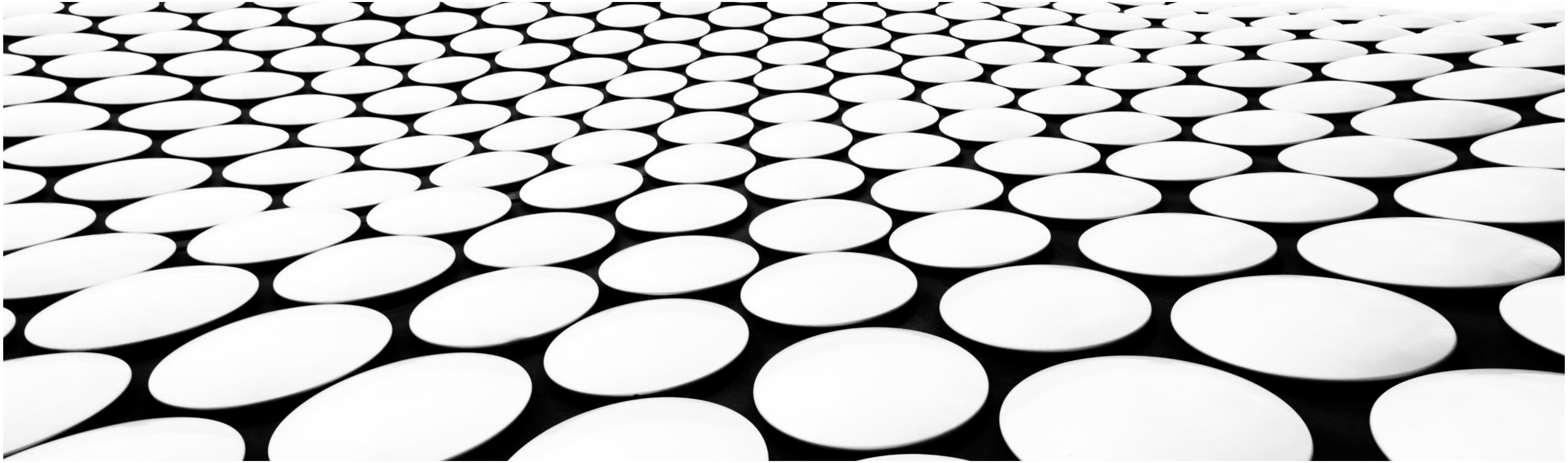
深度学习

邱怡轩



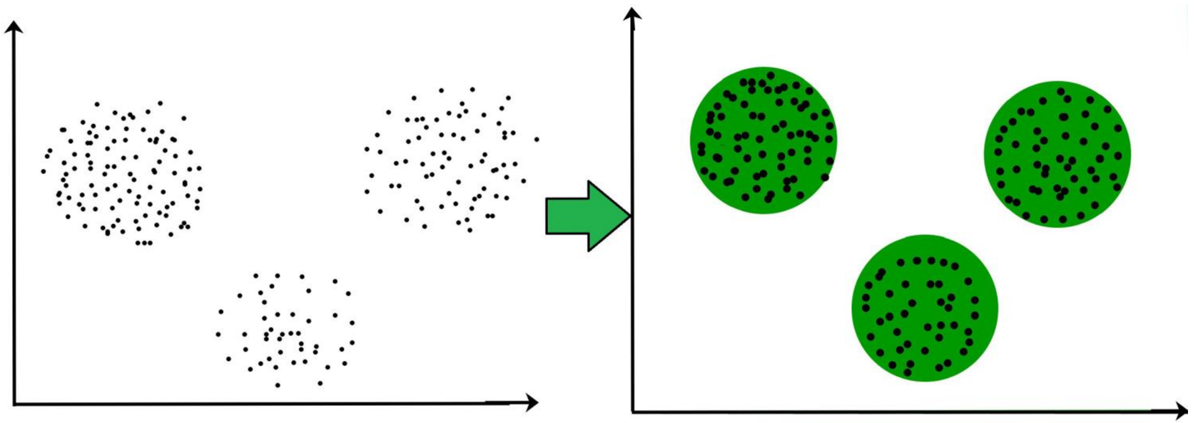
今天的主题

- 深度生成模型
- 变分自编码器　扩散模型
- 生成对抗网络

无监督学习

- 数据：X，没有标记“标签”
- 目标：了解数据的分布或结构

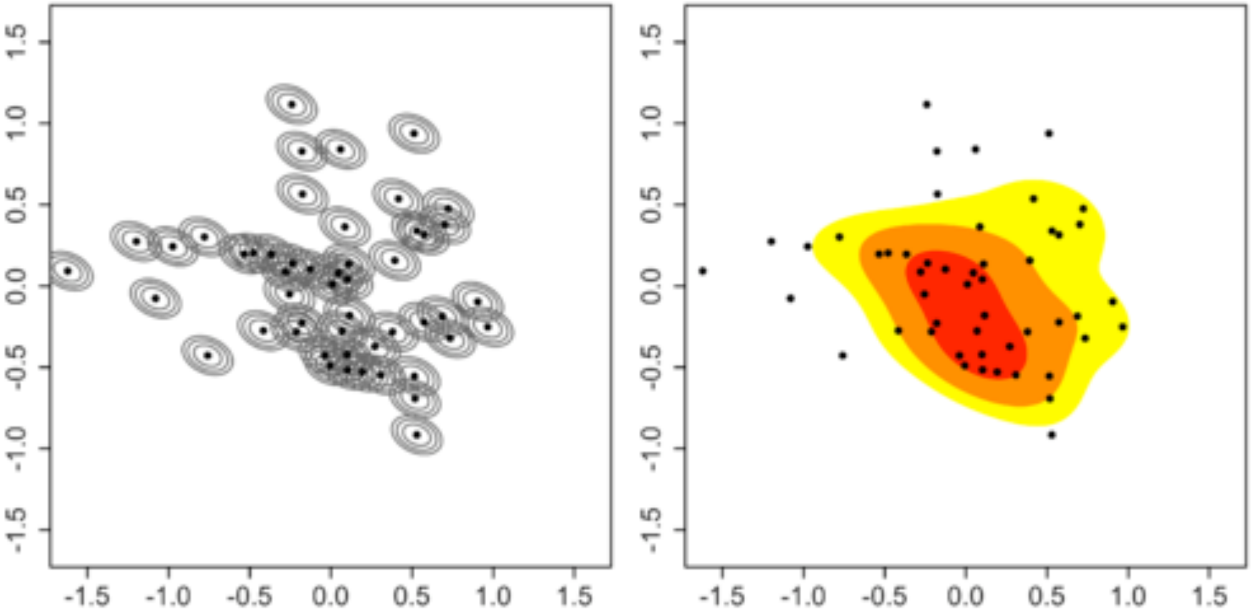
聚类



无监督学习

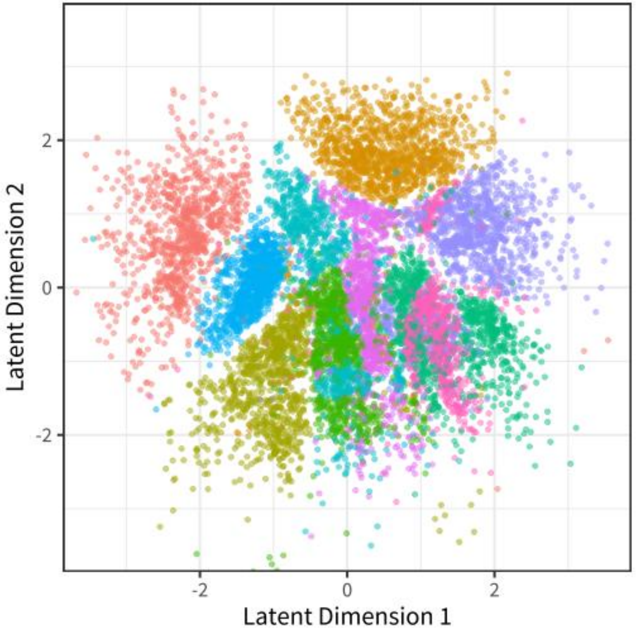
- 数据：X，没有标记“标签”
- 目标：了解数据的分布或结构

密度估计



无监督学习

- 数据：X，没有标记“标签”
- 目标：了解数据的分布或结构

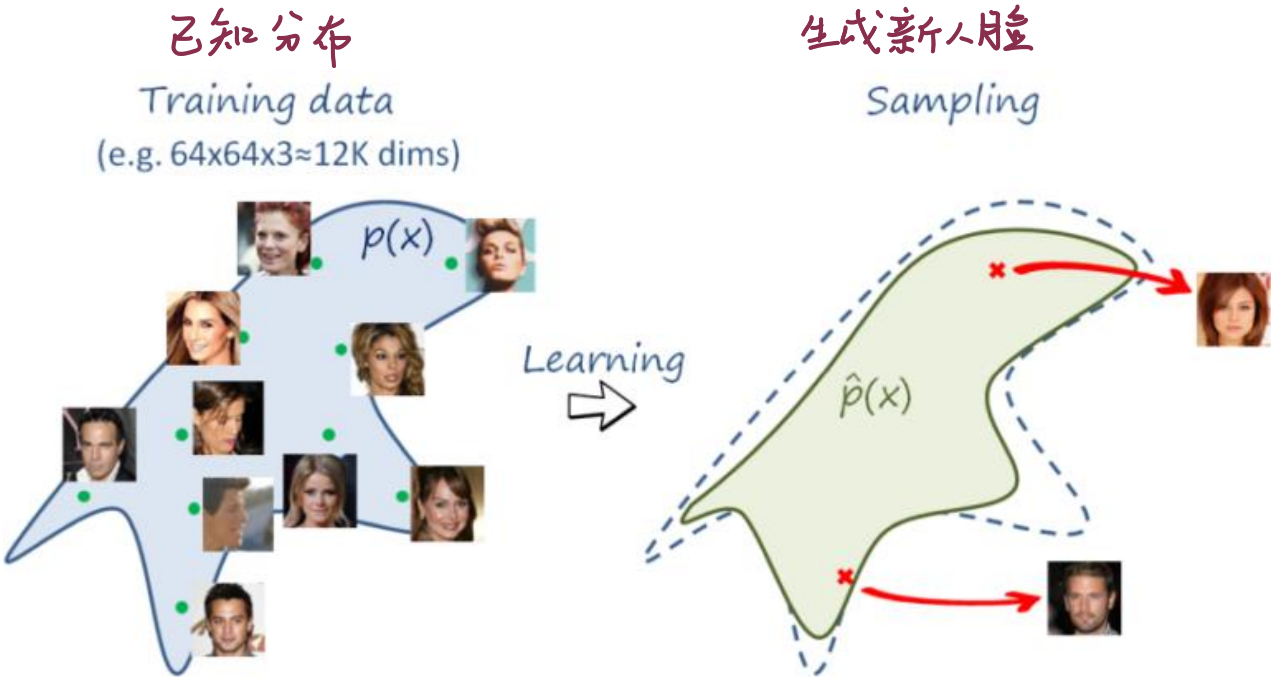


深度生成模型

无监督学习

生成模型

- 给定样本数据，得到一个“生成器”，用以生成同分布的样本



生成模型

- 经典统计中的密度估计就是一种生成模型
- 利用了神经网络结构的生成模型一般称为深度生成模型

- 变分自编码器（variational autoencoder)
- 生成对抗网络（generative adversarial network) GAN
- 流模型（normalizing flow)
- 扩散模型（diffusion model) 很多
- ……

作用

- 理解数据的统计分布（统计建模的核心问题）
- 数据扩充，图像生成，视觉艺术
- 统计模拟　　生成新的样本
- 提取特征　　内在潜变量的表达
　　　　　　　类似 PCA
　　　　　　　非线性
- ……

变分自编码器

（从统计学的视角）

VAE

- Variational autoencoders, VAE

  变分 自编码器

- VAE 的文献通常是以编码器/解码器的视角来介绍该模型
- 我们从更“统计”的角度来引入和理解

建模目标

本质：对数据分布进行建模

- 回顾生成模型的建模目标
- 给定数据 $X_1, \ldots, X_n$，希望刻画其分布 $p(x)$
- 如何构建 $p(x)$，使其能拟合复杂的数据？

传统方法

- 假设 $p(x)$ 来自某个分布族 $p_\theta(x)$
- 例如正态分布 $N(\mu, \Sigma)$，$\theta = (\mu, \Sigma)$

- 缺点：形式受限，不够灵活

核心思想1

- 混合分布建模
- 将 $p_\theta(x)$ 设定为两个分布的混合
- $p_\theta(x) = \int \pi(z) p_\theta(x|z) \mathrm{d}z$

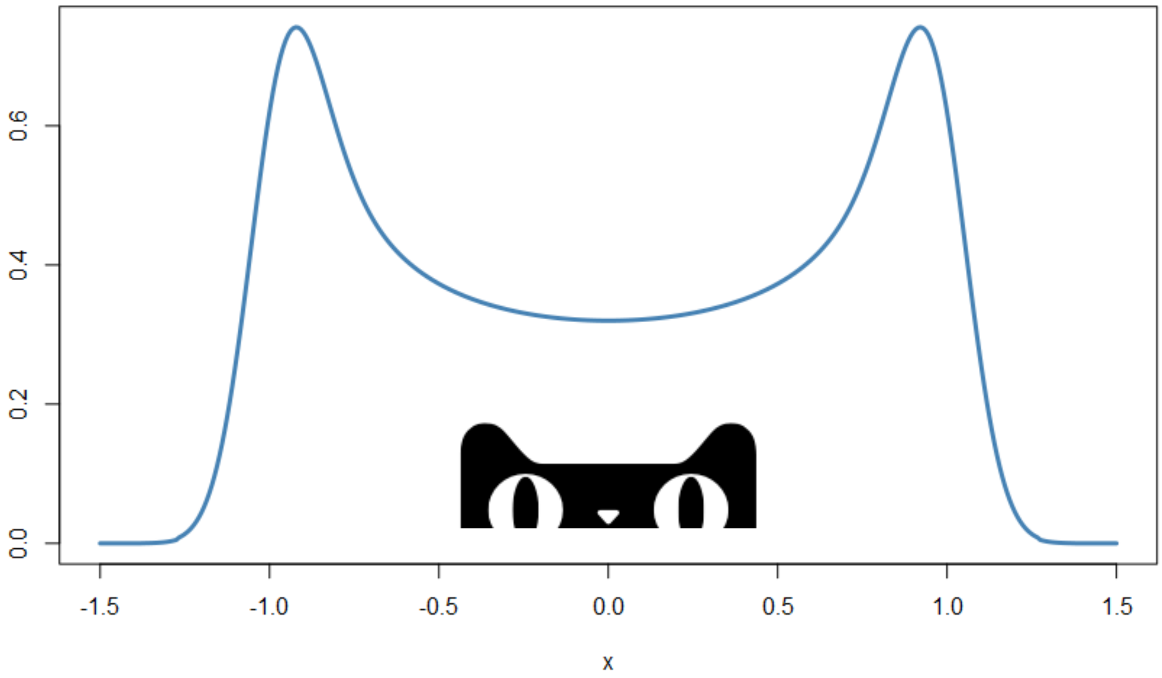
x与z联合，再消掉z

- 简单分布 + 简单分布 => 复杂分布
- 简单分布 + 复杂分布 => 更复杂的分布

混合分布

- $Z \sim N(0,1)$ 独立正态
- $X|\{Z = z\} \sim N(\sin(z), 0.01)$

条件正态分布



VAE 建模

- $p_\theta(x) = \int \pi(z) p_\theta(x|z) \mathrm{d}z$
- 在 VAE 中，$\pi(z)$ 固定为标准正态 $N(0, I)$ 简单
- $p_\theta(x|z)$ 利用神经网络刻画
- 连续数据： $p_\theta(x|z) = N(g_\theta(z), \tau^2 I)$， $g_\theta(z)$ 为神经网络，τ 为常数/超参数
- 0-1数据： $p_\theta(x|z) = Bernoulli(\sigma(g_\theta(z)))$ ，$g_\theta(z)$ 为神经网络，σ 为 Sigmoid 函数

参数估计

- 设定好模型后，接下来的工作是估计参数 θ
- 经典方法：极大似然估计
- $l(\theta; x) = \log p_\theta(x)$
- $\max_\theta \; n^{-1} \sum_{i=1}^{n} l(\theta; X_i)$

问题

复杂模型，简单计算

- $p_\theta(x) = \int \pi(z) p_\theta(x|z) \mathrm{d}z$ 是一个复杂的积分
- $l(\theta; x)$ 难以计算!

核心思想2

- 似然函数不等式
- 《关于VAE，记住这个公式就够了》
- 反正大概率记不住

一定是非负的(可以严格证明)

- $$\log p(x) - \mathrm{KL}(q(z|x)\|p(z|x)) = \mathrm{E}_{z\sim q(z|x)}[\log p(x|z)] - \mathrm{KL}(q(z|x)\|\pi(z))$$

细节

- $\mathrm{KL}(q\|p)$ 称为 KL divergence，用以衡量两个分布之间的不匹配程度
- $\mathrm{KL}(q\|p)=\int q(z)\log\frac{q(z)}{p(z)}\mathrm{d}z$
- KL divergence 非负，$\mathrm{KL}(q\|p)\geq 0$
- 当 $q=p$ 时，$\mathrm{KL}(q\|p)=0$

细节

好计算

$z \sim p(z)$
$x|z \sim p(x|z)$ $\Big\} \Rightarrow p(x,z) = p(x|z)p(z)$

$= p(z|x)p(x)$

q有极高的自由度

- $\log p(x) - \mathrm{KL}(q(z|x)||p(z|x)) = \mathrm{E}_{z\sim q(z|x)}[\log p(x|z)] - \mathrm{KL}(q(z|x)||\pi(z))$
- 该等式对于任意的分布 $q(z|x)$ 都成立

ELBO

似然函数最大值
提高下界

- 因为 $\mathrm{KL}(q(z|x)\|p(z|x)) \geq 0$，得到似然函数的一个下界
- $\log p(x) \geq \mathrm{E}_{z\sim q(z|x)}[\log p(x|z)] - \mathrm{KL}(q(z|x)\|\pi(z))$ ✓ 这个式子可计算
- 不等号右侧称为 evidence lower bound （ELBO）

ELBO

- 如果 $q(z|x)$ 选取得当，可以计算得到 ELBO 的无偏估计
- 当似然函数难以优化时，就优化似然函数的一个下界

依赖 q 、下界与似然函数接近

核心思想3

怎么找q

- "最优" 编码器
- 在 VAE 中，$q(z|x)$ 被称为编码器（encoder）
- 如果 $q(z|x) = p(z|x)$，那么似然函数=ELBO
- $q(z|x)$ 越接近 $p(z|x)$，ELBO 的近似效果越好

编码器

- 然而编码器的形式决定了 ELBO 计算的复杂程度
- 需要在近似精度与计算效率之间进行平衡
- 在 VAE 中，通常选取 $q_\phi(z|x) = N\left(\mu_\phi(x), \mathrm{diag}\{\sigma_\phi^2(x)\}\right)$
- μ_ϕ 和 σ_ϕ^2 是两个神经网络，ϕ 是可学习的参数
- $\phi^* = \mathrm{argmin}_\phi\ \mathrm{KL}(q_\phi(z|x) \| p(z|x))$

优化方法

理论上先φ后θ

- 综合起来，要优化生成网络 $p_\theta(x|z)$ 的参数 θ 和编码器 $q_\phi(z|x)$ 的参数 ϕ
- $(\theta^*, \phi^*) = \operatorname{argmax}_{\theta,\phi} ELBO$ 相互促进
- $\mathrm{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - \mathrm{KL}(q_\phi(z|x) \| \pi(z))$
- 第一项用 Monte Carlo 方法近似，第二项有显式解

交互演示

图片 → z控制

z→ 10维潜变量 → 不同抽象特征

z → x

- https://www.siarez.com/projects/variational-autoencoder
- https://spinthil.github.io/towards-an-interpretable-latent-space

类似图片分布较类似

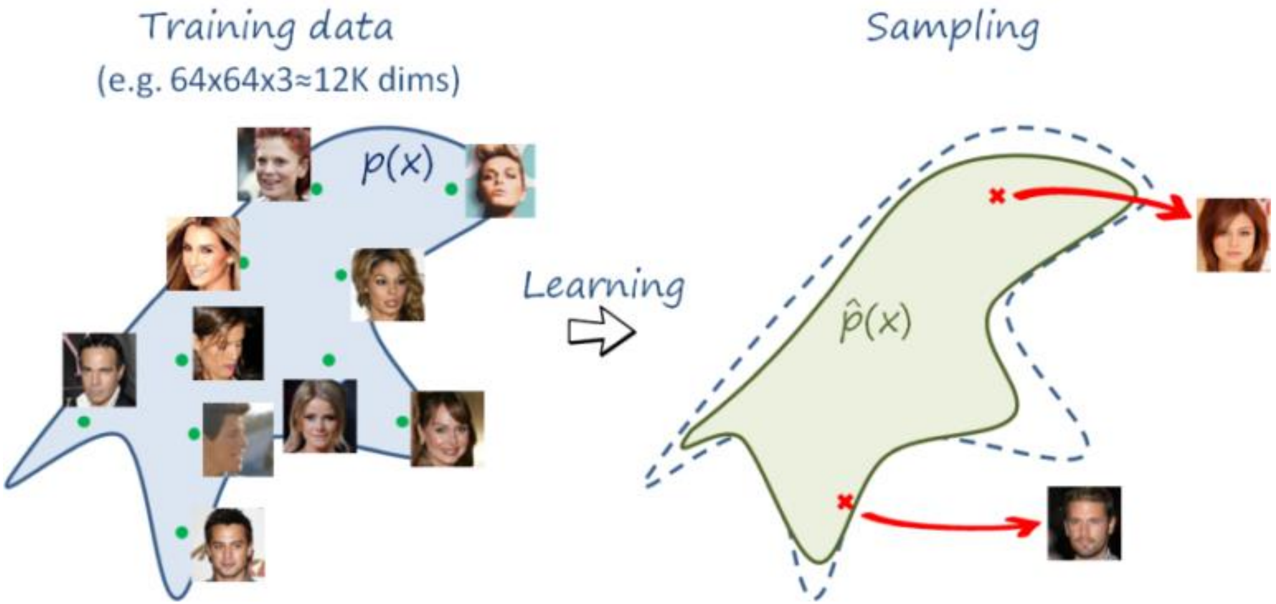
推荐阅读

- Doersch, C. (2016). Tutorial on variational autoencoders.
- https://arxiv.org/pdf/1606.05908.pdf

回顾：深度生成模型

生成模型

- 给定样本数据，得到一个“生成器”，用以生成同分布的样本



“生成器”

- 生成器可以有各种不同的形式

- 显式、可以计算的密度函数
- 显式、近似的密度函数
- 隐式分布

Taxonomy of Generative Models

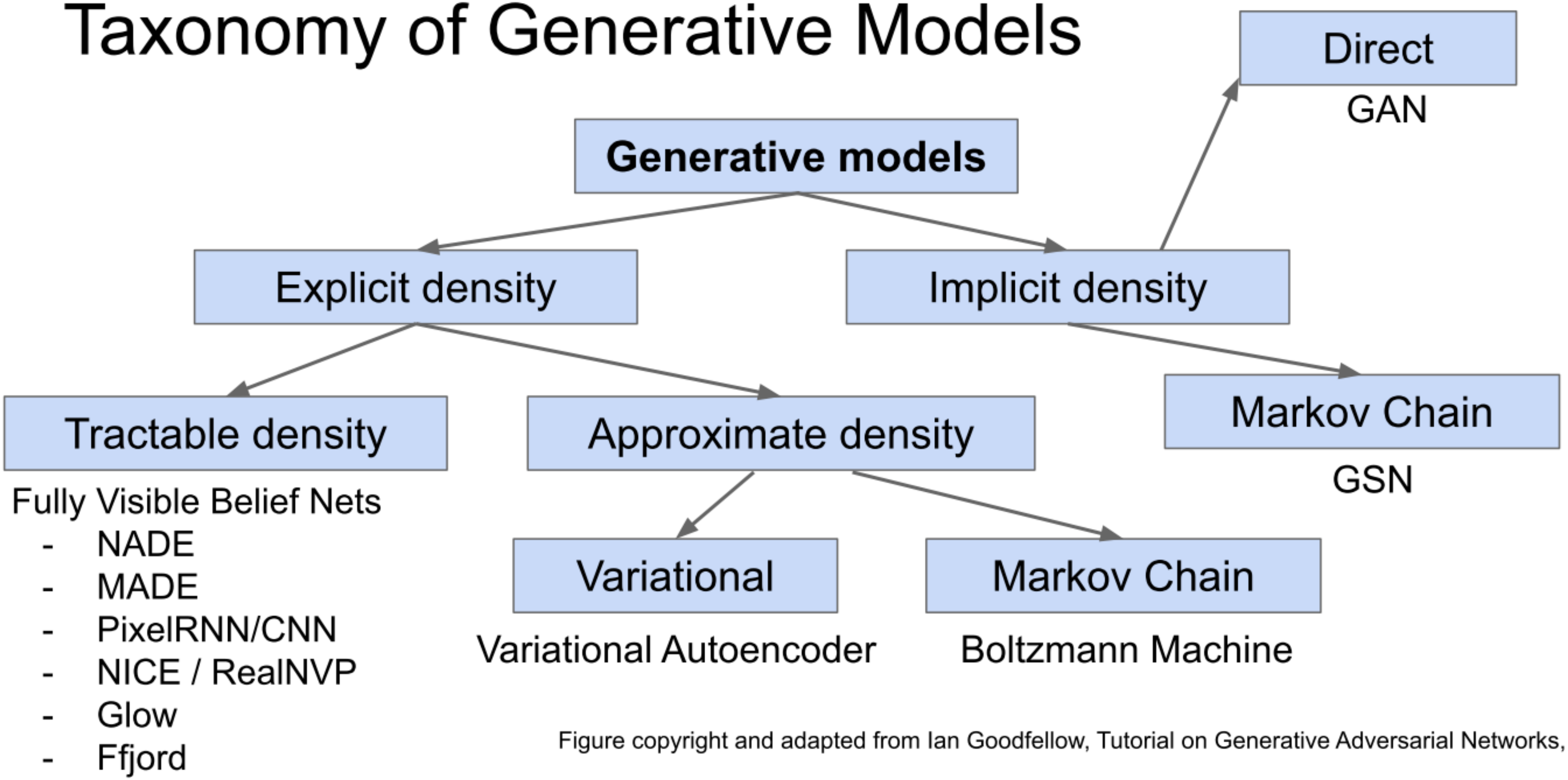


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Taxonomy of Generative Models

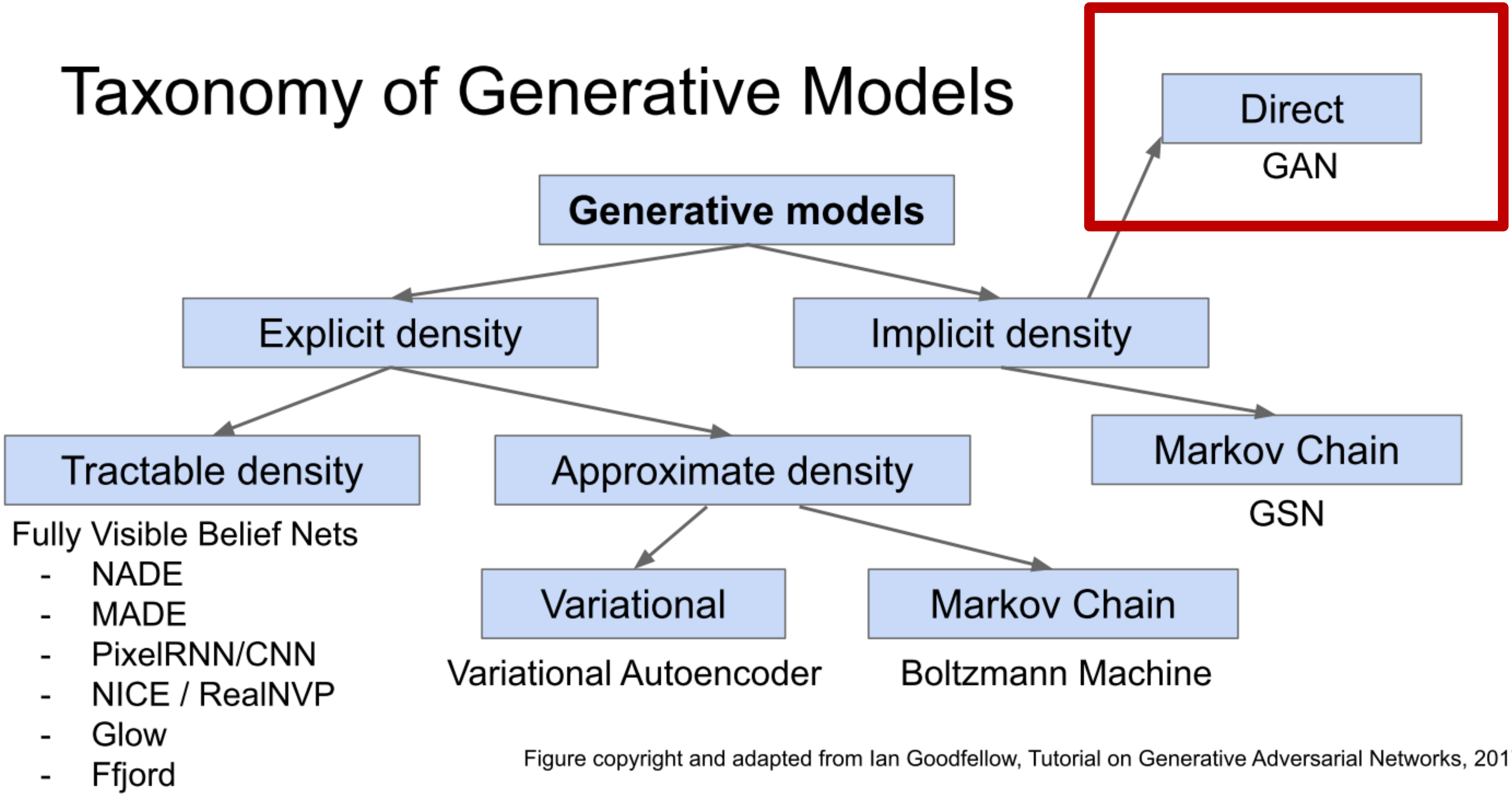


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

GAN

- Generative adversarial network, GAN

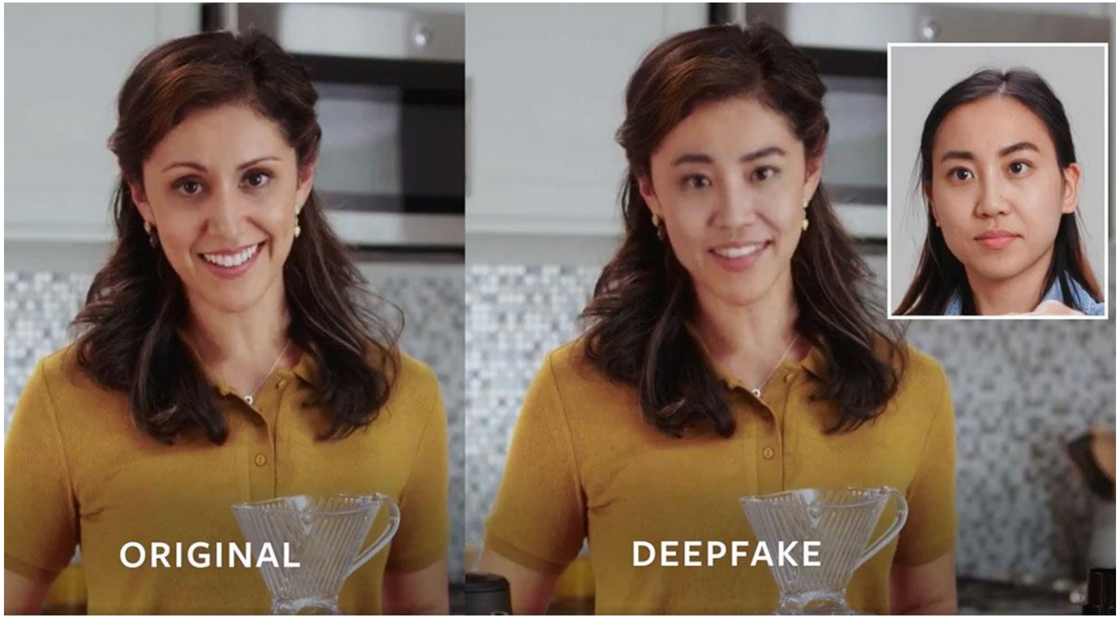
- 没有显式的密度函数
- 但可以进行无限的抽样
- 被广泛应用于图像生成

GAN

StyleGan: https://github.com/NVlabs/stylegan
https://www.bilibili.com/video/BV1rb41187Wv

AI 伦理

- DeepFake



- 技术是否可以被无限使用甚至滥用?
- 当图片和影像不再完全可信的时候，如何合理接收和辨别信息?

AI 伦理

- “不懂得进攻的方法，就无从防御。”

GAN 模型原理

核心思想1

- 随机变量的变换

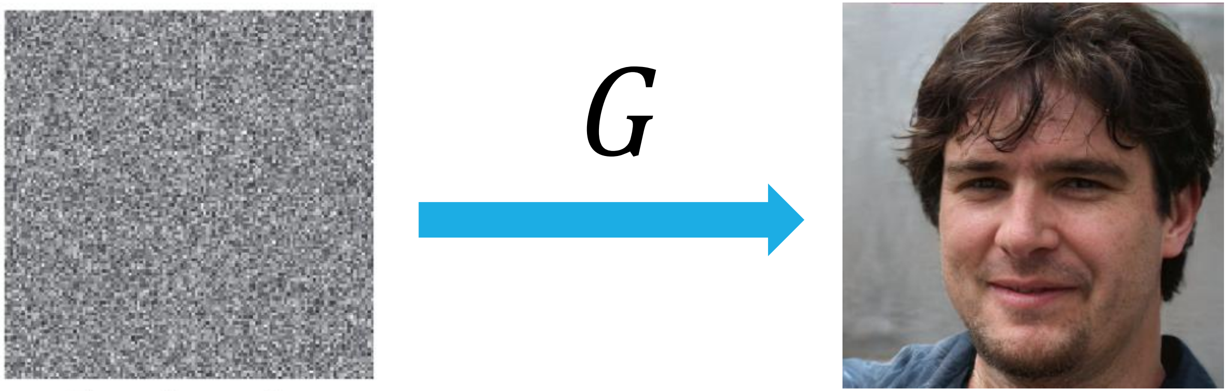
- GAN 的生成器基于一个基本的原理
- 将简单的随机变量进行复杂的非线性变换，可以得到复杂的分布

分布变换

- $U \sim Unif(0,1)$， $-\log U \sim ?$
- $Z_1, Z_2 \sim^{iid} N(0,1)$， $Z_1^2 + Z_2^2 \sim ?$
- $U_1, U_2 \sim^{iid} Unif(0,1)$，
  $\sqrt{-2\log U_1}\cos(2\pi U_2) \sim ?$

GAN生成器

- GAN 试图估计一个变换映射 $G: \mathrm{R}^r \rightarrow \mathrm{R}^p$
- 当输入一个“噪音”随机向量 $Z \sim N(0, I_r)$
- 可以得到一张“有意义”的图片 $X = G(Z)$



Gaussian noise

统计含义

- 如果 $Z \sim N(0, I_r)$
- 那么 $G(Z)$ 也是一个随机向量，记其分布为 p_g
- 给定样本 $X_1, \dots, X_n \sim p^*$，估计映射 G，使得 $p_g \approx p^*$
- 如何衡量 p_g 与 p^* 之间的差距?

极大似然

- 传统方法中，衡量模型分布 p_θ 与数据分布 p^* 的差距常用 KL divergence
- $\mathrm{KL}(p^* \| p_\theta) = \mathrm{E}_{p^*} \log p^*(x) - \mathrm{E}_{p^*} \log p_\theta(x)$
- $\min \ \mathrm{KL}(p^* \| p_\theta) \Leftrightarrow \max \ \mathrm{E}_{p^*} \log p_\theta(x)$
- 也就是极大似然！
- 然而这需要 p_θ 有显式的表达式

核心思想2

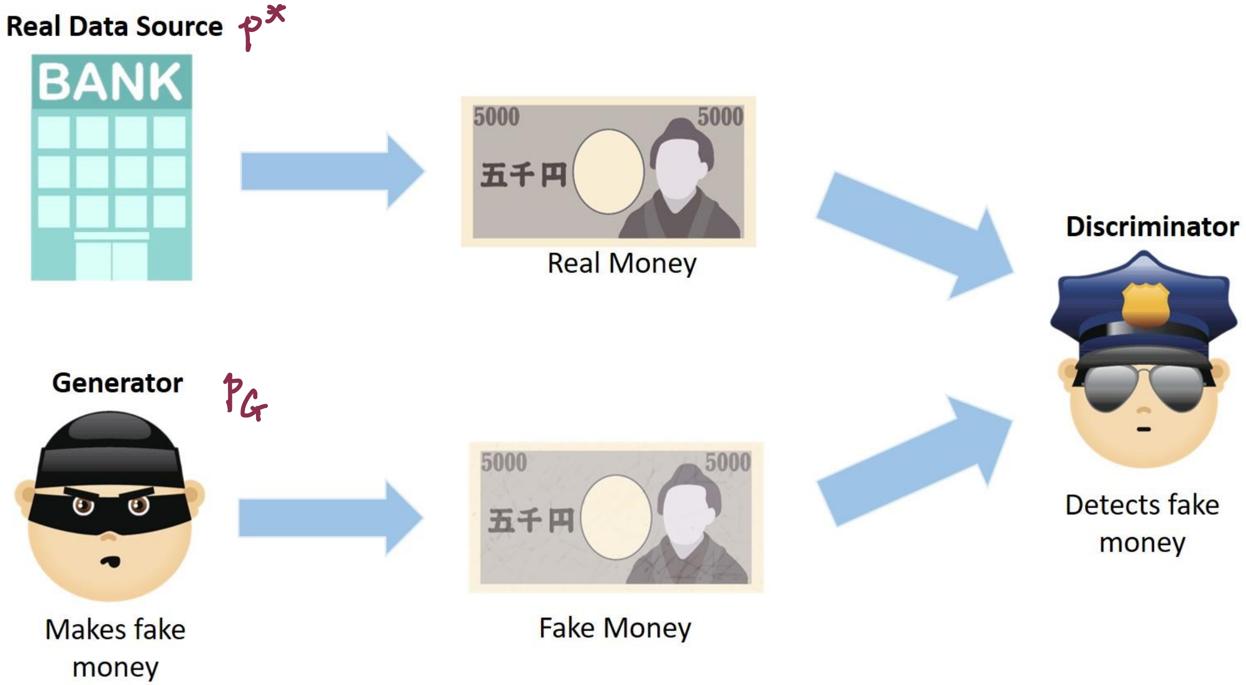
- 判别器

- 在 GAN 中，p_g 没有显式的密度函数
- 但创造性地引入了判别器的思想
- 如果能同时得到 p_g 和 p^* 的样本，并且很难用一个分类器去区分它们，那么就可以认为 $p_g \approx p^*$

"警察与假币"

判别器

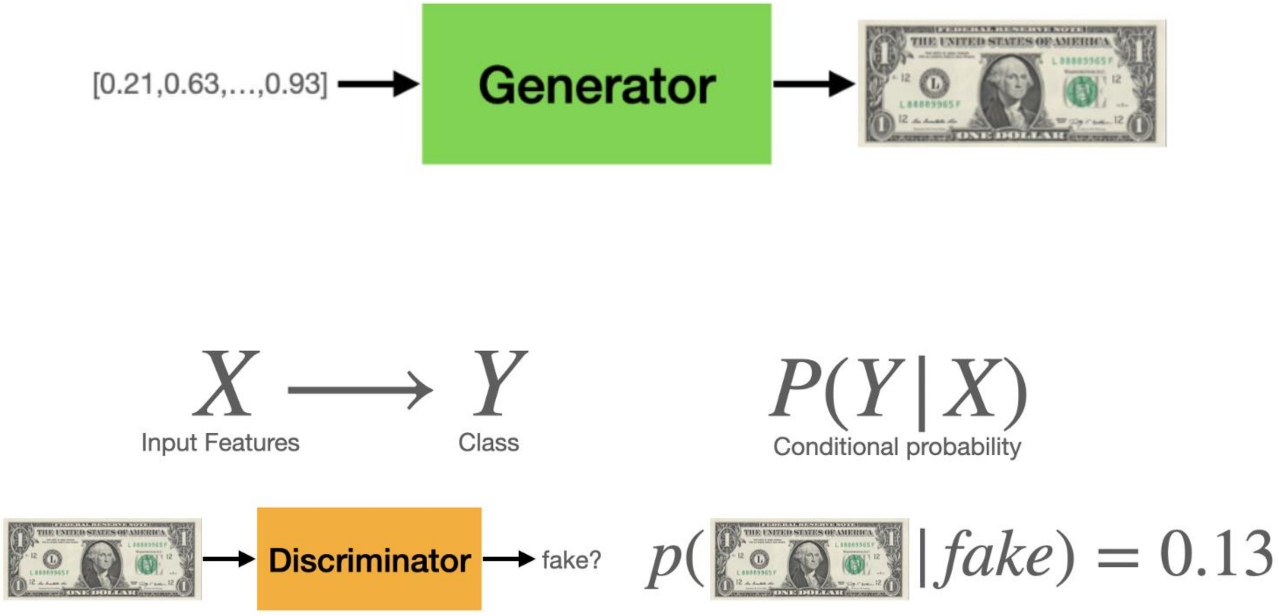
- GAN 的论文中引入了一个“警察与假币”的比喻



图片来源：https://www.macnica.co.jp/business/ai_iot/columns/135130/

判别器

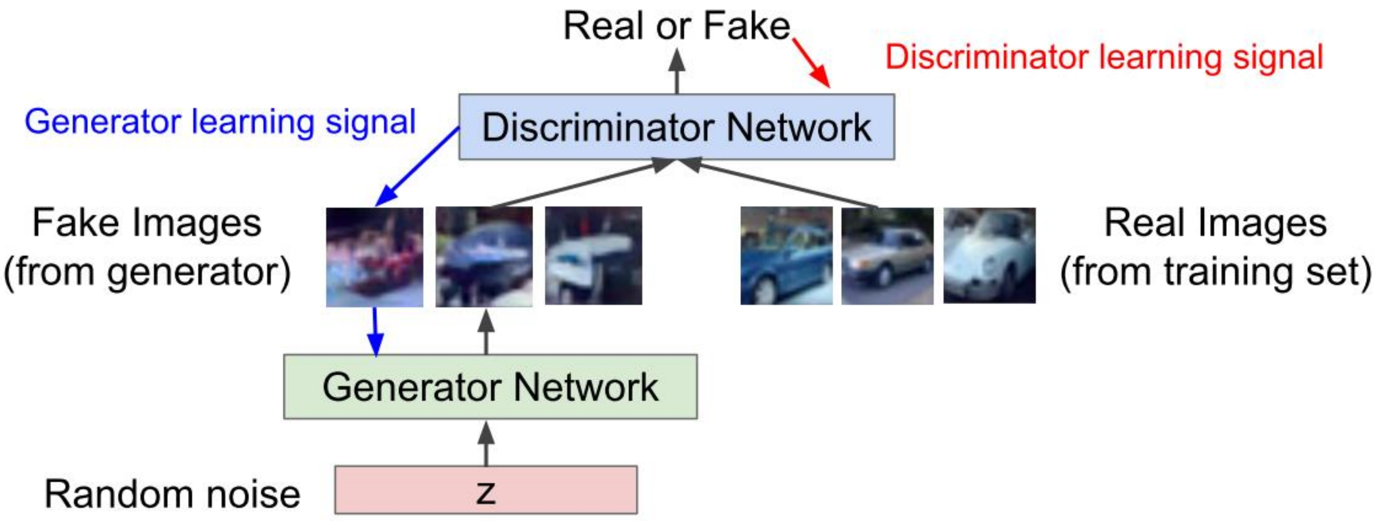
- GAN 的论文中引入了一个“警察与假币”的比喻



图片来源：https://towardsdatascience.com/intuition-behind-generative-adversarial-networks-52628d3119f5

判别器

- GAN 定义了一个判别器 D，用于区分真实的数据和生成的样本



Fake and real images copyright Emily Denton et al. 2015.

判别器

- 最优判别器
- $\max_{D} \mathrm{E}_{x\sim p^*} \log D(x) + \mathrm{E}_{z\sim p(z)} \log[1 - D(G(z))]$

核心思想3

- 对抗

- GAN 同时估计生成器 G 和判别器 D
- D 的目的是最大区分生成样本和真实数据
- G 的目的是最小化生成样本和真实数据的差异

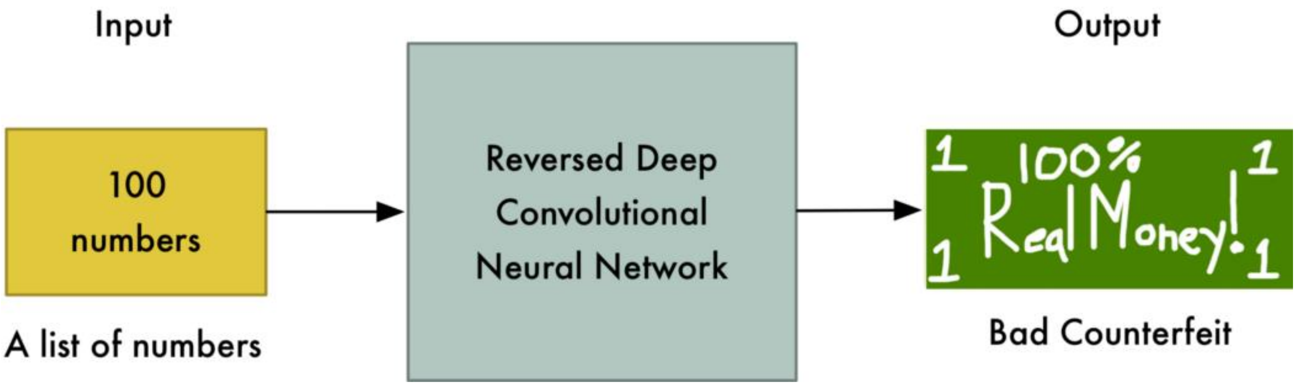
- 两个网络相互对抗，不断调整参数，最终目的是使判别器无法判断生成的样本是否是真的

对抗

- 回到“警察与假币”的比喻
- 警察不断提高鉴别能力
- 假币制造者不断提高造假技术
- 最终的结果是真币与假币无法区分开来

对抗

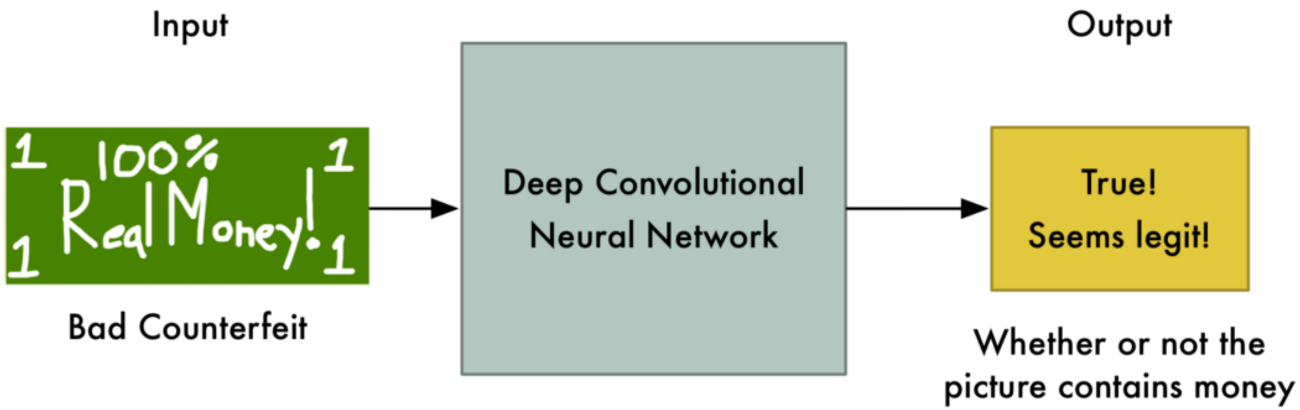
- 一个很差的生成器



图片来源：https://towardsdatascience.com/intuition-behind-generative-adversarial-networks-52628d3119f5

对抗

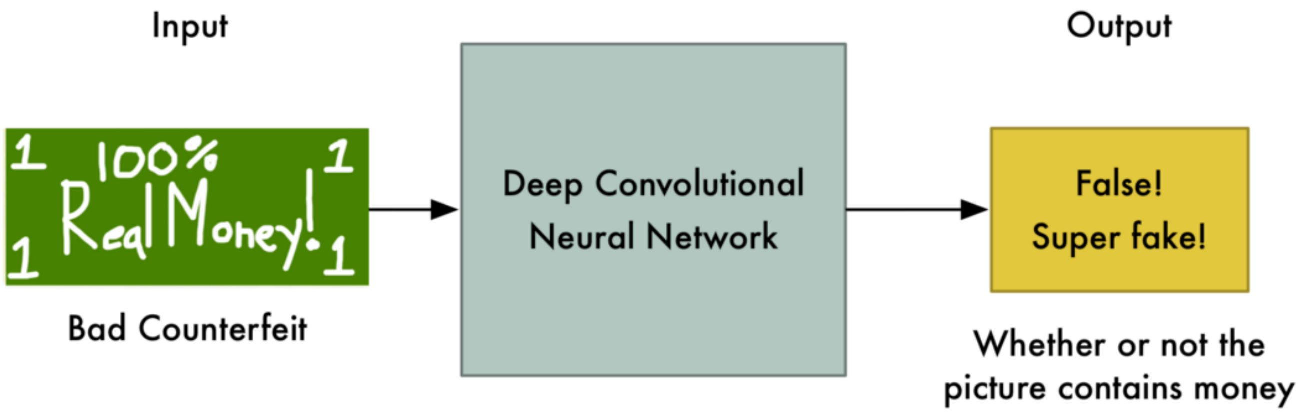
- 一个很差的判别器



图片来源：https://towardsdatascience.com/intuition-behind-generative-adversarial-networks-52628d3119f5

对抗

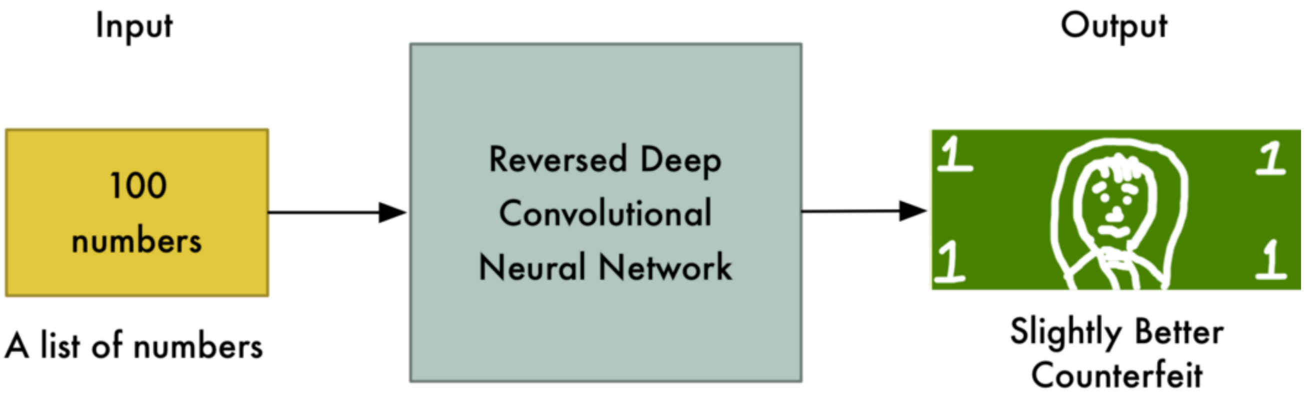
▪ 改进后的判别器



图片来源：https://towardsdatascience.com/intuition-behind-generative-adversarial-networks-52628d3119f5

对抗

改进后的生成器



图片来源：https://towardsdatascience.com/intuition-behind-generative-adversarial-networks-52628d3119f5

对抗

- Minimax 优化问题
- $\min_{G} \max_{D} \mathrm{E}_{x\sim p^*} \log D(x) + \mathrm{E}_{z\sim p(z)} \log[1 - D(G(z))]$

判

优化

反复调参、训练过程人工干预

- 实际操作中，GAN 的优化是一大难题
- 牵涉到生成器与判别器的博弈
- 不如 VAE 稳定
- 伴有梯度消失的问题

深入理解

GAN统计理解

似然比与分类器

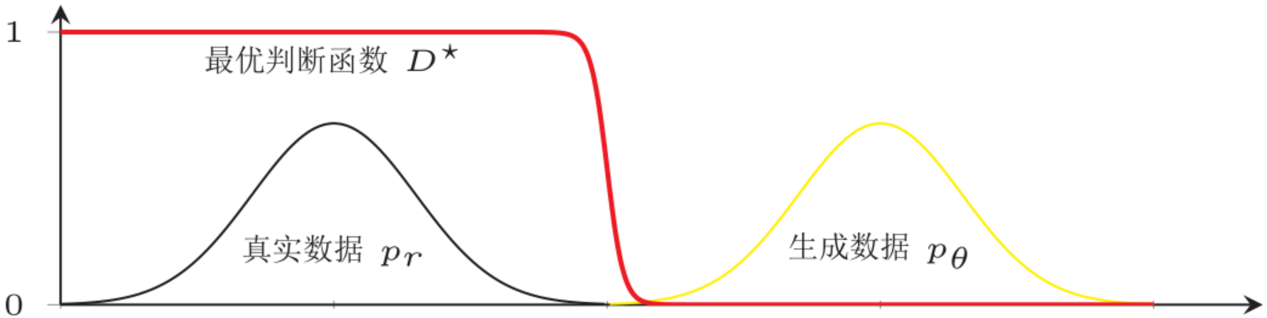
- 首先了解两个重要事实
- 结论1：最优判别器具有显式解 $D^* = \frac{p^*}{p^* + p_g}$，即 D^* 是以下优化问题的解
- $\max_D \mathrm{E}_{x\sim p^*} \log D(x) + \mathrm{E}_{z\sim p(z)} \log[1 - D(G(z))]$
- 结论2：优化目标等于
- $\mathrm{KL}(p^* \| p_a) + \mathrm{KL}(p_g \| p_a) - 2\log 2$，其中 $p_a = \frac{1}{2}(p^* + p_g)$

J-S Divergence

- 换言之，GAN 的优化等价于找到生成器 G，使得 $\mathrm{KL}(p^* \| p_a) + \mathrm{KL}(p_g \| p_a)$ 最小
- 这个量除以2也被称为 Jensen–Shannon divergence

J-S Divergence

- J-S Divergence 的问题在于，当两个分布没有重叠时，其取值恒等于常数 log2 求导
- 此时对生成器 G 来说，关于参数的梯度为0

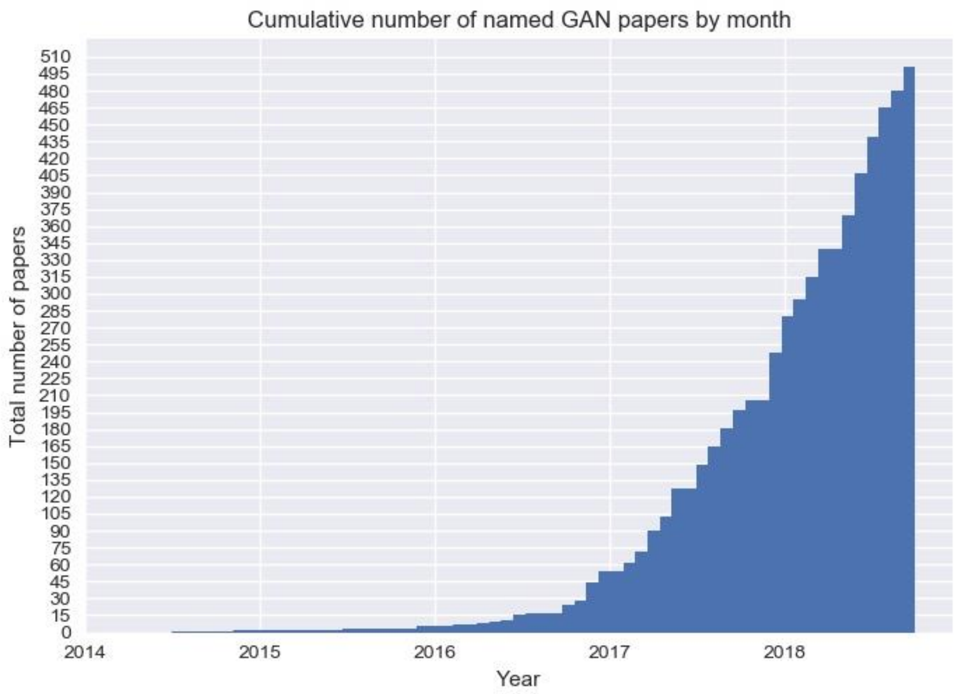


邱锡鹏《神经网络与深度学习》

扩展

- 针对这一问题，有非常多的工作试图对 GAN 进行改进
- 一度成为深度学习的热门课题
- “The GAN Zoo”

https://github.com/hindupuravinash/the-gan-zoo



扩展

- 而其中 Wasserstein GAN 或许是近年来对 GAN 最重要的一个改进

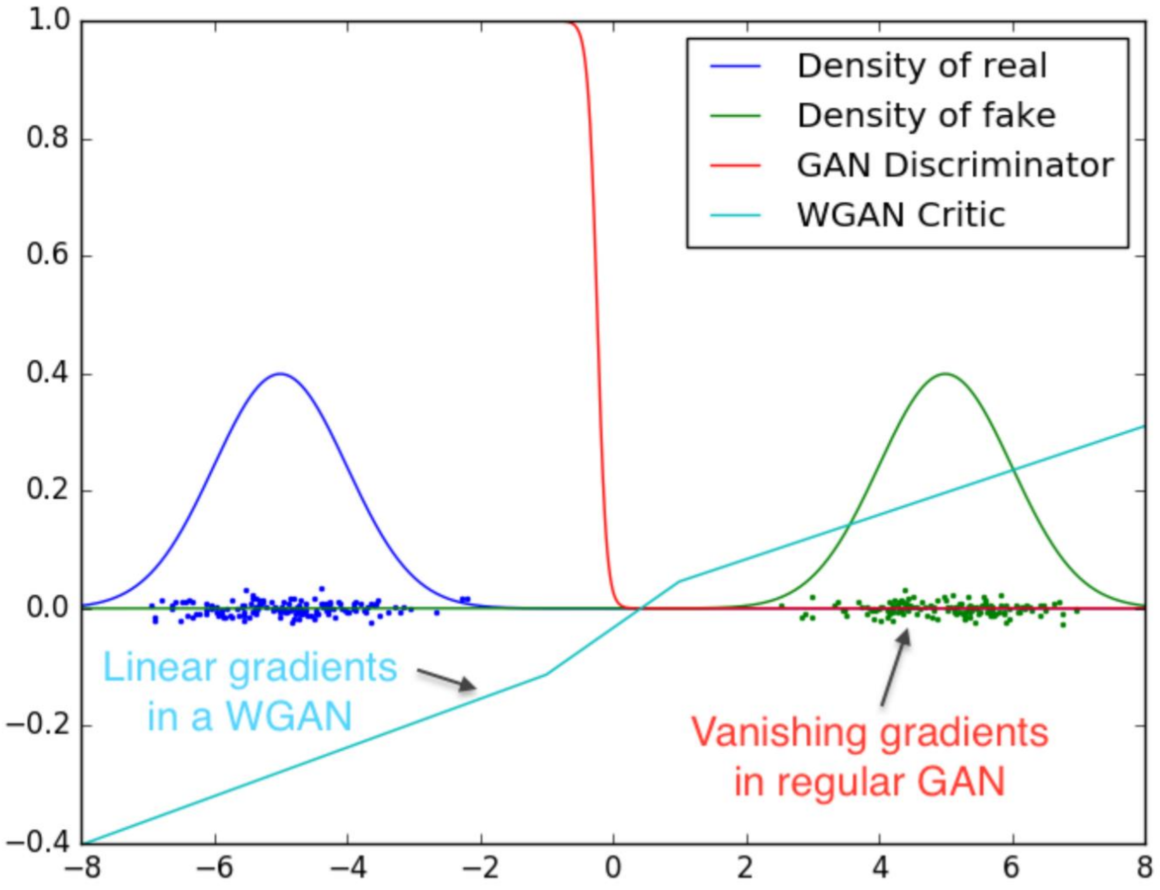
WGAN

- WGAN 的核心是利用 Wasserstein 距离来衡量两个分布之间的距离

$$W_p(q_1, q_2) = \left(\inf_{\gamma(x,y)\in\Gamma(q_1,q_2)} \mathbb{E}_{(x,y)\sim\gamma(x,y)}[d(x,y)^p] \right)^{\frac{1}{p}}$$

- Wasserstein 距离的优势在于，即使两个分布的重叠很少，依然可以反映它们之间的差异，且梯度不会消失

WGAN



Density of real
Density of fake
GAN Discriminator
WGAN Critic
Linear gradients
in a WGAN
Vanishing gradients
in regular GAN
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−8
−6
−4
−2
0
2
4
6
8