题　　目：《生成对抗网络》

选 课 班：2 班

选课序号：29

姓　　名：汪广鑫

学　　号：1120200298

学　　院：信息科学技术学院

注：原文学习并翻译

二〇二一年五月

0 Abstract

我们提出了一个新的框架，通过一个对抗的过程来估计生成模型，在此过程中我们同时训练两个模型:一个生成模型 G 捕获数据分布，和一种判别模型 D，它估计样本来自训练数据而不是 G 的概率。G 的训练程序是最大化 D 犯错的概率，这个框架对应于一个极小极大的双人游戏。在任意函数 G 和 D 的空间中，存在唯一解，G 可以重现训练数据分布，D 处处等于 1/2。在 G 和 D 由多层感知器定义的情况下，整个系统可以通过反向传播进行训练。在训练或生成样本的过程中，不需要任何马尔科夫链或展开的近似推理网络。通过对生成的样本进行定性和定量评估，实验证明了该框架的潜力。

1 Introduction

深度学习的前景是发现丰富的分层模型，它代表人工智能应用中遇到的各种数据的概率分布，如自然图像、包含语音的音频波形和自然语言语料库中的符号。到目前为止，在深度学习中最显著的成功涉及到判别模型，通常是那些将高维、丰富的感官输入映射到类标签的模型。这些惊人的成功主要是基于反向传播和 dropout 算法，使用分段线性单元，具有特别良好的梯度。由于在极大似然估计和相关策略中出现的许多难以处理的概率计算的近似性，以及由于难以在生成环境中利用分段线性单元的优点，深度生成模型的影响较小。我们提出了一种新的生成模型估计方法来克服这些困难。

在提出的对抗网框架中，生成模型与对手进行了比较:一个学习确定样本是来自模型分布还是来自数据分布的判别模型。生成模型可以被认为类似于一组伪造者，他们试图制造假币并在不被发现的情况下使用它，而判别模型则类似于警察，试图发现假币，这个游戏的竞争促使两队改进他们的方法，直到仿冒品无法从真品中辨别出来。

该框架可以生成针对多种模型的特定训练算法和优化算法，在这篇文章中，我们探讨了生成模型通过一个多层感知器传递随机噪声来生成样本的特殊情况，而判别模型也是一个多层感知器，我们把这种特殊情况称为对抗网络。在这种情况下，我们可以只使用非常成功的反向传播和 dropout 算法来训练这两个模型，并且只使用正向传播来训练生成模型的样本，不需要近似推论或马尔科夫链。

2 Related work

有潜在变量的有向图形模型的另一种选择是有潜在变量的无向图形模型，如限制玻尔兹曼机(RBMs)，深玻尔兹曼机(DBMs)及其众多变体。这些模型中的相互作用被表示为未归一化势函数的乘积，由随机变量所有状态的全局求和/积分进行归一化。这个数量(配分函数)和它的梯度是棘手的，但最琐碎的情况下，虽然他们可以由马尔可夫链蒙特卡罗(MCMC)方法估计。对于依赖于 MCMC 的学

习算法来说，混合是一个很重要的问题。

深度置信网络(DBNs)[16]是包含一个无向层和多个有向层的混合模型。虽然存在一种快速的分层近似训练准则，但 DBNs 存在与无向和有向模型相关的计算困难。

也有人提出了不近似或不限制对数似然的替代标准，如分数匹配和噪声对比估计(NCE)，这两种方法都要求所学习的概率密度被解析指定为一个归一化常数。请注意，在许多具有多层潜在变量(如 DBNs 和 DBMs)的有趣生成模型中，甚至不可能导出可处理的非规范化概率密度，一些模型，如去噪自动编码器[30]和收缩自动编码器的学习规则非常类似于分数匹配应用于 RBMs。在 NCE 中，与本文一样，使用了判别训练准则来拟合生成模型。然而，生成模型本身用于从固定噪声分布的样本中区分生成的数据，而不是拟合一个单独的判别模型。由于 NCE 使用一个固定的噪声分布，当模型学习到即使是在观察变量的一个小子集上的一个近似正确的分布之后，学习速度也会显著减慢。

最后，一些技术不涉及明确定义概率分布，而是训练生成机器从期望的分布中抽取样本，这种方法的优点是可以通过反向传播来训练这些机器。近期主要的工作包括生成随机网络(GSN)框架：它扩展了广义去噪自动编码器：两者都可以看作是定义一个参数化的马尔科夫链，即一个人学习机器的参数，执行一个步骤的生成马尔科夫链。与 GSNs 相比，对抗网的采样不需要马尔科夫链，由于反求网络在生成过程中不需要反馈环，所以它们能够更好地利用分段线性单元，这提高了反向传播的性能，但在使用反馈环时存在无限制激活的问题。通过反向传播训练生成机器的最新例子包括自动编码变分贝叶斯和随机反向传播。

3 Adversarial nets

当模型都是多层感知器时，对抗性建模框架最容易应用。为了学习生成器在数据 x 上的分布 p_g ,我们定义了一个输入噪声变量 $\mathrm{p}_z(z)$，$G(z,\theta_g)$ 表示将噪声变量映射到数据空间, G 是一个可微函数，表示为一个参数为 θ_g 的多层感知器。我们还定义一个多层感知器 $D(x;\theta_d)$ 输出一个标量，D(x)表示 x 来自数据集而不是 p_g 的概率。我们训练 D 最大限度地将正确的标签分配给训练样本和来自 G 的样本的概率，我们同时训练 G，使得 log(1 - D(G(z))) 最小化。

换句话说，D 和 G 玩了一个具有值函数 V (G,D)的二人极大极小博弈:

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

在下一节中，我们将对对抗网进行理论分析，主要说明当 G 和 D 具有足够的容量时，训练准则允许恢复数据生成分布，例如在非参数极限下。请参见图 1，其中对该方法进行了不太正式的、更具教育性的解释。在实践中，我们必须使用迭代的数值方法来实现游戏。优化完成内环的训练在计算上是禁止的，对于有限

的数据集会导致过度拟合。相反，我们在优化 D 的 k 个步骤和优化 G 的一个步骤之间交替进行，只要 G 变化足够慢，D 就会保持在其最优解附近，这种策略类似于 SML/PCD：训练从一个学习步骤到下一个学习步骤保持来自马尔可夫链的样本，该过程在算法 1 中正式给出。

在实际应用中，公式(1)可能无法为 G 提供足够的梯度来学习。在学习的早期，当 G 较差时，D 可以很有信心地拒绝样本，因为它们与训练数据明显不同。在这种情况下，log(1 - D(G(z)))饱和，与其训练 G 去最小化 log(1 - D(G(z))不如训练 G 去最大化 log D(G(z))这一目标函数的结果与动态函数相同，但在学习中提供了更强的学习效果。

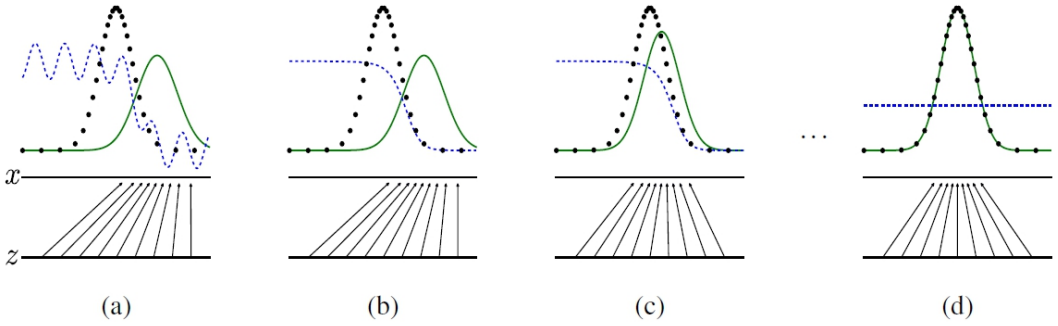

图 1：通过同时更新判别分布（D，蓝色，虚线）来训练生成对抗网络，以便区分生成数据的分布（黑色，虚线）p_x 的样本与生成分布 $p_g(G)$ 的样本之间的区别（绿色， 实线）。 下方的水平线是从中采样 z 的域，在这种情况下是均匀的。 上面的水平线是 x 的域的一部分。向上的箭头显示映射 x = G（z）如何将非均匀分布 p_g 施加到转换后的样本上。G 在高密度区域收缩，在 p_g 低密度区域膨胀。（a）考虑一个接近收敛的对抗对： p_g 与 p_{data} 相似，D 是部分准确的分类器。（b）在算法的内部循环中，训练 D 来区分数据中的样本，收敛到 $D^*(x)=\frac{P_{data}(x)}{P_{data}(x)+P_g(x)}$。（c）在更新 G 之后，D 的坡度已引导 G（z）流向更可能被归类为数据的区域。（d）经过几步训练后，如果 G 和 D 具有足够的能力，则它们将达到无法提高的点，因为 $P_g = P_{data}$。 判别器无法区分两个分布，即 D（x）= 0.5。

4 Theoretical Results

生成器 G 隐式地定义了一个概率分布 p_g，作为 $z \sim P_z$ 得到的样本 G(z)的分布。因此，如果有足够的能力和训练时间，我们将使用算法 1 来收敛 P_{data}。本节的结果是在非参数的情况下得到的，例如我们通过研究概率密度函数空间的收敛性来表示一个具有无限容量的模型。我们将在第 4.1 节中展示这个极大极小游戏

对于 $P_g = P_{data}$ 有一个全局最优。我们将在 4.2 节中展示算法 1 对公式(1)进行了优化，从而得到了我们想要的结果。

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.

4.1 Global Optimality of $P_g = P_{data}$

首先考虑任意给定发生器 G 的最优鉴别器 D。

命题 1：对于固定的 G，最优鉴别器 D 为：

$$D^*_G(x) = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)} \tag{2}$$

证明：给定任何生成器 G，鉴别器 D 的训练准则是使量 V（G，D）最大化

$$\begin{aligned} V(G,D) &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_{\boldsymbol{z}} p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz \\ &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx \end{aligned} \tag{3}$$

对于任意的 $(a,b) \in \mathbb{R}^2 \setminus \{0,0\}$，函数 $y \to a\log(y) + b\log(1-y)$ 达到最大值在点 $\frac{a}{a+b}$.鉴别器不需要在 $\text{Supp}\left(p_{\text{data}}\right) \cup \text{Supp}\left(p_g\right)$ 之外定义，包括证明。

请注意 D 的训练目标可以解释为最大化对数似然率以估计条件概率 P(Y = y | x)，其中 Y 表示 x 是来自 p_{data}（y = 1）还是来自 p_g（y = 0）。 公式(1)中的 minimax 博弈现在可以将公式(1)改写为：

$$
\begin{aligned}
C(G) &= \max_D V(G,D) \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right]
\end{aligned} \tag{4}
$$

定理 1：当且仅当 $P_g = P_{data}$ 时，得到虚拟训练准则 C(G)的全局最小值。此时，C(G) 等于 -log4。

证明：对于 $P_g = P_{data}, D^*_G(x) = \frac{1}{2}$，(考虑公式 2).因此，通过检查公式 4 在 $D^*_G(x) = \frac{1}{2}$ 处，我们发现 $C(G) = \log\frac{1}{2} + \log\frac{1}{2} = -\log 4$，为了找到 C(G)的最佳可能值仅对于 $P_g = P_{data}$，可以发现：

$$
\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[-\log 2] + \mathbb{E}_{\mathbf{x}\sim p_g}[-\log 2] = -\log 4
$$

通过从 $C(G) = V(D^*_G, G)$ 中减去该表达式，我们得到：

$$
C(G) = -\log(4) + KL\left(p_{\text{data}} \| \frac{p_{\text{data}} + p_g}{2}\right) + KL\left(p_g \| \frac{p_{\text{data}} + p_g}{2}\right) \tag{5}
$$

其中 KL 是 Kullback-Leibler 散度。 我们在前面的表达式中认识到模型的分布与数据生成过程之间的詹森-香农差异：

$$
C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right) \tag{6}
$$

由于两个分布之间的詹森-香农散度总是非负的，并且只有在它们相等时才为零，因此我们证明了 $C^* = -\log(4)$ 是 C（G）的全局最小值，唯一的解是 $P_g = P_{data}$，即生成模型完美地复制了数据生成过程。

4.2 Convergence of Algorithm 1

命题 2：如果 G 和 D 具有足够的容量，并且在算法 1 的每个步骤中，允许鉴别器达到其最佳给定 G，并更新 P_g 以改进准则

$$
\mathbb{E}_{\mathbf{x}\sim p_{\text{dana}}}\left[\log D_G^*(x)\right] + \mathbb{E}_{\mathbf{x}\sim p_g}\left[\log\left(1 - D_G^*(x)\right)\right]
$$

那么，P_g 收敛于 P_{data}

证明：按照上述准则，将 $V(G,D) = U(P_g, D)$ 视为 P_g 的函数。 注意，$U(p_g, D)$ 在 P_g 中是凸的。 凸函数的极值的子导数包括在达到最大值时的函数导数。 换句

话说，如果 $f(x)=\sup_{\alpha\in A} f_\alpha(x)$ 且 $f_\alpha(x)$ 在 x 中对于每个 α 是凸的，则当 $\beta=\arg\sup_{\alpha\in A} f_\alpha(x)$ 时，$\partial f_\beta(x)\in\partial f$。这等效于在给定相应 G 的情况下，在最佳 D 下计算 P_g 的梯度下降更新。$\sup_D U(p_g,D)$ 在 P_g 中是凸的，具有在定理 1 中证明的唯一全局最优值，因此 P_g 的更新足够小，P_g 收敛到 P_x，得出证明。

实际上，对抗网络通过函数 $G(z,\theta_g)$ 表示有限的 P_g 分布族，我们优化 θ_g 而不是 P_g 本身。 使用多层感知器定义 G 会在参数空间中引入多个临界点。 但是，多层感知器在实践中的出色表现表明，尽管缺乏理论上的保证，它们还是可以使用的合理模型。

5 Experiments

我们在包括 MNIST, the Toronto Face Database (TFD),和 CIFAR-10 一系列数据集上训练了对抗网络。生成网络使用 rectifier linear and sigmoid 两种激活函数，而判别器使用 maxout 激活。应用 dropout 训练判别器网络。虽然我们的理论框架允许在生成器的中间层使用 dropout 和其他噪声，但我们只使用噪声作为生成器网络最底层的输入。

我们通过对 G 生成的样本拟合高斯 Parzen 窗口，并报告该分布下的对数似然，来估计 P_g 下测试集数据的概率，高斯分布的参数 σ 是通过交叉验证的验证集。结果见表 1。这种估计可能性的方法有一些高的方差，在高维空间中表现不好，但它是我们所知道的最好的方法。能抽样但不能估计可能性的进化能直接激发对如何评价这些模型的进一步研究。

Model	MNIST	TFD
DBN [3]	138 ±2	1909 ±66
Stacked CAE [3]	121 ±1.6	**2110** ± **50**
Deep GSN [6]	214 ±1.1	1890 ±29
Adversarial nets	**225** ± **2**	**2057** ± **26**

表 1：基于 Parzen 窗口的对数似然估计。MNIST 上报告的数字是测试集上样本的均值对数似然，并通过示例计算出均值的标准误。在 TFD 上，我们计算了数据集折叠的标准误差，并使用每个折叠的验证集选择了不同的标准误差。在 TFD 上，对每个折叠进行交叉验证，并计算每个折叠的平均对数似然率。对于 MNIST，我们将其与数据集的实值（而不是二进制）版本的其他模型进行比较。

在图 2 和图 3 中，我们展示了训练后从生成器网络中抽取的样本。虽然我们不能断言这些样本比现有方法生成的样本更好，但我们认为这些样本至少与文献中更好的生成模型具有竞争力，并突出了对抗性框架的潜力。

图 2：可视化来自模型的样本。 最右列显示了邻近样本的最近训练示例，以证明该模型没有存储训练集。 样本是公平的随机抽签，并非精心挑选。 与大多数其他深层生成模型的可视化不同，这些图像显示了来自模型分布的实际样本，而不是给定隐藏单元样本的条件均值。此外，这些样本是不相关的，因为采样过程不依赖于马尔可夫链混合。 a）MNIST b）TFD c）CIFAR-10（全连接模型）d）CIFAR-10（卷积鉴别器和“反卷积”生成器）

	Deep directed graphical models	Deep undirected graphical models	Generative autoencoders	Adversarial models
Training	Inference needed during training.	Inference needed during training. MCMC needed to approximate partition function gradient.	Enforced tradeoff between mixing and power of reconstruction generation	Synchronizing the discriminator with the generator. Helvetica.
Inference	Learned approximat e inference	Variational inference	MCMC-base d inference	Learned Approxim ate inference
Sampling	No difficulties	Requires Markov chain	Requires Markov chain	No difficulties
Evaluating $p(x)$	Intractable,may be approximated with AIS	Intractable, may be approximated with AIS	Not explicitly represented, may be approximated with Parzen density estimation	Not explicitly represented, may be approximated with Parzen density estimation
Model design	Nearly all models incur extreme difficulty	Careful design needed to ensure multiple properties	Any differentiable function is theoreticall y permitted	Any differentiable function is theoreticly permitted

表 2：生成建模的挑战：对于涉及模型的每个主要操作，采用不同方法进行深度生成建模所遇到的困难的摘要。

图 3：通过在完整模型的 z 空间中的坐标之间进行线性插值而获得的数字。

6 Advantages and disadvantages

与以前的建模框架相比，此新框架具有优点和缺点。缺点主要是没有 $p_g(x)$ 的明确表示，并且在训练过程中 D 必须与 G 很好地同步（特别是，在没有更新 D 的情况下，G 不能被过多训练，以避免出现“ Helvetica 场景” 其中，G 将太多的 z 值折叠成与 x 相同的值，以至于没有足够的多样性来建模 P_{data} ），就像必须在学习步骤之间保持 Boltzmann 机器的负链更新一样。 优点是不再需要马尔可夫链，仅使用 backprop 即可获得梯度，在学习过程中无需进行推理，并且可以将多种功能集成到模型中。 表 2 总结了生成对抗网络与其他生成建模方法的比较。

前述优点主要是计算上的。 对抗模型还可以从生成器网络中获得一些统计上的优势，该生成器网络不直接使用数据示例进行更新，而仅使用流经鉴别器的梯度进行更新。 这意味着输入的组成部分不会直接复制到生成器的参数中。 对抗网络的另一个优点是它们可以表现出非常尖锐的分布，甚至是简并的分布，而基于马尔可夫链的方法则要求分布有些模糊，以使链能够在模式之间进行混合。

7 Conclusions and future work

这个框架允许许多简单的扩展：

(1) 将 c 作为 G 和 D 的输入，可以得到条件生成模型 p(x | c)。

(2) 学习近似推理：可以利用一个辅助网络在给定 x 时来预测 z。这与 wake-sleep 算法训练的推理网络类似，但具有在生成器网络完成训练后，可以对固定生成器网络进行推理网络训练的优点。

(3) 通过训练一系列共享参数的条件模型，可以近似地对所有条件 $p(x_S \mid x_{\not S})$ 进行建模，其中 s 是 x 指标的子集。本质上，我们可以使用对抗网来实现确定性 MP-DBM[11]的随机扩展。

(4) 半监督学习:当有限的标记数据可用时，鉴别器或推理器的特性可能会降低分类器的性能。

(5) 效率改进:在培训过程中，通过划分更好的方法来协调 G 和 D，或者确定更好的 z 分布，可以大大加快训练的速度。

本文证明了对抗性建模框架的可行性，表明这些研究方向是有用的。