再看变分自动编码器 VAE

赖泽强

2020 年 10 月 26 日

目录

1 Introduction

VAE 时 Variational Autoencoder 的缩写，中文名通常翻译为变分自动编码器。这个模型自从论文 Auto-Encoding Variational Bayes[1] 提出以来，受到了广泛的讨论和应用，并产生了许多变种，如 Conditional VAE 等等。虽然 VAE 中有一个 autoencoder，但 VAE 最初并不是从 autoencoder 引出的，而是推导结果中恰好有类似 autoencoder 的形式。

网络上关于 VAE 的教程很多，不过各个教程侧重的点比较分散，有些点可能这个教程有涉及，但另一个就没有。本文主要是基于 Tutorial on Variational Autoencoders[2] 这篇文章，对 VAE 再次进行一个比较详细的总结。

1.1 Background

考虑这么一个场景，我们有一堆数据 $X = x_1, x_2, ..., x_n$，我们想要**学习这些数据满足的分布**，用数学语言来说，我们希望找到一个分布 P，使得公式1所述的似然概率最大。

$$l(\theta) = P(x_1)P(x_2)...P(x_n) \tag{1}$$

虽然可以这样直接对数据 X 进行建模，但一般来说，我们不想直接对 X 进行建模，我们希望能在一个更紧凑的空间对 X 进行建模，因为这将有利于许多后续的任务。

换句话说，我们更想要的是用某个低维隐空间 Z 中的隐变量 z 对 X 中的个体进行建模，我们期望这个紧凑的隐空间足以描述原数据的所有特征，并且我们希望能够学习某个确定但还未知的

某个函数 f，能够将隐空间的变量映射到原数据空间 X 上去 — 以一个分布的形式，即输入一个 z，我们希望 f 能够输出这个 z 对应的 x 的分布。

用数学语言来说，我们实际上是在使用全概率公式将 P(x) 转换成公式2所示的形式，并将对 P(x) 建模转换成对 P(x|z) 和 P(z) 进行建模。

$$P(x) = \int P(x|z)P(z)dz \tag{2}$$

公式2的转换留给了我们三个问题，1. 首先是 P(x|z) 的分布形式；2. 其次是 P(z) 分布形式的选取；3. 最后是如何计算这个积分。第一个问题容易解决，而后两个问题则将由 VAE 进行解决。

1.2 $P(x|z;\theta)$ 的分布

P(x|z) 这个分布与数据的类型有关，通常我们会假设 P(x|z) 服从某种分布，例如对于连续型的数据，我们假设 P(x|z) 服从正态分布，而对于离散型的数据，我们假设 P(x|z) 服从伯努利分布。

事实上，对于 P(x|z)，我们可以随意选取其分布的形式，只要保证它可以计算，并且在其参数空间连续可导。这将保证我们可以使用梯度下降法拟合出分布函数的未知参数。

在 VAE 中，P(x|z) 取的是高斯分布，即 $P(X \mid z;\theta) = \mathcal{N}\left(X \mid f(z;\theta), \sigma^2 * I\right)$，其中均值和方差都是未知数，均值由一个函数 f 从输入 z 映射而来，并且这个函数将由神经网络实现。方差不参与运算，我们可以把它当作一个未知的确定值（对于某个 z）。

在确定了 P(x|z) 的分布形式后，**我们还需要确定 P(z) 的分布形式**。

1.3 $P(z)$ 的分布

一般来说，z 这个隐变量的真实分布是很复杂的，不同维的信息可能代表不同含义，不同维也可能互相关联。我们很难说用先验，我们也不想要（太复杂了）去手工定义 z 的分布形式。

对于 P(z)，VAE 的做法是“先从一个简单的分布中对 z 进行取样，比如说标准正态分布 $\mathcal{N}(0, I)$，然后通过一个足够复杂的函数（可以用神经网络表示）将这个正态分布映射到其他任何分布[1]。

回到前面的 $P(x \mid z;\theta) = \mathcal{N}\left(x \mid f(z;\theta), \sigma^2 * I\right)$，如果我们使用一个多层神经网络对这个分布进行建模，我们可以想象这个神经网络用它的前几层先将从标准值正态分布中取出的 z 映射到隐空间；然后后几层再将这个隐变量映射到最终的结果上（实际上是一系列的均值）

至此，我们已经确定了最初这个等式右边两项的形式了。

$$P(x) = \int P(x|z;\theta)P(z)dz$$

接下来的主要问题是，我们要如何最大化公式1所示的似然概率，并求出未知数 θ（即 P(x|z) 中的神经网络参数）。

2 VAE

2.1 Intractablility

既然我们要求最大值，一个很自然的想法就是对 $f(\theta) = P(x)$ 求导，因为 P(x) 实际上关于 θ 的函数，对 P(x) 求导并不能直接去掉积分符号，而当 $P(x|z;\theta)$ 很复杂的时候，这个积分是 intractable 的（这个证明我也没找到，有兴趣的可以自己找找看），因此我们没有办法用解析的方法最大化那个似然概率，求出未知数 θ。

[1] Any distribution in d dimensions can be generated by taking a set of d variables that are normally distributed and mapping them through a sufficiently complicated function[2]。

另一个也很自然的方法是用近似的方法来计算 P(X)：我们先随机取一系列的 $z = z_1, ..., z_n$，然后用 $P(X) \approx \frac{1}{n} \sum_i P(X \mid z_i)$ 对 P(X) 做一个近似。一些文章也会用蒙特卡洛法这个术语，意思是一样的。

这种方法可以反向传播（虽然我不知道怎么做），但是仍然有一个问题，那就是我们需要采样的数目要很大，才能得到一个比较满意的对 P(X) 的估计。

因为对大部分的 z，P(X|z) 都是 0，对我们估计 P(X) 一点用都没有。换句话说 θ 的变化并不会引起 P(X|z) 的变化，这部分 z 的梯度是 0，这些采样点对优化 θ 没有任何帮助。

2.2 Variational Inference

VAE 改进了这个 sample 的过程，它是怎么做的呢?

核心思想是尝试从更可能产生 X 的 z 中进行 sample，为此，VAE 添加了一个新的函数 Q(z|X)，可以输入一个 x，输出一个可能产生 x 的 z 的分布。

在这个函数的基础上，我们可以比较容易的计算出 $E_{z\sim Q}P(X|z) = \int P(X|z)Q(z|X)$ 这个期望（仍然要采样，但是采样数可以减少）。

但是问题是我们想要优化的是 $P(x) = E_{z\sim P(z)}P(x|z)$，我们需要想办法把这两个式子联系起来。

首先考虑近似分布 Q(z|X) 和真实分布 P(z|X) 的 KL divergence：

$$\mathcal{D}[Q(z|X)\|P(z \mid X)] = E_{z\sim Q}[\log Q(z|X) - \log P(z \mid X)]$$

使用贝叶斯法则引入 $P(X|z;\theta)$ 和 $P(X)$：

$$\mathcal{D}[Q(z|X)\|P(z \mid X)] = E_{z\sim Q}[\log Q(z|X) - \log P(X \mid z) - \log P(z)] + \log P(X)$$

重新组合 $\log Q(z|X)$ 和 $\log P(z)$ 成 $\mathcal{D}[Q(z|X)\|P(z)]$，移项得

$$\log P(X) - \mathcal{D}[Q(z \mid X)\|P(z \mid X)] = E_{z\sim Q}[\log P(X \mid z)] - \mathcal{D}[Q(z \mid X)\|P(z)]$$

至此，我们得到了 VAE 中最重要的一个公式。观察这个公式，我们可以看到：

- 等式左边有我们要最大化的 log 似然，$\log P(X)$
- 选取合适的 Q 的形式，右边可以使用随机梯度下降
- 右边的形式很像 autoencoder
- 因为 $\mathcal{D}[Q(z \mid X)\|P(z \mid X)]$ 大于 0，所以等式右边是 $\log P(X)$ 的一个下限（ELBO）

当我们在最大化等式右边时，我们事实上是在同时最大化 $\log P(X)$ 和最小化 $\mathcal{D}[Q(z \mid X)\|P(z \mid X)]$，其中后者没有解析的计算形式，无法直接优化。

我们通过优化等式右边来优化 $\log P(X)$ 基于以下的一个假设或者说 Intuition：即当我们用一个表达能力够强的模型来对 $Q(z \mid X)$，在优化时它将以很大概率快速收敛到 $P(z \mid X)$，即 $\mathcal{D}[Q(z \mid X)\|P(z \mid X)] = 0$，这时候优化 ELBO 等价于优化 $\log P(X)$。

至此，我们就将 intractable 的原式 $P(x) = \int P(x|z;\theta)P(z)dz$ 变成 tractable 的了。

2.3 $D_{KL}[N(\mu(X),\Sigma(X))\|N(0,1)]$ 的推导

这部分的推导源自“变分自编码器 VAE：原来是这么一回事”[2]。

我们只考虑各分量独立的多元正态分布，因此只用考虑一元正态分布的情况，推导如下：

$$\begin{aligned}
&KL\left(N\left(\mu,\sigma^2\right)\|N(0,1)\right)\\
=&\int\frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/2\sigma^2}\left(\log\frac{e^{-(x-\mu)^2/2\sigma^2}/\sqrt{2\pi\sigma^2}}{e^{-x^2/2}/\sqrt{2\pi}}\right)dx\\
=&\int\frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/2\sigma^2}\log\left\{\frac{1}{\sqrt{\sigma^2}}\exp\left\{\frac{1}{2}\left[x^2-(x-\mu)^2/\sigma^2\right]\right\}\right\}dx\\
=&\frac{1}{2}\int\frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/2\sigma^2}\left[-\log\sigma^2+x^2-(x-\mu)^2/\sigma^2\right]dx
\end{aligned}\tag{3}$$

整个结果分为三项积分，

- 第一项实际上就是乘以概率密度的积分 $-\log\sigma^2$，也就是 1，所以结果是 $-\log\sigma^2$。
- 第二项实际是正态分布的二阶矩，熟悉正态分布的朋友应该都清楚正态分布的二阶矩为 $\mu^2+\sigma^2$。
- 而根据方差的定义定义，第三项 $-1/\sigma^2*E_{x\sim N(\mu,\sigma^2)[(x-\mu)^2]}$，实际上就是“-方差除以方差 =-1”。

所以总结果就是：

$$KL\left(N\left(\mu,\sigma^2\right)\|N(0,1)\right)=\frac{1}{2}\left(-\log\sigma^2+\mu^2+\sigma^2-1\right)$$

3 Implementation

3.1 Reconstruction Loss

为什么实现的时候 reconstruction loss 不需要再 sample，而且 decoder 只输出 mean，不输出 std（对正态分布假设来说）？

简要总结 Stackoverflow 上的一个回答[3]，并做一些额外解释：

ELBO:

$$E_{z\sim q}[\log P(x|z)]-\mathrm{KL}(q(z)||p(z))$$

我们首先是用单个 sample 去估计这个期望 $E_{z\sim q}[\log P(x|z)]$。其次，由于 $P(x|z)\sim\mathcal{N}(\mu,\sigma^2)$，且 $\mu=f(z;\theta)$，最小化 $\log P(x|z)$ 等价于最小化

$$logP(x|z)=log(\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma}})=-\frac{(x-\mu)^2}{2\sigma}-log(\sigma\sqrt{2\pi})\propto-(x-\mu)^2$$

方差这里可能不是常量，但是因为和参数 θ 无关，可以视为一个常量，因此最大化 $logP(x|z)$ 等价于最小化 $(x-\mu)^2$。

[2] https://www.sohu.com/a/226209674_500659

[3] Shouldn't we sample from the output of variational auto-encoder?

References

[1] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". en. In: *arXiv:1312.6114 [cs, stat]* (May 2014). arXiv: 1312.6114. URL: http://arxiv.org/abs/1312.6114 (visited on 09/30/2020) (cit. on p. 1).

[2] Carl Doersch. "Tutorial on Variational Autoencoders". en. In: *arXiv:1606.05908 [cs, stat]* (Aug. 2016). arXiv: 1606.05908. URL: http://arxiv.org/abs/1606.05908 (visited on 10/09/2020) (cit. on pp. 1, 2).