Notes about Generative Adversarial Networks

赵浩翰

2024 年 7 月 11 日

目录

§I Original GAN

1 Introduction

Generative Adversarial Networks (GANs) [1]，生成对抗网络。

生成模型，通过一个生成器 (Generator, G) 和一个鉴别器 (Discriminator, D) 的对抗性训练，G 用来估计真实数据的概率分布，D 用来估计样本来自真实数据的概率。G 的训练过程，即为最大化 D 的犯错概率。由此，G 和 D 之间形成了一个对抗性的博弈，G 努力学习真实数据分布，D 努力提升辨别真假数据分布的能力，形成一个 minimax 双人游戏。在 G 和 D 的任意函数空间中，存在唯一解 – 纳什均衡，使得 G 重现真实数据分布，D 无法区分真假数据，即概率判断为 $\frac{1}{2}$。

G 和 D 都是多层感知器 (Multilayer Perceptrons)，G 的输入是一个随机噪声，输出是一个样本，D 的输入是一个样本，输出是一个概率值。二者可以通过后向传播 (Backpropagation) 进行训练，从而无需马尔科夫链 (Markov Chain) 以及近似推断 (Approximate Inference)。

GAN 利用以下观察结果，研究生成过程中的反向传播导数：

$$\lim_{\sigma \to 0} \nabla_{\boldsymbol{x}} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})} f(x+\epsilon) = \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) \tag{1.1.1}$$

2 Adversarial Nets

对抗性模型框架最直接的应用是其生成器 G 和鉴别器 D 都是多层感知器。为了学习生成器在数据 $\boldsymbol{x}$ 上的分布 $\boldsymbol{p_g}$，先验地定义一个输入的噪音变量 $p_{\boldsymbol{z}}(\boldsymbol{z})$，并将其在数据空间上的映射表示为 $G(\boldsymbol{z};\theta_g)$，其中 G 是一个以多层感知器表示的可微函数，参数为 θ_g。同时，定义第二个多层感知器 $D(\boldsymbol{x};\theta_d)$，输出为一个标量，其中，$D\boldsymbol{x}$ 代表 $\boldsymbol{x}$ 来自真实数据而非 p_g 的概率。

训练 D 以最大化正确识别训练样本和来自生成器 G 的样本的概率，并同时训练 G 以最小化 $\log(1-D(G(\boldsymbol{z})))$。这个过程可以被看作是 D 和 G 的 minimax 二人游戏，其价值函数 $V(G,D)$ 为：

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))] \tag{1.2.1}$$

当 G 和 D 被给予足够的容量，如无参数限制时，以上训练准则足以恢复真实数据分布。在实际中，需要使用迭代的数值方法执行以上 game 的训练过程。为避免有限数据集上的过拟合风险，需要在计算上禁止在训练的内循环中优化 D 直到结束。正确的方式应该是：迭代地训练 k 步 D，然后训练 1 步 G，如此重复。此时，只要 G 的改变足够缓慢，D 将会被维持在其最优解附近。训练过程如算法 1 所示。

在实践中，1.2.1 可能会导致 G 的梯度消失。在训练的早期，G 的能力较弱，D 可以轻松的识别真、假样本，导致 $\log(1-D(G(\boldsymbol{z}))) \approx 0$，G 的梯度消失，训练速度缓慢。为了解决这个问题，可以在训练初期使用 $\log D(G(\boldsymbol{z}))$ 代替 $\log(1-D(G(\boldsymbol{z})))$，这个目标函数在 G 和 D 相互作用时有相同的固定点，但在学习早期提供了更强的梯度。

3 Theoretical Results

如前所述，真实的数据 $\boldsymbol{x}$ 服从某个特定的分布 $p_{data}(\boldsymbol{x})$，而生成器 G 隐含地为其生成的样本 $G(\boldsymbol{z})$, $\boldsymbol{z}\sim p_{\boldsymbol{z}}$ 定义了一个概率分布 p_g。因此，在训练过程中，G 的目标便是学习一个分布 p_g，使得 $p_g = p_{data}$，即两个分布的“距离”越近越好，由此产生三个问题：

1. 如何度量两个分布的“距离”？
2. $p_g(\boldsymbol{z}) = p_{data}(\boldsymbol{x})$ 是否为生成器 G 的全局最优解？
3. 上述训练算法是否可以使得 $p_g(\boldsymbol{z})$ 收敛于 $p_{data}(\boldsymbol{x})$？

Algorithm 1 Original GAN

Minibatch stochastic gradient descent training of generative adversarial nets.
The number of steps to apply to the discriminator, k, is a hyperparameter.

for number of training iterations **do**
  for k steps **do**
    Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \cdots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{data}(\boldsymbol{x})$.
    Update the discriminator by ascending its stochastic gradient:

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

  end for
  Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \cdots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  Update the generator by descending its stochastic gradient:

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

end for

3.1 KL & JS Divergence

KL 散度 (Kullback-Leibler Divergence) 和 JS 散度 (Jensen-Shannon Divergence) 是用于度量两个分布之间的“距离”的方法。

对于两个连续的概率分布 p, q，KL 散度定义为：

$$KL(p||q) = \int_{-\inf}^{\inf} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} d\boldsymbol{x} \tag{1.3.1}$$

KL 散度具有非负性，当两个分布完全相同，对于任意 $\boldsymbol{x}$，有 $p(\boldsymbol{x}) = q(\boldsymbol{x})$，此时 $\log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} = 0$，KL 散度为 0。当两个分布不完全相同，根据吉布斯不等式 (Gibbs' Inequality) 可证明 KL 散度为正数。注意到 KL 散度不满足对称性，即 $KL(p||q) \neq KL(q||p)$。

JS 散度解决了 KL 散度不对称的问题，定义为：

$$JS(p||q) = \frac{1}{2} KL(p||\frac{p+q}{2}) + \frac{1}{2} KL(q||\frac{p+q}{2}) \tag{1.3.2}$$

JS 散度为两项 KL 散度之和，当 p, q 两个分部完全相同，两项 KL 散度均为 0，JS 散度为 0。JS 散度同样满足非负性。JS 散度与 KL 散度的不同之处在于：(1) KL 散度无上界，而 JS 散度有上界 $\log 2$；(2) JS 散度满足对称性，即 $JS(p||q) = JS(q||p)$。

3.2 Global Optimality of $p_g = p_{data}$

考虑任意给定的生成器 G 下的最优鉴别器 D。

命题 1 (对于给定的生成器 G，最优鉴别器 D 为：)**.**

$$D_{\boldsymbol{G}}^{*}(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \tag{1.3.3}$$

证明：对于任意给定的 G，最优鉴别器 D 的目标是最大化价值函数 $V(D,G)$：

$$
\begin{aligned}
V(D,G) &= \int_{\boldsymbol{x}} p_{data}(\boldsymbol{x})\log(D(\boldsymbol{x}))d\boldsymbol{x} + \int_{\boldsymbol{z}} p_{\boldsymbol{z}}(\boldsymbol{z})\log(1-D(G(\boldsymbol{z})))d\boldsymbol{z} \\
&= \int_{\boldsymbol{x}} [p_{data}(\boldsymbol{x})\log(D(\boldsymbol{x})) + p_g(\boldsymbol{x})\log(1-D(\boldsymbol{x}))]d\boldsymbol{x}
\end{aligned}
\tag{1.3.4}
$$

对于任意的 $(a,b)\in\mathbb{R}^2\backslash\{0,0\}$，在 $[0,1]$ 区间上，函数 $y\to a\log(y)+b\log(1-y)$ 在点 $\frac{a}{a+b}$ 处取得最大值。同时，鉴别器无需在 $Supp(p_{data})\cup Supp(p_g)$ 之外定义。由此得证。

$\square$

由于 D 的训练目标可以视作最大化估计条件概率 $P(Y=y|\boldsymbol{x})$ 的对数似然，其中 Y 为二值随机变量，表示样本来自真实数据 ($y=1$ when $\boldsymbol{x}\sim p_{data}$) 或生成器 G ($y=0$ when $\boldsymbol{x}\sim p_g$)。因此，(1.2.1) 中的 minimax 游戏可以重新表述为：

$$
\begin{aligned}
C(G) &= \max_D V(G,D) \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1-D_G^*(G(\boldsymbol{z})))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1-D_G^*(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x})+p_g(\boldsymbol{x})}] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log \frac{p_g(\boldsymbol{x})}{p_{data}(\boldsymbol{x})+p_g(\boldsymbol{x})}]
\end{aligned}
\tag{1.3.5}
$$

定理 1

虚拟训练准则 $C(G)$ 的全局最小值当且仅当 $p_g=p_{data}$ 时取得。在该点处，$C(G)=-\log 4$。

证明：对于 $p_g=p_{data}$，$D_G^*(\boldsymbol{x})=\frac{1}{2}$ (1.3.3)，由此，根据 (1.3.5)，有 $C(G)=\log\frac{1}{2}+\log\frac{1}{2}=-\log 4$。对于任意 $p_g\neq p_{data}$，首先，将 $C(G)$ 的期望改写为积分形式：

$$
\begin{aligned}
C(G) = &\int_{\boldsymbol{x}} \left[p_{data}(\boldsymbol{x})\log\frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x})+p_g(\boldsymbol{x})} + p_g(\boldsymbol{x})\log\frac{p_g(\boldsymbol{x})}{p_{data}(\boldsymbol{x})+p_g(\boldsymbol{x})}\right]d\boldsymbol{x} \\
= &\int_{\boldsymbol{x}} \left\{p_{data}(\boldsymbol{x})\left[-\log 2+\log\frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x})+p_g(\boldsymbol{x})}+\log 2\right]\right. \\
&\left.+p_g(\boldsymbol{x})\left[-\log 2+\log\frac{p_g(\boldsymbol{x})}{p_{data}(\boldsymbol{x})+p_g(\boldsymbol{x})}+\log 2\right]\right\}d\boldsymbol{x}
\end{aligned}
\tag{1.3.6}
$$

移项可得：

$$
\begin{aligned}
C(G) = &-\log 2\int_{\boldsymbol{x}}[p_{data}(\boldsymbol{x})+p_g(\boldsymbol{x})]\,d\boldsymbol{x} \\
&+\int_{\boldsymbol{x}} p_{data}(\boldsymbol{x})\log\left[\frac{p_{data}(\boldsymbol{x})}{(p_{data}(\boldsymbol{x})+p_g(\boldsymbol{x}))/2}\right]d\boldsymbol{x} \\
&+\int_{\boldsymbol{x}} p_g(\boldsymbol{x})\log\left[\frac{p_g(\boldsymbol{x})}{(p_{data}(\boldsymbol{x})+p_g(\boldsymbol{x}))/2}\right]d\boldsymbol{x} \\
= &-2\log 2+KL(p_{data}||\frac{p_{data}+p_g}{2})+KL(p_g||\frac{p_{data}+p_g}{2}) \\
= &-\log 4+2JS(p_{data}||p_g)
\end{aligned}
\tag{1.3.7}
$$

由于 JS 散度非负，当且仅当 $p_g=p_{data}$ 时，JS 散度取最小值 0,。此时，$C(G)$ 取得全局最小值 $-\log 4$。因此，$p_g=p_{data}$ 是生成器 G 的全局最优解的充要条件。

$\square$

3.3 Convergence of Algorithm 1

命题 2. 当 G 和 D 有足够的容量，并且在 Algorithm 1 的每一步训练中，D 可以达到给定 G 下的最优状态，且 p_g 以提升以下准则为目标进行更新时，p_g 收敛于 p_{data}。

$$\mathbb{E}_{\boldsymbol{x}\sim p_{data}}\left[\log D_G^*(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log(1 - D_G^*(\boldsymbol{x}))\right] \tag{1.3.8}$$

证明： 当以上述准则进行训练时，$V(G, D)$ 可以视作 p_g 的函数 $U(p_g, D)$。由于 D 可以达到给定 G 下的最优状态，则 $U(p_g, D)$ 是 p_g 的凸函数。凸函数的上确界的子导数包括函数在最大值处的导数。即：如果 $f(x) = sup_{\alpha\in\mathcal{A}} f_\alpha(x)$ 且 $f_\alpha(x)$ 对任意 α 在 x 上是凸函数，那么 $\partial f_\beta(x) \in \partial f\ when\ \beta = argsup_{\alpha\in\mathcal{A}} f_\alpha(x)$。这相当于在给定相应的 G 的最优的 D 下计算 p_g 的梯度下降更新。$sup_D U(p_g, D)$ 对 p_g 是凸函数，且由定理 1 可得其有唯一全局最优解，因此，当 p_g 不断地、足够小幅度地更新时，p_g 收敛于 p_{data}。

□

注： 在实践中，对抗网络通过函数 $G(z; \theta_g)$ 代表了一个 p_g 的有限分布族，我们优化的是 θ_g 而不是 p_g 本身，所以证明并不适用。

§ II Conditional GAN

1 Introduction

Conditional GAN (cGAN) [2], 条件生成对抗网络。

通过在模型中添加额外信息作为条件，来引导数据的生成过程，这种条件可以基于类别标签、用于内绘的部分数据，如 [3]，甚至是来自不同模态的数据。

GAN 可以拓展至 Conditional GAN，通过对生成器和鉴别器添加额外的信息 $\boldsymbol{y}$。$\boldsymbol{y}$ 可以是任何形式的辅助信息，如类别标签或来自其他模态的数据。我们可以将 $\boldsymbol{y}$ 作为附加输入层输入到鉴别器和生成器中，从而进行调节。

在生成器中，先前的输入噪声 $p_{\boldsymbol{z}}(\boldsymbol{z})$ 和 $\boldsymbol{y}$ 被组合为一个联合的隐藏表示，而对抗性训练的框架允许各种灵活的表示方式。

在鉴别器中，$\boldsymbol{x}$ 和 $\boldsymbol{y}$ 被输入至一个判别函数中，同样，该判别函数可以是任意的多层感知器。

双方的 minmax 游戏的目标函数如式所示：

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))] \tag{2.1.1}$$

§ III A Deep Generative Approach to Conditional Sampling

本文 [4] 利用噪声外包引理，提出了一种基于条件分布和广义非参数回归函数的统一公式下条件分布样本的深度生成方法。该方法旨在学习一个条件生成器，接受从参考分布中抽取的样本，产生一个来自目标条件分布中的随机样本。条件生成器利用 KL 散度匹配合适的联合分布，并利用神经网络进行非参数估计。此方法允许预测器或响应为高维变量，且可以处理离散和连续类型的预测器和响应。文章证明了在温和的条件下，该方法中的条件发生器一致收敛至底层条件分布。

1 Introduction

为响应变量 Y 和预测器 X 之间的关系建模是统计学中的一个重要问题，这种模型可以基于新的 X 值生成 Y 的样本，也可以在给定 X 的条件下分析 Y 的变动。比如经典的回归模型，就关注于给定预测变量下响应变量的条件均值或中位数。但是，当条件分布式多模型或非对称时，条件均

值和中位数就不再能充分代表 Y 和 X 之间的关系。因此，为了全面地理解响应变量如何依赖于预测变量，我们需要学习一个条件分布，以全面地描述 Y 和 X 之间的关系。

本文提出的是一个从条件分布中抽样的非参数生成方法，称之为 Generative Conditional Distribution Sampler (GCDS)。对于一个给定的预测变量 $X = x$，GCDS 会估计一个 η 和 x 的函数 $G(\eta, x)$，η 是一个来自简单的参考分布的随机变量，如正态或均匀分布，以使得 $G(\eta, x)$ 可以模拟给定 $X = x$ 时 Y 的条件分布。本文使用神经网络来非参数地估计 G，从而使预测和响应变量都可以是高维的。

该模型的好处有，其一，支持高维的预测变量和响应变量；其二，可以处理离散和连续类型的预测变量和响应变量；其三，模型基于一个简单的参考分布学习了底层条件分布的生成函数，因而可以通过 Monte Carlo 方法获得底层条件分布统计量的估计，如条件矩和分位数；其四，GCDS 可以用于处理复杂和高维数据，如图像生成和重构；最后，本文证明了 GCDS 是一致的，因为它生成的样本弱收敛于潜在的目标条件分布。

2 Generative representation of conditional distribution

考虑一对随机向量 $(X, Y) \in \mathcal{X} \times \mathcal{Y}$，其中 X 是预测因子向量，Y 是响应变量或标签向量。对于回归问题，我们有 $Y \subseteq \mathbb{R}^q, q \geq 1$；对于分类问题，$Y$ 是有限多个标签的集合。我们假设空间 $X \subseteq \mathbb{R}^d d \geq 1$。预测因子 X 可以包含连续和分类成分。假设 $(X, Y) \sim P_{X,Y}$，具有边际分布 $X \sim P_X$ 和 $Y \sim P_Y$。用 $P_{Y|X}$ 表示在给定 X 的条件下 Y 的条件分布。对于给定的 X 的值 x，我们也将条件分布表示为 $P_{Y|X=x}$。设 η 为一个独立于 X 的随机向量，具有已知分布 P_η。例如，我们可以取 P_η 为标准多变量正态分布 $N(\mathbf{0}, \boldsymbol{I}_m)$，其中 $m \geq 1$。注意 m 不必与 q 相同，q 是 Y 的维度。

我们的目标是找到一个函数 $G : \mathbb{R}^m \times X \to Y$，使得在给定 $X = x$ 时，$G(\eta, X)$ 的条件分布与 Y 在 $X = x$ 时的条件分布相同。由于 η 与 X 独立，这相当于找到一个满足下述条件的函数 G：

$$G(\eta, x) \sim P_{Y|X=x} \quad x \in X \tag{3.2.1}$$

一个直观的问题就是：函数 G 是否存在？由最小化条件下概率论的外生噪音理论，我们可以证明 G 的存在性 (Lemma 2.1 in [4])。

G 的估计，就是将给定 $x \in \mathcal{X}$ 下 $G(\eta, x)$ 和 $P_{Y|X}$ 的条件分布进行匹配，而这一匹配，在 X 的边缘分布确定时，与将 $X, G(\eta, X)$ 和 (X, Y) 的联合分布进行匹配是等价的 (Lemma 2.2 in [4])。所以，估计 G 的方式转化为：找到一个联合分布 $P_{X,G(\eta,X)}$，使得 $P_{X,G(\eta,X)}$ 与 $P_{X,Y}$ 的联合分布匹配。

§ IV Distribution matching estimation

概括：使用 $f-$ 距离的变分形式度量两个联合分布之间的距离，具体到 KL 散度上，即最小化 KL 散度形式的变分 $f-$ 距离。由此得到 Con-GAN 的损失函数：

$$\mathcal{L}(G, D) = \mathbb{E}_{(X,\eta)\sim P_X P_\eta}[D(X, G(\eta, X))] - \mathbb{E}_{(X,Y)\sim P_{X,Y}}[\exp(D(X, Y))] \tag{4.0.1}$$

参考文献

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in Neural Information Processing Systems, 27, 2014.

[2] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[3] Ian Goodfellow, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Multi-prediction deep boltzmann machines. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 26. Curran Associates, Inc., 2013.

[4] Xingyu Zhou, Yuling Jiao, Jin Liu, and Jian Huang. A deep generative approach to conditional sampling. Journal of the American Statistical Association, 118(543):1837–1848, February 2022.