6 算例 C:«引入深度学习模型 ϕ,简化 X 和 Z——变分自编码器 (VAE)»

在上一章中，我们处理了一个“混合推断”模型，其中既有需要逐个优化的经典因子，也有通过神经网络分摊计算压力的摊销因子。为了更透彻地理解摊销推断（Amortized Inference）的力量，本章将探讨其在深度学习领域的旗舰应用——标准变分自编码器（VAE）。

6.1 变分自编码器 (VAE) 的算例定义

为了透彻理解摊销推断（Amortized Inference）的纯粹形式，我们将变分自编码器（VAE）作为一个标准算例引入。在 VAE 框架下，所有观测样本共享同一套推断逻辑。

6.1.1 变量定义与逻辑对齐

VAE 的变量设定与本讲义之前关于“摊销因子 Z_1”的推导具有完美的逻辑一致性。在本算例中，我们不再保留经典的坐标更新因子 Z_2, Z_3，而是聚焦于全局映射：

符号	描述	在本讲义中的逻辑对应
X	观测变量（如高维图像、基因表达谱）	对应前章的观测值 X
Z	低维隐变量（Latent Representation）	对应前章的摊销因子 Z_1
ϕ	变分参数（Encoder 神经网络权重）	对应推断网络参数 ϕ
θ	生成参数（Decoder 神经网络权重）	对应生成模型参数（通常视为超参数或全局参数）

6.1.2 概率模型设定：先验与似然

在 VAE 作为一个生成模型时，其概率结构定义如下：

1. **隐变量先验 (Prior)**：假设隐空间服从标准正态分布，为所有样本提供统一的流形约束：

$$P(Z) = \mathcal{N}(Z|0, \mathbf{I})$$

2. **观测似然 (Likelihood)**：由生成网络（Decoder）参数化。给定隐变量 Z，观测值 X 的生成过程为：

$$P_\theta(X|Z) = \mathcal{N}(X|f_\theta(Z), \sigma^2\mathbf{I}) \quad \text{或} \quad \text{Bernoulli}(f_\theta(Z))$$

这里 f_θ 是由参数 θ 定义的深度神经网络。

3. **变分后验 (Variational Posterior)**：由于真实后验 $P(Z|X)$ 不可积，我们使用推断网络（Encoder）进行摊销估计：

$$q_\phi(Z|X) = \mathcal{N}(Z|\mu_\phi(X), \sigma^2_\phi(X)) \tag{6-1}$$

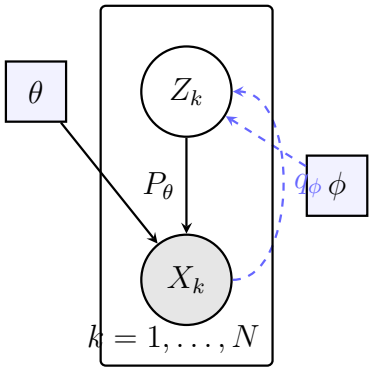

图 4: 标准 VAE 的 PGM 表示。实线代表生成模型 $P_\theta(X|Z)P(Z)$，蓝色虚线代表由编码器参数化的摊销推断路径 $q_\phi(Z|X)$。

6.1.3 概率图模型 (PGM)

VAE 的概率图反映了生成路径（实线）与推断路径（虚线）的结合。与第 5 章的混合模型相比，它的结构更为纯粹——所有样本 X_i 的隐变量推断完全依赖于全局参数 ϕ。

为了更直观地理解 VAE 的工作原理，我们将模型分解为生成路径与推断路径。这种拆分有助于理解全局参数 θ 和 ϕ 是如何分别作用于所有样本的。

(a) 生成模型 $P_\theta(X|Z)$ (b) 摊销推断 $q_\phi(Z|X)$

图 5: VAE 的生成与推断路径对比。(a) 展示了从隐变量到观测数据的生成逻辑，由参数 θ 控制；(b) 展示了从观测数据到隐空间的摊销推断逻辑，由参数 ϕ 控制。

6.2 VAE 的目标函数：ELBO

在摊销推断框架下，我们的目标是最大化对数边缘似然 $\log P(X)$。由于其不可积性，我们转而优化其下界（ELBO）。本节将详细展示如何从概率分解的角度推导出 VAE 的损失函数。

6.2.1 ELBO 的基本定义

对于单个样本 X，引入变分分布 $q_\phi(Z|X)$，根据 Jensen 不等式，ELBO 定义为：

$$\mathcal{L}_{ELBO}(\phi, \theta) = \mathbb{E}_{q_\phi(Z|X)} \left[\log \frac{P_\theta(X, Z)}{q_\phi(Z|X)} \right]$$

利用联合概率分解 $P_\theta(X, Z) = P_\theta(X|Z)P(Z)$，我们可以将上式展开为两项：

$$\mathcal{L}_{ELBO}(\phi, \theta) = \underbrace{\mathbb{E}_{q_\phi(Z|X)}[\log P_\theta(X|Z)]}_{\text{重建项 (Reconstruction)}} + \underbrace{\mathbb{E}_{q_\phi(Z|X)} \left[\log \frac{P(Z)}{q_\phi(Z|X)} \right]}_{\text{正则化项 (Regularization)}}$$

1. 重建项与交叉熵的关系 第一项 $\mathbb{E}_{q_\phi(Z|X)}[\log P_\theta(X|Z)]$ 衡量了从隐变量 Z 恢复观测值 X 的准确度。在给定 Z 的情况下，$\log P_\theta(X|Z)$ 实际上是观测分布的对数似然。

- **情形 A (伯努利分布)**：若 $X \in \{0,1\}^D$ 是二值数据（例如黑白图像），$P_\theta(X|Z)$ 为伯努利分布，则该项的负数 $-\mathbb{E}_q[\log P]$ 恰好等于 X 与生成值 $f_\theta(Z)$ 之间的 **二元交叉熵 (Binary Cross Entropy)**。
- **情形 B (高斯分布)**：若 $P_\theta(X|Z) = \mathcal{N}(X|f_\theta(Z), \sigma^2\mathbf{I})$，则该项对应于 **均方误差 (MSE)** 重建损失。

证明（情形 A）：

若观测数据 $X \in \{0,1\}^D$ 为二值向量，我们假设每个维度 x_i 独立服从以神经网络输出 $\hat{x}_i = f_\theta(Z)_i$ 为参数的伯努利分布：

$$P_\theta(X|Z) = \prod_{i=1}^{D} \hat{x}_i^{x_i}(1-\hat{x}_i)^{1-x_i}$$

对似然函数取对数：

$$\log P_\theta(X|Z) = \sum_{i=1}^{D} [x_i \log \hat{x}_i + (1-x_i)\log(1-\hat{x}_i)]$$

在深度学习中，二元交叉熵 (BCE) 的定义为：

$$\mathrm{BCE}(X, \hat{X}) = -\sum_{i=1}^{D} [x_i \log \hat{x}_i + (1-x_i)\log(1-\hat{x}_i)]$$

因此，最大化对数似然等价于最小化 BCE 损失：

$$-E_{q_\phi}[\log P_\theta(X|Z)] = E_{q_\phi}[\mathrm{BCE}(X, f_\theta(Z))] \tag{6-2}$$

证明（情形 B）：

高斯分布与均方误差 (MSE) 若观测数据 $X \in \mathbb{R}^D$ 为连续变量（例如基因表达量或彩色图像），假设 $P_\theta(X|Z)$ 服从均值为 $f_\theta(Z)$、方差为 σ^2 的多元高斯分布：

$$P_\theta(X|Z) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(-\frac{|X - f_\theta(Z)|^2}{2\sigma^2}\right)$$

取对数似然：

$$\begin{aligned} \log P_\theta(X|Z) &= \log\left(\tfrac{1}{(2\pi\sigma^2)^{D/2}}\right) - \tfrac{|X-f_\theta(Z)|^2}{2\sigma^2} \\ &= -\tfrac{1}{2\sigma^2}|X - f_\theta(Z)|^2 - \tfrac{D}{2}\log(2\pi\sigma^2) \end{aligned}$$

忽略常数项 $\frac{D}{2}\log(2\pi\sigma^2)$，其负对数似然与均方误差 (MSE) 成正比：

$$-\log P_\theta(X|Z) \propto |X - f_\theta(Z)|^2$$

若固定 $\sigma^2 = 0.5$（即常见简化设定），则重建项的负值恰好等于 MSE 损失：

$$-E_{q_\phi}[\log P_\theta(X|Z)] = E_{q_\phi}[\mathrm{MSE}(X, f_\theta(Z))] + \mathrm{const} \tag{6-3}$$

2. 正则化项与 KL 散度的转换 第二项可以利用 KL 散度的定义进行恒等变形：

$$\begin{aligned}\mathbb{E}_{q_\phi(Z|X)}\left[\log \frac{P(Z)}{q_\phi(Z|X)}\right] &= \int q_\phi(Z|X)\log\frac{P(Z)}{q_\phi(Z|X)}\,dZ \\ &= -\int q_\phi(Z|X)\log\frac{q_\phi(Z|X)}{P(Z)}\,dZ \\ &= -D_{KL}(q_\phi(Z|X)\|P(Z))\end{aligned}$$

这一项强制变分后验 q_ϕ 靠近先验 $P(Z)$（通常是标准正态分布），起到了正则化的作用，防止隐空间坍缩。

6.2.2 −ELBO 成为最终损失函数

综合以上推导，VAE 在最小化目标（Loss Function）时的表达式为：

$$\mathcal{J}(\phi,\theta) = -\mathcal{L}_{ELBO} = \underbrace{\mathbb{E}_{q_\phi(Z|X)}[-\log P_\theta(X|Z)]}_{\text{重建损失 (如交叉熵或 MSE)}} + \underbrace{D_{KL}(q_\phi(Z|X)\|P(Z))}_{\text{隐空间约束}} \tag{6-4}$$

6.3 神经网络实现 ELBO 设置的两个 Trick

6.3.1 第一个 Trick：重建项的重参数化表达

由于重建项 $\mathbb{E}_{q_\phi(Z|X)}[\log P_\theta(X|Z)]$ 中的期望算子依赖于待优化的推断网络参数 ϕ，直接求导会产生极大的方差。为了实现端到端的梯度传导，我们引入**重参数化技巧**。[8]

令隐变量 Z 表达为观测值 X 与独立噪声 ϵ 的确定性映射函数：

$$Z(\epsilon;\phi) = \mu_\phi(X) + \sigma_\phi(X)\odot\epsilon, \quad \text{其中} \quad \epsilon\sim\mathcal{N}(0,\mathbf{I})$$

此时，重建项可以改写为对噪声分布 ϵ 的期望，从而将随机性从参数 ϕ 中解耦出来：

$$\mathbb{E}_{q_\phi(Z|X)}[\log P_\theta(X|Z)] = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[\log P_\theta(X \mid \mu_\phi(X)+\sigma_\phi(X)\odot\epsilon)\right] \tag{6-5}$$

在该表达形式下，针对 ϕ 的优化目标梯度可以穿过采样层直接计算：

$$\nabla_\phi\mathcal{L}_{recon} \approx \frac{1}{L}\sum_{l=1}^{L}\nabla_\phi \log P_\theta(X \mid \mu_\phi(X)+\sigma_\phi(X)\odot\epsilon^{(l)})$$

其中 $\epsilon^{(l)}$ 是从标准正态分布中提取的第 l 个随机样本。

[8]重参数化技巧。由于上式中的期望算子 $\mathbb{E}_{q_\phi}$ 依赖于参数 ϕ，直接对 ϕ 求导会导致高方差。因此，我们必须引入**重参数化技巧**：令 $Z=\mu_\phi(X)+\sigma_\phi(X)\odot\epsilon$，其中 $\epsilon\sim\mathcal{N}(0,\mathbf{I})$。此时，梯度可以平滑地流经随机采样节点：

$$\nabla_\phi\mathbb{E}_{q_\phi(Z|X)}[f(Z)] = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I})}[\nabla_\phi f(\mu_\phi(X)+\sigma_\phi(X)\odot\epsilon)]$$

这使得整个 ELBO 可以通过标准的分批随机梯度下降（Minibatch SGD）进行端到端优化。

6.3.2 第二个 Trick：KL 散度的解析展开

假设隐变量 Z 的维度为 d。在标准 VAE 中，为了保证计算的解析性，通常对推断模型与先验模型做如下分布设定：

- **变分后验分布**：$q_\phi(Z|X) = \mathcal{N}(Z; \mu, \sigma^2\mathbf{I})$，其中 μ 和 σ^2 是由 Encoder 神经网络输出的参数向量。
- **先验分布**：$P(Z) = \mathcal{N}(Z; 0, \mathbf{I})$，即假设隐空间服从标准正态分布。

根据信息论中两个 d 维多元高斯分布 $p = \mathcal{N}(\mu_1, \Sigma_1)$ 与 $q = \mathcal{N}(\mu_2, \Sigma_2)$ 之间 KL 散度的通用计算公式：

$$D_{KL}(p \parallel q) = \frac{1}{2}\left[\mathrm{tr}(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T\Sigma_2^{-1}(\mu_2 - \mu_1) - d + \log\frac{|\Sigma_2|}{|\Sigma_1|}\right]$$

我们将 VAE 的具体参数代入上述公式，即令 $p = q_\phi$（$\mu_1 = \mu, \Sigma_1 = \Sigma$）且 $q = P(Z)$（$\mu_2 = 0, \Sigma_2 = \mathbf{I}$）。由于 Σ_2 为单位矩阵，其逆矩阵与行列式均为 1，公式可简化为：

$$D_{KL}(q_\phi(Z|X) \parallel P(Z)) = \frac{1}{2}\left[\mathrm{tr}(\Sigma_1) + \mu^T\mu - d - \log|\Sigma_1|\right]$$

由于在 VAE 中我们假设 Σ_1 是对角矩阵 $\mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2)$，矩阵的迹等于对角元素之和，行列式的对数等于对角元素对数的求和。据此，我们可以将上式展开为各维度分量相加的闭式解：

$$D_{KL}(q_\phi(Z|X) \parallel P(Z)) = \frac{1}{2}\sum_{j=1}^{d}\left(\mu_j^2 + \sigma_j^2 - \log(\sigma_j^2) - 1\right) \tag{6-6}$$

6.3.3 VAE 核心 Trick 的 PyTorch 实现

VAE 核心 Trick 的 PyTorch 实现

```python
import torch
import torch.nn.functional as F

def vae_loss_function(recon_x, x, mu, log_var):
    """
    recon_x: Decoder输出的重建值
    x: 原始观测值
    mu: Encoder输出的均值
    log_var: Encoder输出的对数方差 log(sigma^2)
    """

    # --- Trick 1: 重参数化采样 (Reparameterization) ---
    # Z = mu + sigma * epsilon
    std = torch.exp(0.5 * log_var) # sigma = exp(0.5 * log(sigma^2))
    eps = torch.randn_like(std) # 采样标准噪声 epsilon ~ N(0, I)
    z = mu + eps * std # 核心公式

    # --- 计算重建项 (以二元交叉熵为例) ---
    # 对应 E_q[log P(X|Z)]
    recon_loss = F.binary_cross_entropy(recon_x, x, reduction='sum')

    # --- Trick 2: KL 散度的解析展开 ---
    # 对应 0.5 * sum(mu^2 + sigma^2 - log(sigma^2) - 1)
    kl_div = -0.5 * torch.sum(1 + log_var - mu.pow(2) - log_var.exp())

    return recon_loss + kl_div
```

6.4 总结

不同于上章所提的混合推断模型，标准 VAE 假设所有隐变量均由全局映射函数（神经网络）控制。因此，其变分分布不再包含坐标更新项，而是一个纯粹的摊销分布 $q_\phi(Z|X)$ [cite: 29]。

其 ELBO 的表达式简化为：

$$\text{ELBO}(\phi,\theta) = \underbrace{E_{q_\phi(Z|X)}[\log P_\theta(X|Z)]}_{\text{重建似然}} - \underbrace{D_{KL}(q_\phi(Z|X)\|P(Z))}_{\text{正则化约束}}$$

为什么说 VAE 是摊销推断的极致？

我们可以从以下两个关键点看到 VAE 对推断范式的革新：

1. **完全取消局部参数**：在之前的 Z_2, Z_3 更新中，我们仍需维护坐标均值 μ_2, μ_3 。但在 VAE 中，对于数百万个样本，我们只需要存储一套全局参数 ϕ。任何新样本的后验分布都通过 $f_\phi(X)$ 瞬间计算得出]。

2. **重参数化梯度的普适化**：VAE 将上一章中提到的“重参数化处理”应用到了极致 [cite: 42, 74]。通过 $Z = \mu_\phi(X) + \sigma_\phi(X) \odot \epsilon$，它将复杂的随机推断问题彻底转化为了一个标准的、可利用自动微分工具（如 PyTorch/TensorFlow）求解的端到端神经网络优化问题。

总结：从 VAE 到分层模型

标准 VAE 是摊销推断的一个“极简点”。当我们理解了 VAE 如何利用 ϕ 处理单个隐变量后，就能更轻松地理解后面章节中“多模态基因表征对齐”的分层耦合结构——那本质上是多个 VAE 模块通过共享空间 S 进行协同推断的产物。