VAE & VQ-VAE

Tags

一.前置知识

1.自编码器（AE）

自编码器是一类用于无监督特征学习的神经网络模型，其核心思想是通过编码器将高维输入数据映射到低维潜在空间，并通过解码器从潜在空间重构输入。

（1）自编码器的数学定义

给定输入数据 $\mathbf{x} \in \mathbb{R}^n$，自编码器的目标是通过优化网络参数，使得输出数据 $\hat{\mathbf{x}}$与输入 $\mathbf{x}$尽可能接近。编码器和解码器可以表示为两个函数：

编码器

将输入 $\mathbf{x}$ 映射到潜在表示 $\mathbf{z}$：

$$\mathbf{z} = f_\theta(\mathbf{x}) = \sigma(\mathbf{W}_e\mathbf{x} + \mathbf{b}_e)$$

其中，$\mathbf{W}_e$ 和 $\mathbf{b}_e$ 分别是编码器的权重矩阵和偏置，σ 是激活函数。

解码器

$$\hat{\mathbf{x}} = g_\phi(\mathbf{z}) = \sigma'(\mathbf{W}_d\mathbf{z} + \mathbf{b}_d)$$

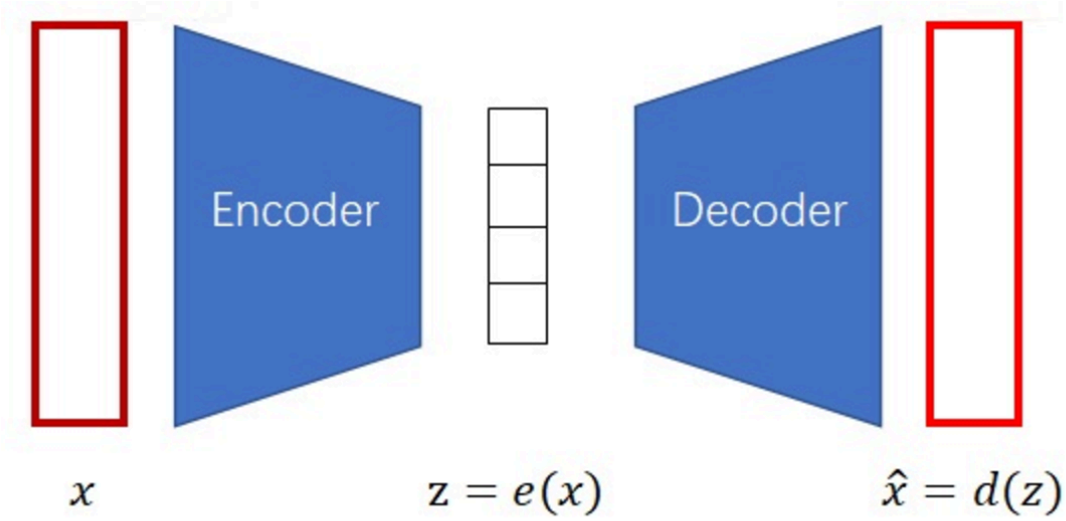
从潜在表示 $\mathbf{z}$重构输入数据 $\hat{\mathbf{x}}$

$\mathbf{W}_d$ 和 $\mathbf{b}_d$ 是解码器的权重矩阵和偏置，σ'是另一种激活函数。

损失函数

自编码器的训练目标是最小化重构误差，常使用以下损失函数：

$$L(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{x}_i - \hat{\mathbf{x}}_i||_2^2$$



$$z = argmin_z ||\hat{x} - x||^2$$

（2）用自编码器生成图像的原理

学习潜在空间分布

传统自编码器的潜在空间没有概率分布约束，这使得直接在潜在空间采样生成图像变得困难。因此，**改进模型**（如变分自编码器或对抗自编码器）通常假设潜在空间**服从某种分布**（如高斯分布），从而简化采样和生成过程。

采样与生成

采样：从潜在空间的分布中随机采样 $\mathbf{z}$，例如从$\mathcal{N}(\mathbf{0}, \mathbf{I})$采样。

生成：将采样的 $\mathbf{z}$ 输入解码器 g_ϕ，生成新图像：

$$\hat{\mathbf{x}} = g_\phi(\mathbf{z})$$

2.变分推断

在变分自编码器（VAE）中，变分推断的目标是通过优化一个变分下界（ELBO）来近似真实后验分布$p(\mathbf{z}|\mathbf{x})$。

贝叶斯推断的困难

假设有一个包含观测数据 $\mathbf{x}$和潜在变量 $\mathbf{z}$ 的模型，我们关心的是如何计算**后验分布** $p(\mathbf{z}|\mathbf{x})$：

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}$$

这里， $p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$是**证据**。计算这个积分非常复杂，因此，不能直接计算后验分布 $p(\mathbf{z}|\mathbf{x})$，必须用变分推断的方法进行近似。

变分推断的基本思想

变分推断的核心思想是，通过引入一个

变分分布 $q_\phi(\mathbf{z})$，来近似真实的后验分 $p(\mathbf{z}|\mathbf{x})$，常用的衡量距离的方式是**KL散度。**

$$D_{\mathrm{KL}}(q_\phi(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) = \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}\right]$$

变分下界（ELBO）的推导

计算KL散度需要知道$p(\mathbf{z}|\mathbf{x})$，因此KL散度无法直接计算，需通过推导变分下界（ELBO）来最小化近似误差。

从对数边际似然出发，对**证据 p(x)**取对数，**引入变分分布并进行重写**

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} = \log \int q_\phi(\mathbf{z}) \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z})} d\mathbf{z}$$

根据Jensen不等式有：

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z})}\right]$$

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z}) + \log p(\mathbf{z}) - \log q_\phi(\mathbf{z})\right]$$

将其简化为：

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z})\right] - D_{\mathrm{KL}}\left(q_\phi(\mathbf{z})||p(\mathbf{z})\right)$$

第一项是重构误差，表示给定潜在变量
z 后，数据x的重构质量；第二项是KL散度，表示近似后验
$q_\phi(\mathbf{z})$ 与先验分布 $p(\mathbf{z})$之间的差异。

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{q(z)}[\log p(x|z)] - D_{\mathrm{KL}}(q(z) \parallel p(z))$$

通过最大化变分下界最小化KL散度的原理

目标：最小化 KL 散度

$$q^*(z) = \arg\min_{q(z)} D_{\mathrm{KL}}(q(z) \parallel p(z|x))$$

$$D_{\mathrm{KL}}(q(z) \parallel p(z|x)) = \int q(z) \log \frac{q(z)}{p(z|x)}\, dz = \int q(z) \log \frac{q(z)}{p(z)p(x|z)/p(x)}\, dz$$

拆解为两部分：

$$D_{\mathrm{KL}}(q(z) \parallel p(z|x)) = \int q(z) \log \frac{q(z)}{p(z)p(x|z)}\, dz + \log p(x)$$

$\log p(x)$ 是观测数据的对数边际似然，与 q(z) 无关，是**一个常数**。

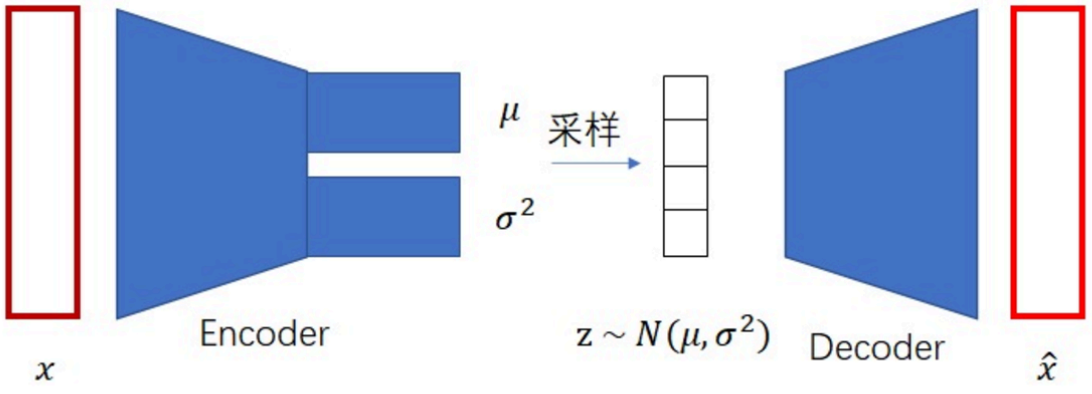
第一项是负的变分下界（ELBO）的定义。

因此：

$$D_{\mathrm{KL}}(q(z) \parallel p(z|x)) = -\mathcal{L}_{\mathrm{ELBO}} + \log p(x)$$

最大化变分下界 $\mathcal{L}_{\mathrm{ELBO}}$等价于最小化 KL 散度。

二. VAE:一种正则化的自编码器



loss: $||\hat{x} - x||^2 - sim(N(\mu, \sigma^2), N(0, I))$

VAE的核心思想

①**学习数据的潜在表示**：通过引入潜在变量的**概率模型**，将数据生成过程建模为一个**概率分布**。将高维数据 x 的复杂分布 p(x) 表示为隐变量 z 的分布 p(z) 与条件分布 p(x丨z) 的联合分布。

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

②生成新数据:从潜在空间z中采样，生成与训练数据相似的新样本

关键技巧

①变分推断：通过优化一个变分下界（ELBO），逼近真实后验分布 p(z丨x)，避免直接求解高维积分，同时保证编码器的输出分布合理。

②参数化技巧

采样操作本身不是可导的，因此在训练过程中会导致梯度无法传播，无法进行有效的反向传播。为了解决这个问题，VAE引入了 **重参数化技巧**，将潜在变量 z 表示为：

$$\mu(x) + \sigma(x) \cdot \epsilon$$

其中, μ(x) 和 σ(x) 是编码器输出的均值和标准差，ϵ是从标准正态分布 $\mathcal{N}(0, I)$中采样的噪声项。通过这个技巧，潜在变量的采样过程变得可微，允许使用反向传播来优化模型。

VAE模型结构

编码器：将数据映射到潜在空间。

解码器：从潜在变量生成数据。

损失函数

VAE的损失函数是 **证据下界（ELBO）**，其目的是在潜在空间中找到既能 **精确重建数据**，又能保持潜在空间结构 **简单和有规律** 的方法。

ELBO包含两部分：

$$\mathcal{L}_{VAE}(x) = \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z))$$

第一项重构损失：

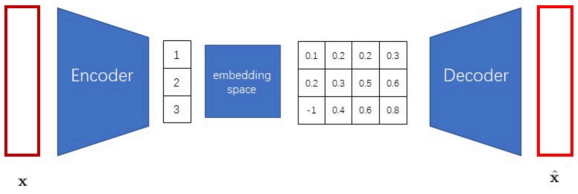
> 编码器输出潜在变量分布后，解码器根据这些潜在变量z 生成数据x 的概率对数。就是生成的x 和真实的x 之间的相似度。

第二项KL散度：

编码器学习到的潜在变量分布 q(z | x) 和 **潜在变量的先验分布**之间的差异。

这样做的目的是让编码器学习的潜在空间具有良好的结构，使得在潜在空间中采样时生成的新数据是合理的。

三.VQ-VAE



VQVAE

VQ-VAE的核心思想

通过引入离散的潜在变量（向量量化），使得潜在空间不再是连续的高维变量，而是由有限数量的离散向量（来自一个代码本）组成。能够更好地捕捉和利用数据的离散结构。

关键技巧

停止梯度

在训练过程中，由于向量量化是一个不可导操作，因此VQ-VAE引入了**停止梯度**操作。同时VQ-VAE使用代码本的**软更新**，即通过计算梯度来更新代码本中的离散向量，而不是直接通过量化过程更新编码器的输出。

模型结构

编码器

将输入数据 $\mathbf{x}$映射到一个连续的潜在空间表示 $\mathbf{z}_{\text{cont}}$

向量量化层

VQ-VAE对编码器输出的连续潜在向量 $\mathbf{z}_{\text{cont}}$ 进行离散化。将连续潜在向量 $\mathbf{z}_{\text{cont}}$映射到一个预定义的**代码本**codebook中的离散向量里。这个代码本包含 K 个嵌入向量，每个嵌入向量 具有与 $\mathbf{z}_{\text{cont}}$相同的维度。

公式表示为：

$$\mathbf{z}_{\text{discrete}} = \text{Quantize}(\mathbf{z}_{\text{cont}}) = \mathbf{c}_k, \quad \text{where} \quad k = \arg\min_j \|\mathbf{z}_{\text{cont}} - \mathbf{c}_j\|$$

解码器

将离散的潜在空间映射回原始数据空间

字典

字典中每个向量 c_i代表了潜在空间中的一个“离散”元素

损失函数

重构损失：

理想的目标是：

$$\mathcal{L}_{\text{reconstruction}} = \|x - \text{decoder}(z_q)\|_2^2$$

其中 z_q 是量化后的潜在向量，它是通过最小化连续潜在向量 z'和离散代码簿向量 c_i 之间的距离来得到的

问题在于，**argmin这个操作是没梯度的**

> VQ-VAE使用了一种叫做”straight-through estimator”的技术来完成梯度复制。这种技术是说，前向传播和反向传播的计算可以不对应。你可以为一个运算随意设计求梯度的方法。基于这一技术，VQ-VAE使用了一种叫做sg(stop gradient，停止梯度)的运算：
>
> $$sg(x) = \begin{cases} x \ (in\ forward\ propagation) \\ 0 \ (in\ backward\ propagation) \end{cases}$$

基于这种运算，我们可以设计一个把梯度从$z_e(x)$复制到$z_q(x)$的误差：

$$L_{reconstruct} = ||x - decoder(z_e(x) + sg(z_q(x) - z_e(x)))||_2^2$$

量化误差：

连续潜在表示 z′ 与量化后的离散向量 z 之间的差异。

$$\mathcal{L}_{\text{quantization}} = \|\text{sg}(z') - z\|^2$$

其中，sg() 表示停止梯度操作，避免对量化过程进行直接反向传播

字典更新：

字典更新损失用来优化字典中的潜在向量，使得它们更好地表示数据的潜在结构。

$$\mathcal{L}_{\text{codebook}} = \|z - c_i\|^2$$

VQ-VAE的总体损失函数是：

$$\begin{aligned} L =& ||x - decoder(z_e(x) + sg(z_q(x) - z_e(x)))||_2^2 \\ &+ \alpha||sg(z_e(x)) - z_q(x)||_2^2 + \beta||z_e(x) - sg(z_q(x))||_2^2 \end{aligned}$$

VQ-VAE如何生成图像

①训练VQ-VAE的编码器和解码器，使得VQ-VAE能把图像变成「小图像」，也能把「小图像」变回图像。

②训练**PixelCNN**，让它学习怎么生成「小图像」，拟合潜在空间

③随机采样时，先用PixelCNN采样出「小图像」，再用VQ-VAE把「小图像」翻译成最终的生成图像