GAN: Generative Adversarial Nets

——He Sun

from USTC

1 简介

1.1 怎么来的?

有一天，**Goodfellow**在酒吧和一位朋友讨论如何用计算机更好地自动生成想要的数据，灵光一现，想到了用两个模型对抗的思想来提高生成模型的质量。回家实现了想法，并且在**NIPS2014**上发表了论文**Generative adversarial nets**。

Generative Adversarial Nets

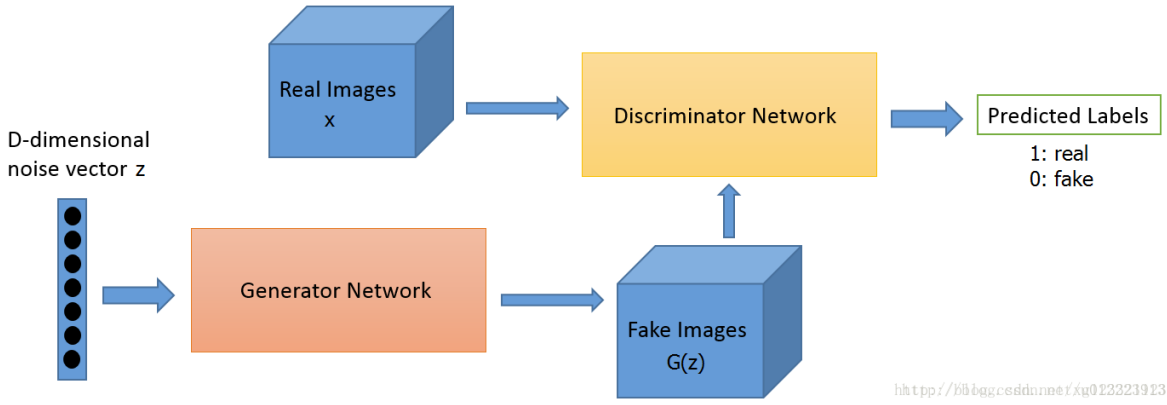
Ian J. Goodfellow, Jean Pouget-Abadie,* Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,† Aaron Courville, Yoshua Bengio‡
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

1.2 是什么?

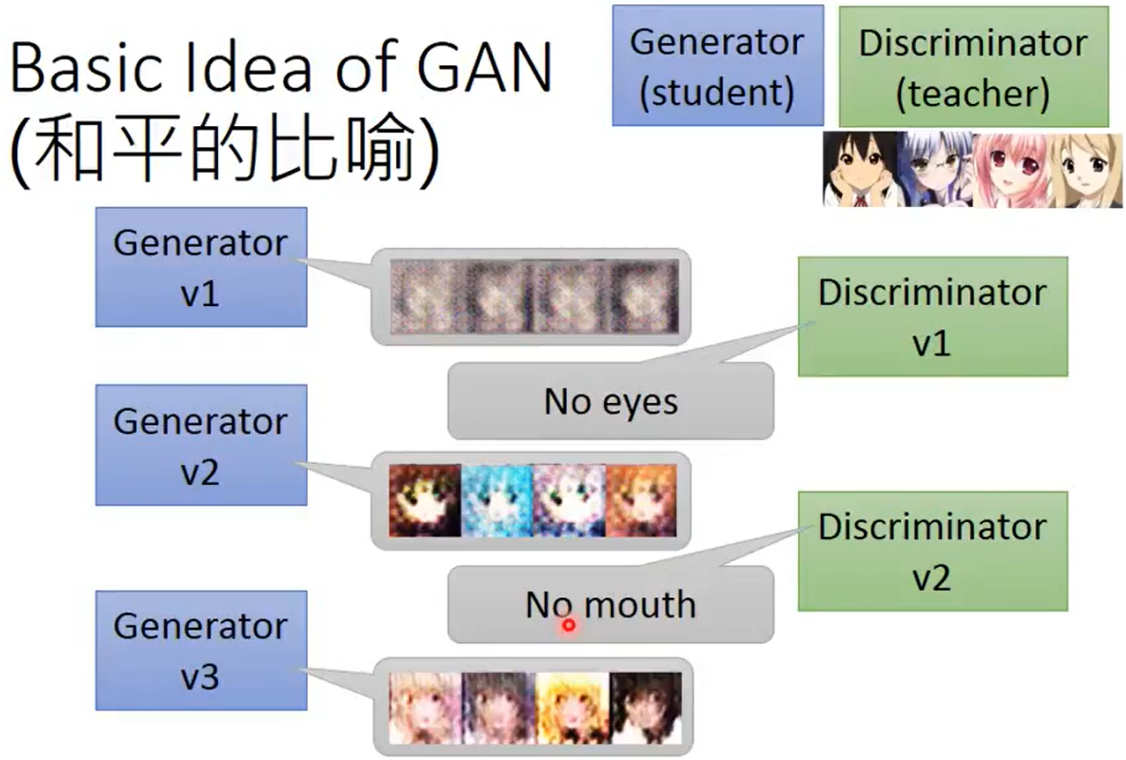
GAN是用对抗方法来生成数据的一种模型， 和其他机器学习模型相比，**GAN**引人注目的地方在于给机器学习引入了对抗这一理念。

GAN是一种二人零和博弈思想**(two-player game)**，博弈双方的利益之和是一个常数。

主要由①生成器**G**（专门造假的）②判别器**D**（专门鉴别的）组成.



举个例子:

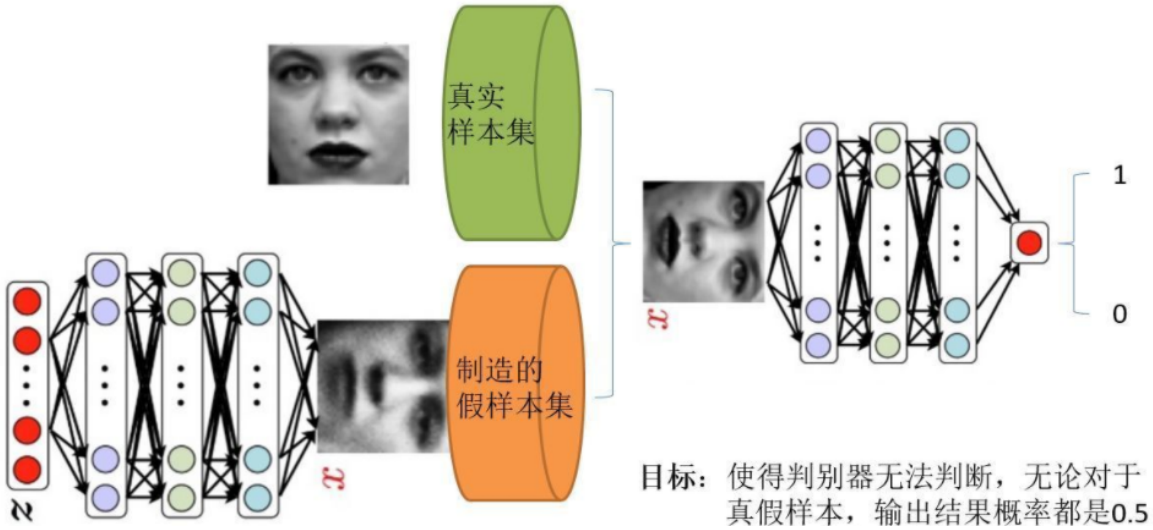


1.3 评价

“Adversarial training is the coolest thing since sliced bread”——Yann LeCun

2 数学模型

2.1 对抗网络

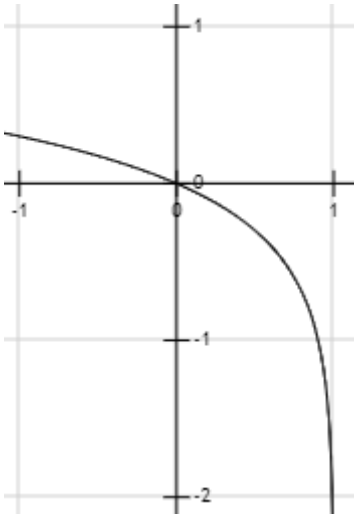


1.为了在数据x上学习生成器的分布p_g，生成网络$G(.)$接收概率密度为p_z的随机输入z，返回训练后服从目标概率分布的结果$x_g = G(z;\theta_g)$，G由多个感知器表示的可微函数，参数是θ_g.

2. 判别网络$D(.)$的输入x可以是真实的样本$(x_t$，概率密度$p_t)$也可以是生成样本$(x_g$，其概率密度p_g受到p_z影响$)$，判别网络$D(x,\theta_d)$输出为一个标量，表示x数据来自数据而不是p_g的概率。

训练D——以最大化为G训练样本和样本分配正确标签的概率。

训练G——以最小化$log(1-D(G(z)))$



也就是D和G使用函数$V(G;D)$进行以下双人最小最大值游戏：

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

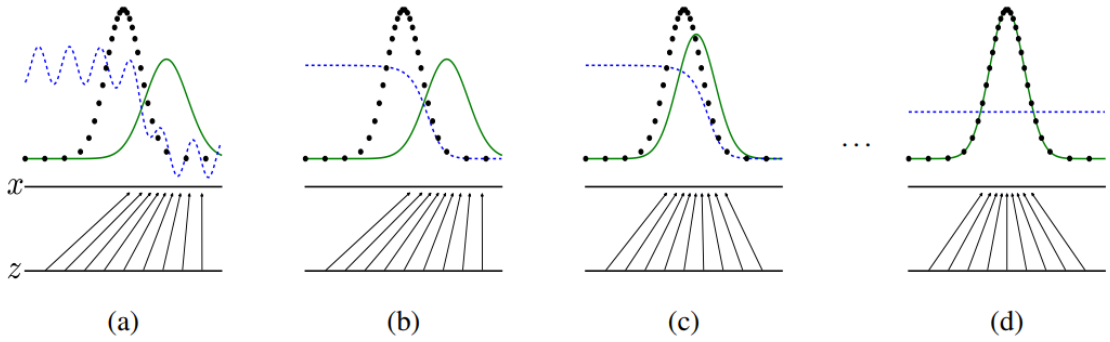


Figure 1: Generative adversarial nets are trained by simultaneously updating the **d**iscriminative distribution (D, blue, dashed line) so that it discriminates between samples from the data generating distribution (black, dotted line) $p_{\boldsymbol{x}}$ from those of the **g**enerative distribution p_g (G) (green, solid line). The lower horizontal line is the domain from which $\boldsymbol{z}$ is sampled, in this case uniformly. The horizontal line above is part of the domain of $\boldsymbol{x}$. The upward arrows show how the mapping $\boldsymbol{x} = G(\boldsymbol{z})$ imposes the non-uniform distribution p_g on transformed samples. G contracts in regions of high density and expands in regions of low density of p_g. (a) Consider an adversarial pair near convergence: p_g is similar to p_{data} and D is a partially accurate classifier. (b) In the inner loop of the algorithm D is trained to discriminate samples from data, converging to $D^*(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})+p_g(\boldsymbol{x})}$. (c) After an update to G, gradient of D has guided $G(\boldsymbol{z})$ to flow to regions that are more likely to be classified as data. (d) After several steps of training, if G and D have enough capacity, they will reach a point at which both cannot improve because $p_g = p_{\text{data}}$. The discriminator is unable to differentiate between the two distributions, i.e. $D(\boldsymbol{x}) = \frac{1}{2}$.

2.2 理论分析

2.2.1 模型训练算法

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.

2.2.2 全局最优解 $p_{data} = p_g$

p_g被定义为当生成器G输入$z \sim p_z$时所生成的数据的概率分布。如果给定充足的时间和存储，算法**1**将会收敛到良好的预期。

首先给定任意生成器G，我们考虑最优判别器D：.

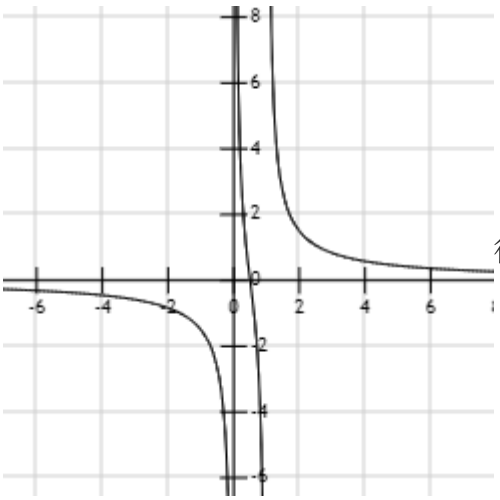
$$D_G^*(x) = \frac{p_{data}(x)}{p_{data(x)} + p_g(x)}$$

Prof: 此时目标是最大化目标函数$V(G; D)$

$$\begin{aligned} V(G, D) &= \int_x p_{data} log(D(x)) dx + \int_z p_z(z) log(1 - D(g(z))) dz \\ &= \int_x p_{data}(x) log(D(x)) + p_g(x) log(1 - D(x)) dx \end{aligned}$$

$$\Longrightarrow y \longrightarrow a log(y) + b log(1 - y) \quad \text{其中} (a, b) \in R/(0, 0)$$

$$V' = \frac{a}{y} - \frac{b}{1 - y} = 0$$



得$y = \frac{a}{a+b}$ 此时目标函数最大

V'

判别函数D的训练目标就是最大化条件概率$P(Y=y|x)$对数似然估计，如果x来自p_{data}，则$y=1$；如果x来自p_g，则$y=0$。那么，$minmaxV(G,D)$函数可以表示为：

$$\begin{aligned} C(G) &= \max_D V(G,D) \\ &= E_{x\sim p_{data}}[log(D_G^*(x))] + E_{z\sim p_z}[log(1-D_G^*(G(z)))] \\ &= E_{x\sim p_{data}}[log(D_G^*(x))] + E_{x\sim p_g}[log(1-D_G^*(x))] \\ &= E_{x\sim p_{data}}[log(\frac{p_{data}(x)}{p_{data}(x)+p_g(x)})] + E_{x\sim p_g}[log(\frac{p_g(x)}{p_{data}(x)+p_g(x)})] \end{aligned}$$

Theorem 1. 当且仅当$p_{data}=p_g$时候，$C(G)$达到虚拟训练标准的全局最小。此时$C(G)$的值为$-log4$

Prof:

当$p_g=p_{data}$时，$D_G^*=\frac{1}{2}$。此时$C(G)=log\frac{1}{2}+log\frac{1}{2}=-log4$，当$p_{data}=p_g$，$C(G)$是否最优，观测到：

$$E_{x\sim p_{data}}[-log2]+E_{x\sim p_g}[-log2]=-log4$$

从$C(G)=V(D_G^*,G)$减去上式，得：

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

判别模型与生成模型之间的**JS**散度**:**

$$C(G)=-log4+2\cdot JSD(p_{data}||p_g)$$

由于两个分布之间的**JS**散度总是非负的。并且当两个分布相等时，值为**0**，因此$C^*=-log4$为$C(G)$的全局极小值，并且唯一解为$p_g=p_{data}$，这就表示生成模型可以完美复制数据的生成过程。

相关知识：

1. KL散度

KL散度又称为相对熵，信息散度，信息增益。KL散度是是两个概率分布P和Q 差别的非对称性的度量。 KL散度是用来 度量使用基于Q的编码来编码来自P的样本平均所需的额外的位元数。 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。

定义如下：

$$D_{KL}(P//Q)=-\sum_{x\in X}P(x)log\frac{1}{P(x)}+\sum_{x\in X}P(x)log\frac{1}{Q(x)}$$

因为对数函数是凸函数，所以 KL散度的值为非负数。

2. JS散度(Jensen-Shannon)

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。定义如下：

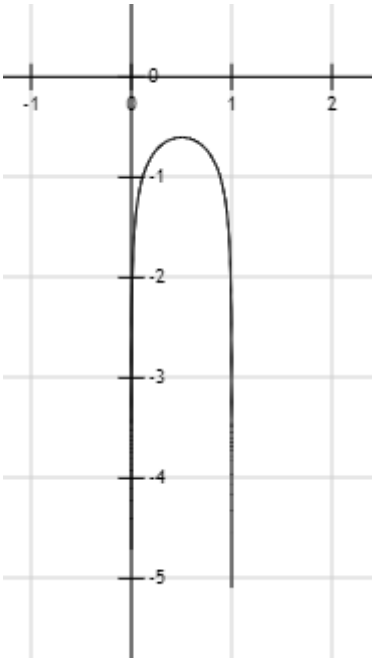
$$JS(P_1||P_2)=\frac{1}{2}KL(P_1||\frac{P_1+P_2}{2})+\frac{1}{2}KL(P_2||\frac{P_1+P_2}{2})$$

2.3 算法的收敛性

如果**G**和**D**有足够的性能，对于算法**1**，给定**G**，判别器能够达到它的最优，并且通过更新p_g来提高这个判别准则：

$$E_{x\sim p_{data}}[logD^*_G(x)]+E_{x\sim p_g}[log(1-D^*_G(x))]$$

那么p_g收敛为p_{data}



V

由于$V(G,D)$是凸函数，那么给定对应的**G**和最优的**D**，计算p_g的梯度下降更新时，如果时间足够，则p_g会收敛到P_{data}。

Reference

[1] Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.

[2] https://zhuanlan.zhihu.com/p/33752313

[3] https://libertydream.github.io/2020/04/19/一文了解对抗生成网络/