法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。



关注 小象学院

第14讲 生成对抗网络

计算机视觉

主讲人：叶梓

上海交通大学博士
主要研究方向：机器学习、深度学习、人工智能

本章结构

- 变分自编码器
- 对抗生成网络(GAN)
- 常见的改进式GAN
  - DCGAN
  - WGAN
  - SRGAN
- GAN在图像生成中的应用
- GAN在传统CV问题中的应用
- 应用实例：GAN

判别式模型与生成式模型

- 判别式模型：已知观察变量X和隐含变量z，它对p(z|X)进行建模，它根据输入的观察变量x得到隐含变量z出现的可能性。
  - 根据原始图像推测图像具备的一些性质，例如根据数字图像推测数字的名称等；
- 生成式模型则相反，它对p(X|z)进行建模，输入是隐含变量，输出是观察变量的概率。
  - 通常给出的输入是图像具备的性质，而输出是性质对应的图像。

生成模型

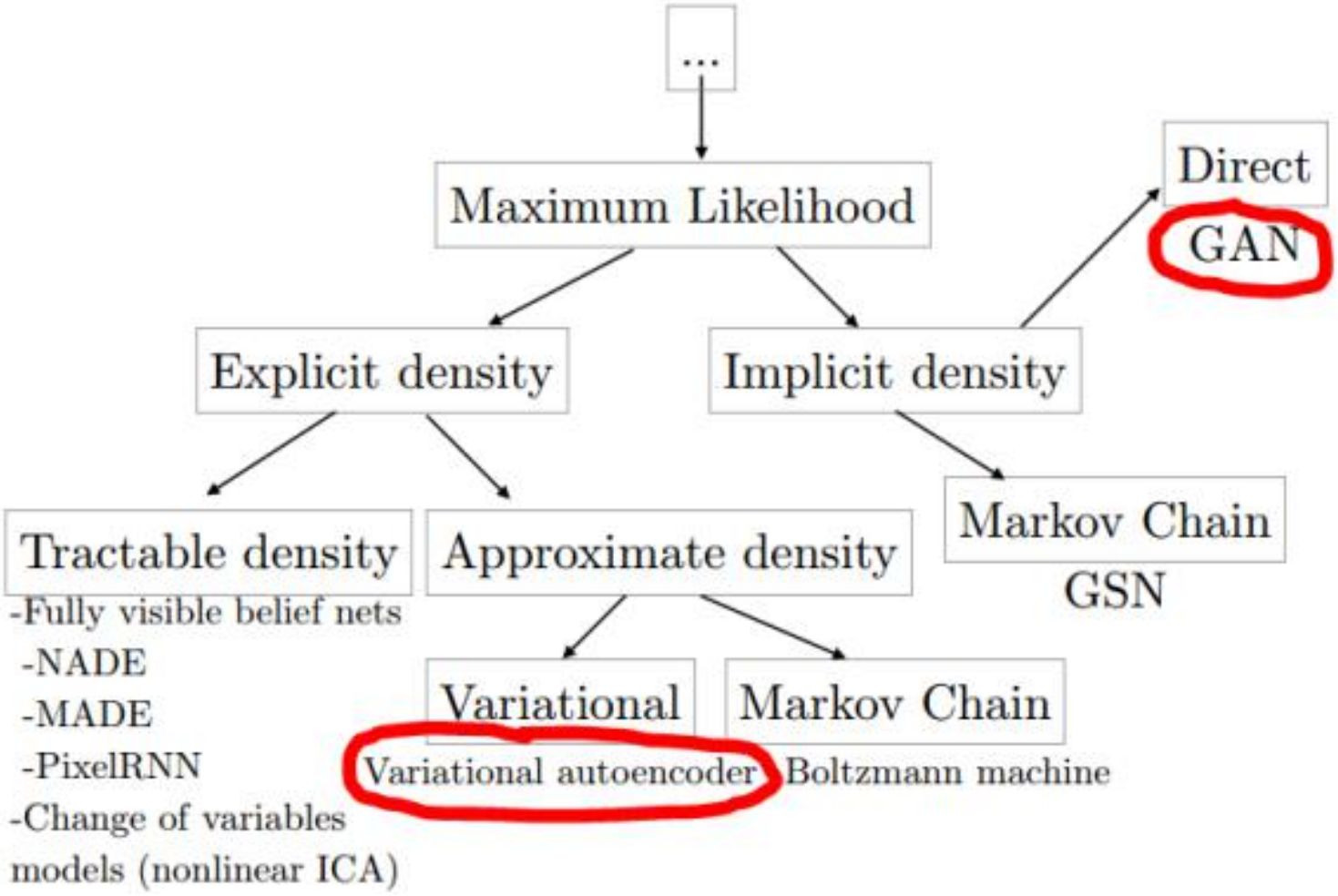
模型目标：

- 训练数据集的模型： $\boldsymbol{x} \sim p_{\text{train}}(\boldsymbol{x})$
- 生成样本的模型： $\boldsymbol{x} \sim p_{\text{model}}(\boldsymbol{x})$
- 令 $p_{\text{model}}(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x})$

解决问题：

- 构建高维、复杂概率分布
- 数据缺失
- 多模态输出
- 真实输出任务
- 未来数据预测

生成模型


...
Maximum Likelihood
Direct
GAN
Explicit density
Implicit density
Tractable density
Approximate density
Markov Chain
GSN
-Fully visible belief nets
-NADE
-MADE
-PixelRNN
-Change of variables models (nonlinear ICA)
Variational
Markov Chain
Variational autoencoder
Boltzmann machine

MLE

- 样本集中的样本都是独立同分布，可以只考虑一类样本集D，来估计参数向量θ。记已知的样本集为：

$$D=\{x_1,x_2,\cdots,x_N\}$$

- 似然函数（linkehood function）：联合概率密度函数p(D|θ)称为相对于的θ的似然函数。

$$l(\theta)=p(D|\theta)=p(x_1,x_2,\cdots,x_N|\theta)=\prod_{i=1}^{N}p(x_i|\theta)$$

- 如果θ^是参数空间中能使似然函数l(θ)最大的θ^值，则θ^应该是"最可能"的参数值，那么θ^就是θ的极大似然估计量。

MLE

- MLE：
- 求使得出现该组样本的概率最大的θ值。

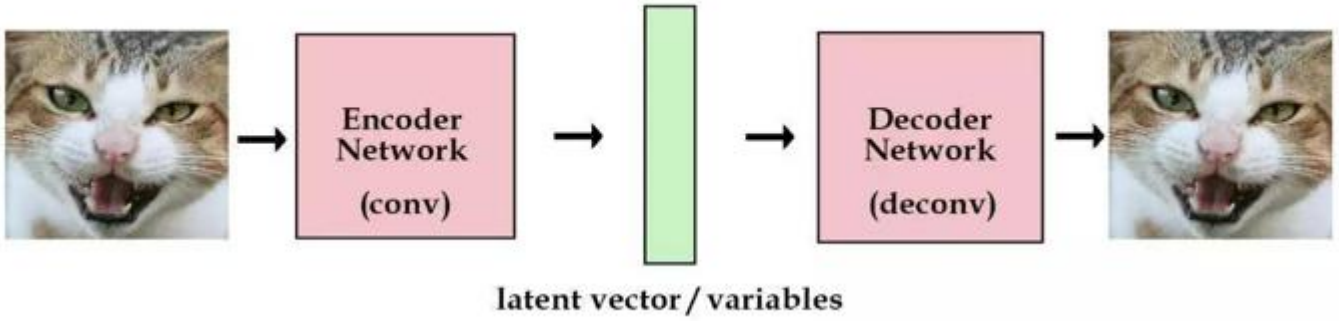
$$\hat{\theta} = \arg\max_{\theta} l(\theta) = \arg\max_{\theta} \prod_{i=1}^{N} p(x_i \mid \theta)$$

- 实际中为了便于分析，定义了对数似然函数

$$\hat{\theta} = \arg\max_{\theta} H(\theta) = \arg\max_{\theta} \ln l(\theta) = \arg\max_{\theta} \sum_{i=1}^{N} \ln p(x_i \mid \theta)$$

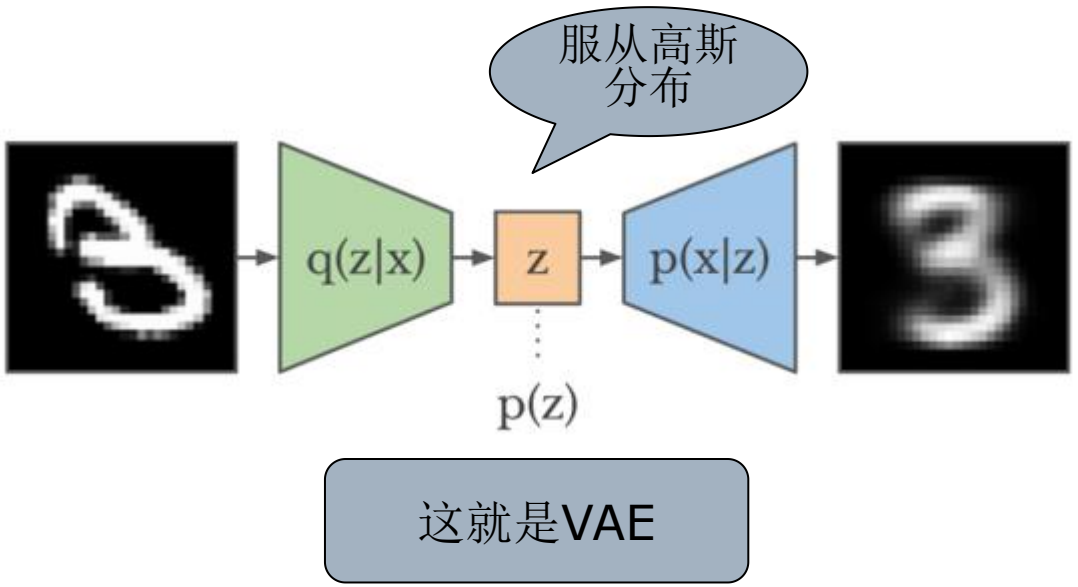
VAE

- 经典的自编码机：
- 左侧把原始图像编码卷积（编码）成向量；解卷积层则能把这些向量"解码"回原始图像。
- 可以用尽可能多的图像来训练网络，如果保存了某张图像的编码向量，我们随时就能用解码组件来重建该图像。



VAE

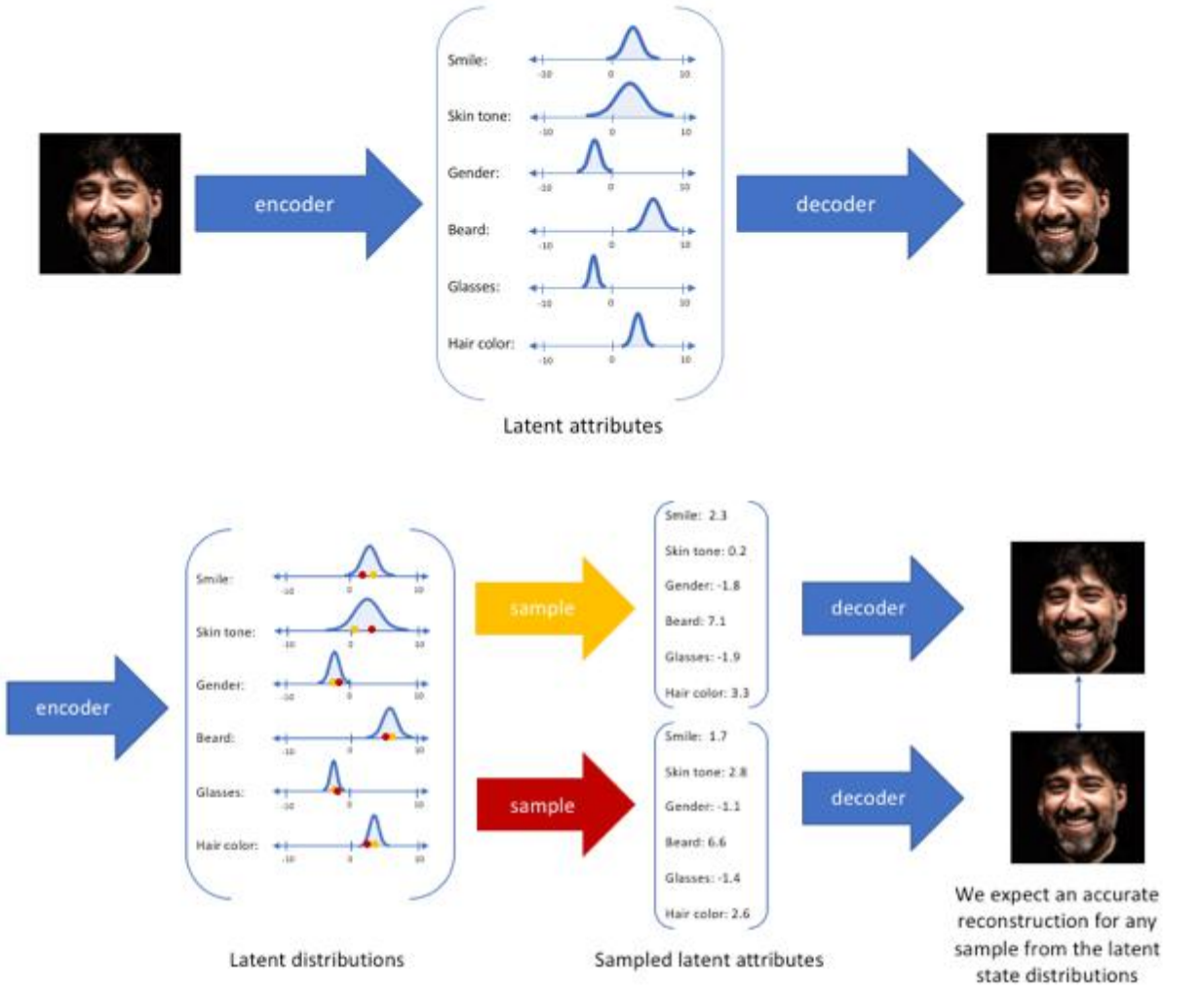
- 问题：潜在向量除了从已有图像中编码得到，能否凭空创造出这些潜在向量来？
- 简单的办法：给编码网络增加一个约束，使它所生成的潜在向量大体上服从单位高斯分布。
- 生成新的图像就变得容易了：只需从单位高斯分布中采样出一个潜在向量，并将其传到解码器。



VAE

- 假定认为输入数据的数据集D(显变量) 的分布完全由一组隐变量 z操控，而这组隐变量之间**相互独立而且服从高斯分布**。
- VAE让 encoder 取学习输入数据的隐变量模型，也就是去学习这组隐变量的高斯概率分布的参数：隐变量高斯分布的均值(μ)和方差(σ)的log值。
- 而隐变量 z 就可以从这组分布参数的正态分布中采样得到：z~ N(μ,σ), 再通过 decoder 对z隐变量进行解码来重构输入。

VAE



本质上是实现了连续的，平滑的潜在空间表示

VAE

- 误差项精确度与潜在变量在单位高斯分布上的契合程度，包括两部分内容：
  - 生成误差，用以衡量网络重构图像精确度的均方误差；
  - 潜在误差，用以衡量潜在变量在单位高斯分布上的契合程度的KL散度。
- 总的目标函数：

$$\mathcal{L}(x,\hat{x}) + \sum_{j} KL\left(q_j\left(z|x\right)||p\left(z\right)\right)$$

VAE

- 假设现在有一个样本集中两个概率分布p、q，其中p为真实分布，q为非真实分布。
- 则按照真实分布p来衡量识别一个样本所需要的编码长度的期望为：

$$\sum_i p(i) \cdot log\left(\frac{1}{p(i)}\right)$$

- 如果采用错误的分布q来表示来自真实分布p的平均编码长度，则应该是：

$$\sum_i p(i) \cdot log\left(\frac{1}{q(i)}\right)$$

- 此时就将H(p,q)称之为交叉熵。

VAE

- **KL散度**又称为**相对熵**，是是两个概率分布P和Q **差别**的非对称性的度量。 典型情况下，P表示数据的真实分布，Q表示数据的理论分布（模型分布或P的近似分布）。

$$\mathrm{D(P||Q)} = \sum P(x) \log \frac{P(x)}{Q(x)}$$

- KL散度不是对称的，并不满足距离的性质；

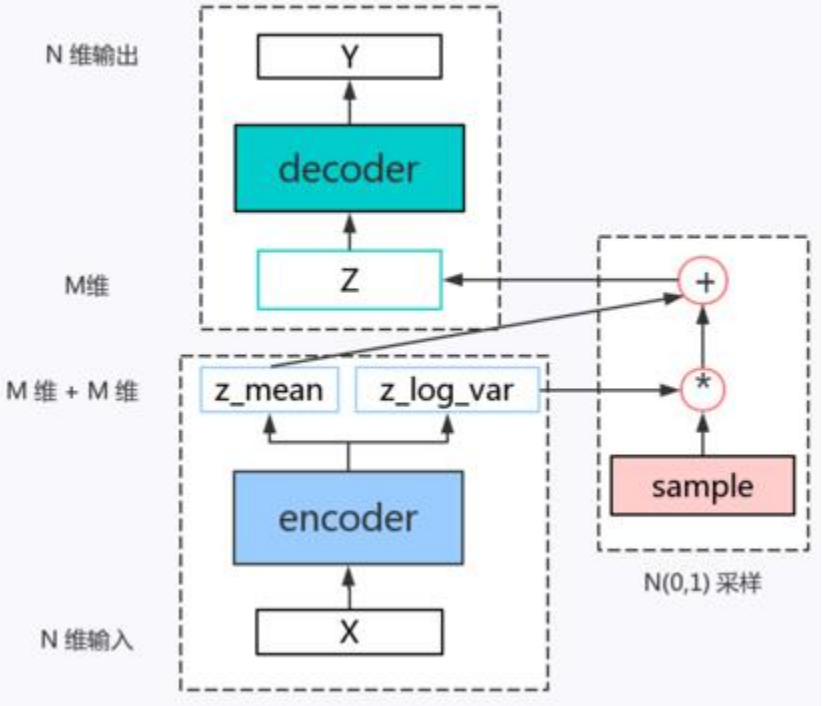
$$D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$$

- **JS散度**度量了两个概率分布的相似度，基于KL散度的变体，解决了KL散度非对称的问题。一般地，JS散度是对称的，其取值是0到1之间。

$$JS(P||Q) = \frac{1}{2}KL(P(x)||\frac{P(x)+Q(x)}{2})+\frac{1}{2}KL(Q(x)||\frac{P(x)+Q(x)}{2})$$

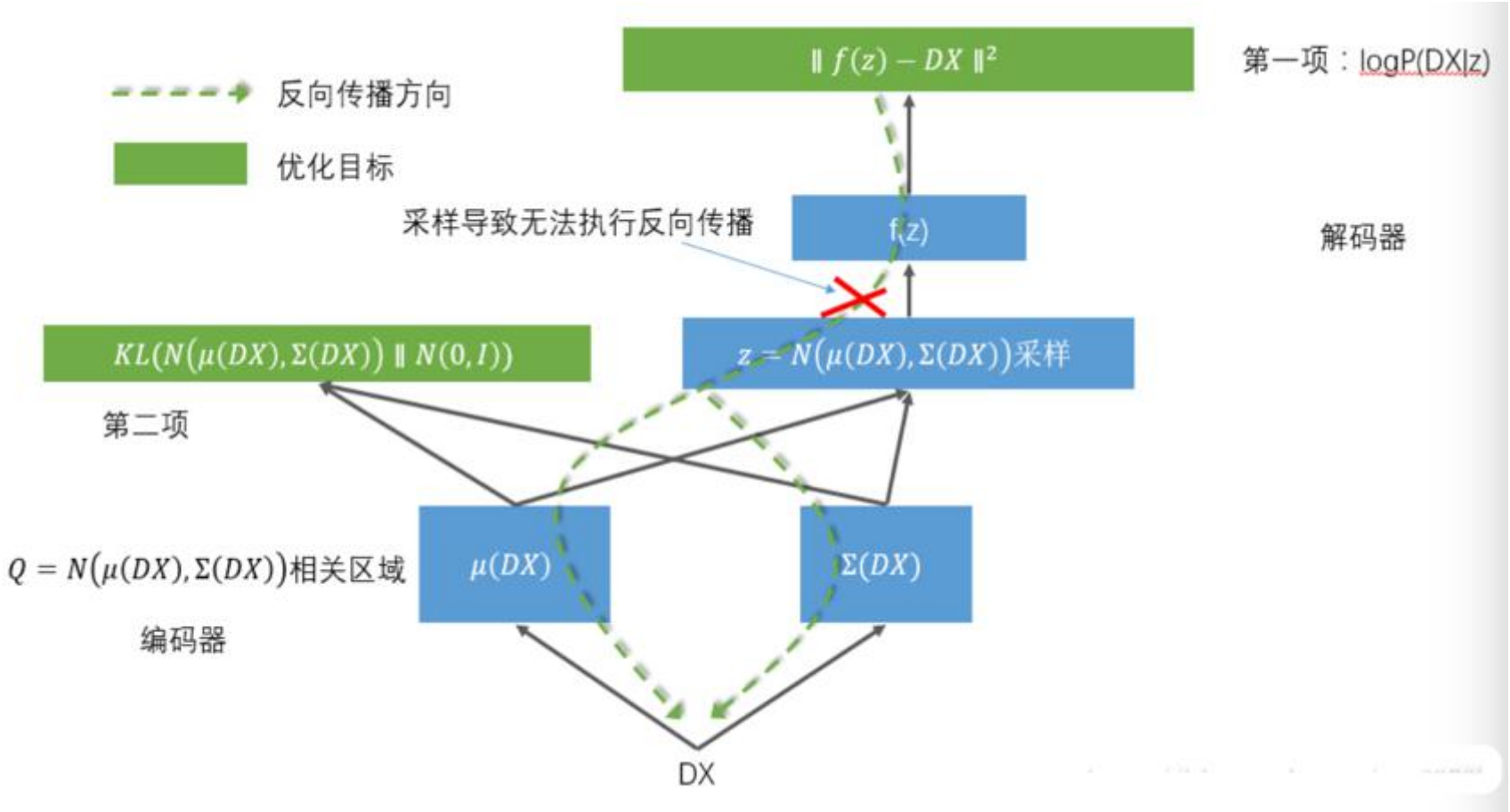
VAE

- Reparameterization：
- VAE模型并没有真正的用z~ N(μ,σ)来采样得到z变量，因为这样采样之后，没有办法对(μ,σ)进行求导。
- 先采样一个标准高斯分布(正态分布):ϵ~ N(0,1),然后z=μ+ϵ* σ，这样得到的z就是服从z~ N(μ,σ)，同时也可以正常的对(μ,σ)进行求导了。



VAE

VAE全过程：


反向传播方向
优化目标
采样导致无法执行反向传播
$\| f(z) - DX \|^2$
第一项：logP(DX|z)
f(z)
解码器
$KL(N(\mu(DX), \Sigma(DX)) \| N(0, I))$
第二项
$z = N(\mu(DX), \Sigma(DX))$采样
$Q = N(\mu(DX), \Sigma(DX))$相关区域
编码器
$\mu(DX)$
$\Sigma(DX)$
DX

VAE

- 优点：遵循“编码-解码”模式，能直接把生成的图像同原始图像进行对比。
- 不足：由于它是直接采用均方误差，其神经网络倾向于生成较为模糊的图像。

第一代



第九代

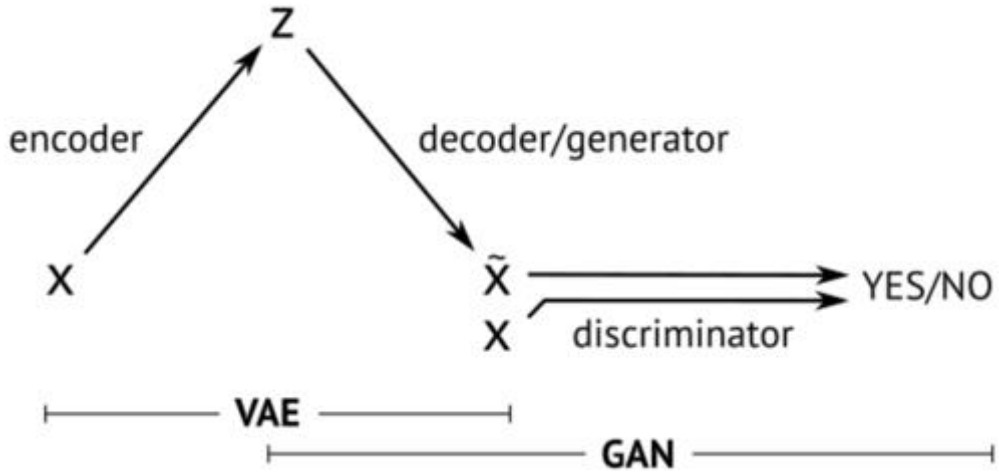


真实样本



VAE与GAN



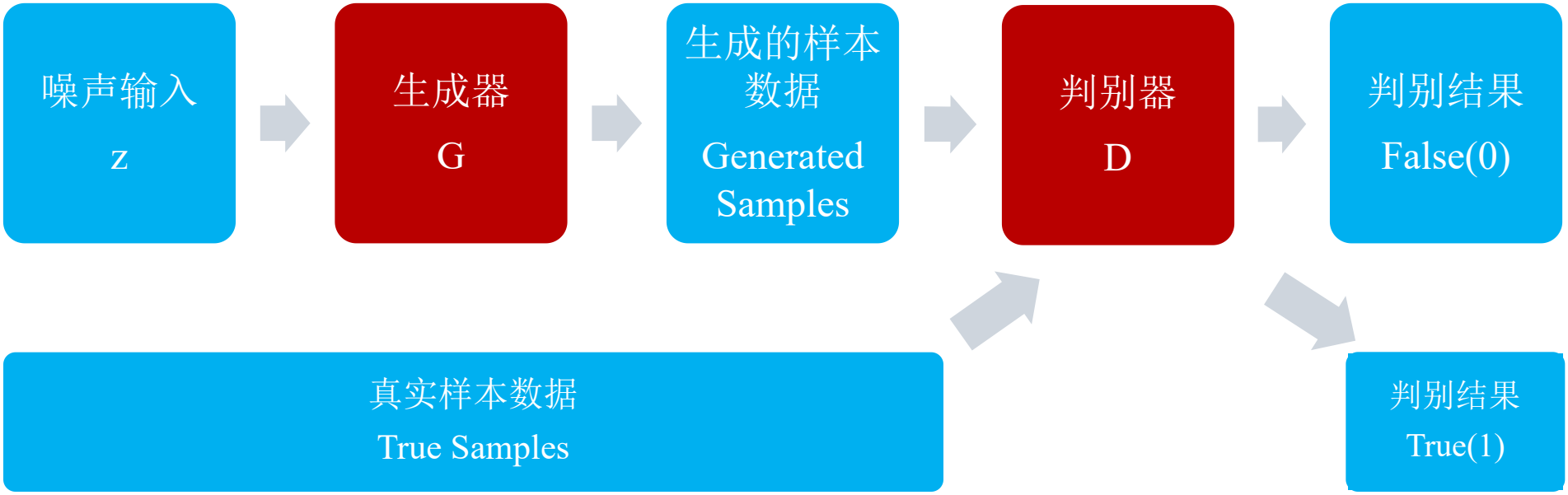


VAE和GAN的区别：GAN是VAE的“后一半”再加上了一个鉴别网络。由此导致了完全不同的训练方式。

生成式对抗网络（GAN）

结构：

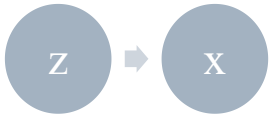
- 生成器（Generator）
- 判别器（Discriminator）



生成式对抗网络（GAN）

生成器网络（Generator Network）：

- 负责生成样本数据
  - 输入：高斯白噪声向量 z
  - 输出：样本数据向量 x



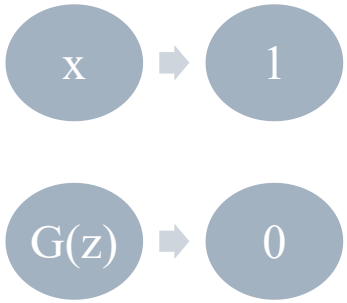
$$\boldsymbol{x} = G(\boldsymbol{z}; \boldsymbol{\theta}^{(G)})$$

- 必须可微分
- 深度神经网络（CNN）

生成式对抗网络（GAN）

判别器网络（Discriminator Network）

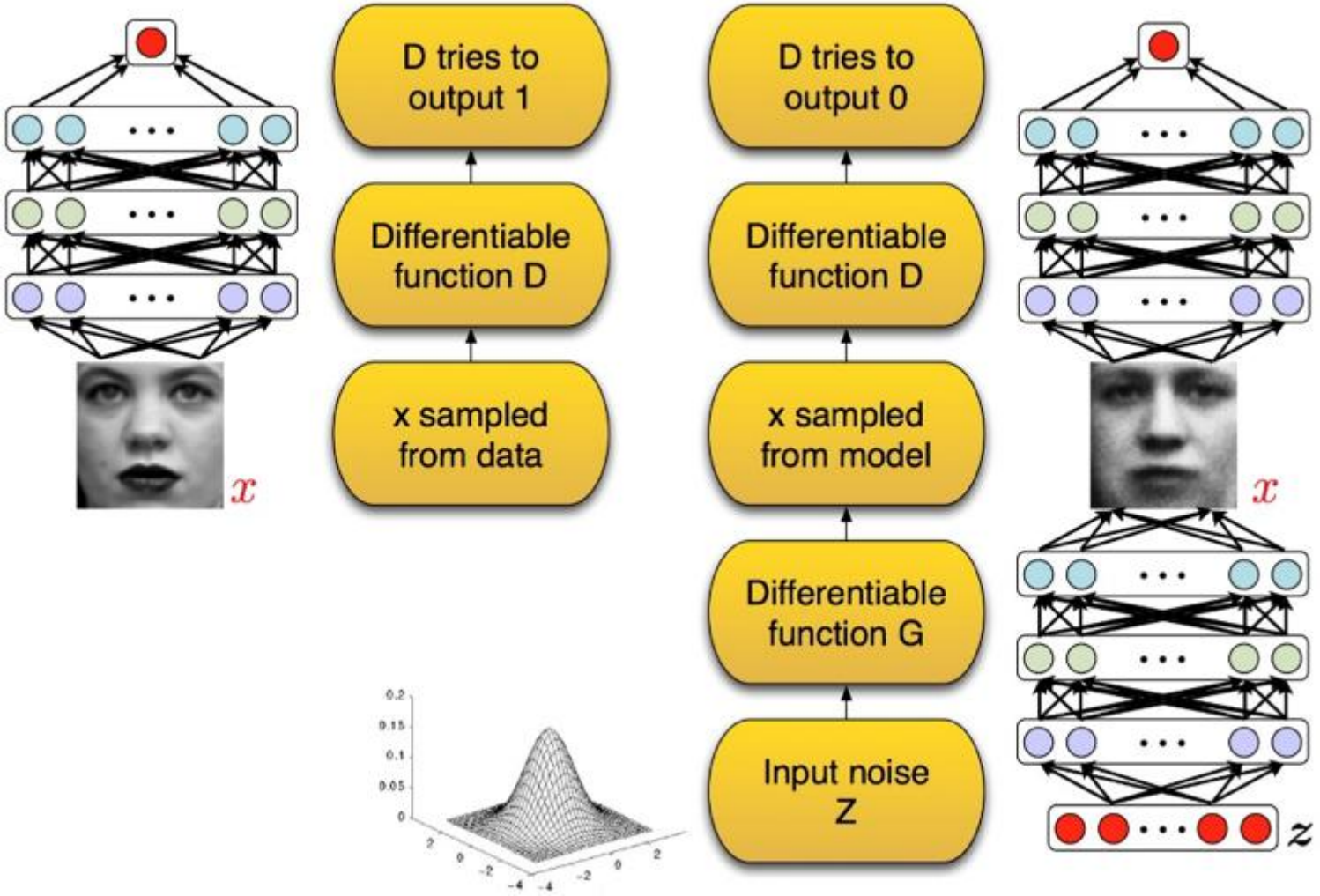
- 负责检测样本数据真假
  - 输入：真实或生成的样本数据
  - 输出：真/假标签



$$y = D(x, \theta^D)$$

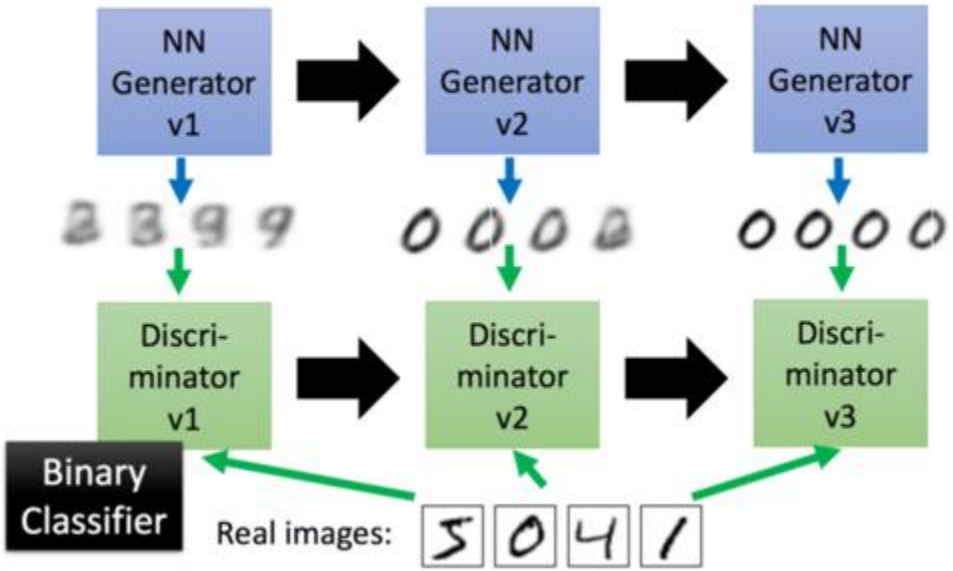
- 必须可微分
- 深度神经网络（CNN）

生成式对抗网络（GAN）


D tries to output 1
Differentiable function D
x sampled from data
x
D tries to output 0
Differentiable function D
x sampled from model
Differentiable function G
Input noise Z
x
z

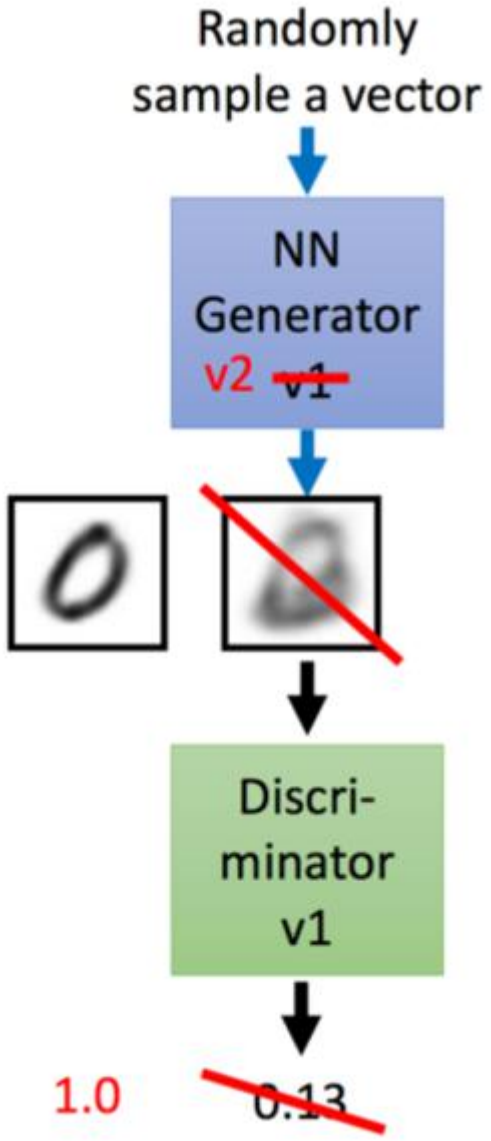
生成式对抗网络（GAN）

- 让第一代G产生一些图片，然后把这些图片和一些真实的图片丢到第一代的D里面去学习，让第一代的D能够分辨生成的图片和真实的图片。
- 然后训练第二代的G，第二代的G产生的图片，能够骗过第一代的D，再训练第二代的D，依此迭代。



生成式对抗网络（GAN）

- 如何训练新一代的G来骗过上一代的D？
- 很简单，可以把新一代的G和上一代的D连起来形成一个新的NN，我们希望最终的输出接近1，训练之……
- 然后我们就可以拿中间的结果当作我们的新的图片的输出。



生成式对抗网络（GAN）

优化目标

- 价值函数（Value Function）

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

- 生成器G固定后，使用$\max_D V(D,G)$来评价p_{data}和p_z之间的差异。
- 优化方式
  - 生成器优化方向：最**小**化价值函数
  - 判别器优化方向：最**大**化价值函数
  - 交替优化，直到达到**纳什均衡**点（Nash equilibrium）

生成式对抗网络（GAN）

Minimax Game

- D-step：

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

- G-step：

$$J^{(G)} = -J^{(D)} = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

- 生成器最小化目标
  - 判别器将**生成数据**识别为**假**的**概率**的**log**值
- 均衡点是判别器代价函数的鞍点（Saddle Point）
- 类似于Jensen-Shannon divergence

生成式对抗网络（GAN）

$$\mathbb{E}_{x\sim P_r}[\log D(x)] + \mathbb{E}_{x\sim P_g}[\log(1-D(x))]$$

$$\mathbb{E}_{x\sim P_r}\log\frac{P_r(x)}{\frac{1}{2}[P_r(x)+P_g(x)]} + \mathbb{E}_{x\sim P_g}\log\frac{P_g(x)}{\frac{1}{2}[P_r(x)+P_g(x)]} - 2\log 2$$

- 变换成这个样子是为了引入KL散度和JS散度

$$KL(P_1||P_2) = \mathbb{E}_{x\sim P_1}\log\frac{P_1}{P_2}$$

$$JS(P_1||P_2) = \frac{1}{2}KL(P_1||\frac{P_1+P_2}{2}) + \frac{1}{2}KL(P_2||\frac{P_1+P_2}{2})$$

$$2JS(P_r||P_g) - 2\log 2$$

- 在最优判别器下，GAN定义的生成器loss可等价变换为最小化真实分布与生成分布之间的JS散度。

生成式对抗网络（GAN）

训练算法

默认为1

重复多次

- 执行D-step的minibatch优化k次：
  - 从先验分布$p_g(z)$随机生成m个随机噪声向量$z^{(i)}$
  - 从数据集分布$p_{data}(x)$里随机获取m个样本$x^{(i)}$
  - 使用**梯度上升优化判别器**的代价函数：

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

- 执行G-step的minibatch优化1次：
  - 从先验分布$p_g(z)$随机生成m个随机噪声向量$z^{(i)}$
  - 使用**梯度下降优化判别器**的代价函数：

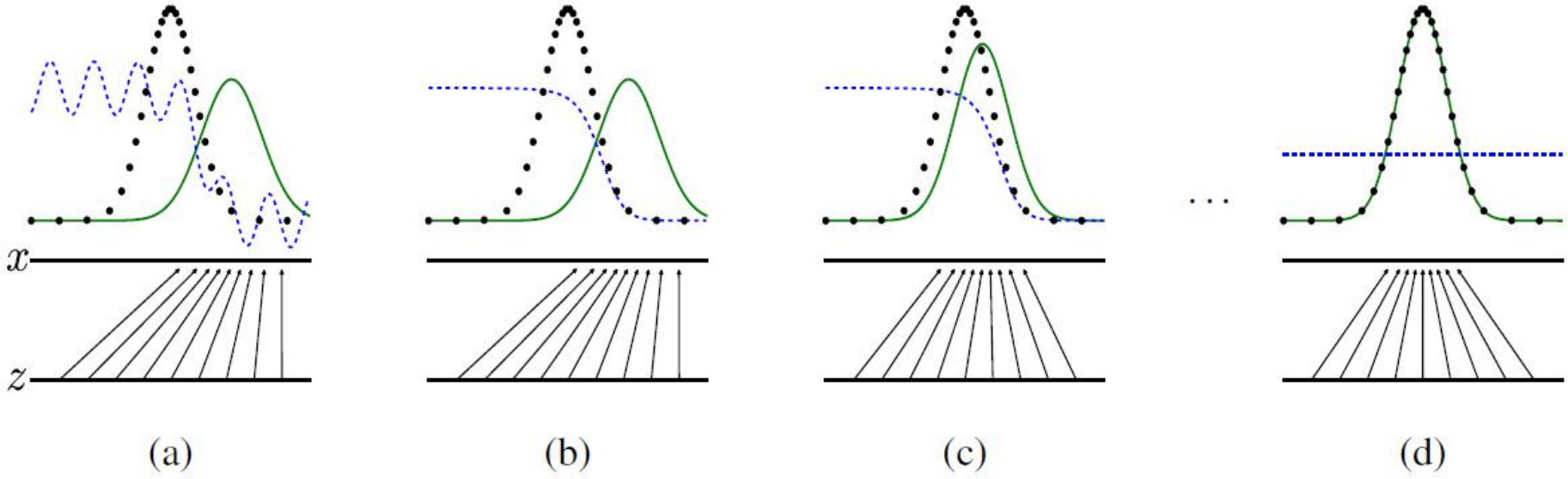
$$\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$$

生成式对抗网络（GAN）

训练过程：

- 绿线：生成器的数据分布
- 黑线：真实数据分布
- 蓝线：判别器的结果分布

随着绿色的线与黑色的线的偏移，蓝色的线下降了，也就是判别器准确率也下降了。



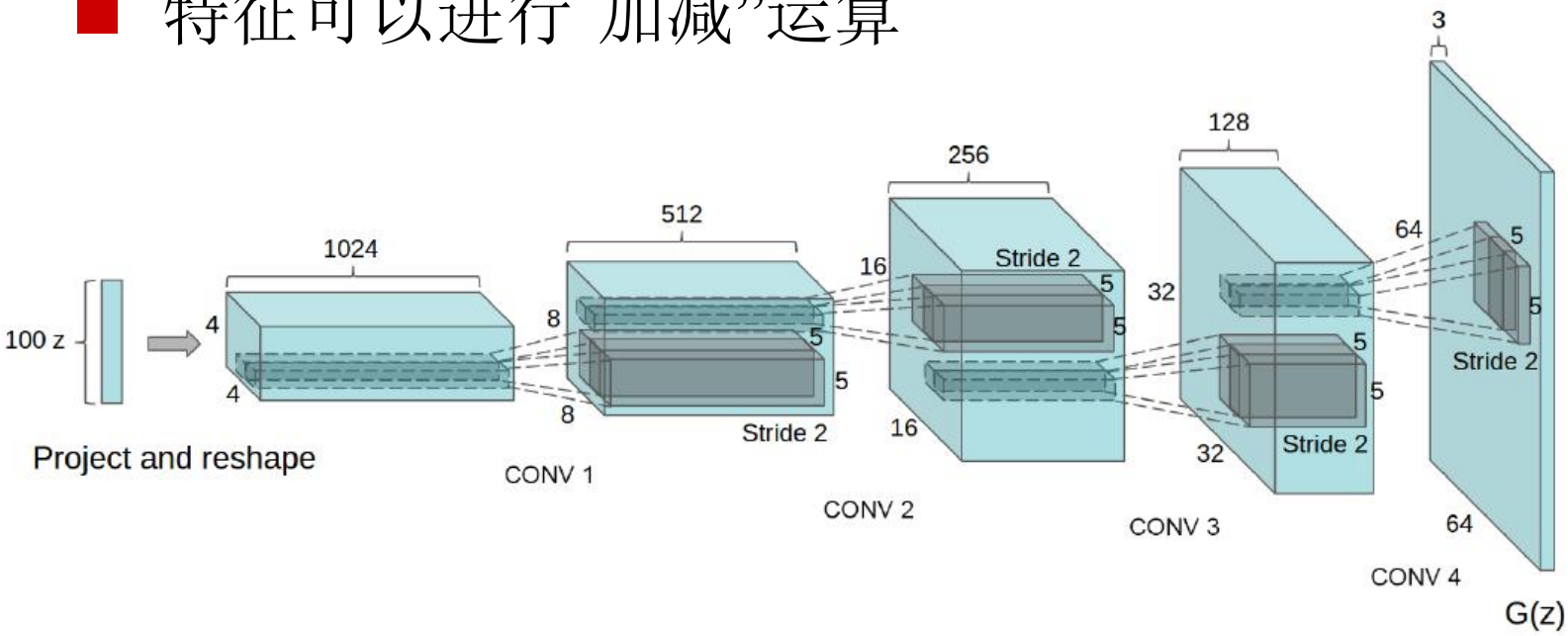
生成式对抗网络（GAN）

问题及挑战：

- 训练困难
  - 收敛问题：很难达到纳什均衡点
  - 无法有效监控收敛状态
- 模型崩溃（Model collapse）
  - 判别器快速达到最优，能力明显强于生成器
  - 生成器将数据集中生成在判别器认最认可的空间点上，即输出多样性很低
- 不适用于离散输出（不可微分）
  - 文本生成

DCGAN

- 为GAN的训练提供了一个适合的网络结构。
- 表明生成的特征具有向量的计算特性。
  - 特征可以进行“加减”运算



DCGAN

- 模型稳定训练的技巧（tricks）
  - 全连接层→卷积层
  - 池化层、上采样层→卷积层
    - 生成器：带步长的卷积(Fractional strided convolution)
  - 使用批量归一化（batch normalization）
  - 生成器中使用ReLU激活函数，输出使用Tanh
  - 判别器中使用Leaky ReLU激活函数
  - 使用adam优化器训练，学习率最好是0.0002

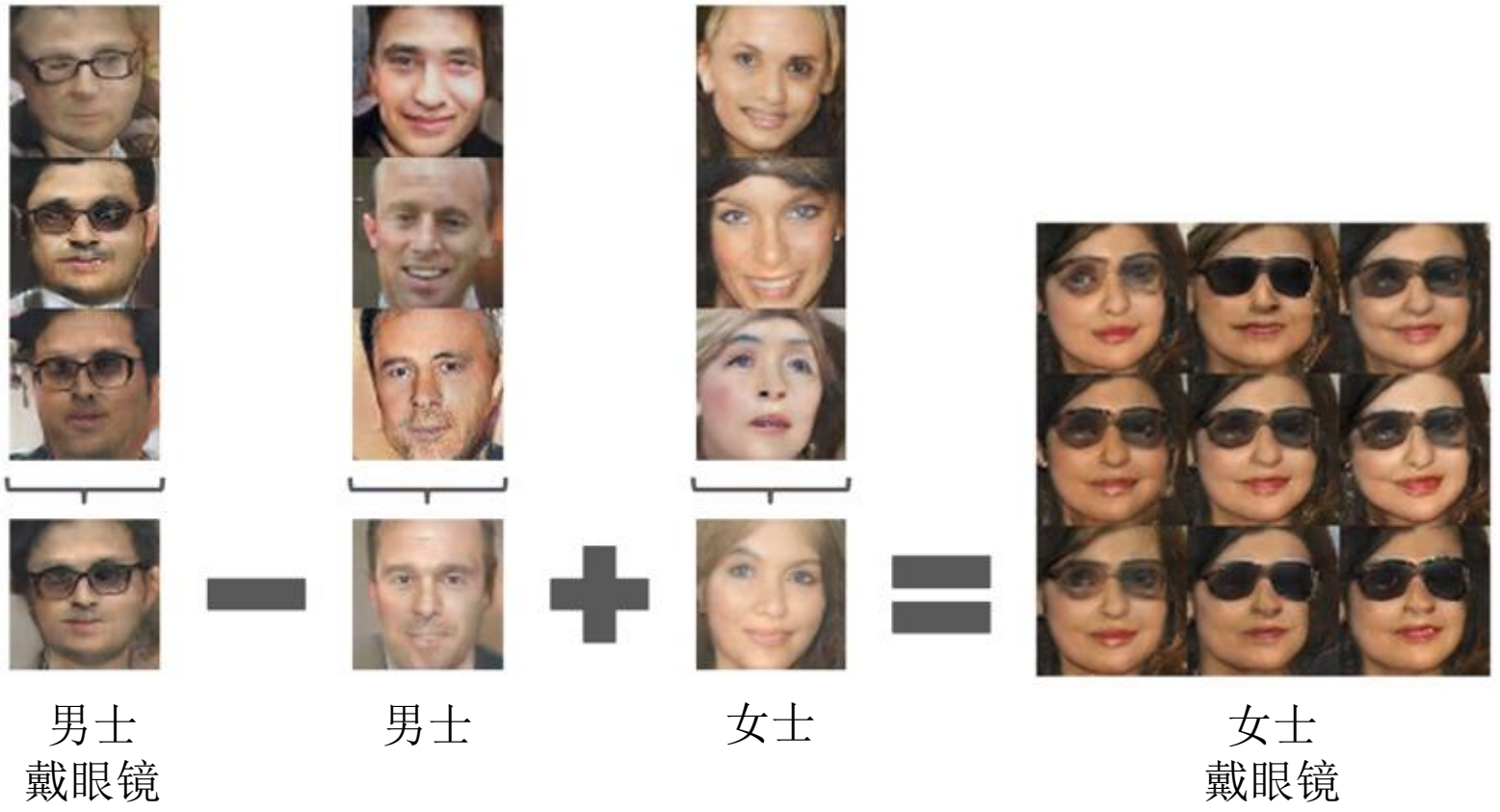
DCGAN

- 生成效果（LSUN数据集上的效果）
  - 仅支持低分辨率图片
  - 无法捕捉物体结构特性



DCGAN

- Z向量的计算特性



图片生成DCGAN

- Z向量的插值特性



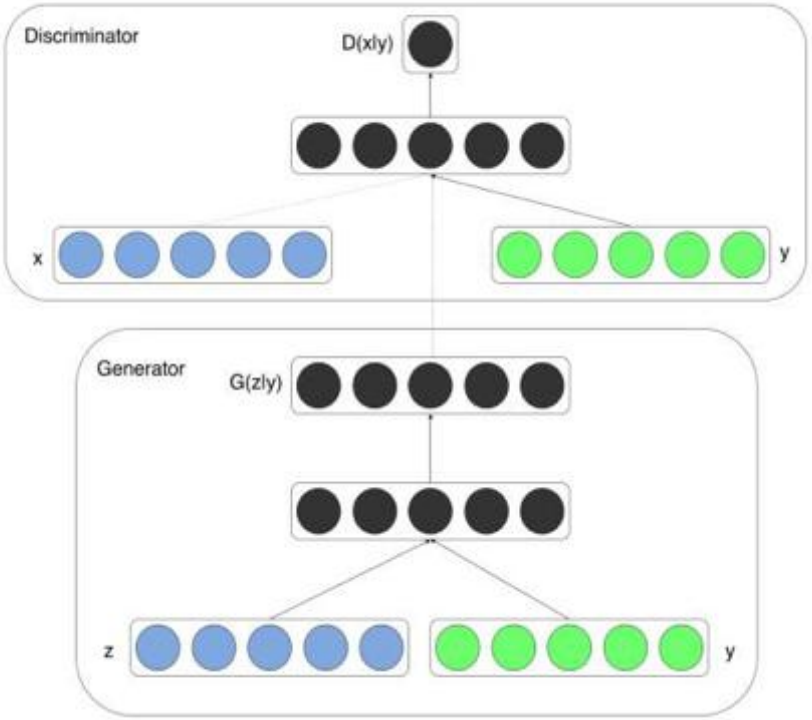
向左看　　向右看

CGAN

- GAN中输入是随机的数据，那么很自然就会想到能否用输入改成一个有意义的数据？
- 最简单的就是数字字体生成，能否输入一个数字，然后输出对应的字体。
- 这就是CGAN思路。

CGAN

- 实现方式：
- 在G网络的输入在z的基础上连接一个输入y；
- 然后在D网络的输入在x的基础上也连接一个y。



目标函数：

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

Wasserstein GAN

- WGAN相比原始GAN的算法只改了四点：
  - 判别器最后一层去掉sigmoid
  - 生成器和判别器的loss不取log
  - 每次更新判别器的参数之后把它们的绝对值截断到不超过一个固定常数c
  - 不要用基于动量的优化算法（包括momentum和Adam），推荐RMSProp，SGD也行

Wasserstein GAN

- Wasserstein GAN实现了以下关键点：
  - 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collapse mode的问题，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训练得越好，代表生成器产生的图像质量越高；
  - 以上一切好处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以做到。

Wasserstein GAN

- 只要p_r和p_g没有一点重叠或者重叠部分可忽略，JS散度就固定是常数，而这对于梯度下降方法意味着梯度为0。
- p_r和p_g不重叠或重叠部分可忽略的可能性有多大？不严谨的说是：非常大。
- 在（近似）最优判别器下，最小化生成器的loss等价于最小化p_r和p_g之间的JS散度，而由于p_r和p_g几乎肯定有可忽略的重叠，所以无论它们相距多远JS散度都是常数，最终导致生成器的梯度（近似）为0，梯度消失。

Wasserstein GAN

- Wasserstein距离又叫Earth-Mover距离：

$$W(P_r, P_g) = \inf_{\gamma \sim \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||]$$

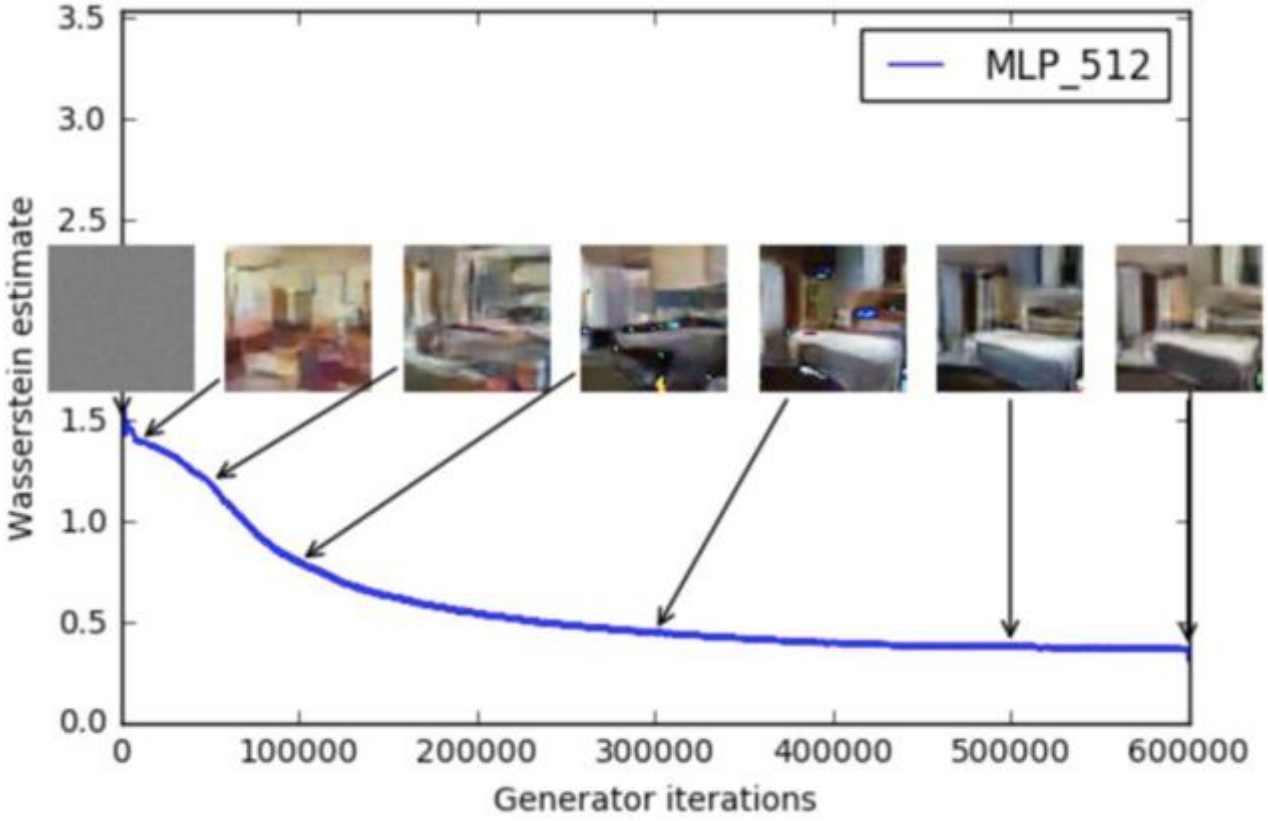
- 可理解为在这个"路径规划"下把p_r这堆"土"挪到p_g"位置"所需的"消耗"，而$W(p_r, p_g)$就是"最优路径规划"下的"最小消耗"。
- Wasserstein距离相比KL散度、JS散度的优越性在于，即便两个分布没有重叠，Wasserstein距离仍然能够反映它们的远近。

Wasserstein GAN

- 原GAN的目标函数中的log(x)不能用，因为它的导数没有上界，就不是Lipschitz连续。
  - Lipschitz连续——其实就是在一个连续函数上面额外施加了一个限制。
- 原GAN的判别器做的是二分类任务，所以最后一层是sigmoid；现在判别器做的是近似拟合Wasserstein距离，属于回归任务，所以要把最后一层的sigmoid拿掉。
- 这样得到了WGAN的两个loss：
  - WGAN生成器loss函数 $-\mathbb{E}_{x\sim P_g}[f_w(x)]$
  - WGAN判别器loss函数 $\mathbb{E}_{x\sim P_g}[f_w(x)]-\mathbb{E}_{x\sim P_r}[f_w(x)]$

Wasserstein GAN

- 判别器所近似的Wasserstein距离与生成器的生成图片质量高度相关。



Wasserstein GAN

在所有WGAN的实验中未观察到collapse mode。

- WGAN如果用类似DCGAN架构，生成图片的效果与DCGAN差不多。
- 但是厉害的地方在于：WGAN不用DCGAN各种特殊的架构设计也能做到不错的效果。



Super-ResolutionGAN

- I^{LR}是高分辨率图像I^{HR}的低分辨率副本。高分辨率图像全部来自于数据库。I^{LR}是对I^{HR}进行高斯滤波然后进行下采样得到的。
- SRGAN的博弈公式：

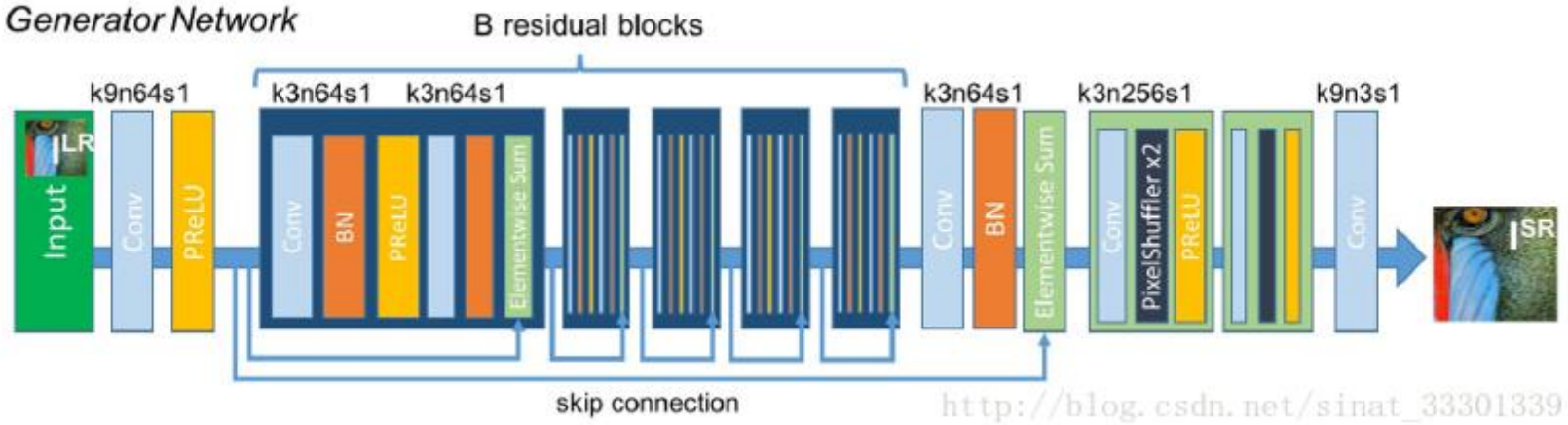
$$\min_{\theta_G}\max_{\theta_D}\ \mathbb{E}_{I^{HR}\sim p_{\text{train}}(I^{HR})}[\log D_{\theta_D}(I^{HR})]+$$
$$\mathbb{E}_{I^{LR}\sim p_G(I^{LR})}[\log(1-D_{\theta_D}(G_{\theta_G}(I^{LR})))]$$

- 参数θ_G通过优化一个特殊的损失函数l^{SR}得到的，即：

$$\hat{\theta}_G=\arg\min_{\theta_G}\frac{1}{N}\sum_{n=1}^{N}l^{SR}(G_{\theta_G}(I_n^{LR}),I_n^{HR})$$

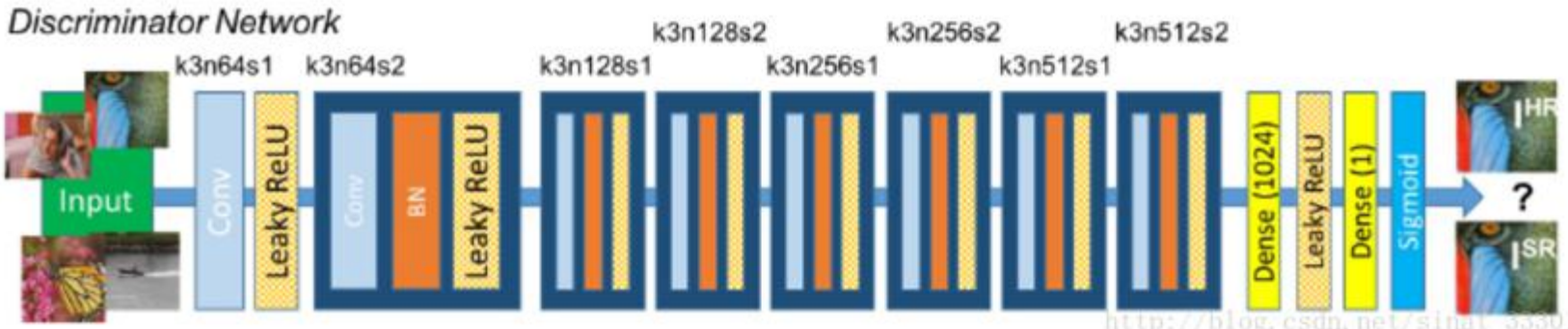
Super-ResolutionGAN

- 生成网络：应用了分布相同的B残差块，每个残差块都有两个卷积层。
- 卷积层后面加上batch-normalization，并用PReLU作为激活函数。
- 卷积层的卷积核都是3×3，并有64个特征图。
- 还有跃层连接。



Super-ResolutionGAN

- 判别网络：由连续的卷积块构成，卷积块包括卷积层、Leaky ReLU层和BN层。
- 卷积层的卷积核都是3×3。
- 最后是两个dense层，并通过sigmoid进行鉴别判断。



Super-ResolutionGAN

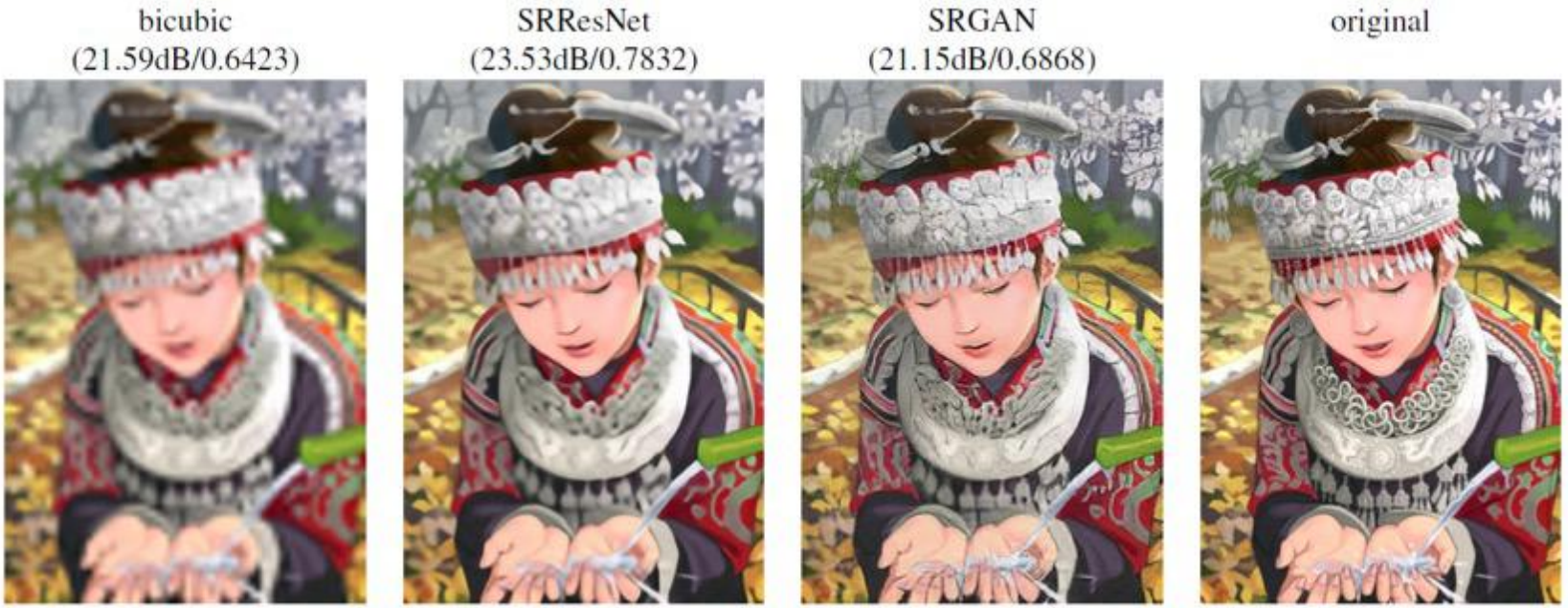
模型使用：

- 价值函数 $$\min_{\theta_G}\max_{\theta_D}\ \mathbb{E}_{I^{HR}\sim p_{\text{train}}(I^{HR})}[\log D_{\theta_D}(I^{HR})]+ \mathbb{E}_{I^{LR}\sim p_G(I^{LR})}[\log(1-D_{\theta_D}(G_{\theta_G}(I^{LR}))]$$
- 生成器的感知代价函数 $$l^{SR} = \underbrace{\underbrace{l^{SR}_{\text{X}}}_{\text{content loss}} + \underbrace{10^{-3}l^{SR}_{Gen}}_{\text{adversarial loss}}}_{\text{perceptual loss (for VGG based content losses)}}$$
  - 内容代价函数（Content loss – VGG loss）
$$l^{SR}_{VGG/i.j} = \frac{1}{W_{i,j}H_{i,j}}\sum_{x=1}^{W_{i,j}}\sum_{y=1}^{H_{i,j}}(\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$
  - 对抗代价函数（Adversarial loss）
$$l^{SR}_{Gen} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

Super-ResolutionGAN

效果展示

- 捕捉人类视觉感知代价(human-like perceptual loss)



参考资料

- 李宏毅，深度学习
- https://www.sohu.com/a/162863895_697750
- https://www.cnblogs.com/frombeijingwithlove/p/7077448.html
- https://www.camcard.com/info/1599920cdf149bc24c95739e9
- https://zhuanlan.zhihu.com/p/25071913

问答互动

在所报课的课程页面，

1、点击“全部问题”显示本课程所有学员提问的问题。

2、点击“提问”即可向该课程的老师和助教提问问题。



联系我们

小象学院：互联网新技术在线教育领航者

– 微信公众号：**小象学院**

