《Generative Adversarial Nets》

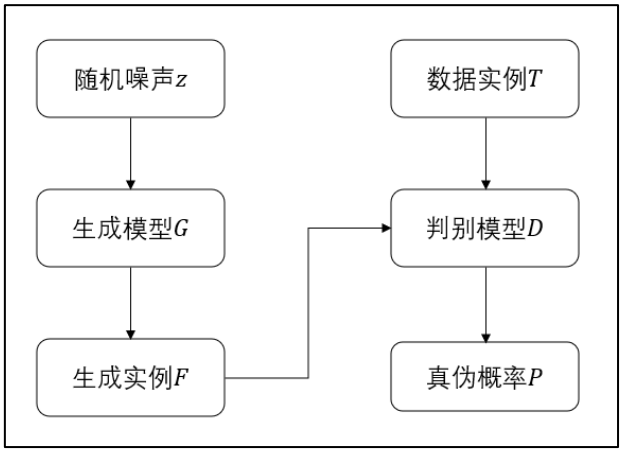
摘要：

1. **论文背景：**深度学习已经在判别模型中取得了一系列较好的成果，但在生成模型领域中尚未有优秀、合适的框架被提出。
2. **论文的贡献：**提出了一个通过对抗过程来估计生成模型的新神经网络框架，它包含两个训练模型：**模拟数据分布的生成模型 G** 和**判别样本数据真伪的判别模型 D**。
3. **主要创新点：**所提出的架构可以有效使用反向传播来进行训练，而不需要运用任何类似“马尔科夫链”复杂的推理网络。此外，框架还结合了一定的博弈理论。
4. **实验分析：**论文中简单证明了对抗性建模框架（**生成对抗网络**）的可行性，并且表明了这些研究方向可能是有用的。【该框架如今已被广泛使用】

GAN 网络整体框架：

在 GAN 中包含有两个模型，一个是**生成模型 G（Generative model）**，一个是**判别模型 D（Discriminative model）**。生成模型 G 的任务是生成看起来自然真实的、和原始数据相似的实例；而判别模型 D 的任务是判断给定的实例看起来是自然真实的，还是人为伪造的（真实实例来源于数据集，伪造实例来源于生成模型 G）。

具体网络结构如图所示：



1）生成模型 G：G 是一个生成伪造实例的网络，它首先接收一个随机的噪声z，然后通过这个噪声来生成伪造（生成）实例F，记做$G(z)$。

2）判别模型 D：D 是一个用于判别的网络，其主要功能就是判别一个实例是不是“真实的”。它的输入是x（x代表一个实例），输出是$D(x)$，它代表x为真实实例T的概率P。如果$P=1$，则代表 100% 是真实实例；而如果$P=0$，则代表不可能是真实实例。

注意：这里的网络是指 MLP（多层感知器）。

GAN 模型优化训练：

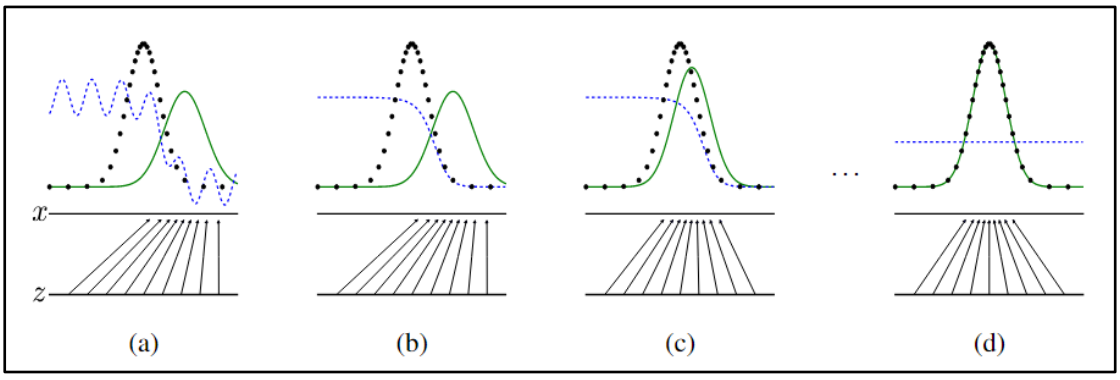
GAN 的目标优化函数如下：

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

训练的目标就是优化判别网络D，使得最大概率地分对训练样本的标签（即最大化$log\ D(x)$和$log\ (1-D(G(z)))$）；优化生成网络G，使得最小化$log(1-D(G(z)))$，即最大化D的损失。

而在训练过程中，首先固定一方网络，然后再更新另一个网络的参数，并交替迭代，使得对方的错误最大化【类似于博弈过程】。最终，生成模型 G 就能估测出样本数据的分布，即使得生成的样本更加真实。

对抗过程可视化：



如图所示，在模型的对抗训练中，可能会出现以上几种情况：

图(a)：D（蓝色线）刚开始训练，其本身判别能力有限，结果存在波动，但还是能初步区分实际数据和生成数据。

图(b)：在 D 经过有效训练后，可以明显地区分出生成数据。

图(c)：G 优化自己的输出分布（绿色线），使得其更加接近真实数据的分布（黑色线），从而使得 D 更加难以判别实际数据和生成数据。

图(d)：由于 G 的不断提升，其输出分布几乎已经等同于真实数据的分布，从而使得 D 难以判别二者，因此 D 的输出结果趋近于1/2。至此，网络 G 和 D 处于**纳什均衡状态**，无法再进一步优化。

整体算法的实现步骤：

如图所示：

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

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

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

  end for

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

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

end for

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

首先，用k步优化判别模型 D 的参数θ_d。然后，固定 D，优化生成模型 G 的参数θ_g（采用反向传播来优化网络参数）。以上过程进行n次迭代。

相关数学证明：

当生成器G确定时，判别器D的最优解为：

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

Proof.

$$\because \mathbb{E}_{x \sim p(x)} f(x) = \int_x p(x) f(x)\, dx$$

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

$$= \int_x p_{data}(x) log D(x)\, dx + \int_z p_z(z) \log\Big(1 - D\big(G(z)\big)\Big)\, dz$$

$$= \int_x p_{data}(x) log D(x) + p_g(x) \log\,(1 - D(x))\, dx$$

∴ 对于$p_{data}(x) log D(x) + p_g(x) \log\big(1 - D(x)\big)$的最大值可近似于：

$$f(y) = a log y + b log(1 - y), \text{对} f(y) \text{求最大值}$$

$$\because f'(y) = \frac{a}{y} - \frac{b}{1 - y}, \text{且} f(y) \text{为凸函数}$$

$$\therefore f(y) \text{的最大值为} f'(y) = 0, \text{即} y = \frac{a}{a + b}$$

$$\therefore D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

得证。

下面求生成器G的最优解。

由于已知$D_G^*(x)$，因此有：

$$C(G) = \max_D V(G, D)$$

$$= \mathbb{E}_{x \sim p_{data}(x)}[log D_G^*(x)] + \mathbb{E}_{z \sim p_z(z)}\left[\log\Big(1 - D_G^*\big(G(z)\big)\Big)\right]$$

$$= \mathbb{E}_{x \sim p_{data}(x)}[log D_G^*(x)] + \mathbb{E}_{x \sim p_g(x)}\big[\log\big(1 - D_G^*(x)\big)\big]$$

$$= \mathbb{E}_{x \sim p_{data}(x)}\left[log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}\right] + \mathbb{E}_{x \sim p_g(x)}\left[\log \frac{p_g(x)}{p_{data}(x) + p_g(x)}\right]$$

∴ 对$C(G)$求最小值，当且仅当$p_{data}(x) = p_g(x)$，且值为$-log4$

Proof.

$$\because log \frac{p_{data}}{p_{data} + p_g} = log \frac{p_{data}}{\frac{p_{data} + p_g}{2}} * \frac{1}{2}$$

$$= log \frac{p_{data}}{\frac{p_{data} + p_g}{2}} - log2$$

$$\text{又} \because \mathbb{E}_{x \sim p_{data}}\left[log \frac{p_{data}}{\frac{p_{data} + p_g}{2}}\right] = KL\left(p_{data} \middle\| \frac{p_{data} + p_g}{2}\right)$$

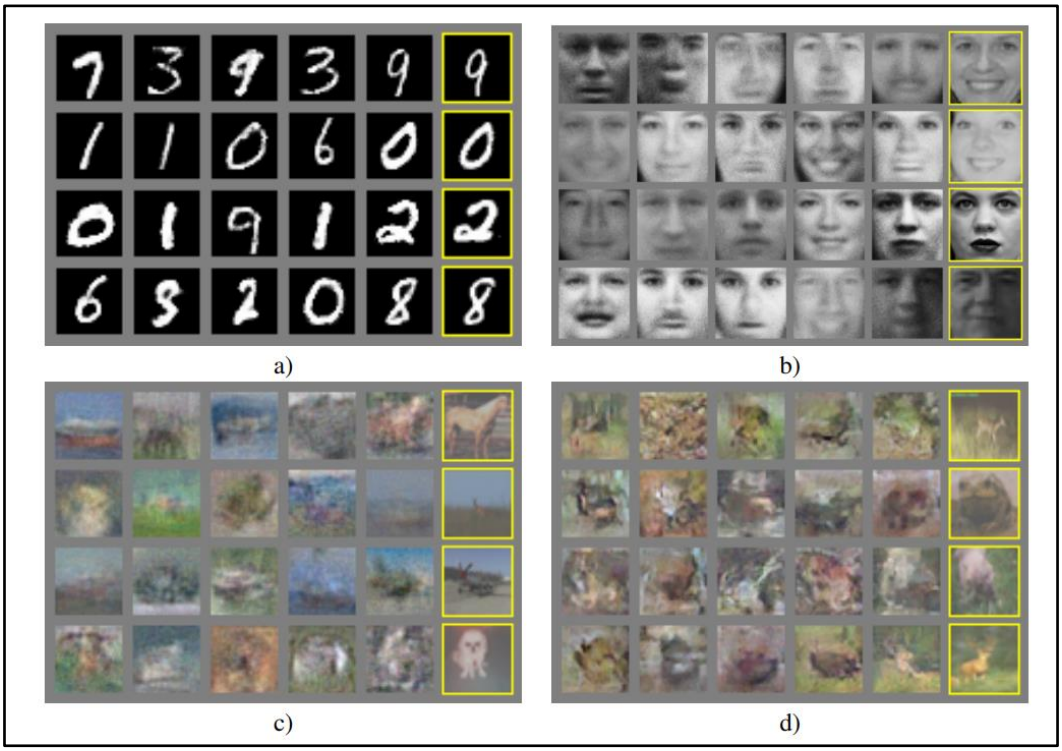
$$\therefore C(G) = -log4 + KL\left(p_{data} \middle\| \frac{p_{data} + p_g}{2}\right) + KL\left(p_g \middle\| \frac{p_{data} + p_g}{2}\right)$$

$$\because KL\left(p_{data}\left\|\frac{p_{data}+p_g}{2}\right.\right)\geq 0, \text{且当} p_{data}=\frac{p_{data}+p_g}{2}\text{时，值为 0}$$

$$\therefore C^*=-log4, \text{当且仅当} p_{data}(x)=p_g(x)$$

得证。

实验结果：



注：最后一列为 G 生成的图片。

总结：

优点：

1）GAN 是一种以半监督方式训练分类器的方法。

2）G 的参数更新不是直接来自数据样本，而是使用来自 D 的反向传播。所以理论上，只要是可微分函数都可以用于构建 D 和 G。

3）GAN 可以比推理网络更快的产生样本，因为它不需要在采样序列下生成不同的数据。

4）GAN 框架只用到了反向传播，不涉及马尔科夫链。

缺点：

1）训练 GAN 需要达到纳什均衡，虽然有时候可用梯度下降法做到，但有时

候却做不到，而本文还没有找到一个很好的达到纳什均衡的方法。

2）GAN 很难去学习、生成离散的数据，例如文本数据。

对本文的感悟：

GAN 的提出使得生成模型领域有了重大突破。作为一个开创性的方法（框架），它有效结合了博弈理论，并使得深度学习研究有了更宽阔的思路，以至于而后的研究者们在此之上做出了大量的改进和模型变种。