Outline

- Introduction
- Related Work
- Adversarial Nets
- Theoretical Results
- Experiments
- Conclusions
- Applications
Introduction

- **Deep learning**
  - Discover hierarchical models that represent probability distributions
- **Discriminative models**
  - Map high-dimensional input to a class label
- **Backpropagation with well-behaved gradient**
- **Deep generative models**
  - Difficult to approximate intractable probabilistic computations
  - Discover probability distribution of data
- **Adversarial nets**
  - Police against counterfeiters
Related Work

● Deep generative models
  ○ Parametric specification of a probability distribution function
  ○ Maximize log-likelihood

● Boltzmann machine (Restricted Boltzmann Machine, RBM)
  ○ Require numerous approximation

● Variational autoencoders (VAE)
  ○ Perform approximate inference

● Generative stochastic networks (GSN)
  ○ Stochastic backpropagation
  ○ Use Markov chains
Related Work
Related Work

Reconstruction

- These biases are new
- Visible layer
- Hidden layer 1
- Reconstructions are the new output
- Activations are the new input
- Weights are the same

\[ r = b + \]

\[ w_i \ldots w_n \]
Related Work

- **Noise-contrastive estimation (NCE)**
  - Discriminate data from a fixed noise distribution
  - Ratio of the probability densities of the noise distribution and the model distribution
  - Backpropagate through both densities

- **Predictability minimization**
  - Two neural networks compete
  - Sole training criterion
  - Statistically independent between hidden units
  - Compare outputs
  - Optimization problem
Adversarial Nets

- Generative model $G$
- Discriminative model $D$
- Minimax two-player game
- No approximate inference or Markov chains
Adversarial Nets

- Train $D$ to maximize the probability of classifying training samples and samples from $G$ correctly
- Train $G$ to minimize $\log(1-D(G(z)))$

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

(a)  

(b)  

(c)  

(d)  

...
Adversarial Nets

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.

\[
\text{for number of training iterations do} \\
\quad \text{for } k \text{ steps do} \\
\quad\quad \bullet \text{Sample minibatch of } m \text{ noise samples } \{z^{(1)}, \ldots, z^{(m)}\} \text{ from noise prior } p_g(z). \\
\quad\quad \bullet \text{Sample minibatch of } m \text{ examples } \{x^{(1)}, \ldots, x^{(m)}\} \text{ from data generating distribution } p_{\text{data}}(x). \\
\quad\quad \bullet \text{Update the discriminator by ascending its stochastic gradient:} \\
\quad\quad \quad \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D(x^{(i)}) + \log \left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right]. \\
\quad \text{end for} \\
\quad \bullet \text{Sample minibatch of } m \text{ noise samples } \{z^{(1)}, \ldots, z^{(m)}\} \text{ from noise prior } p_g(z). \\
\quad \bullet \text{Update the generator by descending its stochastic gradient:} \\
\quad \quad \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D\left(G\left(z^{(i)}\right)\right)\right). \\
\text{end for} \\
\text{The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.}
Theoretical Results

- Want probability distribution of generator $G$ to be the same as the probability distribution of the data
- Want Algorithm 1 to converge to a good estimator of the probability distribution of the data
- Non-parametric
- Global optimality
- Algorithm 1 optimizes $V(D, G)$
Theoretical Results

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

We first consider the optimal discriminator $D$ for any given generator $G$.

Proposition 1. For $G$ fixed, the optimal discriminator $D$ is

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$  \hspace{1cm} (2)
Theoretical Results

Proof. The training criterion for the discriminator $D$, given any generator $G$, is to maximize the quantity $V(G, D)$

$$V(G, D) = \int_x p_{data}(x) \log(D(x))dx + \int_z p_z(z) \log(1 - D(g(z)))dz$$

$$= \int_x p_{data}(x) \log(D(x)) + p_g(x) \log(1 - D(x))dx$$

For any $(a, b) \in \mathbb{R}^2 \setminus \{0, 0\}$, the function $y \rightarrow a \log(y) + b \log(1 - y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$. The discriminator does not need to be defined outside of $\text{Supp}(p_{data}) \cup \text{Supp}(p_g)$, concluding the proof. \qed
Theoretical Results

\[ C(G) = \max_D V(G, D) \]

\[ = \mathbb{E}_{x \sim p_{\text{data}}} [\log D_G^*(x)] + \mathbb{E}_{z \sim p_z} [\log (1 - D_G^*(G(z)))] \]

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

\[ = \mathbb{E}_{x \sim p_{\text{data}}} \left[ \log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)} \right] + \mathbb{E}_{x \sim p_g} \left[ \log \frac{p_g(x)}{p_{\text{data}}(x) + p_g(x)} \right] \]
Theoretical Results

**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* $-\log 4$. 
Theoretical Results

Proof. For \( p_g = p_{\text{data}} \), \( D_G^*(\mathbf{x}) = \frac{1}{2} \), (consider Eq. 2). Hence, by inspecting Eq. 4 at \( D_G^*(\mathbf{x}) = \frac{1}{2} \), 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

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

and that by subtracting this expression from \( C(G) = V(D_G^*, G) \), we obtain:

\[
C(G) = -\log(4) + KL \left( p_{\text{data}} \parallel \frac{p_{\text{data}} + p_g}{2} \right) + KL \left( p_g \parallel \frac{p_{\text{data}} + p_g}{2} \right)
\]

(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:

\[
C(G) = -\log(4) + 2 \cdot JSD(p_{\text{data}} \parallel p_g)
\]

(6)

Since the Jensen–Shannon divergence between two distributions is always non-negative, and zero iff 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 distribution.
Experiments

● Datasets
  ○ MNIST
  ○ Toronto Face Database (TFD)
  ○ CIFAR-10

● Generator
  ○ Mixture of rectified linear activations and sigmoid activations

● Discriminator
  ○ Maxout activations
  ○ Dropout

● Gaussian Parzen window
  ○ Kernel density estimation
## Experiments

<table>
<thead>
<tr>
<th>Model</th>
<th>MNIST</th>
<th>TFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stacked CAE [3]</td>
<td>121 ± 1.6</td>
<td>2110 ± 50</td>
</tr>
<tr>
<td>Adversarial nets</td>
<td>225 ± 2</td>
<td>2057 ± 26</td>
</tr>
</tbody>
</table>

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 $\sigma$ chosen using the validation set of each fold. On TFD, $\sigma$ 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.
Experiments
Conclusions

- Generated samples are at least competitive with the better generative models
- Highlight the potential of the adversarial framework
- No explicit representation of the probability distribution of the generator
- A conditional generative model can be obtained
- Demonstrated the viability of the adversarial modeling framework
## Conclusions

<table>
<thead>
<tr>
<th></th>
<th>Deep directed graphical models</th>
<th>Deep undirected graphical models</th>
<th>Generative autoencoders</th>
<th>Adversarial models</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training</strong></td>
<td>Inference needed during training.</td>
<td>Inference needed during training. MCMC needed to approximate partition function gradient.</td>
<td>Enforced tradeoff between mixing and power of reconstruction generation</td>
<td>Synchronizing the discriminator with the generator. Helvetica.</td>
</tr>
<tr>
<td><strong>Inference</strong></td>
<td>Learned approximate inference</td>
<td>Variational inference</td>
<td>Learned approximate inference</td>
<td></td>
</tr>
<tr>
<td><strong>Sampling</strong></td>
<td>No difficulties</td>
<td>Requires Markov chain</td>
<td>Requires Markov chain</td>
<td>No difficulties</td>
</tr>
<tr>
<td><strong>Evaluating $p(x)$</strong></td>
<td>Intractable, may be approximated with AIS</td>
<td>Intractable, may be approximated with AIS</td>
<td>Not explicitly represented, may be approximated with Parzen density estimation</td>
<td>Not explicitly represented, may be approximated with Parzen density estimation</td>
</tr>
<tr>
<td><strong>Model design</strong></td>
<td>Models need to be designed to work with the desired inference scheme — some inference schemes support similar model families as GANs</td>
<td>Careful design needed to ensure multiple properties</td>
<td>Any differentiable function is theoretically permitted</td>
<td>Any differentiable function is theoretically permitted</td>
</tr>
</tbody>
</table>

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.
Applications

- Text to image generation
- Image to image translation
- Increasing image resolution
- Predicting next video frame
Applications
Thank you!