Notes on Variational Autoencoders

Xavier Garcia

April 24, 2019

In these notes, I want to explain the ideas behind variational autoencoders and their friends. In the first section, we explore the motivations and objects we will be interested in studying through out these notes. In the second section, we will define the variational autoencoder and make some remarks as to why it works. In the third section, we explore possible failure modes that may occur during training, as well as give some quantitative and qualitative approaches to measure the success of training. Finally, in the last section, we present some links to other generative models.

1 Introduction

In these notes, we will have data $x_1, ..., x_N \in \mathbb{R}^n$ viewed as samples of some random variable X. Our initial goal is to able to generate more data according to the distribution of X, but inevitably we'll desire much more. Generally, interesting data follows a complicated distribution which makes it difficult to sample directly from it, even if we had an exact form for it. The principal assumption we'll make is that there exists a latent random variable $Z \in \mathbb{R}^m$ such that both Z and $X|Z = z$ are random variables with simple distributions, which allow for efficient sampling as well as other benefits which we'll explore later on. This assumption is not only convenient, but also quite natural.

For example, if one was to draw a random digit, it would be easier to first choose uniformly at random what digit to draw e.g. 5, then draw some noisy version of 5. In this example, X corresponds to the random variable of possible drawings of digits and the latent variable Z corresponds to a uniform discrete random variable, supported on {0,1,...,9}, which denotes which number we'll draw (not how we draw it!). Guided by this example, an optimist might hope that for any naturally-occuring random variable X, we'll be able find latent variables with obvious meaning. Formally, one can interpret the question "How much meaning is encoded in the latent variables?" to mean "What can we say about distribution $Z|X = x$?" For our example, notice that in this case, $Z|X = x$ is equal to the digit that x is supposed to be a drawing off. In general, we won't be able to assign meaning to Z in such a simple fashion, but the posterior $Z|X = x$ does give us a mathematical object as a proxy for meaning. Putting it all together, the objects we will be interested are the following:

- The conditional distribution $X|Z = z$
- The latent (or prior) distribution Z
- The posterior distribution $Z|X = x$

Given these, we can recover anything we want about X. For example, we can get the distribution of X by

$$p(x) = \mathbb{E}_{z\sim Z}[p(x|z)].$$

Moreover, we can sample from X efficiently by sampling from z from Z, then sampling from $X|Z = z$, without having to compute the potentially-intractable integral $\mathbb{E}_{z\sim Z}[p(x|z)]$.

2 Variational Autoencoders

In this section, we set out to formally define variational autoencoders. We continue with the setting of the previous section and set out to mathematically define the equations we'll need before diving into practical concerns.

2.1 The ELBO Objective

As is tradition, we will assume that $X|Z = z$ and $Z|X = x$ belong to (or are well-approximated by) some parametrized family of distributions. More explicitly,

$$X|Z = z \sim p_\theta(\cdot|z)$$
$$Z|X = x \sim q_\phi(\cdot|x)$$

for some parameters θ and ϕ. We will discuss explicit choices for parametrized family of distributions later on but for the time being, we make no restrictions. Additionally, we'll assume the distribution of Z is known a priori and hence it will not depend on the parameters.

We follow the tried-and-true approach of maximizing the average (log-)likelihood to find the optimal parameters: i.e. maximize the quantity

$$\begin{aligned}\mathbb{E}_{x\sim X} \log p_\theta(x) &= \mathbb{E}_{x\sim X} \log \mathbb{E}_{z\sim Z}[p_\theta(x|z)] \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \log \mathbb{E}_{z\sim Z} p(x_i|z)\end{aligned}$$

There are three problems with this equation. First, the log function is outside the inner expectation, which will make the quantity harder to estimate by Monte Carlo methods (as is done in practice). Second, even without the log, the Monte Carlo estimate may perform poorly with limited samples due to the fact that $p(x|z)$ could be quite small if there is a mismatch between x and z. Consider

the example described in Section 1. If x is a drawing of the number 3 and z is not equal to 3, we should expect the $p(x|z)$ to be 0. Since Z is uniformly distributed in {0,1,...,9} then for 90% of the samples, this estimate will be 0. In particular, this will force us to use a large number of samples, which is highly non-desirable. Third, this equation doesn't involve ϕ at all, which gives us some unused flexibility. We address these three issues as follows.

The first point is addressed by using the concavity of log and Jensen's inequality to move the log inside the expecation, at the cost of the equality becoming an inequality. We deal with the second and third point in a single swipe. Namely, instead of sampling Z using the prior distribution, we should instead use a suitable distribution which gives more importance to relevant samples for a given x, a procedure known as importance sampling. The most natural choice for this would be the distribution of $Z|X = x$ i.e. $q(z|x)$. This motivates the following sequence of operations:

$$\begin{aligned}
\log p_\theta(x) &= \log \mathbb{E}_{z\sim Z}[p_\theta(x|z)] \\
&= \log \mathbb{E}_{z\sim q_\phi(z|x)}\left[p_\theta(x|z)\frac{p(z)}{q_\phi(z|x)}\right] \\
&\geq \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left(p_\theta(x|z)\frac{p(z)}{q_\phi(z|x)}\right)\right] \\
&= \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\frac{p_\phi(z|x)}{p(z)}\right] \\
&= \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - \mathrm{KL}(q_\phi(z|x)||p(z))
\end{aligned}$$

where KL refers to the Kullback-Liebler divergence. After computing the expectation of both sides with respect to $x \sim X$, we call the resulting quantity on the right the Expected Lower Bound or **ELBO**. We can summarize the algebraic manipulations by the following statement: "We are allowed to move the log inside at by trading equality for inequality. We can swap the prior distribution with the posterior by incurring a cost given by the Kullback-Liebler divergence between the two." Given that most of optimization theory is phrased in terms of minimizing objectives, we re-write this in terms of the average negative log-likelihood:

$$\begin{aligned}
-\mathbb{E}_{x\sim X}\log p_\theta(x) &\leq \mathbb{E}_{x\sim X}\left[\mathbb{E}_{z\sim q_\phi(z|x)}[-\log p_\theta(x|z)] + \mathrm{KL}(q_\phi(z|x)||p(z))\right] \\
&\approx \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{z\sim q_\phi(z|x_i)}[-\log p_\theta(x|z)] + \mathrm{KL}(q_\phi(z|x_i)||p(z))
\end{aligned}$$

The first term on the right is traditionally referred to as the **reconstruction loss**. This is the natural objective we should be trying to minimize. The second term is usually considered to be regularization and we'll justify its existence through more palatable means than seemingly-magical algebraic manipulations.

2.2 The canonical example: The Gaussian case

To convince the reader that this framework we established can actually be utilized in practice, we present a fundamental example. In this case, we'll assume that all the distribution we are interested in are diagonal Gaussians. More explicitly, we assume the following to be true:

$$\begin{aligned} X|Z = z &\sim \mathcal{N}\left(f_\theta(z), \frac{1}{2} I_n\right) \\ Z|X = x &\sim \mathcal{N}(\mu_\phi(x), \operatorname{diag}(\sigma_\phi(x)) \\ Z &\sim \mathcal{N}(0, I_m) \end{aligned}$$

where μ_ϕ, σ_ϕ and f_θ are some parametrized family of functions e.g. neural networks, I_m corresponds to the $m \times m$ identity matrix, and $\operatorname{diag}(\sigma_\phi(x))$ refers to the diagonal matrix whose entries are given by the vector $\sigma_\phi(x)$. The first of these assumptions say that given z, X is given by $f_\theta(z)$ plus some Gaussian noise. The second one tell us that given x, the most relevant z values will be clustered around some $\mu_\phi(x)$. The last one says that with no fixed x, z follows standard Gaussian. In this case, we can write the reconstruction loss as follows:

$$\begin{aligned} \mathbb{E}_{z\sim q_\phi(z|x)}[-\log p(x|z)] &= \mathbb{E}_{z\sim q_\phi(x|z)}(x - f_\theta(z)^2 + \frac{n}{2}\log\pi \\ &= \mathbb{E}_{z\sim\mathcal{N}(0,I)}[(x - f_\theta(\mu_\phi(x) + \sigma_\phi(x)z)^2] + \frac{n}{2}\log\pi \end{aligned}$$

This says the reconstruction loss is actually the square error between the reconstruction $f(\mu_\phi(x) + \sigma_\phi(x)z)$ and the original input x. Moreover, the second equality can easily be approximated by Monte Carlo methods and is usually done so with a single sample. More surprisingly, we can also explicitly compute the KL term as follows:

$$\mathrm{KL}(\mathcal{N}(\mu_\phi(x), \sigma_\phi(x)), \mathcal{N}(0, I)) = \frac{1}{2}\left(||\mu_\phi(x)||^2 + ||\sigma_\phi(x)||^2 - ||\log\sigma_\phi^2(x)||^2 - m\right)$$

where $||\cdot||$ corresponds to the traditional Euclidean norm. In particular, every term in the ELBO objective has an explicit form which amenable to gradient descent.

2.3 The Reparametrization Trick

The most important attributes of our assumptions in the previous example are the following:

1. An analytic expression for the KL term
2. An expression for the reconstruction term as an expectation of a distribution which did not depend on the parameters.

The reason why these two factors are so important is that both terms are originally written as expectations against $q_\phi(z|x)$. In practice, we replace these expectation by their Monte Carlo approximations, which would make it impossible

to differentiate with respect to the parameters. By changing the distribution that we are expectating against, we managed to subvert this problem. More explicitly, we used the fact we could write the posterior distribution $Z|X = x$ as a function (depending on the parameters) of a random variable whose distribution did not depend on the parameters:

$$Z|X = x \sim h(Z\,|x, \phi) := \sigma_\phi(x)\tilde{Z} + \mu_\phi(x)$$

where $\tilde{Z} \sim \mathcal{N}(0, I)$ does not depend on ϕ, θ or x. We call this the **reparametrization trick**. Notice that since the KL term is also an expectation with respect to the posterior distribution, we could also substitute attribute 1 with the requirement that the KL term can be written as an expression which is amenable to the reparametrization trick.

3 Failure Modes

In this section, we explore potential failures that could arise during the training of VAEs. As before, recall that we can write the (negative) ELBO objective as:

$$-\mathbf{ELBO}(x) = \mathbb{E}_{x\sim X}\left[\mathbb{E}_{z\sim q(z|x)}[-\log p(x|z)] + \mathrm{KL}(q(z|x)||p(z))\right].$$

Earlier, we referred to the first term on the right as the reconstruction term and the second one as regularization. We attempt to demystify how the KL term applies regularization by viewing it as a measure of the amount of information encoded into the latent space by conditioning on a sample x from X. The more $q(z|x)$ and $p(z)$ differ, the more influence x has on $q(z|x)$ and hence the conditional distribution must be capturing some properties of x. In the next two subsections, we examine the two extreme cases for this quantity.

3.1 Information overload: $\mathbb{E}_{x\sim X}\mathbf{KL}(q(z|x)||p(z)) = \infty$

The first failure mode comes from what we'll call information overload i.e.

$$\mathrm{KL}(q(z|x)||p(z)) = \infty.$$

In this case, we've endowed too much information on the posterior $q(z|x)$. An infinite KL divergence can arise from the posterior distributions being supported on $p(z)$-measure-zero set and hence almost all (with respect to the prior distribution) latent vectors don't correspond to any meaningful code. In particular, if we tried to sample z from Z then sample $X|Z = z$, we'd most likely obtain something far different from the data. As an example, let's return to the Gaussian case of Section 2.2. Suppose that in that case, we chose a degenerate Gaussian where $\sigma_\phi(x) = 0$ for all x i.e. $q(z|x)$ is the Dirac delta distribution $\delta_{\mu_\phi(x)}$. By the KL formula from before, this implies that the KL is infinite. Informally, it says that there is no flexibility in the latent space. Any minor modification in the latent space would yield a point that could be drastically

different from the original input, causing a type of discontinuity of the latent space. We can see this more directly by noticing the following:

$$p(z) = \mathbb{E}_{x\sim X}[p(z|x)] \approx \frac{1}{N}\sum_{i=1}^{N} p(z|x_i) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\mu_\phi(x_i)}.$$

In particular, this says that the distribution of the latent space is roughly supported on a finite set of points and we would require many samples to be able to roughly get a good approximation of the real distribution. This failure mode tends not to happen too often, but it can arise as a by-product of solving the next failure mode.

3.2 Posterior collapse: $\mathbf{KL}(q(z|x)||p(z)) = 0$

The other extreme is the case when the KL term drops to 0. At a glance, this isn't so bad. After all, we should expect that both the KL term and the reconstruction term to become very small assuming training goes well. Morally, however, this seems quite corrupt. If the KL term should measure the amount of information encoded into the latent space by the posterior, then it seems impossible for the reconstruction term to drop to 0 as well. Nevertheless, it is sometimes possible for flexbile enough decoders to approximate the distribution of $X|Z = z$ without actually using z. If this is the case, then then the VAE will then focus on minimizing the KL loss by reducing the posterior to the prior i.e. setting $q(z|x) = p(z)$ for all x, hence the name **posterior collapse**.

If the goal is simply to generate new data, one might argue that this isn't really a failure mode. After all, if by some magic the decoder can construct $X|Z = z$ without using z, then it can recover the distribution of X and thus we can sample. Nevertheless, any proponent of magic in the 21st century should be met with skepticism. One should expect that if the decoder can magically reproduce the distribution of $X|Z = z$ without using z it must mean that it is either cheating or casting an illusion. To give an example of the latter, suppose that $X \sim \mathcal{N}(0, \epsilon)$ for small ϵ and let the decoder $f_\theta(z) = 0$ for all z. The reconstruction error is approximately the variance of X and hence ϵ. Moreover, since f_ϕ doesn't actually use the latent code, we can freely set the posterior equal to the prior and hence collapse the KL term to 0. However, this is not a satisfying approximation. We are exploiting the fact that X is very well approximated by the constant 0 and nothing more, hence the illusion.

An example of cheating is a little more subtle. If the decoder is truly reproducing $X|Z = z$ without using z explicitly, then it must be gaining the information that it would gain from z through some other illegimate sources hence cheating. The most common situation where this occurs is when trying to model sequential data. In this setting, a sample point $\mathbf{x} = (x_1, ..., x_T)$ consists of some sequence of points in space and the decoder is constructed in an

autoregressive way. More precisely, we model the distribution of $\mathbf{X}|Z = z$ by

$$p(x_1, ..., x_T|z) = p(x_1|z) \prod_{t=1}^{T-1} p(x_{t+1}|x_1, ..., x_t, z)$$

and model the density $p(x_{t+1}|x_t, ..., x_1, z)$ by using the hidden states of a recurrent decoder f_ϕ, evaluated at the point x_{t+1}. The hidden state at time t will capture the information regarding z and the rest of the sequence $x_1, ..., x_t$. During training, we pass $\mathbf{x}$ to both the encoder and decoder. At inference, however, the decoder constructs the output in autoregressive fashion i.e. it uses its own predictions rather than the original inputs. This is necessarily so since at inference, we are generating new data and therefore don't have future data. This mismatch between training and inference is what allows the cheating occur. Namely, the decoder can learn to always expect meaningful data to come in as input, hence it can ignore the latent code and simply use the input, devolving the decoder into something akin to a language model.

The important question is, are these phenomena of practical concern? The answer is yes, and in multiple ways. It has been demonstrated by empirical experiments that early on in training, the KL can term can actually grow to a large value, which will cause the optimizer to fret and instantly push to minimize it,causing the collapse from which the optimizer cannot recover from, see e.g. [1]. There are many purported explanations for this initial growth, but we don't delve into those here. Moreover, it's much easier for the decoder to collapse the posterior than it is to actually lower the reconstruction error, since to collapse the posterior, it only needs to set $\mu_\phi = 0$ and $\sigma_\phi = 1$, which makes it more likely for the optimizer to find the collapsing posterior solution before a reasonable minima for the ELBO.

It is thus recommended to monitor both the KL term and the reconstruction loss independently to assess whether collapse has occured. In the sequential data case, one should also monitor validation loss with and without teacher-forcing to make sure information leaking is not occuring.

3.3 Qualitative metrics

It can be difficult to tell if the VAE is experiencing posterior collapse or if the problem doesn't require that much information encoded in the latent space. Similarly, it may be possible that one requires very large KL values and is not actually experiencing information overload. In order to assess if one has entered one of the two failure modes, we should consider additional qualitative metrics to establish whether the VAE has actually learned something of interest. In particular, we will focus on whether the VAE utilizes the latent space in an interesting manner.

There are many ways to formally pose this question. The most obvious way would be to consider the question, "Does (almost-)every point in latent space decode back to a meaningful point?" We can test this by doing sampling points from Z and decoding them back into data points and see if the output seems

like it could arise as a sample from X. Depending on the setting, one could potentially eyeball this e.g. images. A more rigorous approach might be to look at the quantity $p(x|z)$ where x is the decoded point and see if the value seems reasonable.

A more interesting approach is to exploit the fact that latent space should be continuous. We saw earlier that in both information overload and posterior collapse, some notions of continuity were broken. In some sense, the continuity of the latent space is one of the most interesting features of VAE. Even if the data is discrete or does not support natural operations such as addition, the latent structure created by the VAE is naturally a vector space. To that end, one could always check whether interpolations make sense. More rigorously, given two data points x_0 and x_1, we can encode them into the latent space to produce z_0 and z_1 respectively then define a path $\varphi : [0, 1] \to \mathbb{R}^m$ where $\varphi(0) = z_0$ and $\varphi(1) = z_1$. We should then check that for every t, the latent vector $\varphi(t)$ naturally decodes back to a meaningful data point and the path $\{\varphi(t)\}_t$ serves as a smooth interpolation between the two data points x_0 and x_1. The most natural choice of a path would be linear interpolation, $\varphi(t) = x_0+t(x_1-x_0)$. However, depending on the prior distribution on Z, this may be ill-advised. For example, it is known that m-dimensional Gaussian distributions concentrate heavily around the sphere of radius $\sqrt(m)$ for large m. Linear interpolation could pass through a vector with low norm which would correspond to an unlikely vector to be sampled from the distribution. As a way to address, some have used spherical linear interpolation (slerp) instead, see e.g. [6].

One can similarly exploit the vector space properties of the latent space. The idea here is to grab all data points with a shared property in common and compute the average of the latent code to produce the **attribute vector**. One way to check the quality of the model is by taking an item without the attribute, computing its latent vector, adding the attribute vector to the latent code, then check if the item obtained by decoding this new vector possess the attribute.

4 Relationships with other models

In this section, we discuss other generative models and how they relate with the VAE architecture. We begin by deriving a generalization of the ELBO and use it to construct various different models in the literature.

4.1 Maximizing mutual information

One might object in using the MLE objective since it doesn't directly involve the latent space and it reduces to an inequality. A natural objective to consider then is to maximize the **mutual information** $I_{\theta,\phi}(X, Z)$ between the random variables X and Z i.e.

$$I(X, Z) = \mathbb{E}_{z\sim Z, x\sim X}\left[\log \frac{p_{X,Z}(x, z)}{p_X(x)p_Z(z)}\right]$$

where p_X and p_Z denote the densities of X and Z respectively and $p_{X,Z}$ is the joint distribution. By writing $p_{X,Z}(x,z) \approx p_Z(z)p_\theta(x|z)$, we can write the mutual information as follows:

$$\begin{aligned} I_{\theta,\phi}(X,Z) &= \mathbb{E}_{z\sim Z, x\sim X}[\log p_\theta(x|z)] - \mathbb{E}_{z\sim Z, x\sim X}[\log p_X(x)] \\ &= \mathbb{E}_{x\sim X}\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] + \mathcal{H}(X) \end{aligned}$$

where $\mathcal{H}(X)$ is the entropy of X. This term only depends on the distribution of the data and thus its independent of the network. Therefore,

$$\begin{aligned} \text{argmax}_{\theta,\phi} I_{\theta,\phi}(X,Z) &= \text{argmax}_{\theta,\phi}\mathbb{E}_{x\sim X}\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] \\ &= \text{argmin}_{\theta,\phi}\mathbb{E}_{x\sim X}\mathbb{E}_{z\sim q(z|x)}[-\log p_\theta(x|z)] \end{aligned}$$

This is the loss function for **Uncertainty Autoencoders** (UAE), see the work of Grover and Ermon [2]. A closer inspection reveals this to be the reconstruction loss from VAE.

4.2 β-VAE

Given the connection we established between the UAE's loss function and the VAE's loss function, we might become concerned by the missing KL term. Notice that its absence agrees with the previously stated intuition regarding the KL term and encoding information. It makes sense that in order to maximize mutual information, one should not penalize encoding information into the posterior distribution. Nevertheless, to prevent an information overload as before, one could impose a bound on the KL divergence. More explicitly, we could replace the mutual information objective with the following restricted optimization problem:

$$\begin{aligned} &\min_{\theta,\phi} \mathbb{E}_{x\sim X}\mathbb{E}_{z\sim q(z|x)}[-\log p_\theta(x|z)] \\ &\text{such that } \mathbb{E}_{x\sim X}\text{KL}(q_\phi(z|x)||p(z)) \leq \lambda \end{aligned}$$

In this problem, we've explicitly limited the size of the KL term to avoid explosions. With sufficient regularity, the KKT conditions allow us to write this optimization problem as the following:

$$\min_{\theta,\phi} \mathbb{E}_{x\sim X}\mathbb{E}_{z\sim q(z|x)}[-\log p_\theta(x|z)] + \beta\text{KL}(q_\phi(z|x)||p(z))$$

for some β related to λ. This final objective is the **β-VAE** objective, see [3]. The authors originally considered the base $\beta > 1$ to increase the regularization power, but nowadays most work consider either $\beta < 1$ or an annealing schedule to increase β from 0 until some desired value.

4.3 Free Bits

Interestingly enough, Kingma et. al. [4] considered the opposite problem:

$$\begin{aligned} &\text{argmin}_{\theta,\phi}\mathbb{E}_{x\sim X}\mathbb{E}_{z\sim q(z|x)}[-\log p_\theta(x|z)] \\ &\text{such that } \mathbb{E}_{x\sim X}\text{KL}(q_\phi(z|x)||p(z)) \geq \lambda \end{aligned}$$

At the time, they were trying to avoid the posterior collapse by requiring the VAE to have non-trivial KL. One can equivalently formulate the problem in its dual form as follows:

$$\text{argmin}_{\theta,\phi}\mathbb{E}_{x\sim X}\mathbb{E}_{z\sim q(z|x)}[-\log p_\theta(x|z)] + max(KL(q(z|x)||p(z) - \lambda, 0)$$

This is this the so-called free bits formulation. The intuition is that VAE has a budget of λ bits (if using base 2) to invest in encoding information and will get penalized if it uses more. This is an effective way of prevent posterior collapse, but it presents optimization challenges due to the non-smoothness of the penalty.

4.4 f-VAE

In Section 2.3, we highlighted the power of the reparametrization trick. Given this success, one could directly choose posterior and conditional distributions which are amenable to this trick. More precisely, we can choose a function $F : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$ and define the following:

$$Z|X = x \sim F(\tilde{Z}, x)$$

where $\tilde{Z} \sim \mathcal{N}(0, I_m)$. This framework is known as **f-VAE**, see [5]. Notice that choosing $F(z, x) = \mu_\phi(x) + \sigma_\phi(x)z$ recovers the usual VAE posterior. Moreover, a judicious choice of F can also introduce a version of **normalizing flows**.

References

[1] Bowman, S., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., Bengio, S. "Generating Sentences from a Continuous Space." *SIGNLL Conference on Computational Natural Language Learning (CONLL).* 2016

[2] Grover, A., Ermon , S. "Uncertainty Autoencoders: Learning Compressed Representations via Variational Information Maximization" *https://arxiv.org/pdf/1812.10539.pdf.* 2019

[3] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A. "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework." *https://openreview.net/pdf?id=Sy2fzU9gl*, 2017

[4] Kingma, D., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., Welling, M. "Improved Variational Inference with Inverse Autoregressive Flow." *https://arxiv.org/abs/1606.04934* 2016

[5] Su, J., Wu, G. "f-VAEs: Improve VAEs with Conditional Flows." *arXiv:1809.05861 2018.*

[6] White, T. "Sampling Generative Networks: Notes on a few effective techniques." *preprint https://arxiv.org/pdf/1609.04468.pdf.* 2016