Introduction to Variational Autoencoder

Group U8: Lake Yin, Zhiqi Wang

Spring 2023

1 Motivation: From Autoencoder to Variational Autoencoder

1.1 Review on Autoencoder

Autoencoder is a Deep Learning model that encode high dimensional information into latent space (encoder) to learn a lower dimension representation while ignoring the noise.

In assignment 4, we've worked with a simple autoencoder model that encode the FashionMNIST dataset into 1 hidden layer and then reconstruct it to a output layer, which is the reconstructed image.

We could think about the latent space representation as features of an image, if we just use the generator of the autoencoder to generate a new image with the latent space representation, we could get a new image that is similar to the original image.

2 Model of Variational Autoencoder

2.1 VAE structure From Autoencoder

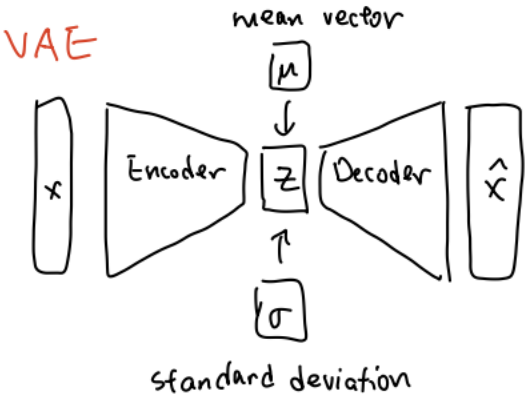

Figure 1: VAE structure

For VAE to work, we need to make some assumption about the distribution of the latent space variable. We assume that the dataset X is generated by some random process involving a unknown random variable z. This z is generated from some prior distribution $p_{\theta^*}(z)$, and the data point $x \in X$ is generated from some conditional distribution $p_{\theta^*}(x|z)$.

As the graph suggests above, we have a encoder that maps the data point x to the latent space variable z, and a decoder that maps the latent space variable z to the data point $\hat{x}$. What's different of this graph comparing to a classic autoencoder is that we have a prior distribution $p_{\theta^*}(z)$. A common choice for the prior distribution is a standard normal distribution. As the graph suggests, we have a mean vector and a standard deviation vector that maps the latent space variable z to the mean and standard deviation of the distribution.

2.2 Latent Variables

Let's define latent variables as the variables that are not directly observed. In the case of VAE, the latent variables are the variables z that are not directly observed. Since they are not observed in the input data, they are a part of our model. Therefore, we can model the observed data x's distribution as a integral of the joint distribution of x and z:

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

$p_\theta(x)$ is called the marginal likelihood. If our goal is to find θ^* such that $p_{\theta^*}(x) \approx p*(x)$. We want to model the true distribution of x.

2.3 Intractabilities

According to the definition of latent variables, we can model the observed data x's distribution as a integral of the joint distribution of x and z. However, this integral is intractable since we don't know the true distribution of z.

Also, the posterior distribution

$$p_\theta(z|x) = \frac{p_\theta(x, z)}{p_\theta(x)} \tag{2}$$

is intractable since we know that $p_\theta(x)$ is intractable. To model the distribution of x, we could use VAE.

2.4 VAE's objective function

We introduce a parametric inference model $q_\phi(z|x)$, which corresponds to the encoder in the of autoencoder/VAE. We want to find ϕ^* such that $q_{\phi^*}(z|x) \approx p^*_\theta(z|x)$.

To find ϕ^*, we want to minimize the KL divergence between $q_\phi(z|x)$ and $p^*_\theta(z|x)$:

$$\mathrm{KL}(q_\phi(z|x)||p^*_\theta(z|x)) = -\sum_z q_\phi(z|x)\log\frac{p^*_\theta(z|x)}{q_\phi(z|x)} \tag{3}$$

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x,z)}{p(x)} \tag{4}$$

$$\mathrm{KL}(q_\phi(z|x)||p^*_\theta(z|x)) = -\sum_z q_\phi(z|x)\log\frac{p^*_\theta(x,z)}{p^*_\theta(x)q_\phi(z|x)} \tag{5}$$

$$\mathrm{KL}(q_\phi(z|x)||p^*_\theta(z|x)) = -\sum_z q_\phi(z|x)\log\left[\frac{p^*_\theta(x,z)}{q_\phi(z|x)}\cdot\frac{1}{p_\theta(x)}\right] \tag{6}$$

$$\mathrm{KL}(q_\phi(z|x)||p^*_\theta(z|x)) = -\sum_z q_\phi(z|x)\log\frac{p^*_\theta(x,z)}{q_\phi(z|x)} + \sum_z q_\phi(z|x)\log p_\theta(x) \tag{7}$$

$$\mathrm{KL}(q_\phi(z|x)||p^*_\theta(z|x)) = -\sum_z q_\phi(z|x)\log\frac{p^*_\theta(x,z)}{q_\phi(z|x)} + \log p_\theta(x) \tag{8}$$

The first term in RHS in equation (8) is the ELBO (Evidence Lower Bound) and we write it as $\mathcal{L}_{\phi,\theta}(x)$. Our goal is to minimize the KL divergence between $q_\phi(z|x)$ and $p^*_\theta(z|x)$, which is equivalent to maximizing the ELBO acoording to (8).

From equation (8), we have a equation for the ELBO:

$$\mathcal{L}_{\phi,\theta}(x) = \log p_\theta(x|z) - D_{\mathrm{KL}}(q_\phi(z|x)||p_{\theta^*}(z|x)) \tag{9}$$

Define a new term ELBO (Evidence Lower Bound) as:

$$\begin{aligned}\mathcal{L}_{\phi,\theta}(x) &= \log p_\theta(x|z) - D_{\mathrm{KL}}(q_\phi(z|x)||p_{\theta^*}(z)) &(10)\\ &= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \mathbb{E}_{q_\phi(z|x)}[\log q_\phi(z|x)] &(11)\\ &= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z) - \log q_\phi(z|x)] &(12)\end{aligned}$$

3 SGD and Reparameterization Trick

3.1 SGD

The objective function of VAE is a non-convex function, so we need to use SGD to optimize the objective function. The objective function, when training with a batch of data, is:

$$\mathcal{L}_{\phi,\theta}(D) = \frac{1}{|D|}\sum_{x\in D}\mathcal{L}_{\phi,\theta}(x) \tag{13}$$

And the gradient of the objective function w.r.t. the generative model parameters θ is:

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\phi,\theta}(x) &= \nabla_\theta \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z) - \log q_\phi(z|x)] && (14)\\ &= \mathbb{E}_{q_\phi(z|x)}[\nabla_\theta \log p_\theta(x|z) - \nabla_\theta \log q_\phi(z|x)] && (15)\\ &\approx \nabla_\theta(\log p_\theta(x|z) - \log q_\phi(z|x)) && (16)\\ &= \nabla_\theta \log p_\theta(x|z) && (17)\end{aligned}$$

And the gradient of the objective function w.r.t. the variational parameters ϕ is harder to approach, since:

$$\begin{aligned}\nabla_\phi \mathcal{L}_{\phi,\theta}(x) &= \nabla_\phi \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z) - \log q_\phi(z|x)] && (18)\\ &\neq \mathbb{E}_{q_\phi(z|x)}[\nabla_\phi \log p_\theta(x|z) - \nabla_\phi \log q_\phi(z|x)] && (19)\end{aligned}$$

This is because the gradient of the objective function w.r.t. the variational parameters ϕ is a stochastic gradient, and we can't use the stochastic gradient descent to optimize the objective function in this form, since the expectation is taken over the variational parameters ϕ.

3.2 Reparameterization Trick

To solve the problem of the stochastic gradient, we can express the random variable z as a deterministic function of the variational parameters ϕ, the data point x, and a random variable ϵ that is sampled from a prior distribution $p(\epsilon)$:

$$z = g_\phi(x, \epsilon, \phi) \tag{20}$$

This is called the reparameterization trick, and the ϵ is usually sampled from a standard normal distribution $p(\epsilon) \sim \mathcal{N}(0, 1)$.

3.3 Gradient of Expectation with Reparameterization Trick

Given the change of variable $z = g_\phi(x, \epsilon, \phi)$, we can rewrite the gradient of the objective function w.r.t. the variational parameters ϕ as:

$$\mathbb{E}_{q_\phi(z|x)}[f(z)] = \mathbb{E}_{p(\epsilon)}[f(g(x, \epsilon, \phi))] = \mathbb{E}_{p(\epsilon)}[f(z)] \tag{21}$$

Then we could take gradient of equation (13) w.r.t. ϕ:

$$\begin{aligned}\nabla_\phi \mathbb{E}_{q_\phi(z|x)}[f(z)] &= \nabla_\phi \mathbb{E}_{p(\epsilon)}[f(z)] && (22)\\ &= \mathbb{E}_{p(\epsilon)}[\nabla_\phi f(z)] && (23)\\ &\approx \nabla_\phi f(z) && (24)\end{aligned}$$

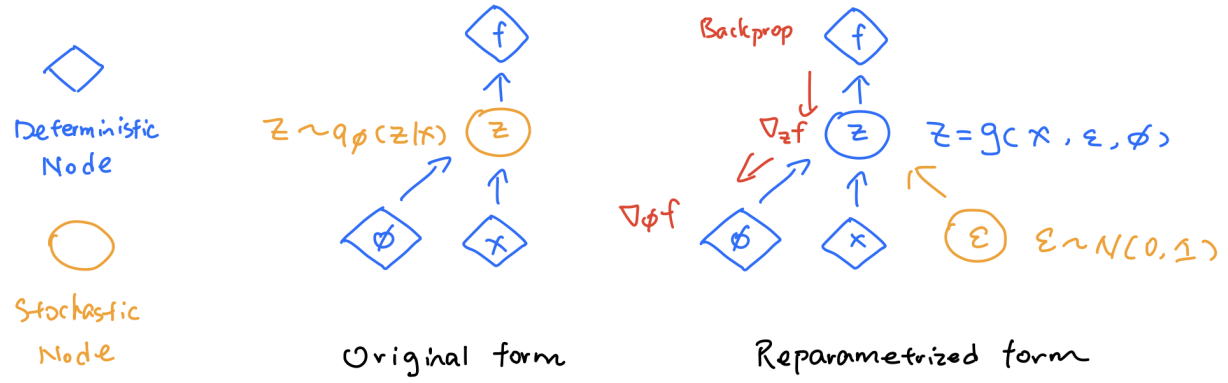

Figure 2: Reparameterization trick demonstration: on the left is the original form (without) reparameterization trick, and on the right is the reparameterization trick form. As it can be seen, the reparameterization trick enables us to take gradient of the expectation w.r.t. the variational parameters ϕ, therefore we can backprogagate the gradient to the generative model parameters θ.

3.4 Gradient of Objective Function with Reparameterization Trick

Now we can take gradient of the objective function w.r.t. the variational parameters ϕ:

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z) - \log q_\phi(z|x)] \tag{25}$$
$$= \mathbb{E}_{p(\epsilon)}[\log p_\theta(x|z) - \log q_\phi(z|x)] \tag{26}$$
$$\tag{27}$$

From equation (27), we can form our ELBO in with the reparameterized parameter z:

$$\epsilon \sim p(\epsilon) \tag{28}$$
$$z = g_\phi(x, \epsilon, \phi) \tag{29}$$
$$\tilde{\mathcal{L}}_{\theta,\phi}(x) = \log p_\theta(x|z) - \log q_\phi(z|x) \tag{30}$$

$\tilde{\mathcal{L}}_{\theta,\phi}(x)$ is a Monte Carlo estimator of the objective ELBO of a single data point x now we could use it for SGD.

4 Further Reading

This lecture note is based on the following papers:

- Auto-Encoding Variational Bayes

- An Introduction to Variational Autoencoders

We started from autoencoder to variational autoencoder, then we defined the objective based on the encoder-decoder structure, and then we derived the reparameterization trick. The original paper of VAE is a bit hard to understand since it started with the motivation from probability, so I hope this lecture note can help you understand the VAE better.