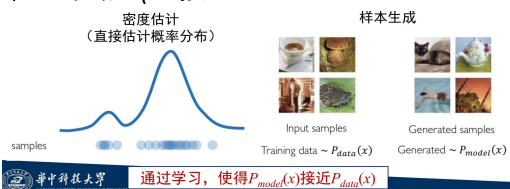


应用 自然图像生成 文字转图像
动画头像生成 图像编辑
图像风格迁移 图像修复

思想: 生成模型

从概率分布 $P_{data}(x)$ 中采样若干样本构成训练集, 学习 $P_{model}(x)$
模拟真实概率分布 $P_{data}(x)$



作用: 生成特殊场景数据, 为强化学习提供环境!

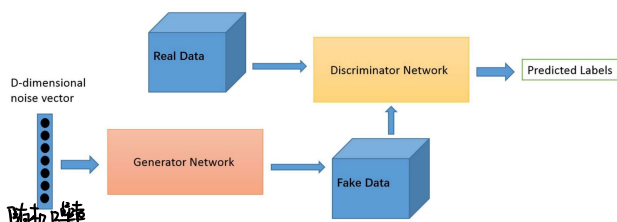
补充隐藏信息

生成模型类型 ① 自编码器/变分自编码器

② 生成对抗网络



GAN框架



随机噪声
GAN学习目标

• 假设 G 为生成器, D 为判别器, 真实数据 x 服从 $p_{data}(x)$ 分布, 噪声 z 服从 $p_z(z)$ 分布, 目标函数 $V(D, G)$ 的优化可表示为:

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

- $D(\cdot)$ 计算数据的真实性概率 (1 为真, 0 为假)。
- 生成器的目标是生成尽可能真实的“假数据”, 即 $D(G(z)) \rightarrow 1$;
- 判别器的目标是尽可能判断出数据的真假, 即 $D(x) \rightarrow 1, D(G(z)) \rightarrow 0$ 。

目标函数可看作真实数据和生成数据的交叉熵之和。

实际操作分两步:

• 训练判别器 D 时, 固定生成器 G 参数不变:

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

• 训练生成器 G 时, 固定判别器 D 参数不变:

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

• 循环优化, 直至收敛。

GAN特点

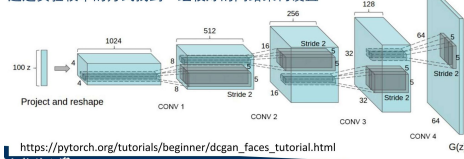
- 特点
 - ✓ 存在两个不同的网络, 且训练方式是对抗训练
 - ✓ 生成器的梯度更新信息来自判别器, 而不是数据样本
- 优点
 - ✓ 未局限网络结构的具体形式
 - ✓ 生成器的参数更新是使用来自判别器的反向传播, 降低 (某些场景下) 损失函数设计的困难
- 缺点
 - ✓ 可解释性差, 生成模型的分布没有显式的表达
 - ✓ 比较难训练, 生成器与判别器之间需要很好的同步
 - ✓ 存在训练不稳定、梯度消失、模式崩溃的问题

GAN变种

① DCGAN

Deep Convolutional GAN, 将深度卷积神经网络CNN与生成对抗网络GAN结合, 设计了一种通用的图像生成网络结构。

通过实验枚举的方式找到一组较好的网络架构设置



https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

不足: 生成器/判别器 Loss 无法指导训练; 样本缺乏多样性易出现模式崩溃

② WGAN

Wasserstein: 度量概率分布差异 \rightarrow 推土机距离 概率分布-验证 \rightarrow 易目标形状

利用Wasserstein距离构造Loss

WGAN 的具体改进如下:

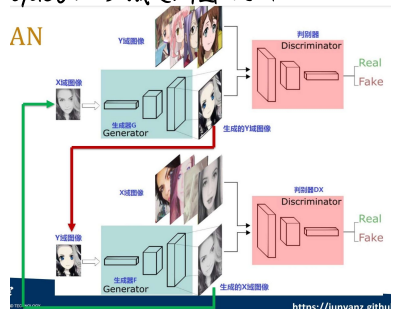
- 判别器最后一层去掉Sigmoid
- 生成器和判别器的损失函数不取Log, 采用Wasserstein 距离
- 每次更新判别器的参数后, 把参数截断在某个范围内
- 不使用基于动量的优化算法如Momentum和Adam, 推荐使用RMSProp等

改动简单, 却有奇效

优点

- 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
- 基本解决了模式崩塌问题, 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训练的越好, 代表生成器产生的图像质量越高
- 不需要精心设计的网络架构, 最简单的多层全连接网络就可做到

③ CycleGAN \Rightarrow 域迁移/画风迁移



<https://junyanz.github.io>