生成对抗网络的几何观点以及算法优化

项目研究成果

生成对抗网络与其几何解释

项目编号：201910055055.

负 责 人：崔嘉珩.

所在院系：数学科学学院 统计学.

联系电话：18622705306.

指导教师：吴春林.

摘　要

生成对抗网络（GAN）模型作为一个生成模型，在近些年很受欢迎。它可以在不知道目标解析表达式的情况下给出一个从简单分布 (例如高斯分布) 到目标分布的映射。生成对抗网络模型由生成器 G 和判别器 D 构成，它们都是深度神经网络。因此，生成对抗网络能够学习并生成图像甚至视频等高维对象。

为了解决在生成对抗网络中出现的梯度消失问题，人们提出了许多方法，引人关注的 WGAN 就是其中之一。它采用了最优传输理论中的 Wasserstein 距离代替 GAN 中的 JS 散度，来作为判别器的损失函数，解决了梯度消失问题。同时，通过最优传输中的 Kantorovich 方法，我们可以知道在最优传输的观点下，生成对抗网络的生成器 G 中的映射等价于最优传输问题中的最优传输映射；生成对抗网络的判别器 D 中的 Wasserstein 距离等价于最优传输问题中的 Kantorovich 势函数。我们证明：在代价函数为凸函数的情况下，二者是等价的。

关键词**:** 生成对抗网络；WGAN；最优传输理论

Abstract

Generative adversarial network (GAN) model is popular as a generative model in recent years. It can give a mapping from a simple distribution, e.g. Gaussian distribution, to a target distribution, even without knowing the explicit expression of the target distribution. The GAN model consists of a generator G and a discriminator D, both of which are deep neural networks. Therefore, GAN models are able to learn and generate high-dimensional objects such as images and even videos.

To solve the gradient vanishing problem that arises in generative adversarial networks, many methods have been proposed, and the intriguing WGAN is one of them. It adopts the Wasserstein distance in optimal transportation theory instead of the JS divergence in GAN as the loss function of the discriminator, so as to solve the gradient vanishing problem. Meanwhile, by the Kantorovich's approach in optimal transportation, we can know that the mapping in the generator G is equivalent to the optimal transportation map under the viewpoint of optimal transportation; the Wasserstein distance in the discriminator D is equivalent to the Kantorovich potential function . We prove that these two are equivalent in the case where the cost function is convex.

Key Words: Generative Adversarial Networks; WGAN; Optimal Transportation Theory

目录

目录

第一章　生成对抗网络 (GAN)

第一节　GAN 的模型

GAN 的模型如下图[2] 所示：

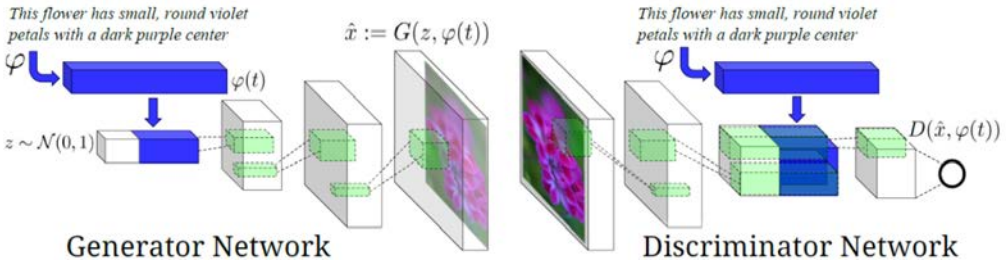

图 1: GAN 的模型

从图 1 中可见，GAN[1] 由两部分组成：生成器 (Generator)G 和判别器 (Discriminator)D。

从数学上来讲，生成器 G 是一个从给定的简单分布 (如高斯分布) 到目标分布 (如人脸图片的分布) 的映射。给生成器一个简单分布的随机向量，它可以一个样本。生成器由一个深度神经网络来实现。

从数学上来讲，判别器 D 是一个度量，它用 JS 散度[4] 给出两个分布之间的“距离”（严格来说，JS 散度并不是一种距离，因为其不满足对称性），用来衡量样本是由生成器生成还是来自训练集。判别器也由一个深度神经网络来实现。

生成器 G 和判别器 D 是同时相互进行训练的，我们希望生成器生成的样本越来越贴近目标分布中的样本，同时也希望判别器的判别能力能够提高。生成器和判别器的工作像有一种“矛盾关系”，因此我们称 GAN 的训练为“对抗训练”。这种“对抗”使得整个网络能提高生成样本和判别样本的能力，直到最后达到纳什均衡，也就是说训练到假冒品和真品几乎无法区分为止。

第二节　GAN 的原理

1.2.1　GAN 的对抗框架

原作者[1] 在设计 GAN 时，认为两者都使用多层感知机 (multilayer perceptron)[5] 时，对抗框架最容易实现。其提出了如下的对抗框架：

- 生成器 G：为了从训练集 X_{train} 中学习生成器的分布 p_g，定义一个输入噪声变量 $p_z(z)$，把从它到数据空间的映射表示为 $G(z;\Theta_g)$。这里的 G 是一个由参数为 Θ_g 的可微函数，由一个多层感知机实现。

- 判别器 D：第二个多层感知机 $D(x;\Theta_d)$，它输出一个标量，参数为 Θ_d。$D(x)$ 表示样本 x 来自训练集而不是 p_g 的概率。
- D 和 G 的目标：我们训练 D，使得它为训练集样本和 G 生成的样本分配正确标签的可能性最大化，同时训练 G 来最小化生成样本分布和给定分布之间的距离，即在给定 $D(x)$ 时最小化 $log(1-D(G(z)))$ 的期望。

简而言之，D、G 进行如下的 min-max 博弈：

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))] \tag{1.1}$$

理论上我们想使得最后 $p_g = p_{data}$，此时判别器无法区别两个分布，$D(x) = 1/2$，说明生成器生成的样本已经达到了“以假乱真”的程度。具体的训练过程可以由下图[1] 来表示：

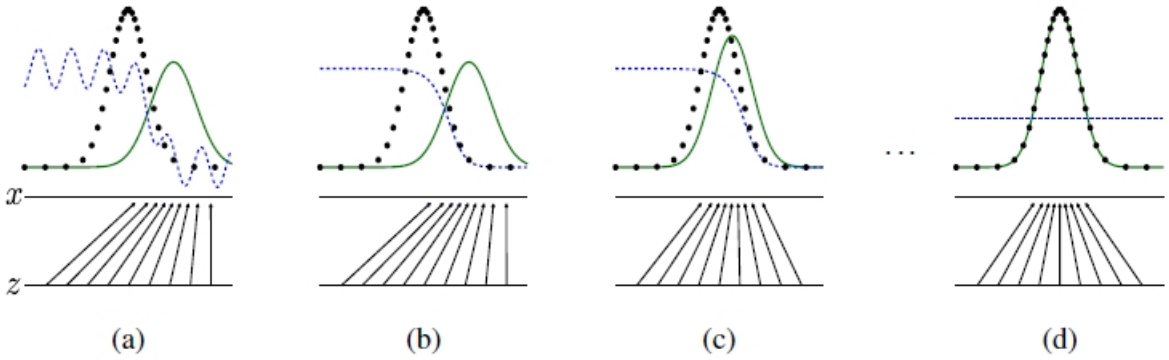

图 2: 我们要更新判别分布 (D，蓝色虚线)，使其能够区分训练集数据分布 p_x(黑色虚线) 与生成分布 p_g(G，绿色实线) 的样本。下方的水平线是 z 的采样域，在本图中进行的是均匀采样。我们用映射 $x = G(z)$ 将 z 映到 x(黑色箭头)。
(a) 考虑一个接近收敛的模型：p_g 与 p_{data} 相似，D 部分精确。
(b) 更新 D，使其能对样本和数据进行区分，其会收敛到 $D(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$ 。
(c) 更新 G（实际情况下，应该更新 k 步 G 后更新一次 D，k 是一个整数超参数），D 的梯度引导 $G(z)$“流向”更有可能被分类为数据的区域。
(d) 训练收敛的情况：如果 G 和 D 有足够的容量，它们会达到一个点，在这个点上 $p_g = p_{data}$。此时判别器无法区分两种分布，即 $D^*(x) = 1/2$。

1.2.2 GAN 的训练算法

GAN 的训练算法伪代码如下，其中超参数 m 为每一批学习的样本个数，超参数 k 指更新 k 步 G 后更新一次 D，这里随机梯度上升/下降可以换成任何其他基于梯度的学习方法。

Algorithm 1 GAN 的小批量随机梯度下降训练

for 训练迭代次数 **do**

for k 次 **do**

- 从噪声先验 $p_g(z)$ 选取 m 个小批量样本 $z^{(1)}, z^{(2)}, ..., z^{(m)}$
- 从生成器的生成分布 $p_{data}(x)$ 选取 m 个小批量样本 $x^{(1)}, x^{(2)}, ..., x^{(m)}$
- 通过随机梯度上升更新判别器：

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right] \tag{1.2}$$

end for

- 从噪声先验 $p_g(z)$ 选取 m 个小批量样本 $z^{(1)}, z^{(2)}, ..., z^{(m)}$
- 通过随机梯度下降更新生成器：

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \tag{1.3}$$

end for

1.2.3 $p_g = p_{data}$ 时的全局最优性

下面叙述几个定理来说明图 2(d) 中 $p_g = p_{data}$ 时训练结束的合理性，具体证明请见 [1]：

Theorem 1. 对于固定的 G，最优的判别器 D 是：

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \tag{1.4}$$

现在可以将判别函数改写为

$$\begin{aligned} C(G) &= \max_D V(G, D) \\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \left[\log D_G^*(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}} \left[\log\left(1 - D_G^*(G(\boldsymbol{z}))\right)\right] \\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \left[\log D_G^*(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{x} \sim p_g} \left[\log\left(1 - D_G^*(\boldsymbol{x})\right)\right] \\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{da}}} \left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x} \sim p_g} \left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] \end{aligned} \tag{1.5}$$

Theorem 2. 当且仅当 $p_g = p_{data}$，虚拟训练准则 $C(G)$ 达到全局最小值，此时 $C(G)$ 的值为 $-log4$，这样生成器模型完美地复制了数据分布。

同时我们在这里给出算法的收敛性[1]：

Theorem 3. 如果 G，D 有足够的容量，而且在上述算法中的每个步骤中，判别器都达

到了在给定的 G 下的最佳值，并且 p_g 也被更新从而满足标准

$$\mathbb{E}_{\boldsymbol{x}\sim p_{data}}\left[\log D_G^*(\boldsymbol{x})\right]+\mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log\left(1-D_G^*(\boldsymbol{x})\right)\right] \tag{1.6}$$

这样 p_g 收敛到 p_{data}。

在实践中，生成对抗网络通过 $G(z;\Theta_g)$ 表示有限的 p_g，优化了 Θ_g 而不是 p_g 本身，所以无法直观判断 $p_g = p_{data}$，因此只能依靠循环算法来尽量尝试让生成图片逼近真实图片 (即肉眼看起来更像)。值得庆幸的是，从经验上来看，这个算法在训练次数较多时，一大部分生成图片比较真实，能达到人眼不细看分辨不出来真假的情况。

第三节　GAN 的优势

- 当真实数据的概率分布不可或难以计算的时候（尤其对于图片、视频、语音等高维数据），传统依赖于数据内在解释的生成模型无法直接应用，但是生成对抗网络模型依然可以使用。
- 机器学习需要大量数据样本，而生成对抗网络模型自身理论上可以生成无尽的样本，从而降低了对于训练数据的需求，因此极其适合无监督学习。
- 理论上该框架可以训练所有的生成模型。

GAN 现在已经应用在许多领域，主要用途是用来生成常规情况下无法获得的大量样本，以及图像的修复和分辨率提高。例如 Andrew Brock 等人的 Large Scale GAN Training for High Fidelity Natural Image Synthesis[6] 中提到的生成现实图片，Deepak Pathak 等人在 Context Encoders: Feature Learning by Inpainting[7] 中提出的图片修复功能，都有很好的成果。

第四节　GAN 的缺点

- GAN 的训练过程是不可视的，也就是俗称的“黑箱算法”。这导致 GAN 是一个很难解释的模型，导致我们很难解释其生成结果为什么好或为什么坏。近年来，一些学者从几何和最优传输理论方面对 GAN 进行了一些解释[9][10][11]，详见我们的结项报告。
- GAN 存在梯度消失问题。简单地说，判别器训练得越好，梯度就越趋向于 0，越容易出现梯度消失情况。此时优化就无从谈起。GAN 的优化模型 WGAN[12] 解决了这个问题，我们将在第二章介绍 WGAN。

- GAN 存在模式崩溃[11][13][14] 的问题。比如当训练的图片种类只有猫一种时，可以得到不错的效果；但是如果同时混入了猫狗两种训练样本，此时 GAN 的生成结果就并不好。此外还存在一些现象：有时虽然涵盖了训练的所有模式，但是生成的样本有一些是与训练样本完全无关的，无意义的。而且 GAN 的生成结果非常随机，难以完全复现。

第二章　Wasserstein GAN

针对 GAN 存在的诸多问题，有许多的研究成果做出了一定的优化。其中有一种引人注目的方法——Wasserstein GAN(简称为 WGAN)[12]，其引入了最优传输理论中的 Wasserstein 距离[15]。WGAN 采用 Wasserstein 距离作为判别器的损失函数，这样当生成分布和训练集样本分布的支撑集没有相交部分时，WGAN 就可以得到一个用来优化生成器的合适梯度。WGAN 作者进行了非常繁复的数学推导，而算法上的改动却非常简单。

第一节　JS 散度与 Wasserstein 距离

WGAN 对 GAN 的优化是基于几个概念的。GAN 用 JS 散度来衡量生成分布与训练集分布的距离，而 WGAN 使用 Wasserstein 距离 (又称 Earth-Mover 距离，简称 EM 距离)。

Definition 1. *Kullback-Leibler(KL)* 散度[3]

$$KL\left(\mathbb{P}_r \| \mathbb{P}_g\right) = \int_{\mathcal{X}} \log\left(\frac{P_r(x)}{P_g(x)}\right) P_r(x) d\mu(x) \tag{2.1}$$

KL 散度是不对称的，且要求 $P_g(x) \neq 0, \quad \forall x \in \mathcal{X}$，因此其不是一个距离。

Definition 2. *Jensen-Shannon(JS)* 散度[4]

$$JS\left(\mathbb{P}_r, \mathbb{P}_g'\right) = KL\left(\mathbb{P}_r \| \mathbb{P}_m\right) + KL\left(\mathbb{P}_g \| \mathbb{P}_m\right) \tag{2.2}$$

这里的 $\mathbb{P}_m = \frac{(\mathbb{P}_r + \mathbb{P}_g)}{2}$，这种散度是对称的而且总是有定义的（因为我们可以取 $\mu = \mathbb{P}_m$）。

Definition 3. *Wasserstein* 距离

$$W\left(\mathbb{P}_r, \mathbb{P}_g\right) = \inf_{\gamma \in \Pi\left(\mathbb{P}_r, \mathbb{P}_g\right)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|] \tag{2.3}$$

这里的 $\Pi\left(\mathbb{P}_r, \mathbb{P}_g\right)$ 表示所有边际分布是 $\mathbb{P}_r$ 和 $\mathbb{P}_g$ 的联合分布 $\gamma(x, y)$. 直观地说，$\gamma(x, y)$ 表示要从 x 到 y 传输多少质量才能转换分布 $\mathbb{P}_r$ 到 $\mathbb{P}_g$[15][16][17][18]。这样 Wasserstein 距离就是最优传输方法的成本。

第二节　WGAN 对 GAN 的改进

我们先给出 WGAN 的小批量随机梯度下降训练的算法伪代码。

算法中的超参数如下：学习速率 α，裁剪参数 c，每批数据个数 m，每次生成器迭代时的判别器的迭代次数 k，初始判别器参数 ω_0，初始生成器参数 θ_0。

$RMSProp$ 指的是采用 RMSProp 算法的优化器[20]。

$clip(w, -c, c)$ 被称为权重裁剪 (weight clipping)，指的是将权重 w 裁剪至闭区间 $[-c, c]$ 内，这样能保证 g_w 是 Lipschitz 连续的。这样 Wasserstein 距离处处连续，几乎处处可微，从而避免梯度消失问题[12]。

Algorithm 2 WGAN 的小批量随机梯度下降训练

while θ 尚未收敛 **do**
  for k 次 **do**
    • 从真实数据分布 $\mathbb{P}_r$ 选取样本 $\{x^{(i)}\}_{i=1}^m$
    • 从给定先验分布 $p(z)$ 选取样本 $\{z^{(i)}\}_{i=1}^m$
    •$g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m \int_w \left(x^{(i)}\right) - \frac{1}{m}\sum_{i=1}^m \int_w \left(g_\theta\left(z^{(i)}\right)\right)\right]$
    •$w \leftarrow w + \alpha \cdot \text{RMSProp}\left(w, g_w\right)$
    •$w \leftarrow \text{clip}(w, -c, c)$
  end for
  • 从给定先验分布 $p(z)$ 选取样本 $\{z^{(i)}\}_{i=1}^m$
  •$g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m \int_w \left(g_\theta\left(z^{(i)}\right)\right)$
  •$\theta \leftarrow \theta - \alpha \cdot RMSProp\left(\theta, g_\theta\right)$
end while

与 GAN 相比 WGAN 所做的改动主要是三点：

- 生成器与判别器的损失函数没取对数.
- 判别器最后一层没有使用 sigmoid 函数.
- 更新判别器参数后进行权重裁剪.

作者对 WGAN 进行了大量的实验，其实验结果能说明两点好处：

- 得到了与生成器的收敛性以及生成样本质量相关的有意义的度量。在算法中的损失函数是一个对 Wasserstein 距离的估计，其趋向于一个与设置的 Lipschitz 常数有

关的常数。但是这个估计也是模糊的，缺少明显的计算方法。虽然如此，这仍然是对 GAN 的一个巨大改进，解决了梯度消失的问题。

- 提高了优化过程的稳定性。在众多实验中均未观测到模式崩溃现象。但无法证明模式崩溃问题已经解决。

第三节　WGAN 的后续改进

WGAN 也存在一定缺点，比如它训练困难，收敛速度慢。有一些人对 WGAN 提出了改进，比如蒙特利尔大学的研究者发表的论文 Improved Training of Wasserstein GANs[21] 中就提出了 WGAN-GP 模型。

他们发现 WGAN 仍然存在着模式崩溃现象，而原因通常是由于在 WGAN 中使用了权重裁剪来在判别器函数上实现 Lipschitz 约束。他们提出了一种替代权重裁剪的办法：根据判别器的输入来惩罚判别器的梯度的范数（即 GP）。这种方法比 WGAN 的性能更好，并且在不调参的前提下对多种 GAN 架构进行训练。

他们认为通过权重裁剪这种方法调整会使判别器偏向更简单的函数，从而无法捕捉数据分布的高阶细节。他们提出了另外一种方法实施 Lipschitz 约束：由于一个可微函数是 1-Lipschtiz 的，当且仅当他在任何地方有至多为 1 的梯度范数，所以他们考虑针对判别器输入直接限制判别器输出的梯度范数。为了实现可处理性，他们用一个对随机样本 $\widehat{x} \sim \mathbb{P}_{\widehat{x}}$ 放松了约束条件。他们使用的新的判别器函数是

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})]}_{\text{WGAN 的损失函数}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\alpha}}}\left[\left(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1\right)^2\right]}_{\text{新增的梯度惩罚项}}. \tag{2.4}$$

在后面的实验中研究者均使用了四种不同的 GAN 进行训练：使用 WGAN-GP，WGAN，DCGAN，最小二乘 GAN，使用了 WGAN-GP 是唯一一种使用同一种默认超参数，并在每个架构下都成功训练的方法。因此 WGAN-GP 要优于 WGAN。

第三章　最优传输理论对 GAN 的解释

在最优传输理论的观点下，生成器和判别器之前并不是相互竞争的关系。相反，在 L^2 成本函数下，生成器和判别器的训练是相互促进的。我们应该充分利用它的这一特点，共享部分计算内容来提高效率。在我们的算法中实现了这一点。

第一节　生成器与判别器

生成器 G 可以看作一个从隐空间 $\mathcal{Z}$ 到样本空间 $\mathcal{X}$ 的一个映射 ($g_\theta : \mathcal{Z} \to \mathcal{X}$)，其中隐空间 $\mathcal{Z}$ 是样本空间 $\mathcal{X}$ 的特征空间。我们需要用深度网络学习参数 θ。

令 ζ 为隐空间 $\mathcal{Z}$ 上的一个简单分布 (例如高斯分布)，则生成器 G 使隐空间 $\mathcal{Z}$ 上的一个简单分布 ζ 变成 $\mu_\theta = g_\theta^* \zeta$ (其中 g_θ^* 为映射 g_θ 的导出映射，细节请见结项报告)，μ_θ 就是样本空间 $\mathcal{X}$ 中一个由隐空间 $\mathcal{Z}$ 上的一个简单分布 ζ 导出的分布。

判别器 D 利用 Wasserstein 距离 $W_c(\mu_\theta, \nu)$ 计算生成的样本分布 μ_θ 和目标分布 ν 之间的误差，其中 Wasserstein 距离 $W_c(\mu_\theta, \nu)$ 等价于最优传输当中的 Kantorovich 势函数 φ_ξ，它由另一个深度神经网络通过参数 ξ 确定。

总结来说，生成器 G 通过优化参数 θ，使得 μ_θ 逼近 ν；判别器 D 通过优化参数 ξ，使得 φ_ξ 逼近 Wasserstein 距离 $W_c(\mu_\theta, \nu)$。

用数学公式可以表示为以下的优化过程：

$$\min_\theta \max_\xi \mathbb{E}_{x\sim\zeta} \left(\varphi_\xi \left(g_\theta(x)\right) + \mathbb{E}_{y\sim\nu} \left(\varphi_\xi^c(y)\right)\right) \tag{3.1}$$

在最优传输的观点下[8]，生成对抗网络的生成器 G 中的映射 g_θ 等价于最优传输问题中的最优传输映射 T；生成对抗网络的判别器 D 中求解 Wasserstein 距离 $W_c(\mu_\theta, \nu)$，等价于求解最优传输问题中的 Kantorovich 势函数 φ_ξ。

第二节　最优传输映射与 Kantorovich 势函数的等价性

令 $\mathcal{X} \subset \mathbb{R}^n$ 为训练集样本空间，$\mathcal{P}(\mathcal{X})$ 表示 X 上的概率测度全体，$\nu \in \mathcal{P}(\mathcal{X})$ 表示真实样本的概率分布，则生成器训练的过程实际上是产生一个逼近 ν 的参数族 $\mu_\theta = g_\theta^* \zeta$，其中 ζ 为隐空间 $\mathcal{Z}$ 上的一个简单分布 (高斯分布)；判别器实际上是一个特别的损失函数，利用 Wasserstein 距离 $W_c(\mu_\theta, \nu)$ 计算生成的样本分布 μ_θ 和目标分布 ν 之间的误差，其中 Wasserstein 距离 $W_c(\mu_\theta, \nu)$ 定义如下：

$$W_c(\mu_\theta, \nu) = \min_{\gamma \in \mathcal{P}(\mathcal{X}\times\mathcal{X})} \left\{ \int_{\mathcal{X}\times\mathcal{X}} c(x,y) d\gamma(x,y) | \pi_x^* \gamma = \mu_\theta, \pi_y^* \gamma = \nu \right\} \tag{3.2}$$

我们的优化对象实际为[12][9]: $\min_\theta W_c(\mu_\theta, \nu)$。

利用 Kantorovich 对偶问题[15] 我们可以知道：

$$\min_{\gamma\in\mathcal{A}(\mu,\nu)} \int_{X\times Y} c(x,y) d\gamma(x,y) = \max_{\varphi,\psi} \{ \int_X \varphi(x) d\mu(x) + \int_Y \psi(y) d\nu(y) \} \tag{3.3}$$

从而

$$W_c(\mu_\theta, \nu) = \max_{\varphi,\psi} \left\{ \int_{\mathcal{Z}} \varphi\left(g_\theta(z)\right) d\zeta(z) + \int_{\mathcal{X}} \psi(y) d\nu(y); \varphi(x) + \psi(y) \leq c(x,y) \right\} \tag{3.4}$$

所以优化对象实际为:

$$\min_\theta W_c(\mu_\theta, \nu) = \min_\theta \max_{\varphi,\psi} \left\{ \int_{\mathcal{Z}} \varphi\left(g_\theta(z)\right) d\zeta(z) + \int_{\mathcal{X}} \psi(y) d\nu(y); \varphi(x) + \psi(y) \leq c(x,y) \right\} \tag{3.5}$$

利用 c-concave 函数 φ[16] 则可以写成

$$\min_\theta \max_{\varphi} \left\{ \int_{\mathcal{Z}} \varphi\left(g_\theta(z)\right) d\zeta(z) + \int_{\mathcal{X}} \varphi^{c+}(y) d\nu(y) \right\} \tag{3.6}$$

回顾生成对抗网络的数学公式

$$\min_\theta \max_{\xi} \mathbb{E}_{x\sim\zeta}\left(\varphi_\xi\left(g_\theta(x)\right)\right) + \mathbb{E}_{y\sim\nu}\left(\varphi_\xi^c(y)\right) \tag{3.7}$$

由此可以知道，Kantorovich 势函数 φ^* 实际上是判别器所优化的最终对象，而在实际训练中用神经网络通过参数 ξ 确定函数 φ_ξ 就是一个逼近 φ^* 的过程。我们对于生成对抗网络的训练就是先确定参数 ξ 得到函数 φ_ξ，再根据一次又一次的对抗修正 φ_ξ 最终逼近 φ^* 的过程。

综上所述，从最优传输的角度来看，我们可以通过解 Kantorovich 问题来进行判别器的训练，通过求解给定分布与训练集分布之间的 Wasserstein 距离来进行生成器的训练。

参考文献

[1] Goodfellow, I. J., Generative Adversarial Networks, arXiv e-prints, 2014.

[2] Li, H.. Website: https://speech.ee.ntu.edu.tw/ hylee/mlds/2018-spring.html, 2018

[3] Kullback, S. (1959), Information Theory and Statistics, John Wiley and Sons. Republished by Dover Publications in 1968; reprinted in 1978: ISBN 0-8446-5625-9.

[4] Ôsterreicher, F.; I. Vajda (2003). A new class of metric divergences on probability spaces and its statistical applications. Ann. Inst. Statist. Math. 55 (3): 639–653. doi:10.1007/BF02517812.

[5] Hastie, Trevor. Tibshirani, Robert. Friedman, Jerome. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, NY, 2009.

[6] Brock, A., Donahue, J., and Simonyan, K., Large Scale GAN Training for High Fidelity Natural Image Synthesis, arXiv e-prints, 2018.

[7] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A., Context Encoders: Feature Learning by Inpainting, arXiv e-prints, 2016.

[8] Xianfeng Gu, Feng Luo, Jian Sun, and Shing-Tung Yau. Variational principles for minkowski type problems, discrete optimal transport, and discrete monge-ampere equations. Asian Journal of Mathematics (AJM), 20(2):383 C 398, 2016.

[9] Lei, N., Su, K., Cui, L., Yau, S.-T., and Xianfeng Gu, D., A Geometric View of Optimal Transportation and Generative Model, arXiv e-prints, 2017.

[10] Lei, N., Luo, Z., Yau, S.-T., and Xianfeng Gu, D., Geometric Understanding of Deep Learning, arXiv e-prints, 2018.

[11] Lei, N., Guo, Y., An, D., Qi, X., Luo, Z., Yau, S.-T., and Xianfeng Gu, D., (2019). Mode Collapse and Regularity of Optimal Transportation Maps.

[12] Martin Arjovsky, Soumith Chintala, and L´eon Bottou. Wasserstein GAN. In International Conference on Machine Learning, pages 214–223, 2017.

[13] Arjovsky, M. and Bottou, L., Towards Principled Methods for Training Generative Adversarial Networks, arXiv e-prints, 2017.

[14] Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y., Spectral Normalization for Generative Adversarial Networks, arXiv e-prints, 2018.

[15] Peyré, G. and Cuturi, M., Computational Optimal Transport, arXiv e-prints, 2018.

[16] Santambrogio, F.. (2015) Optimal Transport for Applied Mathematicians. Progress in Nonlinear Differential Equations and Their Applications vol 87. Springer, Berlin, Heidelberg.

[17] Villani, C.. (2008) Optimal Transport-Old and New. Springer Berlin, Heidelberg, NewYork, HongKong, London, Milan, Paris, Tokyo.

[18] Yann Ollivier, Hervé Pajot, Cedric Villani. Optimal Transportation Theory and Applications. Cambridge University Press.

[19] B. Riemann (1867).

[20] Keras, Website: https://keras.io/api/optimizers/rmsprop/

[21] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A., Improved Training of Wasserstein GANs, arXiv e-prints, 2017.