278–1286. JMLR Workshop and Conference Proceedings, 2014.

[11] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *CoRR*, abs/1601.06759, 2016.