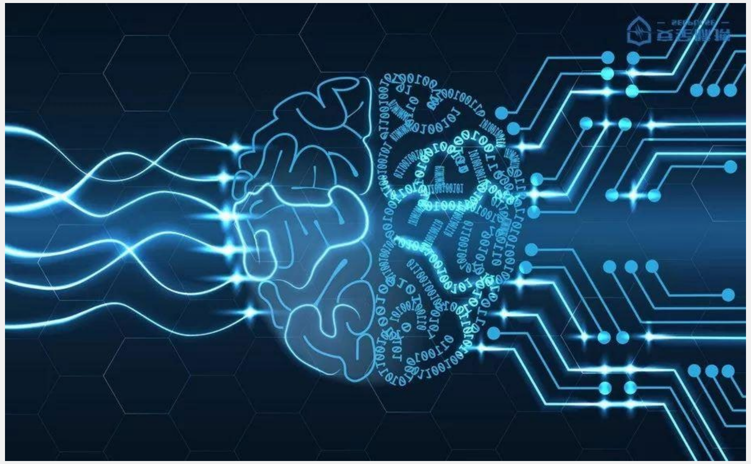


机器学习

Machine Learning

第八章：生成对抗网络（GAN）

目录

CONTENTS

为人师表

1. 生成对抗网络

- 生成对抗网络（Generative adversarial network），简称GAN，2014年Ian J. Goodfellow等人在《Generative Adversarial Nets》中提出的一个通过对抗过程估计生成模型的新框架。其应用包括：

✓ 产生图像数据

✓ 卡通人物创作

✓ 模拟人脸老化

1. 生成对抗网络

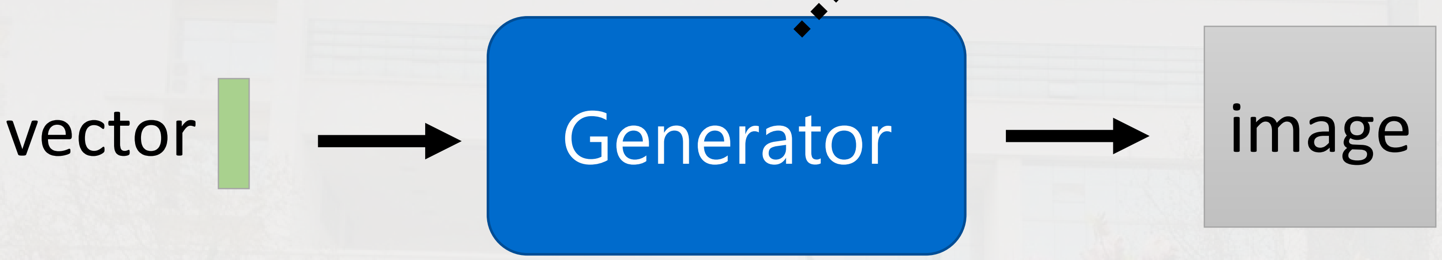
- 生成对抗网络（Generative adversarial network），简称GAN，2014年Ian J. Goodfellow等人在《Generative Adversarial Nets》中提出的一个通过对抗过程估计生成模型的新框架。其应用包括：
  - ✓ 根据文字生成图像

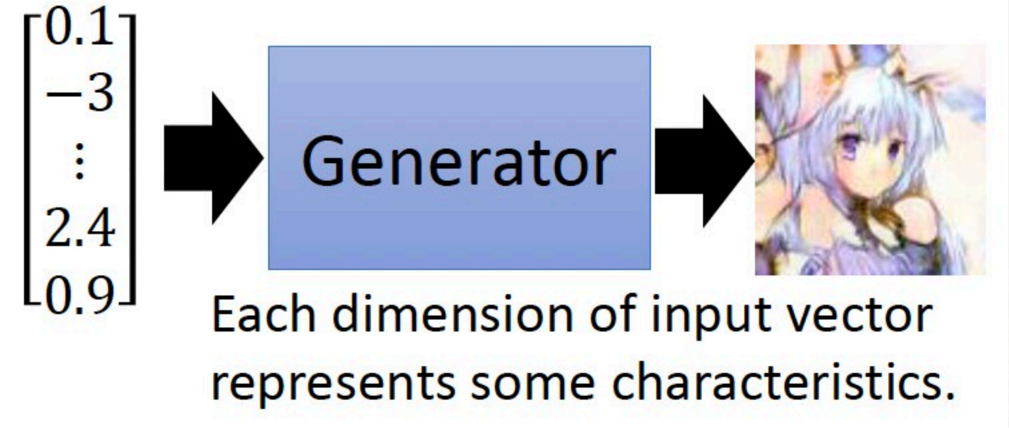
1. 生成对抗网络-结构

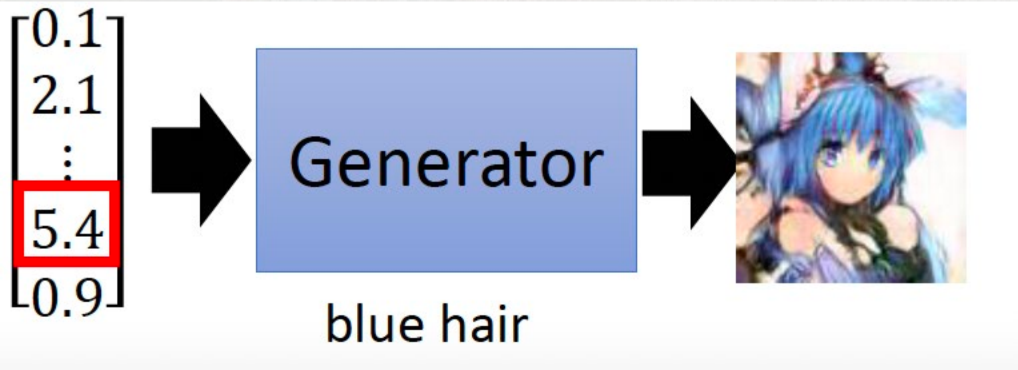
➢ GAN由两个主要网络构成：一个是Generator Network(生成器)，一个是Discriminator Network(判别器)

✓ **生成器**

It is a neural network (that is, a function).







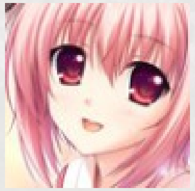
1. 生成对抗网络-结构

➢ GAN由两个主要网络构成：一个是Generator Network(生成器)，一个是Discriminator Network(判别器)

✓ **判别器**

It is a neural network (that is, a function).

image → Discriminator → **Scalar**: Larger means real, smaller value fake.


→ Discriminator → 1.0


→ Discriminator → 1.0


→ Discriminator → 0.1


→ Discriminator → 0.1

1. 生成对抗网络-结构

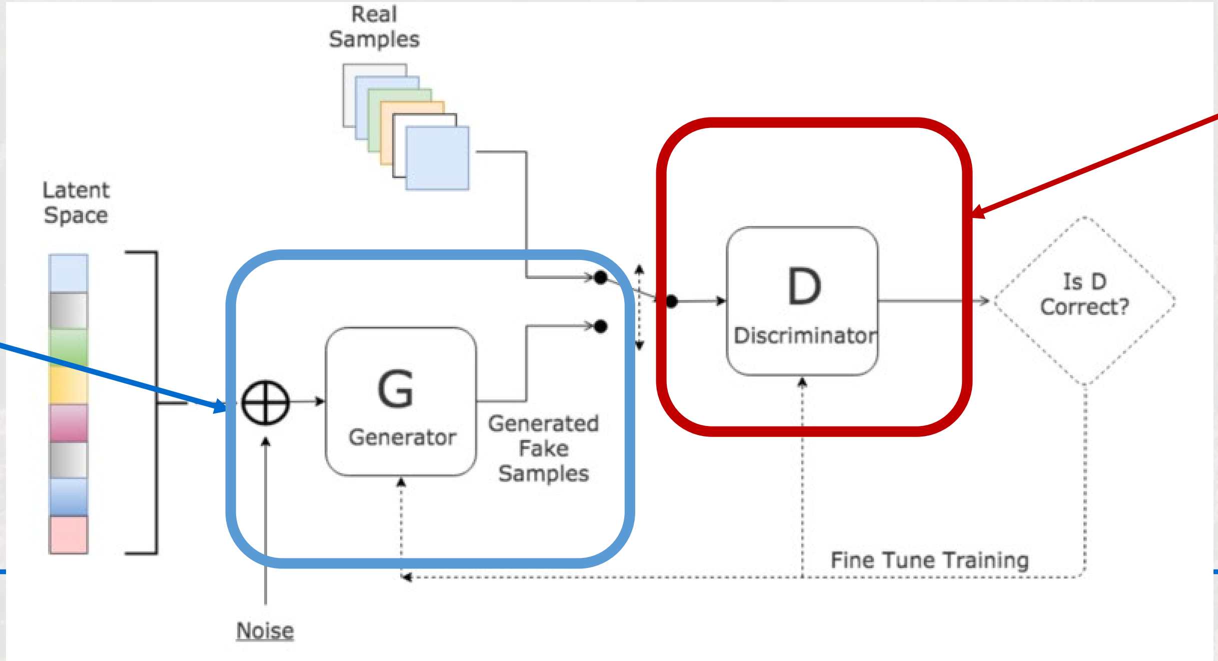
➢ GAN由两个主要网络构成：一个是Generator Network(生成器)，一个是Discriminator Network(判别器)

vector → Generator → image → Discriminator → **Scalar**

➢ GAN网络的核心逻辑是生成器和判别器相互对抗、相互博弈

生成器（Generator)

判别器（Discriminator)



1. 生成对抗网络-结构

➢ 生成对抗网络的基本思想

Generator

Brown

veins

Butterflies are not brown

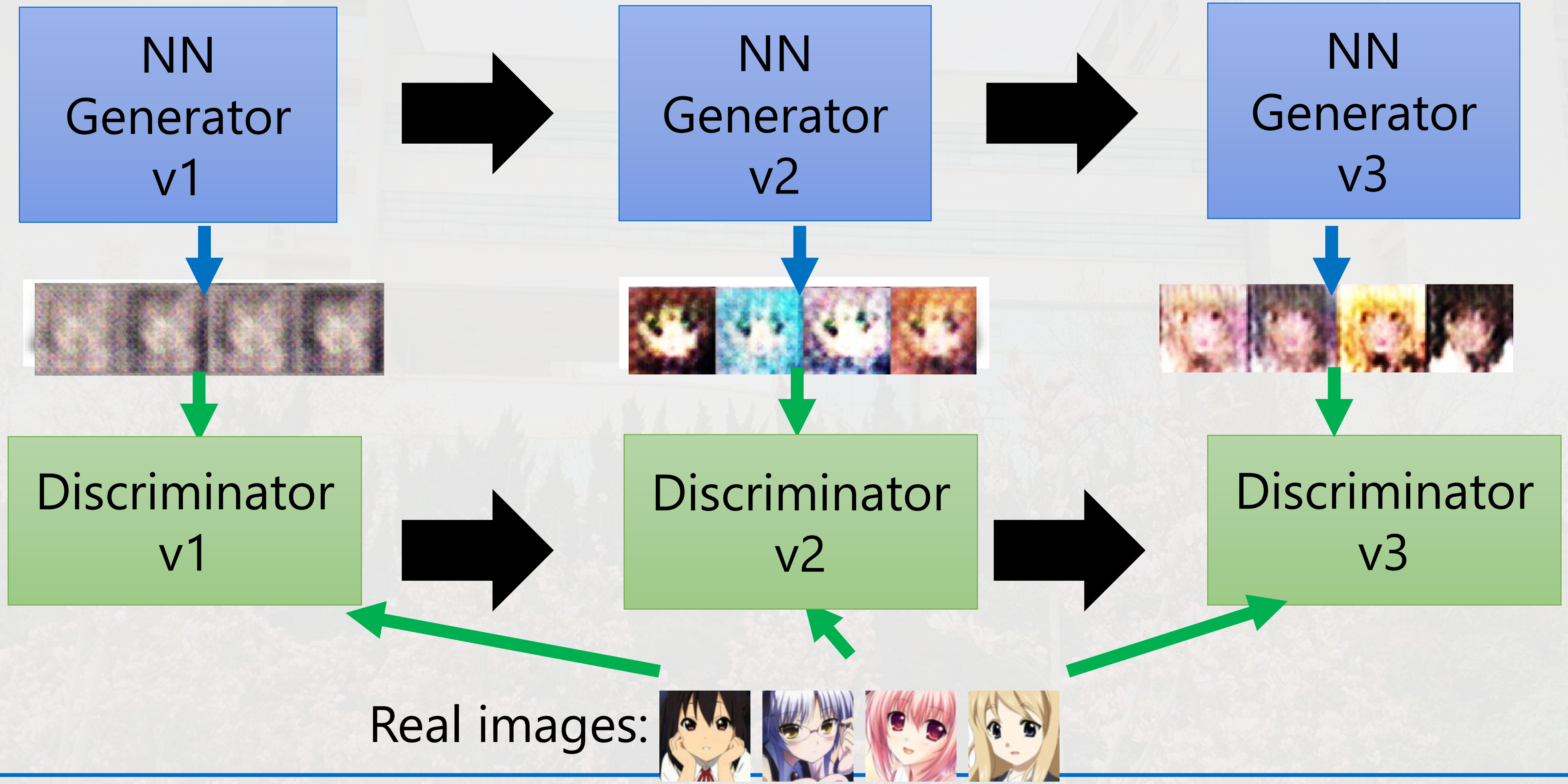
Butterflies do not have veins

....

Discriminator

1. 生成对抗网络-结构

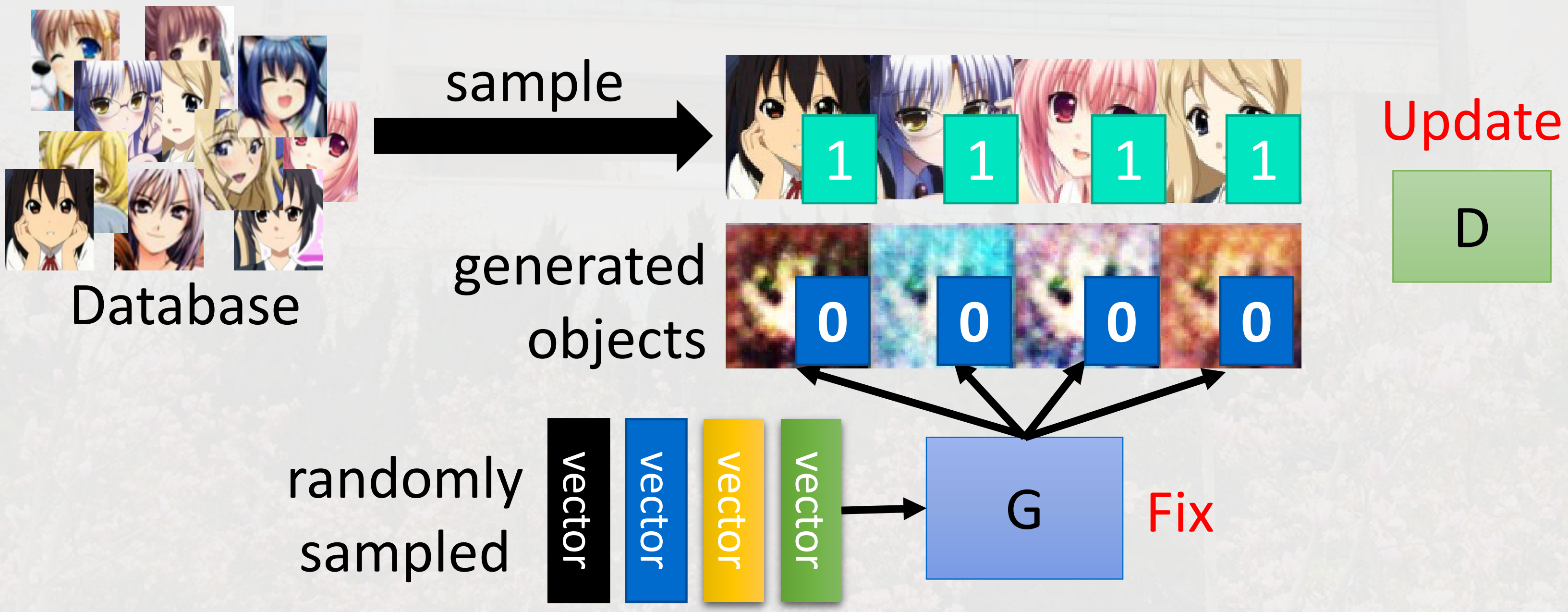
➢ 生成对抗网络的基本思想



1. 生成对抗网络-算法

- 初始化生成器和判别器
- 在每一轮迭代中：

Step 1: 训练判别器。固定生成器的参数，真实样本输入判别器后输出的结果标签为1，随机噪声 z 输入生成器得到 G(z)，再输入判别器后得到的输出结果标签为0，训练判别器到收敛

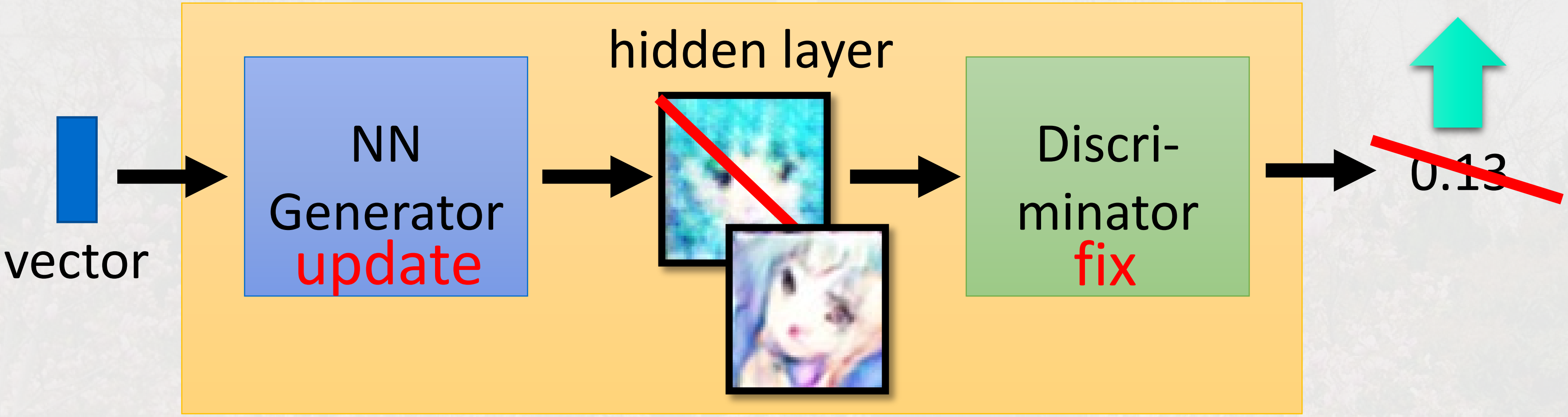


1. 生成对抗网络-算法

- 初始化生成器和判别器
- 在每一轮迭代中：

G D

<u>***Step 2***</u>: 训练生成器：固定判别器的参数，随机噪声输入生成器，得到假图，输入判别器，得到一个概率值，生成器更新参数来使判别器对生成图的输出尽可能地给高分



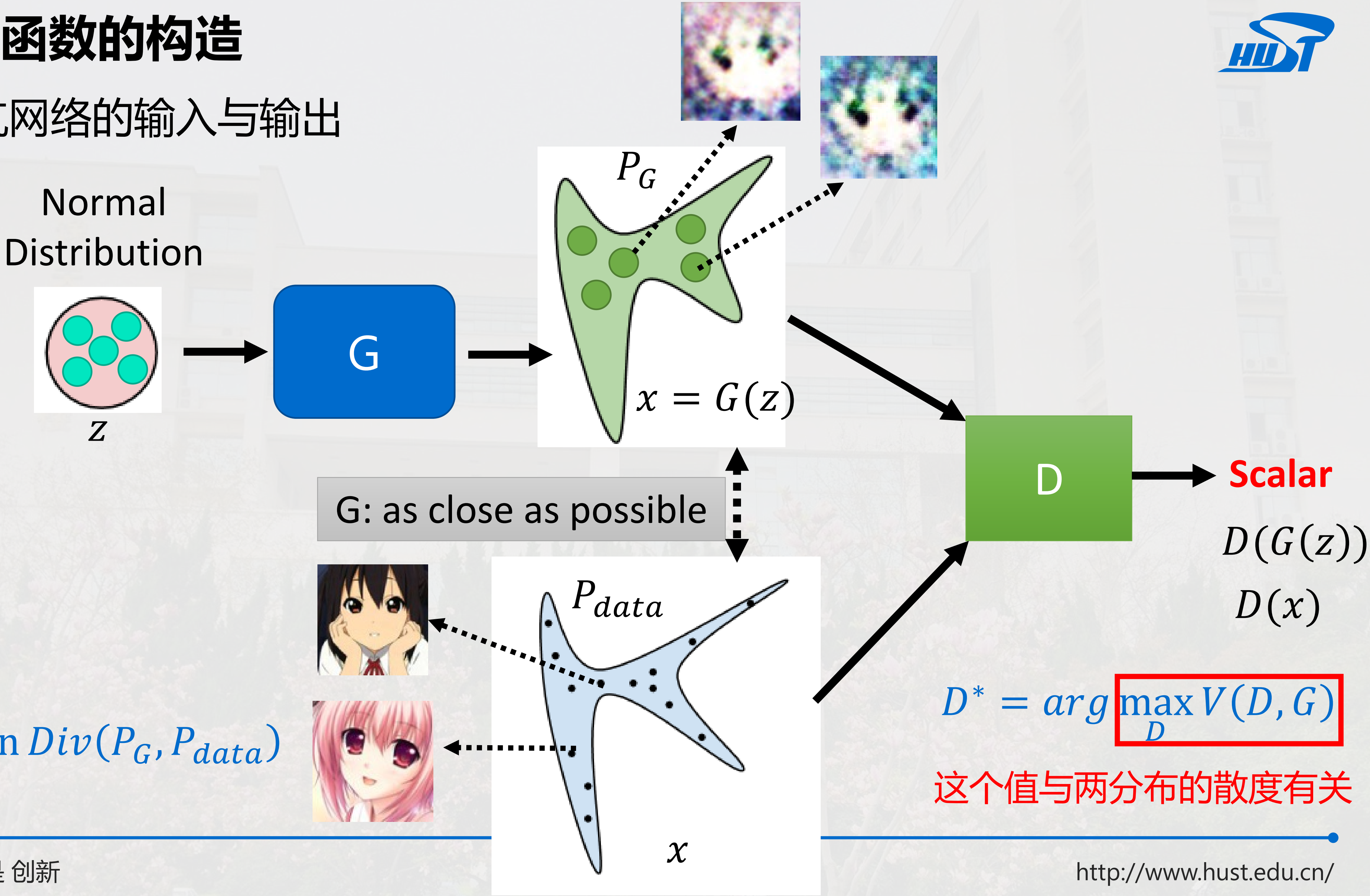
<u>large network</u>

1. 生成对抗网络-算法

- 初始化生成器和判别器
- 在每一轮迭代中：

2. 目标函数的构造

➢ 生成对抗网络的输入与输出



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

$$D^* = arg\,\boxed{\max_D V(D,G)}$$

这个值与两分布的散度有关

2. 目标函数的构造

1) 从数据库中抽取真实样本$\{x_i\}_{i=1}^{N}$: P_{data}

2) 生成器的生成数据服从分布P_g, 在生成对抗网络中，不直接对P_g进行建模，而是通过一个神经网络去逼近这个分布，输入z表示噪声数据，服从分布 $z \sim P_z(z)$

- 从判别器角度来讲：若输入来自于P_{data}，判别器的输出应该较大。反之，其输出应该较小（等价于$1-D(G(z))$ 较大），判别器的目标函数可写为：

$$\max_D E_{x\sim P_{data}(x)}[logD(x)] + E_{z\sim P_z(z)}[\log(1-D(G(z)))]$$

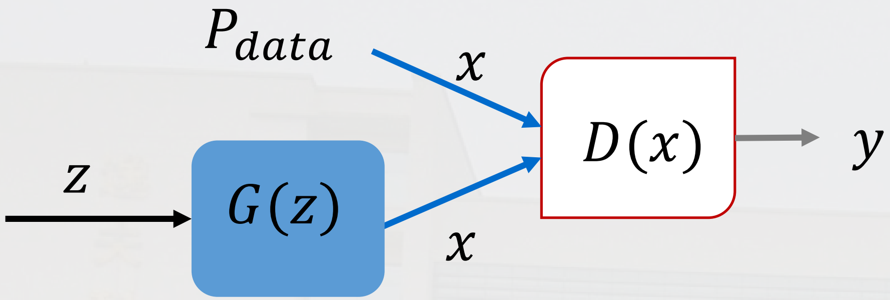
- 从生成器角度来讲：若输入来自于P_z，希望判别器的输出应该较大 （等价于$1-D(G(z))$ 较小），生成器的目标函数可写为：

$$\min_G E_{z\sim P_z(z)}[\log(1-D(G(z)))]$$

- 总目标函数可写为：

$$\min_G \max_D V(D,G) = \min_G \max_D E_{x\sim P_{data}(x)}[logD(x)] + E_{z\sim P_z(z)}[\log(1-D(G(z)))]$$

3 全局最优解



$$V(D,G)=E_{x\sim P_{data}}[logD(x)]+E_{z\sim P_z}[\log(1-D(G(z)))]$$

等价于

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

第一步希望判别器输出最大，那么此时固定G的参数，只训练D，将判别器的目标函数变换成积分形式：

$$\max_D V(D,G)=\int[P_{data}(x)logD(x)+P_g(x)\log(1-D(x))]dx$$

3 全局最优解

$$\max_D V(D,G) = \int [P_{data}(x)\ logD(x) + P_g(x)\log(1-D(x))]dx$$

要找到一个 D使得$V(D,G)$最大，对其求偏导：

$$\frac{\partial V(D,G)}{\partial D} = \int \frac{\partial}{\partial D}\big[P_{data}(x) logD(x) + P_g(x)\log(1-D(x))\big]dx = 0$$

$$\int [P_{data}(x)\frac{1}{D(x)} + P_g(x)\frac{-1}{1-D(x)}]dx = 0$$

将上式进行变化可得：

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

3 全局最优解

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

将推导出的公式代入目标函数中：

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

$$= \min_G E_{x\sim P_{data}}[logD^*(x)] + E_{x\sim P_g}[\log(1 - D^*(x))]$$

$$= \min_G E_{x\sim P_{data}}[log\frac{P_{data}(x)}{P_{data}(x)+P_g(x)}] + E_{x\sim P_g}[\log\frac{P_g(x)}{P_{data}(x)+P_g(x)}]$$

作一些简单的变换：

$$= \min_G E_{x\sim P_{data}}[log\frac{\frac{1}{2}P_{data}(x)}{\frac{P_{data}(x)+P_g(x)}{2}}] + E_{x\sim P_g}[\log\frac{\frac{1}{2}P_g(x)}{\frac{P_{data}(x)+P_g(x)}{2}}]$$

$$= \min_G -2log2 + E_{x\sim P_{data}}[log\frac{P_{data}(x)}{\frac{P_{data}(x)+P_g(x)}{2}}] + E_{x\sim P_g}[\log\frac{P_g(x)}{\frac{P_{data}(x)+P_g(x)}{2}}]$$

3 全局最优解

KL散度（相对熵）： $KL(P||Q)=\int P(x)\log\frac{P(x)}{Q(x)}dx$

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

$$=\min_G -2log2+E_{x\sim P_{data}}[log\frac{P_{data}(x)}{\frac{P_{data}(x)+P_g(x)}{2}}]+E_{x\sim P_g}[\log\frac{P_g(x)}{\frac{P_{data}(x)+P_g(x)}{2}}]$$

$$=\min_G -log4+\mathrm{KL}(P_{data}\parallel\frac{P_{data}(x)+P_g(x)}{2})+\mathrm{KL}(P_g\parallel\frac{P_{data}(x)+P_g(x)}{2})]$$

$$\geq -log4$$

当 $P_{data}=\frac{P_{data}(x)+P_g(x)}{2}=P_g$时成立：

最优 $P_g^*=P_{data}$，此时$D^*=\frac{P_{data}(x)}{P_{data}(x)+P_g(x)}=\frac{1}{2}$

3 全局最优解

KL散度（相对熵）： $KL(P||Q) = \int P(x)\log\frac{P(x)}{Q(x)}dx$

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

$$= \min_G -log4 + \mathrm{KL}(P_{data} \parallel \frac{P_{data}(x)+P_g(x)}{2}) + \mathrm{KL}(P_g \parallel \frac{P_{data}(x)+P_g(x)}{2})]$$

JS散度： $JS(P||Q) = \frac{1}{2}KL(P_1\left|\right|\frac{P_1+P_2}{2}) + \frac{1}{2}KL(P_2\left|\right|\frac{P_1+P_2}{2})$

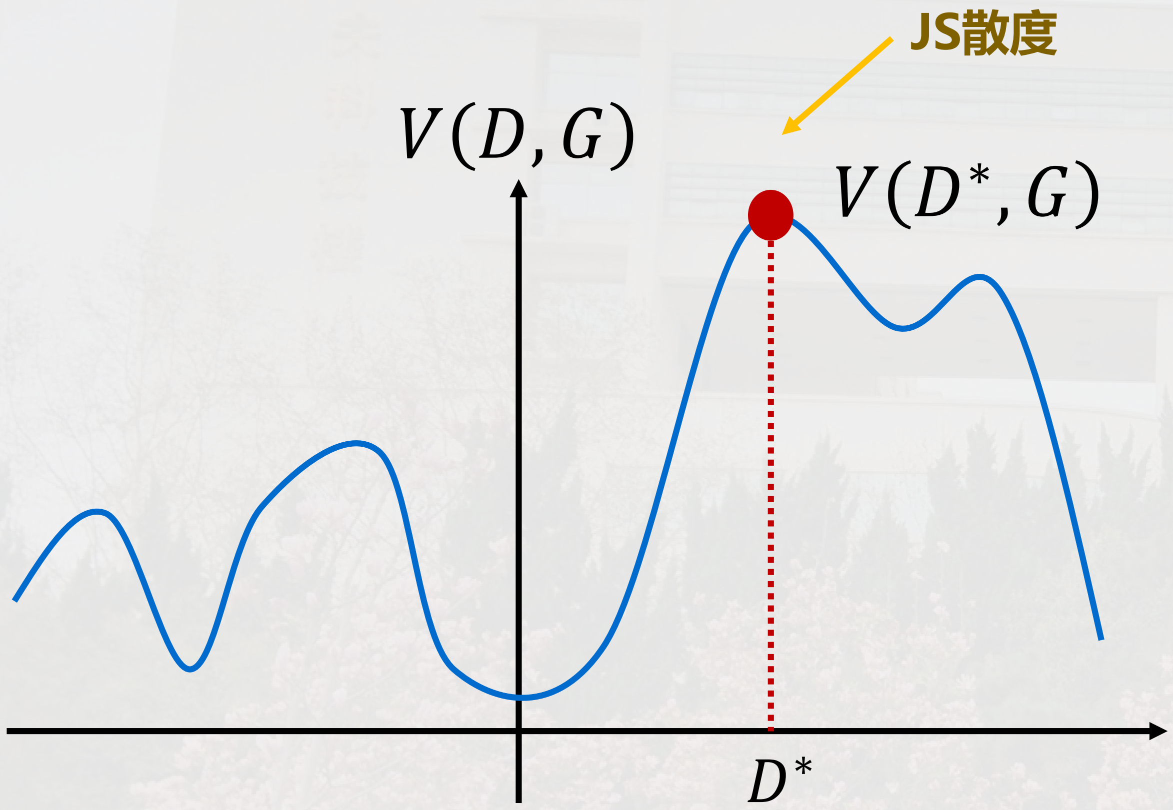
$$= \min_G -log4 + 2\mathrm{JS}(P_{data} \parallel \frac{P_{data}(x)+P_g(x)}{2})$$

- 生成器最小化GAN的目标函数就是最小化真实分布与生成分布之间的JS散度，即最小化两个分布的相对熵。

3 全局最优解

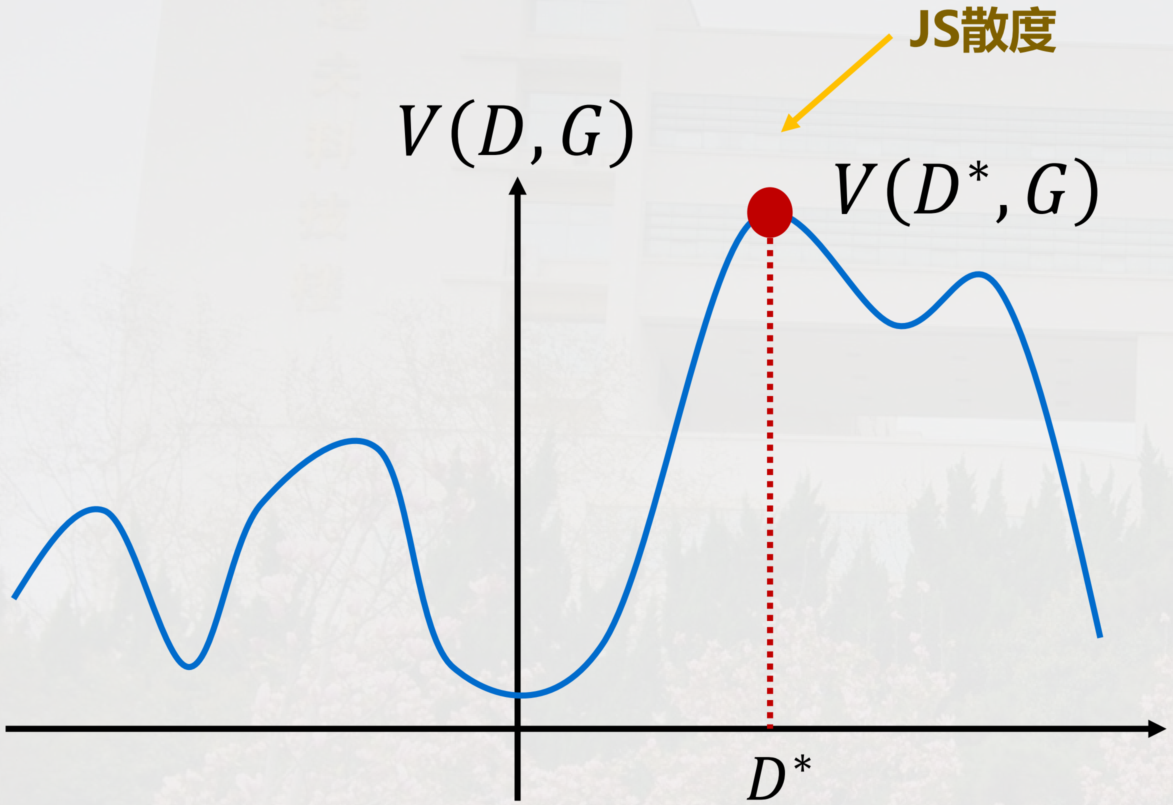
$$\min_G \max_D V(D,G) = \min_G \max_D E_{x\sim P_{data}(x)}[logD(x)] + E_{z\sim P_z(z)}[\log(1-D(G(z)))]$$

图像化理解刚才的结果：



- 对判别器而言，目标函数是最大化 $V(D,G)$，找到左图函数的最高点，对应的判别器为D^*。
- 代入生成器目标函数，得到此时的高度 $V(D^*,G)$。
- 该高度表示生成分布与真实分布的JS散度
- 对生成器而言，目标就是训练G最小化这个JS散度。

4 GAN的训练过程



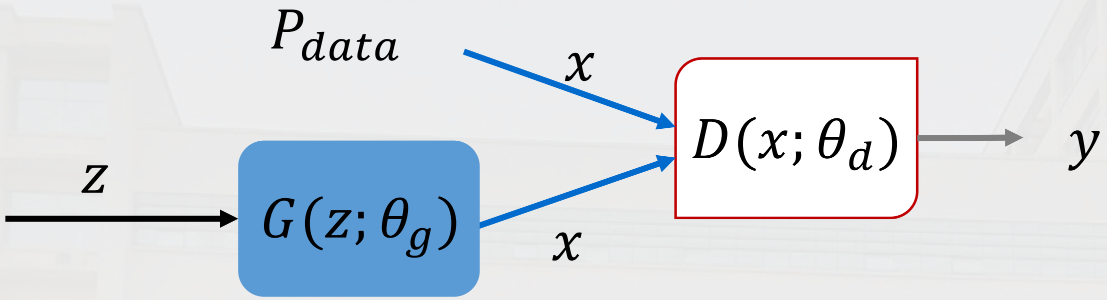
通过判别器找到当前分布与真实分布的JS散度

通过生成器生成数据构成新的生成分布

减小生成分布与真实分布之间的JS散度

怎么求解？　梯度下降法

4 GAN的训练过程



$$\min_G \max_D E_{x\sim P_{data}(x)}[logD(x;\theta_d)] + E_{z\sim P_z(z)}[\log(1 - D(G(z;\theta_g);\theta_d))]$$

- 对于生成器 G，目标是最小化$\max_D V(D,G)$，将$\max_D V(D,G)$记为$\mathcal{L}(G)$，可以采用梯度下降法求解参数：

$$\theta_g = \theta_g - \eta \frac{\partial \mathcal{L}(G)}{\partial \theta_g}$$

4 GAN的训练过程

训练过程可总结如下：

1）固定生成器G，训练判别器D,获得 $\max_{D} V(D,G)$。

2）固定判别器D，训练生成器G,通过梯度下降求解更新参数：

$$\theta_g = \theta_g - \eta \frac{\partial \mathcal{L}(G)}{\partial \theta_g} = \theta_g - \eta \nabla \mathcal{L}(\theta_g)$$

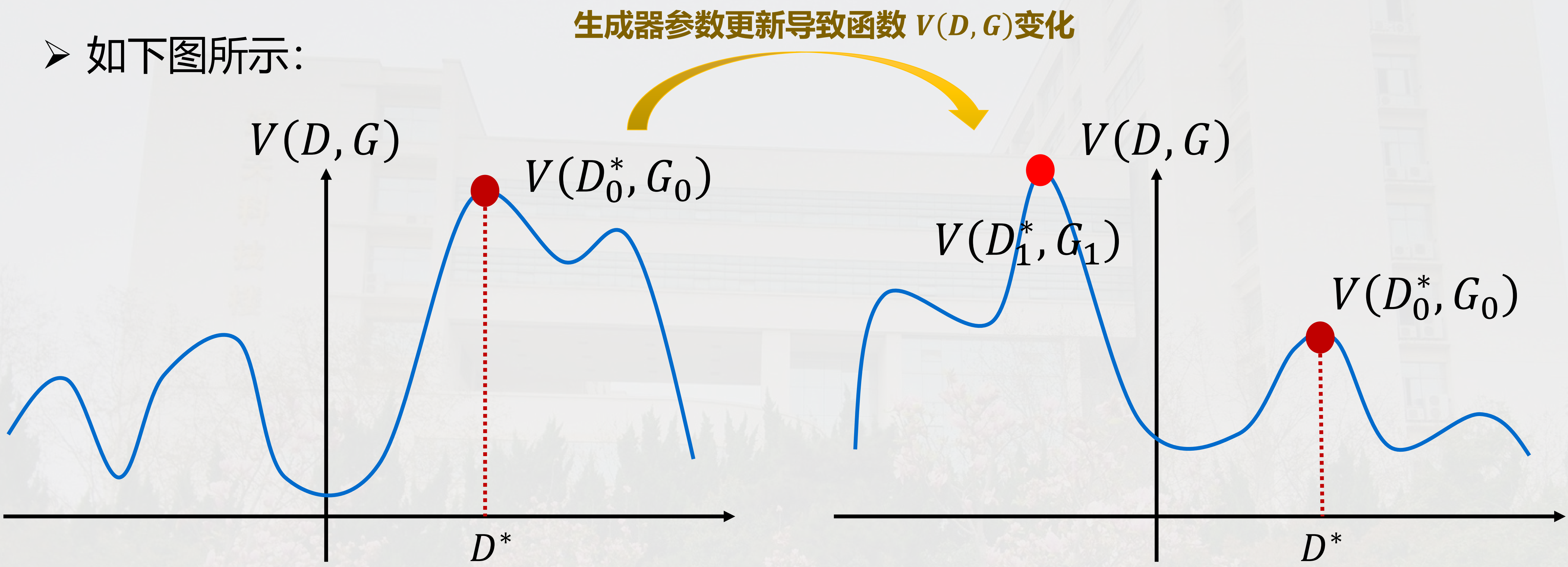
3）重复循环上面两步直到模型收敛。

思考：

如2)中方法，如果固定判别器 D 后训练很多次，生成器G的的参数会更新多次，导致函数$V(D,G)$本身发生变化，而此时判别器参数是固定的，对于判别器而言，变化后函数$V(D,G)$所在的点就不是之前图中最大值所在的点。

4 GAN的训练过程

➢ 如下图所示：



解决方法：训练时减少生成器G的训练次数，相当于减小函数$V(D,G)$的变化，近似的将D_0^*看成变动后 $V(D,G)$最大值的近似值。

4 GAN的训练过程

➢ GAN的训练过程可进一步总结如下：

1）通过抽样方式获得样本，真实样本$x_1, x_2, \dots, x_N \in P_{data}(x)$,噪声样本 $z_1, z_2, \dots, z_N \in P_z(z)$,
生成样本$x'_1, x'_2, \dots, x'_N \in P_g(x)$

2）固定生成器G，训练判别器，以样本分布来近似表示真实分布与生成分布：

$$\max_D V'(D, G) = \frac{1}{N}\sum_{i=1}^{N} \log D(x_i) + \frac{1}{N}\sum_{i=1}^{N} \log(1 - D(x'_i))$$

更新参数： $\theta_d = \theta_d + \eta \nabla V'(\theta_d)$

3）固定判别器D，训练生成器G：

$$\min_G V'(D^*, G) = \frac{1}{N}\sum_{i=1}^{N} \log(1 - D(G(z_i)))$$

更新参数： $\theta_g = \theta_g - \eta \nabla V'(\theta_g)$

4）重复循环上面两步直到模型收敛。

Note: 训练多次判别器后才会训练一个生成器

4 GAN的训练过程

➢ 统一框架F-GAN, 使用 f 散度（包含JS散度）的形式来一般化表示GAN的目标函数，f散度函数：

$$\mathrm{D}_f(P \parallel Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

当f散度函数满足下面两个条件时，可以使用$\mathrm{D}_f(P \parallel Q)$衡量两种概率分布之间的差异：

❑ f函数是一个凸函数；

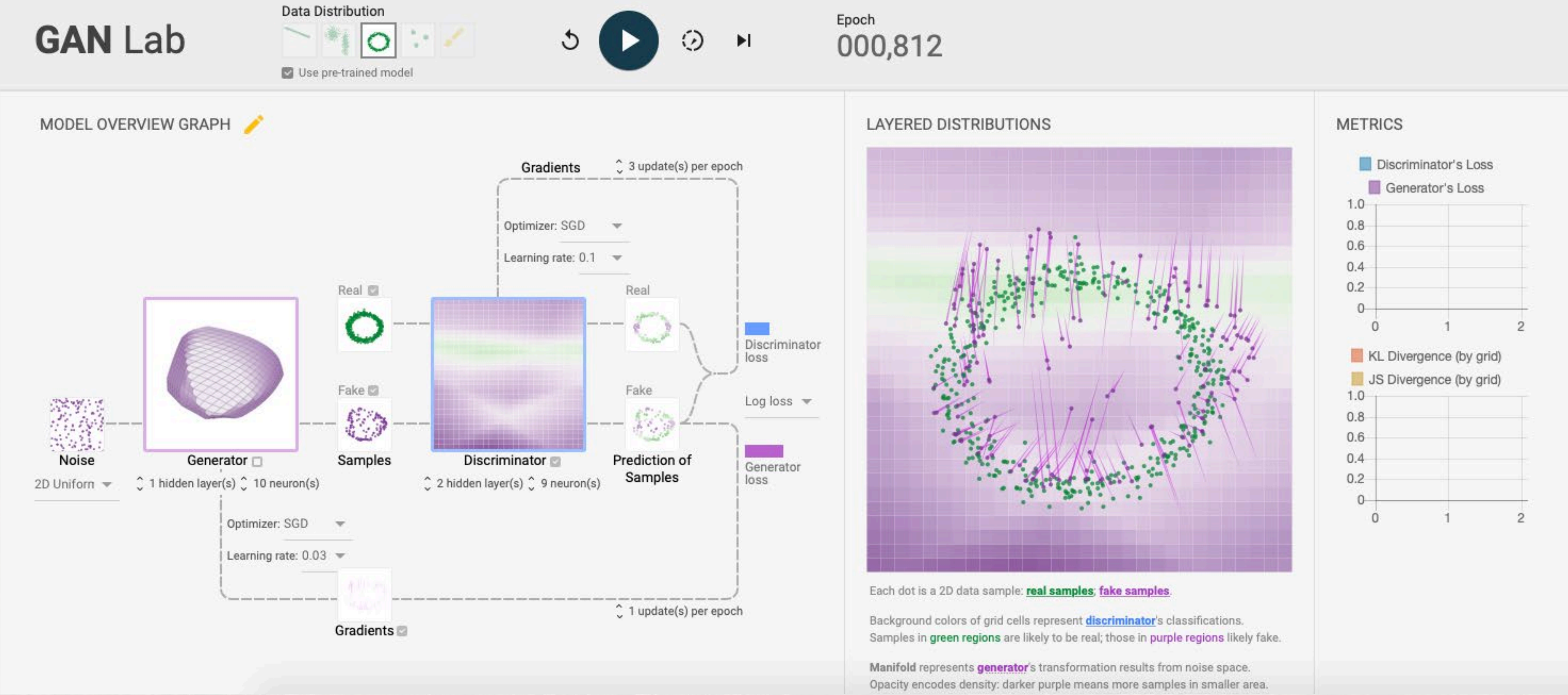
❑ f(1)=0。

➢ 统一框架F-GAN，使用f散度（包含JS散度）的形式来一般化表示GAN的目标函数

度量	表达式	f(u)
总变差	$\frac{1}{2}\int_x \mid p_{data}(x)-p_g(x) \mid dx$	$\frac{1}{2}\mid u-1 \mid$
KL散度	$\int_x p_{data}(x)log[\frac{p_{data}(x)}{p_g(x)}]dx$	$ulog\ u$
逆KL散度	$\int_x p_g(x)log[\frac{p_g(x)}{p_{data}(x)}]dx$	$-log\ u$
Pearson χ^2	$\int_x \frac{(p_g(x)-p_{data}(x))^2}{p_{data}(x)}dx$	$(u-1)^2$
Neyman χ^2	$\int_x \frac{(p_{data}(x)-p_g(x))^2}{p_g(x)}dx$	$\frac{(u-1)^2}{u}$
Hellinger距离	$\int_x(\sqrt{p_{data}(x)}-\sqrt{p_g(x)})^2dx$	$(\sqrt{u}-1)^2$
Jeffrey	$\int_x(p_{data}(x)-p_g(x))log[\frac{p_{data}(x)}{p_g(x)}]dx$	$(u-1)log\ u$
JS散度	$\frac{1}{2}\int_x p_{data}(x)log\frac{2p_{data}(x)}{p_{data}(x)+p_g(x)}+ p_g(x)log\frac{2p_g(x)}{p_{data}(x)+p_g(x)}dx$	$-(u+1)log\frac{1+u}{2}+ ulog\ u$
α散度	$\frac{1}{\alpha(\alpha-1)}\int_x p_{data}(x)[(\frac{p_g(x)}{p_{data}(x)})^\alpha-1]-\alpha(p_g(x)- p_{data}(x))dx$	$\frac{1}{\alpha(\alpha-1)}(u^\alpha-1- \alpha(u-1))$

4 GAN的训练过程和结果可视化

- GAN lab: https://poloclub.github.io/ganlab/



Minsuk Kahng, Nikhil Thorat, Polo Chau, Fernanda Viégas, and Martin Wattenberg. "GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation." *IEEE Transactions on Visualization and Computer Graphics, 25(1) (VAST 2018)*, Jan. 2019.

4 GAN的训练结果

➢ 创建动画角色

Figure 7: Generated samples

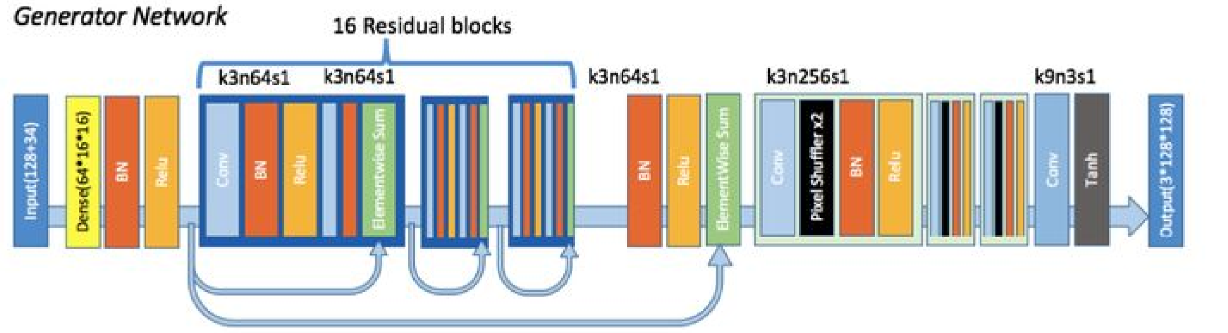


Figure 3: Generator Architecture

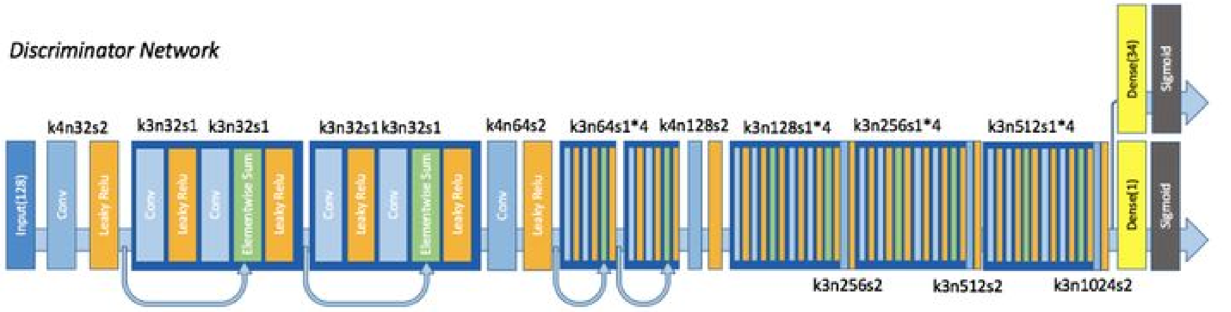
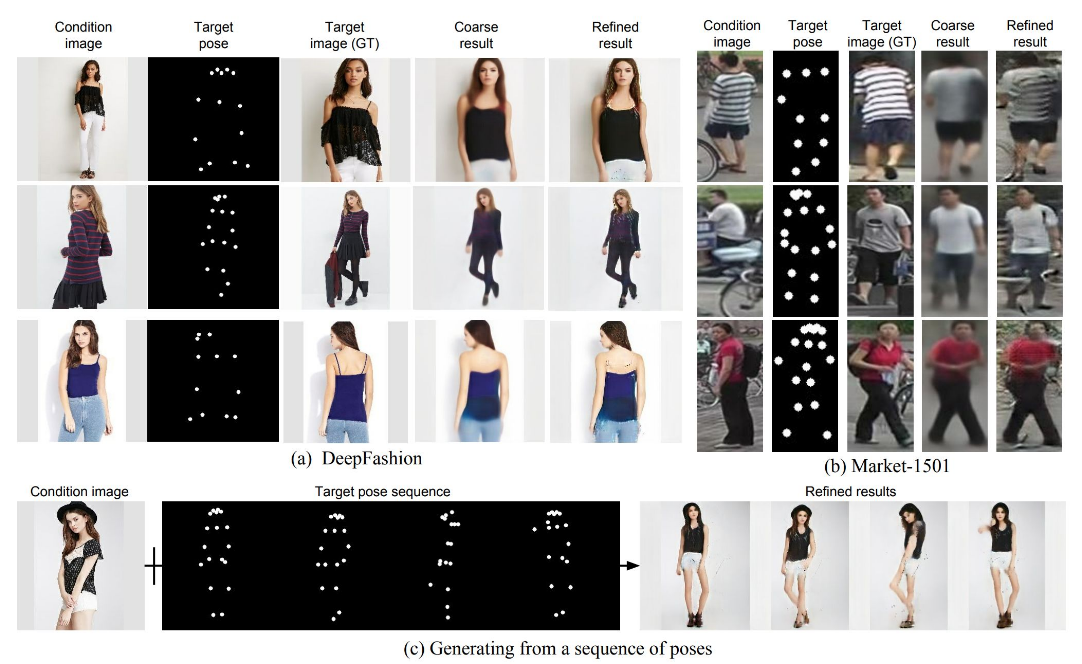


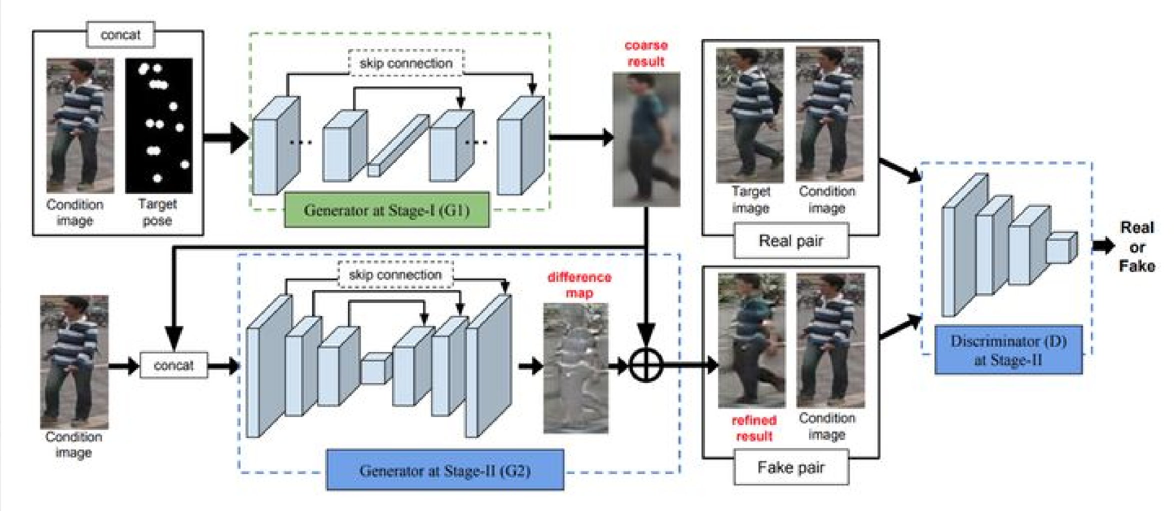
Figure 4: Discriminator Architecture

DCGAN （Deep Convolutional GAN）采用深度网络

4 GAN的训练结果

➢ 改变人体形态

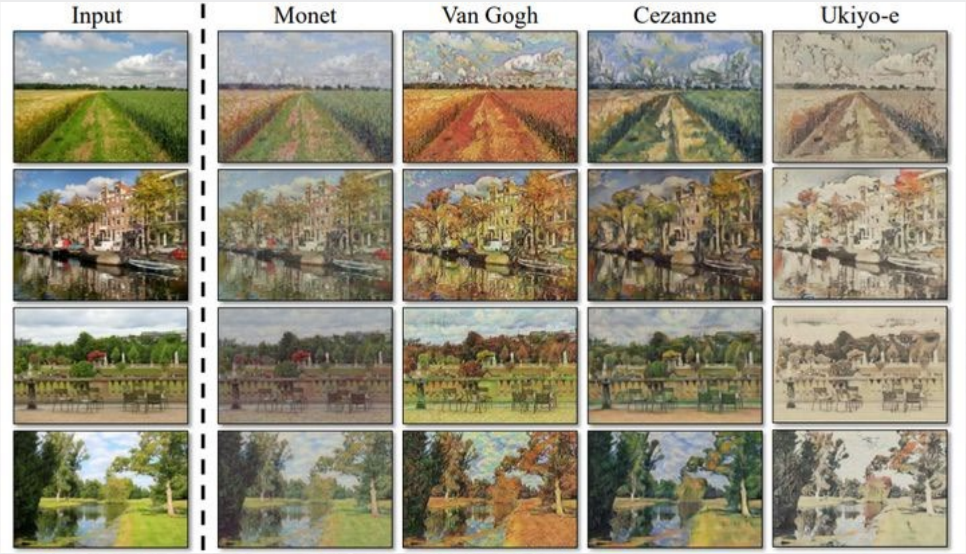


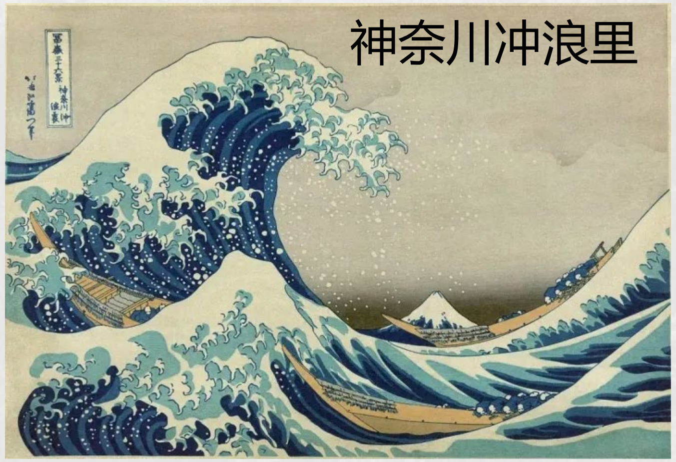


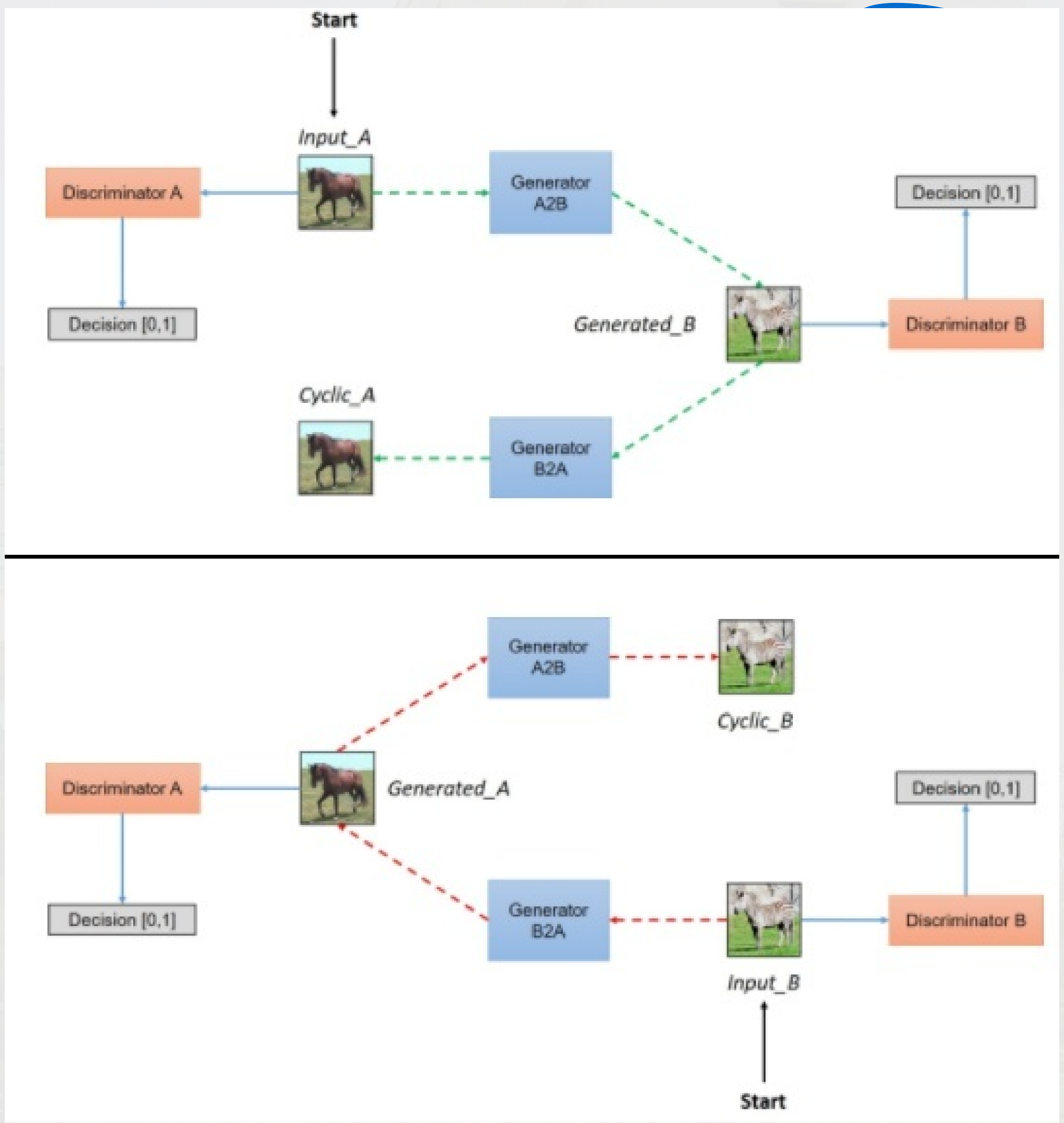
Conditional GAN （CGAN）

4 GAN的训练结果

➢ 风格转换



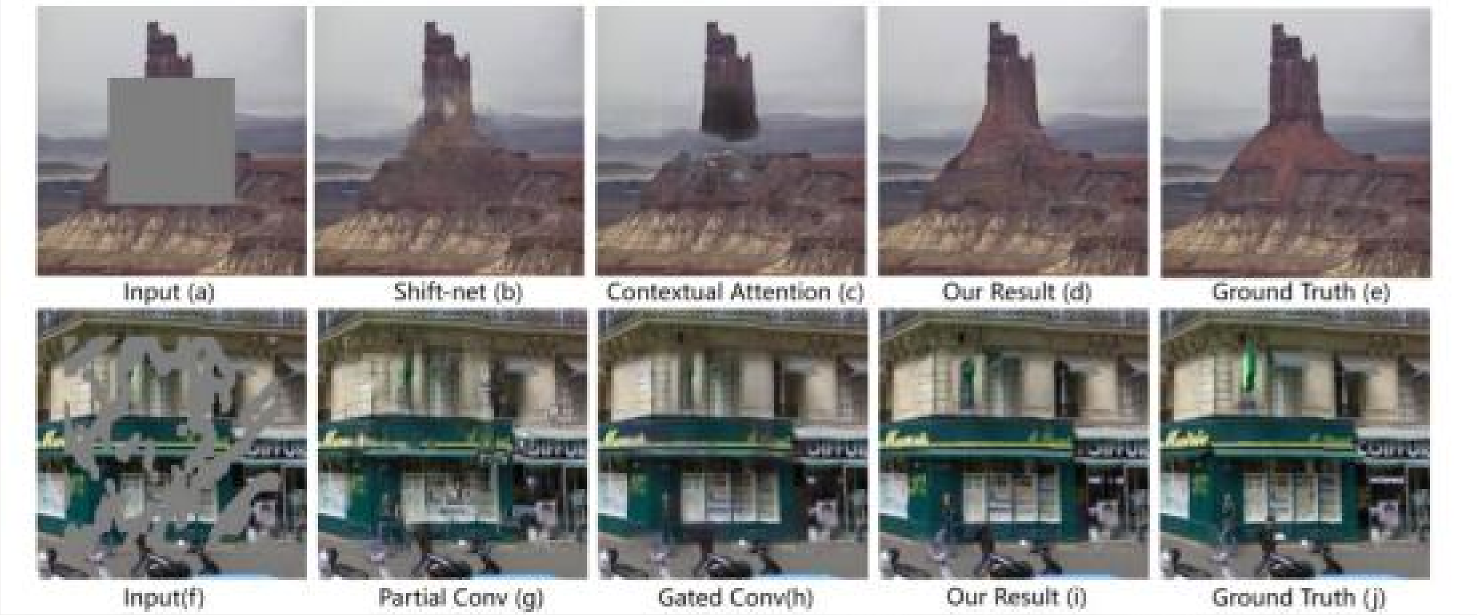


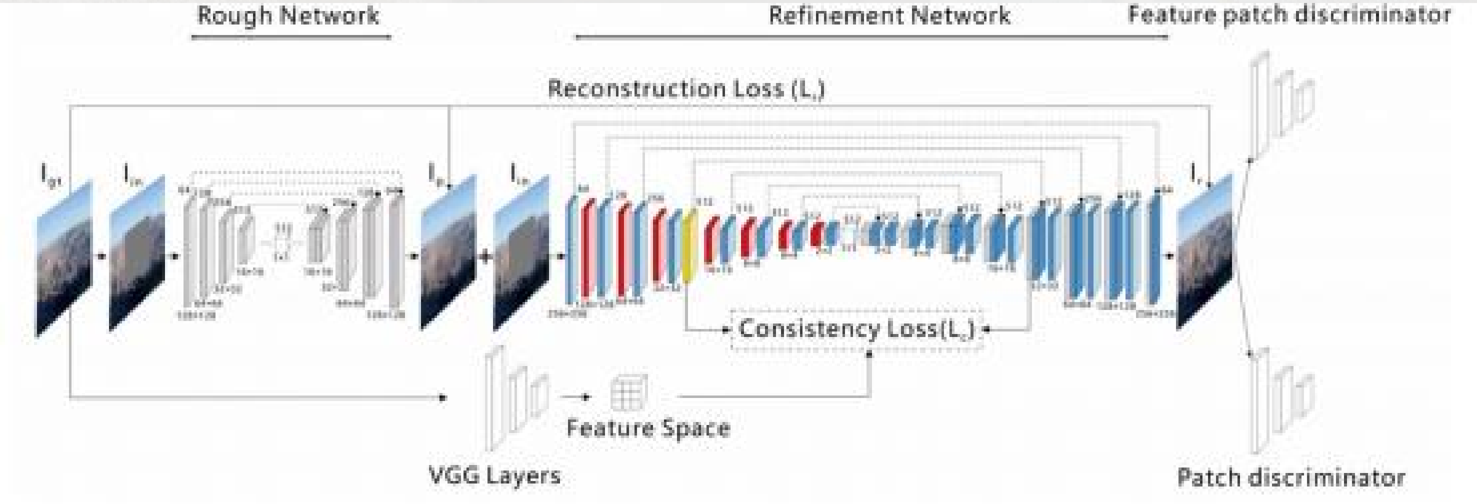


CycleGAN

4 GAN的训练结果

➢ 图像修复



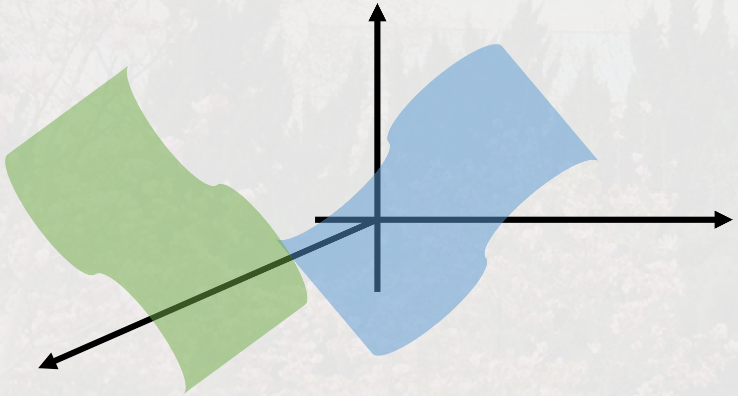


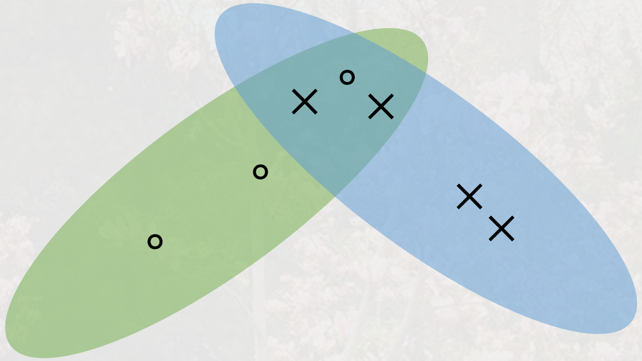
- 传统GAN存在的问题：**梯度消失，JS散度的问题**

传统GAN中训练生成器就是减小真实分布与生成分布的JS散度，从而达到让生成器以假乱真的目的：

$$\min_G \max_D V(D,G) = \min_G V(D^*,G) = \min_G -log4 + 2\mathrm{JS}(P_{data} \parallel \frac{P_{data}(x)+P_g(x)}{2})$$

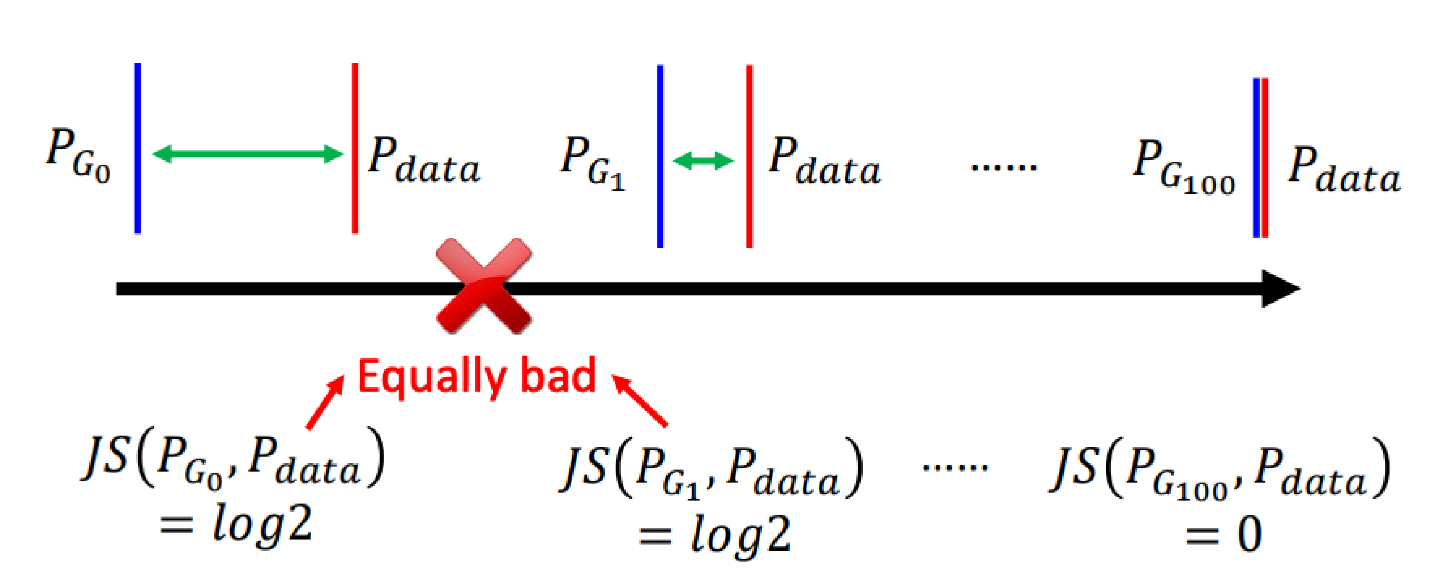
从三维空间理解，两个分布重叠的概率是非常小的，几乎为0，只有交叉点重叠，交叉点比曲线低一个维度，因此交叉点的重叠可以忽略；如果从分布中采样样本点，则重叠几率也会减小





5 GAN的改进

➢ 传统GAN存在的问题： 梯度消失，JS散度的问题



只要两分布没有重叠部分或重叠部分可忽略，JS散度恒为$log2$，目标函数为0。当判别器最优时，生成器无法获得任何梯度信息，梯度消失，无法训练。

➢ 传统GAN存在的问题： 梯度消失，JS散度的问题

解决方法：

1. 不把判别器训练的太好。

2. 给生成数据和真实数据加噪声，强行让生成数据与真实数据在高维空间产生重叠，JS散度就可以发挥作用：

$$\min_G E_{x\sim P_{data+\epsilon}}[logD^*(x)] + E_{x\sim P_{g+\epsilon}}[log(1 - D^*(x))]$$

$P_{data+\epsilon}$表示加入噪声后的真实分布数据，$P_{g+\epsilon}$表示加入噪声后的生成分布数据。

5 GAN的改进

➢ 传统GAN存在的问题： 模式崩溃(mode collaspe)

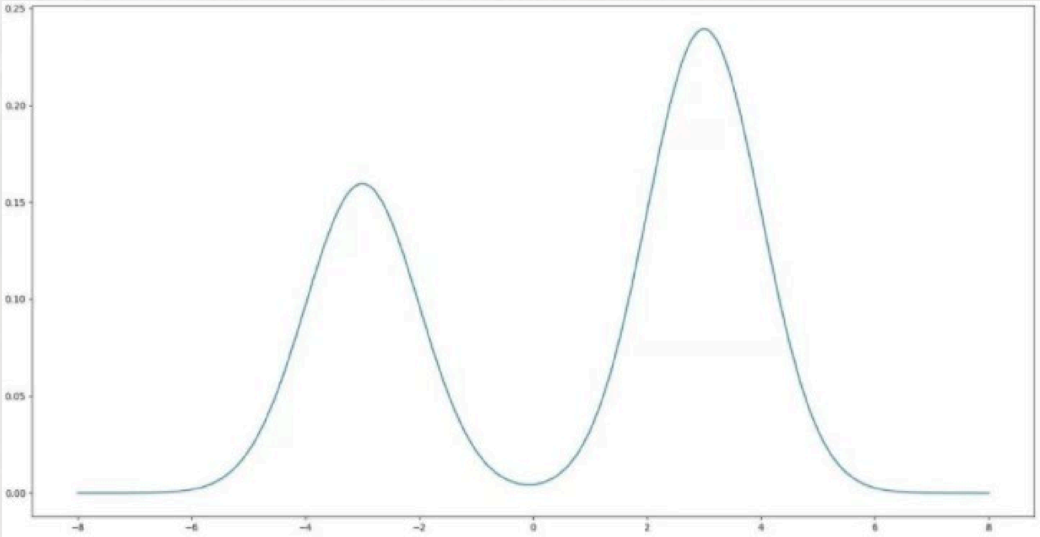
生成图像中相同的图像多次出现，如果继续训练GAN网络，会生成更多相同或相近的图像，发生模式崩溃，造成生成数据多样性不足。

5 GAN的改进

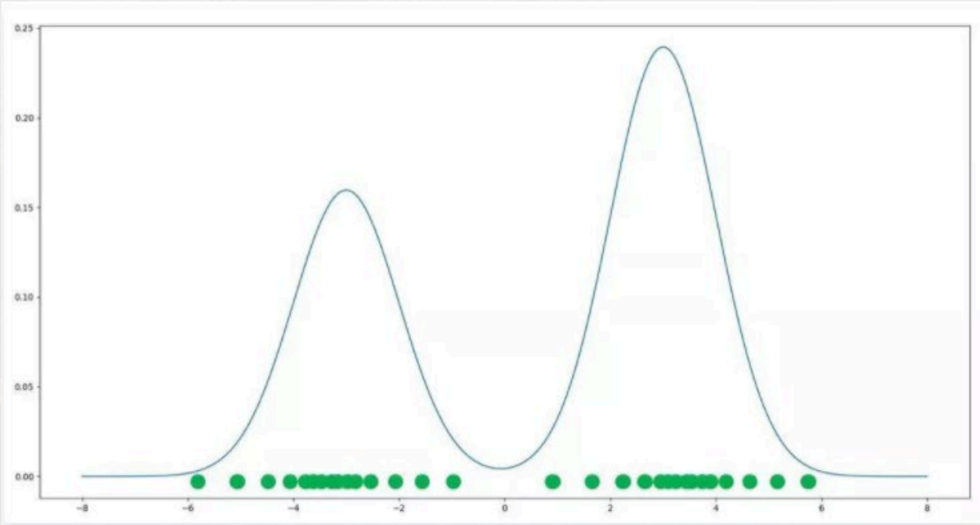
➢ 传统GAN存在的问题： 模式崩溃(mode collaspe)

生成图像中相同的图像多次出现，如果继续训练GAN网络，会生成更多相同或相近的图像，发生模式崩溃，造成生成数据多样性不足。

举例：对于某一个训练数据集，其中样本的概率分布为一个简单的一维高斯混合分布，包含两个峰：



理想状态，生成器生成样本（绿色标记）

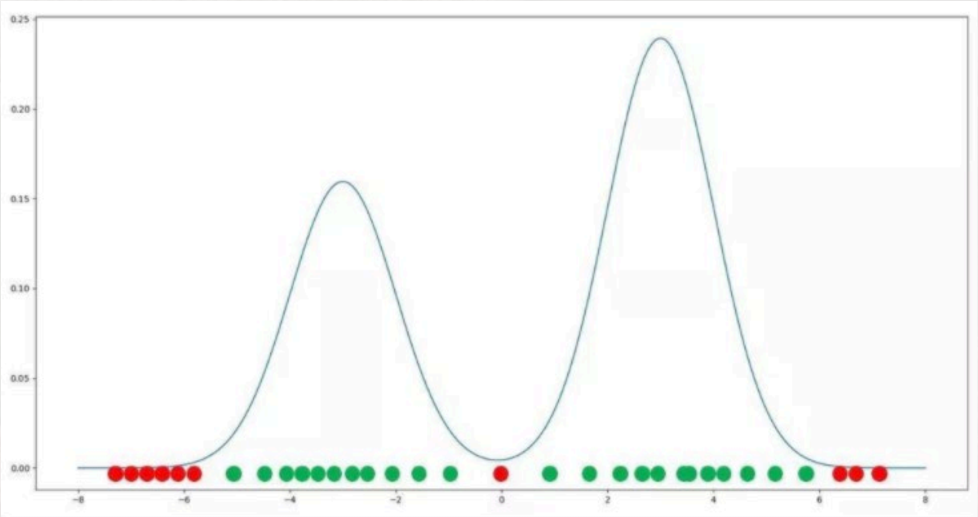


➢ 传统GAN存在的问题： 模式崩溃(mode collaspe)

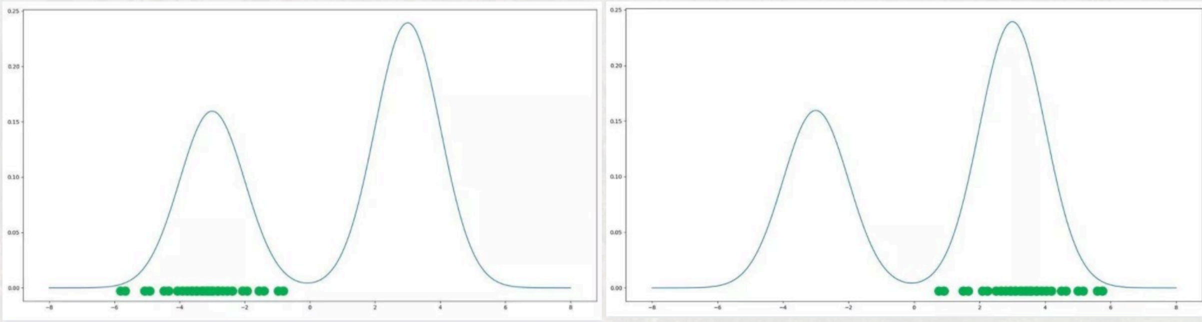
生成图像中相同的图像多次出现，如果继续训练GAN网络，会生成更多相同或相近的图像，发生模式崩溃，造成生成数据多样性不足。

举例：对于某一个训练数据集，其中样本的概率分布为一个简单的一维高斯混合分布，包含两个峰：

正常训练



模式崩溃



- 传统GAN存在的问题： 模式崩溃(mode collaspe)

解决方法：

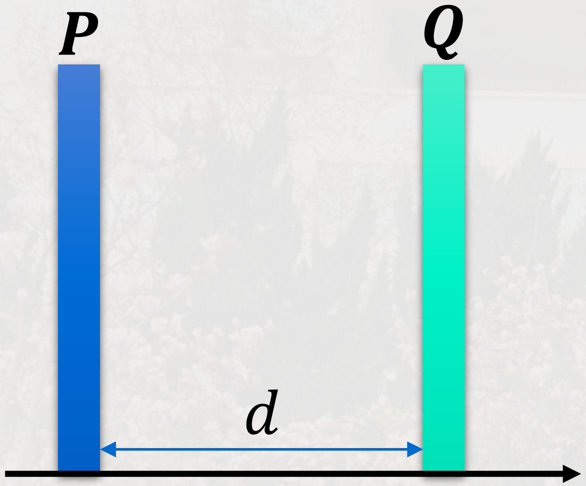
1) 从目标函数考虑：经验发现，当GAN出现模式崩溃问题时，通常判别器在真实样本附近更新参数时，其梯度值非常大。可对判别器在真实样本附近施加梯度惩罚项。试图在真实样本附近构建线性函数，因为线性函数为凸函数具有全局最优解。（DRAGAN）

2) 从网络架构考虑：即使单个生成器会产生模式崩溃的问题，但是如果同时构造多个生成器，且让每个生成器产生不同的模式，则这样的多生成器结合起来也可以保证产生的样本具有多样性。（MAD (Multi-agent diverse)-GAN）

5 GAN的改进

- Wasserstein GAN (WGAN)的提出就是为了解决传统GAN遇到的梯度消失、训练时梯度不稳定以及模式崩溃等问题。
- 使用Wasserstein距离来衡量两分布之间的距离，抛弃了KL散度与JS散度。

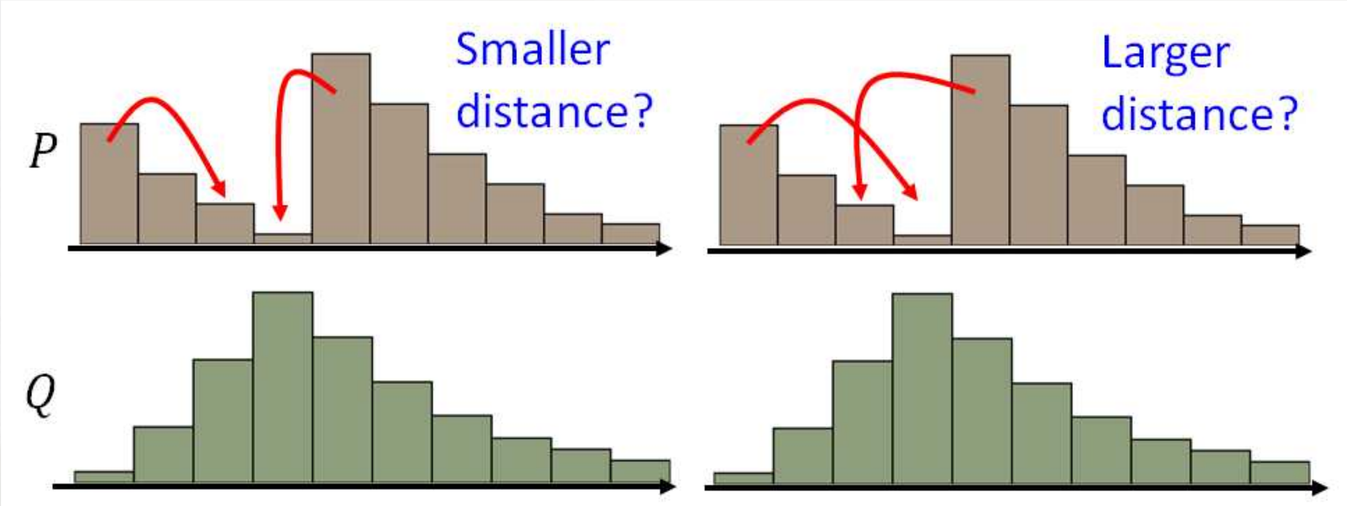
Wasserstein距离也叫EM距离（Earth Mover’s Distance, 推土距离)，假设数据P分布在一维空间中的某个点，数据Q也分布在一维空间中的某个点，两组数据的分布之间的距离就是d。



$$W(P,Q) = d$$

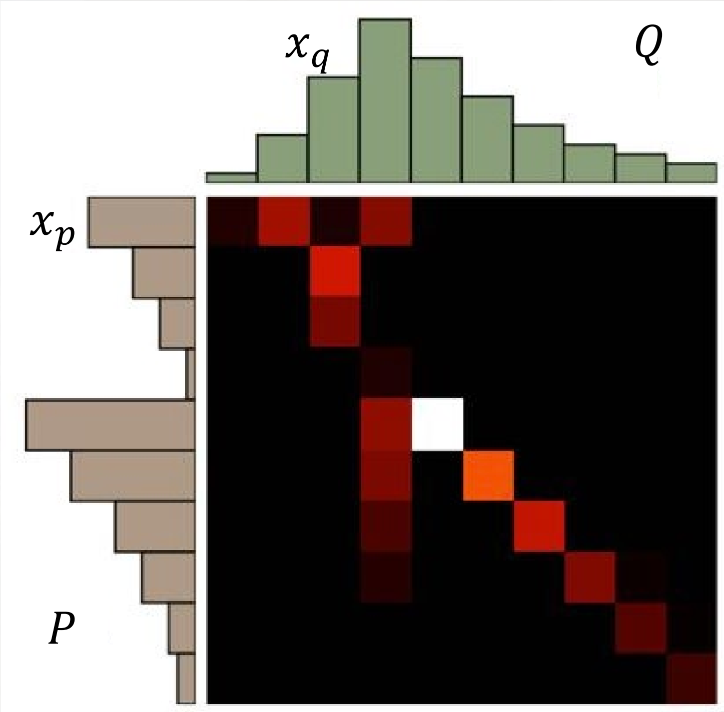
5 GAN的改进

Wasserstein距离也叫EM距离（Earth Mover’s Distance, 推土距离)，假设数据P分布在一维空间中的某个点，数据Q也分布在一维空间中的某个点，两组数据的分布之间的距离就是d。



每一种推土的方法称为一个Moving plan，穷举所有Moving plans，计算出所有Moving plans将数据P推到Q时产生的总代价, 选取最小的总代价作为两个分布的EM距离。

5 GAN的改进



将数据分布P移到分布Q的总代价：

$$B(\gamma) = \sum_{x_p, x_q} \gamma(x_p, x_q) \| x_p - x_q \|$$

$\gamma(x_p, x_q)$表示推土行为，即此次推土要将分布P的多少数据推到分布 Q，将数据从分布P推到到分布 Q的距离 $\| x_p - x_q \|$。

EM距离：

$$\boxed{W(P, Q) = \min_{\gamma \in \Pi} B(\gamma)}$$

Π表示两种分布所有可能的Moving plans。

- EM距离在GAN上的使用，将判别器目标函数改写为：

$$\max_{D\in 1-Lipschitz} E_{x\sim P_{data}}[D(x)] - E_{x\sim P_g}[D(x)]$$

满足$1-Lipschitz$约束条件。 $Lipschitz$函数定义如下：

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

即要求输入的变化乘以K大于输出的变化，输出变化不能太快，从而保证函数是平滑的。当K=1时，此时$Lipschitz$函数为$1-Lipschitz$。

WGAN中，采取Weight clipping的方式让判别器目标函数更平滑。

➢ WGAN总结，与传统GAN的区别：

1）判别器最后一层去掉sigmoid。

2）生成器与判别器的损失不再取log。

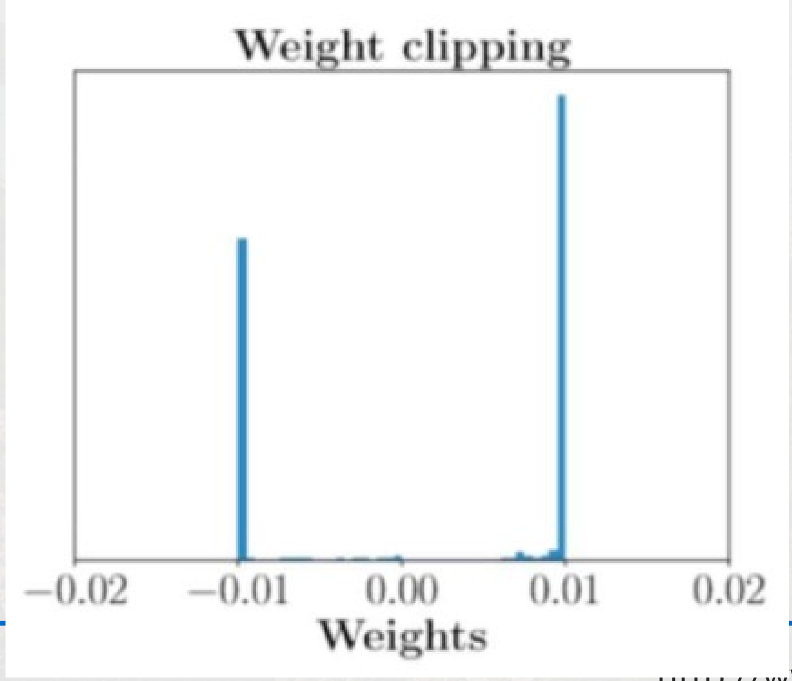
3）weight clipping：训练判别器时，每次参数更新后的值在一个范围(-c,c)，使得D尽量满足$1-Lipschitz$约束条件 。

4）推荐使用RMSProp或SDG优化算法。

5 GAN的改进

- WGAN总结存在的问题，训练困难和收敛缓慢的问题，造成这些问题的原因是Weight clipping：

判别器为了实现返回的损失可以更好的区分真实样本和生成样本，模型参数容易取到最大值或最小值，假设设定范围[-0.01,0.01], 如下图所示会集中在最大值0.01与最小值-0.01上。



➢ WGAN-GP (GP: gradient penalty) 的提出就是为了解决WGAN遇到的问题，不再使用WGAN中Weight clipping的方式来粗暴地限制参数范围。

WGAN-GP： 当判别器D服从1-Lipschitz约束时，等价于判别器D在任意地方的梯度都小于1，公式如下：

$$D \in 1 - Lipschitz \iff \|\nabla_x D(x)\| \leq 1$$

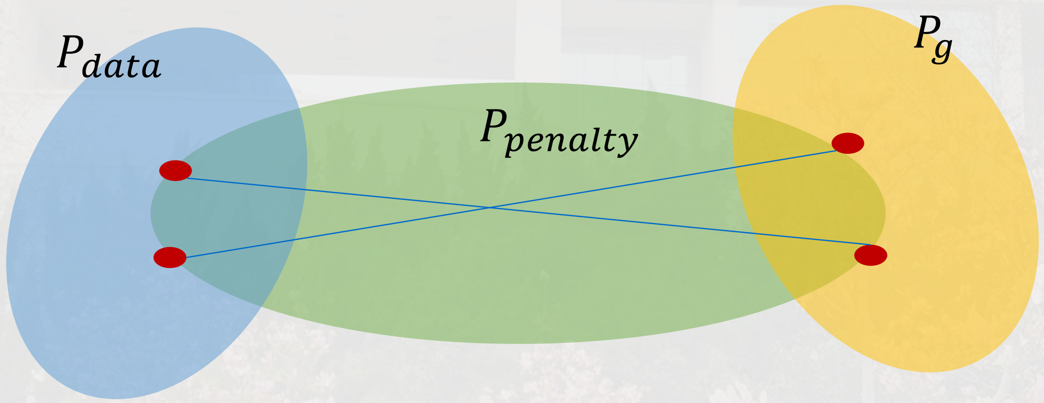
判别器的目标函数可改写为：

$$\max_D E_{x\sim P_{data}}[D(x)] - E_{x\sim P_g}[D(x)] - \lambda \int_x \max(0, \|\nabla_x D(x)\| - 1)\, dx$$

5 GAN的改进

采样最后一项，即定义惩罚样本抽取数据的空间分布 $P_{penalty}$，通过实验发现，$P_{penalty}$ 设置成生成数据空间分布与真实数据空间分布之间的效果比较好，上式可改写为：

$$\max_{D} E_{x\sim P_{data}}[D(x)] - E_{x\sim P_g}[D(x)] - \lambda\, E_{x\sim P_{penalty}}[\max(0, \|\nabla_x D(x)\| - 1)]$$



从生成样本和真实数据空间抽取一些样本点，多个样本之间连线构成的空间就是$P_{penalty}$

总结

- 举例理解了生成对抗网络的运行机制。
- 学习了生成对抗网络的目标函数的构造过程。
- 了解了目标函数的全局最优解。
- 学习了生成对抗网络的训练过程。
- 了解了对传统生成对抗网络的改进方法。