深度生成模型 Course Project 报告

对 Least Squares GAN 的思考与改进

北京大学信息科学技术学院 李喆琛 1901111292
北京大学数学科学学院 初济群 1901210090

2020 年 5 月 28 日

Outline

关于 GAN 的回顾

GAN 的目标函数

GAN 有一个生成器 G 和一个判别器 D:

$$\min_G \max_D V_{GAN}(D, G) = \mathbb{E}_{x\sim P_r}[\log(D(x))] + \mathbb{E}_{z\sim P_g}[\log(1 - D(G(z)))]$$

其中，P_r 表示原始数据的分布，P_g 表示生成数据的分布.

GAN 的一个问题：梯度消失

在 GAN 的训练过程中，如果 D 足够优秀，那么 G 的梯度将会消失，进而停止迭代，使得 GAN 的生成质量不高.

Least Squares GAN

LSGAN 的优化目标

$$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{x\sim P_r}[(D(x)-b)^2] + \frac{1}{2}\mathbb{E}_{z\sim P_g}[(D(G(z))-a)^2]$$
$$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z\sim P_r}[(D(G(z))-c)^2]$$

其中，a 表示假数据，b 表示真实数据，c 表示 G 希望被 D 判别为真的数据.

LSGAN 的思路：为什么传统的 GAN 会造成梯度消失？

- 交叉熵 Sigmoid 损失函数
  - 只关注生成样本的分类，而不关注生成样本与决策边界之间的距离.
  - 这会导致一些距离决策边界很远的样本通过检测 $\rightarrow$ 梯度消失.

解决方法：最小二乘函数 $\rightarrow$ 迫使离群点向决策边界移动.

WGAN

由于 f-散度在一定情况下会产生突变，在 GAN 的训练中产生梯度消失的现象，Martin Arjovsky 等人采用了更为平滑的 Wasserstein 度量来计算分布之间的距离：

$$W(P_1, P_2) = \inf_{\gamma \sim \Pi(P_1, P_2)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||]$$

作者对上式进行了一个变换：

$$W(P_r, P_g) = \frac{1}{K} \sup_{\|f\|_L \leqslant K} \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{x\sim P_g}[f(x)]$$

WGAN 的目标函数是根据 Kantorovich Rubinstein 对偶理论建立的：

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{x\sim P_r}[D(x)] - \mathbb{E}_{z\sim P_g}[D(G(z))]$$

WGAN-GP

北京大學 PEKING UNIVERSITY

WGAN 针对 D 所进行的 weight clipping 虽然可以迫使 D 满足 Lipschitz 条件，但这个方法会导致参数过于集中在两个顶点处，在实验中不够稳定. 针对这种情况，Ishaan 等人提出，可以通过增加 gradient penalty 的方法来取代 weight clipping，从而使得 D 满足 Lipschitz 条件 [5].
WGAN-GP 的方法在实验中效果极佳，是现在 state of the art 的方法.

$$L = \mathbb{E}_{z\sim P_g}[D(G(z))] - \mathbb{E}_{\bar{x}\sim P_r}[D(\bar{x})] + \lambda \mathbb{E}_{\tilde{x}\sim P_{\tilde{x}}}[(\nabla_{\tilde{x}} D(\tilde{x})_2 - 1)^2]$$

Our Method I

我们首先从 WGAN 文中的思路出发来分析 LSGAN 的目标函数:

$$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{x\sim P_r}[(D(x)-b)^2] + \frac{1}{2}\mathbb{E}_{z\sim P_g}[(D(G(z))-a)^2]$$

$$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z\sim P_r}[(D(G(z))-c)^2]$$

不失一般性，设 $c=0$. 对 $V_{LSGAN}(D)$ 进行简单分析即可得到 LSGAN 的最优判别器为:

$$D^* = \frac{bP_r + aP_g}{P_r + P_g}$$

回带入 $V_{LSGAN}(G)$ 中并进行简单变换，可得:

$$V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{x\sim P_r}[D^*(x)^2] + \frac{1}{2}\mathbb{E}_{z\sim P_g}[D^*(z)^2]$$

Our Method II

此时，如果 P_r 和 P_g 的支集（记为 $\textbf{supp } P_r$, $\textbf{supp } P_g$）为高维空间中的低维流形，将有：

$$\mathbb{P}[\mu(\textbf{supp } P_r \cap \textbf{supp } P_g) = 0] = 1$$

换言之，二者支集的交集很可能是一个零测集. 那么给定一个 x，有且仅有如下 4 种情况之一发生：

1. $P_d(x) = 0,\ P_g(x) = 0;$
2. $P_d(x) \neq 0,\ P_g(x) = 0;$
3. $P_d(x) = 0,\ P_g(x) \neq 0;$
4. $P_d(x) \neq 0,\ P_g(x) \neq 0.$

Our Method III

其中第 1 中情况对目标函数无贡献；第 2、3 种情况下 V_{LSGAN} 在一定的邻域里都是常数，因而对梯度的贡献为 0；而由前文的断言又可知，第 4 种情况不会几乎发生. 因此在该情形下，LSGAN 也无法从理论上避免梯度消失的问题.
针对这个问题，我们给 LSGAN 的生成器 G 添加了一个正则项：

$$
\begin{aligned}
V_{LSGAN'}(G) &= \frac{1}{2}\mathbb{E}_{z\sim P_g}[(D(G(z)) - c)^2] + \lambda(\mathbb{E}_{z\sim P_g}D(G(z)) - \mathbb{E}_{x\sim P_r}[D(x)])^2 \\
&= V_{LSGAN}(G) + \lambda(\mathbb{E}_{z\sim P_g}D(G(z)) - \mathbb{E}_{x\sim P_r}[D(x)])^2
\end{aligned}
$$

根据上面的讨论，当 $\textbf{supp}\ P_r$ 与 $\textbf{supp}\ P_g$ 在判别器 D 的识别下差异很大时，我们有：

$$
\lim_{D\to D^*} \nabla_x V_{LSGAN}(G) = 0.
$$

Our Method IV

北京大学 PEKING UNIVERSITY

因此我们可以得到:

$$\lim_{D\to D^*} \nabla_x V_{LSGAN'}(G) = \lim_{D\to D^*} \lambda\nabla_x[(\mathbb{E}_{z\sim P_g} D(G(z)) - \mathbb{E}_{x\sim P_r}[D(x)])^2] \neq 0.$$

即: 添加该正则项可以在理论上避免该情形下的梯度消失.
另一方面，添加正则项不会破坏原先 LSGAN 的最优化条件，因此 [3] 中针对 LSGAN 的下述定理仍然会成立:

定理 ([3])

若 *LSGAN* 中的参数 a, b, c 满足 $b - c = 1$ 且 $b - a = 2$，则其优化过程会最小化 $P_r + P_g$ 与 $2P_g$ 之间的 *Pearson* χ^2 散度 $\chi^2_{Pearson}(P_r + P_g \| 2P_g)$.

Our Method V

即我们的新模型在 $b-c=1$ 且 $b-a=2$ 时，仍然会优化 Pearson χ^2 散度 $\chi^2_{\text{Pearson}}(P_r+P_g\|2P_g)$. 如此便保证了收敛性.
关于参数的选择，同样也有与 LSGAN 相似的两种选择：

- $b-c=1$, $b-a=2$：优化 Pearson χ^2 散度；
- $b=c$：生成尽可能真实的样本.

Evaluation

北京大学 PEKING UNIVERSITY

Dataset

我们在参数 $(1,-1,0)$ 下对 MNIST 数据集进行了实验：MINST 数据集小，程序收敛快，我们可以更快速的看到结果. 为了进行对照，我们选用 DCGAN 的 architecture[6]，使用 tensorlayer 作为程序底层架构，在单片 GeForce RTX 2080 Ti 上进行实验.

Hyperparameter

我们设定 batch_size 为 64，训练的 Epoch 为 20，初始学习率为 $1e^{-4}$，z 的维度为 100，采样个数和 batch_size 相同，为 64. 对于非 WGAN 的模型，我们都采用 Adam 优化器，并设定 Momentum term 为 0.5，对于 WGAN，我们采用了 RMSPROP.

Evaluation

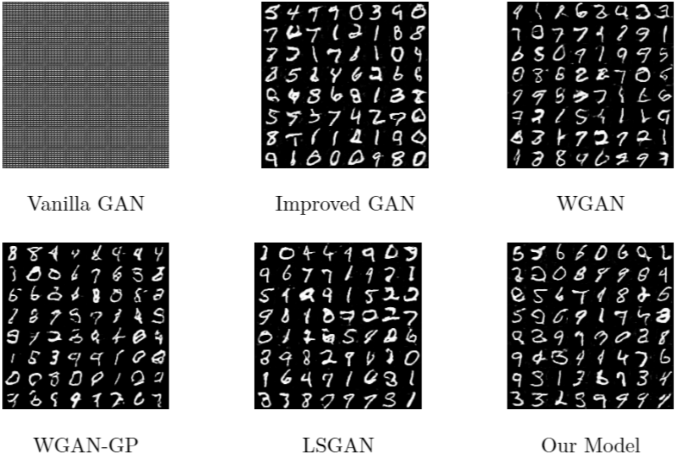

图 1: 几种模型在 20 个 Epoch 之后的图片生成效果对比

Evaluation

北京大学 PEKING UNIVERSITY

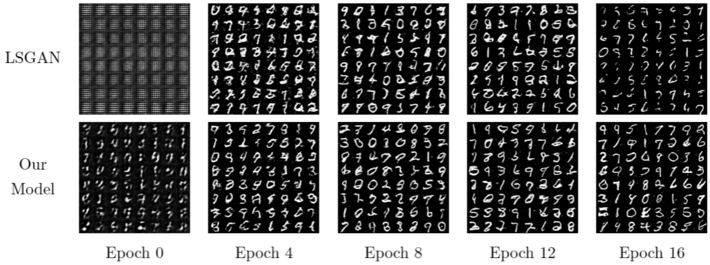

图 2: LSGAN 和我们的新模型在不同 Epoch 的对比

Thanks For Listening!

References I

- Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets [C]// *Advances in Neural Information Processing Systems (NIPS)*. 2014.
- Mao X, Li Q, Xie H, et al. Least squares generative adversarial networks [C]// *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017.
- Mao X, Li Q, Xie H, et al. On the effectiveness of least squares generative adversarial networks [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- Arjovsky M, Chintala S, Bottou L. Wasserstein Generative Adversarial Networks [C]// *International Conference on Machine Learning (ICML)*. 2017.

References II

北京大学 PEKING UNIVERSITY

Gulrajani I, Ahmed F, Arjovsky M, et al. Improved training of wasserstein gans [C]// *Advances in Neural Information Processing Systems (NIPS)*. 2017.

Radford, A., Metz, L., Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks [C]// *The International Conference on Learning Representations (ICLR)*. 2016.