

Course 1

Week 1 Intro to GANs

BCE cost function:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log h(x^{(i)}, \theta) + (1 - y^{(i)}) \log (1 - h(x^{(i)}, \theta))] \quad (1.1)$$

初始化: generator 从随机噪声 ξ 开始生成数据

优化:

(1) 其中 generator 只考虑自己生成的假数据通过 discriminator 的结果, 因此 cost function 只包括后一项. discriminator 需要考虑所有.

(2) Generator 和 discriminator 分别各自优化. Generator 需要 $\max_{\theta} J_1(\theta)$, 而 discriminator 需

要 $\min_{\theta} J_2(\theta)$

BCE 可简化为

$$\min_d \max_g \left\{ -\left[\mathbb{E}(\log(d(x))) + \mathbb{E}(1 - \log(d(g(z)))) \right] \right\} \quad (1.2)$$

这里分别用 x, z 表示真数据和假数据.

Week 2 Basis of NN

无

Week 3 Wasserstein GAN

Mode collapse: 数据本身拥有许多特征, 但其中某一个(几个)很容易骗过 discriminator. 于是 generator 生成的数据只包含简单的 modes.

BCE cost function 的问题: 训练 discriminator 更容易, 由于 sigmoid 激活函数的特征(最终结果是 0~1 的概率), 几个 cycle 后出现 vanishing gradient, generator 学习不到有效信息.

● 解决思路:

换用 Earth Mover's Distance(EMD) 表示两个特征分布间的距离和量. Wasserstein loss 就表示 EMD 的大小, 因此是一个正实数. 不存在 vanishing 问题.

于是可以写出 W-Loss:

$$\mathbb{E}(c(x)) - \mathbb{E}(c(g(z))) \quad (1.3)$$

其中 c 表示用 critic 替代 discriminator. W-Loss 可以解决 vanishing gradient 问题, 同时也缓解 mode collapse.(因为 generator 可以一直获得 feedback)

● 对 critic 的要求: 连续可导, 且任一点 $\|\nabla c(x)\|_2 \leq 1$ (确保有边界).

为了实现最好的学习效果, 一般希望 $\|\nabla c(x)\|_2$ 接近 1. 于是增加一项 gradient penalty. 可以

写出 W-loss:

$$\max_c \min_g [\mathbb{E}(c(x)) - \mathbb{E}(c(g(z))) + \lambda \mathbb{E}(\|\nabla c(\hat{x})\|_2 - 1)^2] \quad (1.4)$$

其中 \hat{x} 表示真实样本和生成样本的线性插值, $\hat{x} = \epsilon x + (1 - \epsilon)z$. 这样可以尽可能多的代表 feature space.

Week 4 Controllable GAN

生成特定类的内容: 为 generator 提供 one-hot vector(与 ξ 合并), 同时 discriminator 也得到这一信息(例如图片可以增加 one-hot channel)

特征迁移: 发现 generator 中某一特征所代表的 \vec{d} (如头发颜色)之后, 可以用 $\vec{\xi} + \vec{d}$ 改变其他结果(如为所有人改变头发颜色)

Course 2

Week 1 Evaluation of GANs

评价标准: fidelity & diversity

图片比较: 使用 feature distance 而不是 pixel distance.

- 利用 ImageNet/inception-v3 等成熟的神经网络, 替换最后的几层(一般是 pooling layer 之后)训练得到需要的若干特征.
- Feature embedding: 将这些特征编码为 vector, $x \rightarrow \phi(x)$.

得到 feature embedding 之后, 可以计算 distance(Frechet Distance)

- 单变量: $(\mu_X - \mu_Y)^2 + (\sigma_X^2 + \sigma_Y^2 - 2\sigma_X\sigma_Y)$
- 多变量(Multivariate Normal Frechet Distance)

$$\text{FID} = \|\mu_X - \mu_Y\|_2 + \text{Tr}(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X \Sigma_Y}) \quad (2.1)$$

其中 Σ 为 covariance matrix.

缺陷: FID 假设了所有变量为正态分布; pre-trained model 中可能未训练出需要的特征.

☆ 一种 FID 的替代方法(但 FID 更常用): 利用 KL divergence 计算 Inception Score(IS, 基于 Inception-V3)

$$\text{IS} = \exp[\mathbb{E}_{x \sim p_g}(D_{KL}[p(y|x)||p(y)])] \quad (2.2)$$

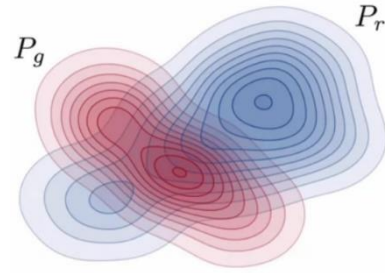
其中 $D_{KL}[p(y|x)||p(y)] = p(y|x) \log\left(\frac{p(y|x)}{p(y)}\right)$.

缺陷: 并没有与真实样本进行比较

✧ 使用 precision/recall

precision = $\frac{\text{overlap fake}}{\text{all fake}}$, 表征 fidelity.

recall = $\frac{\text{overlap real}}{\text{all real}}$, 表征 diversity.



Week 2 GAN disadvantages & bias

无

Week 3 StyleGAN & Advancements

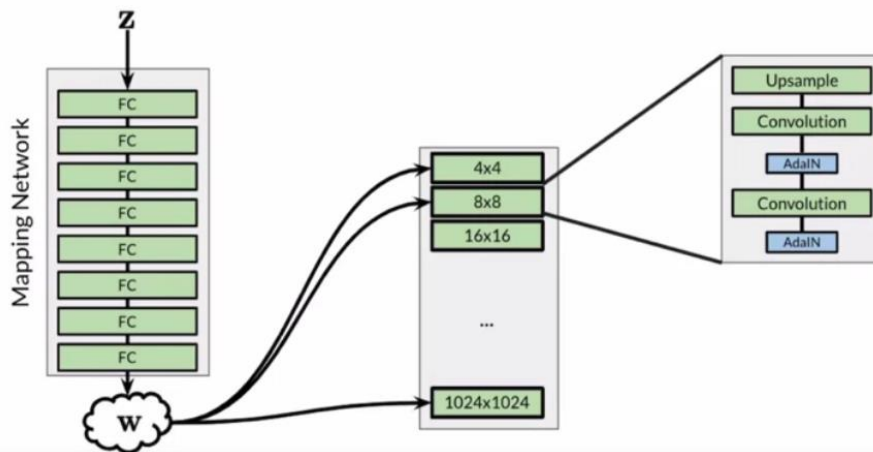
可提升的方向: diversity & stability. 可以将 standard deviation 加入 critic 中, 避免最终陷入 local minima(生成的图片都很接近). 另外也可以把几次 checkpoints 的结果取平均值.

StyleGAN: 主要包括 noise mapping network, adaptive instance normalization(AdaIN) 和 progressive growing.

(1) Progressive growing: 从低分辨率开始训练(e.g. 8×8), 逐渐增加 Conv layer 到高分辨率.

(2) Noise mapping network: 添加一个包含 8 个 FC layer 的神经网络(mapping network), 将 512×1 的 \tilde{z} 生成一个中间 noise vector \tilde{w} , 以实现不同特征之间的解耦

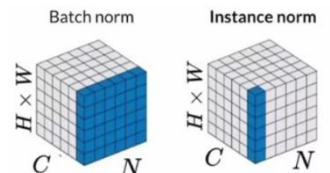
(3) AdaIN



1. 首先对卷积层结果进行 instance normalization. 与 batch normalization 的区别是前者只对每个 batch 中的每个 sample 进行归一化.
 2. 之后是添加 adaptive style(来自于 \tilde{w} , 其经过 FC 层生成的 scale 和 bias).
- 综合而言即

$$\text{AdaIN}(x_i, y) = y_{s,i} \cdot \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i} \quad (2.3)$$

\tilde{w} 会添加到每一个 Conv block(从 4×4 到 1024×1024).



- Style variation(mixing): 可以训练两种风格的 mapping network. 之后每次设置两个 \tilde{z} , 得到的 \tilde{w}_1, \tilde{w}_2 可以一个传入 Conv stacks 的前半部分做 AdaIN, 另一个传入后半部分.

- 另外也可以给 AdaIN 传入随机噪声($\lambda = 0.001 \sim 0.5$)以增加 variation.

Course 3

Week 1 Data Augmentation by GAN

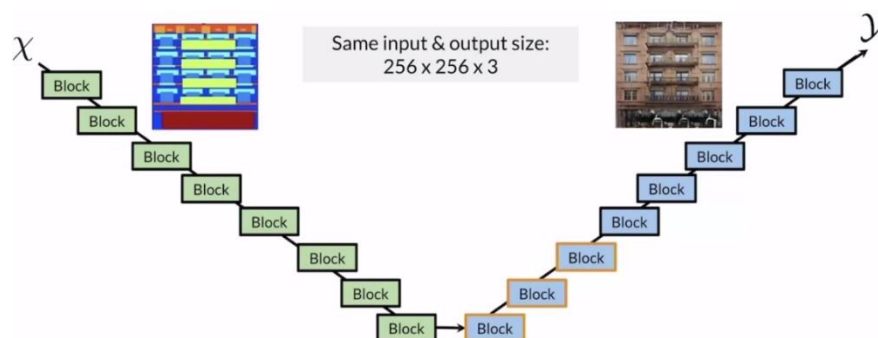
无

Week 2 Image-to-Image Translation (Pix2Pix)

Generator: 输入 mask, 输出生成的图片. (这里 noise vector 对结果的影响不大, 因此不作为输入, 进而用 dropout 增加 robustness)



- U-Net: encoder-decoder 结构. 其中: (1) decoder 前 3 个 block 可以在训练中增加 dropout 以提高泛化能力 (2) 相同尺寸的 encoder/decoder block 之间可以增加 skip connection 以减轻 gradient vanishing.
- Generator loss 可以增加 pixel loss ($\lambda|\text{generated} - \text{real}|$, L1 regularization)



Discriminator: 输入 mask + generated image 或 mask + real image. (PatchGAN). 输出 0~1 的 matrix, 表示真实的概率.(使用 BCE loss)

Week 3 CycleGAN

只改变 style, 不改变图片内容(content) – 无监督过程

2 ↑ generator & 2 ↑ discriminator

- Cycle consistency loss: 这里将 BCE loss 修改为

$$\text{least square loss} + \lambda(\text{style A difference} + \text{style B difference})$$