# Generative Adversarial Nets

**论文原地址：https://arxiv.org/abs/1406.2661**
作者：Ian J. Goodfellow等

翻译：七月在线DL翻译组
译者：杨智友　彭博　张永彬
责编：翟惠良　July


## Abstract

我们提出一个通过对抗过程来估计生成模型的新框架，在这个框架中我们同时训练两个模型：生成模型G－－用来捕获数据分布，判别模型D－－用来估计样本来自训练数据而不是G 的概率，G的训练过程目的是最大化D产生错误的概率。这个框架相当于一个极小化极大的双方博弈。在任意函数G 和D 的空间中存在唯一的解，此时G 恢复训练数据分布，并且D处处都等于$\frac{1}{2}$。在G和D 由多层感知器构成的情况下，整个系统可以用反向传播进行训练。在训练或生成样本时不需要任何马尔科夫链或展开的近似推理网络。实验通过对生成的样本定性和定量评估来展示这个框架的潜力。

## 1　介绍

深度学习的任务是发现丰富的层次模型，这些模型在人工智能领域里用来表达各种数据的概率分布，例如自然图像，包含语音的音频波形以及自然语言语料库中的符号等。到目前为止，在深度学习领域最为成功的的模型便是判别式模型，通常它们将高维丰富的感知器输入映射到类别标签。这些显著的成功主要是基于反向传播和丢弃算法来实现的，特别是具有特别良好梯度的分段线性单元。由于在最大似然估计和相关策略中会遇见许多难以解决的概率计算困难，而且在生成上下文时很难利用使用分段线性单元的好处，导致深度生成模型的影响很小。我们提出一个新的生成模型估计程序来避开这些难题。

在提到的对抗网络框架中，生成模型对抗着一个对手：一个通过学习去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伪造团队，试图产生假货并在不被发现的情况下使用它，而判别模型类似于警察，试图检测假币。在这个游戏中的竞争驱使两个团队不断改进他们的方法，直到真假难分为止。

针对多种模型和优化算法，这个框架可以提供特定的训练方法。在这篇文章中，我们探讨了生成模型将随机噪声传输到多层感知机来生成样本的特例，同时判别模型也是通过多层感知机实现的。我们称这个特例为对抗网络。在这种情况下，我们可以仅使用非常成熟的反向传播和丢弃算法训练两个模型，生成模型在生成样本时只使用前向传播算法。并且不需要近似推理和马尔可夫链作为前提。

## 2　相关工作

含隐变量的有向图模型可以由含隐变量的无向图模型替代，例如受限波兹曼机（RBM），深度波兹曼机（DBM）和它们很多的变种。这些模型之间的相互影响可以表达为非标准化的势函数的乘积，再通过随机变量的所有状态的全局整合来标准化。这个数量（配分函数）和它的梯度的估算是很棘手的，尽管他们能够使用马尔可夫链和蒙特卡罗（MCMC）算法来估计，同时依靠MCMC算法的混合也会引发一个严重的问题。

深度置信网络（DBN）是一个包含一个无向层和若干有向层的混合模型。当使用快速逐层训练法则时，DBNS 会引发无向模型和有向模型相关的计算难题。

已经有人提出不采用似然函数的估计或约数的替代准则，例如分数匹配和噪音压缩评估（NCE）。他们都需要知道先验概率密度知识以分析指定一个规范化的常量。请注意,许多有趣的带有一些隐层变量的生成模型（如DBN和DBM），它们甚至不需要难以处理的非标准化的概率密度先验知识。一些模型如自动编码降噪机和压缩编码的学习准则与分数匹配在RBM上的应用非常相似。在NCE 中，使用一个判别训练准则来拟合一个生成模型。然而,生成模型常常被用来判别从一个固定噪音分布中抽样生成的数据，而不是拟合一个独立的判别模型。由于NCE 使用一个固定的噪音分布，仅仅是从观测变量的一个小子集中学习到一个大致正确的分布后，模型的学习便急剧减慢。

最后，一些技术并没有用来明确定义概率分布，而是用来训练一个生成器来从期望的分布中拟合出样本。这个方法优势在于这些机器学习算法能够设计使用反向传播算法训练。这个领域最近比较突出的工作包含生成随机网络（GSN），它扩展了广义的除噪自动编码器:两者都可以看作是定义了一个参数化的马尔可夫链，即一个通过执行生成马尔科夫链的一个步骤来学习机器参数的算法。同GSNs 相比，对抗网络不需要使用马尔可夫链来采样。由于对抗网络在生成阶段不需要循环反馈信息，它们能够更好的利用分段线性单元，这可以提高反向传播的性能。更多利用反向传播算法来训练生成器的例子包括变分贝叶斯自动编码和随机反向传播.

## 3 对抗网络

当模型是多层感知器时，对抗模型框架是最直接的。为了学习生成器关于数据$\boldsymbol{x}$ 上的分布$p_g$, 我们定义输入噪声的先验变量$p_z(z)$,用$G(z;\theta_g)$ 来代表数据空间的映射。这里$\boldsymbol{G}$ 是一个由含有参数$\theta_g$ 的多层感知机表示的可微函数。我们再定义了一个多层感知机$D(\boldsymbol{x};\theta_d)$用来输出一个单独的标量。$D(\boldsymbol{x})$ 代表$\boldsymbol{x}$ 来自于真实数据分布而不是$p_g$的概率，我们训练$D$来最大化分配正确标签的概率，不管数据是来自于训练样例还是$G$生成的样例.我们同时训练$G$来最小化$log(1-D(G(z)))$。换句话说，$D$和$G$的训练是关于值函数$V(G,D)$的极小化极大的二人博弈问题：

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

在下一节中，我们提出了对抗网络的理论分析，本质上表明基于训练准则可以恢复数据生成分布，当给予$G$和$D$足够的容量，即在非参数极限。如图1展示了该方法的一个非正式却更加直观的解释。实际上，我们必须使用迭代数值方法来实现这个过程。在训练的内部循环中完成$D$ 的优化是禁止的，并且有限的数据集将导致过拟合。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。只要$G$变化足够慢，可以保证$D$保持在其最佳解附近，这个策略类似SML/PCD training。该过程如算法1 所示。

实际上，方程1可能无法为$G$提供足够的梯度来学习。训练初期，当$G$ 的生成效果很差时，$D$ 会以高置信度来拒绝生成样本，因为它们与训练数据明显不同。因此，$log(1-D(G(z)))$饱和。因此我们选择最大化$logD(G(z))$而不是最小化$log(1-D(G(z)))$ 来训练$G$，该目标函数使$G$和$D$的动力学稳定点相同，并且在训练初期，该目标函数可以提供更强大的梯度。

## 4 理论结果

当$\boldsymbol{z}\sim p_{\boldsymbol{z}}$时，生成器$G$隐式的定义概率分布$p_g$ 为$G(\boldsymbol{z})$ 获得的样本的分布。因此，如果模型容量和训练时间足够大时，我们希望算法1收敛为$p_{\text{data}}$的良好估计量。本节的结果是在非参数设置下完成的，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节中显示，这个极小化极大问题的全局最优解为$p_g = p_{\text{data}}$。我们将在4.2节中展示使用算法1来优化等式1，从而获得期望的结果。

### 4.1 全局最优：$p_g = p_{\text{data}}$

首先任意给生成器$G$，考虑最优判别器$D$。

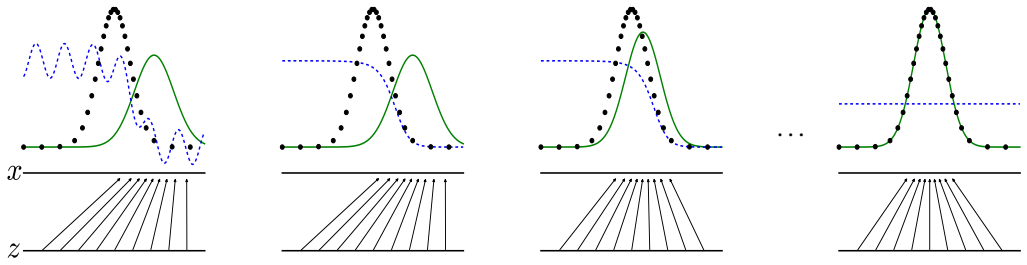


Figure 1: 在训练生成对抗网络时，同时更新判别分布（$D$，蓝色虚线）使D能区分数据生成分布$p_{\boldsymbol{x}}$（黑色虚线）中的样本和生成分布$p_g$ ($G$，绿色实线) 中的样本。下面的水平线为均匀采样$\boldsymbol{z}$的区域，上面的水平线为$\boldsymbol{x}$ 的部分区域。朝上的箭头显示映射$\boldsymbol{x} = G(\boldsymbol{z})$如何将非均匀分布$p_g$ 作用在转换后的样本上。$G$ 在$p_g$高密度区域收缩，且在$p_g$的低密度区域扩散。(a)考虑一个接近收敛的对抗的模型对：$p_g$与$p_{\text{data}}$相似，且$D$ 是个部分准确的分类器。(b)算法的内循环中，训练$D$来判别数据中的样本，收敛到：$D^*(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})+p_g(\boldsymbol{x})}$。(c)在$G$的1次更新后，$D$的梯度引导$G(\boldsymbol{z})$ 流向更可能分类为数据的区域。(d)训练若干步后，如果$G$和$D$性能足够，它们接近某个稳定点并都无法继续提高性能，因为此时$p_g = p_{\text{data}}$。判别器将无法区分训练数据分布和生成数据分布，即$D(\boldsymbol{x}) = \frac{1}{2}$。

**Algorithm 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**

  **for** $k$ steps **do**

  - 在噪声先验分布为$p_g(\boldsymbol{z})$的$m$个噪声样本$\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$中采一个minibatch。
  - 在数据分布为$p_{\text{data}}(\boldsymbol{x})$的$m$个训练样本$\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$中采一个minibatch。
  - 通过随机梯度上升来更新判别器：

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

  **end for**

  - 在噪声先验分布为$p_g(\boldsymbol{z})$ 的$m$个噪声样本$\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$中采一个minibatch。
  - 通过随机梯度下降来更新生成器：

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

**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。

**Proposition 1.** 固定$G$，最优判别器$D$为：

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

*Proof.* 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$P(Y = y|\boldsymbol{x})$的最大似然估计，当$y = 1$时，$\boldsymbol{x}$来自于$p_{\text{data}}$；当$y = 0$时，$\boldsymbol{x}$来自$p_g$。公式1 中的极小化极大问题可以变形为：

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

**Theorem 1.** 当且仅当$p_g = p_{data}$时，$C(G)$达到全局最小。此时，$C(G)$ 的值为$-\log 4$。

*Proof.* $p_g = p_{\text{data}}$时，$D^*_G(\boldsymbol{x}) = \frac{1}{2}$（公式2）。再根据公式4 可得，$C(G) = \log \frac{1}{2} + \log \frac{1}{2} = -\log 4$。为了确定仅当$p_g = p_{\text{data}}$时$C(G)$是否是最优的，观测

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

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。我们在表达式中识别出了模型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

**Proposition 2.** 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$p_{data}$。

*Proof.* 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$U(p_g, D)$ 为$p_g$的凸函数。该凸函数上确界的次导数包含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$\alpha$，$f_\alpha(x)$ 是关于$x$的凸函数，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，对抗的网络通过函数$G(\boldsymbol{z}; \theta_g)$ 表示$p_g$分布的有限簇，并且我们优化$\theta_g$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优良性能表明了这是一个合理的模型。

## 5 实验

我们在一系列数据集上，包括MNIST、多伦多面数据库（TFD）和CIFAR-10，来训练对抗网络。生成器的激活函数包括修正线性激活（ReLU）和sigmoid 激活，而判别器使用maxout激活。Dropout被用于判别器网络的训练。虽然理论框架可以在生成器的中间层使用Dropout和其他噪声，但是这里仅在生成网络的最底层使用噪声输入。

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

Table 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的平均对数似然以及在样本上平均计算的标准误差。在TFD上，我们计算数据集的不同折之间的标准误差，在每个折的验证集上选择不同的$\sigma$。在TFD 上，在每一个折上对$\sigma$进行交叉验证并计算平均对数似然函数。对于MNIST，我们与真实值（而不是二进制）版本的数据集的其他模型进行比较。

我们通过对$G$生成的样本应用高斯Parzen窗口并计算此分布下的对数似然，来估计测试集数据的概率。高斯的$\sigma$参数通过对验证集的交叉验证获得。Breuleux 等人引入该过程且用于不同的似然难解的生成模型上。结果报告在表1 中。该方法估计似然的方差较大且高维空间中表现不好，但确实目前我们认为最好的方法。生成模型的优点是可采样而不直接估计似然，从而促进了该模型评估的进一步研究。

训练后的生成样本如下图2图3所示。虽然未声明该方法生成的样本优于其它方法生成的样本，但我们相信这些样本至少和文献中较好的生成模型相比依然有竞争力，也突出了对抗框架的潜力。

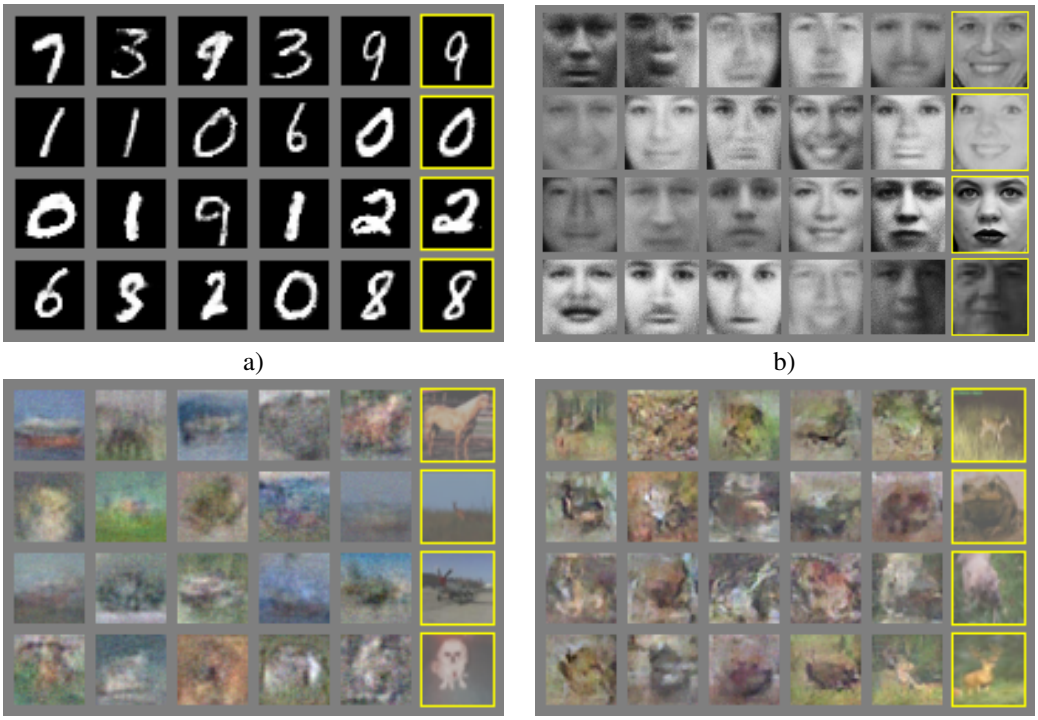


Figure 2: 来自模型的样本的可视化。最右边的列示出了相邻样本的最近训练示例，以便证明该模型没有记住训练集。样品是完全随机抽取，而不是精心挑选。与其他大多数深度生成模型的可视化不同，这些图像显示来自模型分布的实际样本。此外，这些样本是完全不相关的，因为，采样过程并不依赖马尔科夫链混合。a) MNIST；b) TFD；c) CIFAR-10（全连接模型）；d) CIFAR-10（卷积判别器和“解卷积”生成器）

Figure 3: 通过在完整模型的$\boldsymbol{z}$空间的坐标之间进行线性内插获得的数字。

| | 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 |

Table 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 优势和劣势

新框架相比以前的模型框架有其优缺点。缺点主要为$p_g(\boldsymbol{x})$是隐式表示，且训练期间，$D$和$G$必须很好地同步（尤其，不更新$D$时$G$ 不必过度训练，为避免“Helvetica 情景”。否则，$\mathbf{x}$值相同时$G$丢失过多$\mathbf{z}$ 值以至于模型$p_{\text{data}}$多样性不足），正如Boltzmann机在学习步间的不断更新。其优点是无需马尔科夫链，仅用反向传播来获得梯度，学习间无需推理，且模型中可融入多种函数。表2 总结了生成对抗网络与其他生成模型方法的比较。

上述优势主要在计算上。判别式模型可以从生成模型中获得一些统计优势，生成模型并未直接通过数据更新，而是仅用流过判别器的梯度。这意味输入部分未直接复制进生成器的参数。对抗的网络的另一优点是可表示很尖，甚至退化的分布，而基于马尔科夫链的方法为混合模式而要求模糊的分布。

## 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$\boldsymbol{c}$作为$G$ 和$D$ 的输入来获得。
2. 给定$\boldsymbol{x}$，可以通过训练一个任意的模型来学习近似推理，以预测$\boldsymbol{z}$。这和wake-sleep算法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. 能够用来近似模型所有的条件概率$p(\boldsymbol{x}_S \mid \boldsymbol{x}_{\not S})$，其中$S$通过训练共享参数的条件模型簇的关于$\boldsymbol{x}$索引的一个子集。本质上，可以使用生成对抗网络来随机拓展MP-DBM。
4. 半监督学习：当标签数据有限时，判别网络或推理网络的特征会提高分类器效果。
5. 效率改善：为协调$G$和$D$设计更好的方法，或训练期间确定更好的分布来采样$\mathbf{z}$，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。完。

译者后记

**关于我们**

七月在线DL翻译组是由一群热爱翻译、热爱DL、英语六级以上的研究生或博士组成，有七月在线的学员，也有非学员。本翻译组翻译的所有全部论文仅供学习交

流，宗旨是：汇集顶级内容帮助全球更多人。目前已经翻译数十篇顶级DL论文，详见：https://ask.julyedu.com/question/7612

**加入我们**

如果你过了英语六级、是研究生或博士、且熟练DL、热爱翻译，欢迎加入我们翻译组，微博私信@研究者July

**GAN课程**

为了帮助更多人更好的了解、学习、入门GAN，今年上半年，我们七月在线亦会开《生成对抗网络班》，从头到尾详解GAN的原理及其实战应用，敬请期待。

七月在线DL翻译组、二零一七年三月七日