Variational Autoencoder Framework Derivation

Parth Shah[*]
IIT Guwahati

Variational Auto Encoders(VAEs) are hugely successful generative and disentangling models. Variational Auto Encoders are more intuitive than other VAE based disentangling models because of logic-driven assumptions that reduce the general "black box" like nature of ML models. Beginning from just the graphical model, we derive the framework without omitting any results or assumptions.

I. The Problem

The problem setup is quite simple. Given data X we want to learn a generative model that can reconstruct the data $\hat{X}$. Figure 1 shows the generative process.[1] Here Z is the latent representation learned by VAE[2] and X is the data of dimensions N. With the problem well defined now, we can start deriving the framework.[3]

II. VAE Framework Derivation

From Figure 1, we can write the following:

$$p(z,x) = p(z)p(x|z) \tag{1}$$

Inspired from the Autoencoders[4], the terms in Eq. (1) can be imagined as follows: $p(x|z)$ can be imagined as the probabilistic decoder & $p(z|x)$ can be imagined as the probabilistic encoder.[5]

From Bayes theorem

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{2}$$

Using conditioning, we can rewrite Eq. (2) as

$$p(z|x) = \frac{p(x|z)p(z)}{\int p(x|u)p(u)du} \tag{3}$$

Assume:

$$p(Z) \sim N(0,I) \tag{4}$$

[*] Work done while interning at IISc Bangalore.

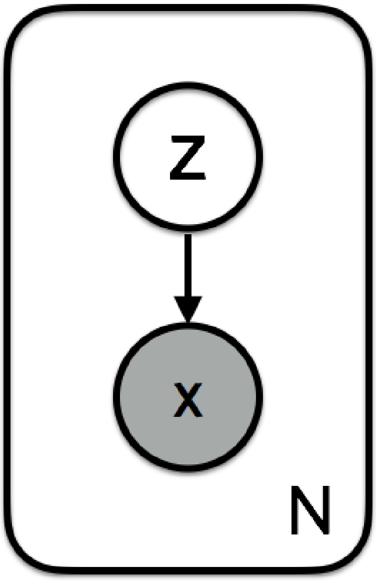

Fig. 1 The VAE generative Model

And, we can paramaterize $p(X|Z)$ for $f \in F$

$$p(X|Z) \sim N(f(Z), cI) \tag{5}$$

where F is a family of functions and $c > 0$. For now assume f is known.

Even though we know $p(x|z)$ & $p(z)$, we can't estimate $p(z|x)$ from Eqn. (3) because of the intractability of the denominator. Therefore, we resort to variational inference.

Approximate $p(z|x)$ with $q_x(z)$ s.t.,

$$q_x(z) \sim N(g(x), h(x)) \tag{6}$$

where $g \in G$ and $h \in H$, G and H are families of functions.

We want to fine $(g, h) \in G \times H$ s.t, the approximation captures most information of $p(z|x)$. We use KL Divergence to measure the *distance* between $q_x(z)$ and $p(z|x)$.

i.e.,

$$
\begin{aligned}
(g^*, h^*) &= \underset{(g,h)\in G\times H}{\text{argmin}}\ KL(q_x(z)\ ||\ p(z|x)) \\
&= \underset{(g,h)\in G\times H}{\text{argmin}}\ E[log\ q_x(z)] - E[log\ p(z|x)] && (\because KL(p(x)\ ||\ q(x) = E[log\ p(x)] - E[log(q(x))]) \\
&= \underset{(g,h)\in G\times H}{\text{argmin}}\ E[log\ q_x(z)] - E[\frac{log\ p(x|z)p(z)}{p(x)}] && (\because p(z|x) = \frac{p(x|z)p(z)}{p(x)}) \\
&= \underset{(g,h)\in G\times H}{\text{argmin}}\ E[log\ q_x(z)] - E[\frac{log\ p(z,x)}{log\ p(x)}] && (From\ (1)) \\
&= \underset{(g,h)\in G\times H}{\text{argmin}}\ E[log\ q_x(z)] - E[log\ p(z,x)] - E[log\ p(x)] && (\because E[A+B] = E[A] + E[B]) \\
&= \underset{(g,h)\in G\times H}{\text{argmin}}\ E[log\ q_x(z)] - E[log\ p(z,x)] - log\ p(x) && (\because E(c) = c)
\end{aligned}
$$

This is still intractable because of the presence of $p(x)$.

Define,

$$ELBO(g,h) := E[log\ p(z,x)] - E[log\ q_x(z)] \tag{7}$$

$$\therefore\ log\ p(x) = ELBO(g,h) + KL(q_x(z)\ ||\ p(z|x)) \tag{8}$$

From Gibb's Inequality, we know that KL Divergence is always positive. i.e.,

$$KL(p(x)\ ||\ q(x)) \leq 0$$

Since LHS of Eqn (8) is constant, maximising $ELBO$ will minimise KL Divergence.

$\therefore$ We can rewrite the minimisation problem as,

$$
\begin{aligned}
(g^*, h^*) &= \underset{(g,h)\in G\times H}{\text{argmax}}\ ELBO(g,h) \\
&= \underset{(g,h)\in G\times H}{\text{argmax}}\ E[log\ p(z,x)] - E[log\ q_x(z)] && (From\ (7)) \\
&= \underset{(g,h)\in G\times H}{\text{argmax}}\ E[log\ p(x|z) + log\ p(z)] - E[log\ q_x(z)] && (From\ (1)) \\
&= \underset{(g,h)\in G\times H}{\text{argmax}}\ E[log\ p(x|z)] - (\ E[log\ q_x(z)] - E[log\ p(z)]\ ) && (\because E[A+B] = E[A] + E[B]) \\
&= \underset{(g,h)\in G\times H}{\text{argmax}}\ E[log\ p(x|z)] - KL(q_x(z)\ ||\ p(z)) && (\because KL(p(x)\ ||\ q(x) = E[log\ p(x)] - E[log(q(x))])
\end{aligned}
$$

In Assumption (5), we assumed that f was known. Hence, now we must find out f s.t., the likelihood of the data given

z is maximum. i.e.,

$$f^* = \underset{f \in F}{\operatorname{argmax}}\ E[log\ p(x|z)]$$

$$\therefore f^*_{MLE} = f^*_{least\ squares} \qquad\qquad \because p(x|z) \sim N(f(z), cI)$$

$$\therefore f^* = \underset{f \in F}{\operatorname{argmax}}\ E[-\frac{||x - f(z)||^2}{2c}]$$

Now, our joint optimisation has become:

$$(f^*, g^*, h^*) = \underset{(f,g,h) \in F \times G \times H}{\operatorname{argmax}}\ E[-\frac{||x - f(z)||^2}{2c}] - KL(q_x(z)\ ||\ p(z)) \tag{9}$$

Therefore, our final objective function would be:

$$l_i = -ELBO(f, g, h) \tag{10}$$

Where $ELBO(f, g, h)$ is the relation from Eqn. (9). Since none of the data points share latent variables, we can use Gradient Descent (or Mini Batch or SGD) to minimise the loss measure from Eqn. (10).

References

[1] Altosar, J., "What is a variational autoencoder?" 2018. URL `https://jaan.io/what-is-variational-autoencoder-vae-tutorial/`.

[2] Kingma, D., and Welling, M., "Auto-Encoding Variational Bayes," 2014.

[3] Weng, L., "From Autoencoder to Beta-VAE," *lilianweng.github.io/lil-log*, 2018. URL `http://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html`.

[4] Hinton, G. E., and Salakhutdinov, R. R., "Reducing the Dimensionality of Data with Neural Networks," *Science*, Vol. 313, No. 5786, 2006, pp. 504–507. https://doi.org/10.1126/science.1127647, URL https://science.sciencemag.org/content/313/5786/504.

[5] Rocca, J., "What is a variational autoencoder?" *Towards Data Science*, 2019. URL https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73.