Everything within the VAE ELBO loss function

Yao Fu, Columbia University

yao.fu@columbia.edu

THU OCT 04TH 2018

While there are a bunch of blogs and tutorials about Variational AutoEncoder, I find many of them either too theoretial, or too engineering. So here is a blog to bridge the theory and the code. In this blog, I will analyze everything in the loss function of VAE, and point out how they are implemented. We assume the reader has the basic knowledge of VAE and Bayesian Learning.

Let's first take a look at the log probability of the data X and see how the ELBO loss is derived:

$$\begin{aligned}
\log p(X) &= \log \int_Z p(X, Z) dZ \quad // \text{ We introduce a latent variable Z here} \\
&= \log \int_Z \frac{p(X, Z) q(Z)}{q(Z)} dZ \quad // \text{ We use } q \text{ to approximate the pdf of } Z \\
&= \log \mathbb{E}_{q(Z)} [\frac{p(X, Z)}{q(Z)}] \quad // \text{ We write the integral as an expectation} \\
&\geq \mathbb{E}_{q(Z)} \log \frac{p(X, Z)}{q(Z)} \quad // \text{ Jensen's inequality}
\end{aligned}$$

As $p(X)$ is often called the "evidence" in Bayesian learning, the RHS gives a lower bound of $\log p(X)$, this is why it is called "evidence lower bound, ELBO".

Now we will tear the ELBO apart and interpret every component of it.

First:

$$E_{q(z)}[\log \frac{p(X, Z)}{q(Z)}] = E_{q(z|X)}[\log \frac{p(X, Z)}{q(Z|X)}]$$

This is because we assume: $q(Z) = q(Z|X)$, i.e. the latent variable Z is independent of the observed sample X. Intuitively, since Z generates X, i.e. Z is the cause of X, so Z is independent of X.

Then with Bayes' rule:

$$E_{Q(z|X)}[\log\frac{p(X,Z)}{q(Z|X)}] = E_{q(z|X)}[\log\frac{p(X|Z)p(Z)}{q(Z|X)}]$$
$$= E_{Q(z|X)}[\log p(X|Z) + \log\frac{p(Z)}{q(Z|X)}]$$

Here we have many comments on this equation:

- The first term of the RHS $\log p(X|Z)$ is implemented as **the decoder**. It takes a Z and decode it back to (reconstruct) X.
    - This is why it is called the **reconstruction loss**
    - During training, **the reparameterization trick** is caused by this term. See the paper *Autoencoding Variational Bayes*.
    - The interesting thing is, I find out that **The tensorflow probability library** can do the reparameterization for you. When you use this library, you can directly write the probability as the loss function to optimize.
    - During Inference, we can directly sample a Z and decode it.
- In the second term, the numeritor $q(Z|X)$ is **the encoder**. It takes an X and encode it into Z
- $p(Z)$ is the prior of the latent variable, it can either be **learned or fixed**.
    - In the original VAE paper *Autoencoding Variational Bayes*. $p(Z)$ is fixed as a Gaussian.
    - When $p(Z)$ and $q(Z|X)$ are all Gaussian, the term $E_{q(z|X)}[\log\frac{p(Z)}{q(Z|X)}] = \mathrm{KL}(q(Z|X)||p(Z))$ is the **KL-divergence** of the two, and it can **be solved** analytically. We have seen this in the paper *Autoencoding Variational Bayes*.
    - However, $p(Z)$ can also **be learned**, as is in the paper *Disentangled Sequential Autoencoder*.
    - In the implementation of *Disentangled Sequential Autoencoder*, all the probabilities are writen as they are because they use the tensorflow probability library. I think this is interesting.
    - What's more, $p(Z)$ can be other priors than Gaussian. In the paper *Generating Sentences by Editing Prototypes* the author proposed **a uniform distribution over the surface of a unit sphere** and shows it can be trained **without KL annealing**.

There are all the notes for the loss function. If you know other interesting interpretations, I will be very happy to learn~