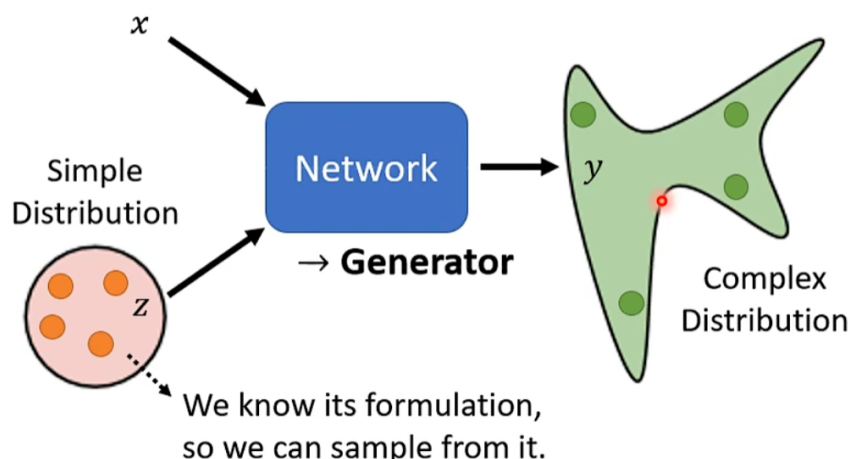


# GAN

## 引入

走迷宫左走和右走都是可能的，但是一定**不能同时**生成分别向右和向左的两个影像。为了解决这个问题，要让network的输出是一个概率问题。



因此，在输入的时候加入一个概率 $z$ ，从而使得输出不在固定（同时包含向左和向右转两个可能）→ 要让任务具有一定的**创造力**。

## 无条件生成

输入只有 $z$ ，输出就是 $y$ 。（这里 $z$ 是一个普通的低维向量，生成器输出一个高维向量 $y$ ）

辨别器：输入图像，输出一个标量（评分越高越真）

GAN的思路是：利用辨别器辨别不断辨别主要特征评分，从而调整生成器参数。

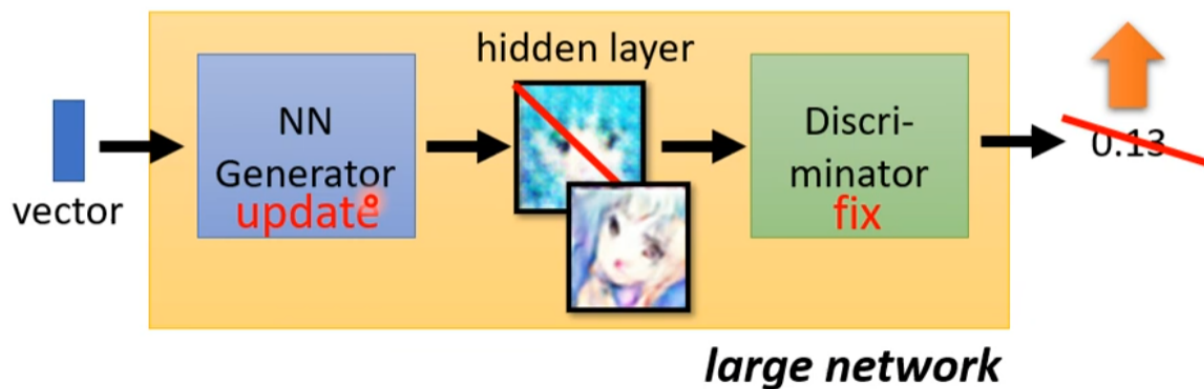
## 训练

初始化参数

训练迭代：

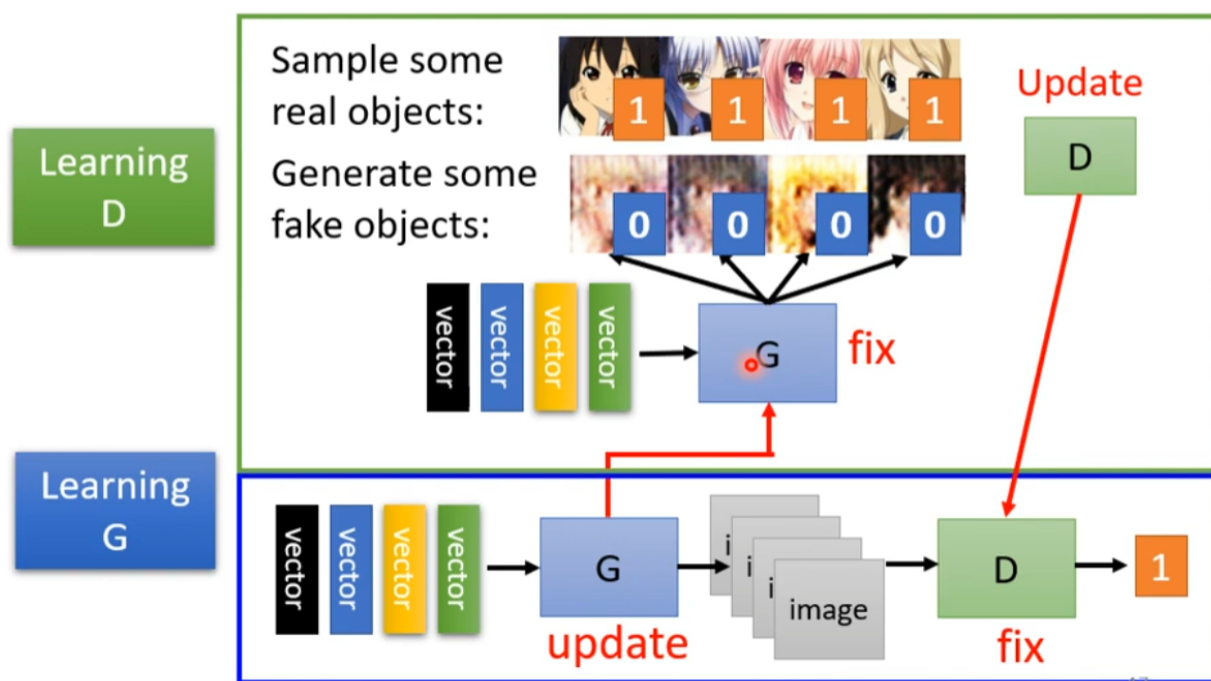
step1: 固定生成器，更新辨别器（准确分别真假）

step2: 固定判别器, 更新生成器 (欺骗, 让编码器的输出分数更高)



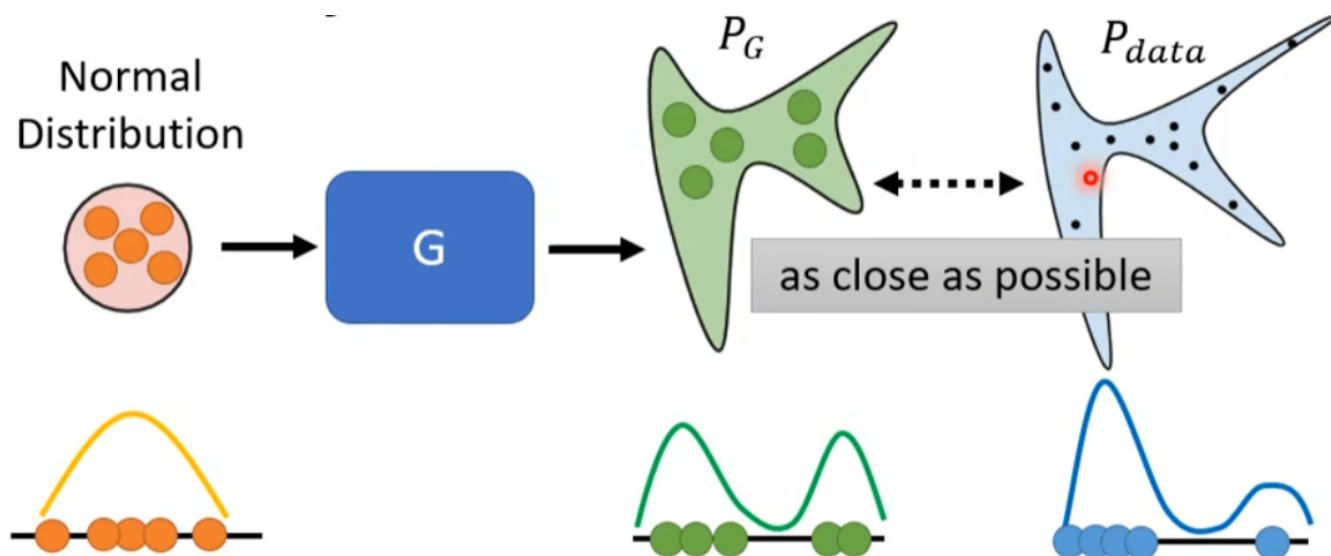
看作一个巨大的网络结构, 冻结编码器训练

step3: 不断重复step1和2



## 损失

核心是要真假图片分布 $P_G$ 和 $P_{data}$ 的距离差异 $Biv$ 更小 ( $P_G$ 和 $P_{data}$ 分布更像)  $G^* = \arg \min_G \text{Div}(P_G, P_{data})$ 。但核心是我们无法拿到准确的真假图像完整的分布信息。



$$G^* = \arg \min_G \underline{Div}(P_G, P_{data})$$

Divergence between distributions  $P_G$  and  $P_{data}$

尽管我们不知道  $P_{data}$  具体是怎么分布的，但是我们可以从database中采样选数据。从生成器中产生一定数量的采样作为  $P_G$ 。但其实  $V(G, D)$  等同于负的交叉熵函数，其实本质就是一个对真和假图像的 binary 分类任务

利用  $D^* = \arg \max_D V(D, G)$  训练编码器（能够辨别max真实图片和生成图像），这里  $D$  的目标函数是：

$$V(G, D) = E_{y \sim P_{data}} [\log D(y)] + E_{y \sim P_G} [\log(1 - D(y))]$$

我们希望给来自  $P_{data}$  的样本高分，给来自  $P_G$  的样本低分，

- $y$ 如果是从  $P_{data}$  采样出来的， $D(y)$ 越大越好
- $y$ 如果是从  $P_G$  采样出来的， $D(y)$ 越小越好  $\rightarrow 0$

所以就是要最大化这个目标方程。那么：此时就会出现：

- 如果判别能力强（真假特征提取差距大）（ $E_{y \sim P_G} [D(y)] \rightarrow 0$ ,  $E_{y \sim P_{data}} [D(y)] \rightarrow 1$ ）， $E_{y \sim P_G} [\log(1 - D(y))] \rightarrow 0$ ,  $E_{y \sim P_{data}} [\log D(y)] \rightarrow 0$ 。那么我们就可以计算出  $V(D, G)$  就会很小。
- 如果判别能力弱（真假特征提取差异小）， $V(D, G)$  就会很大

$V(D, G)$  符合JS散度，因此还可以利用其他JS散度函数。

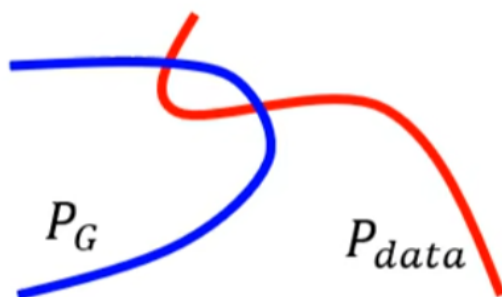
而训练判别器的时候，需要固定 $G$ 时，那么 $\max V(G, D)$ 就表示 $P_G$ 和 $P_{data}$ 之间的差异。因此，可以直接替代 $\text{Div}(P_G, P_{data})$ ，所以损失函数可以写作：

$$G^* = \arg \min_G \max_D V(D, G)$$

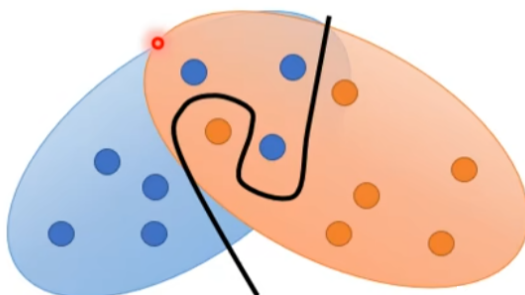
## 如何训练GAN

实际上， $P_{data}$ 和 $P_G$ 基本是不重叠的，因为：

(1)  $P_{data}$ 和 $P_G$ 都是自然的数据从高维空间到低维空间的manifold（很难在二维空间找到重叠的）

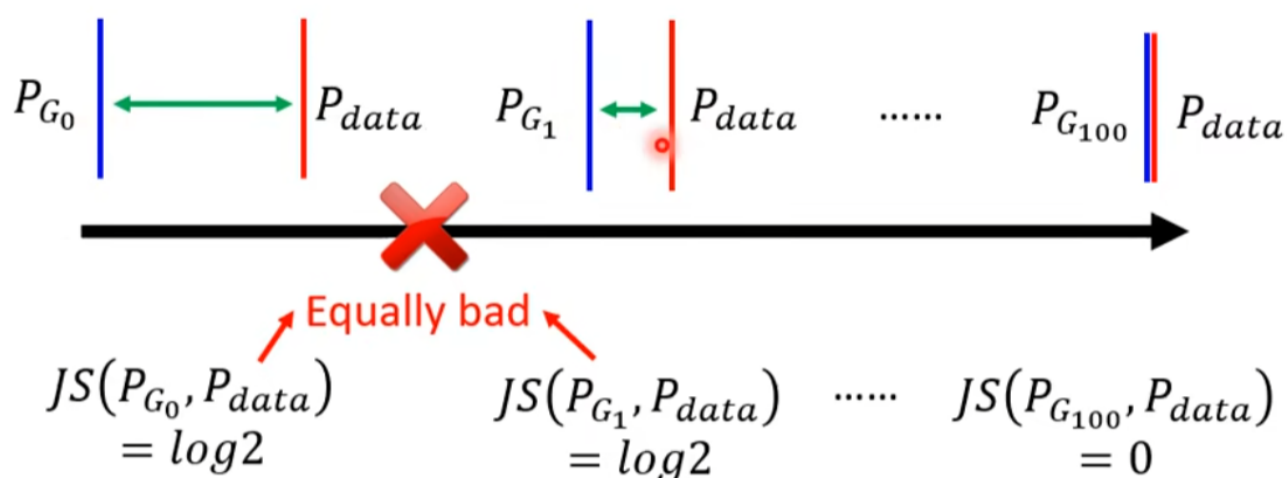


(2) 对于真假的判断只是来自采样 $P_{data}$ 和 $P_G$ ，可能理解不了重叠部分。



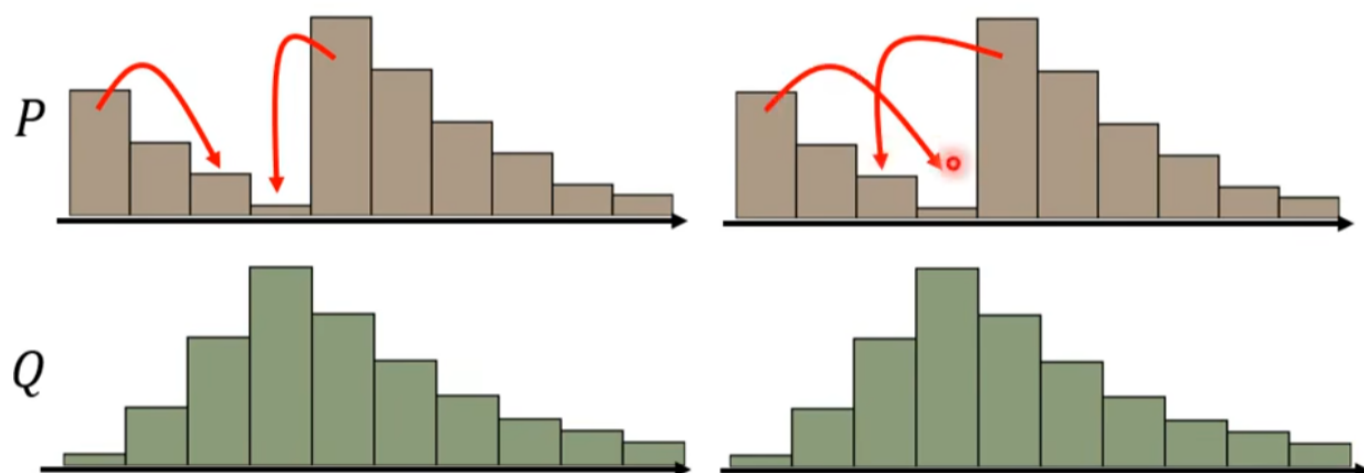
这就出现了一个问题，JS散度计算两个不重叠的分布结果总是 $\log 2$ 。但第二个虽然不重叠，但是生成的效果比第一个好（此时，准确率或损失毫无意义。无法往更好方向更新）

JS divergence is always  $\log 2$  if two distributions do not overlap.



## 利用Wasserstein distance改进

- Wasserstein distance: 移动土堆的距离



There are many possible "moving plans".

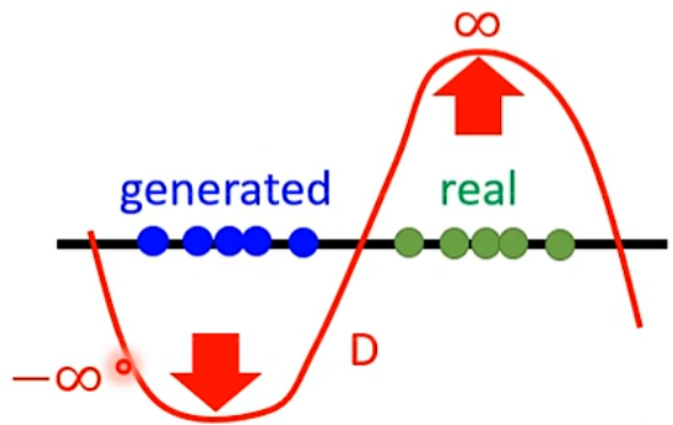
但在考虑比较复杂的分布时，各种移动方法也会导致多种距离结果。

Wasserstein distance是使用平均距离最小的“移动计划”。WGAN利用这个公式计算

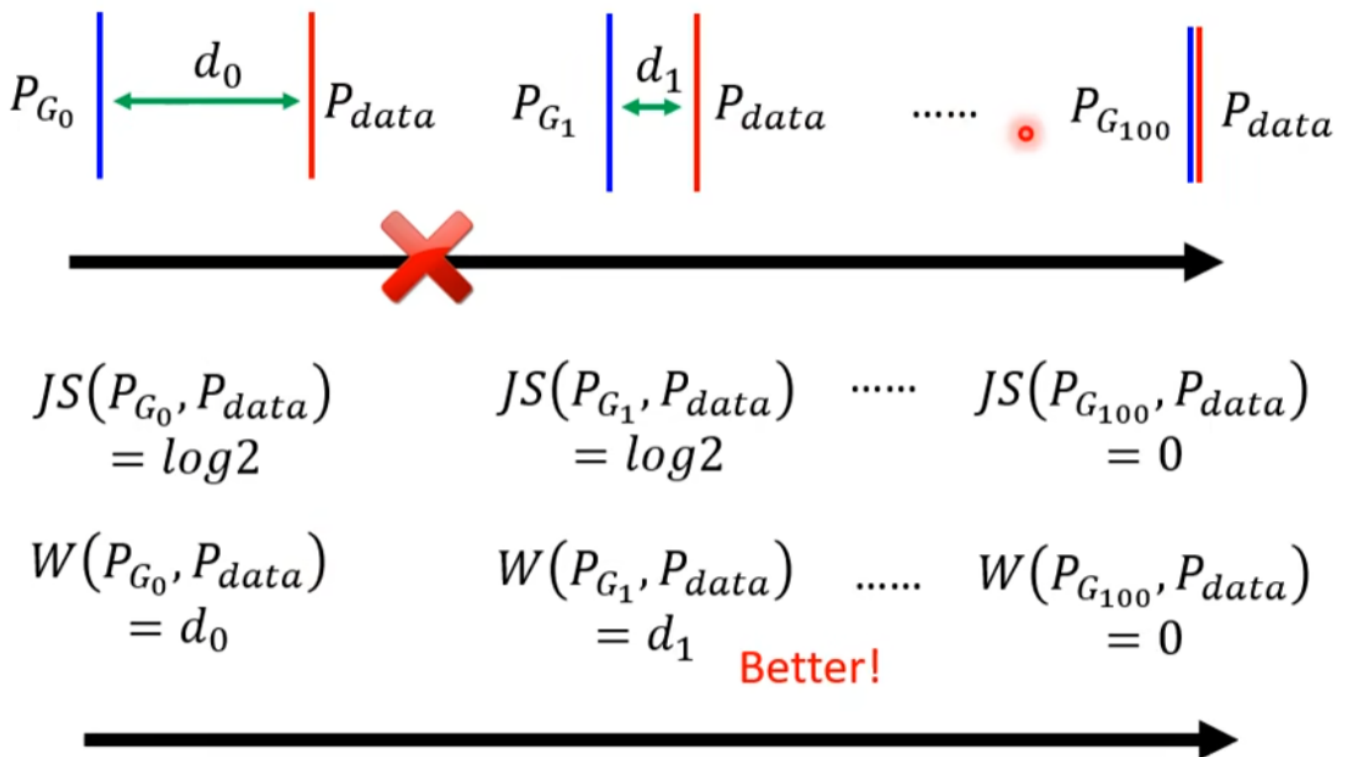
$$\max_{D \in 1\text{-Lipschitz}} \{ \mathbb{E}_{x \sim P_{data}} [D(y)] - \mathbb{E}_{x \sim P_G} [D(y)] \}$$

其中  $D$  必须是一个比较平滑的公式，否则可能会出现无限的情况（变化剧烈）

Without the constraint, the training of D will not converge.



如果利用Wasserstein distance替代JS散度，就可以接解决这个问题。



## 条件生成

条件+普通分布 $z$ 同时输入生成器，生成结果既需要满足条件又要满足真实性（要和pair算分数），训练的时候还需要考虑估计错配条件和图像。

## 其他运用改进

as close as possible  
Cycle consistency

