Shanghai Jiao Tong University

图像生成（VAE）实验报告

王宇昊　519030910410

2022.05

1 VAE 原理

1.1 简介

VAE，全称为变分自编码器，是一个经典的隐变量模型。顾名思义，该模型结合了变分推断和自编码器，不仅能够进行数据的特征提取，还能够进行数据的生成。为了更好地解释 VAE 的 intuition，我们从自编码器 AE 开始讲起。

自编码器是一个常见的无监督式学习模型，主要由 encoder 和 decoder 组成。从贝叶斯理论的角度进行理解，encoder 相当于后验 $p(z|x)$，能够将所给的输入 X 映射到隐空间的某一分布上；而 decoder 相当于似然 $p(x|z)$，能够根据所给的隐变量生成对应样本 x 的分布。但在 AE 模型中，先验 $p(z)$ 是不可知的，因此我们无法对随机变量 z 进行有效的采样（可理解为有可能采样出无意义的隐变量），因此 AE 并不适合用于进行生成任务，而 VAE 正是为了解决了这一问题而提出的。

1.2 VAE 公式推导

对于随机变量 x 与 z，VAE 作出了以下假设：

1. 隐变量服从标准高斯分布，即 $z \sim \mathbf{N}(0, I)$
2. 似然服从高斯分布，且可以写为 $x|z \sim \mathbf{N}(f(z), cI)$（**f** 即代表了 decoder ）

回顾 AE 的损失函数，其目的是重构输入数据，即输出与输入尽可能的相同，因此有

$$loss = ||x_{input} - x_{output}||^2 = ||x_{input} - D(E(x_{input}))||^2 \tag{1}$$

但在 VAE 中，我们对隐变量 z 以及似然 x|z 进行了约束，因此不能简单套用上面无约束的目标函数进行优化

我们首先对 encoder（即后验 p(z|x)）的优化进行推导。不妨假设 decoder（即 f）是固定的，那么根据贝叶斯公式，我们便可以从理论上求出后验 $p(z|x)$ 的表达式，即

$$p(z|x) = \frac{p(x,z)}{p(x)} = \frac{p(x|z)p(z)}{\int_z p(x|z)p(z)\mathbf{d}z} \tag{2}$$

但在 VAE 中，隐变量 z 是一个 N 元的高维向量，上述式子的积分项是无法求得的。因此，如何求解该后验分布便涉及到了机器学习的重要问题——**变分推理**。首先，由于样本 X 可以认为是从真实分布中采样得到的，因此我们认为 $p(x)$ 是确定的，那么有如下等式：

$$logp(x) = logp(x,z) - logp(z|x) \tag{3}$$

由于后验无法直接求得，我们使用 $q(z|x)$ 去逼近它，从而求得后验的近似解。为了便于计算，VAE 中假设 $q(z|x)$ 服从高斯分布，即 $z|x \sim \mathbf{N}(h(x), g(x))$。令等式两边同时乘上 $q(z|x)$ 并求期望，得到

$$\begin{aligned}
\textbf{左边} &= \int_z q(z|x) \cdot logp(x)\mathbf{d}z = logp(x) \\
\textbf{右边} &= \int_z q(z|x) \cdot logp(x,z)\mathbf{d}z - \int_z q(z|x) \cdot logp(z|x)\mathbf{d}z \\
&= \int_z q(z|x) \cdot log\frac{p(x,z)}{q(z|x)}\mathbf{d}z - \int_z q(z|x) \cdot log\frac{p(z|x)}{q(z|x)}\mathbf{d}z \\
&= E_{q(z)}[log\frac{p(x,z)}{q(z|x)}] + KL(q(z|x)||p(z|x)) \\
&= ELBO + KL_divergence
\end{aligned} \tag{4}$$

我们知道，KL divergence 始终大于等于零，因此有

$$logp(x) \geq ELBO \tag{5}$$

当且仅当 $q(z|x)$ 与 $p(z|x)$ 完全相等时，上式取等。由于 $p(x)$ 是某一固定的分布，ELBO 是有明确上界的，我们只需要让 ELBO 尽可能接近 log p(x) 即可得到 p(z|x) 的近似 q(z|x)。因此我们此时的目标由求解 $p(z|x)$ 转变为最大化 ELBO ，即

$$\begin{aligned}
h^*, g^* &= argmax_{h,g} ELBO \\
&= argmax_{h,g} E_{q(z|x)}[log\frac{p(x,z)}{q(z|x)}] \\
&= argmax_{h,g} E_{q(z|x)}[logp(x|z) + logp(z) - logq(z|x)] \\
&= argmax_{h,g} E_{q(z|x)}[logp(x|z)] - KL(q(z|x)||p(z)) \\
&= argmax_{h,g} E_{q(z|x)}(-\frac{||x - f(z)||^2}{2c}) - KL(q(z|x)||p(z))
\end{aligned} \tag{6}$$

通过以上推导，我们知道了该如何对 encoder（后验）进行优化，那么接下来我们关注于 decoder（似然）的优化。回顾 VAE 的流程，我们通过 encoder 得到某一样本 $x^{(i)}$ 在隐空间中所对应的高斯分布，然后在该高斯分布中采样得到一个隐变量 $z^{(i)}$，最后将该隐变量输入 decoder 中得到似然 $p(x|z^{(i)})$。非常自然地，我们希望能够最大化 $p(x^{(i)}|z^{(i)})$ 的期望，因此 decoder 的优化目标便为

$$
\begin{aligned}
f^* &= argmax_f \int_z p(x|z) \cdot q(z|x)\mathbf{d}z \\
&= argmax_f \int_z logp(x|z) \cdot q(z|x)\mathbf{d}z \\
&= argmax_f E_{q(z|x)}[logp(x|z)] \\
&= argmax_f E_{q(z|x)}(-\frac{||x-f(z)||^2}{2c})
\end{aligned} \tag{7}
$$

可以看到，decoder 的优化目标其实与 encoder 优化目标的第一项是一致的。因此，VAE 整体的优化目标可以写为

$$
(f^*, g^*, h^*) = argmax_{f,h,g} E_{q(z|x)}(-\frac{||x-f(z)||^2}{2c}) - KL(q(z|x)||p(z)) \tag{8}
$$

从直观的角度想，该 loss 第一项是要最小化重构误差，第二项是要让后验逼近标准正态。

2 网络实现

第一部分中我们从数学角度对 VAE 的优化目标进行了推导。我们可以非常自然地想到，如果使用神经网络建模 f,g,h 三个函数，再使用随机梯度下降法优化目标函数，便可以非常容易地实现 VAE。

因此，VAE 的模型结构大致如图 1所示，要关注的主要有三点：一是重参数化技巧，二是模型结构，三是损失函数。

2.1 重参数化技巧

可以注意到，隐变量 z 的生成是通过 $z = \mu + \epsilon\sigma$ 生成的，这便使用了重参数化技巧。这么做的原因是神经网络只能进行确定性的计算，而 z 是一个随机变量，因此需要通过额外采样一个 ϵ 来引入 z 的随机性。另外，神经网络也不能对随机变量进行梯度的求解。因此重参数化技巧是让模型可以训练的必要手段。

2.2 模型结构

对于模型结构，一般而言 VAE 的 Encoder 和 Decoder 是一个对称的结构。在本次针对 MNIST 的作业中，我们不需要太过复杂的模型框架，既可以使用不同深度的 MLP 作为 Encoder 和 Decoder，也可以使用 CNN 和 Transpose CNN 作为 Encoder 和 Decoder。因此，我尝试了如图 2所示的 4 种模型结构，依次将模型标注为①②③④，并比较了他们的性能。

2.3 Loss 函数

最后，我们需要根据第一部分的推导结果来确定模型的损失函数。对于第一项 $E_{q(z|x)}(-\frac{||x-f(z)||^2}{2c})$，其本质是重构误差，x 为输入图片，f(z) 为 VAE decoder 生成的图片。需要注意的是，求该期望需要对 $q(z|x)||x-f(z)||^2$ 进行积分，但 f 实际上是一个神经网络，我们并没有其解析表达式，该积分无法求得。一般而言，对于无法求解的均值，我们都会使用蒙特卡洛采样进行近似。回顾 VAE 的流程，在得

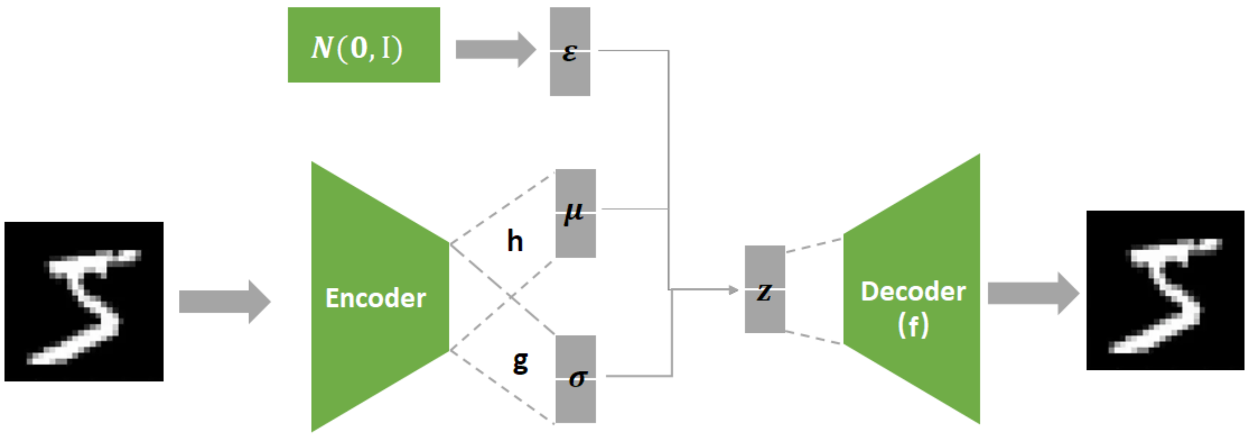


图 1: VAE 模型结构

到 encoder 输出的 μ 和 σ 后，我们将从 $N(\mu,\sigma)$ 中采样得到隐变量 z_i。如果我们在此采样得到 N 个 z_i，即有

$$E_{q(z|x)}(-\frac{||x-f(z)||^2}{2c}) \longrightarrow -\frac{1}{N}\sum_i \frac{||x-f(z_i)||^2}{2c} \quad z_i \sim q(z|x) \tag{9}$$

但在实际训练过程中，f 是一个仍待优化的网络结构，并且多次采样也会让网络前向传播变慢。因此，在实现过程中，我们仅用一次采样来近似期望，即

$$E_{q(z|x)}(-\frac{||x-f(z)||^2}{2c}) \longrightarrow -\frac{||x-f(z_i)||^2}{2c} \quad z_i \sim q(z|x) \tag{10}$$

对于公式 (8) 的第二项，有

$$\begin{aligned} KL(q(z|x)||p(z)) &= KL(N(\mu,\sigma)||N(0,1)) \\ &= log\frac{1}{\sigma}+\frac{\sigma^2+\mu^2}{2}-\frac{1}{2} \end{aligned} \tag{11}$$

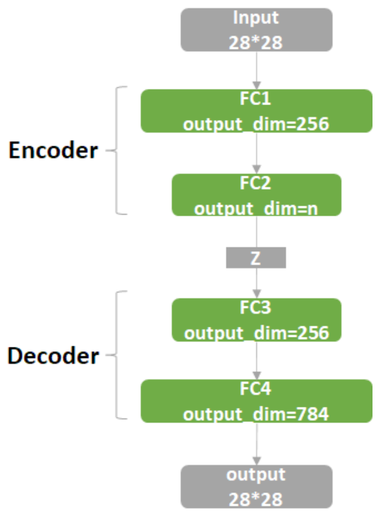
因此，最终的 Loss 函数在理论上便为

$$Loss = \frac{||x-f(z)||^2}{2c} + log\frac{1}{\sigma}+\frac{\sigma^2+\mu^2}{2}-\frac{1}{2} \tag{12}$$

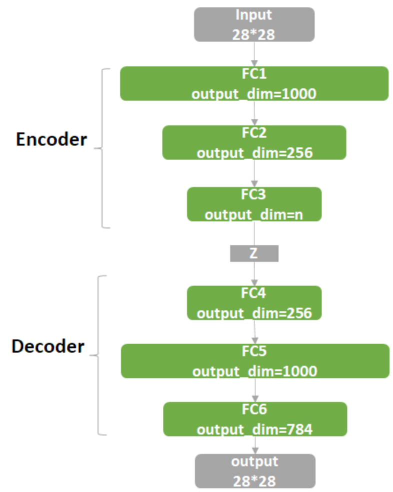
但在具体实现中，我们需要对 loss 做进一步的修改。首先，在神经网络中我们并没有对似然 x|z 的方差进行建模，因此我们并不清楚 c 的具体取值；另外，虽然我们是通过假设似然为高斯分布的方法推导出 Loss 的第一项，但如果假设似然为别的分布，其实我们可以得到不一样的表达式。在这里，我们尝试使用 L1、BCE Loss 来替代第一项重构损失，并观察由此带来的影响。因此，我们可以将 Loss 重写为

$$\begin{gathered} Loss = c\cdot Recons(x,f(z)) + KL(\mu,\sigma) \\ Recons(x,f(z)) = BCELoss(x,f(z)) \quad or \quad L1(x-f(z)) \quad or \quad L2(x-f(z)) \\ KL(\mu,\sigma) = log\frac{1}{\sigma}+\frac{\sigma^2+\mu^2}{2}-\frac{1}{2} \end{gathered} \tag{13}$$

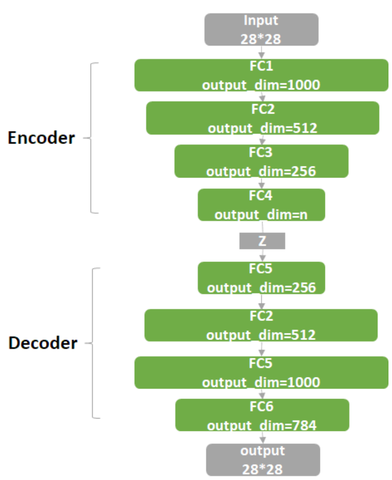
其中，第一项为重构误差，第二项为 KL 距离，c 为某一超参数。我对以上三种 Loss 都进行了尝试，并对比了结果之间的不同。



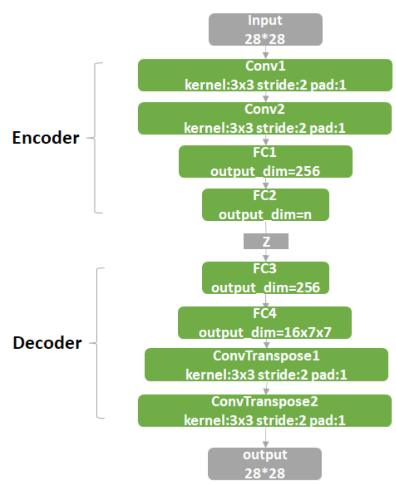
(a) ①两层 MLP 的 Encoder 和 Decoder



(b) ②三层 MLP 的 Encoder 和 Decoder



(c) ③四层 MLP 的 Encoder 和 Decoder



(d) ④基于 CNN 的 Encoder 和 Decoder

图 2: 不同模型结构

表 1: 不同模型结构

model	重构误差	KL 距离	Loss 总和
①	142.78	6.21	148.99
②	131.17	7.26	138.43
③	131.31	7.29	138.60
④	140.09	6.53	146.62

表 2: 不同的重构误差

model	重构误差	KL 距离	Loss 总和
L1	58.58	6.20	64.78
L2	27.87	5.71	33.58
BCE Loss	131.17	7.26	138.43

3 实验结果

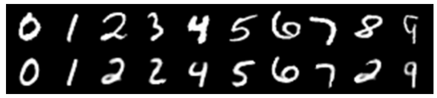
3.1 模型结构

如第二部分所述，我尝试了 4 种不同的 VAE 结构，并设定隐向量的维度为 2，实验结果如表 1所示，

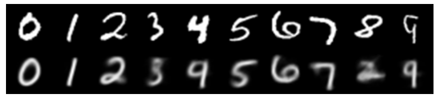
可以看到，当 MLP 由两层加深到三层时，重构误差明显减少，但从三层加深到四层后，模型的表现并没有明显变化，另外，使用 CNN 也没有太好的效果。这主要是由于 MNIST 数据量少，pattern 比较单一，因此使用复杂的网络结构容易过拟合，训练难度较大。根据上述实验结果，后序的实验将使用模型②继续进行。

3.2 Loss 函数

根据上一部分得到的结果，我们使用模型②继续进行 loss 函数的探究。在这一部分中，我们对不同的重构误差进行了实验，结果如表 2所示。由于重构误差的计算方式已经改变，单纯观察重构误差的数值并没有实质意义，因此我们直接观察重构的结果图（如图 3）。从结果来看，使用 BCE Loss 和 L2 Loss 所重构出来的数字都有一定模糊，并且 BCE loss 的重构效果会略好于 L2 Loss；使用 L1 Loss 的重构结果较差，例如数字 3 和 8 都与原图有着较大差别，但是其产生的图片锐度更大，这是因为 L2 和 BCE Loss 在接近目标值时梯度都会逐渐变小，而 L1 Loss 的梯度的绝对值则一直保持稳定，这意味着即使 Loss 变小，其梯度也不会变小，重构图片的像素值更容易趋近于 0 或 1，因此就导致了生成图片的锐度明显较大。



(a) L1



(b) L2

(c) BCE Loss

图 3: 不同形式的重构误差

表 3: 隐变量的不同维度

维度	重构误差	KL 距离	Loss 总和
1	153.90	5.03	158.93
2	131.17	7.26	138.43

4 生成结果

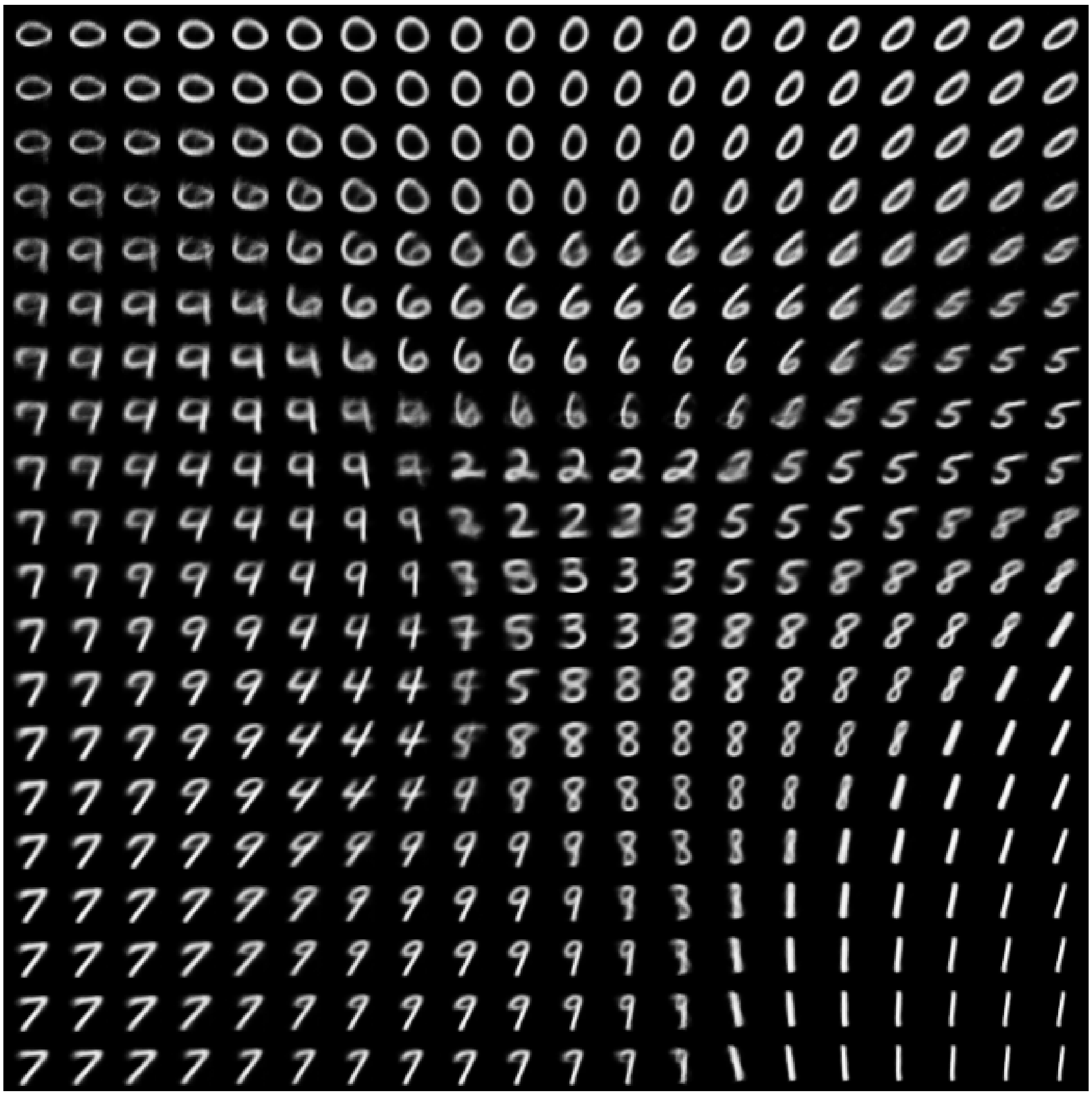
根据以上实验结果，我选择使用模型②以及 BCE Loss 进行图像的生成。首先，我将隐变量的维度设为 1，进行 VAE 的训练，然后从 [-5,5] 区间内得到不同的隐变量，使用训练好的 VAE 进行生成，得到图 4(a) 结果；相似的，我将隐变量维度设为 2，并画出隐层向量的两个维度在 [-5, 5] 区间内对应的图片生成效果（如图 4(b)）。一方面，随着隐变量维数的增多，其表征能力也随之增强，图片的生成效果也更好。当 z 的维度为 1 时，如 4、5、7 这些数字都不能较好地进行生成，而当 z 的维度为 2 时，我们就能在生成结果中找到所有种类的数字了。另一方面，我们可以明显的看到，相同种类的数字在隐空间中都聚集在一起，这意味着 VAE 能够自动完成聚类的任务。

5 思考与拓展

在做该项目的过程中，我查阅了较多资料，发现在许多中文帖子中都将 Loss 函数的第二项 KL 距离解释为模型想让后验 p(z|x) 与标准正态分布尽可能接近，但实际上这种直观的解释是不严谨，甚至错误的。我们考虑极端的情况，如果对于任意样本 $x^{(i)}$ 而言，后验 $z^{(i)}|x^{(i)}$ 都服从标准正态分布，那就意味着所有样本的隐变量的分布都是一致的，即隐变量不包含任何有效信息，这显然是不合理的。事实上，KL 距离这一项只是在 Loss 化简过程中得到的一项，并不能完全从直观的角度进行理解。

另外，在 VAE 中将先验、后验以及似然都简单地假设为高斯分布，这一方面是为了方便神经网络对分布进行建模，另一方面也是为了让 KL 距离有一个简单明了的解析解（如公式（11））。在这种假设

(a) 隐层向量 z 维度为 1



(b) 隐层向量 z 维度为 2

图 4

下，我们有

$$p(z)=\mathcal{N}(0,I)=\int_x p(z|x)p(x)dx=\int_x \mathcal{N}(\mu(x),\sigma(x))p(x)dx\approx\sum_i \mathcal{N}(\mu(x^{(i)}),\sigma(x^{(i)}))p(x^{(i)}) \tag{14}$$

可以想象，以上等式的解并不容易求得，这无形之中增大了 VAE 的训练难度。从直观上想，对于相同类别的样本，其隐变量 z 的分布应该是相似的，因此我们可以将先验 p(z) 假设为一个高斯混合分布，或许会有利于模型的训练优化。但问题在于如何计算高斯混合分布与某一高斯分布的 KL 距离。论文 [1] 中给出了两个具有相同数目 M 个高斯分量的高斯混合分布 KL 距离的上界

$$\begin{aligned}
D_{KL}[p||\hat{p}] &= \int(\sum \pi_m f_m)ln\frac{\sum \pi_m f_m}{\sum \hat{\pi}_m f_m}\\
&\leq \int \sum(\pi_m f_m)ln\frac{\pi_m f_m}{\hat{\pi}_m f_m}\\
&= \sum \pi_m ln\frac{\pi_m}{\hat{\pi}_m}\int f_m+\sum \pi_m \int f_m ln\frac{f_m}{\hat{f}_m}\\
&= \sum \pi_m ln\frac{\pi_m}{\hat{\pi}_m}+\sum \pi_m D_{KL}(f_m||\hat{f}_m)
\end{aligned} \tag{15}$$

因此，对于任一高斯分布 $p=\mathcal{N}(\mu,\sigma)=\sum_m \frac{1}{M}\mathcal{N}(\mu,\sigma)$，我们都可以计算出他与某一混合高斯分布 $\hat{p}$ 的 KL 距离的上界

$$\begin{aligned}
D_{KL}[p||\hat{p}] &\leq \sum\frac{1}{M}ln\frac{1}{M\hat{\pi}_m}+\sum\frac{1}{M}D_{KL}(\mathcal{N}(\mu,\sigma)||\mathcal{N}(\mu_m,\sigma_m))\\
&= \sum\frac{1}{M}ln\frac{1}{M\hat{\pi}_m}+\sum\frac{1}{2M}(2log\frac{\sigma_m}{\sigma}+\frac{\sigma^2+(\mu-\mu_m)^2}{{\sigma_m}^2}-1)
\end{aligned} \tag{16}$$

那么，我们就可以通过最小化上界来间接最小化 KL 散度。因此，使用公式 (16) 来替代 Loss 函数的第二项，便能够让先验 p(z) 变为我们所想要的高斯混合分布了。由于时间有限，我没有对以上思考进行代码上的实现。

6 参考文献

[1] Gjl A , Yang L A , Mzg B , et al. Variational inference with Gaussian mixture model and householder flow. 2019.

[2] Kingma D P , Welling M . Auto-Encoding Variational Bayes[J]. arXiv.org, 2014.

[3] Joseph Rocca. Sep 24, 2019. Understanding Variational Autoencoders. https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73.