生成对抗网

Ian J. Goodfellow，Jean Pouget-Abadie [*]，Mehdi Mirza，Bing Xu，David Warde-Farley
，Sherjil Ozair [†]，Aaron Courville，Aaron Courville，Yoshua Bengio { montr ´eal montr ´
eal，qc h3c 3j7

抽象的

我们提出了一个新的框架，用于通过对抗过程估算生成模型，在该过程中，我们同时训练两个模型：一种生成模型G捕获数据分布的生成模型，以及一个区分模型D估计样品来自训练数据而不是G的可能性。G的培训过程是最大化D犯错的概率。该框架对应于一个Minimax两人游戏。在任意函数G和D的空间中，存在一个唯一的解决方案，G恢复训练数据分布，D等于$\frac{1}{2}$。在G和D的情况下，由多层感知器定义，可以通过反向传播对整个系统进行训练。在培训或生成样本期间，不需要任何马尔可夫链或展开的近似推理网络工作。实验通过对生成样品的定性和定量评估来证明框架的潜力。

1简介

深度学习的承诺是发现富有的，分层的模型[2]，代表人工智能应用中遇到的数据种类的概率分布，例如自然图像，包含语音的音频波形以及自然语言语言中的符号。到目前为止，深度学习中最引人注目的成功涉及判别模型，通常是那些将高维，丰富感官输入映射到类标签的模型[14，22]。这些惊人的成功主要基于反向传播和辍学算法，使用分段线性单元[19，9，10]具有特别良好的梯度。由于很难在最大似然估计和相关策略中产生许多棘手的概率计算，并且由于难以利用在生成的生成环境中的分散线性单位的益处而产生的许多棘手的概率计算。我们提出了一种新的生成模型估计程序，以避开这些困难。[1]

在提出的*adversarial nets*框架中，生成模型是针对对手的：一个学会确定样本是来自模型分布还是数据分布的歧视模型。可以将生成模型视为类似于伪造者的团队，试图生产假货币并在没有发现的情况下使用它，而判别模型类似于警察，试图检测伪造的货币。该游戏中的竞争驱使两支球队都改善了他们的方法，直到伪造物与真实文章无关。

[*]Jean Pouget-Abadie is visiting Université de Montréal from Ecole Polytechnique.
[†]Sherjil Ozair is visiting Université de Montréal from Indian Institute of Technology Delhi
[‡]Yoshua Bengio is a CIFAR Senior Fellow.
[1]All code and hyperparameters available at `http://www.github.com/goodfeli/adversarial`

该框架可以为多种模型和优化算法产生特定的培训算法。在本文中，我们探讨了当生成模型通过通过多层感知器传递随机噪声生成样品的特殊情况，而区分模型也是多层感知器。我们将此特殊情况称为*adversarial nets*。在这种情况下，我们只能使用非常成功的反向传播和辍学算法[17]训练这两个模型，并仅使用正向传播从生成模型中进行采样。不需要大概的推理或马尔可夫链。

2相关工作

具有潜在变量的有向图形模型的替代方法是具有潜在变量的无向图形模型，例如受限的Boltzmann机器（RBMS）[27，16]，Deep Boltzmann机器（DBMS）[26]及其许多变量。此类模型中的相互作用表示为非均衡电位函数的乘积，该函数通过随机变量的所有状态上的全局汇总/集成归一化。该数量（*partition function*）及其梯度对于除最微不足道的实例以外的所有梯度都是棘手的，尽管可以通过马尔可夫链蒙特卡洛（MCMC）方法估算它们。混合对依赖MCMC的学习算法提出了一个重要的问题[3，5]。

深信仰网络（DBNS）[16]是包含单个无向层和几个定向层的混合模型。尽管存在快速近似层的训练标准，但DBN会产生与无向模型和指向模型相关的计算困难。

还提出了不近似或绑定对数似然的替代标准，例如分数匹配[18]和噪声对抗性估计（NCE）[13]。这两者都要求将学习的概率密度在分析上指定至归一化常数。请注意，在许多具有多层潜在变量（例如DBN和DBMS）的有趣的生成模型中，甚至不可能得出可拖动的不当概率密度。一些模型，例如Denoing Auto-编码器[30]和Contricive AutoCoders的学习规则与应用于RBM的得分匹配非常相似。在NCE中，就像在这项工作中一样，采用了判别训练标准来拟合生成模型。但是，生成模型本身并没有拟合单独的判别模型，而是用于将生成的数据从样品区分出固定的噪声分布。由于NCE使用固定的噪声分布，因此在模型学会了在观察到的一小部分的一小部分之后，学习却大大减慢。

最后，某些技术不涉及明确定义概率分布，而是训练生成机器以从所需的分布中绘制样品。这种方法的优点是可以设计这些机器可以通过反向传播进行训练。该领域的最新工作包括生成随机网络（GSN）框架[5]，该框架扩展了广义的deno化自动编码器[4]：两者都可以看作是定义的定义参数化的马尔可夫链，即，一个人学习了一台计算机的参数，该参数执行了生成的Markov链的一步。与GSN相比，对抗网络框架不需要马尔可夫链进行抽样。由于对抗网不需要在一代中需要反馈循环，因此它们能够更好地利用分段线性单元[19，9，10]，从而改善了反向传播的性能，但在使用INA反馈循环时会出现无界激活的问题。训练生成机的最新示例是通过向后传播中的生成机，包括有关自动编码变化贝叶斯[20]和随机反向传播的最新工作[24]。

3个对抗网

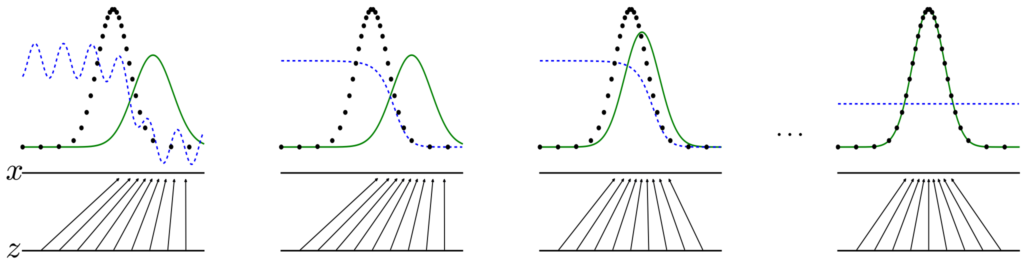
当模型都是多层感知器时，对抗建模框架最为直接地应用。要通过数据$\boldsymbol{x}$学习发电机的分布p_g，我们在输入噪声变量$p_{\boldsymbol{z}}(\boldsymbol{z})$上定义了先验，然后将映射到数据空间为$G(\boldsymbol{z};\theta_g)$，其中$G$是由具有参数$\theta_g$的多层感知器表示的可区分函数。我们还定义了第二个多层感知器$D(\boldsymbol{x};\theta_d)$输出一个标量。 $D(\boldsymbol{x})$表示$\boldsymbol{x}$来自数据而不是p_g的概率。我们训练D，以最大化将正确标签分配给培训示例和G样本的概率。我们同时训练G以最小化log（$1-D(G(\boldsymbol{z}))$）：

In other words, D and G play the following two-player minimax game with value function $V(G, D)$:

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

在下一部分中，我们将对对抗网的理论分析进行理论分析，从本质上表明，训练标准允许人们将生成分布的数据恢复为G和D的能力足够，即在非参数限制中。有关方法的形式，请参见图1。实际上，我们必须使用迭代，数值方法实施游戏。优化D在训练的内部循环中完成的计算效率是过于良好的，并且在有限的数据集上将导致过度拟合。相反，我们在k优化D的步骤和优化G的一个步骤之间进行交替。只要G变化得足够缓慢，这会导致D在其最佳解决方案附近维护。该策略类似于SML/PCD [31，29]培训从一个学习步骤到下一个学习步骤的样本，以避免在马尔可夫链中燃烧，这是学习的内部学习循环的一部分。该过程在算法1中正式显示。

在实践中，方程1可能无法为G学习良好提供足够的梯度。在学习的早期，当G很差时，D可以拒绝具有很高信心的样本，因为它们与培训数据显然不同。在这种情况下，log（1 − $D(G(\boldsymbol{z}))$）饱和。而不是训练G以最小化log（1 − $D(G(\boldsymbol{z}))$），而是可以训练G以最大化log $D(G(\boldsymbol{z}))$。该目标函数导致G和D动力学的相同固定点，但在学习早期提供了更强的梯度。



(a) (b) (c) (d)

图1：通过同时更新区分分布（D，蓝色，虚线）来训练生成的对抗网络，以便将样品与数据生成分布（黑色，虚线）$p_{\boldsymbol{x}}$区分开，从较低的水平线是在这种情况下均匀地采样$\boldsymbol{z}$的域。上面的水平线是$\boldsymbol{x}$域的一部分。向上箭头显示映射$\boldsymbol{x} = G(\boldsymbol{z})$如何在变换后的样本上施加非均匀分布$p_g$。$G$在高密度区域的收缩，并在$p_g$的低密度区域扩展。（a）考虑接线附近的对抗对：$p_g$类似于$p_{\text{data}}$，$D$是部分准确的分类。（b）在算法的内环中$D$经过训练以区分样品与数据，收敛到$D^*(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})+p_g(\boldsymbol{x})}$。（c）在更新$G$之后，$D$的梯度将$G(\boldsymbol{z})$引导到流动到更可能被分类为数据的区域。（d）经过几个步骤的训练，如果G和D具有足够的容量，它们将达到两者无法改进的地步，因为$p_g = p_{\text{data}}$。鉴别器无法区分两个分布，即$D(\boldsymbol{x}) = \frac{1}{2}$。

4理论结果

生成器G隐式将概率分布p_g定义为$\boldsymbol{z} \sim p_{\boldsymbol{z}}$时获得的样品$G(\boldsymbol{z})$的分布。因此，如果给出了足够的容量和训练时间，我们希望算法1收敛到p_{data}的良好估计器。本节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性来代表具有无限能力的模型。

我们将在第4.1节中表明，该最小值游戏具有$p_g = p_{\text{data}}$的全局最佳选择。然后，我们将在第4.2节中显示算法1优化了EQ 1，从而获得了所需的结果。

算法1 Minibatch随机梯度下降训练生成对抗网。应用于鉴别器k的步骤数是一个超参数。在我们的实验中，我们使用了$k = 1$，这是最便宜的选择。

```
用于训练迭代的数量
  $k$步骤确实
    • $m$噪声样本$\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$的示例MiniBatch先前$p_g(\boldsymbol{z})$。
    • 从数据生成分布$p_{\text{data}}(\boldsymbol{x})$的$m$示例$m$的示例示例。
    • 通过升高其随机梯度来更新鉴别器:
        $$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$
  结束
  • $m$噪声样本$\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$的示例MiniBatch先前$p_g(\boldsymbol{z})$的样本。
  • 通过下降其随机梯度来更新发电机:
        $$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$
结束
基于梯度的更新可以使用任何基于标准的基于梯度的学习规则。我们在实验中使用了Momentum。
```

4.1 $p_g = p_{\textbf{data}}$的全局最优性

我们首先考虑任何给定的生成器G的最佳区分D。

命题1。*For G fixed, the optimal discriminator D is*

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

Proof. 给定任何发电机G的鉴别器D的训练标准是最大化数量$V(G, D)$

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

对于任何（a, b）$\in \mathbb{R}^2 \setminus$ { 0 , 0 }，函数$y \to a \log$（y）$+ b \log$（$1 - y$）在[0 , 1] in {0 , 1] at $\frac{a}{a+b}$ in$\frac{a}{a+b}$中的最大值。不需要在$Supp(p_{\text{data}}) \cup Supp(p_g)$之外定义歧视者，以结束证明。 □

请注意，D的训练目标可以解释为最大化条件概率$P(Y = y|\boldsymbol{x})$的日志类近样性，其中Y指示$\boldsymbol{x}$是否来自p_{data} ((($y = 1$ $y = 1$) v4} { 0)。等式中的minimax游戏。 1现在可以重新重新重新出现：

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

Theorem 1. *The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{\text{data}}$. At that point, $C(G)$ achieves the value $-\log 4$.*

Proof. For $p_g = p_{\text{data}}$, $D^*_G(\boldsymbol{x}) = \frac{1}{2}$, (consider Eq. 2). Hence, by inspecting Eq. 4 at $D^*_G(\boldsymbol{x}) = \frac{1}{2}$, we find $C(G) = \log \frac{1}{2} + \log \frac{1}{2} = -\log 4$. To see that this is the best possible value of $C(G)$, reached only for $p_g = p_{\text{data}}$, observe that

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

and that by subtracting this expression from $C(G) = V(D^*_G, G)$, we obtain:

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

where KL is the Kullback–Leibler divergence. We recognize in the previous expression the Jensen–Shannon divergence between the model's distribution and the data generating process:

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

Since the Jensen–Shannon divergence between two distributions is always non-negative and zero only when they are equal, we have shown that $C^* = -\log(4)$ is the global minimum of $C(G)$ and that the only solution is $p_g = p_{\text{data}}$, i.e., the generative model perfectly replicating the data generating process. $\square$

4.2 Convergence of Algorithm 1

Proposition 2. *If G and D have enough capacity, and at each step of Algorithm 1, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion*

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

then p_g converges to p_{data}

Proof. Consider $V(G, D) = U(p_g, D)$ as a function of p_g as done in the above criterion. Note that $U(p_g, D)$ is convex in p_g. The subderivatives of a supremum of convex functions include the derivative of the function at the point where the maximum is attained. In other words, if $f(x) = \sup_{\alpha\in\mathcal{A}} f_\alpha(x)$ and $f_\alpha(x)$ is convex in x for every α, then $\partial f_\beta(x) \in \partial f$ if $\beta = \arg\sup_{\alpha\in\mathcal{A}} f_\alpha(x)$. This is equivalent to computing a gradient descent update for p_g at the optimal D given the corresponding G. $\sup_D U(p_g, D)$ is convex in p_g with a unique global optima as proven in Thm 1, therefore with sufficiently small updates of p_g, p_g converges to p_x, concluding the proof. $\square$

In practice, adversarial nets represent a limited family of p_g distributions via the function $G(\boldsymbol{z}; \theta_g)$, and we optimize θ_g rather than p_g itself. Using a multilayer perceptron to define G introduces multiple critical points in parameter space. However, the excellent performance of multilayer perceptrons in practice suggests that they are a reasonable model to use despite their lack of theoretical guarantees.

5 Experiments

We trained adversarial nets an a range of datasets including MNIST[23], the Toronto Face Database (TFD) [28], and CIFAR-10 [21]. The generator nets used a mixture of rectifier linear activations [19, 9] and sigmoid activations, while the discriminator net used maxout [10] activations. Dropout [17] was applied in training the discriminator net. While our theoretical framework permits the use of dropout and other noise at intermediate layers of the generator, we used noise as the input to only the bottommost layer of the generator network.

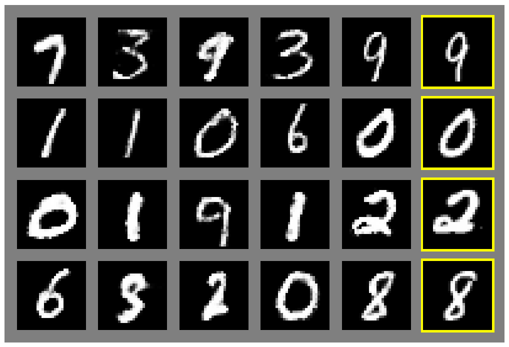
We estimate probability of the test set data under p_g by fitting a Gaussian Parzen window to the samples generated with G and reporting the log-likelihood under this distribution. The σ parameter

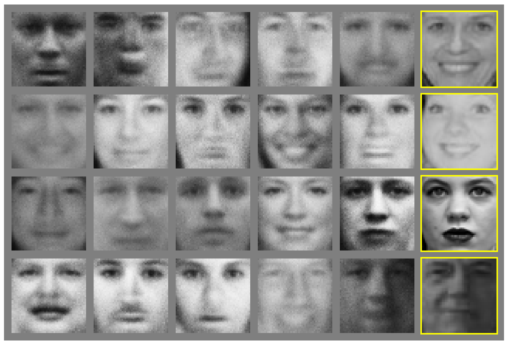
Model	MNIST	TFD
DBN [3]	138 ± 2	1909 ± 66
Stacked CAE [3]	121 ± 1.6	$\mathbf{2110 \pm 50}$
Deep GSN [6]	214 ± 1.1	1890 ± 29
Adversarial nets	$\mathbf{225 \pm 2}$	$\mathbf{2057 \pm 26}$

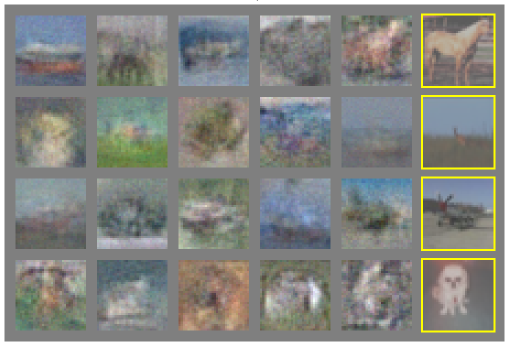
Table 1: Parzen window-based log-likelihood estimates. The reported numbers on MNIST are the mean log-likelihood of samples on test set, with the standard error of the mean computed across examples. On TFD, we computed the standard error across folds of the dataset, with a different σ chosen using the validation set of each fold. On TFD, σ was cross validated on each fold and mean log-likelihood on each fold were computed. For MNIST we compare against other models of the real-valued (rather than binary) version of dataset.

of the Gaussians was obtained by cross validation on the validation set. This procedure was introduced in Breuleux *et al.* [8] and used for various generative models for which the exact likelihood is not tractable [25, 3, 5]. Results are reported in Table 1. This method of estimating the likelihood has somewhat high variance and does not perform well in high dimensional spaces but it is the best method available to our knowledge. Advances in generative models that can sample but not estimate likelihood directly motivate further research into how to evaluate such models.

In Figures 2 and 3 we show samples drawn from the generator net after training. While we make no claim that these samples are better than samples generated by existing methods, we believe that these samples are at least competitive with the better generative models in the literature and highlight the potential of the adversarial framework.


a)


b)


c)

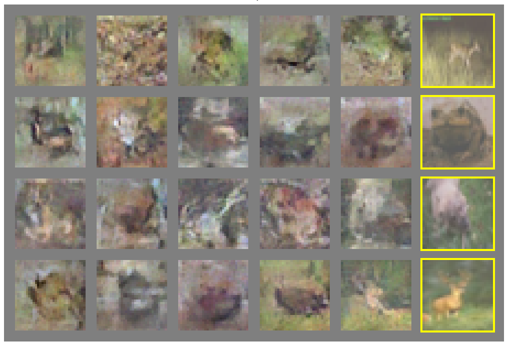

d)

Figure 2: Visualization of samples from the model. Rightmost column shows the nearest training example of the neighboring sample, in order to demonstrate that the model has not memorized the training set. Samples are fair random draws, not cherry-picked. Unlike most other visualizations of deep generative models, these images show actual samples from the model distributions, not conditional means given samples of hidden units. Moreover, these samples are uncorrelated because the sampling process does not depend on Markov chain mixing. a) MNIST b) TFD c) CIFAR-10 (fully connected model) d) CIFAR-10 (convolutional discriminator and "deconvolutional" generator)

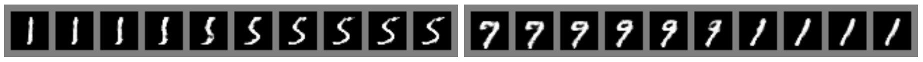


Figure 3: Digits obtained by linearly interpolating between coordinates in $\boldsymbol{z}$ space of the full model.

	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: Challenges in generative modeling: a summary of the difficulties encountered by different approaches to deep generative modeling for each of the major operations involving a model.

6 Advantages and disadvantages

This new framework comes with advantages and disadvantages relative to previous modeling frameworks. The disadvantages are primarily that there is no explicit representation of $p_g(\boldsymbol{x})$, and that D must be synchronized well with G during training (in particular, G must not be trained too much without updating D, in order to avoid "the Helvetica scenario" in which G collapses too many values of $\mathbf{z}$ to the same value of $\mathbf{x}$ to have enough diversity to model p_{data}), much as the negative chains of a Boltzmann machine must be kept up to date between learning steps. The advantages are that Markov chains are never needed, only backprop is used to obtain gradients, no inference is needed during learning, and a wide variety of functions can be incorporated into the model. Table 2 summarizes the comparison of generative adversarial nets with other generative modeling approaches.

The aforementioned advantages are primarily computational. Adversarial models may also gain some statistical advantage from the generator network not being updated directly with data examples, but only with gradients flowing through the discriminator. This means that components of the input are not copied directly into the generator's parameters. Another advantage of adversarial networks is that they can represent very sharp, even degenerate distributions, while methods based on Markov chains require that the distribution be somewhat blurry in order for the chains to be able to mix between modes.

7 Conclusions and future work

This framework admits many straightforward extensions:

1. A *conditional generative* model $p(\boldsymbol{x} \mid \boldsymbol{c})$ can be obtained by adding $\boldsymbol{c}$ as input to both G and D.
2. *Learned approximate inference* can be performed by training an auxiliary network to predict $\boldsymbol{z}$ given $\boldsymbol{x}$. This is similar to the inference net trained by the wake-sleep algorithm [15] but with the advantage that the inference net may be trained for a fixed generator net after the generator net has finished training.

3. One can approximately model all conditionals $p(\boldsymbol{x}_S \mid \boldsymbol{x}_{\not S})$ where S is a subset of the indices of $\boldsymbol{x}$ by training a family of conditional models that share parameters. Essentially, one can use adversarial nets to implement a stochastic extension of the deterministic MP-DBM [11].
4. *Semi-supervised learning*: features from the discriminator or inference net could improve performance of classifiers when limited labeled data is available.
5. *Efficiency improvements:* training could be accelerated greatly by divising better methods for coordinating G and D or determining better distributions to sample $\mathbf{z}$ from during training.

This paper has demonstrated the viability of the adversarial modeling framework, suggesting that these research directions could prove useful.

Acknowledgments

We would like to acknowledge Patrice Marcotte, Olivier Delalleau, Kyunghyun Cho, Guillaume Alain and Jason Yosinski for helpful discussions. Yann Dauphin shared his Parzen window evaluation code with us. We would like to thank the developers of Pylearn2 [12] and Theano [7, 1], particularly Frédéric Bastien who rushed a Theano feature specifically to benefit this project. Arnaud Bergeron provided much-needed support with LaTeX typesetting. We would also like to thank CIFAR, and Canada Research Chairs for funding, and Compute Canada, and Calcul Québec for providing computational resources. Ian Goodfellow is supported by the 2013 Google Fellowship in Deep Learning. Finally, we would like to thank Les Trois Brasseurs for stimulating our creativity.

References

[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 networks 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] Hyvärinen, 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 International 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.