

## Recent\_Note

LiXiaoLong

July 18, 2024

# Contents

<b>1</b>	<b>GAN</b>	<b>2</b>
1.1	简介 . . . . .	2
1.1.1	设计思路 . . . . .	2
1.1.2	损失函数和训练策略 . . . . .	3
1.1.3	数学推导 . . . . .	3
1.1.4	Code . . . . .	3
<b>2</b>	<b>VAE model</b>	<b>5</b>

# Chapter 1

## GAN

### 1.1 简介

#### 1.1.1 设计思路

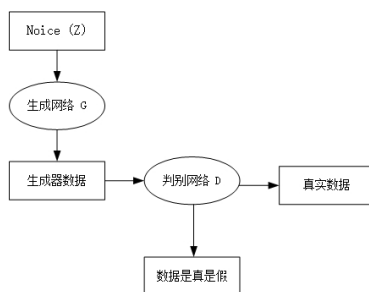


Figure 1.1: GAN 设计思路

GAN 包括两个模型，一个是生成模型  $G$  (Generator)，一个是判别模型  $D$  (Discriminator)。它们的功能分别是：

$G$  负责生成图片，接收一个随机的噪声  $z$ ，通过该噪声生成图片，记为  $G(z)$ 。

$D$  负责判别一张图片是否“真实”。其输入是  $x$ ，代表一张图片，输出  $D(x)$  表示  $x$  为真实图片的概率。输出为 1 代表真实图片的概率为 100%，而输出为 0 则代表图片不可能是真实的（真实实例来源于数据集，伪造实例来源于生成模型）。

有一个很好的比喻，就是枯叶蝶的演化过程类似于树叶，枯叶蝶不需要认识树叶，但能通过变异逃避捕食者（筛选器），这样的自然选择使枯叶蝶越来越像树叶。同理，生成器产生的图片概率分布也会越来越接近真实数据集的概率分布。

### 1.1.2 损失函数和训练策略

#### (1) 损失函数:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (1.1)$$

含义很直接, 对生成器, 尽可能让  $\mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$  更小, 也就是  $D(G(z))$  尽可能大, 前半段不涉及  $z$ , 当常数处理。

对判别器, 就是真图像判别的结果越接近 1 越好, 假图像越接近 0 越好。

训练策略:

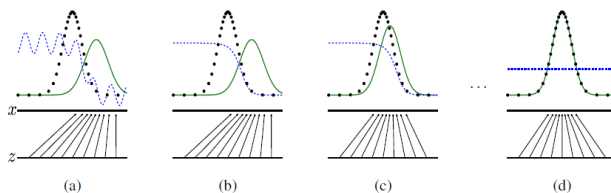


Figure 1.2: (2) 训练过程

Note: 判别器是小蓝, 真实图片的概率分布是小黑, 生成器整的映射是下半边的箭头, 由全数据区域到生成的图像空间, 绿色是生成图像空间的概率分布。

- 生成器和真实图像的概率分布偏差大, 性能是比较差, 判别器虽然大体在真是图像概率大的地方高, 但是不稳定。
- 判别器被迭代了几轮, 区分良好。
- 生成器被迭代更新, 概率分布趋近了真实图像分布一些。
- 大结局, 以假乱真, 判别器 out。

### 1.1.3 数学推导

推荐文章:<https://www.cnblogs.com/LXP-Never/p/9706790.html>

### 1.1.4 Code

[点击这里查看代码文件](#)

#### (1): 伪代码:

---

**for** number of training iterations **do**

**for**  $k$  steps **do**

- Sample minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from noise prior  $p_g(z)$ .
- Sample minibatch of  $m$  examples  $\{x^{(1)}, \dots, x^{(m)}\}$  from data generating distribution  $p_{\text{data}}(x)$ .
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[ \log D(x^{(i)}) + \log \left( 1 - D(G(z^{(i)})) \right) \right].$$

**end for**

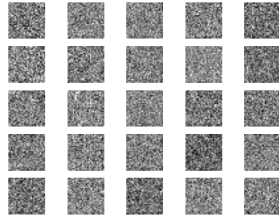
- Sample minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from noise prior  $p_g(z)$ .
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log \left( 1 - D(G(z^{(i)})) \right).$$

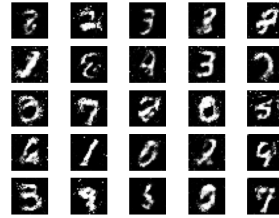
**end for**

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

---



(a) 训练 0 个 epoch 的生成结果



(b) 训练 200 个 epoch 生成的结果

Figure 1.3: 训练不同 epoch 数的生成结果对比

## Chapter 2

# VAE model