# Varying Variational Autoencoders

**Nathan Kong**
University of Toronto
nathan.kong@mail.utoronto.ca

**Jingyao (Jason) Li**
University of Toronto
jingyao.li@mail.utoronto.ca

## Abstract

In variational inference, the approximate posterior distribution that is chosen is very important. It needs to be computationally tractable, yet flexible enough to approximate the true posterior. In this paper, we discuss an application of variational inference in dimensionality reduction. We experiment with the variational autoencoder (VAE), which was developed by Kingma and Welling (2013), by comparing two different variational inference methods. The first method is the vanilla VAE and the second method improves variational inference by introducing normalizing flows, developed by Rezende and Mohamed (2015). Normalizing flows increase the complexity of an initial simple distribution, so that more complex true posteriors can potentially be approximated.

## 1 Introduction

Nowadays, with increasingly large amounts of data, making posterior inferences is intractable since the evidence in Bayes' Rule consists of a computationally intractable integral. Stochastic variational inference was developed that makes this inference more tractable, by turning the inference problem into an optimization problem. In variational inference, an intractable posterior distribution is approximated by a simpler probability distribution, whose parameters are optimized.

Unfortunately, extremely complex posterior distributions may not be successfully approximated using such simple distributions, so novel methods must be developed to improve the approximations. In this paper, we discuss various methods that improve posterior distribution approximations and also compare the performance of two different methods of variational inference.

## 2 Formal Description

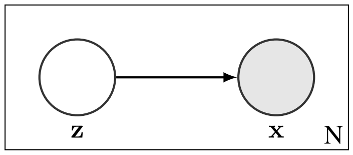

Figure 1: Probabilistic graphical model with latent variables, $\mathbf{z}$, and observed variables, $\mathbf{x}$.

From a probabilistic graphical model perspective, we have a latent space, governed by $\mathbf{z}$, and an observed space, which are our data, $\mathbf{x}$. The observed variables depend on the latent variables. Figure 1 illustrates this model. Using this framework, the joint density is: $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x} \,|\, \mathbf{z})\, p(\mathbf{z})$. $p(\mathbf{z})$ is the prior over the latent variables and $p(\mathbf{x} \,|\, \mathbf{z})$ is the likelihood of the data given the latent variables.

## 2.1 Variational Lower Bound

In order to obtain the marginal likelihood $p(\mathbf{x})$, we must integrate over $\mathbf{z}$, which is intractable. So, a posterior distribution, $q(\mathbf{z}\,|\,\mathbf{x})$, is introduced allowing us to obtain a lower bound on the marginal likelihood:

$$
\begin{aligned}
\log p(\mathbf{x}) &= \log \int_{\mathbf{z}} p(\mathbf{x}\,|\,\mathbf{z})\, p(\mathbf{z})\, \mathrm{d}\mathbf{z} && (1)\\
&= \log \left( \mathbb{E}_q \left[ \frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z}\,|\,\mathbf{x})} \right] \right) && (2)\\
&\geq \mathbb{E}_q \left[ \log p(\mathbf{x},\mathbf{z}) \right] - \mathbb{E}_q \left[ \log q(\mathbf{z}\,|\,\mathbf{x}) \right] && (3)\\
&= \log p(\mathbf{x}) - \mathbb{D}_{\mathrm{KL}} \left[ q(\mathbf{z}\,|\,\mathbf{x}) \,||\, p(\mathbf{z}\,|\,\mathbf{x}) \right] && (4)\\
&= -\mathbb{D}_{\mathrm{KL}} \left[ q(\mathbf{z}\,|\,\mathbf{x}) \,||\, p(\mathbf{z}) \right] + \mathbb{E}_{q(\mathbf{z}\,|\,\mathbf{x})} \left[ \log p(\mathbf{x}\,|\,\mathbf{z}) \right] && (5)
\end{aligned}
$$

Equation 3 follows from Equation 2 by applying Jensen's inequality. Equation 5 is known as the variational lower bound, which we want to maximize. Note that from Equation 4, maximizing the lower bound minimizes the Kullback-Leibler (KL) divergence between the approximate posterior and the true posterior and maximizes the marginal likelihood since the KL divergence is always positive. We want $q(\mathbf{z}\,|\,\mathbf{x})$ to be computationally tractable, but flexible enough to be able to match $p(\mathbf{z}\,|\,\mathbf{x})$ such that the KL divergence is close to 0.

## 2.2 Vanilla VAE

In an autoencoder, there are two main stages: the encoder stage and the decoder stage, which are both neural networks. The input to the encoder, or recognition, stage is the high dimension feature vector that we want to reduce into a space with lower dimensionality. In a VAE, the output of the encoder stage are parameters to the approximate posterior, $q(\mathbf{z}\,|\,\mathbf{x})$. If $q(\mathbf{z}\,|\,\mathbf{x})$ is a Gaussian, the parameters that are output would be the mean, $\boldsymbol{\mu}$, and the variance, $\boldsymbol{\sigma}^2$. In this case, the covariance matrix is a diagonal matrix. So, both $\boldsymbol{\mu} \in \mathbb{R}^D$ and $\boldsymbol{\sigma}^2 \in \mathbb{R}^D$, where $D$ is the dimension of the Gaussian.

The inputs to the decoder, or generator, are sampled values, $\mathbf{z}$, from the distribution, $q(\mathbf{z}\,|\,\mathbf{x})$. For $q(\mathbf{z}\,|\,\mathbf{x}) \sim \mathcal{N}(\mathbf{z}\,|\,\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$, the sampled values are: $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. The output of the decoder stage is a feature vector which has the same dimension as that of the input vector. They are parameters to $p(\mathbf{x}\,|\,\mathbf{z})$.

In order to train the VAE, we want to minimize the negative variational lower bound (negative of Equation 5). It is: $\mathbb{D}_{\mathrm{KL}} \left[ q(\mathbf{z}\,|\,\mathbf{x}) \,||\, p(\mathbf{z}) \right] - \mathbb{E}_{q(\mathbf{z}\,|\,\mathbf{x})} \left[ \log p(\mathbf{x}\,|\,\mathbf{z}) \right]$. The first term measures how different the approximate posterior is from $p(\mathbf{z})$, so that a smaller value indicates a better approximation. The second term is the reconstruction error, which describes how faithful the reconstructed input is to the actual input. The loss function for the VAE is:

$$
\mathcal{F}(\mathbf{x}) = -\frac{1}{2} \sum_{d=1}^{D} \left( 1 + \log \boldsymbol{\sigma}_d^2 - \boldsymbol{\mu}_d^2 - \boldsymbol{\sigma}_d^2 \right) - \left( \sum_{k=1}^{K} \mathbf{x}_k \log \hat{\mathbf{x}}_k + (1 - \mathbf{x}_k) \log (1 - \hat{\mathbf{x}}_k) \right) \qquad (6)
$$

where $K$ is the dimension of the input vector and $\hat{\mathbf{x}}$ being the reconstructed input. This is computed for every sampled value of $\mathbf{z}_i \sim q(\mathbf{z}\,|\,\mathbf{x})$ and then averaged.

## 2.3 VAE with Normalizing Flow

In Rezende and Mohamed (2015), variational inference is improved by introducing normalizing flows. Recall that in variational inference, we want $q(\mathbf{z}\,|\,\mathbf{x})$ to be flexible enough to approximate the true posterior, $p(\mathbf{z}\,|\,\mathbf{x})$, since the true posterior can be extremely complicated. Figure 2 illustrates how inferences are made using normalizing flow and how output is generated.

Normalizing flow describes a series of transformations on the initial probability distribution. These transformations are invertible mappings and at the end of the series, a new probability distribution is obtained. The transformations, $f$, must be chosen such that they are smooth and invertible (i.e. $f^{-1} = g$, where $g(f(\mathbf{z})) = \mathbf{z}$). Also, if $\mathbf{z} \sim q(\mathbf{z})$, then the transformed random variable, $\mathbf{z}_1 = f(\mathbf{z})$, has the following distribution, $q(\mathbf{z}_1)$:

$$
\mathbf{z}_1 \sim q(\mathbf{z}_1) = q(\mathbf{z}) \left| \det \frac{\partial f}{\partial \mathbf{z}} \right|^{-1} \qquad (7)
$$

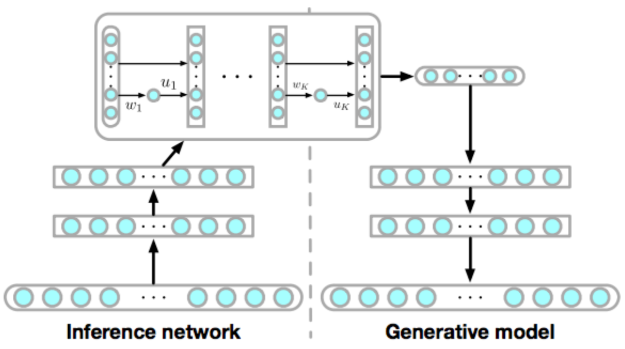

Figure 2: VAE with normalizing flow — flow diagram, from Rezende and Mohamed (2015).

We can perform $K$ transformations to the initial random variable, $\mathbf{z}_0$, to obtain the final random variable, $\mathbf{z}_K$, where:

$$\mathbf{z}_K = f_K(f_{K-1}(\cdots(f_2(f_1(\mathbf{z}_0))))) \tag{8}$$

$$\log q_K(\mathbf{z}_K) = \log q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}} \right| \tag{9}$$

The transformations, $f$, that are used are of the form:

$$f(\mathbf{z}) = \mathbf{z} + \mathbf{u} \cdot h(\mathbf{w}^\top \mathbf{z} + b) \tag{10}$$

where $\mathbf{u}, \mathbf{w} \in \mathbb{R}^D$, $b \in \mathbb{R}$, and $h$ is a smooth non-linearity that is at least once differentiable. This is called a planar flow. We use $h(\cdot) = \tanh(\cdot)$ and $h'(\cdot) = 1 - \tanh^2(\cdot)$ in our experiments. Plugging this into Equation 9, we get:

$$\log q_K(\mathbf{z}_K) = \log q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \log \left| 1 + \mathbf{u}_k^\top \psi_k(\mathbf{z}_{k-1}) \right| \tag{11}$$

where $\psi_k(\mathbf{z}) = h'(\mathbf{w}^\top \mathbf{z} + b)\mathbf{w}$. With these equations, the function we want to minimize becomes:

$$\mathcal{F}(\mathbf{x}) = \mathbb{E}_{q_0}\left[\log q_0(\mathbf{z}_0)\right] - \mathbb{E}_{q_0}\left[\log p(\mathbf{x}, \mathbf{z}_K)\right] - \mathbb{E}_{q_0}\left[\sum_{k=1}^{K} \log \left| 1 + \mathbf{u}_k^\top \psi_k(\mathbf{z}_{k-1}) \right|\right] \tag{12}$$

## 3 Related Work

Rezende and Mohamed (2015) also introduce another type of flow called radial flow. Radial flow differs from planar flow in the transformation function, $f$. The transformation is $f(\mathbf{z}) = \mathbf{z} + \beta \cdot h(\alpha, r)(\mathbf{z} - \mathbf{z}_0)$, where $\alpha \in \mathbb{R}^+$ and $\beta \in \mathbb{R}$. The changes to the initial distribution, $q_0$, differ between the two transformation functions. Planar flows apply contractions and expansions perpendicular to the hyperplane, $\mathbf{w}^\top \mathbf{z} + b$, whereas radial flows apply contractions and expansions radially around $\mathbf{z}_0$.

Inverse autoregressive flow, which is a part of the family of normalizing flows, was introduced by Kingma et al. (2016). In this flow, the series of transformations is initialized by sampling $\mathbf{z}_0 \sim q(\mathbf{z} \,|\, \mathbf{x})$, as before. Then, each successive $\mathbf{z}$ is computed using the transformation: $\mathbf{z}_t = \boldsymbol{\mu}_t + \boldsymbol{\sigma}_t \odot \mathbf{z}_{t-1}$, where $t$ indexes the transformations. After $T$ transformations, the log distribution becomes: $\log q(\mathbf{z}_T \,|\, \mathbf{x}) = -\sum_{i=1}^{D} \left( \frac{1}{2}\epsilon_i^2 + \frac{1}{2}\log(2\pi) + \sum_{t=0}^{T} \log \sigma_{t,i} \right)$ The resulting log marginal probabilities proved to be larger than those computed using previous methods of variational inference.

## 4 Comparison

### 4.1 Baseline (Vanilla) VAE

### 4.2 VAE with Normalizing Flow

## 5 Limitations

## 6 Conclusions

## References

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

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems 29*, pages 4743–4751. 2016.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, pages 1530–1538, 2015.