24 Jun

生成扩散模型漫谈（十九）：作为扩散ODE的GAN

By 苏剑林 | 2023-06-24 | 31058位读者 引用

在文章《生成扩散模型漫谈（十六）：W距离 ≤ 得分匹配》中，我们推导了Wasserstein距离与扩散模型得分匹配损失之间的一个不等式，表明扩散模型的优化目标与WGAN的优化目标在某种程度上具有相似性。而在本文，我们将探讨《MonoFlow: Rethinking Divergence GANs via the Perspective of Wasserstein Gradient Flows》中的研究成果，它进一步展示了GAN与扩散模型之间的联系：GAN实际上可以被视为在另一个时间维度上的扩散ODE！

这些发现表明，尽管GAN和扩散模型表面上是两种截然不同的生成式模型，但它们实际上存在许多相似之处，并在许多方面可以相互借鉴和参考。

思路简介

我们知道，GAN所训练的生成器是从噪声$\boldsymbol{z}$到真实样本的一个直接的确定性变换$\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z})$，而扩散模型的显著特点是“渐进式生成”，它的生成过程对应于从一系列渐变的分布$p_0(\boldsymbol{x}_0), p_1(\boldsymbol{x}_1), \cdots, p_T(\boldsymbol{x}_T)$中采样（注：在前面十几篇文章中，$\boldsymbol{x}_T$是噪声，$\boldsymbol{x}_0$是目标样本，采样过程是$\boldsymbol{x}_T \to \boldsymbol{x}_0$，但为了便于下面的表述，这里反过来改为$\boldsymbol{x}_0 \to \boldsymbol{x}_T$）。看上去确实找不到多少相同之处，那怎么才能将两者联系起来呢？

很明显，如果想要从扩散模型的视角理解GAN，那么就要想办法构造出一系列渐变的分布出来。生成器$\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z})$本身就是一个一步到位的变换，不存在渐变，然而我们知道模型的优化是渐变的，可否用参数$\boldsymbol{\theta}$的历史轨迹$\boldsymbol{\theta}_t$来构建这一系列渐变分布呢？具体来说，假设生成器初始化为$\boldsymbol{\theta}_0$，经过T步对抗训练后得到最优参数$\boldsymbol{\theta}_T$，训练过程的中间参数为$\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_{T-1}$，那么我们定义$\boldsymbol{x}_t = \boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z})$，不就定义了一系列渐变的$\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_T$，从而也就定义了渐变的分布$p_0(\boldsymbol{x}_0), p_1(\boldsymbol{x}_1), \cdots, p_T(\boldsymbol{x}_T)$了？

如果这个思路可行的话，那么GAN就可以诠释为梯度下降的（虚拟）时间维度上的扩散模型！下面我们就沿着这个思路进行探索。

梯度之流

首先，我们需要重温上一篇文章《梯度流：探索通向最小值之路》关于Wasserstein梯度流的结果：它指出方程

$$\frac{\partial q_t(\boldsymbol{x})}{\partial t} = -\nabla_{\boldsymbol{x}}\cdot\big(q_t(\boldsymbol{x})\nabla_{\boldsymbol{x}}\log r_t(\boldsymbol{x})\big) \tag{1}$$

在最小化$p(\boldsymbol{x})$和$q_t(\boldsymbol{x})$的KL散度，即$\lim\limits_{t\to\infty} q_t(\boldsymbol{x}) = p(\boldsymbol{x})$，这里$r_t(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{q_t(\boldsymbol{x})}$。如果$p(\boldsymbol{x})$代表真实样本的分布，那么如果能实现从$q_t(\boldsymbol{x})$采样的话，那么逐渐推到$t\to\infty$时，就可以实现从$p(\boldsymbol{x})$采样了。根据《测试函数法推导连续性方程和Fokker-Planck方程》，从$q_t(\boldsymbol{x})$采样可以通过下述ODE实现：

$$\frac{d\boldsymbol{x}}{dt} = \nabla_{\boldsymbol{x}}\log r_t(\boldsymbol{x}) \tag{2}$$

然而，上式中的$r_t(\boldsymbol{x})$是未知的，所以我们还无法通过上式进行采样，需要先想办法估算$r_t(\boldsymbol{x})$。

判别估计

这时候登场的是GAN的判别器。以最早的Vanilla GAN为例，它的训练目标是

$$\max_D\, \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\log\sigma(D(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log(1-\sigma(D(\boldsymbol{x})))] \tag{3}$$

这里的D是判别器，$\sigma(t)=1/(1+e^{-t})$是sigmoid函数，$p(\boldsymbol{x})$是真样本的分布，$q(\boldsymbol{x})$是假样本的分布。可以证明（不清楚的读者可以参考《RSGAN：对抗模型中的“图灵测试”思想》中的“补充证明”一节），上式中判别器D的理论最优解是

$$D(\boldsymbol{x}) = \log\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \tag{4}$$

更一般化的f-GAN（参考《f-GAN简介：GAN模型的生产车间》、《Designing GANs：又一个GAN生产车间》）结果会稍有不同，但可以证明的是它们判别器的理论最优解都

是$\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}$的函数。也就是说，只要我们可以实现从$p(\boldsymbol{x})$和$q_t(\boldsymbol{x})$中采样，那么通过GAN的判别器训练$(3)$就可以估算出$r_t(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{q_t(\boldsymbol{x})}$出来。

向前一步

这时候可能有读者疑惑：这不就进入“鸡生蛋、蛋生鸡”的循环论证了吗？我们估算$r_t(\boldsymbol{x})$不就是为了利用式(2)实现从$q_t(\boldsymbol{x})$中采样吗？现在你又假设能从$q_t(\boldsymbol{x})$采样才来估算$r_t(\boldsymbol{x})$？不着急，经典的一笔就要来了。

假设我们有生成器$\boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z})$，它的采样生成结果就等于从$q_t(\boldsymbol{x})$采样的结果，即

$$\left\{\boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z})\middle|\,\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})\right\} \quad = \quad \left\{\boldsymbol{x}_t\middle|\,\boldsymbol{x}_t\sim q_t(\boldsymbol{x})\right\} \tag{5}$$

那么现在我们就可以利用它和式(3)来估算$r_t(\boldsymbol{x})$。注意这只是t时刻的$r_t(\boldsymbol{x})$，其他时刻的$r_t(\boldsymbol{x})$我们并不知道，所以无法直接通过式(2)完成最终的采样过程，但是我们可以往前推一小步：

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \epsilon\nabla_{\boldsymbol{x}_t}\log r_t(\boldsymbol{x}_t) = \boldsymbol{x}_t + \epsilon\nabla_{\boldsymbol{x}_t} D(\boldsymbol{x}_t) \tag{6}$$

这里的ϵ是一个很小的正数，代表步长。那么，现在我们就有了下一步采样的结果，我们希望它继续能等价于下一步的生成器的采样结果，即

$$\begin{aligned}\left\{\boldsymbol{g}_{\boldsymbol{\theta}_{t+1}}(\boldsymbol{z})\middle|\,\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})\right\} \quad =& \quad \left\{\boldsymbol{x}_{t+1}\middle|\,\boldsymbol{x}_{t+1}\sim q_{t+1}(\boldsymbol{x})\right\} \\ =& \quad \left\{\boldsymbol{x}_{t+1}\middle|\,\boldsymbol{x}_t + \epsilon\nabla_{\boldsymbol{x}_t} D(\boldsymbol{x}_t), \boldsymbol{x}_t\sim q_t(\boldsymbol{x})\right\}\end{aligned} \tag{7}$$

换句话说，我们想要**将扩散模型中样本的运动转化为生成器参数的运动**！为了达到这个目标，我们通过如下损失去求$\boldsymbol{\theta}_{t+1}$：

$$\boldsymbol{\theta}_{t+1} = \mathop{\text{argmin}}_{\boldsymbol{\theta}}\,\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\left\Vert\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z}) - \boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z}) - \epsilon\nabla_{\boldsymbol{g}} D(\boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z}))\right\Vert^2\right] \tag{8}$$

也就是说，拿$\boldsymbol{x}_t = \boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z})$往前迭代一步得到$\boldsymbol{x}_{t+1}$，然后希望新的$\boldsymbol{g}_{\boldsymbol{\theta}_{t+1}}(\boldsymbol{z})$能尽量等于$\boldsymbol{x}_{t+1}$。完成这一轮后，再用$\boldsymbol{\theta}_{t+1}$替代原本的$\boldsymbol{\theta}_t$开始新一轮的迭代，也就是式$(3)$和式$(8)$交替执行，是不是就有GAN的味道了？

点睛之笔

如果这还不够，我们还可以继续完善一下，将它变得跟GAN更加一致。注意到式(8)的被期望函数的梯度是：

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}}\Vert \boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z}) - \boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z}) - \epsilon\nabla_{\boldsymbol{g}} D(\boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z}))\Vert^2 \\ =&\, 2\big\langle \boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z}) - \boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z}) - \epsilon\nabla_{\boldsymbol{g}} D(\boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z})), \nabla_{\boldsymbol{\theta}}\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z})\big\rangle\end{aligned}\tag{9}$$

代入当前值$\boldsymbol{\theta}=\boldsymbol{\theta}_t$，那么结果是

$$-2\epsilon\big\langle \nabla_{\boldsymbol{g}} D(\boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z})), \nabla_{\boldsymbol{\theta}_t}\boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z})\big\rangle = -2\epsilon\nabla_{\boldsymbol{\theta}_t} D(\boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z}))\tag{10}$$

也就是说，如果用基于梯度的优化器只优化一步的话，那么以式(8)为损失函数，跟以下式为损失函数，结果是等价的（因为梯度只差一个常数倍）：

$$\boldsymbol{\theta}_{t+1} = \mathop{\text{argmin}}_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[-D(\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z}))]\tag{11}$$

这便是常见的生成器损失之一。式(3)和式(11)交替训练，就是一个常见的GAN变体。

特别地，原论文还证明了生成器的损失函数可以一般化为

$$\boldsymbol{\theta}_{t+1} = \mathop{\text{argmin}}_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[-h(D(\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{z})))]\tag{12}$$

其中$h(\cdot)$是任意单调递增函数，它也对应于Wasserstein梯度流(1)中的$\log r_t(\boldsymbol{x})$可以换成$h(\log r_t(\boldsymbol{x}))$，这应该也是MonoFlow一词的来源（Monotonically increasing function + Wasserstein flow）。这个证明过程就不展开了，大家自行看原论文就好。

意义思考

总的来说，将GAN理解为扩散模型的思路是

$$\cdots \xrightarrow{\quad} \boldsymbol{g}_{\boldsymbol{\theta}_t}(\boldsymbol{z}) \xrightarrow{\text{式}(3)} r_t(\boldsymbol{x}) \xrightarrow{\text{式}(6)} \boldsymbol{x}_{t+1} \xrightarrow{\text{式}(8)} \boldsymbol{g}_{\boldsymbol{\theta}_{t+1}}(\boldsymbol{z}) \xrightarrow{\quad} \cdots$$

其中，核心的式子是(6)，它源于Wasserstein梯度流的式(1)和式(2)，这部分我们在上一篇文章《梯度流：探索通向最小值之路》讨论过了。

可能有读者想问：这个视角看上去并没有得到比GAN更多的东西，为什么要费这番大功夫去重新理解GAN呢？首先，在笔者看来，从扩散模型角度理解GAN，或者说将扩散模型和GAN统一起来，它本身就是一件很有趣、很好玩的事情，并不一定需要发挥什么实际作用，有趣、好玩就是它最大的意义。

其次，如同作者在原论文所说，已有的GAN的推导过程跟它实际的训练过程是不一致的，而本文所讨论的扩散视角，则是跟训练过程是一致的。也就是说，以训练过程为标准的话，GAN已有的推导过程是错的，本文的扩散视角才是对的。怎么理解这一点呢？以前面提到的GAN为例，判别器和生成器的目标分别是：

$$\max_D \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\log\sigma(D(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log(1-\sigma(D(\boldsymbol{x})))] \tag{13}$$

$$\min_q \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[-D(\boldsymbol{x})] \tag{14}$$

通常的证明方式是，证明D的最优解是$\log\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}$，然后代入生成器的损失函数，发现它在最小化$q(\boldsymbol{x}),p(\boldsymbol{x})$的KL散度，所以最优解是$q(\boldsymbol{x})=p(\boldsymbol{x})$。但是，这样的证明对应的训练方式应该是先针对任意的$q(\boldsymbol{x})$，将$\max\limits_D$这一步都求解出来（求出的D应该是$q(x)$的函数，或者说应该是生成器参数$\boldsymbol{\theta}$的函数），然后再去执行$\min\limits_q$这一步，而不是实际上用的交替训练。而基于扩散模型的理解，它在设计上就是交替的，所以它跟训练过程更加一致。

总的来说，从扩散模型的角度来理解GAN，不单单是理解GAN的一种新视角，而且还是一种更贴近训练过程的视角。比如，我们可以解释为什么GAN的生成器不能训练太多步，因为只有单步优化时，式(11)和式(8)才等价，如果GAN要进行更多步的优化，那么应该使用式(8)为损失函数。事实上，式(8)就相当于笔者之前在《用变分推断统一理解生成模型（VAE、GAN、AAE、ALI）》所提出的$KL\left(q(x)\Vert q^o(x)\right)$项，它保证了生成器的“传承”而不仅仅是“创新”。

文章小结

本文介绍了MonoFlow，它展示了我们可以将GAN理解为在另一个时间维度上的扩散ODE，从而建立了一种基于扩散模型理解GAN的新视角。特别地，这是一种比GAN的常

规推导更加贴近训练过程的视角。

如果您需要引用本文，请参考：

苏剑林. (Jun. 24, 2023). 《生成扩散模型漫谈（十九）：作为扩散ODE的GAN 》[Blog post]. Retrieved from https://spaces.ac.cn/archives/9662

```
@online{kexuefm-9662,
    title={生成扩散模型漫谈（十九）：作为扩散ODE的GAN},
    author={苏剑林},
    year={2023},
    month={Jun},
    url={\url{https://spaces.ac.cn/archives/9662}},
}
```