VAE

理论部分

从直觉方面看VAE

主要参考资料：BV1sM4y1W7jx

直觉上，作者想做的事情就是将原本仅仅是一个x和一个z，再从z到y这种映射的关系，得到一个均匀平滑的z。让z的每一个值都能得到一个有意义的y，也就是最终得到$p(x|z)$这个过程，这也是生成模型的本质，即将用一个均匀平滑的分布映射到真实数据的分布。为了得到均匀平滑的z，作者假设z服从高斯分布或者伯努利分布，伯努利分布和高斯分布具体的差异在于，伯努利分布的z的数据是二值的，也就是生成的值只能是二值的，也就是只能适用于Mnist这种数据集，而高斯分布什么数据集都可以使用。

LOSS推导

当假设z服从高斯分布后，作者就需要得到$p(z|x)$这个将真实数据转换为隐空间变量的过程。对于这个过程而言如果直接算$p(z|x)=\frac{p(x|z)p(z)}{p(x)}=\frac{p(x|z)p(z)}{\int p(x)\,dx}$在这里由于分母部分的积分是真实世界的数据集，永远不可能求出值。因此我能只能用逼近的方法。因此为了分布，这里就需要引入KL散度。

假设我们有个近似的后验分布$q_\theta(z|x)$，其中θ代表神经网络中的参数，因此我们用$D_{KL}[q_\theta(z|x)||p(z|x)]$来衡量近似分布和真实分布之间的距离。

我们展开这个式子可以得到

$$D_{KL}[q_\theta(z|x)||p(z|x)]=\mathbb{E}_{z\sim q_\theta(z|x)}[log\frac{q_\theta(z|x)}{p(z|x)}]=\mathbb{E}_{z\sim q_\theta(z|x)}[q_\theta(z|x)-logp(z|x)]=\mathbb{E}_{z\sim q_\theta(z|x)}[q_\theta(z|x)-logp(x|z)-logp(z)+logp(x)]$$

其中$logp(x)$与期望变量z无关所以可以写成

$$=\mathbb{E}_{z\sim q_\theta(z|x)}[q_\theta(z|x)-logp(x|z)-logp(z)]+logp(x)$$

进一步整理，前后项进行调整，可以得到最终的形式

$$logp(x)=D_{KL}[q_\theta(z|x)||p(z|x)]+\mathbb{E}_{z\sim q_\theta(z|x)}[logp(x|z)+logp(z)-q_\theta(z|x)]$$

在这里x是一个固定的值，表示数据集中的图片，所以$logp(x)$我们可以看做是一个常量

所以为了最小化$D_{KL}[q_\theta(z|x)||p(z|x)]$，我们可以最大化$\mathbb{E}_{z\sim q_\theta(z|x)}[logp(x|z)+logp(z)-q_\theta(z|x)]$

$\mathbb{E}_{z\sim q_\theta(z|x)}[logp(x|z)+logp(z)-q_\theta(z|x)]$也在变分推断里面被称为Evidence Lower Bound，也就是ELBO

令$\mathcal{L}=\mathbb{E}_{z\sim q_\theta(z|x)}[logp(x|z)+logp(z)-q_\theta(z|x)]=\mathbb{E}_{z\sim q_\theta(z|x)}[logp(x|z)-log\frac{q_\theta(z|x)}{p(z)}]=\mathbb{E}_{z\sim q_\theta(z|x)}[logp(x|z)]-D_{KL}[q_\theta(z|x)||p(z)]$

最终我们得到了$\mathcal{L}=\mathbb{E}_{z\sim q_\theta(z|x)}[logp(x|z)]-D_{KL}[q_\theta(z|x)||p(z)]$

Encoder部分

Encoder对应的是$q_\theta(z|x)$部分，即将一个数据集中的图片x，通过网络得到一个分布，这个分布与我们最开始设置z的分布趋近一致，因此$q_\theta(z|x)$同样也是一个标准的正态分布或者对二值数据集而言的伯努利分布。

Decoder部分

对应的网络是$p(x|z)$部分，因为高斯分布（或正态分布）在自然界和许多领域（包括图像和声音等）中都非常常见，且许多独立随机过程的总和往往会接近高斯分布（这是由于中心极限定理的结果）。因此，假设 $p(x|z)$服从高斯分布是一个通常的、实用的假设。所以我们认为$p(x|z)$同样服从高斯分布。

从联合概率密度看VAE

主要参考资料：苏剑林：变分自编码器(一)（二）

VAE本质上就是AE引入了高斯噪声作为中间变量，从而能够产生生成的效果。

VAE主要是想利用一个简单的分布$p(z)$，$p(z)$服从与高斯分布。从$p(z)$采样的点能够生成原始的分布$p(x)$，也就是得到$q(x|z)$这个这个条件分布。

联合概率密度 $p(x,z)$ 捕捉了数据 x 和潜在变量 z 的整个生成过程。这意味着我们考虑了从潜在空间到数据空间的所有可能路径以及它们发生的概率。

通过 $p(x,z)$ 我们可以直接讨论和建模潜在变量 z 如何生成数据 x 的问题，同时也能反向推断出给定数据 x 时潜在变量 z 的分布。这涵盖了整个模型的编码（推断）和解码（生成）过程。

LOSS推导

所以可以从联合概率分布开始推导，我们从真实的联合概率密度分$p(x,z)$和近似的联合概率密度分布$q(x,z)$的KL散度入手。用$q(x,z)$来近似$p(x,z)$

从而优化这个KL散度。对应KL散度，我们进行变化，可以让其得到KL散度为$\mathbb{E}_{p(x)}\left[\int p(z|x)\ln\left(\frac{\tilde{p}(x)p(z|x)}{q(x,z)}\right)dz\right]$，继续将其拆解可以发现其中的$\mathbb{E}_{x\sim p(x)}\left[\int p(z|x)\ln\tilde{p}(x)p(z|x)\,dz\right]=\mathbb{E}_{x\sim p(x)}\left[\ln\tilde{p}(x)\int p(z|x)\,dz\right]=\mathbb{E}_{x\sim p(x)}\left[\ln\tilde{p}(x)\right]$是一个常数

因此我们可以得到**我们所需要的损失**
$L=KL散度-常数=\mathbb{E}_{x\sim p(x)}\left[\int p(z|x)\ln\left(\frac{p(z|x)}{q(x,z)}\right)dz\right]=\mathbb{E}_{x\sim p(x)}\left[\mathbb{E}_{z\sim p(z|x)}\left[-\ln q(x|z)\right]+KL\left(p(z|x)\parallel q(z)\right)\right]$

这时可以看见L由两个部分组成，左边为$q(x|z)$ 也就是decoder的部分，右边为$p(z|x)$也需要进一步的近似，得到$q(z|x)$为encoder的部分。

可以看到 当我们优化这个L的时候两部分的Loss会相互抵抗，达成一个平衡效果。

在这个L中，我们对$q(x|z),p(z|x),q(z)$未知

Encoder部分

如果我们假设z服从高斯分布，那么$q(z)$就知道了

对于$p(z|x)$仍然需要近似，近似$p(z|x)$的方法同理与近似$p(x,z)$的方法，都是用KL散度进行近似，可以得到一个

$\hat{p}(z|x)=q(z|x)=\frac{q(z|x)q(z)}{q(x)}=\frac{q(z|x)q(z)}{\int q(z|x)q(z)dz}$，但是分母上的积分是不可能的事情

所以我们直接假设$p(z|x)$服从正态分布，均值和方差由神经网络得到$p(z|x)=\frac{1}{\prod_{k=1}^{D}\sqrt{2\pi\sigma_{(k)}^2(x)}}\exp\left(-\frac{1}{2}\left\|\frac{z-\mu(x)}{\sigma(x)}\right\|^2\right)$

其中d是分量的维度，并且之所以指数函数不需要对各分量做累积的原因是，指数内部的范数平方已经包含了累积的步骤
$\left\|\frac{z-\mu(x)}{\sigma(x)}\right\|^2=\sum_{k=1}^{d}\left(\frac{z_k-\mu_k(x)}{\sigma_k(x)}\right)^2$

那么最终的得到的**Encoder的部分的损失**$KL\left(p(z|x)\parallel q(z)\right)=\frac{1}{2}\sum_{k=1}^{d}\left(\mu_k^2(x)+\sigma_k^2(x)-\ln\sigma_k^2(x)-1\right)$

Decoder部分

对于Decoder，我们可以直接同样用认为$q(x|z)$服从高斯分布

这样$q(x|z)=\frac{1}{\prod_{k=1}^{D}\sqrt{2\pi\sigma_{(k)}^2(z)}}\exp\left(-\frac{1}{2}\left\|\frac{x-\tilde{\mu}(z)}{\tilde{\sigma}(z)}\right\|^2\right)$

$-\ln q(x|z)=\frac{1}{2}\left\|\frac{x-\tilde{\mu}(z)}{\tilde{\sigma}(z)}\right\|^2+\frac{D}{2}\ln 2\pi+\frac{1}{2}\sum_{k=1}^{D}\ln\tilde{\sigma}_{(k)}^2(z)$

这里的$\tilde{\sigma}(z)$和$\tilde{\mu}(z)$都是decoder网络计算出的参数

如果我们假设$\tilde{\sigma}$为常数的话，那么可以接着等价于优化以下的公式

$-\ln q(x|z)\sim\frac{1}{2\tilde{\sigma}^2}\|x-\tilde{\mu}(z)\|^2$

采样技巧

在VAE中我们神经网络中的一个batch的每一个x都从$p(z|x)$中采样一个专属于这个x的z，然后接着用这个z去计算$-\ln q(x|z)$。由于只采样了一个样本，因此我们内部的期望可以去掉。最终得到的损失函数如下：
$L=\mathbb{E}_{x\sim p(x)}\left[-\ln q(x|z)+KL\left(p(z|x)\parallel q(z)\right)\right]$

网络构建

LOSS推导

在之前的推导中，我们已经得到了Loss为$\mathcal{L}=\mathbb{E}_{z\sim q_\theta(z|x)}[logp(x|z)]-D_{KL}[q_\theta(z|x)||p(z)]$

由于我们是用的近似的方法，所以在这里$p(x|z)$和$q_\theta(z|x)$都是神经网络

在这里$\log p(x|z)=\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}\right)$

如果我们假设$\sigma=1$，那么这个公式就可以简化为：

$\log p(x|z)=-\frac{(x-\mu)^2}{2}+\text{const}$

其中，"const"是不依赖于x或μ的常数项。

因此，我们可以将$\mathbb{E}_{z\sim q_\theta(z|x)}[\log p(x|z)]$的优化问题转化为最小化MSE的问题，具体的公式为：

$\mathbb{E}_{z\sim q_\theta(z|x)}[\log p(x|z)] \approx \mathbb{E}_{z\sim q_\theta(z|x)}[-\frac{1}{2}(x-\mu)^2]$

同样的，对于$D_{KL}[q_\theta(z|x)||p(z)]$$而言有

$D_{KL}[q_\theta(z|x)||p(z)] = \frac{1}{2}\sum_i\left(\mu_i^2+\sigma_i^2-\log(\sigma_i^2)-1\right)$

Encoder部分

对于Encoder部分，我们这里需要让原始数据x映射到多种z上从而引入随机性，所以所以这个网络应该输出的是高斯分布的均值和方差，并最终利用重采样技术得到网络输出的分布$p(z|x)$，所以我们需要构建一个神经网络，该网络输出的是均值和方差。在这个部分，输出的值会用于计算，这部分的损失为KL散度$D_{KL}[q_\theta(z|x)||p(z)] = \frac{1}{2}\sum_i\left(\mu_i^2+\sigma_i^2-\log(\sigma_i^2)-1\right)$

Decoder部分

这里我们同样需要需要输出值，在理论部分中，我们认为这个网络输出的仍然是一个高斯分布的参数，但是在实际过程中我们通常在编码器的输出端进行采样，这样就可以引入所需的随机性，使得我们的模型能够生成多样性的输出。然后，我们使用这些采样的潜在变量 z 来重构输入图像。

然而，在解码器的输出端，我们通常不进行采样，而是直接使用解码器输出的均值向量作为重构图像，这主要有两个原因：

1. **计算方便**：进行采样会引入额外的随机性和复杂性，这可能会使得训练过程更加困难。而直接使用均值向量可以简化计算，使得训练过程更加稳定。
2. **质量和多样性的权衡**：在生成模型中，我们通常需要在生成质量和生成多样性之间进行权衡。进行采样会增加生成多样性，但可能会降低生成质量。而直接使用均值向量可以保证生成质量，但可能会降低生成多样性。在许多应用中，我们更加关心生成质量，因此选择直接使用均值向量。

因此，这里输出的是均值，但是并且不进行继续采样。用均值计算就有重建损失$\log p(x|z) = -\frac{(x-\mu)^2}{2} + \text{const}$