

# 深入理解生成对抗网络

翟俊海

河北大学

2017.07.21

# Outlines

- 1 解决的问题
- 2 解决问题的思路
- 3 结构选择问题
- 4 损失函数
- 5 GANs的训练
- 6 关于GANs及其训练的几个问题

# Outlines

- 1 解决的问题
- 2 解决问题的思路
- 3 结构选择问题
- 4 损失函数
- 5 GANs的训练
- 6 关于GANs及其训练的几个问题

- **解决的问题:** 概率密度估计问题.
- **问题的描述:** 给定一个未知概率分布  $P_{data}$  的数据集  $X$  和一个已知概率分布(如, 高斯分布)的数据集  $Z$ , 用某种方法估计  $P_{data}$ .

- **解决的问题:** 概率密度估计问题.
- **问题的描述:** 给定一个未知概率分布  $P_{data}$  的数据集  $X$  和一个已知概率分布(如, 高斯分布)的数据集  $Z$ , 用某种方法估计  $P_{data}$ .

# Outlines

- 1 解决的问题
- 2 解决问题的思路
- 3 结构选择问题
- 4 损失函数
- 5 GANs的训练
- 6 关于GANs及其训练的几个问题

## 解决问题的思路

- 用生成对抗网络(见下图), 通过学习估计  $P_{data}$ . 生成对抗网络由一个生成器  $G$  和一个判别器  $D$  构成.
- 生成器  $G$  的输入是  $\mathbf{z} \in Z$ , 输出为  $G(\mathbf{z})$ . 判别器  $D$  的输入为  $G(\mathbf{z})$  和  $\mathbf{x} \in X$ , 其职责是判别输入的样例是来自  $X$ ? 还是来自生成器  $G$ ?
- 通过训练或学习, 使得判别器  $D$  不能判别其输入是来自  $X$  还是来自  $G$ . 这时, 我们说生成器和判别器达到了纳什均衡.
- 学到的  $G$  (不是显式的概率分布函数, 而是隐式的概率分布函数  $P_{model}$ ) 就是  $P_{data}$  的逼近.

## 解决问题的思路

- 用生成对抗网络(见下图), 通过学习估计  $P_{data}$ . 生成对抗网络由一个生成器  $G$  和一个判别器  $D$  构成.
- 生成器  $G$  的输入是  $\mathbf{z} \in Z$ , 输出为  $G(\mathbf{z})$ . 判别器  $D$  的输入为  $G(\mathbf{z})$  和  $\mathbf{x} \in X$ , 其职责是判别输入的样例是来自  $X$ ? 还是来自生成器  $G$ ?
- 通过训练或学习, 使得判别器  $D$  不能判别其输入是来自  $X$  还是来自  $G$ . 这时, 我们说生成器和判别器达到了纳什均衡.
- 学到的  $G$  (不是显式的概率分布函数, 而是隐式的概率分布函数  $P_{model}$ ) 就是  $P_{data}$  的逼近.

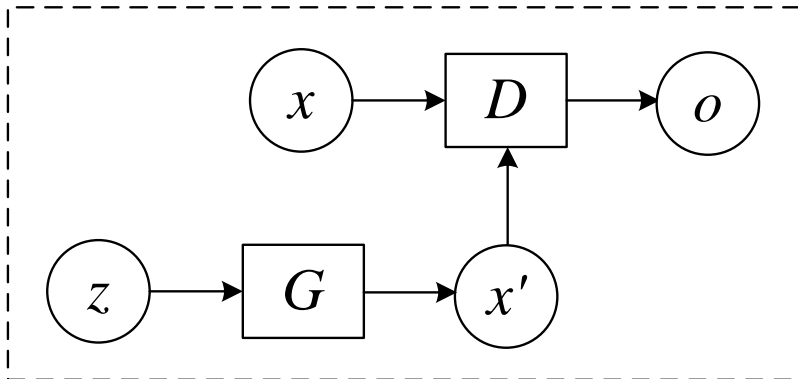


## 解决问题的思路

- 用生成对抗网络(见下图), 通过学习估计  $P_{data}$ . 生成对抗网络由一个生成器  $G$  和一个判别器  $D$  构成.
- 生成器  $G$  的输入是  $\mathbf{z} \in Z$ , 输出为  $G(\mathbf{z})$ . 判别器  $D$  的输入为  $G(\mathbf{z})$  和  $\mathbf{x} \in X$ , 其职责是判别输入的样例是来自  $X$ ? 还是来自生成器  $G$ ?
- 通过训练或学习, 使得判别器  $D$  不能判别其输入是来自  $X$  还是来自  $G$ . 这时, 我们说生成器和判别器达到了纳什均衡.
- 学到的  $G$  (不是显式的概率分布函数, 而是隐式的概率分布函数  $P_{model}$ ) 就是  $P_{data}$  的逼近.

## 解决问题的思路

- 用生成对抗网络(见下图), 通过学习估计  $P_{data}$ . 生成对抗网络由一个生成器  $G$  和一个判别器  $D$  构成.
- 生成器  $G$  的输入是  $\mathbf{z} \in Z$ , 输出为  $G(\mathbf{z})$ . 判别器  $D$  的输入为  $G(\mathbf{z})$  和  $\mathbf{x} \in X$ , 其职责是判别输入的样例是来自  $X$ ? 还是来自生成器  $G$ ?
- 通过训练或学习, 使得判别器  $D$  不能判别其输入是来自  $X$  还是来自  $G$ . 这时, 我们说生成器和判别器达到了纳什均衡.
- 学到的  $G$  (不是显式的概率分布函数, 而是隐式的概率分布函数  $P_{model}$ ) 就是  $P_{data}$  的逼近.



# Outlines

- 1 解决的问题
- 2 解决问题的思路
- 3 结构选择问题
- 4 损失函数
- 5 GANs的训练
- 6 关于GANs及其训练的几个问题

## 结构选择问题

- 一般地,  $G$ 和 $D$ 都是神经网络(如卷积神经网络、自动编码器、受限玻耳兹曼机、ELM网络、多层感知器等;
- 针对具体形式, 应进行具体分析, 如果它们是自动编码器的话, 那么如何分析? 如果它们是ELM的话, 那么又如何分析?

## 结构选择问题

- 一般地,  $G$ 和 $D$ 都是神经网络(如卷积神经网络、自动编码器、受限玻耳兹曼机、ELM网络、多层感知器等;
- 针对具体形式, 应进行具体分析, 如果它们是自动编码器的话, 那么如何分析? 如果它们是ELM的话, 那么又如何分析?

# Outlines

- 1 解决的问题
- 2 解决问题的思路
- 3 结构选择问题
- 4 损失函数
- 5 GANs的训练
- 6 关于GANs及其训练的几个问题

## 损失函数

- 在经典GANs中, 提出了两种损失函数, 即minmax GAN(MMGAN)损失函数和non-saturating GAN(NSGAN)损失函数.
- 对于前者, 判别器 $D$ 的损失函数为 $L_D^{MMGAN} = -E_{\mathbf{x} \sim p_{data}}[\log(D(\mathbf{x}))] + E_{\mathbf{z} \sim p_{noise}}[\log(1 - D(G(\mathbf{z})))]$ , 生成器 $G$ 的损失函数为 $L_G^{MMGAN} = -L_D^{MMGAN}$ .
- 对于后者, 判别器 $D$ 的损失函数为 $L_D^{NSGAN} = L_D^{MMGAN}$ , 生成器 $G$ 的损失函数为 $L_G^{NSGAN} = E_{\mathbf{z} \sim p_{noise}}[\log(D(G(\mathbf{z})))]$ .
- 将生成器 $G$ 和判别器 $D$ 放在一起考虑(i.e. two-player minimax game), 相应的优化问题为 $\min_G \max_D V(D, G) = E_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}))] + E_{\mathbf{z} \sim p_{noise}}[\log(1 - D(G(\mathbf{z})))]$ .



## 损失函数

- 在经典GANs中, 提出了两种损失函数, 即minmax GAN(MMGAN)损失函数和non-saturating GAN(NSGAN)损失函数.
- 对于前者, 判别器 $D$ 的损失函数为 $L_D^{MMGAN} = -E_{\mathbf{x} \sim p_{data}}[\log(D(\mathbf{x}))] + E_{\mathbf{z} \sim p_{noise}}[\log(1 - D(G(\mathbf{z})))]$ , 生成器 $G$ 的损失函数为 $L_G^{MMGAN} = -L_D^{MMGAN}$ .
- 对于后者, 判别器 $D$ 的损失函数为 $L_D^{NSGAN} = L_D^{MMGAN}$ , 生成器 $G$ 的损失函数为 $L_G^{NSGAN} = E_{\mathbf{z} \sim p_{noise}}[\log(D(G(\mathbf{z})))]$ .
- 将生成器 $G$ 和判别器 $D$ 放在一起考虑(i.e. two-player minimax game), 相应的优化问题为 $\min_G \max_D V(D, G) = E_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}))] + E_{\mathbf{z} \sim p_{noise}}[\log(1 - D(G(\mathbf{z})))]$ .

## 损失函数

- 在经典GANs中, 提出了两种损失函数, 即minmax GAN(MMGAN)损失函数和non-saturating GAN(NSGAN)损失函数.
- 对于前者, 判别器 $D$ 的损失函数为 $L_D^{MMGAN} = -E_{\mathbf{x} \sim p_{data}}[\log(D(\mathbf{x}))] + E_{\mathbf{z} \sim p_{noise}}[\log(1 - D(G(\mathbf{z})))]$ , 生成器 $G$ 的损失函数为 $L_G^{MMGAN} = -L_D^{MMGAN}$ .
- 对于后者, 判别器 $D$ 的损失函数为 $L_D^{NSGAN} = L_D^{MMGAN}$ , 生成器 $G$ 的损失函数为 $L_G^{NSGAN} = E_{\mathbf{z} \sim p_{noise}}[\log(D(G(\mathbf{z})))]$ .
- 将生成器 $G$ 和判别器 $D$ 放在一起考虑(i.e. two-player minimax game), 相应的优化问题为 $\min_G \max_D V(D, G) = E_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}))] + E_{\mathbf{z} \sim p_{noise}}[\log(1 - D(G(\mathbf{z})))]$ .

## 损失函数

- 在经典GANs中, 提出了两种损失函数, 即minmax GAN(MMGAN)损失函数和non-saturating GAN(NSGAN)损失函数.
- 对于前者, 判别器 $D$ 的损失函数为 $L_D^{MMGAN} = -E_{\mathbf{x} \sim p_{data}}[\log(D(\mathbf{x}))] + E_{\mathbf{z} \sim p_{noise}}[\log(1 - D(G(\mathbf{z})))]$ , 生成器 $G$ 的损失函数为 $L_G^{MMGAN} = -L_D^{MMGAN}$ .
- 对于后者, 判别器 $D$ 的损失函数为 $L_D^{NSGAN} = L_D^{MMGAN}$ , 生成器 $G$ 的损失函数为 $L_G^{NSGAN} = E_{\mathbf{z} \sim p_{noise}}[\log(D(G(\mathbf{z})))]$ .
- 将生成器 $G$ 和判别器 $D$ 放在一起考虑(i.e. two-player minimax game), 相应的优化问题为 $\min_G \max_D V(D, G) = E_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}))] + E_{\mathbf{z} \sim p_{noise}}[\log(1 - D(G(\mathbf{z})))]$ .

# Outlines

- 1 解决的问题
- 2 解决问题的思路
- 3 结构选择问题
- 4 损失函数
- 5 GANs的训练**
- 6 关于GANs及其训练的几个问题

---

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

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

**end for**

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

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

**end for**

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

# Outlines

- 1 解决的问题
- 2 解决问题的思路
- 3 结构选择问题
- 4 损失函数
- 5 GANs的训练
- 6 关于GANs及其训练的几个问题

## 关于GANs及其训练的几个问题

- GANs用简单分布(高斯分布)逼近复杂分布, 其数学原理是什么? 是概率论中的中心极限定理?
- 如果 $X$ 是图像数据, 那么生成器 $G$ 和判别器 $D$ 都是卷积神经网络, 对 $Z$ 的要求是什么? 是二维高斯分布?
- 如果 $X$ 是 $d$ 维表格数据, 那么生成器 $G$ 和判别器 $D$ 都是MLP? 对 $Z$ 的要求是什么? 是 $d$ 维高斯分布?
- 如果 $X$ 是文本数据, 那么生成器 $G$ 和判别器 $D$ 都是LSTM?  $Z$ 是什么样子的数据? 如何得到?
- 关于生成器 $G$ 和判别器 $D$ 结构选择问题. 针对具体应用, 生成器 $G$ 和判别器 $D$ 结构选择有无标准遵循? 异构网络可以吗? 不是神经网络可以吗? MMD GANs中的判别器就不是神经网络.
- 损失函数问题. 针对不同应用, 什么样的损失函数为最佳选择? 或设计或选择.

## 关于GANs及其训练的几个问题

- GANs用简单分布(高斯分布)逼近复杂分布, 其数学原理是什么? 是概率论中的中心极限定理?
- 如果 $X$ 是图像数据, 那么生成器 $G$ 和判别器 $D$ 都是卷积神经网络, 对 $Z$ 的要求是什么? 是二维高斯分布?
- 如果 $X$ 是 $d$ 维表格数据, 那么生成器 $G$ 和判别器 $D$ 都是MLP? 对 $Z$ 的要求是什么? 是 $d$ 维高斯分布?
- 如果 $X$ 是文本数据, 那么生成器 $G$ 和判别器 $D$ 都是LSTM?  $Z$ 是什么样子的数据? 如何得到?
- 关于生成器 $G$ 和判别器 $D$ 结构选择问题. 针对具体应用, 生成器 $G$ 和判别器 $D$ 结构选择有无标准遵循? 异构网络可以吗? 不是神经网络可以吗? MMD GANs中的判别器就不是神经网络.
- 损失函数问题. 针对不同应用, 什么样的损失函数为最佳选择? 或设计或选择.



## 关于GANs及其训练的几个问题

- GANs用简单分布(高斯分布)逼近复杂分布, 其数学原理是什么? 是概率论中的中心极限定理?
- 如果 $X$ 是图像数据, 那么生成器 $G$ 和判别器 $D$ 都是卷积神经网络, 对 $Z$ 的要求是什么? 是二维高斯分布?
- 如果 $X$ 是 $d$ 维表格数据, 那么生成器 $G$ 和判别器 $D$ 都是MLP? 对 $Z$ 的要求是什么? 是 $d$ 维高斯分布?
- 如果 $X$ 是文本数据, 那么生成器 $G$ 和判别器 $D$ 都是LSTM?  $Z$ 是什么样子的数据? 如何得到?
- 关于生成器 $G$ 和判别器 $D$ 结构选择问题. 针对具体应用, 生成器 $G$ 和判别器 $D$ 结构选择有无标准遵循? 异构网络可以吗? 不是神经网络可以吗? MMD GANs中的判别器就不是神经网络.
- 损失函数问题. 针对不同应用, 什么样的损失函数为最佳选择? 或设计或选择.

## 关于GANs及其训练的几个问题

- GANs用简单分布(高斯分布)逼近复杂分布, 其数学原理是什么? 是概率论中的中心极限定理?
- 如果 $X$ 是图像数据, 那么生成器 $G$ 和判别器 $D$ 都是卷积神经网络, 对 $Z$ 的要求是什么? 是二维高斯分布?
- 如果 $X$ 是 $d$ 维表格数据, 那么生成器 $G$ 和判别器 $D$ 都是MLP? 对 $Z$ 的要求是什么? 是 $d$ 维高斯分布?
- 如果 $X$ 是文本数据, 那么生成器 $G$ 和判别器 $D$ 都是LSTM?  $Z$ 是什么样子的数据? 如何得到?
- 关于生成器 $G$ 和判别器 $D$ 结构选择问题. 针对具体应用, 生成器 $G$ 和判别器 $D$ 结构选择有无标准遵循? 异构网络可以吗? 不是神经网络可以吗? MMD GANs中的判别器就不是神经网络.
- 损失函数问题. 针对不同应用, 什么样的损失函数为最佳选择? 或设计或选择.

## 关于GANs及其训练的几个问题

- GANs用简单分布(高斯分布)逼近复杂分布, 其数学原理是什么? 是概率论中的中心极限定理?
- 如果 $X$ 是图像数据, 那么生成器 $G$ 和判别器 $D$ 都是卷积神经网络, 对 $Z$ 的要求是什么? 是二维高斯分布?
- 如果 $X$ 是 $d$ 维表格数据, 那么生成器 $G$ 和判别器 $D$ 都是MLP? 对 $Z$ 的要求是什么? 是 $d$ 维高斯分布?
- 如果 $X$ 是文本数据, 那么生成器 $G$ 和判别器 $D$ 都是LSTM?  $Z$ 是什么样子的数据? 如何得到?
- 关于生成器 $G$ 和判别器 $D$ 结构选择问题. 针对具体应用, 生成器 $G$ 和判别器 $D$ 结构选择有无标准遵循? 异构网络可以吗? 不是神经网络可以吗? MMD GANs中的判别器就不是神经网络.
- 损失函数问题. 针对不同应用, 什么样的损失函数为最佳选择? 或设计或选择.

## 关于GANs及其训练的几个问题

- GANs用简单分布(高斯分布)逼近复杂分布, 其数学原理是什么? 是概率论中的中心极限定理?
- 如果 $X$ 是图像数据, 那么生成器 $G$ 和判别器 $D$ 都是卷积神经网络, 对 $Z$ 的要求是什么? 是二维高斯分布?
- 如果 $X$ 是 $d$ 维表格数据, 那么生成器 $G$ 和判别器 $D$ 都是MLP? 对 $Z$ 的要求是什么? 是 $d$ 维高斯分布?
- 如果 $X$ 是文本数据, 那么生成器 $G$ 和判别器 $D$ 都是LSTM?  $Z$ 是什么样子的数据? 如何得到?
- 关于生成器 $G$ 和判别器 $D$ 结构选择问题. 针对具体应用, 生成器 $G$ 和判别器 $D$ 结构选择有无标准遵循? 异构网络可以吗? 不是神经网络可以吗? MMD GANs中的判别器就不是神经网络.
- 损失函数问题. 针对不同应用, 什么样的损失函数为最佳选择? 或设计或选择.

## 说明

- 生成器 $G$ 隐式地定义了一个抽样 $G(\mathbf{z})$ 的概率分布函数 $p_{model}$ ，而抽样是由 $\mathbf{z} \sim p_{noise}$ 作为 $G(\mathbf{z})$ 的输入得到的；
- 生成对抗网络的优化目标是JENSEN-SHANNON散度：
$$JSD(p_{data} \parallel p_{model}) = \frac{1}{2} KL(p_{data} \parallel p_{average}) + \frac{1}{2} KL(p_{model} \parallel p_{average})$$
；
- 其中， $p_{average} = \frac{1}{2}[p_{data} + p_{model}]$ .

## 说明

- 生成器 $G$ 隐式地定义了一个抽样 $G(\mathbf{z})$ 的概率分布函数 $p_{model}$ ，而抽样是由 $\mathbf{z} \sim p_{noise}$ 作为 $G(\mathbf{z})$ 的输入得到的；
- 生成对抗网络的优化目标是JENSEN-SHANNON散度：
$$JSD(p_{data} \parallel p_{model}) = \frac{1}{2}KL(p_{data} \parallel p_{average}) + \frac{1}{2}KL(p_{model} \parallel p_{average})$$
；
- 其中， $p_{average} = \frac{1}{2}[p_{data} + p_{model}]$ .

## 说明

- 生成器 $G$ 隐式地定义了一个抽样 $G(\mathbf{z})$ 的概率分布函数 $p_{model}$ ，而抽样是由 $\mathbf{z} \sim p_{noise}$ 作为 $G(\mathbf{z})$ 的输入得到的；
- 生成对抗网络的优化目标是JENSEN-SHANNON散度：
$$JSD(p_{data} \parallel p_{model}) = \frac{1}{2} KL(p_{data} \parallel p_{average}) + \frac{1}{2} KL(p_{model} \parallel p_{average})$$
；
- 其中， $p_{average} = \frac{1}{2}[p_{data} + p_{model}]$ 。

# The End