Variational Auto Encoder*

Fahim Faisal Niloy

1 Introduction

In this document, we will address two fundamental problems in machine learning,

- Density estimation: Modeling $p(x)$. This distribution helps us to quantify how probable a new data point is.
- Classification: Modeling the distribution $p(y|x)$, which gives us the class distribution given the input data distribution

At first, we will look into the first problem, that is modeling the data generating process.

2 Density Estimation

$p_\theta(x)$ is comparatively easier to model if all the variables of the system are observed in data. However, if there are latent variables associated in the system, modeling $p(x)$ is difficult because the latent variables have to be taken into account while modeling $p_\theta(x)$ with the following equation.

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

The distribution $p_\theta(x, z)$ can be assumed by making suitable choices for $p_\theta(z)$ and $p_\theta(x|z)$. Because,

$$p_\theta(x, z) = p_\theta(x|z)p_\theta(z)$$

So $p_\theta(x, z)$ is easy to calculate for any given x or z. The intractability of $p_\theta(x)$ stems from the integration operation in (1). Because, the distribution $p_\theta(x|z)$ is usually characterized with neural networks, which makes the integration operation in (1) intractable. Even if we assume the distributions $p_\theta(x|z)$ and $p_\theta(z)$ with known probability distributions, only in the case of conjugate pairs this integration is tractable. So, in most of the cases it is not possible to find an analytical solution to the integration. Also,

*This is a draft version. The document hasn't been completed yet

$$p_\theta(x,z) = p_\theta(z|x)p_\theta(x)$$
$$p_\theta(z|x) = \frac{p_\theta(x,z)}{p_\theta(x)}$$

So, intractable $p_\theta(x)$ makes $p_\theta(z|x)$ intractable. Estimating $p_\theta(z|x)$ using variational distribution $q_\phi(z|x)$ is one solution. **Our sole objective is to estimate** $p_\theta(x)$. If $q_\phi(z|x)$ is a close estimate of $p_\theta(z|x)$ we can estimate $p_\theta(x)$ by

$$p_\theta(x) \sim \frac{p_\theta(x,z)}{q_\phi(x|z)}$$

In fact [3] uses a similar method (importance sampling).

$$log\, p_\theta(x) = log\, E_{q_\phi(z|x)}[\frac{p(x,z)}{q(x|z)}]$$

There are several methods to variational inference (approximating the true posterior with $q(z|x)$).

- Comparatively older literature used to perform classical methods (MCMC, Mean field, conjugate priors etc.) to approximate $p(z|x)$ with $q(z|x)$.
- At around 2014 stochastic variational inference emerged where stochastic gradient descent is used by performing reparamitarization trick, which makes calculating gradients with respect to variational parameters possible.
- Later at 2016 came normalization flow [1] based variational inference.
- Recently adversarial techniques to estimate the posterior are becoming popular.

Now getting back to our objective. We want to make the approximate posterior $q_\phi(z|x)$ close to the true posterior $p_\theta(z|x)$. Essentially, we want to minimize the divergence between the two distributions.

$$\begin{aligned}
D_{KL}\left[q_\phi(z|x) \,||\, p_\theta(z|x)\right] &= E_{q_\phi(z|x)}[log[\frac{q_\phi(z|x)}{p_\theta(z|x)}]] \\
&= E_{q_\phi(z|x)}[log[\frac{q_\phi(z|x)p_\theta(x)}{p_\theta(z|x)p_\theta(x)}]] \\
&= E_{q_\phi(z|x)}[log[\frac{q_\phi(z|x)p_\theta(x)}{p_\theta(x,z)}]] \\
&= E_{q_\phi(z|x)}[-log[\frac{p_\theta(x,z)}{q_\phi(z|x)p_\theta(x)}]] \\
&= E_{q_\phi(z|x)} log\, p_\theta(x) - E_{q_\phi(z|x)}[log[\frac{p_\theta(x,z)}{q_\phi(z|x)}]] \\
&= log\, p_\theta(x) - \underbrace{E_{q_\phi(z|x)}[log[p_{\theta(x,z)} - logq_\phi(z|x)]}_{\text{ELBO}} \qquad (2)
\end{aligned}$$

The last term is always positive. So it acts as a lower bound. It is named as *Variational Lower Bound* or *Evidence Lower Bound*, in short **ELBO** [2]. We could also come at the ELBO objective from the log marginal distribution:

$$
\begin{aligned}
log\, p_\theta(x) &= E_{q_\phi(z|x)}[log\, p_\theta(x)] \\
&= E_{q_\phi(z|x)}[log[\frac{p_\theta(x,z)}{p_\theta(z|x)}]] \\
&= E_{q_\phi(z|x)}[log[\frac{p_\theta(x,z)q_\phi(z|x)}{p_\theta(z|x)q_\phi(z|x)}]] \\
&= E_{q_\phi(z|x)}[log[\frac{p_\theta(x,z)}{q_\phi(z|x)}]] + E_{q_\phi(z|x)}[log[\frac{q_\phi(z|x)}{p_\theta(z|x)}]] \\
&= \underbrace{E_{q_\phi(z|x)}[log[p_\theta(x,z) - logq_\phi(z|x)]}_{\text{ELBO}} + D_{KL}\,[q_\phi(z|x)\,||\,p_\theta(z|x)]
\end{aligned}
$$

Maximizing ELBO with respect to θ and ϕ maximises the probability of observing the data. So, the ELBO objective $L_{\theta,\phi}(x)$ is:

$$
L_{\theta,\phi}(x) = E_{q_\phi(z|x)}[log\, p_\theta(x,z) - log\, q_\phi(z|x)] \tag{3}
$$

There are different modifications to the ELBO objective in literature. All of them can be derived from (3). One common (probably more prevalent) modification is:

$$
\begin{aligned}
L_{\theta,\phi}(x) &= E_{q_\phi(z|x)}[log\, p_\theta(x,z) - log\, q_\phi(z|x)] \\
&= E_{q_\phi(z|x)}[log\,[p_\theta(x|z)p_\theta(z)] - log\, q_\phi(z|x)] \\
&= E_{q_\phi(z|x)}[log\, p_\theta(x|z) + log\, p_\theta(z) - log\, q_\phi(z|x)] \\
&= E_{q_\phi(z|x)}[log\, p_\theta(x|z) + log\, \frac{p_\theta(z)}{q_\phi(z|x)}] \\
&= E_{q_\phi(z|x)}[log\, p_\theta(x|z) - log\, \frac{q_\phi(z|x)}{p_\theta(z)}] \\
&= E_{q_\phi(z|x)} log\, p_\theta(x|z) - E_{q_\phi(z|x)} log\, \frac{q_\phi(z|x)}{p_\theta(z)} \\
&= E_{q_\phi(z|x)} log\, p_\theta(x|z) - D_{KL}\,(q_\phi(z|x)\,||\,p_\theta(z))
\end{aligned} \tag{4}
$$

Here, $q_\phi(z|x)$ is the encoder, $p_\theta(z)$ is prior and $p_\theta(x|z)$ is the decoder. The encoder and decoder parameters ϕ and θ are usually learned with deep neural networks.

Our objective is to maximize the lower bound with respect to ϕ and θ. It serves two purposes. From (2) it can be observed,

$$
ELBO = log\, p_\theta(x) - \; D_{KL}\,[q_\phi(z|x)\,||\,p_\theta(z|x)]
$$

$ELBO$ is maximized if $log\ p_\theta(x)$ is maximized and $D_{KL}\ [q_\phi(z|x)\ ||\ p_\theta(z|x)]$ becomes 0. So maximizing the $ELBO$ ensures that the probability of observing the data is maximized and simultaneously the variational posterior distribution becomes close to the true posterior distribution.

3 Maximizing ELBO: The Hardships

Now, the optimization should be straightforward, that is, we should maximise $L_{\theta,\phi}(x)$ with respect to both θ and ϕ. Let's see the gradient calculation with respect to theta. But before that, it should be noted that if gradient operation is interchangeable with expectation operation, then the calculation of gradients becomes easy. That is,

$$\nabla_\theta E_{p(x)}[f(x)] = E_{p(x)}[\nabla_\theta f(x)]$$

Because, practically we approximate the expectation most of the time. That is the approximation of the expectation will give us a single value. We can not take gradient of that single value. Let's check if we can interchange the gradient and expectation operation for ELBO. We first take gradient with respect to θ in Eqn (3)

$$\begin{aligned}
\nabla_\theta L_{\theta,\phi}(x) &= \nabla_\theta E_{q_\phi(z|x)}[log\ p_\theta(x,z) - log\ q_\phi(z|x)] \\
&= \nabla_\theta \int log\ p_\theta(x,z) q_\phi(z|x) dz - \nabla_\theta \int log\ q_\phi(z|x) q_\phi(z|x) dz \\
&= \int \nabla_\theta\ log\ p_\theta(x,z) q_\phi(z|x) dz - \int \nabla_\theta\ log\ q_\phi(z|x) q_\phi(z|x) dz \\
&= \int \nabla_\theta\ log\ p_\theta(x,z) q_\phi(z|x) dz - 0 \\
&= \int q_\phi(z|x)\ \nabla_\theta log\ p_\theta(x,z) dz \\
&= E_{q_\phi(z|x)}[\nabla_\theta\ log\ p_\theta(x,z)]
\end{aligned}$$

So, we can interchange the expectation sign with the gradient. Now, the gradient calculation is straightforward, that is, calculate the gradient, and sample from $q_\phi(z|x)$ to calculate the monte-carlo expectation.

However, gradient calculation with respect to ϕ is problematic. Because we can not interchange the expectation sign with gradient in this case.

$$\begin{aligned}
\nabla_\phi L_{\theta,\phi}(x) &= \nabla_\phi E_{q_\phi(z|x)}[log\ p_\theta(x,z) - log\ q_\phi(z|x)] \\
&= \nabla_\phi \int log\ p_\theta(x,z) q_\phi(z|x) dz - \nabla_\phi \int log\ q_\phi(z|x) q_\phi(z|x) dz \\
&= \int log\ p_\theta(x,z) \nabla_\phi q_\phi(z|x) dz - \int \nabla_\phi log\ q_\phi(z|x) q_\phi(z|x) dz \\
&\neq E_{q_\phi(z|x)}[\nabla_\phi log\ p_\theta(x,z)] - E_{q_\phi(z|x)}[\nabla_\phi log\ q_\phi(z|x)]
\end{aligned}$$

We can get away with this by calculating the score function. However, the score function gradient estimate shows high variance.

The main obstacle during the calculation is that the gradient is being taken with respect to the distribution parameter. So, **the distribution term can not be taken out from the gradient operation to later show as expectation**. To solve this problem, we can take resort to change of variable technique and change the distribution variable. This way, the distribution can be taken out from gradient operation.

4 Reparameterization aka Change of Variable

We know, from probability theory, if there is some function of random variable $y = g(x)$, and y and x have the distribution $p(y)$ and $p(x)$ respectively, then,

$$\int y\, p(y) dy = \int g(x)\, p(x) dx$$
$$E_{p_y}[y] = E_{p_x}[g(x)]$$

Which is intuitive and can be proved with change of variable technique. However, for a rigorous proof check appendix.

Let,

$$z = g(\epsilon, \phi, x)$$

Now, the expectation is,

$$E_{q_\phi(z|x)}[log\, p_\theta(x,z) - log\, q_\phi(z|x)] = E_{p(\epsilon)}[log\, p_\theta(x, g(\epsilon,\phi,x)) - log\, q_\phi(g(\epsilon,\phi,x)|x)]$$

So, our changed ELBO is now,

$$L_{\theta,\phi}(x) = E_{p(\epsilon)}[log\, p_\theta(x, g(\epsilon,\phi,x)) - log\, q_\phi(g(\epsilon,\phi,x)|x)]$$

Taking gradient with respect to ϕ, we get,

$$\begin{aligned}
\nabla_\phi L_{\theta,\phi}(x) &= \nabla_\phi E_{p(\epsilon)}[log\, p_\theta(x, g(\epsilon,\phi,x)) - log\, q_\phi(g(\epsilon,\phi,x)|x)] \\
&= \int \nabla_\phi log\, p_\theta(x, g(\epsilon,\phi,x)) p(\epsilon) d\epsilon - \int \nabla_\phi log\, q_\phi(g(\epsilon,\phi,x)|x) p(\epsilon) d\epsilon \\
&= \int p(\epsilon) \nabla_\phi log\, p_\theta(x, g(\epsilon,\phi,x)) d\epsilon - \int p(\epsilon) \nabla_\phi log\, q_\phi(g(\epsilon,\phi,x)|x) d\epsilon \\
&= E_{p(\epsilon)}[\nabla_\phi log\, p_\theta(x, g(\epsilon,\phi,x))] - E_{p(\epsilon)}[\nabla_\phi log\, q_\phi(g(\epsilon,\phi,x)|x)]
\end{aligned}$$

So, taking gradient with respect to both θ and ϕ is now possible.

Whole ELBO can be written as sum of individual data point ELBO, that is:

$$log\, L_{\theta,\phi}(\mathcal{D}) = \sum_{x^i \epsilon \mathcal{D}} log\, L_{\theta,\phi}(x^i)$$

Where $\mathcal{D}$ is the set of input data. So, **Individual-datapoint ELBO** is:

$$log\, L_{\theta,\phi}(x^i) = E_{p(\epsilon)}[log\, p_\theta(x^i, z) - log\, q_\phi(z|x^i)] \quad (5)$$
$$\text{Where, } z = g_\phi(\epsilon, x^i)$$

5 Approximating the Expectation

We can form a simple Monte Carlo estimator to estimate the expectation of the individual datapoint ELBO in equation (5). Monte Carlo estimation of the expectation states that, for a random variable x with distribution $p(x)$, the expectation of any function $f(x)$ can be estimated from:

$$E_{p(x)}(f(x)) \approx \frac{1}{L}\sum_{l=1}^{L} f(x^l) \quad ;\ x^l \sim p(x)$$

So, the steps of Monte Carlo estimation for (5) are:

- Take L number of samples from the noise distribution: $\epsilon^{(l)} \sim p(\epsilon)$
- Calculate $z^{(i,l)} = g_\phi(\epsilon^{(i,l)}, x^i)$
- Calculate $log\, \widetilde{L}_{\theta,\phi}(x) = \frac{1}{L}\sum_{l=1}^{L}[log\, p_\theta(x^i, z^{(i,l)}) - log\, q_\phi(z^{(i,l)}|x^i)]$

If we use a **single noise sample** ϵ from $p(\epsilon)$ that is $L = 1$, we get,

$$\begin{aligned} log\, \widetilde{L}_{\theta,\phi}(x) &= log\, p_\theta(x^i, z) - log\, q_\phi(z|x^i) \\ &= log\, p_\theta(x^i|z) + log\, p_\theta(z) - log\, q_\phi(z|x^i) \end{aligned}$$

Estimating gradients with respect to ϕ and θ are now straightforward because the expectation sign gets omitted. Here, $q_\phi(z|x)$ is the encoder, $p_\theta(z)$ is prior and $p_\theta(x|z)$ is the decoder. Variational auto encoder comprises of all the three components.

6 Assumptions and Calculation

6.1 Distribution Assumptions

By calculating the inverse distribution we get,

$$\begin{aligned} q_\phi(z|x^i) &= p_\epsilon(g_\theta^{-1}(z))(g_\theta^{-1}(z))' \\ log\, q_\phi(z|x^i) &= log\, p(\epsilon) + log\, d_\phi(x^i, \epsilon) \end{aligned} \quad (6)$$

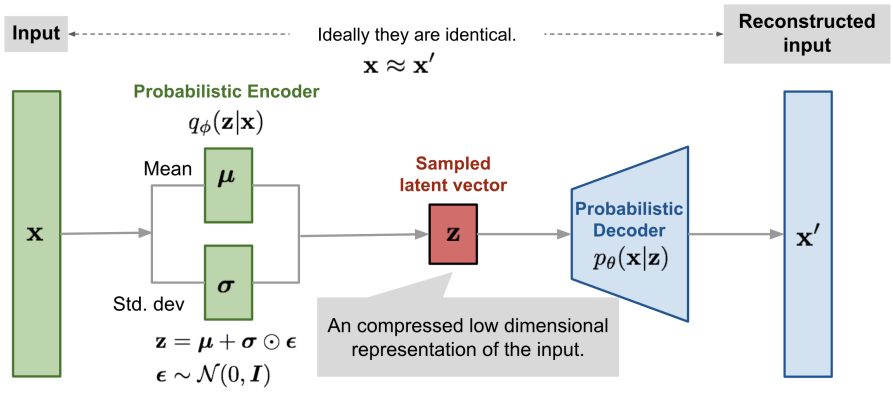

Figure 1: Overview of the VAE

Here,

$$log\, d_\phi(x^i, \epsilon) = log|det(\frac{\partial z}{\partial \epsilon})|$$

Let's assume, ϵ to be multivariate normal with $\mathbf{d}$ dimensions.

$$\epsilon \sim \mathcal{N}(0, I)$$

Also let's define $z = g_\theta(\epsilon, x^i)$ the following way,

$$z = \mu + L\epsilon$$

L is lower/upper triangular matrix. The off-diagonal elements define the correlations (covariances) of the elements in z. This way the covariance matrix can be written as $\Sigma = LL^T$ using the Cholesky decomposition. These are learned using a neural network.

$$(\mu_i, log\ \sigma_i, L_i^{'}) \leftarrow EncoderNeuralNet_\theta(x^i)$$
$$L_i \leftarrow L_{mask} \odot L_i^{'} + log\ \sigma$$

L_{mask} is a masking matrix with zeros on and above the diagonal, and ones below the diagonal for lower triangular L.

As a result, $q_\phi(z|x^i)$ becomes multivariate gaussian with d dimensions. That is,

$$q_\phi(z|x^i) = \mathcal{N}(z; \mu, \Sigma)$$

Please note, we can't explicitly assume $q_\phi(z|x)$, because from equation (6), the distribution of $q_\phi(z|x)$ depends on how we define $p(\epsilon)$ and the relation between z and ϵ, that is g_ϕ. We assumed $p(\epsilon)$ and g_ϕ such way to make $q_\phi(z|x)$ a multivariate gaussian. And we will assume $p_\theta(x|z)$ to be factorized bernoulli.

Algorithm 2: Computation of unbiased estimate of single-datapoint ELBO for example VAE with a full-covariance Gaussian inference model and a factorized Bernoulli generative model. $\mathbf{L}_{mask}$ is a masking matrix with zeros on and above the diagonal, and ones below the diagonal.

Data:

$\mathbf{x}$: a datapoint, and optionally other conditioning information

$\boldsymbol{\epsilon}$: a random sample from $p(\boldsymbol{\epsilon}) = \mathcal{N}(0, \mathbf{I})$

$\boldsymbol{\theta}$: Generative model parameters

$\boldsymbol{\phi}$: Inference model parameters

$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$: Inference model

$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$: Generative model

Result:

$\tilde{\mathcal{L}}$: unbiased estimate of the single-datapoint ELBO $\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$

$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}, \mathbf{L}') \leftarrow \text{EncoderNeuralNet}_{\boldsymbol{\phi}}(\mathbf{x})$

$\mathbf{L} \leftarrow \mathbf{L}_{mask} \odot \mathbf{L}' + \text{diag}(\boldsymbol{\sigma})$

$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$

$\mathbf{z} \leftarrow \mathbf{L}\boldsymbol{\epsilon} + \boldsymbol{\mu}$

$\tilde{\mathcal{L}}_{\text{logqz}} \leftarrow -\sum_i(\frac{1}{2}(\epsilon_i^2 + \log(2\pi) + \log \sigma_i))_i$ ▷ $= q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$

$\tilde{\mathcal{L}}_{\text{logpz}} \leftarrow -\sum_i(\frac{1}{2}(z_i^2 + \log(2\pi)))$ ▷ $= p_{\boldsymbol{\theta}}(\mathbf{z})$

$\mathbf{p} \leftarrow \text{DecoderNeuralNet}_{\boldsymbol{\theta}}(\mathbf{z})$

$\tilde{\mathcal{L}}_{\text{logpx}} \leftarrow \sum_i(x_i \log p_i + (1 - x_i) \log(1 - p_i))$ ▷ $= p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$

$\tilde{\mathcal{L}} = \tilde{\mathcal{L}}_{\text{logpx}} + \tilde{\mathcal{L}}_{\text{logpz}} - \tilde{\mathcal{L}}_{\text{logqz}}$

Figure 2: Calculation Summary

$$p_\theta(x^i|z) = Bernoulli(x^i, p)$$

$p(z)$ is also assumed to be multivariate normal, that is,

$$p(z) = \mathcal{N}(0, I)$$

Please note, although z is sampled from $p(\epsilon)$ using the relation $z = \mu + L\epsilon$, we have defined the distribution $p(z)$ as of our choice.

6.2 Calculation

Defining L that particular way in last section helps us to compute the jacobian $\frac{\partial z}{\partial \epsilon}$ easily.

$$log|det(\frac{\partial z}{\partial \epsilon})| = \sum_i log\,|L_{ii}|$$
$$= \sum_d log\,\sigma_d$$

Let's now compute $log\,p(\epsilon)$

$$p(\epsilon) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(\epsilon-\mu)^T\Sigma^{-1}(\epsilon-\mu))$$
$$= \frac{1}{(2\pi)^{d/2}|\mathbf{I}|^{1/2}} exp(-\frac{1}{2}\epsilon^T\mathbf{I}^{-1}\epsilon)$$
$$= \frac{1}{(2\pi)^{d/2}} exp(-\frac{1}{2}\epsilon^T\epsilon\mathbf{I})$$

We know, $|\mathbf{I}| = 1$ and $\mathbf{I}^{-1} = \mathbf{I}$. So,

$$p(\epsilon) = \frac{1}{(2\pi)^{d/2}} exp(-\frac{1}{2}\epsilon^T\epsilon\mathbf{I})$$
$$= \frac{1}{(2\pi)^{d/2}} exp(-\frac{1}{2}\sum_d \epsilon^2)$$
$$log\,p(\epsilon) = \sum_d (\frac{1}{2}(\epsilon_d^2 + log(2\pi))$$

So, from (6), we can write $log\,q_\phi(z|x^i)$ as

$$log\,q_\phi(z|x^i) = \sum_d (\frac{1}{2}(\epsilon_d^2 + log(2\pi) + log\sigma_d)$$

As we defined $p(z) \sim \mathcal{N}(0, I)$, we get,

$$log\,p(z) = \sum_d (\frac{1}{2}(z_d^2 + log(2\pi))$$

z_d are calculated from ϵ_d. We assumed, $p_\theta(x|z)$ to be factorized Bernoulli. So,

$$log\,p_\theta(x^i|z) = \sum_d (x_d^i log p_d + (1-x_d^i) log(1-p_d)$$

7 Normalizing Flow

More flexible posteriors help to better capture the manifold of the data. Also as we are defining $g_\phi()$ the variational posterior $q(z|x)$ are also being specified. For this reason, a more sophisticated $g_\phi()$ is used so that the variational distribution becomes more flexible.

$$
\begin{aligned}
\epsilon_o &\sim p(\epsilon) \\
for\ \ t &= 1....T \\
\epsilon_t &= f_t(\epsilon_{t-1}, x) \\
z &= \epsilon_T
\end{aligned}
$$

So, the Jacobian is factorized as:

$$log\ |det(\frac{dz}{d\epsilon_o})| = \sum_{t=1}^{T} log\ |det(\frac{d\epsilon_t}{d\epsilon_{t-1}})|$$

References

[1] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational inference with inverse autoregressive flow. *arXiv preprint arXiv:1606.04934*, 2016.

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

[3] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.