生成对抗网络

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair[y], Aaron Courville, Yoshua Bengio[z]
Departement´ d'informatique et de recherche operationnelle´
Universite´ de Montreal´
Montreal,´ QC H3C 3J7

摘 要

通过对抗过程，我们提出了一个估计生成模型的新的框架，在新的框架中，我们同时训练两个模型：捕获数据分布的生成模型 G，和估计样本来自训练数据而不是生成器 G 的概率的判别模型 D。G 的训练过程是使 D 犯错误的概率最大化。这个框架相当于最小化最大的双人博弈。在任意函数 G 和 D 的空间中，存在唯一解，其中，G 恢复训练数据分布，D 处处都是 1/2。在 G 和 D 由多层感知机定义的情况下，整体系统可以用反向传播训练。在训练或者生成模型期间，不需要任何的马尔科夫链或者展开近似推理网络。实验通过对生成样本的定性和定量评估来展示框架的潜力。

1 介绍

深度学习的任务是发现丰富的层次模型[2]，表达在人工智能领域中各种数据的概率分布，比如自然图像、语音的音频波形和自然语言语料库的符号。到目前为止，深度学习中最显著的成功之一的框架是判别模型。通常他们将高维丰富的感知器输入映射到类标签上[14,22]。这些成功主要基于反向传播和丢弃算法实现的，特别是具有良好梯度的分段线性单元[19,9,10]。由于极大似然估计和相关策略中出现了难以解决的概率计算的困难，以及很难利用在生成上下文中使用分段线性单元获得的好处，这些原因导致了深度生成模型的影响力很小。所以，我们提出了一个新的生成估计模型，分步处理这些困难。

提到的对抗式网络框架中，生成模型对应着一个对手：学习判别样本是来自模型分布还是数据分布的判别模型。生成模型可以被认为是一个造假团队，他们试图制造假币而且在不被发现的情况下使用它，而判别模型被看作是警察，试图检测假币。游戏中的两个竞争者改进他们的方法直到真假难辨。

Jean Pouget-Abadie 将从 Ecole Polytechnique 参观蒙特利尔大学.

Sherjil Ozair 正在德里印度理工学院参观蒙特利尔大学.

Yoshua Bengio 是 CIFAR 的高级研究员.

所有代码和超参数均可从 http://www.github.com/goodfeli/adversarial 获得

这个框架针对各类模型和优化算法可以提供特定的训练算法。在这篇论文中，我们探讨了将生成模型通过随机噪声传输到多层感知机生成样本的特例，同时判别模型也是由多层感知机实现的。我们称这个特例为对抗网络。在这个例子中，我们可以使用非常成熟的反向传播和丢弃算法[17]训练两个模型，生成模型训练样本是仅使用前向传播算法，不需要马尔科夫链和近似推理。

2 相关工作

具有潜在变量的定向图形模型的替代方案是具有潜变量的无向图形模型，例如受限制的玻尔兹曼机器（RBM）[27,16]，深玻尔兹曼机器（DBM）[26]及其众多变体。 这些模型中的相互作用表示为非标准化势函数的乘积，通过随机变量的所有状态的全局总结/积分进行归一化。 这个数量（分区函数）及其梯度对于除了最平凡的实例之外的所有实例都是难以处理的，尽管它们可以通过马尔可夫链蒙特卡罗（MCMC）方法来估计。 混合对于依赖 MCMC 的学习算法提出了一个重要问题[3,5]。

深信念网络（DBN）[16]是包含单个无向层和多个定向层的混合模型。 虽然存在快速近似分层训练标准，但 DBN 会引起与无向和定向模型相关的计算困难。

不接近或约束对数似然的替代标准也被提出，例如得分匹配[18]和噪声对比估计（NCE）[13]。这两者都要求将学习的概率密度分析地指定为归一化常数。请注意，在许多具有多层潜在变量（例如 DBN 和 DBM）的有趣生成模型中，甚至不可能得出易处理的非标准化概率密度。某些模型（如去噪自动编码器[30]和收缩自动编码器）的学习规则与应用于 RBM 的分数匹配非常相似。在 NCE 中，与本研究一样，采用判别训练标准来拟合生成模型。然而，生成模型本身不是用于拟合单独的判别模型，而是用于将生成的数据与样本区分为固定的噪声分布。因为 NCE 使用固定的噪声分布，所以在模型在一小部分观察到的变量上学习了大致正确的分布后，学习速度显着减慢.

最后，一些技术不涉及明确定义的概率分布，而是训练生成器从所需分布中抽取样本。这种方法的优点是可以将这种生成器设计成通过反向传播进行训练。最近在该领域的突出工作包括生成随机网络（GSN）框架[5]，它扩展了广义去噪自动编码器[4]：两者都可以看作是定义参数化马尔可夫链，即学习机器的参数执行生成马尔可夫链的一步。与 GSN 相比，对抗性网络框架不需要马尔可夫链进行采样。因为对抗网络在生成期间不需要反馈回路，所以它们能够更好地利用分段线性单元[19,9,10]，这提高了反向传播的性能，但是当在反馈回路中使用时存在无界激活的问题。最近通过反向传播训练生成器的例子包括最近关于自动编码变分贝叶斯[20]和随机反向传播[24]的工作。

3 对抗网络

当模型是多层感知器时，对抗建模框架能被最直接应用。 为了在数据 x 上学习生成器的分布 p_g ，我们在输入噪声变量$p_{\boldsymbol{z}}(\boldsymbol{z})$上定义先验，然后将数据空间的映射表示为$G(\boldsymbol{z};\theta_g)$，其中 G 是由多层感知器表示的可微函数,参数为 θ_g。 我们还定义了第二个多层感知器 $D(\boldsymbol{x};\theta_d)$，它输出一个标量。 $D(\boldsymbol{x})$表示 x 来自数据而不是 p_g 的概率。 我们训练 D 以最大化为 G 训练样本和样本分配正确标签的概率。我们同时训练 G 以最小化 $\log(1-D(G(\boldsymbol{z})))$:

换句话说，D 和 G 使用值函数 V（G; D）进行以下双人最小最大值游戏:

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

在下一节中，我们提出了对抗网络的理论分析，基本上表明训练标准允许人们恢复数据生成分布，因为 G 和 D 被赋予足够的容量（在非参数限制中）。有关该方法的不太正式，更具教学意义的解释，请参见图 1。在实践中，我们必须使用迭代的数值方法来实现游戏。在训练的内循环中优化 D 到完成在计算上是不可行的，并且在有限数据集上将导致过度拟合。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。这导致 D 保持接近其最优解，只要 G 变化足够慢。这种策略类似于 SML / PCD [31,29]训练将马尔可夫链中的样本从一个学习步骤维持到下一个学习步骤的方式，以避免作为学习内循环的一部分在马尔可夫链中燃烧。该过程在算法 1 中正式呈现。

在实践中，等式 1 可能无法为 G 学习提供足够的梯度。 在学习初期，当 G 很差时，D 可以高度自信地拒绝样本，因为它们与训练数据明显不同。 在这种情况下，$\log(1 - D(G(\boldsymbol{z})))$饱和。 我们可以训练 G 以最大化$D(G(\boldsymbol{z}))$，而不是训练 G 以最小化$\log(1 - D(G(\boldsymbol{z})))$。 该目标函数导致 G 和 D 动力学的相同固定点，但在学习早期提供更强的梯度。

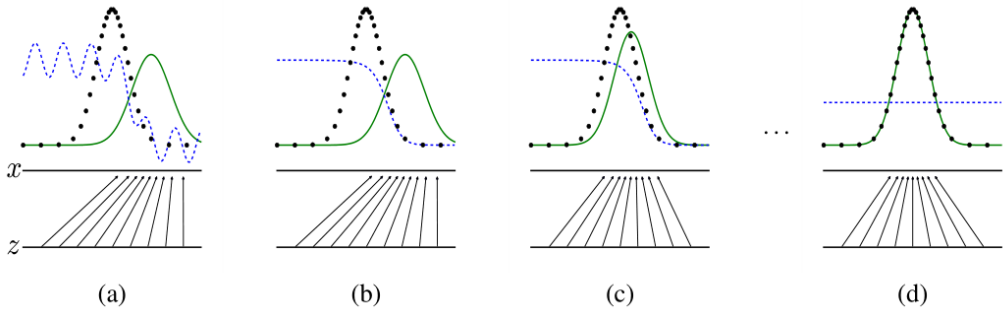

图 1：生成对抗网络通过同时更新判别分布（D，蓝色，虚线）进行训练，以便区分来自数据生成分布（黑色，虚线） $p_{\boldsymbol{x}}$ 的样本与生成分布p_g (G)的样本 （绿色，实线）。 下面的水平线是从中采样 z 的域，在这种情况下是均匀的。 上面的水平线是 x 域的一部分。 向上箭头表示映射 x = G（z）如何在变换样本上施加非均匀分布 p_g。 G 在高密度区域收缩，在低密度区域扩展。 （a）考虑靠近收敛的对抗对：p_g 类似于 p_{data}，D 是部分精确的分类器。（b）在算法的内环中，训练 D 以区分样本和数据，收敛到 D*(x)= $\frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})+p_g(\boldsymbol{x})}$。（c）在更新 G 之后，D 的梯度引导 G（z）流向更可能被分类为数据的区域。 （d）经过几个步骤的训练后，如果 G 和 D 有足够的容量，它们将达到两个都无法改善的点，因为 p_g = p_{data}。 鉴别器不能区分两个分布，即$D(\boldsymbol{x}) = \frac{1}{2}$

4 理论结果

生成器 G 隐含地将概率分布 p_g 定义为当 $\boldsymbol{z} \sim p_{\boldsymbol{z}}$ 时获得的样本 G（z）的分布。 因此，如果给定足够的容量和训练时间，我们希望算法 1 收敛到 p_{data} 的良好预期。 该部分的结果是在非参数设置中完成的，例如， 我们通过研究概率密度函数空间中的收敛来表示具有无限容量的模型。

我们将在 4.1 节中说明这个 minimax 游戏对于 p_g = p_{data} 具有全局最优。 然后我们将在 4.2 节中展示算法 1 优化等式 1，从而获得所需的结果。

算法 1 生成对抗网络的 minibatch 随机梯度下降训练。判别器的训练步数，k，是一个超参数。在我们的试验中使用 k=1，使消耗最小。

for number of training iterations **do**（对于训练的迭代次数）

for k steps **do**（共 K 步）

λ在噪声先验概率分布为 $p_g(\boldsymbol{z})$ 中抽取 M 个样本 $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$。

λ在数据生成分布为 $p_{\text{data}}(\boldsymbol{x})$ 中抽取 M 个样本 $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$。

通过提升其随机梯度来更新鉴别器:

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

end for

· λ在噪声先验概率分布为 $p_g(\boldsymbol{z})$ 中抽取 M 个样本 $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$。

· λ通过随机梯度下降更新判别器

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

end for

基于梯度的更新可以使用任何标准的基于梯度的学习准则。我们在实验中使用了动量准则。.

4.1 全局最优 $p_g = p_{\text{data}}$

首先任意给生成器 G，我们要考虑最优判别器 D。

命题 1：固定 G，最优判别器 D 为：

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \tag{2}$$

证明：给定任意生成器 G，判别器 D 的训练标准为最大化目标函数 V(G,D)

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

对于任意的 $(a,b) \in \mathbb{R}^2 \setminus \{0,0\}$，函数 $y \to a\log(y) + b\log(1-y)$ 在[0,1]中的 $\frac{a}{a+b}$ 处达到最大值。无需在 $Supp(p_{\text{data}}) \cup Supp(p_g)$ 外定义判别器，证毕。

注意到，判别器 D 的训练目标可以看作为条件概率 P(Y=y|x)的最大似然估计，当 y=1 时，x 来自于 p_{data}；当 y=0 时，x 来自 p_g。公式 1 中的极小化极大问题可以变形为

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

定理一：当且仅当 $p_g = p_{\text{data}}$ 时，C(G)达到虚拟训练标准的全局最小。此时，C(G)的值为 −log4。

证明：$p_g = p_{\text{data}}$ 时，$D_G^*(\boldsymbol{x}) = \frac{1}{2}$（公式 2）。再根据公式 4 的 $D_G^*(\boldsymbol{x}) = \frac{1}{2}$ 可得，$C(G) = \log \frac{1}{2} + \log \frac{1}{2} = -\log 4$。为了看当 $p_g = p_{\text{data}}$ 时，C(G)是否是最优的，观测到：

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

然后从 $C(G) = V(D_G^*, G)$ 减去上式，可得：

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

其中 KL 为 Kullback-Leibler 散度，也称相对熵。我们在之前的表达式中识别出了模型的判别和数据生成过程之间的 Jensen-Shannon 散度：

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

由于两个分布之间的 Jensen-Shannon 散度总是非负的，并且当两个分布相等时，值为 0。因此 $C^* = -\log(4)$ 为 C(G)的全局极小值，并且唯一解为 $p_g = p_{\text{data}}$，即生成模型能够完美的复制数据的生成过程。

4.2 算法 1 的收敛性

命题二：如果 G 和 D 有足够的性能，对于算法 1 中的每一步，给定 G 时，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

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

则 p_g 收敛为 p_{data}。

证明：如上述准则，考虑 $V(G, D) = U(p_g, D)$ 为关于 p_g 的函数。注意到 $U(p_g, D)$ 是关于 p_g 的凸函数。该凸函数上确界的二次导数包括达到最大值处的该函数的导数。换句话说，如果 f(x)= $\sup_{\alpha \in \mathcal{A}} f_\alpha(x)$，且对于每一个 α，$f_\alpha(x)$ 是凸函数，那么如果 $\beta = \arg\sup_{\alpha \in \mathcal{A}} f_\alpha(x)$，则 $\partial f_\beta(x) \in \partial f$。这等价于给定对应的 G 和最优的 D，计算 p_g 的梯度下降更新。如定理 1 所证明，$\sup_D U(p_g, D)$ 是关于 p_g 的凸函数且有唯一的全局最优解，因此，当 p_g 的更新足够小时，p_g 收敛到 p_x，证毕。

实际上，对抗网络通过函数 $G(\boldsymbol{z}; \theta_g)$ 表示 p_g 分布的有限族，并且我们优化 θ_g 而不是 p_g 本身。使用一个多层感知机来定义 G 在参数空间引入了多个临界点。然而，尽管缺乏理论证明，但在实际中多层感知机的良好表现能表明了这是一个合理的模型。

5 实验

我们训练了对抗网络的一系列数据集，包括 MNIST [23]，多伦多人脸数据库（TFD）[28]和 CIFAR-10 [21]。 生成网络使用线性激活[19,9]和 S 形激活的混合激活层，而判别网络使用 Maxout [10]激活。随机丢弃算法（Dropout）[17]用于训练判别网络。 虽然我们的理论框架允许在生成器的中间层使用压差和其他噪声，但我们使用噪声作为生成网络最底层的输入。.

我们通过将高斯 Parzen 窗口拟合到用 G 生成的样本并在该分布下报告对数似然来估计测试集数据在 p_g 下的概率。 参数σ

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

表 1：Parzen 基于窗口的对数似然估计。 在 MNIST 上报告的数字是测试集上样本的平均对数似然，并且在示例中计算的平均值的标准差。 在 TFD 上，我们计算了数据集折叠的标准差，使用每个折叠的验证集选择不同的标准差。 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。 对于 MNIST，我们将与其他数据集的实值（而不是二进制）版本进行比较。

通过验证集上的交叉验证获得高斯算子。 该程序在 Breuleux 等人的研究中[8]引入并用于各种生成模型，其确切的可能性是不易处理的[25,3,5]。 结果报告在表 1 中。这种估计概率的方法具有稍高的方差，并且在高维空间中表现不佳，但它是我们所知的最佳方法。 可以采样但不能估计概率的生成模型的进步直接激发了对如何评估此类模型的进一步研究。

在图 2 和图 3 中，我们显示了训练后从生成网络中抽取的样本。 虽然我们没有声称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能与文献中更好的生成模型竞争，并突出了对抗框架的潜力。

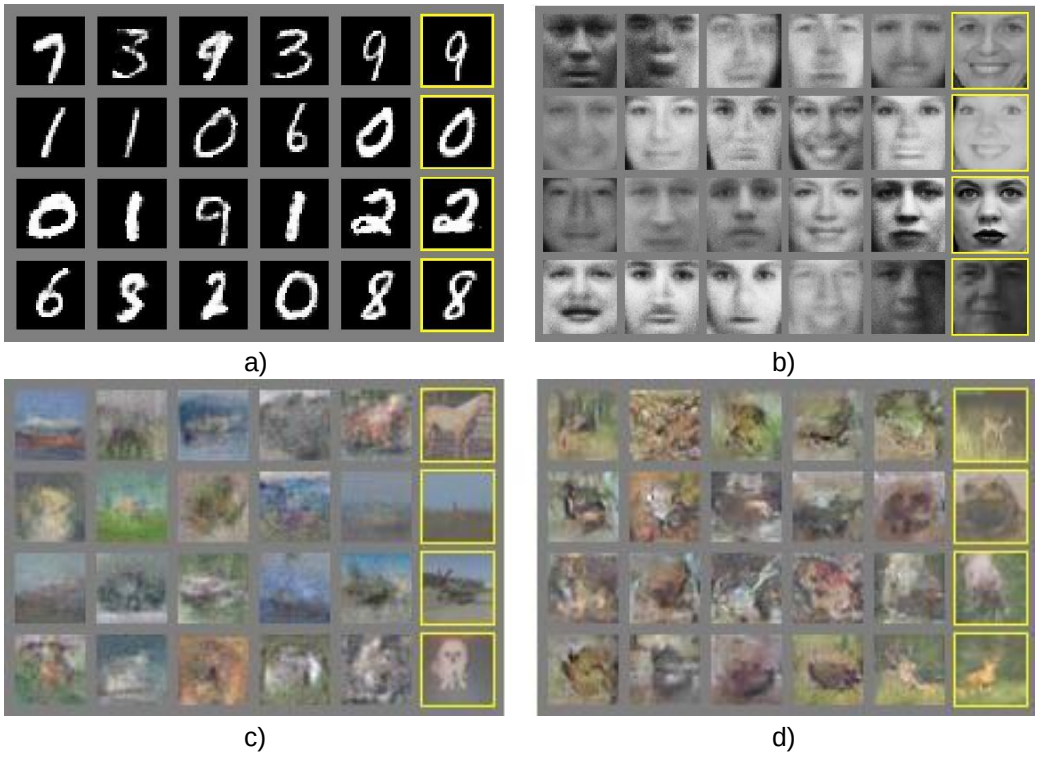

图 2：来自模型的样本的可视化。 最右边的列显示最近的相邻样本的训练示例，以证明模型没有记住训练集。 演示图像是公平的随机抽取，而不是挑选。 与深度生成模型的大多数其他可视化不同，这些图像显示来自模型分布的实际样本，而不是给定隐藏单元样本的条件均值。 而且，这些样品是不相关的，因为取样过程不依赖于马尔可夫链混合。 a）MNIST b）TFD c）CIFAR-10（完全连接模型）d）CIFAR-10（卷积判别器和“反卷积”生成器）。

图 3：通过在完整模型的 z 空间中的坐标之间线性插值获得的数字。

	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 approximate inference	Variational inference	MCMC-based inference	Learned approximate 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 theoretically permitted	Any differentiable function is theoretically permitted

表 2：生成建模中的挑战：对涉及模型的每个主要操作的深度生成建模的不同方法遇到的困难的总结。

6 优缺点

与以前的建模框架相比，这个新框架具有优点和缺点。缺点主要在于没有$p_g(\boldsymbol{x})$的明确表示，并且 D 在训练期间必须与 G 很好地同步（特别是，在不更新 D 的情况下，G 不得过多训练，以避免“Helvetica 场景”“其中 G 将太多的 z 值折叠到 x 的相同值以具有足够的多样性来模拟 p_{data}”，就像 Boltzmann 机器的负链必须在学习步骤之间保持最新一样。优点是永远不需要马尔可夫链，只有反向传播用于获得渐变，学习期间不需要推理，并且可以将多种功能合并到模型中。表 2 总结了生成对抗网络与其他生成建模方法的比较。

上述优点主要是在计算上。对抗模型也可能从生成网络中获得一些统计优势，而不是直接用数据示例更新，而是仅通过流经判别器的梯度。这意味着输入的组件不会直接复制到生成器的参数中。对抗性网络的另一个优点是它们可以表示非常尖锐，甚至是较为初始的分布，而基于马尔可夫链的方法要求分布有些模糊，以便链能够在模式之间混合.

7 结论和未来研究方向

该框架允许许多直接的扩展:

1. 条件生成模型$p(\boldsymbol{x} \mid \boldsymbol{c})$可以通过将 c 作为 G 和 D 的输入来获得。

2. 给定 x，可以通过训练一个辅助的网络来学习近似推理，达到预测 z 的目的。这和 wake-sleep 算法[15]训练出的推理网络类似，但是它具有一个优势，就是在生成器训练完成后，这个推理网络可以针对固定的生成器进行训练。

3.可以通过训练共享参数的条件模型族来近似地对所有条件概率 $p(\boldsymbol{x}_S \mid \boldsymbol{x}_{\not S})$ 进行建模，其中 S 是 x 下标的子集。本质上，可以使用敌对网络来实现确定性 MP-DBM[11]的随机扩展。

4.半监督学习：当有限的标签数据可用时，来自鉴别器或推理网络的特征可以改善分类器的性能。

5.改善效率：通过为协调 G 和 D 设计更好的方法，或在训练期间确定更好的分布来采样 z，能够极大的加速训练。

本文已经展示了对抗模型框架的可行性，表明这些研究方向是有用的。

致 谢

我们要感谢 Patrice Marcotte，Olivier Delalleau，Kyunghyun Cho，Guillaume Alain 和 Jason Yosinski 的有益讨论。 Yann Dauphin 与我们分享了他的 Parzen 窗口评估代码。 我们要感谢 Pylearn2 [12]和 Theano [7,1]的开发人员，特别是 Fred'eric'Bastien，他们专门为该项目提供了一个 Theano 功能。 Ar-naud Bergeron 为 LATEX 排版提供了急需的支持。 我们还要感谢 CIFAR 和加拿大研究主席的资助，以及 Compute Canada 和 Calcul Quebec'提供的计算资源。 Ian Goodfellow 得到 2013 年谷歌深度学习奖学金的支持。 最后，我们要感谢 Les Trois Brasseurs 激发我们的创造力.

参考文献

[1] Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I. J., Bergeron, A., Bouchard, N., and Bengio, Y. (2012). Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

[2] Bengio, Y. (2009). Learning deep architectures for AI. Now Publishers.

[3] Bengio, Y., Mesnil, G., Dauphin, Y., and Rifai, S. (2013a). Better mixing via deep representations. In ICML'13.

[4] Bengio, Y., Yao, L., Alain, G., and Vincent, P. (2013b). Generalized denoising auto-encoders as generative models. In NIPS26. Nips Foundation.

[5] Bengio, Y., Thibodeau-Laufer, E., and Yosinski, J. (2014a). Deep generative stochastic networks trainable by backprop. In ICML'14.

[6] Bengio, Y., Thibodeau-Laufer, E., Alain, G., and Yosinski, J. (2014b). Deep generative stochastic net-works trainable by backprop. In Proceedings of the 30th International Conference on Machine Learning (ICML'14).

[7] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: a CPU and GPU math expression compiler. In Proceedings of the Python for Scientific Computing Conference (SciPy). Oral Presentation.

[8] Breuleux, O., Bengio, Y., and Vincent, P. (2011). Quickly generating representative samples from an RBM-derived process. Neural Computation, 23(8), 2053–2073.

[9] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In AISTATS'2011.

[10] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013a). Maxout networks. In ICML'2013.

[11] Goodfellow, I. J., Mirza, M., Courville, A., and Bengio, Y. (2013b). Multi-prediction deep Boltzmann machines. In NIPS'2013.

[12] Goodfellow, I. J., Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., Bergstra, J., Bastien, F., and Bengio, Y. (2013c). Pylearn2: a machine learning research library. arXiv preprint arXiv:1308.4214.

[13] Gutmann, M. and Hyvarinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In AISTATS'2010.

[14] Hinton, G., Deng, L., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. (2012a). Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine, 29(6), 82–97.

[15] Hinton, G. E., Dayan, P., Frey, B. J., and Neal, R. M. (1995). The wake-sleep algorithm for unsupervised neural networks. Science, 268, 1558–1161.

[16] Hinton, G. E., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18, 1527–1554.

[17] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012b). Improving neural networks by preventing co-adaptation of feature detectors. Technical report, arXiv:1207.0580.

[18] Hyvarinen,¨ A. (2005). Estimation of non-normalized statistical models using score matching. J. Machine Learning Res., 6.

[19] Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? In Proc. International Conference on Computer Vision (ICCV'09), pages 2146–2153. IEEE.

[20] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Proceedings of the Interna-tional Conference on Learning Representations (ICLR).

[21] Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

[22] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. In NIPS'2012.

[23] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

[24] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. Technical report, arXiv:1401.4082.

[25] Rifai, S., Bengio, Y., Dauphin, Y., and Vincent, P. (2012). A generative process for sampling contractive auto-encoders. In ICML'12.

[26] Salakhutdinov, R. and Hinton, G. E. (2009). Deep Boltzmann machines. In AISTATS'2009, pages 448– 455.

[27] Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart and J. L. McClelland, editors, Parallel Distributed Processing, volume 1, chapter 6, pages 194–281. MIT Press, Cambridge.

[28] Susskind, J., Anderson, A., and Hinton, G. E. (2010). The Toronto face dataset. Technical Report UTML TR 2010-001, U. Toronto.

[29] Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In W. W. Cohen, A. McCallum, and S. T. Roweis, editors, ICML 2008, pages 1064–1071. ACM.

[30] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In ICML 2008.

[31] Younes, L. (1999). On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates. Stochastics and Stochastic Reports, 65(3), 177–228.