原始 GAN 到 WGAN-GP 的演化

1 原始GAN（2014）

原始WGAN在训练时非常困难，表现出了以下两个主要问题：

1.1 梯度消失(判别器优化得越好，越容易发生梯度消失问题)

第一个问题的病灶在于原始损失函数的缺陷，原始GAN中，判别器要最小化如下的损失函数：

$$-\mathbb{E}_{x\sim P_r}[\log D(x)] - \mathbb{E}_{x\sim P_g}[\log(1-D(x))]$$

其中P_r是真实样本分布，P_g是生成器生成的样本分布，判别器的目标即为尽可能分真实样本为正，生成样本为负。
代入一个具体的x，有：

$$-P_r(x)\log D(x) - P_g(x)\log[1-D(x)]$$

其中，$P_r(x)$表示x来自P_r的概率。令上式关于$D(x)$的导数为0，有：

$$-\frac{P_r(x)}{D(x)} + \frac{P_g(x)}{1-D(x)} = 0$$

化简得到最优判别器：

$$D^*(x) = \frac{P_r(x)}{P_r(x)+P_g(x)}$$

而对于生成器，损失函数为：

$$\mathbb{E}_{x\sim P_g}[\log(1-D(x))]$$

对这个损失函数添加一项，有：

$$\mathbb{E}_{x\sim P_r}[\log D(x)] + \mathbb{E}_{x\sim P_g}[\log(1-D(x))]$$

由于新添加的项不影响表达式关于$D(x)$的单调性，所以最小化上式等价于最小化生成器损失函数。
当我们把以上求得的最佳判别器代入时，有如下表达式：

$$\mathbb{E}_{x\sim P_r}\log\frac{P_r(x)}{\frac{1}{2}[P_r(x)+P_g(x)]} + \mathbb{E}_{x\sim P_g}\log\frac{P_g(x)}{\frac{1}{2}[P_r(x)+P_g(x)]} - 2\log 2$$

此处变换是为了引入KL散度和JS散度的概念：

$$KL(P_1||P_2) = \mathbb{E}_{x\sim P_1} \log \frac{P_1}{P_2}$$

$$JS(P_1||P_2) = \frac{1}{2}KL(P_1||\frac{P_1+P_2}{2}) + \frac{1}{2}KL(P_2||\frac{P_1+P_2}{2})$$

这样，生成器需要最小化的目标就变为了：

$$2JS(P_r||P_g) - 2\log 2$$

也即，**判别器越接近最优，最小化生成器的loss也就会越近似于最小化P_r和P_g之间的JS散度。**
在计算JS散度前，我们需要考虑一个重要前提T：**当P_r与P_g的支撑集（support）是高维空间中的低维流形（manifold）时，P_r与P_g重叠部分测度（measure）为0的概率为1。**

> *对这一前提的理解：n维流形指高维空间中曲线、曲面概念的拓广，如三维空间中的一个曲面是一个二维流形，因为它只有两个方向的自由度，同理曲线就是一维流形。在三维乃至高维空间中，曲面或者曲线重叠的可能性很低。*

而由于生成器一般从一个低维的随机分布（如100维）中采样出编码向量，再通过一个神经网络生成一个高维样本（如64*64,4096），所以尽管其也有高维度，但是其也仅由低维的随机分布唯一确定，其支撑集最多只有100维，“撑不满”高维空间。所以，满足前提T。
因此，P_r与P_g几乎不可能重叠，$P_r(x)!=0$并且$P_g(x)!=0$的情况几乎不存在，而当$P_r(x)=0$并且$P_g(x)!=0$或者相反时，$JS(P_1||P_2)=log2$，为常数，梯度自然为0，产生梯度消失。

1.2 模型崩塌问题

为了降低问题一的影响，GAN原作者提出了第二种改进的生成器的损失函数（“trick”）：

$$\mathbb{E}_{x\sim P_g}[-\log D(x)]$$

而经过等价变换后，最小化上式即最小化：

$$KL(P_g||P_r) - 2JS(P_r||P_g)$$

这个等价最小化目标存在两个严重的问题。第一是它同时要最小化生成分布与真实分布的KL散度，却又要最大化两者的JS散度，一个要拉近，一个却要推远。在数值上表现为梯度不稳定；第二则是KL散度是不对称的（分母和分子）：

- 当$P_g(x) \to 0$而$P_r(x) \to 1$时，$P_g(x)\log\frac{P_g(x)}{P_r(x)} \to 0$，对$KL(P_g||P_r)$贡献趋近0
- 当$P_g(x) \to 1$而$P_r(x) \to 0$时，$P_g(x)\log\frac{P_g(x)}{P_r(x)} \to +\infty$，对$KL(P_g||P_r)$贡献趋近正无穷

叠加影响下，**生成器宁可多生成一些重复但是很“安全”的样本，也不愿意去生成多样性的样本**，产生了模型崩塌问题。

2 中间阶段

为了解决KL和JS的突变问题，人们尝试了其他loss函数来优化网络。

- **f-GAN**（2016，Nowozin等人）：通过更一般的f-散度（如KL、Pearson χ^2散度）替代JS散度，提供更灵活的损失函数选择。
- **LSGAN**（Least Squares GAN，2016，Mao等人）：用最小二乘损失替代原始GAN的二元交叉熵，缓解梯度消失问题，生成更稳定的样本。
- **噪声添加**：对生成样本和真实样本加噪声，直观上说，使得原本的两个低维流形“弥散”到整个高维空间，强行让它们产生不可忽略的重叠。而一旦存在重叠，JS散度就能真正发挥作用，此时如果两个分布越靠近，它们“弥散”出来的部分重叠得越多，JS散度也会越小而不会一直是一个常数。同时在训练中逐步对噪声退火，降低其影响。
  这些方法从损失函数设计、网络结构、训练技巧等角度改进GAN，但核心问题（如梯度不稳定、模式崩溃）仍未彻底解决。
  此外，还有CGAN的问世。CGAN对G部分：输入标签信息进行编码（嵌入层表示学习），再用合理的方法把噪声和编码信息合成（连接、乘起来等）对D部分：同时输入标签，将标签编码，和输入的假信息合成。

3 WGAN（2017）

WGAN的提出即是为了优化上述问题。

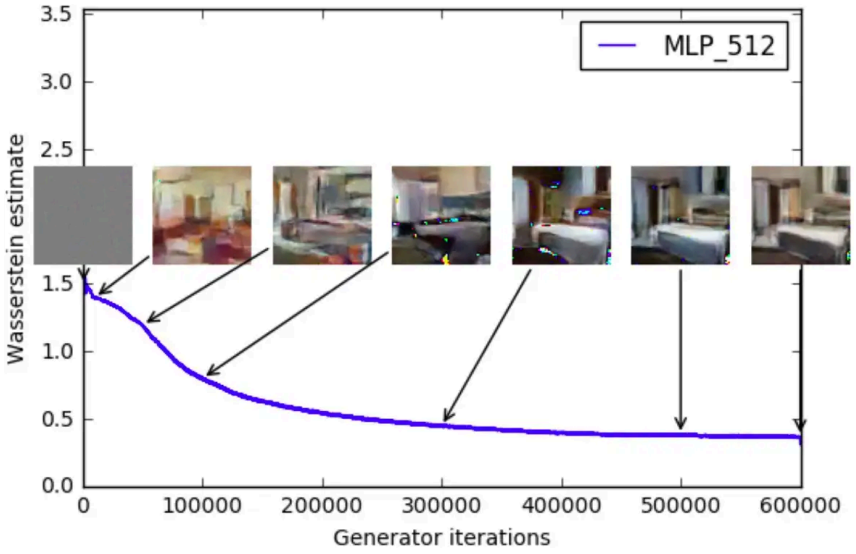
Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.
Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

WGAN的优化成果如下：

- 彻底解决GAN训练不稳定的问题，不再需要小心平衡生成器和判别器的训练程度
- 基本解决了collapse mode的问题，确保了生成样本的多样性
- 训练过程中终于有一个像交叉熵、准确率这样的数值来指示训练的进程，这个数值越小代表GAN训练得越好，代表生成器产生的图像质量越高：



WGAN的主要改动为使用了Wasserstein距离（Earth-Mover距离）来替代原本的目标函数。在WGAN中，它被变换为如下形式：

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

直观理解，Wasserstein距离为在γ这个“路径规划”下把P_r这堆“沙土”挪到P_g“位置”所需的“消耗”，而$W(P_r, P_g)$就是“最优路径规划”下的“最小消耗”。**它相比KL散度、JS散度的优越性在于，即便两个分布没有重叠，Wasserstein距离仍然能够反映它们的远近。**
进行以上变换的前提是满足一个叫做**Lipschitz连续**的条件：要求存在一个常数$K \geq 0$使得定义域内的任意两个元素x_1和x_2都满足：

$$|f(x_1) - f(x_2)| \leq K|x_1 - x_2|$$

类似于一个函数f的导数不超过某个Lipschitz常数K 。作者通过限制神经网络的所有参数 w 不超过某个范围$[-c, c]$满足这一约束。原式子的求解转化为：

$$K \cdot W(P_r, P_g) \approx \max_{w:|f_w|_L \leq K} \mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{x \sim P_g}[f_w(x)]$$

于是可以构造一个含参数w的判别器f_w，在参数不超过某个范围的条件下，最大化下面的式子，即：

$$\max L = \mathbb{E}_{x\sim P_r}[f_w(x)] - \mathbb{E}_{x\sim P_g}[f_w(x)]$$

此时L就会近似真实分布与生成分布之间的Wasserstein距离。最终的损失函数正好是原始GAN损失函数去除log的形式。
故WGAN的改动总结如下：

- 判别器最后一层去掉sigmoid（分类任务-->回归任务）
- **生成器和判别器loss修改**
- **每次更新判别器时把参数截断到一定范围内（满足约束，但导致了问题）**
- 建议不用基于动量的优化算法（Adam等）

4 WGAN-GP（2017）

原WGAN存在以下问题：

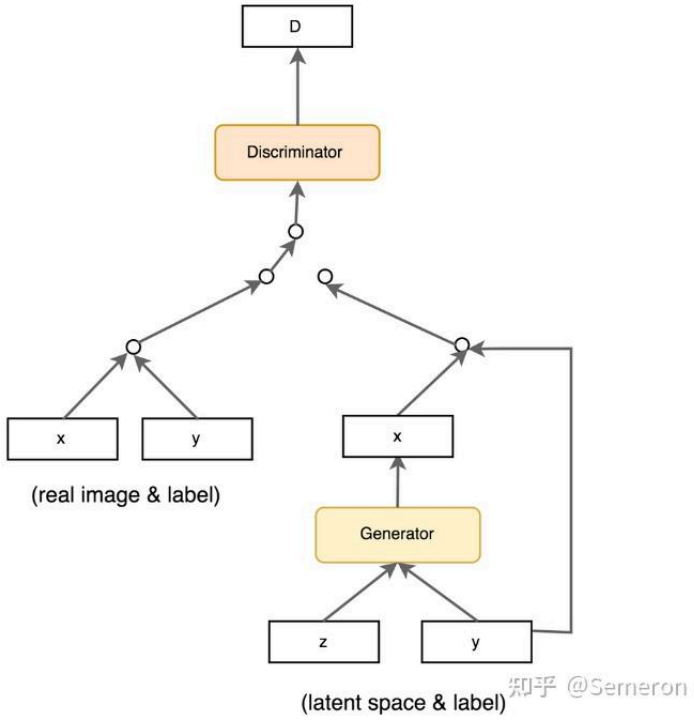
- 参数裁剪会导致参数基本都在限制的边界值，极大浪费了模型的参数。
- 还是很容易梯度消失或者梯度爆炸，需要仔细的调参
  WGAN-GP的改进为用更好的梯度惩罚（Gradient Penalty）替代权重裁剪；判别器的训练目标为最大化下面的L并使得GP惩罚项接近于1：

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}\left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]}_{\text{Their gradient penalty}}.$$

$$\hat{\boldsymbol{x}} = \epsilon\tilde{\boldsymbol{x}} + (1-\epsilon)\boldsymbol{x}, \epsilon \sim U[0,1]$$

5 Conditional GAN

CGAN将真实标签作为特征的一部分，输入生成器和判别器用于训练。主要用于图像转换（翻译），即输入图像和一些信息，生成新图像。



需要注意的是，我们可以认为，**图像生成和判别是在知晓真实标签的这一条件下完成的**，所以目标函数可以如下表示：

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[logD(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[log(1-D(G(z)))\right]$$

$$\downarrow$$

$$\min_G \max_D V'(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[logD(x|y)\right] + \mathbb{E}_{z\sim p_z(z)}\left[log(1-D(G(z|y)))\right]$$

5.1 生成器G

主要考虑latent space和label的结合，结合的方法会因为输入信息维度的不同而相应地变化，如特征标签都为图像时，在通道的维度，对特征与标签进行合并；当涉及信息升维时，要对标签信息进行embedding或者全连接，再用合理的方法把噪声和编码信息合成（连接、乘起来等）

5.2 判别器D

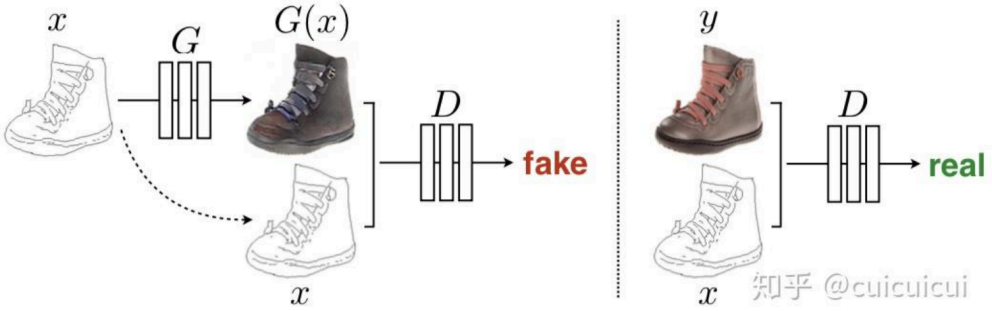
同样地，同时输入真实信息、标签和生成器生成信息，将标签编码、与真实信息合成、再和输入的生成信息合成。

6 pix2pix GAN

pix2pix是在CGAN的基础上将标签输入改为输入图像作为标签，以进行图像风格的变化和图像翻译。但是它的网络结构和CGAN有很大不同。

6.1 生成器G

生成器G使用了U-Net进行图像生成（在Auto-Encoder的基础上添加skip-connection，因为“输入和输出图像的外表面(surface appearance)应该不同而潜在的结构(underlying structure)应该相似”，即使用U-Net）

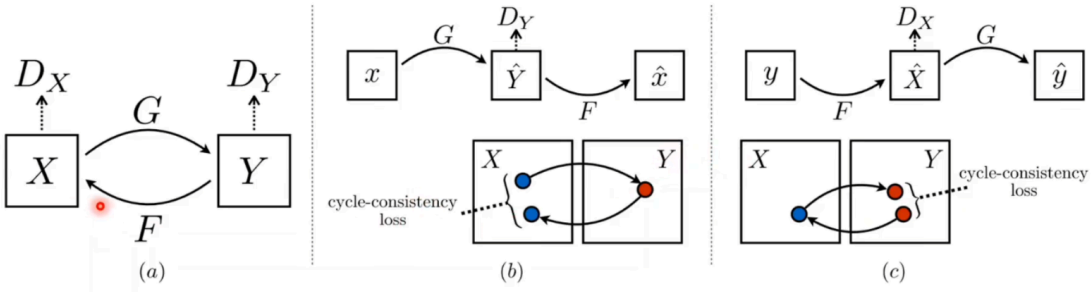


6.2 判别器D

新提出的条件判别器条件判别器PatchGAN，对输入图像的局部小块（Patch）进行判别，而不是对整张图像进行判别。这种设计的好处是可以关注到细节纹理和局部，而L1 loss和L2 loss并不能很好的恢复图像的高频部分。

7 CYCLE GAN

传统CGAN和pix2pix需要图像和标签的配对，采集数据困难，于是CYCLEGAN尝试在未配对数据集上做。
具体的方法是使用了两个生成器、两个判别器（可复用）来进行图像的转换，这两个GAN可以从无标记数据集中学习风格特征。
网络结构如下：



光看图比较难以理解，训练中的顺序如下：

也就是有两个不同方向的G（也可以是一个同时做双向的任务）来负责转化A到B和B到A，但除了使用D对转化结果进行原损失函数评估外，转化结果还会再被另外一个方向的G转化一次（类似重建）得到二次转化结果，再将二次转化结果与原输入做比较，进行一个新损失函数$cycle-consistency\ loss$的优化。
此外，还引入了$id-loss$，用来防止转化过激。

> 本题主要参考资料：
> 论文：Towards Principled Methods for Training Generative Adversarial Networks, Wasserstein GAN
> 【pix2pix】 https://www.bilibili.com/video/BV1EX4y1M7wz/?share_source=copy_web&vd_source=021bb0d047e3a0689c157da3d7b12c77
> 令人拍案叫绝的Wasserstein GAN - 郑华滨的文章 - 知乎 https://zhuanlan.zhihu.com/p/25071913