Variational Autoencoder

xinli*

August 9, 2024

Abstract

这里记录了我在学习 VAE 过程中的一些思考和记录, 帮助我更好的理解 VAE。(仅供参考)

1 Introduction

在学习 VAE 的过程中，常常会思考几个问题。1、原始的 AE 的流程以及它的缺陷是什么? 2、VAE 相对于 AE 来说，它的改进点是什么? 3、VAE 中的数学推理以及它最终的 loss 是什么? 4、VAE 还有哪些缺点? 接下来让我们带着这些问题来开始学习吧。

2 Solve the problem

2.1 AE 的介绍

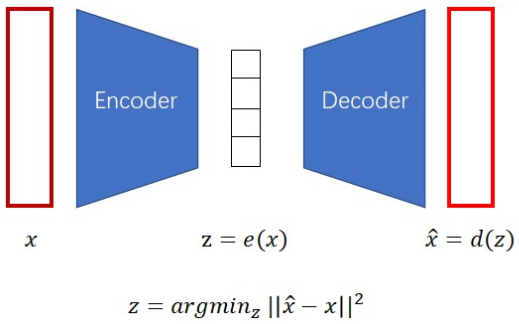

Figure 1: autoencoder architecture

如图(1)所示，AE 首先是一个编码器-解码器的结构, 其中编码器用于将输入数据 x 映射成一个低维潜在表示 z。接下来解码器负责将 z 进行重建, 它的目标是重建成 x。AE 的 loss 函数是原始输入 x 和重建的输出 x1 之间的差距, 我们可以使用 MSE-loss 来表示。

在 AE 中, 它的潜在变量是一个具体的值, 可以是 1, 2, 3 这些。但是，这样的编码结构就会带来一个问题, 如图(2)所示，即使模型的 encoder 和 decoder 足够复杂, 它可以对输入的数据进行很好的拟合, 但是假如我们输入了 1.5 这个数, 它的输出结果并不符合我们的预期。也就是说, AE 的这种编码方式, 让编码本身就失去了语义信息, 编码之间的插值并不能表示图像语义之间的插值了。

*https://github.com/xinli2008

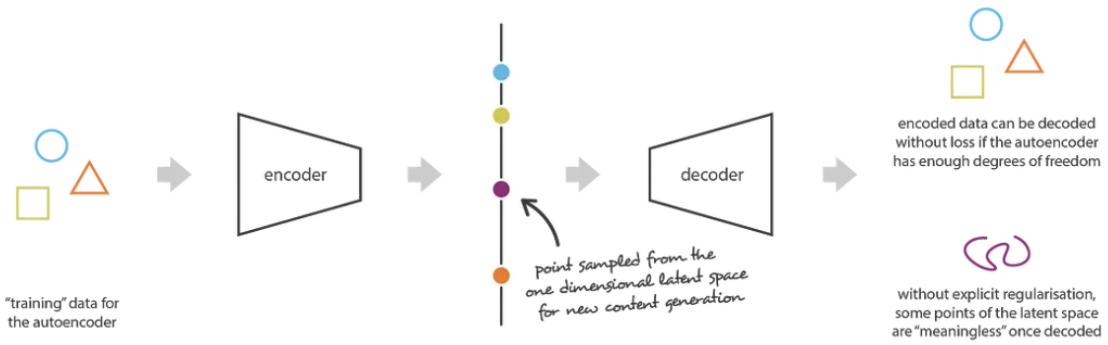

Figure 2: autoencoder shortcoming

2.2 引出 VAE

在上一小节，我们介绍了 AE 的不足。即使我们设计了参数量足够多的 encoder 和 decoder 结构，并且模型的收敛效果足够好时，AE 模型的泛化性还是不行的，或者说，AE 模型已经产生了过拟合的现象。那么，VAE 是如何解决这个问题的呢？VAE 相对于 AE，它的改进点并不是在模型结构上，更多关注的是**潜在变量的概率建模**以及**正则化和潜在空间的连续性**。

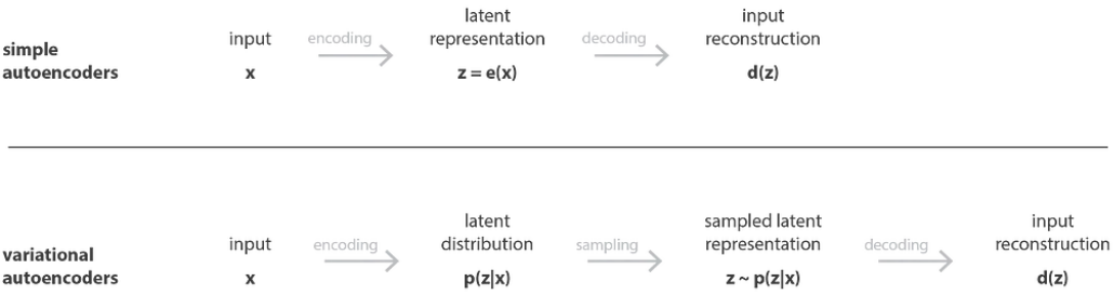

Figure 3: VAE architecture

对于第一点，我的理解是：在 AE 中，潜在变量 Z 是一个具体的值，而在 VAE 中，潜在变量 z 是被建模成了一个概率分布。编码器并不是直接输出潜在变量，而是尝试输出一个分布（具体体现为均值和方差），然后我们可以从这个分布中进行采样，生成新的潜在变量。

实例：假设我们在处理手写数字的图像。AE 的编码器可能会将一个数字“3”映射到潜在空间中的一个具体点，**而 VAE 的编码器会将“3”映射到一个高斯分布，表示为均值 μ 和 σ^2。从这个分布中采样，我们可以得到不同的“3”，这些“3”可能会有细微的变化，比如不同的笔画粗细或角度。这种概率建模使得生成的图像更加多样化和灵活**。

对于第二点，我的理解是：在 VAE 中，它引入了 KL 散度，VAE 强制潜在变量的分布接近标准正态分布。这使得潜在空间更加平滑，此时潜在变量之间的插值也更加的合理。

实例：继续上面的手写数字例子，如果我们想在潜在空间中从“3”插值到“8”，AE 可能会产生奇怪的、不合理的中间形状，因为它们的潜在表示是离散的具体点。而在 VAE 中，由于潜在变量被正则化成标准正态分布，插值路径上的所有点都对应于合理的数字形状。这样，我们可以平滑地生成从“3”到“8”的连续过渡图像，这在图像生成和数据增强中非常有用。

2.3 preliminary

我们先介绍一些预备的数学知识，以便于我们接下来更好的理解。

2.3.1 凹函数、凸函数、Jensen 不等式

我们可以通过二阶导数来有效的判断一个函数在某个区间上是凹函数还是凸函数：如果 $f''(x) \geq 0$ 在区间上成立，则 $f(x)$ 是凸函数。相反，如果 $f''(x) \leq 0$，则 $f(x)$ 是凸函数。

Jensen 不等式是关于凸函数和凹函数的一个重要不等式。对于一个凸函数 f 和一个随机变量 X, 不等式表示如下：

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)] \tag{1}$$

对于一个凹函数 f 和一个随机变量 X, 不等式表示如下：

$$f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)] \tag{2}$$

在 VAE 中，我们引用该不等式来处理对数期望：

$$\log \mathbb{E}_{q(z|x)}\left[\frac{p(x,z)}{q(z \mid x)}\right] \geq \mathbb{E}_{q(z|x)}\left[\log \frac{p(x,z)}{q(z \mid x)}\right] \tag{3}$$

2.3.2 先验分布和后验分布

先验分布表示在观察到任何数据之前，我们对某个参数的信念和知识。它是根据先验知识来确认的。数学上，如果我们有一个参数 z，先验分布可以表示为 $p(z)$。它表示的是我们在没有看到数据 D 的情况下，对参数 z 的初始相信程度。

而后验分布是观察到数据 X 之后，对参数 z 的不确定的新描述。它结合了先验分布和观测数据的似然函数，根据贝叶斯公式计算得到。贝叶斯定理的公式为：

$$P(z \mid x) = \frac{P(x \mid z)P(z)}{P(x)} \tag{4}$$

其中：$p(z \mid x)$ 是后验分布，表示在给定数据 X 后，对参数 z 的相信程度。$p(x \mid z)$ 是似然函数，表示在参数 z 下，数据 X 出现的概率。$p(z)$ 是先验分布。$p(x)$ 是边际似然，计算为所有可能参数值的先验分布的积分。

2.3.3 KL 散度

KL 散度经常用于衡量两个概率分布 $p_1(x)$ 和 $p_2(x)$ 之间的差异。它的定义如下：

$$KL\left(p_1(x) \| p_2(x)\right) = \int p_1(x) \log\left(\frac{p_1(x)}{p_2(x)}\right) dx \tag{5}$$

$$= \int p_1(x) \log p_1(x) dx - \int p_1(x) \log p_2(x) dx \tag{6}$$

$$= -\int p_1(x) \log p_2(x) dx - \left(-\int p_1(x) \log p_1(x) dx\right) \tag{7}$$

设 $p(z)=\mathcal{N}\left(z;\mu,\sigma^2\right)$ 和 $q(z)=\mathcal{N}\left(z;0;1\right)$, 他们的概率函数分别为：

$$p(z)=\frac{1}{\left(2\pi\sigma^2\right)^{J/2}}\exp\left(-\frac{1}{2}\sum_{j=1}^{J}\frac{\left(z_j-\mu_j\right)^2}{\sigma_j^2}\right) \tag{8}$$

$$q(z)=\frac{1}{(2\pi)^{J/2}}\exp\left(-\frac{1}{2}\sum_{j=1}^{J}z_j^2\right) \tag{9}$$

其中 J 是潜在变量 z 的维度。接下来我们尝试将两个概率密度函数带入公式(5)中，

$$\log\frac{p(z)}{q(z)}=\log\left(\frac{\frac{1}{\left(2\pi\sigma\sigma^2\right)^{J/2}}\exp\left(-\frac{1}{2}\sum_{j=1}^{J}\frac{\left(z_j-\mu_j\right)^2}{\sigma_j^2}\right)}{\frac{1}{(2\pi)^{J/2}}\exp\left(-\frac{1}{2}\sum_{j=1}^{J}z_j^2\right)}\right) \tag{10}$$

将分子分母分别计算：

$$\log p(z)=-\frac{J}{2}\log(2\pi)-\frac{1}{2}\sum_{j=1}^{J}\log\sigma_j^2-\frac{1}{2}\sum_{j=1}^{J}\frac{\left(z_j-\mu_j\right)^2}{\sigma_j^2} \tag{11}$$

$$\log q(z)=-\frac{J}{2}\log(2\pi)-\frac{1}{2}\sum_{j=1}^{J}z_j^2 \tag{12}$$

所以，

$$\log\frac{p(z)}{q(z)}=-\frac{1}{2}\sum_{j=1}^{J}\log\sigma_j^2-\frac{1}{2}\sum_{j=1}^{J}\frac{\left(z_j-\mu_j\right)^2}{\sigma_j^2}+\frac{1}{2}\sum_{j=1}^{J}z_j^2 \tag{13}$$

我们将公式(13)带入公式(5)中，可以得到：

$$\mathrm{KL}(p\|q)=\int\mathcal{N}\left(z;\mu,\sigma^2\right)\left(-\frac{1}{2}\sum_{j=1}^{J}\log\sigma_j^2-\frac{1}{2}\sum_{j=1}^{J}\frac{\left(z_j-\mu_j\right)^2}{\sigma_j^2}+\frac{1}{2}\sum_{j=1}^{J}z_j^2\right)dz \tag{14}$$

由于对高斯分布积分的性质，我们可以分别计算这三个部分的期望值：

$$\mathbb{E}_{\mathcal{N}(z;\mu,\sigma^2)}\left[-\frac{1}{2}\sum_{j=1}^{J}\log\sigma_j^2\right]=-\frac{1}{2}\sum_{j=1}^{J}\log\sigma_j^2 \tag{15}$$

$$\mathbb{E}_{\mathcal{N}(z;\mu,\sigma^2)}\left[-\frac{1}{2}\sum_{j=1}^{J}\frac{(z_j-\mu_j)^2}{\sigma_j^2}\right]=-\frac{1}{2}\sum_{j=1}^{J}1=-\frac{J}{2} \tag{16}$$

$$\mathbb{E}_{\mathcal{N}(z;\mu,\sigma^2)}\left[\frac{1}{2}\sum_{j=1}^{J}z_j^2\right]=\frac{1}{2}\sum_{j=1}^{J}(\mu_j^2+\sigma_j^2) \tag{17}$$

将这些部分结合在一起，我们可以得到：

$$\begin{aligned}\mathrm{KL}(p\|q)&=-\frac{1}{2}\sum_{j=1}^{J}\log\sigma_j^2-\frac{J}{2}+\frac{1}{2}\sum_{j=1}^{J}\left(\mu_j^2+\sigma_j^2\right)\\&=\frac{1}{2}\sum_{j=1}^{J}\left(\mu_j^2+\sigma_j^2-\log\sigma_j^2-1\right)\end{aligned} \tag{18}$$

2.3.4 变分推断

变分推导是一种用于近似复杂概率分布的方法，尤其在贝叶斯推断中用于近似后验分布。其核心思想是用一个简单的分布去近似复杂的后验分布，通过优化使得两者尽可能接近。

我们可以这样理解变分推断：我们现在有一个复杂的后验分布 $p(z|x)$，这在大多数情况下难以直接计算。从我们的认知中，给定一堆样本 X，我们想要直接根据这群样本推断出它的隐变量 z 的分布，这是比较困难的。**变分推导是引入一个参数化的简单分布 $q(z \mid x)$ 来近似后验分布 $p(z \mid x)$**。

2.3.5 联合概率分布

两个随机变量 x 和 z 的联合概率分布，记作 $p(x, z)$，可以表示为条件概率 $p(z \mid z)$ 和边缘概率 $p(z)$ 的乘积。

$$p(x, z) = p(x|z)p(z) \tag{19}$$

2.3.6 重参数化技巧

在 VAE 中，我们需要从近似后验分布 $q(z \mid x)$ 中采样潜在变量 z，这个分布通常被假设为高斯分布 $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$。直接从这个分布中采样会使得采样结果不可微，从而无法通过标准的反向传播来训练模型。

重参数化技巧通过将采样过程重参数化为可微的操作来解决这个问题。具体来说，我们可以将从高斯分布 $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ 中的采样过程表示为：

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon} \tag{20}$$

其中：$\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$ 是从编码器中得到的均值和标准差。$\boldsymbol{\epsilon}$ 是从标准正态分布 $\mathcal{N}(0, 1)$ 中采样的随机变量。$\odot$ 表示元素逐元素相乘。

通过这个重参数化，采样过程变得可微，因为 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$ 是神经网络的输出，它们的梯度可以通过反向传播计算得到。

2.4 VAE 的思考流程

Variational Autoencoder（VAE）的训练目标是最大化数据的边际似然 P（X）。当我们知道了数据 X 的分布，我们就可以造出无穷无尽的样本了。

$$P(X) = \int P(X \mid z)P(z)dz \tag{21}$$

但是由于这个积分在高维空间中很难直接计算，所以 VAE 采用了引入潜在变量 z 的方法来简化这个问题。VAE 通过引入隐变量 z，并假设数据 X 是在潜在变量 z 生成的。解码器通过将输入数据 X 映射到潜在空间中，得到数据分布 $p(z|x)$，而解码器通过从潜在空间中采样潜在变量 z，并将其还原成原始数据 X，计算似然分布 $p(x \mid z)$。**第一点，在 VAE 中，通过引入了后验分布 $p(z \mid x)$ 来解决无法直接计算 $p(x)$ 的问题**。

但是，直接计算后验分布 $p(z \mid x)$ 同样比较困难，所以 VAE 引入了变分推断来解决这个问题，也就是说用神经网络拟合出一个简单的分布 $Q(z \mid x)$，通过最小化这两个分布之间的差异（通常使用 KL 散度）使得 $Q(z \mid x)$ 尽可能接近 $p(z \mid x)$。**第二点，在 VAE 中，通过引入了变分推断的方法来解决无法直接计算后验分布 $p(z \mid x)$ 的问题**。

接下来，我们通过逐步推导出 VAE 的损失函数。

我们希望最大化边际似然 $\log p(x)$，通过引入潜在变量 z:

$$\log p(x) = \log \int p(x, z) dz = \log \int p(x \mid z) p(z) dz \tag{22}$$

然后我们通过变分推断，引入了 $Q(z \mid x)$ 后，我们可以将边缘似然重写为:

$$\log p(x) = \log \int q(z \mid x) \frac{p(x, z)}{q(z \mid x)} dz \tag{23}$$

应用 Jensen 不等式，得到:

$$\log p(x) \geq \int q(z \mid x) \log \frac{p(x, z)}{q(z \mid x)} dz \tag{24}$$

而由于联合分布 $p(x, z) = p(x \mid z)p(z)$，则上面的公式等于:

$$\log p(\mathbf{x}) \geq \int q(\mathbf{z} \mid \mathbf{x}) \log \frac{p(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})} d\mathbf{z} \tag{25}$$

接下来在 log 内的表达式拆开:

$$\log p(\mathbf{x}) \geq \int q(\mathbf{z} \mid \mathbf{x}) (\log p(\mathbf{x} \mid \mathbf{z}) + \log p(\mathbf{z}) - \log q(\mathbf{z} \mid \mathbf{x})) d\mathbf{z} \tag{26}$$

所以，我们现在得到了:

$$\log p(\mathbf{x}) \geq \int q(\mathbf{z} \mid \mathbf{x}) \log p(\mathbf{x} \mid \mathbf{z}) d\mathbf{z} + \int q(\mathbf{z} \mid \mathbf{x}) \log p(\mathbf{z}) d\mathbf{z} - \int q(\mathbf{z} \mid \mathbf{x}) \log q(\mathbf{z} \mid \mathbf{x}) d\mathbf{z} \tag{27}$$

VAE 的损失函数由重构误差和 KL 散度组成，前者衡量重构质量，后者确保潜在空间的连续性和平滑性。**第一项是重构误差**，即 $\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x} \mid \mathbf{z})]$，重构误差是指输入数据与经过编码器和解码器重构后的数据之间的差异。**第二项和第三项合在一起就是 KL 散度**，我们可以通过 KL 散度的公式来验证:

$$\begin{aligned} \mathrm{KL}(q(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})) &= \int q(\mathbf{z} \mid \mathbf{x}) \log \frac{q(\mathbf{z} \mid \mathbf{x})}{p(\mathbf{z})} d\mathbf{z} \\ &= \int q(\mathbf{z} \mid \mathbf{x}) \log q(\mathbf{z} \mid \mathbf{x}) d\mathbf{z} - \int q(\mathbf{z} \mid \mathbf{x}) \log p(\mathbf{z}) d\mathbf{z} \end{aligned} \tag{28}$$

为了要让 $p(x)$ 最大，我们需要让右面的数最大，则我们可以将 VAE 的 loss 定义为:

$$\mathcal{L}_{VAE} = -\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x} \mid \mathbf{z})] + \mathrm{KL}(q(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})) \tag{29}$$

由于公式中的 $p(z)$ 代表着隐变量 z 的先验分布，我们通常认为它是高斯分布。则我们按照之前推理的 KL 散度(18)的公式，将它简化为:

$$\mathcal{L}_{VAE} = -\mathbb{E}_{q_\phi(z|x)} \left[\underbrace{\log p_\theta(x|z)}_{\text{how good your decoder is}} \right] + \underbrace{\mathcal{D}_{\mathrm{KL}} \left(\underbrace{q_\phi(z|x)}_{\text{a Gaussian}} \parallel \underbrace{p(z)}_{\text{a Gaussian}} \right)}_{\text{how good your encoder is}} \tag{30}$$

$$\mathcal{L}_{VAE} = -\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x} \mid \mathbf{z})] + \frac{1}{2} \sum_{j=1}^{d} \left(\sigma_j^2(\mathbf{x}) + \mu_j^2(\mathbf{x}) - 1 - \log \sigma_j^2(\mathbf{x}) \right) \tag{31}$$