中国科学: 信息科学 2021 年 第 51 卷 第 4 期: 602–617
SCIENTIA SINICA Informationis
论文

生成式对抗网络的不稳定性分析及其处理技术

谭宏卫[1,2], 周林勇[2], 王国栋[1], 张自力[1,3*]

1. 西南大学计算机与信息科学学院, 重庆 400715, 中国
2. 贵州财经大学数统学院, 贵阳 550025, 中国
3. School of Information Technology, Deakin University, Geelong VIC 3220, Australia
* 通信作者. E-mail: zhangzl@swu.edu.cn

收稿日期: 2019–11–14; 修回日期: 2020–01–06; 接受日期: 2020–03–01; 网络出版日期: 2020–11–20

国家自然科学基金 (批准号: 61732019) 资助项目

摘要 生成式对抗网络 (generative adversarial networks, GANs) 训练的不稳定性问题一直是 GANs 研究领域最具挑战性的问题之一. 目前, 仍未从理论上找到影响 GANs 训练稳定性的根本原因及有效的解决办法. 本文通过理论分析发现, GANs 训练的不稳定性主要是由于训练最优判别器与最小化生成器之间相互矛盾所致. 经逐步分析得出, 控制判别器的 Lipschitz 常数是解决 GANs 不稳定性问题的关键, 进而提出一种有针对性的梯度惩罚技术来解决此问题. 最后, 本文从损失函数的振荡幅度 (收敛性)、梯度总体变化趋势, 以及网络整体性能 3 个方面进行了全面对比实验. 结果显示, 本文所提出的惩罚技术对处理 GANs 训练的不稳定性问题具有显著的效果.

关键词 生成式对抗网络, 不稳定性分析, 惩罚技术, 梯度范数, Lipschitz 常数

1 引言

GANs (generative adversarial networks) 是由 Goodfellow 等 [1] 受博弈论中二人零和博弈思想的启发所提出的一种深度生成模型, 其基本网络结构是由判别器网络 (discriminator, D) 和生成器网络 (generator, G) 所构成. 目前, 虽然 GANs 已广泛应用于图像生成 [2,3]、超分辨率图像合成 [4,5]、语义分割 [6,7] 等多个领域, 但是探究 GANs 训练不稳定性问题的根本原因以及彻底地解决这个问题仍有难度. GANs 训练的不稳定性常造成网络不收敛、梯度更新不稳定、网络整体性能较差、生成低质量的样本等问题 [1,8~10].

在不稳定性理论分析方面, Arjovsky 等 [8] 从微分流形的角度, 严格阐述 GANs 训练的不稳定性主要是由于生成分布与目标分布之间的 Jensen-Shannon 散度 [11] 导致生成器梯度消失所造成. 为了解决这个问题, 他们提出用 Wasserstein I 型距离代替 Jensen-Shannon 散度来度量两个分布之间的距离, 进

引用格式: 谭宏卫, 周林勇, 王国栋, 等. 生成式对抗网络的不稳定性分析及其处理技术. 中国科学: 信息科学, 2021, 51: 602–617, doi: 10.1360/SSI-2019-0205
Tan H W, Zhou L Y, Wang G D, et al. Instability analysis for generative adversarial networks and its solving techniques (in Chinese). Sci Sin Inform, 2021, 51: 602–617, doi: 10.1360/SSI-2019-0205

© 2021《中国科学》杂志社

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

而提出 WGAN (Wasserstein generative adversarial network) 算法 [12] 来解决生成器梯度消失问题, 从而使网络训练达到稳定. 此后, Nagarajan 等 [13] 从优化理论的角度证明, 在均衡点附近 WGAN 存在不收敛的极限环. 除此之外, Mescheder 等 [14] 也发现, WGAN 不能收敛到一个局部均衡点, 并且梯度更新不稳定, 即使是在他们构造的一种特殊的网络结构 (Dirac-GAN) 上也不能收敛. WGAN 的不收敛性及其梯度更新的不稳定性表明生成分布与目标分布之间的 Jensen-Shannon 散度并不是造成 GAN 训练不稳定的关键原因.

为了克服 GANs 不稳定性问题, 研究者们通过优化网络结构 [2, 4, 15, 16]、改变梯度更新方法 [13, 17, 18]、修改目标函数 [3, 12, 19~21], 以及对判别器实施正则化 [22~24] 或惩罚 [25, 26] 等技术来提高 GANs 训练的稳定性. 其中, 惩罚技术是效果较好的一类方法. WGAN-GP [25] 首次使用梯度范数惩罚 WGAN 准则函数, 有效缓解 WGAN 中 Lipschitz 条件的限制, 显著提高了 WGAN 的稳定性. 尽管如此, Mescheder 等 [14] 证明零中心梯度惩罚比 1 中心梯度惩罚更容易收敛 (WGAN-GP 是典型的 1 中心梯度惩罚). 沿着这条研究路线, Petzka 等 [27] 开发了零中心梯度惩罚的 WGAN 算法 (WGAN-LP), 与 WGAN-GP 唯一不同之处在于, 其罚函数是非平滑函数. 除此之外, 也有一些研究者将这种惩罚技术应用于原始 GANs 上 [28, 29]. 其中, Thanh-Tung 等 [26] 提出平滑的零中心梯度范数惩罚是效果较好的一个 GANs 惩罚算法 (GAN-0GP), 旨在提高 GANs 判别器的泛化能力, 同时对提高 GANs 的稳定性也有一定的促进作用.

基于对 GNAs 训练的不稳定性理论及其处理技术的观察, 本文通过诱导一个一般化的 GANs 生成器目标函数来对 GANs 的不稳定性问题进行深入分析并发现, GANs 训练的不稳定性主要是由于训练最优判别器和最小化生成器之间相互矛盾所致. 具体地说, 在网络训练过程中会出现这样的矛盾: 当生成器收敛到最小时, 判别器无法收敛到最优判别器; 而当判别器逼近最优判别器的过程中, 生成器的梯度更新不稳定. 经进一步分析得出, 控制判别器的变化范围是解决 GANs 不稳定性问题的关键, 而控制判别器的变化范围又等价于控制判别器的 Lipschitz 常数 [30]. 基于这个思路, 本文利用函数的 Lipschitz 常数与其梯度范数的关系 (定理 2) 以及 GANs 判别器网络的更新动力源自其 Sigmoid 函数的内层函数的事实 (定理 3), 提出一种惩罚这个内层函数的梯度范数技术来有效解决 GANs 训练的不稳定性问题.

综上, 本文的贡献主要有以下 3 点: (1) 诱导一个一般化的 GANs 生成器目标函数; (2) 给出 GANs 训练不稳定性的主要原因; (3) 提出一种有针对性的梯度惩罚技术来有效处理 GANs 训练的不稳定性问题. 本文余下内容安排如下: 第 2 节, 简要介绍与本文相关的 GANs 基础知识; 第 3 节, 详细分析 GANs 训练不稳定的原因; 第 4 节, 提出一种有针对性的不稳定性问题解决方案; 第 5 节, 通过实验验证解决方案的有效性; 第 6 节, 本文小结.

2 基础知识

为便于阅读, 本节简要介绍 GANs 的目标函数及相应的判别器网络和生成器网络的损失函数, 同时以引理的形式给出 3 个已有结论.

从优化理论上讲, GANs 的优化问题是一个典型的极小极大问题, 其目标函数为

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

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

其中 p_{data} 和 p_z 分别表示真实数据分布 (目标分布) 和隐分布 (先验分布). 根据式 (1), GANs 的判别器网络及生成器网络的损失函数分别为

$$L_D = -\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D(G(\boldsymbol{z})))], \tag{2}$$

$$L_{G_1} = \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D(G(\boldsymbol{z})))] \text{ (Saturating)}, \tag{3}$$

$$L_{G_2} = -\mathbb{E}_{\boldsymbol{z}\sim p_z}[\log D(G(\boldsymbol{z}))] \text{ (Non-Saturating)}, \tag{4}$$

其中 L_{G_1} 和 L_{G_2} 分别表示生成器 G 的饱和形式和非饱和形式损失函数. 为了防止梯度消失, 使网络训练更加稳定, Goodfellow 等 [1] 建议用非饱和形式的损失函数 L_{G_2} 来训练模型, 也因此有研究者将原始的 GANs [1] 称为非饱和型 GANs (non-saturating GANs, NSGANs) [31]. 令 $V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D(G(\boldsymbol{z})))]$, 对于最优判别器 $D^\star$, Goodfellow 等 [1] 给出如下引理:

引理1 对于一个给定的生成器 G, 其最优判别器 $D^\star(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$.

其中, $\boldsymbol{x} \in \text{Supp}(\mathbb{P}_{\text{data}}) \cup \text{Supp}(\mathbb{P}_g)$, $\mathbb{P}_{\text{data}}$, $\mathbb{P}_g$ 分别表示密度函数 p_{data}, p_g 所对应的分布函数, Supp$(\cdot)$ 表示分布的支撑. 另一方面, 当 $D(\boldsymbol{x}) = D^\star(\boldsymbol{x})$, 即判别器达到最优判别器时, 此时最小化生成器损失函数等价最小化生成分布与目标分布之间的 Jensen-Shannon 散度, 有如下结论 (见文献 [1] 的定理 1):

引理2 令 $C(G) = \max_D V(D,G)$, 若 $D = D^\star$, 则有 $C(G) = 2\text{JSD}(p_{\text{data}} \parallel p_g) - \log 4$, 且当且仅当 $p_{\text{data}} = p_g$ 时, $C(G)$ 取最小值 $-\log 4$.

其中, JSD$(p_{\text{data}} \parallel p_g)$ 表示目标分布 p_{data} 与生成分布 p_g 的 Jensen-Shannon 散度. 引理 2 说明, 当判别器 D 达到最优判别器 $D^\star$ 时, 最小化生成器 G 的目标函数 $C(G)$, 等价于最小化两个分布之间的 JSD$(p_{\text{data}} \parallel p_g)$, 即有如下关系:

$$\min_G C(G) = \min_G \max_D V(D,G) = \min_G\{2\text{JSD}(p_{\text{data}} \parallel p_g) - \log 4\} \Longleftrightarrow \min_G \text{JSD}(p_{\text{data}} \parallel p_g). \tag{5}$$

因此, GANs 训练的最终目的是最小化生成分布与目标分布之间的 Jensen-Shannon 散度 JSD$(\cdot)$, 迫使生成分布无限逼近目标分布, 生成以假乱真的样本. 尽管如此, 当判别器 D 无法达到最优判别器 $D^\star$ 时, 这个结论是不成立的. 在实际中, 几乎所有的 GANs 模型都是通过增加判别器 D 的更新次数, 使得 D 尽量逼近 $D^\star$, 以此近似满足引理 2 的条件. 而当判别器 D 无法达到最优判别器 $D^\star$ 时, 最小化 $C(G)$ 将演变为何种形式? 在 3.1 小节中将给出, 当判别器 D 无法达到最优判别器 $D^\star$ 时, 最小化 $C(G)$ 等价于同时最小化 JSD$(p_{\text{data}} \parallel p_g)$ 和一个对数函数.

虽然在引理 1 中给出了 $D^\star$ 的计算方式, 但在实际中, 由于 p_{data} 和 p_g 没有显示表达式, 使得 $D^\star$ 根本无法求解, 只是理论上存在; 只有严格当 $p_{\text{data}} = p_g$ 时, $D^\star = \frac{1}{2}$; 同理, 我们根本无法断定 $p_{\text{data}} = p_g$ 是否严格成立, 因此 $D^\star = \frac{1}{2}$ 也仅是一个理论最优值. 尽管如此, $D^\star$ 的唯一性为本文的不稳定性分析提供了理论支撑. 对此, 有如下结论 (见文献 [32] 的定理 2.1).

引理3 设 GANs 的生成分布是被集合 E 上的测度 μ 所控制, 而 E 是定义在 $\mathbb{R}^d$ 上的 Borel 子集, d 表示生成分布的维数. 若对于任意的 $\theta \in \Theta$ 都使得 $p_{g_\theta} > 0$ 几乎处处成立, 则 $D_\theta^\star$ 是使得 $V(D,G_\theta)$ 取得最大的唯一最优判别器, 即有 $\{D_\theta^\star\} = \arg\max_{D\in D_\infty} V(D,G_\theta)$. 其中, Θ 表示生成器网络的参数空间, D_∞ 表示全体判别器集合.

接下来, 利用一个参数 ϵ 将一般判别器 D 和最优判别器 $D^\star$ 联系起来, 以此诱导一个一般化的生成器损失函数, 来探究 GANs 训练不稳定性的原因.

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

3 不稳定性理论分析

本节详细阐述 GANs 不稳定性的分析过程. 通过诱导一个一般化的 GANs 生成器损失函数来深入探究不稳定性的原因.

3.1 一般化的生成器损失函数

在 GANs 的实际训练中, 判别器 D 是否逼近或达到最优判别器 $D^\star$, 并无从判断. 设 $D(\boldsymbol{x}) = \epsilon \cdot D^\star(\boldsymbol{x})$, 其中, $\epsilon \in (0, +\infty)$ 是一个参数, 理论上 ϵ 可以取 0, 此时 $D(\boldsymbol{x}) = 0$, 但对于一个具有判别能力的网络, 这种情况几乎不可能发生; 同样, 下面的推导过程也将证实 ϵ 不可能取到无穷大, 而是一个有界量. 因此, 假设 $\epsilon \in (0, M)$, M 为一个常数. 由 $D^\star(\boldsymbol{x})$ 的唯一性可知, ϵ 既能表示 $D(\boldsymbol{x})$ 逼近 $D^\star(\boldsymbol{x})$ 的程度, 也能衡量判别器 $D(\boldsymbol{x})$ 的判别能力. 显然, 当 $\epsilon = 1$ 时, 判别器 $D(\boldsymbol{x})$ 达到最优. 令 $\eta = \frac{D^\star(\boldsymbol{x})}{1 - D^\star(\boldsymbol{x})}$, 现通过 $D(\boldsymbol{x}) = \epsilon \cdot D^\star(\boldsymbol{x})$ 来诱导一个带参数 ϵ 生成器损失函数, 有如下结论:

定理1 设 $D(\boldsymbol{x}) = \epsilon \cdot D^\star(\boldsymbol{x})$, $\epsilon \in (0, \frac{1+\eta}{\eta})$, 则生成器损失函数 $C_\epsilon(G) = 2\mathrm{JSD}(p_{\mathrm{data}} \parallel p_g) - \log 4 + \log[-\eta\epsilon^2 + (1+\eta)\epsilon]$.

证明 由 $D(\boldsymbol{x}) = \epsilon \cdot D^\star(\boldsymbol{x})$, $\epsilon \in (0, +\infty)$, 则有

$$
\begin{aligned}
C_\epsilon(G) &= \operatorname*{semi\,max}_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}}[\log \epsilon \cdot D^\star(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - \epsilon \cdot D^\star(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}}\left[\log \frac{p_{\mathrm{data}}(\boldsymbol{x})}{p_{\mathrm{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \log\epsilon + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log\left(1 - \epsilon \cdot \frac{p_{\mathrm{data}}(\boldsymbol{x})}{p_{\mathrm{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right)\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}}\left[\log \frac{p_{\mathrm{data}}(\boldsymbol{x})}{p_{\mathrm{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\mathrm{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] \\
&\quad + \log\epsilon + \mathbb{E}_{\boldsymbol{x}\sim p_g} \log\left[1 + (1-\epsilon) \cdot \frac{p_{\mathrm{data}}(\boldsymbol{x})}{p_g(\boldsymbol{x})}\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}} \log D^\star(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x}\sim p_g} \log(1 - D^\star(\boldsymbol{x})) + \log\epsilon + \mathbb{E}_{\boldsymbol{x}\sim p_g} \log\left[1 + (1-\epsilon) \cdot \frac{p_{\mathrm{data}}(\boldsymbol{x})}{p_g(\boldsymbol{x})}\right] \\
&= 2\mathrm{JSD}(p_{\mathrm{data}} \parallel p_g) - \log 4 + \log\epsilon + \mathbb{E}_{\boldsymbol{x}\sim p_g} \log\left[1 + (1-\epsilon) \cdot \frac{p_{\mathrm{data}}(\boldsymbol{x})}{p_g(\boldsymbol{x})}\right].
\end{aligned}
$$

而又由 $\frac{p_{\mathrm{data}}(\boldsymbol{x})}{p_g(\boldsymbol{x})} = \frac{D^\star(\boldsymbol{x})}{(1 - D^\star(\boldsymbol{x}))}$, 于是有

$$
C_\epsilon(G) = 2\mathrm{JSD}(p_{\mathrm{data}} \parallel p_g) - \log 4 + \log[-\eta\epsilon^2 + (1+\eta)\epsilon],
$$

即定理 1 得证.

注释1 在定理 1 的证明中, $\operatorname{semi\,max}_D V(D,G)$ 表示当 G 固定时, 每轮次最优判别器 D 下的目标函数, 这相当于一种阶段性最优. 实际上, 在 GANs 的训练中都是默认每轮更新的判别器最优, 此时最小化生成器的损失函数才等价于最小化两个分布之间的 JSD$(\cdot)$. 但事实上, 在每轮仅有的几次更新中, 不可能使判别器达到最优, 因此用 $\operatorname{semi\,max}_D V(D,G)$ 来刻画这种实际情况. 这与引理 2 给出的完全最优判别器 $D^\star$ 下推导的生成器损失函数 $C(G)$ 是有区别的, 引理 2 要求每轮更新判别器 D 都要达到最优判别器 $D^\star$, 这是一种理想状态, 与实际情况并不吻合. 因此, 定理 1 中的 $C_\epsilon(G)$ 才是符合实际情况的一般性损失函数. 很显然, 当 $\epsilon = 1$ 时, $D(\boldsymbol{x}) = D^\star(\boldsymbol{x})$, $C_\epsilon(G) = C(G)$. 因此, 原始 GANs [1] 中的生成器损失函数 $C(G)$ 只是 $C_\epsilon(G)$ 的特殊情况. 令 $Q(\epsilon) = \log[-\eta\epsilon^2 + (1+\eta)\epsilon]$,

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

则 $C_\epsilon(G) = 2\mathrm{JSD}(p_{\mathrm{data}} \parallel p_g) - \log 4 + Q(\epsilon)$, 即 GANs 的一般化损失函数是由 Jensen-Shannon 散度和一个对数函数 $Q(\epsilon)$ 所构成. 由于 $-\eta\epsilon^2 + (1+\eta)\epsilon > 0$ 可推出, $\epsilon \in (0, \frac{1+\eta}{\eta})$, 而 $\frac{1+\eta}{\eta} = \frac{1}{D^\star(\boldsymbol{x})}$, 因此 ϵ 的实际范围是 $(0, \frac{1}{D^\star(\boldsymbol{x})})$.

注释2 定理 1 给出, 当判别器 D 不能严格等于最优判别器 $D^\star$ 时, 最小化生成器损失函数等价于同时最小化 JSD$(\cdot)$ 和对数函数 $Q(\epsilon)$. 可以看出, 当判别器 D 不能严格等于最优判别器 $D^\star$ 时, 最小化 $C_\epsilon(G)$ 的过程, 仍然在最小化 JSD$(\cdot)$, 但此时会受到函数 $Q(\epsilon)$ 的影响. 因此, 降低函数 $Q(\epsilon)$ 对最小化 JSD$(\cdot)$ 的影响, 可能是提高 GANs 性能的一个重要思路. 在第 4 节中, 本文将按此思路设计一种有效的惩罚算法.

3.2 GANs 训练不稳定性原因

现考虑 GANs 训练的不稳性问题. 由于要考察 ϵ 对最小化 $C_\epsilon(G)$ 的影响, 令 $C(\theta, \epsilon) = 2\mathrm{JSD}_\theta(p_{\mathrm{data}} \parallel p_g) - \log 4 + Q(\epsilon)$, 其中 θ 为生成器 G 的参数表示, 则生成器 G 的更新动力由以下两部分构成:

$$\frac{\partial C(\theta,\epsilon)}{\partial \theta} = \frac{\partial 2\mathrm{JSD}_\theta(p_{\mathrm{data}} \parallel p_g)}{\partial \theta}, \tag{6}$$

$$\frac{\partial C(\theta,\epsilon)}{\partial \epsilon} = \frac{\partial Q(\epsilon)}{\partial \epsilon} = \frac{(1-2\epsilon)\eta + 1}{-\epsilon^2\eta + (1+\eta)\epsilon}, \tag{7}$$

实际上, ϵ 可表示为 θ 的函数形式. 在更新生成器 G 的过程中, $\epsilon = \frac{D(\boldsymbol{x})}{D^\star(\boldsymbol{x})}$, $\boldsymbol{x} = G_\theta(\boldsymbol{z})$, 便于分析, 在此 ϵ 不用表示为 θ 的函数. 虽然引理 3 保证了 $D^*(\boldsymbol{x})$ 的唯一性及引理 1 给出了 $D^*(\boldsymbol{x})$ 的理论求解方程, 但在实际中, 由于 $p_{\mathrm{data}}(\boldsymbol{x})$ 和 $p_g(\boldsymbol{x})$ 没有显示解, $D^*(\boldsymbol{x})$ 仍无法精确求解. 这导致 ϵ 同样无法精确求解. 即便如此, 这并不影响本文对 GANs 训练的不稳定性问题的分析. 下面, 我们分别从判别器的收敛性及生成器梯度更新的稳定性两方面着手, 来探究 GANs 训练不稳定的主要原因.

首先, 考虑当 $C(\theta, \epsilon)$ 取得最小值时, 判别器 D 的变化情况. 如果要使 $C(\theta, \epsilon)$ 取得最小值, 当且仅当 $\mathrm{JSD}_\theta(p_{\mathrm{data}} \parallel p_g)$ 和 $Q(\epsilon)$ 同时取最小. 由上可知, 函数 $Q(\epsilon)$ 是由两层函数构成, 外层是严格递增的对数函数, 内层是关于 ϵ 开口向下的二次函数 ($\eta > 0$); 便于叙述, 令这个二次函数为 $H(\epsilon) = -\eta\epsilon^2 + (1+\eta)\epsilon$. 显然, 由 $H(\epsilon) > 0$ 可推出 $\epsilon \in (0, \frac{1}{D^\star(\boldsymbol{x})})$. 因此, 当 $\epsilon \to 0$ 或 $\epsilon \to \frac{1}{D^\star(\boldsymbol{x})}$ 时, $C(\theta, \epsilon)$ 能收敛到最小值, 但此时 $D(\boldsymbol{x})$ 只能逼近 0 或者 1, 而无法收敛到最优判别器 $D^\star(\boldsymbol{x})$. 在生成器 G 的实际训练过程中, 如果 $D(\boldsymbol{x})$ 取 0 或 1, 则 GANs 模型无法生成高质量的样本.

其次, 考虑当判别器 D 收敛到最优判别器 $D^\star$ 的过程中, 函数 $C(\theta, \epsilon)$ 的梯度变化情况. 由梯度方程 (6) 和 (7) 可知, $\mathrm{JSD}_\theta(p_{\mathrm{data}} \parallel p_g)$ 的参数更新受到 ϵ 的影响, 这意味着生成器并非只更新生成分布与真实分布之间的 Jensen-Shannon 散度, 还在更新 $Q(\epsilon)$ 函数. 在判别器 $D(\boldsymbol{x})$ 逼近最优判别器 $D^\star(\boldsymbol{x})$ 的过程中, 由于 ϵ 的不断变化, 梯度 $\frac{\partial C(\theta,\epsilon)}{\partial \epsilon}$ 也相应地变化, 这将导致生成器的梯度更新不稳定. 尤其是在模型训练之初, 判别器能比较准确地辨识假样本, 即此时 $D(\boldsymbol{x}) \to 0$, $\boldsymbol{x} = G_\theta(\boldsymbol{z})$, $\epsilon \to 0$, 梯度 $\frac{\partial C(\theta,\epsilon)}{\partial \epsilon}$ 趋于无穷大. 因此, 在 GANs 模型训练之初, 生成器梯度的变化幅度是最大的. 在实验中, 本文用平均梯度范数来刻画这种梯度变化幅度, 实验结果也充分地证实了这个事实. 只有当 ϵ 严格等于 1, 即 $D(\boldsymbol{x})$ 完全逼近 $D^\star(\boldsymbol{x})$ 时, ϵ 对 $\mathrm{JSD}_\theta(p_{\mathrm{data}} \parallel p_g)$ 的梯度更新影响最小 (此时式 (7) 中的梯度为 $1-\eta$). 而在实际中, 判别器 $D(\boldsymbol{x})$ 几乎不可能完全逼近最优判别器 $D^\star(\boldsymbol{x})$. 因此, 梯度 $\frac{\partial C(\theta,\epsilon)}{\partial \epsilon}$ 的不断变化是造成生成器梯度更新不稳定的主要原因.

综上可得, GANs 训练的不稳定性是由于判别器 D 和生成器 G 之间的相互矛盾所致, 即在 GANs 的训练中会出现这样的矛盾: 当生成器 G 收敛到最小时, 判别器 D 无法收敛到最优判别器 $D^\star$; 而当

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

判别器 D 逼近最优判别器 $D^\star$ 的过程中, 生成器 G 的梯度更新不稳定. 本文认为, 这才是导致 GANs 训练不稳定的根源所在.

4 不稳定性处理技术

本节提出一种判别器梯度范数惩罚技术来处理 GANs 训练的不稳定性问题. 首先, 简述梯度范数惩罚技术为何能解决 GANs 训练中所出现的矛盾; 其次, 证明原始 GANs 判别器网络的更新动力完全源自于判别器网络中 Sigmoid 函数的内层函数的梯度; 在此基础上, 提出一种惩罚 Sigmoid 函数的内层函数的技术来有效处理 GANs 的不稳定性问题.

4.1 控制判别器的 Lipschitz 常数

由 3.1 小节中的不稳定性分析可知, GANs 训练的不稳定性问题主要是由于函数 $Q(\epsilon)$ 造成判别器和生成器相互矛盾所致. 因此, 降低 $Q(\epsilon)$ 对最小化两个分布之间的 JSD$(\cdot)$ 的影响是解决不稳定性问题的重要思路. 显然, 控制 ϵ 的变化范围, 降低其对 $\mathrm{JSD}_\theta(p_{\mathrm{data}} \parallel p_g)$ 参数更新的干扰, 使得整个模型的梯度更新趋于稳定, 同时使得最小化生成器 G 的过程等价于最小化生成分布与真实分布之间的 Jensen-Shannon 散度. 而又由 $D(\boldsymbol{x}) = \epsilon \cdot D^\star(\boldsymbol{x})$ 及 $D^\star(\boldsymbol{x})$ 的唯一性可知, 控制 ϵ 的变化范围等价于控制判别器 $D(\boldsymbol{x})$ 的变化范围, 因此判别器 $D(\boldsymbol{x})$ 的变化范围是解决 GANs 不稳定性的突破口. 进一步, 控制判别器的变化范围又等价于控制其 Lipschitz 常数 [33].

对于一般的 GANs 判别器网络, Lipschitz 连续性条件并不苛刻. 进一步假设, 判别器损失函数 L_D 关于其网络参数是 C^2 连续的, 则有 $|\min(L_D) - \max(L_D)| \leqslant M_1 \cdot L\sqrt{12mn}$ 及 $|\min(\nabla L_D) - \max(\nabla L_D)| \leqslant M_2 \cdot L\sqrt{12mn}$ 成立 [33]. 其中, m, n, M_1, M_2 分别表示输入图像的高度、宽度, 以及 L_D 在 $\boldsymbol{X}$ 中的一阶和二阶最大导数值, $\boldsymbol{X} = \mathrm{Supp}(\mathbb{P}_{\mathrm{data}}) \cup \mathrm{Supp}(\mathbb{P}_g)$. 由此可见, Lipschitz 常数 L 不但可以控制 $D(\boldsymbol{x})$ 的变化范围, 还可以控制判别器损失函数的变化范围及其梯度变化范围. 因此, 控制判别器的 Lipschitz 常数能促进网络收敛及梯度更新平稳. 综合上述分析, 控制判别器的 Lipschitz 常数才是解决 GANs 不稳定性问题的关键所在. 但在 GANs 中控制判别器网络的 Lipschitz 常数是比较困难的事情. 下面的定理提供了一个可行的方案.

定理2 ([34]) 设 $F : \mathbb{R}^p \longrightarrow \mathbb{R}$ 上的一个 C^1 函数, 其中 $\mathbb{R}^p$ 是 p 维欧式空间, 则 $F(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^p$ 是常数为 L 的 Lipschitz 函数的充要条件是 $\| \|\nabla F\|^2 \|_\infty \leqslant L^2$.

定理 2 给出了一个判定 $F(\boldsymbol{x})$ 是 Lipschitz 函数的方法, 只要其最大的梯度范数的平方小于等于一个常数 L^2, 则 $F(\boldsymbol{x})$ 就是 Lipschitz 常数为 L 的函数. 其次, 定理 2 也给出了这样的事实: Lipschitz 常数可以控制梯度的扩张. 鉴于此, 可将 GANs 判别器的目标函数 L_D 修改为带限制条件 $\| \|\nabla D(\boldsymbol{x})\|^2 \|_\infty \leqslant L^2$ 的目标函数, 即可通过控制 Lipschitz 常数 L 来控制梯度范数的变化, 使得梯度更新稳定, 从而使整个网络训练趋于稳定. 于是, 判别器的目标函数可修改为如下的惩罚形式:

$$L'_D = L_D + \lambda \mathbb{E}_{\boldsymbol{x} \sim p_{\mathrm{data}}}[\max\{\|\nabla D(\boldsymbol{x})\|^2, L^2\}], \tag{8}$$

其中 λ 是惩罚因子. 特别指出的是, 这种惩罚与 WGAN-GP [25] 的惩罚方式完全不同; WGAN-GP 是为了处理 WGAN 中 Lipschitz 条件的限制, 惩罚准则函数的 Lipschitz 常数为 1; 其次, WGAN-GP 使用的罚函数是平滑的二次函数. 而在 L'_D 中, Lipschitz 常数不会限制为 1, 理论上可取任何非负数; 其次, 罚函数是非平滑函数.

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

4.2 惩罚 Sigmoid 函数的内层函数

现考虑 GANs 的具体惩罚形式. 令 $D(\boldsymbol{x}) = \text{Sigmoid}(f(\boldsymbol{x}))$, Sigmoid$(\cdot)$ 函数是网络最后一层的激活函数. 相应地, GANs 的优化问题转化为

$$\min_G \max_f \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log \text{Sigmoid}(f(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1-\text{Sigmoid}(f(G(\boldsymbol{z}))))], \tag{9}$$

判别器 D 与生成器 G 的损失函数也作相应地替换, 其梯度形式用如下定理给出:

定理3 如果 $D(\boldsymbol{x}) = \text{Sigmoid}(f(\boldsymbol{x}))$, 则 GANs 判别器和生成器的损失函数的梯度分别为

$$\nabla_\phi L_D = -\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[(1-D(\boldsymbol{x}))\nabla_\phi f(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[D(\boldsymbol{x})\nabla_\phi f(\boldsymbol{x})], \tag{10}$$

$$\nabla_\theta L_G = -\mathbb{E}_{\boldsymbol{z}\sim p_z}[(1-D(G(\boldsymbol{z})))f(G(\boldsymbol{z}))\nabla_{\boldsymbol{x}} f(G(\boldsymbol{z}))\boldsymbol{J}_\theta G(\boldsymbol{z})], \tag{11}$$

其中, ϕ, θ, $\boldsymbol{J}_\theta G(\boldsymbol{z})$ 分别表示判别器网络参数, 生成器网络参数及 $G(\boldsymbol{z})$ 关于 θ 的 Jacobi 矩阵.

证明 由式 (2) 有

$$\nabla_\phi L_D = -\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\frac{\nabla_\phi D(\boldsymbol{x})}{D(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\frac{\nabla_\phi D(\boldsymbol{x})}{(1-D(\boldsymbol{x}))}\right], \tag{12}$$

而又有

$$\nabla_\phi D(\boldsymbol{x}) = \nabla_\phi(\text{Sigmoid}(f(\boldsymbol{x}))) = \nabla_\phi\left[\frac{1}{1+\exp(-f(\boldsymbol{x}))}\right] = D(\boldsymbol{x})(1-D(\boldsymbol{x}))\nabla_\phi(f(\boldsymbol{x})), \tag{13}$$

将式 (13) 代入式 (12), 则有

$$\nabla_\phi L_D = -\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[(1-D(\boldsymbol{x}))\nabla_\phi f(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[D(\boldsymbol{x})\nabla_\phi f(\boldsymbol{x})]$$

成立. 同理可证, $\nabla_\theta L_G = -\mathbb{E}_{\boldsymbol{z}\sim p_z}[(1-D(G(\boldsymbol{z})))f(G(\boldsymbol{z}))\nabla_{\boldsymbol{x}} f(G(\boldsymbol{z}))\boldsymbol{J}_\theta G(\boldsymbol{z})]$ 成立. 证毕.

从定理 3 可以看出, 真正为 GANs 提供更新动力的是函数 $f(\boldsymbol{x})$ 的梯度, 即 Sigmoid 函数的内层函数的梯度. 由于 Sigmoid 函数的 Lipschitz 常数始终是 $\frac{1}{4}$, 于是控制 $f(\boldsymbol{x})$ 的 Lipschitz 常数等价于控制 $D(\boldsymbol{x})$ 的 Lipschitz 常数, 而控制函数 $f(\boldsymbol{x})$ 的变化范围又等价于控制 $D(\boldsymbol{x})$ 的变化范围. 又据定理 2 及式 (8), 本文将内层函数 $f(\boldsymbol{x})$ 的梯度范数惩罚于判别器损失函数 L_D, 以此控制判别器的变化范围. 此时 GANs 损失函数变成如下形式:

$$L_D^* = L_D + \lambda\mathbb{E}_{\boldsymbol{x}^*\sim p^*}[\max\{\|\nabla f(\boldsymbol{x}^*)\|^2, L^2\}], \tag{14}$$

$$L_{G_2} = -\mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(\text{Sigmoid}(f(\boldsymbol{x})))], \tag{15}$$

其中, $L_D = -\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log(\text{Sigmoid}(f(\boldsymbol{x})))] - \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1-\text{Sigmoid}(f(\boldsymbol{x})))]$, λ 是惩罚因子, L 是被控 Lipschitz 常数. 值得注意的是, 这里的 L 并不受任何限制, 可为任意的非负数. 尽管如此, 一般情况 L 都取较小的值, 以防梯度爆炸. 其中, $\boldsymbol{x}^* = t\boldsymbol{x}+(1-t)\boldsymbol{y}$, $t\sim U(0,1)$, $\boldsymbol{x}\in\text{Supp}(\mathbb{P}_{\text{data}})$, $\boldsymbol{y}\in\text{Supp}(\mathbb{P}_g)$, p^* 为真实分布与生成分布的混合分布.

实际上, 从罚函数 $\max\{\|\nabla f(\boldsymbol{x})\|^2, L^2\}$ (14) 的形式可以看出, Lipschitz 常数控制了函数 $f(\boldsymbol{x})$ 的梯度变化幅度, 使得网络参数更新平缓而不致于出现过度的振荡. 因此, 这种惩罚方式能够有效地处理 GANs 训练的不稳定性问题.

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

5 实验

本文从损失函数的振荡幅度 (收敛性)、梯度的总体变化趋势, 以及网络的整体性能 3 个方面来综合反映 GANs 网络训练的稳定性. 所有模型均用 Pytorch [35] 训练, 训练集选择 5 个标准数据集 (所有图片均剪裁为 $3 \times 32 \times 32$):

• CIFAR-10 [36]. 一个较小的数据集, 由 60000 张 $3 \times 32 \times 32$ 图片所构成, 其中有 50000 张图片构成训练集, 10000 张图片构成验证集, 本文用到的数据集是去标签后的训练集 (50000 张).

• SVHN [37]. 街道门牌号无标签数据集, 总共有 73257 张 3×32×32 图片.

• STL-10 [38]. 抽自于 ImageNet [39] 的图像识别数据集, 大小是 $3 \times 96 \times 96$, 由两类组成: 训练集 5000 张带标签图片和 105000 张无标签图片; 测试集 8000 张带标签图片和 100000 张无标签图片. 本文的所有实验均采用测试集中的 100000 张无标签图片.

• CelebA [40]. 人脸识别数据集, 包含 10177 个名人身份的 202599 张图片.

• LSUN (bedroom) [41]. 一个场景图像数据集, 主要包含卧室、固房、客厅、教堂等场景图像, 本文的所有实验仅使用卧室类的场景图片 LSUN (bedroom), 这个子类的图片总共 3033042 张图片, 每张图片大小不一.

首先, 在 5.1 小节中, 配置 GANs 的网络结构及其参数设置, 包括搜索最优的超参数 λ 和 L; 其次, 在 5.2 小节中, 从损失函数的振荡幅度 (收敛性), 梯度的总体变化趋势两个方面对比 3 个同类算法 (WGAN-GP [25] (2017), WGAN-LP [27] (2018), GAN-0GP [26] (2019)) 来验证本文的惩罚技术对解决 GANs 不稳定性问题的有效性. 特别指出, 由于篇幅的限制, 本小节所有实验结果仅在 CIFAR-10 上训练所得, 其他 4 个数据集上的实验结果见补充材料. 最后, 网络整体性能比较安排在 5.3 小节中; 在 5 个数据集上验证本文的惩罚技术的整体性能, 并与近年来的一些经典算法作对比, 其中包括上述 3 种惩罚型算法和两种非惩罚型算法 (LSGAN [3] (2017), SNGAN [22] (2018)). 网络整体性能的评价指标选择 FID (Fréchet inception distance, FID) 值 [17], FID 值越小, 网络整体性能越好.

5.1 网络结构及其参数设置

网络结构及其相关参数设置. GANs 的网络结构有很多种 [2, 4, 5, 7], 精选某种优秀的网络结构并不是本文实验的重点. 本文的网络结构是在 DCGAN (deep convolutional generative adversarial network) [2] 的基础上改进一种轻量型且高效的结构. 在判别器网络上, 将批量标准化 (batch normalization, BN) [42] 去除, 以及谱标准化 (spectrum normalization, SN) [22] 只加入到 Sigmoid 层的前一层, 而其他层均不使用标准化技术. 在生成器上, 删除所有的 311 不变层 (即卷积核 (kernel) 为 3, 步长 (stride) 为 1, 加边数 (padding) 为 1 的卷积层); 在 Pytorch 中, 311 卷积层不会改变图像的尺寸, 故称为 311 不变层. 具体的网络结构细节可参见附录 A. 除此之外, 网络参数设置是: 两个网络学习率均采用 0.0002, 批量大小是 64, 隐变量 $\boldsymbol{z} \sim N_{128}(\boldsymbol{0}, \boldsymbol{I})$, 判别器使用正交初始化 [43], 生成器使用 Glorot 初始化 [44]. 梯度加速算法使用 OAdam (optimistic Adam) 算法 [45], 其 $\beta_1 = 0.5$, $\beta_2 = 0.9$. 两个网络的更新率均采用 1:1, 即在每轮更新中, 每更新 1 次生成器, 判别器也只更新 1 次, 生成器更新 $100k$ 次.

超参数搜索. 以 FID 值为评价标准进行交叉搜索实验, 最终得出 $\lambda = 10$, $L = 0$ 为最优的超参数. 这一系列搜索实验都是在数据集 CIFAR-10 上进行的, 表 1 是当 $L = 0$ 时, 不同的 λ 对应的 FID 值; 可以看出, 当 $\lambda = 10$ 时, 相应的 FID 值最小, 网络性能最佳. 表 2 是当 $\lambda = 10$ 时, 不同的 L 值对应的 FID 值, $L = 0$ 时效果最佳; 同时表 2 的实验结果显示 L 的取值与 FID 值呈正相关. 另一方面, 为了进一步验证选取 $L = 0$ 的合理性, 现考虑网络损失函数的变化. 图 1 是当 $\lambda = 10$, L 分别取 10, 5,

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

表 1　$L=0$ 时取不同的 λ 及其 FID 值 (CIFAR-10)

Table 1　Different λ and its corresponding FID value when $L=0$ (CIFAR-10)

λ	5	10	20	50	100
FID	15.41	**12.91**	16.28	34.61	32.82

表 2　$\lambda=10$ 时取不同的 L 及其 FID 值 (CIFAR-10)

Table 2　Different L and its corresponding FID value when $\lambda=10$ (CIFAR-10)

L	10	5	1	0.5	0
FID	18.59	18.56	16.47	15.40	**12.91**

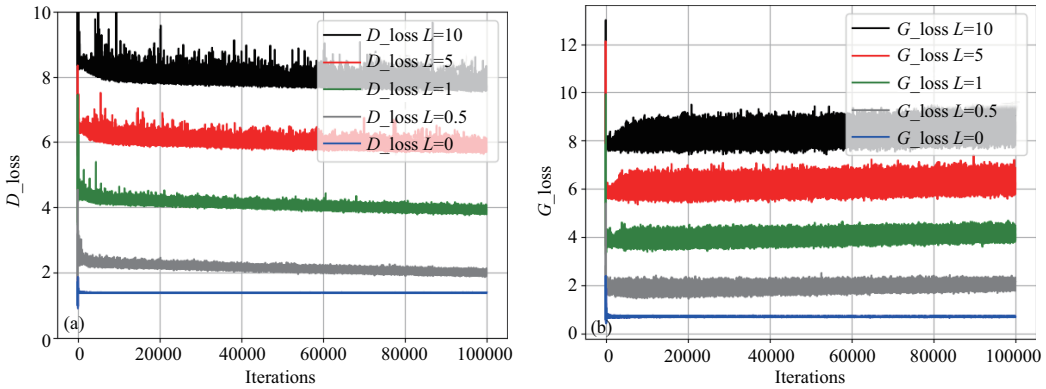


图 1　(网络版彩图) $\lambda=10$ 时不同的 L 值及相应的损失函数变化 (CIFAR-10)

Figure 1　(Color online) Different L value and its corresponding loss function variation when $\lambda=10$ (CIFAR-10). (a) Discriminator loss D_loss; (b) generator loss G_loss

1, 0.5, 0 时, 网络损失函数的变化趋势, 其中图 1(a) 和 (b) 分别表示判别器和生成器损失函数变化, 横轴 (iterations) 是迭代次数, 纵轴 (D_loss, G_loss) 表示损失值. 图 1 显示, 无论是判别器还是生成器损失函数图, 当 $L=0$ 时两个网络的损失函数振荡幅度都最小 (蓝色). 结合表 2 和图 1 的结果, 充分说明 $L=0$ 是最佳的选择, 此时网络训练最平稳, 性能也最佳. 特别指出的是, 在图 1(a) 和 (b) 中, 为了清晰地观察每个 L 值下的损失函数振荡幅度, 除 $L=0$ 外的其他 L 值的损失值均作相应的上下平移. 这里并不关心其收敛值的大小, 只关心反映平稳性的损失函数振荡幅度; 同时, 在同一个坐标系下, 损失值上下平移并不影响相对变化幅度. 后续的对比实验中, 也有类似的平移操作, 以便清晰地观察实验结果. 综上, 这个实验结果再次验证了 Mescheder 等 [14] 关于收敛性证明的结论: 零中心梯度惩罚更容易收敛, 即 $L=0$ 时.

5.2　惩罚技术的有效性验证

目前, 没有硬性的评价指标来衡量 GANs 的稳定性. 一般地, GANs 训练的稳定性与两个网络的损失函数变化, 以及它们的梯度变化密切相关. 具体地说, 一个稳定的 GANs 网络, 其损失函数振荡较小, 趋于收敛; 同时, 梯度变化幅度不大, 总体变化趋势趋于平缓, 不会发生使训练恶化的梯度消失或爆炸现象. 当然, GANs 生成样本的质量也是体现稳定性的重要指标之一. 接下来, 从损失函数的振荡幅度 (收敛性) 及梯度的总体变化趋势角度, 分别选取同类型的 GANs 和 WGAN 惩罚技术作为实验比较对象来验证本文所提出的惩罚技术对解决不稳定性问题的有效性及优越性. 网络整体性能比较实

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

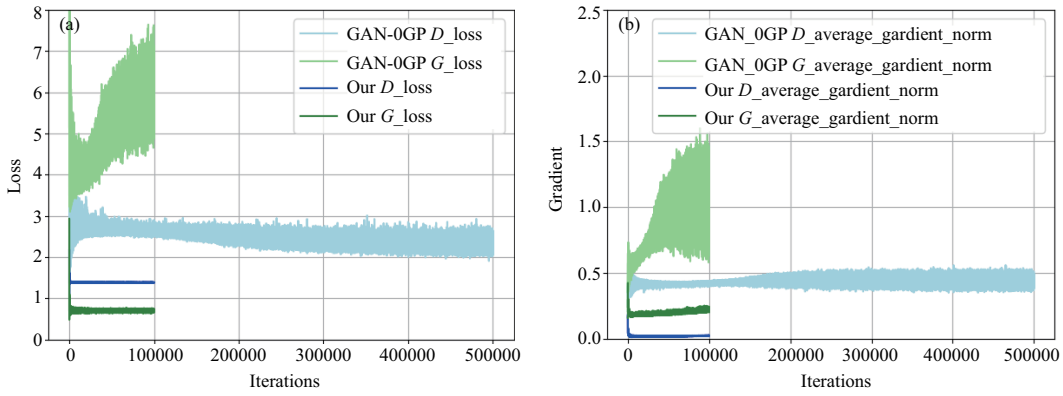


图 2 (网络版彩图) 两种惩罚技术的损失函数变化比较及梯度总体变化趋势比较

Figure 2 (Color online) Comparison of the loss function variation and the general variation trend of gradient between two penalty techniques. (a) The variation of loss function; (b) the general variation trend of gradient

验单独安排在 5.3 小节中.

与同类型的 GAN 惩罚技术比较. 我们选取最近的算法 GAN-0GP [26] 作为比较对象. 下面, 在 CIFAR-10 数据集上分别实验两种惩罚形式, 一种惩罚在 $f(\boldsymbol{x})$ 上 (ours), 另一种惩罚在 Sigmoid($f(\boldsymbol{x})$) 上 (GAN-0GP, 生成器与判别器的更新率是 1:5), 并观察损失函数的收敛性, 梯度的总体变化趋势. 图 2 是两种惩罚形式的损失函数变化 (图 2(a)) 和梯度总体变化趋势 (图 2(b)), 其结果显示, 无论是判别器损失 D_loss 还是生成器损失 G_loss, 本文技术的损失函数振荡幅度显著小于 GAN-0GP 网络的损失函数振荡幅度; 同样, 在梯度的总体变化趋势上, 本文网络的梯度总体变化趋势也显著比 GAN-0GP 网络梯度变化趋势平缓, 这证实了本文的技术能更有效地防止梯度消失或爆炸, 同时也证实了本文技术的梯度更新明显比 GAN-0GP 更加稳定. 从上述两个方面的实验验证, 惩罚 Sigmoid 函数的内层函数 $f(\boldsymbol{x})$ 比直接惩罚 Sigmoid 函数更有效, 极大地提高了 GANs 训练的稳定性. 特别指出的是, 这里是用每次迭代的平均梯度范数 (在一个小批量样本内计算平均梯度范数) 来反映每次迭代的梯度总体变化趋势.

与同类型的 WGAN 惩罚技术比较. 这个实验选取两种典型的 WGAN 惩罚技术作比较对象, 一个是 1 中心梯度惩罚 WGAN-GP [25], 另一个是 0 中心梯度惩罚 WGAN-LP [27]. 同样在 CIFAR-10 数据集上训练, 其结果分别显示在图 3 和 4 中, 前者是 3 种算法的损失函数变化趋势比较, 后者是 3 种算法的梯度总体变化趋势比较. 在损失函数图 3 中, 无论是判别器还是生成器损失函数, 我们的算法的损失函数振荡幅度明显小于 WGAN-GP 和 WGAN-LP 的损失函数振荡幅度. 需要指出的是, 在 WGAN 中充当判别器功能的函数称为准则函数 (critic function) [12], 其优化的目标函数并不是严格意义上的损失函数, 因此在损失函数图 3 中有一些损失值是负值. 但是这完全不影响我们对损失函数振荡性结果的判断. 同样, 在梯度的总体变化趋势上 (图 4), 无论是判别器 (图 4(a)) 还是生成器 (图 4(b)) 的梯度变化, 本文技术的梯度变化幅度及平缓趋势也显著优于 WGAN-GP 和 WGAN-LP. 这个实验再一次充分验证, 本文的技术对处理 GANs 不稳定性问题有极好的效果. 特别指出的是, 由于 WGAN-GP, WGAN-LP 的更新率是 5:1, 而本文的更新率是 1:1, 因此 WGAN-GP 和 WGAN-LP 的判别器损失函数值的数量是本文网络的判别器损失值的 5 倍 (图 3(a)), 判别器的梯度值同样如此 (图 4(a)). 除此之外, 与 WGAN 的两个惩罚技术的对比实验还证实了生成分布与目标分布之间的距离度量方式并不是导致 GANs 不稳定性的主因, 而 GANs 网络判别器和生成器之间的相互矛盾才是导致其不稳定性的

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

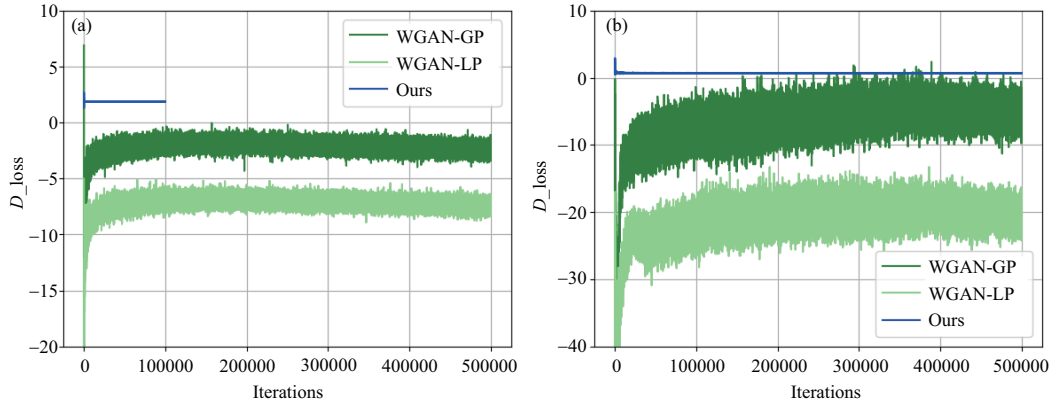


图 3　(网络版彩图) 3 种算法的损失函数变化

Figure 3　(Color online) The variation of loss function in three algorithms. (a) The variation of loss function in discriminator (critic function); (b) the variation of loss function in generator

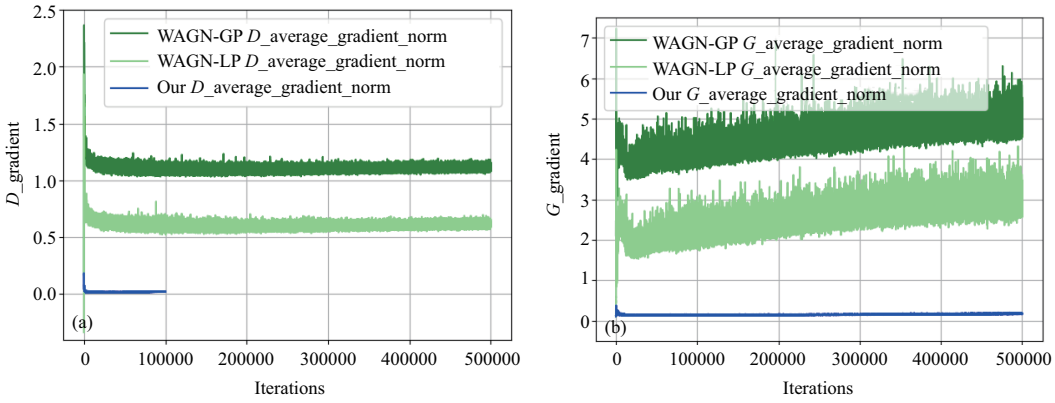


图 4　(网络版彩图) 3 种算法的梯度总体变化趋势

Figure 4　(Color online) The general variation trend of gradient in three algorithms. (a) The general variation trend of gradient in discriminator (critic function); (b) the general variation trend of gradient in generator

关键原因.

除与上述两类模型的比较之外, 我们从图 2(b) 和 4(b) 的实验结果发现: 在训练之初, 所有模型的生成器梯度的振荡幅度达到最大; 随着训练次数的增加, 模型越来越好, 振幅逐步稳定. 由 3.2 小节的不稳定性分析可知, 训练开始时 $\epsilon \to 0$, 梯度 (7) 趋于无穷大, 这导致此时的梯度变化幅度异常大. 随着训练次数增加, 虽然 GAN-0GP (图 2(b) 浅绿色曲线), WGAN-GP (图 4(b) 绿色曲线), WGAN-LP (图 4(b) 浅绿色曲线) 的生成器梯度变化逐步稳定在一个范围内, 但仍有较大的变化幅度影响梯度更新的稳定性. 而本文的模型不但能使生成器梯度趋于稳定, 而且还能控制在一个很小的范围内, 甚至在训练之初也能较好地控制梯度变化范围; 这取决于对 ϵ 的有效控制, 大幅地降低对生成器梯度的扰动 (实际上, 这种控制也有利于判别器梯度更新的稳定性, 图 4(a) 蓝色曲线), 从而使整个模型的训练趋于稳定. 这充分地证实, 通过控制判别器的 Lipschitz 常数来控制 ϵ 的变化范围, 能有效地解决 GANs 训练的不稳定性问题.

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

表 3 算法性能比较 (FID 值越小越好)

Table 3 Algorithms performance comparison (The smaller the FID value, the better)

	CIFAR-10	SVHN	STL-10	CelebA	LSUN (bedroom)
LSGAN (2017)	22.20	3.84	20.17	5.10	**5.23**
SNGAN (2018)	20.70	4.53	18.11	5.56	12.05
WGAN-GP (2017)	21.89	4.09	18.19	5.01	14.61
WGAN-LP (2018)	21.01	3.62	17.40	5.12	15.21
GAN-0GP (2019)	18.91	6.10	14.49	4.53	7.14
Ours	**12.91**	**2.78**	**12.85**	**3.91**	6.85
Real datasets	0.46	0.24	0.84	0.34	0.55

5.3 整体性能比较

GANs 的整体性能也是衡量其稳定性的重要指标之一. 可以肯定的是, GANs 生成样本的质量与其稳定性也密切相关. 本文用 FID 来衡量 GAN 的整体性能, FID 值越小, 性能越高, 生成样本的质量也越好, 其网络也越稳定. 接下来, 我们在 CIFAR-10, SVHN, STL-10, CelebA 及 LSUN (bedroom) 数据集上实验多种算法, 生成器都更新 $100k$ 次, 并计算相应的 FID 值. 其中, LSGAN, SNGAN 是无惩罚的算法, 其余的都是有惩罚的算法, 实验结果显示在表 3 中. 其中, 表 3 中最后一行是真实数据的 FID 值. 可以看出, 无论是有惩罚技术的还是无惩罚技术的算法, 本文提出的算法在多个数据集上都显著优于表 3 中的其他算法 (只有在 LSUN (bedroom) 数据集上表现略差于 LSGAN, GAN-0GP), 这说明本文提出的算法大幅地提高了 GANs 的整体性能. 如果只从整体性能上评价稳定性, 毫无疑问, 本文的算法极大地提高了 GANs 训练的稳定性. 同时, 表 3 的实验结果还表明本文的算法更有助于提高 GANs 生成样本的质量. 特别指出的是, 为了减小 FID 值的计算误差, 本文从真实样本和生成样本中各抽取 50000 个样本来计算 FID 值.

结合 5.2 和 5.3 小节的实验结果, 无论是在 GANs 网络损失函数收敛性及梯度总体变化趋势上还是在网络整体性能上, 本文所提出的惩罚技术的网络稳定性显著优于其他算法, 这充分证实了本文的技术对解决 GANs 的不稳定性问题具有显著的效果.

6 结束语

本文从一般化生成器损失函数的角度, 对 GANs 训练的不稳定性进行深入分析并得出, 影响 GANs 稳定性的关键原因是由于训练最优判别器与最小化生成器相互矛盾所致. 进一步分析得出, 控制判别器的 Lipschitz 常数是解决 GANs 训练不稳性问题的突破口. 为此, 本文利用判别器网络的 Lipschitz 常数及其 Sigmoid 函数的内层函数的梯度范数之间的关系构建一个零中心梯度惩罚模型来有效地处理 GANs 训练的不稳定性问题. 在多个数据集上实验多个 GANs 算法, 其结果显示, 无论是从损失函数的收敛性及梯度总体变化趋势还是从网络整体性能上, 本文所提出的技术都显著优于其他算法. 这充分证实了本文所提出的惩罚技术对处理 GANs 的不稳定性问题具有显著的效果.

虽然收敛性问题不是本文研究的重点, 但是实验结果表明, 本文所提出的惩罚技术的网络损失函数收敛较好, 尤其是在判别器网络上几乎收敛到一个常数上. 因此, 深入探究本文技术的收敛性理论依据将是下一步研究重点.

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

参考文献

1 Goodfellow I, Pougetabadie J, Mirza M, et al. Generative adversarial nets. In: Proceedings of the 28th Conference on Neural Information Processing Systems (NeurIPS), Montréal, 2014. 2672–2680
2 Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. In: Proceedings of the 4th International Conference on Learning Representations (ICLR), San Juan, 2016
3 Mao X D, Li Q, Xie H R, et al. Least squares generative adversarial networks. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), Venice, 2017
4 Karras T, Aila T, Laine S, et al. Progressive growing of GANs for improved quality, stability, and variation. In: Proceedings of the 6th International Conference on Learning Representations (ICLR), Vancouver, 2018
5 Brock A, Donahua J, Simonyan K. Large scale GANs training for high fidelity natural image synthesis. In: Proceedings of the 7th International Conference on Learning Representations (ICLR), Louisiana, 2019
6 Luc P, Couprie C, Chintala S, et al. Semantic segmentation using adversarial networks. In: Proceedings of the 30th Conference on Neural Information Processing Systems (NeurIPS), Barcelona, 2016
7 Zhu W T, Xiang X, Tran T D, et al. Adversarial deep structural networks for mammographic mass segmentation. In: Proceedings of the 15th IEEE International Symposium on Biomedical Imaging, Washington, 2018
8 Arjovsky M, Bottou L. Towards principled methods for training generative adversarial networks. In: Proceedings of the 5th International Conference on Learning Representations (ICLR), Toulon, 2017
9 Wiatrak M, Albrecht S V. Stabilizing generative adversarial network training: a survey. 2019. ArXiv:1910.00927
10 Durall R, Pfreundt F, Keuper J. Stabilizing GANs with octave convolutions. 2019. ArXiv:1905.12534
11 Grey R. Entropy and Information Theory. Berlin: Springer, 2013
12 Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, 2017
13 Nagarajan V, Kolter J Z. Gradient descent GAN optimization is locally stable. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS), California, 2017
14 Mescheder L, Geiger A, Nowozin S. Which training methods for GANs do actually converge? In: Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, 2018
15 Sharma M, Makwana M, Upadhyay A, et al. Robust image colorization using self attention based progressive generative adversarial network. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Califonia, 2019
16 Salimans T, Goodfellow I, Zaremba W, et al. Improved techniques for training GANs. In: Proceedings of the 30th Conference on Neural Information Processing Systems (NeurIPS), Barcelona, 2016
17 Heusel M, Ramsauer H, Unterthiner T, et al. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS), California, 2017
18 Metz L, Poole B, Pfau D, et al. Unrolled generative adversarial networks. In: Proceedings of the 5th International Conference on Learning Representations (ICLR), Toulon, 2017
19 Su Y X, Zhao S L, Chen X X, et al. Parallel wasserstein generative adversarial nets with multiple discriminators. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), Macao, 2019
20 Dam N, Hoang Q, Le T, et al. Three-player wasserstein GAN via amortised duality. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), Macao, 2019
21 Liu H D, Gu X F, Samaras D. Wasserstein GAN with quadratic transport cost. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), Seoul, 2019
22 Miyato T, Kataoka T, Koyama M, et al. Spectral normalization for generative adversarial networks. In: Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS), Montréal, 2018
23 Zhang H, Zhang Z Z, Odena A, et al. Consistency regularization for generative adversarial networks. 2019. ArXiv:1910.12027
24 Kurach K, Mario L, Zhai X H, et al. A large-scale study on regularization and normalization in GANs. In: Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, 2019
25 Gulrajani I, Ahmed F, Arjovsky M, et al. Improved training of Wasserstein GANs. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS), California, 2017
26 Thanh-Tung H, Tran T, Venkatesh S. Improving generalization and stability of generative adversarial networks. In: Proceedings of the 7th International Conference on Learning Representations (ICLR), Louisiana, 2019
27 Petzka H, Fischer A, Lukovnicov D. On the regularization of wasserstein GANs. In: Proceedings of the 6th International

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

Conference on Learning Representations (ICLR), Vancouver, 2018
28 Roth K, Lucchi A, Nowozin S, et al. Stabilizing training of generative adversarial networks through regularization. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS), California, 2017
29 Edraki M, Qi G J. Generalized loss-sensitive adversarial learning with manifold margins. In: Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, 2018
30 Dudley R M. Real Analysis and Probability. Cambridge: Cambridge University Press, 2004
31 Fedus W, Rosca M, Lakshminarayanan B, et al. Many paths to equilibrium: GANs do not need to decrease a divergence at every step. In: Proceedings of the 6th International Conference on Learning Representations (ICLR), Vancouver, 2018
32 Biau G, Cadre B, Sangnier M, et al. Some theoretical properties of GANs. 2018. ArXiv:1803.07819v1
33 Qin Y P, Mitra N, Wonka P. How does Lipschitz regularization influence GANs training? 2019. ArXiv:1811.09567v2
34 Handel R V. Probability in high dimension. 2016. https://web.math.princeton.edu/~rvan/APC550.pdf
35 Paszke, Gross A, Chintala S, et al. Automatic differentiation in PyTorch. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS) Workshop, California, 2017
36 Krizhevsky A. Learning multiple layers of features from tiny images. Dissertation for M.S. Degree. Toronto: University of Toronto, 2009
37 Netzer Y, Wang T, Coates A, et al. Reading digits in natural images with unsupervised feature learning. In: Proceedings of the 25th Conference on Neural Information Processing Systems (NeurIPS), Granada, 2011
38 Coates A, Lee H, Andrew Y. An analysis of single layer networks in unsupervised feature learning. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Florida, 2011
39 Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. Int J Comput Vis, 2015, 115: 211–252
40 Liu Z W, Luo P, Wang X G, et al. Deep learning face attributes in the wild. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), Santiago, 2015
41 Yu F, Seff A, Zhang Y D, et al. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. 2015. ArXiv:1506.03365
42 Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, 2015
43 Saxe A, Mcclelland L, Ganguli S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In: Proceedings of the 2nd International Conference on Learning Representations (ICLR), Banff, 2014
44 Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), Sardinia, 2010
45 Daskalakis C, Ilyas A, Syrgkanis V, et al. Training GANs with optimism. In: Proceedings of the 6th International Conference on Learning Representations (ICLR), Vancouver, 2018

附录 A 改进的 DCGAN 网络结构

表 A1 判别器, 图像尺寸是 3×32×32

Table A1 Discriminator, the image size is 3×32×32

Convolution	Out-channels	Kernel	Stride	Padding	Normalization/activation	Bias
conv	64	3×3	1	1	LReLU(0.2)	True
conv	64	4×4	2	1	LReLU(0.2)	False
conv	128	3×3	1	1	LReLU(0.2)	False
conv	128	4×4	2	1	LReLU(0.2)	False
conv	256	3×3	1	1	LReLU(0.2)	False
conv	256	4×4	2	1	LReLU(0.2)	False
conv	512	3×3	1	1	LReLU(0.2)	False
conv	1	4×4	1	0	SN	True
–	–	–	–	–	Sigmoid()	–

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

表 A2 生成器, 图像尺寸是 3×32×32

Table A2 Generator, the image size is 3×32×32

ConvTranspose	Out-channels	Kernel	Stride	Padding	Normalization/activation	Bias
convT	512	4×4	1	0	BN ReLU	False
convT	256	4×4	2	1	BN ReLU	False
convT	128	4×4	2	1	BN ReLU	False
convT	64	4×4	2	1	BN ReLU	False
convT	3	3×3	1	1	–	True
–	–	–	–	–	Tanh()	–

Instability analysis for generative adversarial networks and its solving techniques

Hongwei TAN[1,2], Linyong ZHOU[2], Guodong WANG[1] & Zili ZHANG[1,3*]

1. *School of Computer and Information Sciences, Southwest University, Chongqing* 400715*, China*;
2. *School of Mathematics and Statistics, Guizhou University of Finance and Economics, Guiyang* 550025*, China*;
3. *School of Information Technology, Deakin University, Geelong* VIC 3220*, Australia*
* Corresponding author. E-mail: zhangzl@swu.edu.cn

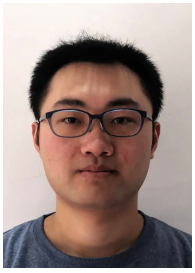
Abstract Training instability in generative adversarial networks (GANs) remains one of the most challenging problems, for which both the theoretical root and an effective solution are needed. In this study, we theoretically determined that the mutual contradiction between training the optimal discriminator and minimizing the generator leads to training instability in GANs. To address this problem, we propose a targeted gradient penalty technique. Unlike other penalty techniques, we penalize the Lipschitz constant of the discriminator, which is the key to dealing with the instability problem (this amounts to controlling the Lipschitz constant of the discriminator). We performed a series of experimental comparisons from three different perspectives: the oscillation amplitude of the loss function (convergence), the general variation trend of the gradient, and the holistic performance of the network. The results demonstrated that the proposed technique has a significant and positive effect on the training instability in GANs.

Keywords generative adversarial networks, instability analysis, penalty technique, gradient norm, Lipschitz constant

https://engine.scichina.com/doi/10.1360/SSI-2019-0205

Hongwei TAN was born in 1983. He received his B.S. degree in mathematics and applied mathematics and M.S. degree in probability theory and mathematical statistics, all from Guizhou Minzu University, Guiyang, China, in 2006 and 2013, respectively. Currently, He is a lecturer at Guizhou University of Finance and Economics, Guiyang, China and is pursuing his Ph.D. degree at the College of Computer and Information Science, Southwest University. His research interests include deep learning, applied statistics, and big data analysis.

Linyong ZHOU was born in 1987. He is a lecturer at Guizhou University of Finance and Economics, Guiyang, China. He received his B.S. degree and M.S. degree in computer science from Guizhou Normal University, Guiyang, China, in 2010 and 2013, respectively. Currently, he is pursuing his Ph.D. degree at the College of Big Data and Computer Science, Guizhou Normal University. His research interests include machine learning and data mining.



Guodong WANG was born in 1990. He received his B.S. degree and M.S. degree from the College of Computer and Information Science, Southwest University, Chongqing, China, in 2012 and 2016, respectively. Currently, he is pursuing his Ph.D. degree at the College of Computer and Information Science, Southwest University. His research interests include deep learning and knowledge graph.

Zili ZHANG was born in 1964. He is a professor at Southwest University, Chongqing, China. He received his B.S. degree from Sichuan University in 1986, M.E. degree from Harbin Institute of Technology in 1989, and Ph.D. degree from Deakin University in 2001, all in computing. His research interests include bio-inspired artificial intelligence, agent-based computing, and big data analysis.

https://engine.scichina.com/doi/10.1360/SSI-2019-0205