Memo for Variational Autoencoder

丁岱宗

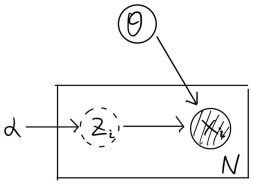
2018 年 1 月 8 日

1 引言

本 memo 是对文章《Auto-Encoding Variational Bayes》的分析。文章先从一个概率图模型切入，在求解此模型的参数时，传统方法是利用变分或者采样的方法，但是这两个方法都具有一定的局限性。为了解上述模型，作者提出了 AEVB 算法，其中在变分推断中估计后验分布时作者提出了 SGVB 算法，使得在采样之后保留分布的参数信息，从而可以用梯度下降的方法来求解。对于 AEVB 算法的应用，作者把 AEVB 中的函数用神经网络来实现，并对隐藏变量等分布做了假设，提出了 Variational Autoencoder 模型，并在图片数据集上做了实验。本 Memo 基于上面的顺序展开。

2 建模

考虑以下的概率图模型：



其中 x 服从分布 $p(x|z,\theta)$，z 服从分布 $p(z|\alpha)$。在模型中观测数据是 x，隐藏变量是 z，参数是 θ，超参数是 α。

那么假设已经有 n 个观测数据点 $\{x^{(1)}, x^{(2)}, \cdots, x^{(n)}\}$，那么可以定义如下的生成模型：

1. 对于第 i 个数据点
   (a) 从分布 $p(z|\alpha)$ 中生成隐藏变量 $z^{(i)}$
   (b) 从分布 $p(x|z^{(i)},\theta)$ 中生成数据点 $x^{(i)}$

对于上述模型中任一数据点 $x^{(i)}$，其概率可以计算为：

$$p(x^{(i)}|\theta) = \int p(x^{(i)}|z,\theta)p(z|\alpha)dz \tag{1}$$

为了求解参数 θ，需要最大化上述的似然概率，取 log 之后即：

$$\max_{\theta} \sum_{i=1}^{N} \log \int p(x^{(i)}|z,\theta)p(z|\alpha)dz \tag{2}$$

解出参数 θ 之后就完成了对这些数据点的建模。

3 目标优化的传统方法

由于上述优化目标中 log 内存在积分，如果对上式直接做梯度下降，一个积分的梯度的表达式很难求得，所以有很多方法被提出解决这个问题。比如 EM 算法，变分推断，采样等，但这几个算法都有各自的问题。本章对优化部分做简单的介绍，从而引入作者提出 AEVB 算法的动机。

3.1 EM 算法

为了求解上述目标，常见的算法为 EM，推导如下。

考虑到条件概率的定义，给定任一 $z \sim p(z)$（为了形式简单我们把 $p(z|\alpha)$ 简写成 $p(z)$）：

$$\begin{aligned} p(x,z|\theta) &= \frac{p(x,z,\theta)}{p(\theta)} \\ p(z|x,\theta) &= \frac{p(z,x,\theta)}{p(x,\theta)} \end{aligned} \tag{3}$$

所以对于任意的 z 都有：

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

对两边取 log，有：

$$\log p(x|\theta) = \log p(x,z|\theta) - \log p(z|x,\theta) \tag{5}$$

现在假设除了 z 的先验分布 $p(z)$，还有一个关于 z 的分布 $q(z|\phi)$，其中 ϕ 是分布的参数。对于上述等式的右式变形：

$$\begin{aligned} \log p(x|\theta) &= \big[\log p(x,z|\theta) - \log q(z|\phi)\big] - \big[\log p(z|x,\theta) - q(z|\phi)\big] \\ &= \log \frac{p(x,z|\theta)}{q(z|\phi)} - \log \frac{p(z|x,\theta)}{q(z|\phi)} \end{aligned}$$

两边同乘上 $q(z|\phi)$，有：

$$q(z|\phi)\log p(x|\theta) = q(z|\phi)\cdot \log \frac{p(x,z|\theta)}{q(z|\phi)} - q(z|\phi)\cdot \frac{p(z|x,\theta)}{q(z|\phi)} \tag{6}$$

现在两边都对 z 做积分，有：

$$\int q(z|\phi)\log p(x|\theta)dz = \int q(z|\phi)\cdot\log\frac{p(x,z|\theta)}{q(z|\phi)}dz - \int q(z|\phi)\cdot\log\frac{p(z|x,\theta)}{q(z|\phi)}dz \tag{7}$$

在左式积分中由于 $\log p(x|\theta)$ 中不包含 z，可以提出来，以及 $\int q(z|\phi)=1$。在右式中把第二项取负，即分式上下互换，可以得到 KL 散度的定义，所以上式可以变为：

$$\log p(x|\theta) = \underbrace{E_{q(z|\phi)}\Big[\log\frac{p(x,z|\theta)}{q(z|\phi)}\Big]}_{Lower\ Bound\ B} + \underbrace{KL\big[q(z|\phi)\|p(z|x,\theta)\big]}_{\geq 0} \tag{8}$$

由于引进了参数 ϕ，所以现在模型需要求解的参数为 $\{\theta,\phi\}$。EM 算法的思想和梯度上升法比较相近，对于上述的目标，做如下的优化：

1. **E Step：** 通过参数 ϕ 找到一个分布 $q(z|\phi)$ 使得这个分布与 z 的后验分布 $p(z|x,\theta)$ 之间的 KL 散度最小，即：

$$\min_{\phi} KL\big[q(z|\phi)\|p(z|x,\theta)\big] \tag{9}$$

2. **M Step：** 基于优化的参数 ϕ 得到似然函数的下界 B，通过优化参数 θ 从而最大化这个下界，即：

$$\max_{\theta} B = E_{q(z|\phi)}\Big[\log\frac{p(x,z|\theta)}{q(z|\phi)}\Big] \tag{10}$$

迭代两个步骤直至收敛。

具体实现上，在 E Step 中，考虑到两个分布等同时候 KL 散度最小，等于 0，这时候 Lower Bound 等于原似然。也就是说，如果两个分布的形式相同，只要找到相应的 ϕ 使得分布 $q(z|\phi)$ 与后验分布 $p(z|x,\theta)$ 完全重合，就完成了这个阶段的优化。举个例子，如果 $p(z|x,\theta)$ 是一个高斯分布，并且 $q(z|\phi)$ 也假设成了高斯分布（ϕ 包含了高斯分布的均值和方差两个参数），那么直接让 $q(z|\phi)$ 的均值和方差都等于 $p(z|x,\theta)$ 的均值和方差，就可以使得两个分布之间的 KL 散度最小。

在 M Step 中，我们对下界做展开，有：

$$\begin{aligned}
B &= E_{q(z|\phi)}\Big[\log\frac{p(x,z|\theta)}{q(z|\phi)}\Big] = E_{q(z|\phi)}\Big[\log\frac{p(x|z,\theta)p(z)}{q(z|\phi)}\Big] \\
&= E_{q(z|\phi)}\Big[\log p(x|z,\theta) - \log\frac{q(z|\phi)}{p(z)}\Big] \\
&= E_{q(z|\phi)}\big[\log p(x|z,\theta)\big] - KL\big[q(z|\phi)\|p(z)\big]
\end{aligned}$$

由于只有第一项含有参数 θ，所以在 M Step 中只需要对第一项做优化。改进后的 EM 算法如下：

1. **E Step：** 找到参数 ϕ 使得分布 $q(z|\phi)$ 与后验分布 $p(z|x,\theta)$ 完全重合。

2. **M Step：** 代入参数 ϕ，优化如下式子：

$$\max_{\theta} E_{q(z|\phi)}\big[\log p(x|z,\theta)\big] \tag{11}$$

但是在很多情况下 E Step 中 z 的后验分布是很难求得的，由于模型只定义了分布 $p(x|z,\theta), p(z)$，为了找到其形式，我们通过贝叶斯定理：

$$p(z|x,\theta) = \frac{p(x|z,\theta)p(z)}{\int p(x|z,\theta)p(z)dz} \tag{12}$$

如果 $p(x|z,\theta)$ 和 $p(z)$ 共轭，那么分母中的积分形式就简单很多，但如果两者不共轭，那么 $p(z|x,\theta)$ 的形式都很难写出来，也就无法简单的让 $q(z|\phi)$ 与其是等同的分布，导致 KL 散度不能得到 0。

除了不能直接让 KL 散度等于 0，当 $p(z|x,\theta)$ 的形式过于复杂时，连 KL 散度的表达式都很难写出来，导致 E Step 中最小化 KL 散度无法实现。换句话说，在 Eq. 8中的右式 KL 散度是写不出来的，无法优化，EM 算法无法使用。

3.2 变分推断

EM 算法存在的问题的核心在于后验分布的形式难以求得，而变分推断则是一种求后验分布的近似的一种方法。首先对于似然函数，如果只优化如下的下界：

$$\log p(x|\theta) \geq B = E_{q(z|\phi)}\Big[\log \frac{p(x,z|\theta)}{q(z|\phi)}\Big] \tag{13}$$

上述不等式也可以通过琴生不等式得到。

那么在优化下界的过程中，由于最大化下界与最小化 KL 散度等价，所以如果我们做如下的 EM 算法优化：

1. **E Step：** 通过变分参数 ϕ 最大化下界 B
2. **M Step：** 通过模型参数 θ 最大化下界 B

那么每一轮的 E Step 之后，由于固定了模型参数 θ 的同时通过调整变分参数 ϕ 最大化了 B，也就意味着通过它最小化了 $KL\big[q(z|\phi)\|p(z|x,\theta)\big]$，考虑到 KL 散度的意义是两个分布之间的距离，也就可以认为通过上述的步骤找到了隐藏变量 z 的后验分布的近似分布。

对于近似分布的表示形式，如果有多个隐藏变量如 z_1, z_2, z_3 等，对于近似分布形式的假设可以通过平均场近似来定理，拆开元素之间的关联，即：

$$q(z_1, z_2, z_3|\phi_1, \phi_2, \phi_3) = q(z_1|\phi_1) \cdot q(z_2|\phi_2) \cdot q(z_3|\phi_3) \tag{14}$$

在 E Step 寻找后验分布的近似分布时就把上述表达式代入求解变分参数 ϕ_i，然后再把表达式代入到 M Step 中求解模型的参数 θ。上述算法就是变分 EM 算法。

变分推断的主要思想就是通过琴生不等式找到似然函数的一个下界，然后通过调整变分参数 ϕ 最大化这个下界，也就实现了最小化与后验分布之间的 KL 散度，换句话说也就得到了后验分布的一种近似。但是变分推断存在以下两个问题：

1. 近似分布的形式的假设需要很谨慎，很可能会出现优化不了的情况，比如在 M Step 中，我们需要对参数 θ 做如下的优化：

$$\max_{\theta} \int q(z|\phi) \log p(x|z,\theta) dz \tag{15}$$

这个问题与后验分布一样，有些情况如果把近似分布 $q(z|\phi)$ 设置的简单或者合理一些，上述积分可以求得，但有些情况下还是无法求得上述的积分，可能会需要一些数值计算的近似计算方法，而这个就给模型的求解上增加了很大的计算量。而且算法很容易局限在某一种特殊场景中，比如 Correlated Topic Model，把 topic 服从的先验分布从 LDA 的狄利克雷改成了多元高斯，模型的变分推断难度就上了几个台阶。

2. 平均场理论中把元素之间的关联全部拆除，这样的假设在简单模型中是合理的，但是当概率图模型复杂了之后，元素之间的联系也非常复杂，这样的近似就不太合理。

3.3 MCMC 采样方法

依据大数定律，如果从一个分布中多次采样数据点，把这个数据点平均之后就可以得到这个分布的期望值，即：

$$E_{p(t)}[t] \approx \frac{1}{L} \sum_{l=1}^{L} t^{(l)} \quad , where \quad t^{(l)} \sim p(t) \tag{16}$$

同样的，也可以对函数的期望做估计，假设 $h(t)$ 是关于 t 的一个函数：

$$E_{p(t)}[h(t)] \approx \frac{1}{L} \sum_{l=1}^{L} h(t^{(l)}) \quad , where \;\; t^{(l)} \sim p(t) \tag{17}$$

在到本问题上，E Step 中需要估计 z 的后验分布的近似分布，那么如果我们把上述的期望值代入到 M Step 中，就有：

$$\max_{\theta} E_{q(z|\phi)} \big[\log p(x|z,\theta) \big] \approx \frac{1}{L} \sum_{l=1}^{L} \log p(x|z^{(l)},\theta) \;\; , where \;\; z^{(l)} \sim p(z|x,\theta) \tag{18}$$

基于已经定义好的概率 $p(x|z,\theta)$，我们可以用梯度下降等方法去求解参数 θ。

上述问题的核心在于如何从后验分布 $p(z|x,\theta)$ 中采样出 z 的样本。随着计算机的发展，有人提出了蒙特卡洛模拟法，即用计算机去生成大量伪随机数，这个方法的应用之一就是从分布中采样样本。采样中最直接的一种方法就是通过累积函数的逆函数来取得样本。假设概率累积函数为 $F(z) \in [0,1]$，其逆函数可以求得为 $F^{-1}(z)$，那么每次随机生成一个 $[0,1]$ 的随机值，通过逆函数就可以得到一个随机的样本。

但是很多情况下这个逆函数很难求得，所以有的解决方法是从已知的近似分布中采样，比如接受-拒绝采样，重要性采样等。但是这样需要让近似分布与真实分布尽可能接近，当真实分布复杂了之后仍然很难处理。

MCMC（Monte Carlo Markov Chain）方法则提供了一种新的采样方法，其思想是如果我们可以算出转移概率 $p(z|z)$，那么给定任意的初始样本 $z^{(0)}$，每一步都从转移概率

$p(z|z^{(l)})$ 中采样出新的样本，以一定概率接受或者拒绝转移，那么经过多次转移之后就可以到达马氏链的平稳状态，平稳状态之后采样出来的样本就是这个分布的样本了。但是上述还有三个问题，第一就是转移概率如何定义，第二就是接受概率如何定义，第三什么时候到达平稳状态。

Metropolis-Hastings 准则对于上述问题做了回答，这里不详细展开。特别的，对于此马尔科夫链的构造，当隐藏变量 z 有多个的时候，比如 z_1, z_2, z_3，那么转移概率就可以间接的构造出来，比如当前进行到第 l 步，那么做如下的采样：

$$\begin{aligned} z_1^{(l+1)} &\sim p(z_1|z_2^{(l)}, z_3^{(l)}, \phi, x) \\ z_2^{(l+1)} &\sim p(z_2|z_1^{(l+1)}, z_3^{(l)}, \phi, x) \\ z_3^{(l+1)} &\sim p(z_3|z_1^{(l+1)}, z_2^{(l+1)}, \phi, x) \end{aligned} \tag{19}$$

那么就认为从 $\{z_1^l, z_2^l, z_3^l\}$ 到 $\{z_1^{l+1}, z_2^{l+1}, z_3^{l+1}\}$ 就完成了一次转移（全部接受），那么等到转移之后样本的分布也没有很大变化之后就认为到达了平稳状态，上述算法就是 Gibbs Sampling。

但是上述算法存在一个很大的问题，就是分布 $p(z_1|z_2, z_3, \phi, x)$ 是后验分布，形式写不出来。所以有一些其他的 MCMC 算法，比如 Hybrid Monte Carlo，如果分布的梯度可以算出来，就可以构造出一条马尔科夫转移链，从而采样出若干个样本。

在给定一个样本 $x^{(i)}$ 的情况下，后验概率为：

$$p(z|\theta, x^{(i)}) = \frac{p(z, x^{(i)}|\theta)}{p(x^{(i)})} \propto p(x|f_\theta(z)) \cdot p(z) \tag{20}$$

那么也就有：

$$\frac{\partial \log p(z|x, \theta)}{\partial z} = \frac{\partial \log p(x|f_\theta(z))}{\partial z} + \frac{\partial p(z)}{z} \tag{21}$$

把这个梯度交给 HMC 算法即可采样出相应的样本，把样本代入概率 $p(x|z, \theta)$ 中，然后通过 Eq. 18计算期望值，然后优化参数 θ，迭代进行，此算法被称为 Sampling-based EM （MCEM），步骤如下：

1. **E Step：**利用 MCMC 方法从后验分布中采样若干个样本 $z^{(l)}$
2. **M Step：**把样本代入原先的优化目标中，调整参数 θ 解出模型。

但是以上的采样算法都是基于马尔科夫链，存在的问题是马尔科夫链的收敛需要很长时间，如果数据集本身很大，对于每个数据点 $x^{(i)}$ 都需要收敛一条马尔科夫链，虽然相对于上述的变分推断来说普适性广了很多，但时间效率是很低的。

4 AEVB 算法

通过上面的分析可以看到，在本文的概率图模型中，目前主流的几种解法，EM 受限于后验分布的表示形式；变分推断的普适性比较差，受限于近似分布的表达形式；采样方法的普适性虽然强，但是马尔科夫链的收敛速度较慢。作者提出了 AEVB 算法，核心是基于 SGVB 的采样器，并且说明了除了本文提到的概率图模型，稍微复杂一点的概率图模型通过 SGVB 也能求解。

4.1 概率图变形

先看 Eq. 8中的下界，把 KL 散度移到左边，可以得到：

$$\log p(x|\theta) - KL\big[q(z|\phi)\|p(z|x,\theta)\big] = \underbrace{E_{q(z|\phi)}\big[logp(x|z,\theta)\big] - KL\big[q(z|\phi)\|p(z)\big]}_{Lower\ Bound} \tag{22}$$

根据变分 EM 算法的思想，应该通过优化参数 ϕ,θ 实现最大化右式的下界。

值得注意的是，在最初的变分假设中 $q(z|\phi)$ 是一个任意的 z 的分布，参数为 ϕ。如果我们对其概率加上一个条件，即给定 x 的情况下 z 的概率，就变成了 $q(z|\phi,x)$。虽然这样的定义不是最严谨的变分，但是通过加上条件 x 可以使得 $q(z)$ 这个分布的选择更小，换句话说更容易学习出来，方程就变成了如下的形式：

$$\log p(x|\theta) - KL\big[q(z|x,\phi)\|p(z|x,\theta)\big] = E_{q(z|x,\phi)}\big[logp(x|z,\theta)\big] - KL\big[q(z|x,\phi)\|p(z)\big] \tag{23}$$

上面这个式子已经在 Helmholtz Machines 等模型中被应用。这个式子就是 VAE 的基础，接下来 VAE 对于其中的分布形式和解法等做了相应的定义。

首先作者提出了 encoder 和 decoder 的概念。我们先看分布 $p(x|z,\theta)$，这个分布的原始概率上定义为给定 z,θ 情况下 x 的分布。如果我们把其形式改一下：

$$x \sim p(x|f_\theta(z)) \tag{24}$$

其中 f 是一个函数，输入为 z，参数为 θ。举个例子，假设 $p(x|z,\theta)$ 为高斯分布，那么：

$$\begin{aligned} x &\sim \mathcal{N}(\mu(z),\Sigma(z)) \\ \mu(z),\Sigma(z) &= f_\theta(z) \end{aligned} \tag{25}$$

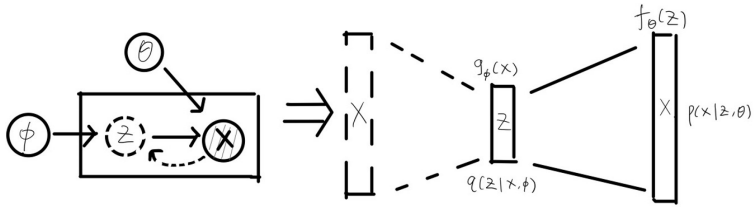
如果我们把分布 $q(z|x,\phi)$ 也做类似的定义：

$$z \sim q(z|g_\phi(x)) \tag{26}$$

那么就可以把函数 f,g 分别看做是 decoder 和 encoder，前者实现把一个 z 解码到 x 上，后者实现把 x 编码到 z 中，那么 VAE 的方程就可以写作：

$$\log \underbrace{p(x|\theta)}_{likelihood} - KL\big[\underbrace{q(z|g_\phi(x))}_{encoder}\|\underbrace{p(z|x,\theta)}_{posterior}\big] = E_{q(z|g_\phi(x))}\big[log\underbrace{p(x|f_\theta(z))}_{decoder}\big] - KL\big[\underbrace{q(z|g_\phi(x))}_{encoder}\|\underbrace{p(z)}_{prior}\big] \tag{27}$$

把 x,z 都假设成实数向量，我们就把下图左边的概率图模型（值得注意的是虚线箭头，代表 $q(z|x,\phi)$，准确的来说这个概率图模型不是严格的有向图模型）画成了右边的 auto-encoder 结构，但本质上还是左边的概率图模型。



4.2 优化方法

4.2.1 下界形式

x 为观测变量，z 是隐藏变量，模型的参数是 θ, ϕ，为了求解这个概率图模型，就需要调整参数 θ, ϕ 使得观测变量的似然概率最大，即 Eq. 27中的 Likelihood。我们使用变分的方法，优化右式中的下界即可。

首先需要定义三个分布：

1. $p(x|f_\theta(z))$：encoder 的输出所服从分布，可以假设成多项分布，高斯分布等。
2. $p(z|g_\phi(x))$：decoder 的输出所服从分布，也可以假设成高斯等分布。
3. $p(z)$：隐藏变量 z 的先验分布（已经从 $p(z|\alpha)$ 简写成 $p(z)$，这里 α 是人为设置的超参数），比如假设 z 服从参数为 α 的高斯分布。

定义了三个分布之后，我们把右式中的下界形式写出来，在给定一个数据点 $x^{(i)}$ 的情况下：

$$\begin{aligned}
\mathcal{L}_1(\phi|x^{(i)}) &= -KL\big[q(z|g_\phi(x^{(i)}))\|p(z)\big] = \int q(z|g_\phi(x^{(i)}))\log\frac{p(z)}{q(z|g_\phi(x^{(i)}))} \\
\mathcal{L}_2(\theta,\phi|x^{(i)}) &= E_{q(z|g_\phi(x^{(i)}))}\big[\log p(x^{(i)}|f_\theta(z))\big] = \int q(z|g_\phi(x^{(i)}))\log p(x^{(i)}|f_\theta(z))dz \\
\mathcal{L}(\theta,\phi|x^{(i)}) &= \mathcal{L}_1(\phi|x^{(i)}) + \mathcal{L}_2(\phi,\theta|x^{(i)})
\end{aligned} \tag{28}$$

那么模型需要最小化 loss function，如果我们采取随机梯度下降的方法，对于数据点 x_i：

$$\{\theta,\phi\} = \{\theta,\phi\} + \lambda\cdot\frac{\mathcal{L}(\theta,\phi|x^{(i)})}{\partial\{\theta,\phi\}} \tag{29}$$

其中 λ 为步长。

但是上述的求导有着一个问题，loss function 的形式未知。在实现的时候，需要把 loss function 的形式输入到类似 tensorflow 的库中，否则无法计算出这个函数的梯度，而在积分之后梯度是很难算的，比如：

$$\nabla\mathcal{L}_2(\theta,\phi|x^{(i)}) = \nabla\int q(z|g_\phi(x^{(i)}))\log p(x^{(i)}|f_\theta(z))dz \tag{30}$$

右式的式子是一个只跟 θ, ϕ 相关的函数（z 被积分掉了），但这个函数的形式需要通过积分之后才能获得，然后再对参数求导。

4.2.2 KL 散度部分

不过幸运的是，在分离出的 $\mathcal{L}_1$ 中，即 KL 散度，在一定情况下是可以求出其具体形式的。假设我们假设 $q(z|g_\phi(x))$ 和 $p(z)$ 都是高斯分布，其中前者（encoder 的输出）服从均值为 $\mu(x)$ 方差为 $\Sigma(x)$ 的高斯分布，且 $\mu(x), \Sigma(x) = g_\phi(x)$；后者（$z$ 的先验）服从均值为 0，方差为 I 的高斯分布，那么两者的 KL 散度就是：

$$KL\big[\mathcal{N}\big(\mu(x), \Sigma(x)\big) \| \mathcal{N}(0, I)\big] = \frac{1}{2}\Big[tr\big(\Sigma(x)\big) + \big[\mu(x)\big]^T - k - \log\det\big(\Sigma(x)\big)\Big] \tag{31}$$

其中 $tr(A)$ 为矩阵的迹，det 为矩阵的行列式，k 是多维高斯的维度。

通过上面的例子可以看出，如果我们适当的把 encoder 输出的分布 $q(z|g_\phi(x))$ 与 z 的先验分布 $p(z)$ 定义成类似的形式，两者的 KL 散度就可以写出来，$\mathcal{L}_1$ 的求导问题就可以解决。

4.2.3 SGVB 算法

但是在很多情况下 $\mathcal{L}_2$ 的形式是很难写出来的，比如 x 服从的分布与 z 的分布类型完全不同，比如 x 服从一个多项分布，而 z 服从高斯分布。或者说即使两个都是高斯分布，由于 log 内不像 KL 散度一样有分母，导致其积分出来的形式也比较复杂。虽然可以通过变分推断的方法去做解出后验分布的近似然后代入，但是通过上一章的介绍，变分推断很容易受到近似分布的形式的影响，不具备很强的普遍性，而在本问题中并没有对几个分布的具体形式定义，所以变分推断不是很适用本模型。同时如果我们采用采样模型，采取多个样本之后取平均得到期望值，除了上一章提到的时间效率问题外，还有如下的问题。

如果现在要对 $\mathcal{L}_2$ 中的 θ 求偏导，那么方程就可以写作：

$$\frac{\partial E_{q(z|g_\phi(x))}\big[\log p(x|f_\theta(z))\big]}{\partial\theta} \approx \frac{\frac{1}{L}\sum_{l=1}^{L}\partial\log p(x|f_\theta(z^{(l)})}{\partial\theta} \quad ,where \;\; z^{(l)} \sim q(z|g_\phi(x)) \tag{32}$$

但是如果我们尝试对参数 ϕ 求偏导：

$$\frac{\partial E_{q(z|g_\phi(x))}\big[\log p(x|f_\theta(z))\big]}{\partial\phi} \approx \frac{\frac{1}{L}\sum_{l=1}^{L}\partial\log p(x|f_\theta(z^{(l)})}{\partial\phi} = 0 \quad ,where \;\; z^{(l)} \sim q(z|g_\phi(x)) \tag{33}$$

由于期望计算值中 $p(x|f_\theta(z))$ 不包含参数 ϕ，所以导致求导值等于 0。这是由于我们从 z 的分布中采样去模拟 z 的值，平均运算之后得到的只有一个数值，而并不包含分布的参数，即采样是一种数值计算的方法，并不包含任何分布参数的信息。但是对于求解我们用的是变分的思想，即寻找到后验分布的一个近似分布表达形式，然而求解的变分参数 ϕ 又包含在采样的分布中，所以导致得不到梯度。问题的根源在于变分和采样是两种不同的思想，变分是对后验分布做近似，然后求解近似分布的参数；采样不用对分布做近似，直接通过数值模拟出期望值代入到目标中。而求下界这件事情已经参考了变分推断的思想，对于后验分布做了近似，采样的话就直接把近似分布给跳过去了，与 MCEM 就完全没有区别。

为了解决上述的问题，作者提出了一种基于“reparameterization trick”的方法，不仅可以实现从一个分布中采样得到期望数值的同时还能保留这个分布的参数信息，还可以实现采样的加速。

以文中的 z 的采样为例，在之前的定义是一个以 $g_\phi(x)$ 为参数的分布 $q(z|g_\phi(x))$，比如 $g_\phi(x)$ 代表了高斯分布的均值与方差。

现在我们换一种方式去采样，定义函数 $\tilde{g}_\phi(\epsilon, x)$，实现在原函数 $g_\phi(x)$ 输出的基础与一个小噪声 $\epsilon \sim p(\epsilon)$ 一一对应，即：

$$z^{(l)} = \tilde{g}_\phi(\epsilon^{(l)}, x), \quad where \quad \epsilon^{(l)} \sim p(\epsilon) \tag{34}$$

那么 $z^{(l)}$ 的值与 $\epsilon^{(l)}$ 就实现了一一对应，所以两个事件的发生概率相同，即：

$$q(z^{(l)}|g_\phi(x)) = p(\epsilon^{(l)}) \tag{35}$$

把 $z = \tilde{g}_\phi(\epsilon, x)$ 代入，由于一一对应关系，可以把 dz 换成 $d\epsilon$，所以可以得到如下的等式：

$$\int q(z|g_\phi(x)) \log p(x|f_\theta(z)) dz = \int p(\epsilon) \log p(x|f_\theta(\tilde{g}_\phi(\epsilon, x))) d\epsilon \tag{36}$$

再对上式做采样，就有如下的近似：

$$\int q(z|g_\phi(x)) \log p(x|f_\theta(z)) dz \approx \frac{1}{L} \sum_{l=1}^{L} \log p(x|f_\theta(\tilde{g}_\phi(\epsilon^{(l)}, x))) \tag{37}$$

也就有：

$$\begin{aligned} \frac{\partial \mathcal{L}_2(\theta, \phi|x^{(i)})}{\partial \phi} &\approx \frac{\partial \Big[\frac{1}{L} \sum_{i=1}^{L} \log p(x^{(i)}|f_\theta(\tilde{g}_\phi(\epsilon^{(l)}, x^{(i)})))\Big]}{\partial \phi} \\ &= \frac{1}{L} \sum_{l=1}^{L} \frac{\partial \log p(x^{(i)}|f_\theta(\tilde{g}_\phi(\epsilon^{(l)}, x^{(i)})))}{\partial \phi} \quad , \quad where \quad \epsilon^{(l)} \sim p(\epsilon) \end{aligned} \tag{38}$$

同理，对于参数 θ，有：

$$\frac{\partial \mathcal{L}_2(\theta, \phi|x^{(i)})}{\partial \theta} \approx \frac{1}{L} \sum_{l=1}^{L} \frac{\partial \log p(x^{(i)}|f_\theta(\tilde{g}_\phi(\epsilon^{(l)}, x^{(i)})))}{\partial \theta} \quad , \quad where \quad \epsilon^{(l)} \sim p(\epsilon) \tag{39}$$

以上采样方法就是作者在文章中提出的 Stochastic Gradient Variational Bayes (SGVB)。结合之前的分析 $\mathcal{L}_1 = -KL\big[q(z|g_\phi(x_i))\|p(z)\big]$ 只包含参数 ϕ，并且其形式通常可以写出来，所以对于下界的优化如下所示：

在实际过程中可以对 $x^{(i)}$ 取 batch，使用 ADAM 做梯度下降优化等操作。把 SGVB 应用到最开始的概率图模型中的算法被作者称为 Auto encoding variational bayes (AEVB)。

对于其中函数 $\tilde{g}$ 的选取，由于在采样过程中我们希望采样出来的值 z 与噪音 ϵ 可以实现一一对应，即 $q(z|g_\phi(x))$ 与 $p(\epsilon)$ 一一对应，所以最直接的想法就是在这两个分布之间建立起一个一一对应的函数 $\tilde{g}_\phi$。对于上述的对应，作者提出了以下三种情况：

Algorithm 1 AEVB 对单数据点更新算法

Initialize parameters θ, ϕ

repeat

Draw a datapoint $x^{(i)}$

Draw noise samples $\epsilon^{(l)} \sim p(\epsilon)$

Update θ with:

$$\theta = \theta + \lambda \cdot \frac{1}{L} \sum_{l=1}^{L} \frac{\partial \log p(x^{(i)}|f_\theta(\tilde{g}_\phi(\epsilon^{(l)}, x^{(i)})))}{\partial \theta} \tag{40}$$

Update ϕ with:

$$\phi = \phi + \lambda \cdot \left[\frac{\partial \mathcal{L}_1(\phi|x^{(i)})}{\partial \phi} + \frac{1}{L} \sum_{i=1}^{L} \frac{\partial \log p(x^{(i)}|f_\theta(\tilde{g}_\phi(\epsilon^{(l)}, x^{(i)})))}{\partial \phi} \right] \tag{41}$$

until convergence

1. 令 $q(z|g_\phi(x))$ 的概率累积函数（CDF）为 $Q(z|g_\phi(x)) : \mathcal{R}^K \to [0,1]$，若 $Q(z|g_\phi(x))$ 的反函数 $Q^{-1}(z|g_\phi(x)) : [0,1] \to \mathcal{R}^K$ 形式容易求得，就可以假设噪音服从一个 $[0,1]$ 的均匀分布，即 $\epsilon \sim \mathcal{U}(0,1)$，从而令 $\tilde{g}_\phi(\epsilon, x) = Q^{-1}(z|g_\phi(x))$。这种方法可以保证两个分布之间是一一对应的关系。

2. 当 CDF 的逆函数不容易求得时，假设分布的形式是属于 location-scale 的形式，location 控制着中心点的位置，scale 控制着分布的形状，假设两者分别为 (a, b)，则分布的累积密度函数 $F(x)$ 可以写作：

$$F(x) = F(\frac{y-a}{b}) \tag{42}$$

其中 $y = ax + b$，且这里 x, y 都可以是多维的向量。那么对于上述形式的分布，作者提出可以从其分布的标准形式 $a = 0, b = I$ 中采样噪声 ϵ，函数 $\tilde{g}_\phi(\epsilon, x)$ 也就可以定义为：

$$\tilde{g}_\phi(\epsilon, x) = a + b \cdot \epsilon \quad , where \quad \{a, b\} = g_\phi(x) \tag{43}$$

比如我们定义 $q(z|g_\phi(x))$ 为多元高斯分布或者多元拉普拉斯分布，那么噪声 ϵ 就可以从标准的多元高斯分布或者标准的多元拉普拉斯分布中采样，由于 location-scale 族的分布经过线性变换之后与标准分布等价，所以上述的对应关系也是一一对应的。

3. 第三种作者说的比较简单，就是表达当一种分布很难去直接找到一种一一对应关系的时候，可以通过分布之间的变换来找到相应的对应关系，比如 Gamma 分布实际上定义的是若干个指数族分布相加的和所服从的分布，也就是说从若干个指数分布中采取样本然后求和与从 Gamma 分布采取样本是一样的，那么如果 $q(z|g_\phi(x))$ 服从的是 Gamma 分布，噪声 ϵ 就可以从指数分布中采取若干个然后求和，两者之间也是一一对应的。

如果上述三种方法都不适用，作者提出为了保证一一对应，可以对概率累积函数的逆函数做数值计算上的近似，文中没有展开。

如果我们把 SGVB 与常见的采样算法做对比，常见的采样算法基于马尔科夫链，但是链进入稳态的时间往往不可预估的，而且通常时间都很长，原因在于 MCMC 方法不需要

知道分布的具体形式，普适性很强。但是 SGVB 算法通过一一对应关系，即把一个噪声和一个样本通过一个函数对应起来，那么每次只需要从噪声分布中随机采取样本即可，免去了等待马尔科夫链收敛这个过程，同时噪声分布的形式由于简单，采样速度更加快了很多，但是带来的负面作用是模型的普适性没有 MCMC 方法强，但是普适性又比变分推断要广一些，可以说是在时间效率和普适性上做了权衡。

4.3 似然概率计算

虽然通过采样的方法使得下界的值可以估计出来，但是似然函数本身的值即 $p(x|\theta)$ 仍然不可估计（下界与似然之间相差一个关于 z 的后验的 KL 散度不好计算），也就是说在观察的时候虽然可以知道目前似然函数的下界提高了多少，但是并不知道真实的似然函数有没有巨大的提高。由于模型本身是一个无监督模型，目标就是提高似然函数，虽然通过变分的方法优化了下界，但如果似然函数本身没有很大的提高，模型是很没有说服力的。所以作者提出了一种估计似然函数的方法，首先假设有一个关于 z 的任意分布 $\tilde{q}(z)$，那么，在给定一个样本 $x^{(i)}$ 的情况下：

$$\begin{aligned}\frac{1}{p(x^{(i)}|\theta)} &= \frac{\int \tilde{q}(z)dz}{p(x^{(i)}|\theta)} = \frac{\int \tilde{q}(z)\frac{p(x^{(i)},z|\theta)}{p(x^{(i)},z|\theta)}dz}{p(x^{(i)}|\theta)} = \int \frac{p(x^{(i)},z|\theta)}{p(x^{(i)}|\theta)}\cdot\frac{\tilde{q}(z)}{p(x^{(i)},z|\theta)}dz \\ &= \int p(z|x^{(i)},\theta)\cdot\frac{\tilde{q}(z)}{p(x^{(i)},z|\theta)}dz = \int p(z|x^{(i)},\theta)\cdot\frac{\tilde{q}(z)}{p(x^{(i)}|z,\theta)p(z)}dz \\ &= E_{p(z|x^{(i)},\theta)}\Big[\frac{\tilde{q}(z)}{p(x^{(i)}|f_\theta(z))p(z)}\Big] \end{aligned} \tag{44}$$

上述的表达式中仍然存在两个未知的分布，一个是很难求得的 z 的后验 $p(z|x^{(i)},\theta)$，一个是未知的分布 $\tilde{q}(z)$。为了解决以上两个问题，作者提出了如下的解决方案。

首先是关于 z 的后验，在上式中是对分式求 z 后验的期望，那么根据采样的思想，如果我们可以从 z 的后验分布中采取若干个样本，通过大数定律对这些样本取平均之后就可以得到分式的期望值，再次利用之前提到的基于梯度的 HMC 算法（Eq. 21）就可以采样出 z 的样本，得到了一些 z 的样本之后就可以对分式的期望值做估计，但是还有一个问题，$\tilde{q}(z)$ 是 z 的分布，仍然是未知。

在给定若干个样本之后去拟合样本的分布，常见思路是对他们的分布做假设，然后利用最大似然的方法去拟合假设的分布的参数，由于 z 的后验分布比较复杂，很难用常见的分布取描述，所以一种方法是利用非参数的方法去拟合样本的分布，比如 Kernel Density Estimator。其思想也很简单，如果给定若干个点，在每个点上都用一个很小的 Kernel 函数拟合（比如很小的高斯分布），那么当点的个数比较多的时候这么多小的分布加起来就是 z 的分布的一种近似拟合了。在文中作者并没有说明怎么去拟合这个分布，我认为是利用这类的 Estimator 去做的。

所以最终似然函数估计步骤如下：

1. 根据梯度 $\nabla_z \log p(z|\theta,x)$，利用基于梯度的 MCMC 方法采样出 z 的若干个样本 z^l。
2. 基于采样得到的样本，拟合分布 $\tilde{q}(z)$（比如用 kernel density estimator）。

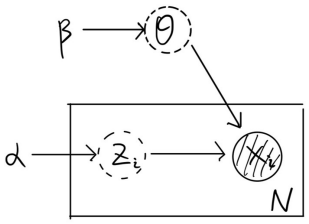
3. 再根据梯度 $\nabla_z \log p(z|\theta, x)$ 重新采样出新的样本 z^l，根据拟合到的分布 $\tilde{q}(z)$，估计似然概率：

$$p(x^{(i)}|\theta) \approx \left[\frac{1}{L}\sum_{l=1}^{L}\frac{\tilde{q}(z^{(l)})}{p(z)p(x^{(i)}|f_\theta(z))}\right]^{-1} \tag{45}$$

通过上述步骤就知道当前似然函数有提高多少，而不仅仅是下界函数提高多少，这样的结果才更有信服力。

4.4 Full Bayesian Inference

在概率图模型中，我们假设 z 的先验是一个已知的分布，其参数是超参数。但是对于参数 θ 却没有假设一个先验分布，在实际 VAE 中就是 decoder 网络中参数 θ 服从什么分布比较合适，那么完全也可以对其加上先验，认为其服从分布 $\theta \sim p(\theta|\beta)$，简写为 $p(\theta)$，那么模型图也就成了：



这里 θ, z 都是隐藏变量，积分掉之后可以得到观测变量的似然概率为：

$$\begin{aligned} p(x|\alpha,\beta) &= \int p(x|\theta)p(\theta|\beta)d\theta \\ &= \int \Big[\int p(x|f_\theta(z))p(z|\alpha)dz\Big]p(\theta|\beta)d\theta \\ &= \int p(x|\theta,z)p(\theta|\beta)p(z|\alpha)dzd\theta \end{aligned} \tag{46}$$

这里由于 α, β 都是人为设置好的超参数，所以不需要通过最大化似然概率来求解概率图中的参数。但是在实际应用中我们往往想知道隐藏变量的分布，比如给定一个样本 $x^{(i)}$ 其隐藏变量 z 的后验分布 $p(z|x^i, \theta)$ 是什么，如果应用在图片上，给定给一个图片 $x^{(i)}$，如果能求得其隐藏变量 z 的分布，那么就可以把隐藏变量的信息认为是其 Embedding 的结果（通常 z 的维度远小于 x）。对隐藏变量 θ 同理，需要计算两者的后验分布，但是基于之前的分析他们的后验分布都十分复杂，难以计算，所以也需要换一个角度，上述问题就变成了找到两个分布 $q(z|\phi_1), q(\theta|\phi_2)$，使得其与后验分布很接近。其中 ϕ_1, ϕ_2 是两个近似分布的参数。

为了求得上述目标，我们首先对 θ 做下界的寻找，我们先对 $p(x|\alpha, \beta)$，即 Eq. 46中的第一行，利用变分的方法：

$$\log p(x) = \underbrace{KL\big[q(\theta|\phi_2)\|p(\theta|x,\beta)\big]}_{\geq 0} + \underbrace{\int q(\theta|\phi_2)\Big[\log p(x|\theta) + \log\frac{\log p(\theta|\beta)}{q(\theta|\phi_2)}\Big]d\theta}_{Lower\ Bound} \tag{47}$$

为了求得 θ 的后验分布的近似，那么目标就可以写作：

$$\min_{\phi_2} KL\big[q(\theta|\phi_2)\|p(\theta|x,\beta)\big] \tag{48}$$

当 KL 散度取到 0 时即两个分布完全重合时候取到最小值，即让近似分布等于后验分布，但是基于前面的分析，大多数后验分布形式都很难写出来，所以这一项的优化也很难，问题就可以转化为：

$$\begin{aligned}
&\max_{\phi_2} \int q(\theta|\phi_2)\Big[\log p(x|\theta) + \log\frac{\log p(\theta|\beta)}{q(\theta|\phi_2)}\Big]d\theta \\
\Leftrightarrow &\max_{\phi_2} E_{q(\theta|\phi_2)}\big[\log p(x|\theta)\big] - KL\big[q(\theta|\phi)\|p(\theta|\beta)\big] \\
\Leftrightarrow &\max_{\phi_1,\phi_2} E_{q(\theta|\phi_2)}\Big[\log\int p(x|f_\theta(z))p(z|\alpha)dz\Big] - KL\big[q(\theta|\phi)\|p(\theta|\beta)\big]
\end{aligned} \tag{49}$$

虽然第二项 KL 散度可以通过手动计算得到，但是如果我们希望通过梯度下降的方法去优化下界，第一项还是很难处理。

第一个难点就是期望值的估计，如果我们使用采样方法，即从近似分布 $q(\theta|\phi_2)$ 中采若干个样本，通过取平均来得到期望值，就会有梯度变成 0 的问题，因为在期望值内 $\log p(x|\theta)$ 中并没有包含变分参数 ϕ_2。第二个难点是 $p(x|\theta)$ 中还带有对 z 的积分，虽然可以通过上一节中的方法来估计 $p(x|\theta)$ 的数值，但是仍然没有包含任何变分参数的信息，导致梯度为 0，无法优化。

所以如果不用梯度下降的方法做，第一个传统方法是采样，假设 z 和 θ 的似然和先验是共轭分布的话，就可以通过 Gibbs 采样的方法构造马尔科夫转移链，然后估计他们的后验分布（如 LDA，BPMF 等），但在本问题中不适用（不是共轭先验）；另一个传统方法是使用变分优化，利用平均场假设，拆开元素之间的联系，即：

$$q(z,\theta|\phi_1,\phi_2) = q(z|\phi_1)\cdot q(\theta|\phi_2) \tag{50}$$

然后结合一些数值计算的方法对第一项做计算（如 LDA，CTM 等）。

如果我们继续基于作者提出的 SGVB 对其优化，首先我们解决第一个难点，即如何从 $q(\theta|\phi_2)$ 中采样若干个样本算期望的同时保留变分参数 ϕ_2 的信息，那么基于 SGVB 的 reparametric trick，假设有一个噪音 η 服从分布 $p(\eta)$，就可以假设一个函数 $r_{\phi_2}(\eta)$，即输入是噪声，参数是 θ 的变分参数 ϕ_2，输出是 θ 的采样值，并且输入和输出是一一对应的关系，即：

$$\theta = r_{\phi_2}(\eta) \quad ,where \quad \eta\sim p(\eta) \tag{51}$$

那么从近似分布 $q(\theta|\phi_2)$ 采样之后 $\log p(x|\theta)$ 的期望值就可以计算为：

$$E_{q(\theta|\phi_2)}\big[\log p(x|\theta)\big] \approx \frac{1}{L}\sum_{l=1}^{L}\log p(x|\theta^{(l)}) \quad ,where \quad \theta^{(l)} = r_{\phi_2}(\eta^{(l)}) \quad and \quad \eta^{(l)}\sim p(\eta) \tag{52}$$

接下来再解决第二个问题，就是如何同时再对 $\log p(x|\theta)$ 做估计，由于其包含 z 的积分，所以我们可以考虑使用文章前面提出的采样方法和对 z 的推导，即假设存在噪音

$\epsilon \sim p(\epsilon)$，并且存在函数 $z = \tilde{g}_{\phi_1}(x, \epsilon)$，那么给定一个数据点 $x^{(i)}$，其似然函数 $p(x|\alpha, \beta)$ 的下界就可以得到如下的近似：

$$\begin{aligned}\mathcal{L}(\phi_1, \phi_2|x^{(i)}) &= E_{q(\theta|\phi_2)}\Big[\log \int p(x|f_\theta(z))p(z|\alpha)dz\Big] - KL\big[q(\theta|\phi)\|p(\theta|\beta)\big] \\ &\approx \frac{1}{L_2}\sum_{c=1}^{L_2}\Big[\frac{1}{L_1}\sum_{l=1}^{L_1}\log p(x|f_{\theta^{(c)}}(z^{(l)}) - KL\big[q(z|g_{\phi_1}(x^{(i)}))\|p(z|\alpha)\big]\Big] \\ &- KL\big[q(\theta|\phi_2)\|p(\theta|\phi_2)\big] \end{aligned} \tag{53}$$

由于两个 KL 散度都可以通过手动计算出来，令：

$$\mathcal{L}_2(\phi_1, \phi_2) = -KL\big[q(z|g_{\phi_1}(x))\|p(z|\alpha)\big] - KL\big[q(\theta|\phi_2)\|p(\theta|\phi_2)\big] \tag{54}$$

同时再做一个简化，对 θ 和 z 同时采样，那么上述下界就可以变成：

$$\begin{aligned}\mathcal{L}(\phi_1, \phi_2|x^{(i)}) &\approx \frac{1}{L}\sum_{l=1}^{L}\log p(x^{(i)}|f_{\theta^{(l)}}(z^{(l)}) + \mathcal{L}_2(\phi_1, \phi_2|x^{(i)}) \\ &\approx \frac{1}{L}\sum_{l=1}^{L}\log p(x^{(i)}|f_{r_{\phi_2}(\eta^{(l)})}(\tilde{g}_{\phi_1}(x, \epsilon^{(l)})) + \mathcal{L}_2(\phi_1, \phi_2|x^{(i)}) \end{aligned} \tag{55}$$

其中：

$$\eta^{(l)} \sim p(\eta) \quad , \quad \epsilon^{(l)} \sim p(\epsilon) \tag{56}$$

举个实例，假设 θ, z 的先验服从高斯分布 $\mathcal{N}(0, I)$，$q(\theta|\phi_1)$ 也服从一个高斯分布，那么两个采样的函数（$r_{\phi_2}(\eta), \tilde{g}_{\phi_1}(\epsilon, x)$）可以定义为：

$$\begin{aligned}\theta &= \mu_t + \sigma_t \odot \eta \quad \eta \sim \mathcal{N}(0, I) \\ z &= \mu_x + \sigma_x \odot \epsilon \quad \epsilon \sim \mathcal{N}(0, I) \end{aligned} \tag{57}$$

其中：

$$\begin{aligned}\{\mu_t, \sigma_t\} &= \phi_2 \\ \{\mu_x, \sigma_x\} &= g_\phi(x) \end{aligned} \tag{58}$$

两个 KL 散度的表达式由于都是高斯分布和高斯先验，所以可以通过手动计算出来。

通过上述的建模，就可以对模型做 Full Bayesian Inference，即对参数 θ 也加入先验分布，从而使得 θ 的值更加准确（在数据量不太大的情况下，先验具有一定的指导意义），但在实际情况中由于数据量比较大，所以不需要对参数也加上这样的先验，但作者通过这样的例子证明了 SGVB 方法的可行性，虽然 Full Bayesian 中的概率图模型通过变分之后已经不能通过 AEVB 算法来解决，但是通过 SGVB 采样方法，在更复杂的情况下也可以适用。

5 VAE

如果我们把 AEVB 中的两个函数用神经网络实现，并且设置好相应的分布，对应的模型作者称之为 Variational Autoencoder，即 VAE。

5.1 建模

基于上述原理，作者首先需要对三个分布做定义：

1. $p(x|f_\theta(z))$：先把图片的矩阵表示顺序展开成一个长向量，即 $x = [x_1, x_2, \cdots, x_M] \in \mathbb{R}^M$，再把图片上每个像素值（0-255）归一化到 $[0, 1]$，那么对于每个点来说都可以假设服从一个伯努利分布。接下来函数 $f_\theta(z)$ 是一个以 $z \in \mathbb{R}^K$ 为输入的多层神经网络，神经网络的参数为 θ，令 $y = f_\theta(z) = [y_1, y_2, \cdots, y_M] \in \mathbb{R}^M$，则似然概率定义如下：

$$p(x|f_\theta(z)) = \prod_{i=1}^{M} y_i^{x_i} \cdot (1 - y_i)^{1-x_i} \quad , where \;\; y = f_\theta(z) \tag{59}$$

除了伯努利分布，似然函数还可以定义为多元高斯分布：

$$\begin{aligned} p(x) &= \frac{1}{\sqrt{(2\pi)^M |\Sigma(z)|}} e^{\frac{1}{2}(x-\mu(z))^T \Sigma(z)^{-1}(x-\mu(z))} \\ &where \;\; \mu(z), \Sigma(z) = f_\theta(z) \end{aligned} \tag{60}$$

2. $q(z|g_\phi(x))$：定义函数 $g_\phi(x)$ 为一个多层神经网络，输入为长向量 x，输出为 $\mu \in \mathbb{R}^K, \sigma \in \mathbb{R}^K$，其中 μ 是 K 维高斯分布的均值，$\sigma^2 I$ 为 K 维高斯分布的协方差矩阵，定义概率如下：

$$\log q(z|g_\phi(x)) = \log \mathcal{N}(\mu, \sigma^2 I) \tag{61}$$

3. $p(z)$：这里定义 z 的先验分布，也为一个 K 维的高斯分布：

$$p(z) = \mathcal{N}(0, I) \tag{62}$$

定义好三个分布之后，给定数据点（n 张图片）$\{x^{(1)}, x^{(2)}, \cdots, x^{(n)}\}$，通过最大化似然函数的下界（$\mathcal{L}(\theta, \phi|x^{(i)}) = \mathcal{L}_1(\phi|x^{(i)}) + \mathcal{L}_2(\theta, \phi|x^{(i)})$）学习出参数 θ, ϕ，根据算法描述，首先需要对 Eq. 27中 KL 散度的形式求出来，代入两个高斯分布，有：

$$\begin{aligned} \mathcal{L}_1(\phi|x^{(i)}) &= -KL\big[q(z|g_\phi(x))\|p(z)\big] = \int q(z|g_\phi(x)) \log \frac{p(z)}{q(z|g_\phi(x))} dz \\ &= \frac{1}{2} \sum_{k=1}^{K} \Big(1 + \log\big((\sigma_k^{(i)})^2\big) - \big((\mu_k^{(i)})^2 - \big((\sigma_k^{(i)})^2\Big) \quad , \{\sigma^{(i)}, \mu^{(i)}\} = g_\phi(x^{(i)}) \end{aligned} \tag{63}$$

接下来再对 $\mathcal{L}_2(\phi, \theta|x^{(i)})$ 做采样估计，这里用上 reparametric trick，由于高斯分布是属于 location-scale 分布，所以采样函数 $\tilde{g}_\phi(\epsilon, x)$ 的设计就可以根据第二种情况定义，最终

的目标近似就可以写作：

$$\mathcal{L}_2(\phi,\theta|x^{(i)}) = \int q(z|g_\phi(x))\log p(x^{(i)}|f_\theta(\tilde{g}_\phi(\epsilon^{(i)},x)))dz$$

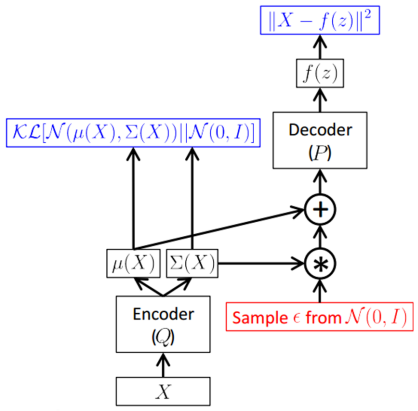
$$\approx \frac{1}{L}\sum_{l=1}^{L}\log p(x^{(i)}|f_\theta(z^{(i,l)})) \tag{64}$$

$$where \quad z^{(i,l)} = \mu^{(i)} + \sigma^{(i)} \odot \epsilon^{(l)} \quad and \quad \epsilon^{(l)} \sim \mathcal{N}(0,I) \tag{65}$$

再利用提出的优化算法，就可以解出参数 θ,ϕ，完成概率图的建模。

5.2 VAE 结构

基于上面的建模，VAE 的结构如下：



其中 Encoder 部分模拟分布 $q(z|\phi,x)$，输入一张图片 $x^{(i)}$ 经过一个神经网络 $g_\phi(x^{(i)})$ 之后输出高斯分布的参数 $\mu(x^{(i)}),\Sigma(x^{(i)})$。接下来从噪声分布 $N(0,I)$ 中采样若干个噪声样本 ϵ，经过一一对应函数 $\tilde{g}_\phi(\epsilon,x^{(i)})$ 之后输出若干个 z 的样本，取平均之后就可以得到近似分布 $q(z|\phi,x^{(i)})$ 的期望值 $z^{(i)}$。基于近似分布的期望值，经过 Decoder（也是一个多层神经网络）就可以输出分布 $p(x|\theta,z^{(i)})$ 的参数 $\mu(z^{(i)}),\Sigma(z^{(i)})$。接下来就可以计算得到似然函数的下界，其中下界的第一项（上图中的右边的蓝色框，图上省略了协方差矩阵）：

$$\log p(x^{(i)}|z^{(i)},\theta) = \frac{1}{2}[x^{(i)} - \mu(z^{(i)})]^T\Sigma(z^{(i)})^{-1}[x^{(i)} - \mu(z^{(i)})] - \frac{1}{2}\log|\Sigma(z)| \tag{66}$$

下界的第二项即 KL 散度部分已经在上一节中求得（上图中左边的蓝色框）。接下来对下界的这两项做优化即可。

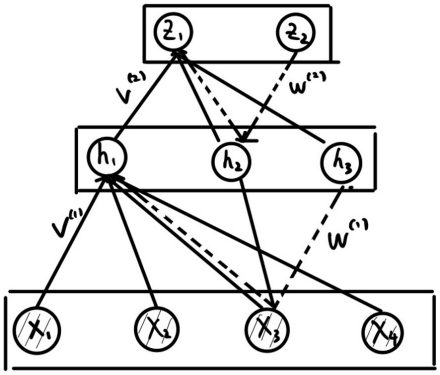
5.3 实验

5.3.1 AEVB 性能探究

为了验证提出的算法效果如何，作者在两个图像数据集上面做了验证，第一个是 MNIST，第二个是 Frey 人脸数据集，两个数据集都划分了训练集与测试集，在训练集上优化参数，在测试集上观察似然函数的概率是多少（具体估计方法在前一章节有描述）。

在模型 baseline 的选取上，由于本文的重点是对于一类概率图，提出了 AEVB 的训练方法，并且以 VAE（把概率图中的结构用神经网络来实现）作为实例跑实验。为了证明 AEVB 的性能，那么选取的 baseline 就是对于此概率图（并且其中结构也是神经网络）的其他训练方法，作者选取了如下两个经典训练方法作为 baseline：

1. **Wake-sleep Agorithm：** WS 算法也是生成式模型中比较经典的一种方法，专门解决无向神经网络（如 DBN）的训练问题。以三层无向神经网络为例，其结构如下：



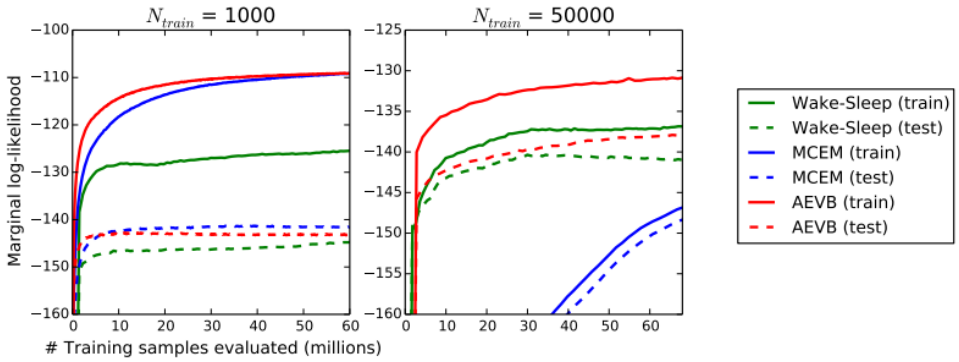
其中 x 是输入层，h 是隐藏层的神经元，z 是最终的特征。在无向神经网络中，从 z 到 x 使用的是权重 w，从 x 到 z 使用的是权重 v。WS 算法的训练方法如下，假设观测数据是 r：

(a) Wake 阶段：通过参数 v 从 r 从下往上生成 h, z，即把这两层神经元的状态设为 h, z。接下来方向反过来，在每一层中对参数 w 做训练。举个例子，通过参数 $w^{(2)}$ 由 z 计算出 h'，通过 h' 与 h 的差异更新参数 $w^{(2)}$。参数 $w^{(1)}$ 类似。

(b) Sleep 阶段：通过参数 w 从 z 从上往下生成 h, x，即把这两层的神经元状态设为 h, x。同样的方向反过来，对于每一层训练参数 v。

2. **MCEM：** 利用 HMC 在 E Step 采样出 z 的后验样本，然后代入 M Step 中取平均计算出下界的期望，通过参数 θ 最大化下界，具体在之前的章节中有介绍。

对于两个 baseline，WS 算法中从 x 到 z 和从 z 到 x 的参数都不同，所以也可以认为是一种 autoencoder 的结构，而 MCEM 则是直接对 z 的后验做采样估计，并没有使用变分参数 ϕ，所以其只有 decoder 的结构，三者的实验效果如下：

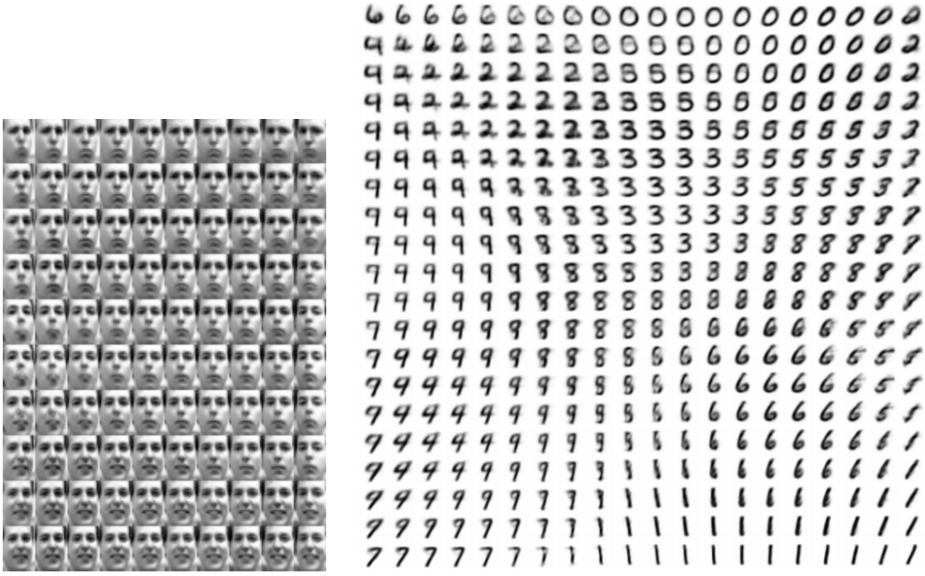


可以看到当数据量增大之后，AEVB 和 WS 算法的效果都远高于 MCEM，并且收敛速度也快很多，从侧面验证了变分结构 $q(z|x, \phi)$ 的合理性。但是同样是具有类似 auto-

encoder 的结构，训练速度也类似，AEVB 的实验效果要比 WS 要好，这个也是合理的，因为 WS 算法只是提出的一种算法，并没有很强的理论基础，只是在之前的实验上做了验证，但是 AEVB 基于变分和采样理论，有着比较强的理论基础，所以其效果好也是可以接受的。

5.3.2 z 的分布

在训练好 VAE 之后，给定任意 $z \sim p(z)$，理论上都可以通过分布 $p(x|\theta, z)$ 看生成的 x 的分布，应用在 VAE 上就是对于随机的一个噪声输入到 decoder 中，观察 x 如何，如下图所

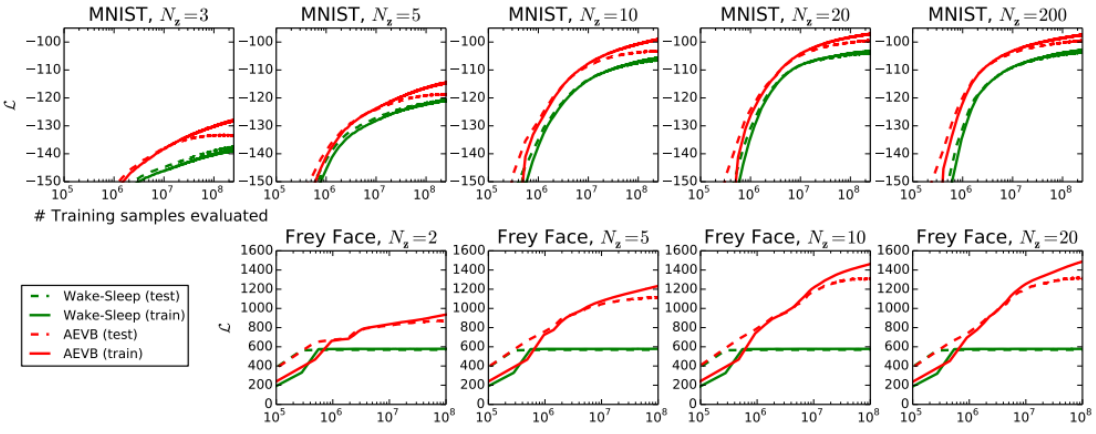


(a) Learned Frey Face manifold　　(b) Learned MNIST manifold

作者把 VAE 的隐藏变量 z 的维度设为 2 维，所以 z 服从一个二维高斯分布。接下来从以 0 为中心的二维坐标中随机采样 z 通过 decoder 生成 x 画在坐标轴上，就得到了上面的图片。可以看到 VAE 的确学习出了 z 的特征，换句话说，对 z 假设的高斯先验是非常合理的。

5.3.3 过拟合探究

作者还做了如下的对比试验：



可以看到随着 z 维度的增加，VAE 与 WS 性能上的差异也逐渐拉大，说明 VAE 的防止过拟合能力很强。这是由于由于对 z 假设了先验分布，模型可以很好的防止过拟合现象，而在 WS 算法中并没有对特征 z 做先验的假设，当 z 的维度增大之后可能会有过拟合的现象。

6 总结

本文针对一类概率图模型提出了一种解法，在此类概率图上传统解法主要是 EM 算法，但是 EM 算法受限于后验分布的表达形式。当后验分布难以写出来时，主流方法分为变分推断和采样，用这两种方法在 E Step 中找到后验分布的近似或者找到后验分布的期望，从而代入 M Step 做模型参数的优化。但是这两种方法都存在一定的问题，变分推断受限于近似分布的形式，如果近似分布比较复杂，在 M Step 中的积分问题还是无法处理，而且当变量复杂起来平均场理论的近似也会有较大的误差；而采样的普适性就强很多，不需要准确表达形式，比如给梯度就能采样出结果，但是现有的采样方法都是基于 MCMC，即构造一条马尔科夫转移链，等转移稳定之后得到的样本拿出来，导致采样的时间通常比较久，很难处理大规模的数据集情况。

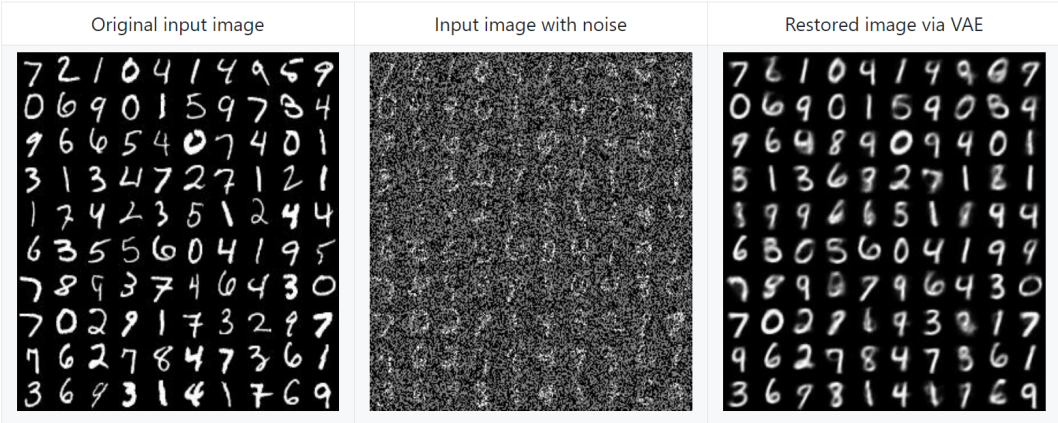
而本文针对此概率图模型，则提出了一种解法，叫做 **AEVB**。首先参照变分推断的思想，通过设置一个近似分布 $q(z)$ 找到原似然函数的下界，然后通过模型参数和变分参数直接对下界做优化。在优化的时候，并没有使用传统变分方法，而是通过采样的方法，找到 z 的期望，从而去掉了下界中积分的形式。本文还提出了 **SGVB** 的采样方法，与常见的 MCMC 构造马尔科夫链采样不同，SGVB 是建立起噪声分布与样本分布的一一对应关系做采样，从而使得采样之后依然保留近似分布中的参数信息，代入下界中做优化。SGVB 有着其他广泛的应用，不止可以应用在 AEVB 算法中，还可以对更复杂的概率图模型做优化。为了验证 AEVB 的有效性，作者把其中的函数用神经网络结构来实现，即 **VAE**，对于此神经网络结构，目前的方法有 **Wake Sleep** 以及 **MCEM**，实验对比之下 MCEM 基于马尔科夫链时间效率很低，WS 不能对 z 设置先验而且其解法没有很强的理论基础，两个 baseline 无论是效果还是收敛速度都不如 VAE，也就验证了 AEVB 和 SGVB 的有效性。SGVB 可以说同时结合了变分与采样方法的优点，相对于变分推断其适用性更广，相对于采样其速度更快。但是对于 SGVB 其要求需要一一对应的映射，除了逆函数能求在文中提到的可行的方法也只有 location-scale 的概率分布，可以认为 SGVB 的局限性就在于只方便对这一类分布建模，但这一类分布通常都够用了，而且用神经网络的结构实现也可以求解。

如果我们把目光放在 VAE 上，对于 **Autoencoder** 已经有工作证明了在 z 没有先验概率的情况下，AE 的目标（最小化重构的 $\tilde{x}$）等于最大化概率 $\log p(x|f_\theta(z))$，也就是 AEVB 中的下界里第一项，而考虑到 z 没有先验，解法就简单了很多，用梯度下降就可以解决。但是 AE 没有设置先验概率可能存在两个问题，第一就是只能对离散的数据点做建模，其抗噪性就差很多（噪声是连续的分布，而离散的点无法对连续分布做建模）；第二就是缺少正则化很可能会出现过拟合，神经网络结构复杂一点或者 z 的维度高了之后都解决不了。对于第一个问题的解决，有工作提出 **DAE**，Denoising Autoencoder，在输入端加噪声，提升模型的抗噪性，但是 DAE 是没有很强的理论基础的，即认为在一个个离散的数据点附近加噪声就可以逐渐逼近数据点真实的分布，唯一的理论基础就是基于流形的

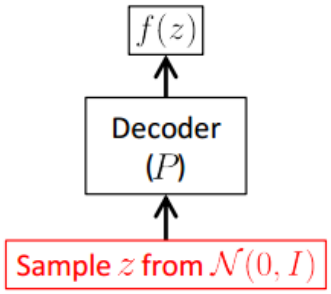
简单解释（但是没有很自圆其说，这里不展开）。而 VAE 把噪声加在 z 上这件事情，看起来是简单的实现，但其背后的理论基础却很强，基于其提出的采样算法 SGVB。

而且 VAE 的结构还有如下两个特性：

1. **抗噪性：** VAE 也具有很强的抗噪性，$z = g_\phi(x)$ 是一个把 x 映射到 z 的函数，哪怕在 x 上任意加噪声，得到的 z 也在 $p(z)$ 分布的定义域上，还是可以通过 $f_\theta(z)$ 得到有意义的 x，如下图：



2. **生成模型：** 在作者的实验中我们可以看到，VAE 可以从任意 $z \sim p(z)$ 中生成 x，即如下的结构：



这个是 autoencoder 所做不到的，autoencoder 如果没有输入就没有重构图像，而 VAE 对 z 假设了分布之后就可以达到上述的效果。

在这篇文章中，把变分下界中的分布分布 $q(z|\phi)$ 变成 $q(z|x,\phi)$，并不是其主要贡献，他的贡献主要是提出了 SGVB，使得采样加速的同时还保留了模型参数。但换个角度，把 $q(z|x,\phi)$ 变成 $q(z|g_\phi(x))$ 之后用 encoder 和 decoder 解释也可以认为是一种贡献。所以在随后，有些工作主要基于其 SGVB 采样方法，也有些工作基于 VAE 这个结构。