GAN and Cycle-GAN

Qingyun Wang
Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio
Why do we need generative models?

- Test our ability to use high-dimensional, complicated probability distributions
- Simulate possible futures for planning or simulated RL
- Handle missing data especially in semi-supervised learning
- Work with multi-modal outputs

GAN compared with other generative models

- Maximum Likelihood
  - Explicit density
    - Approximate density
    - Variational Autoencoder
    - Boltzmann Machine
  - Tractable density
  - Generative Stochastic Networks
- Implicit density
  - Generative Adversarial Networks

Overview

- Use a latent code
- No Markov Chain
- No variational bound
- Play a minmax game

Training Discriminator

Training Generator
MinMax Game

- The Discriminator is trying to maximize its reward
- The Generator is trying to minimize Discriminator’s reward

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

- Discriminator pushes up
- Discriminator’s ability to recognize data as being real
- Discriminator’s ability to recognize generator samples as being fake

- Generator pushes down
Discriminator Strategy

- Optimal strategy for any $p_{\text{model}}(x)$ is always

\[
D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)}
\]
Theoretical properties

- Given infinite data, infinite model capacity, direct updating of generator’s distribution
  - Unique global optimum
  - Optimum corresponds to data distribution ($p_{\text{data}} = p_g$)
  - Convergence to optimum guaranteed
Experiment Results

MNIST

TFD

CIFAR-10 (fully connected)

CIFAR-10 (convolutional)
Advantages

● Sampling (or generation) is straightforward.
● Training doesn't involve Maximum Likelihood estimation.
● Robust to overfitting since Generator never sees the training data.
● Empirically, GANs are good at capturing the modes of the distribution.
Disadvantages

- Probability distribution is implicit
  - Not straightforward to compute $P(X)$
- Training is Hard
  - Non-Convergence
    - Optimization algorithms often approach a saddle point or local minimum rather than a global minimum
    - Game solving algorithms may not approach an equilibrium at all
  - Mode-Collapse
    - Generator learns to map several different input $z$ values to the same output point
    - Generator makes multiple images that contain the same color or texture themes, or multiple images containing different views of the same dog

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Jun-Yan Zhu* Taesung Park* Phillip Isola Alexei A. Efros

https://junyanz.github.io/CycleGAN/
Motivation

- Paired training data are not available for all tasks
- GAN suffer from mode collapse
- Introduce a two-step transformation of source domain image
  - Inspired by back-translation, e.g. a sentence from English to French, and then translate it back from French to English
  - Cycle-consistent
    - Mapping from style 1 to style 2 and back again should give you almost the original image
Cycle Consistency Loss

\[ x \xrightarrow{G} \hat{y} \xrightarrow{F} \hat{x} \]
\[ y \xrightarrow{F} \hat{x} \xrightarrow{G} \hat{y} \]

Reconstruction error:
\[ \|F(G(x)) - x\|_1 \]
\[ \|G(F(y)) - y\|_1 \]
Overview

The discriminator tries to distinguish generated zebra images from real ones.

Discriminator loss: GAN generator objective, i.e. negative log probability D assigns to the sample being real.

Reconstruction loss: squared error between the original image and the reconstruction.

\[
\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F),
\]
## Experiment Results

<table>
<thead>
<tr>
<th>Loss</th>
<th>Map → Photo</th>
<th>Photo → Map</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>% Turkers labeled real</td>
<td>% Turkers labeled real</td>
</tr>
<tr>
<td>CoGAN [30]</td>
<td>0.6% ± 0.5%</td>
<td>0.9% ± 0.5%</td>
</tr>
<tr>
<td>BiGAN/ALI [8, 6]</td>
<td>2.1% ± 1.0%</td>
<td>1.9% ± 0.9%</td>
</tr>
<tr>
<td>SimGAN [45]</td>
<td>0.7% ± 0.5%</td>
<td>2.6% ± 1.1%</td>
</tr>
<tr>
<td>Feature loss + GAN</td>
<td>1.2% ± 0.6%</td>
<td>0.3% ± 0.2%</td>
</tr>
<tr>
<td>CycleGAN (ours)</td>
<td><strong>26.8% ± 2.8%</strong></td>
<td><strong>23.2% ± 3.4%</strong></td>
</tr>
</tbody>
</table>

**AMT ‘real vs fake’ test on maps ↔ aerial**

<table>
<thead>
<tr>
<th>Loss</th>
<th>Per-pixel acc.</th>
<th>Per-class acc.</th>
<th>Class IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoGAN [30]</td>
<td>0.40</td>
<td>0.10</td>
<td>0.06</td>
</tr>
<tr>
<td>BiGAN/ALI [8, 6]</td>
<td>0.19</td>
<td>0.06</td>
<td>0.02</td>
</tr>
<tr>
<td>SimGAN [45]</td>
<td>0.20</td>
<td>0.10</td>
<td>0.04</td>
</tr>
<tr>
<td>Feature loss + GAN</td>
<td>0.06</td>
<td>0.04</td>
<td>0.01</td>
</tr>
<tr>
<td>CycleGAN (ours)</td>
<td><strong>0.52</strong></td>
<td><strong>0.17</strong></td>
<td><strong>0.11</strong></td>
</tr>
</tbody>
</table>

**FCN scores on cityscapes labels → photos**

<table>
<thead>
<tr>
<th>Loss</th>
<th>Per-pixel acc.</th>
<th>Per-class acc.</th>
<th>Class IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoGAN [30]</td>
<td>0.45</td>
<td>0.11</td>
<td>0.08</td>
</tr>
<tr>
<td>BiGAN/ALI [8, 6]</td>
<td>0.41</td>
<td>0.13</td>
<td>0.07</td>
</tr>
<tr>
<td>SimGAN [45]</td>
<td>0.47</td>
<td>0.11</td>
<td>0.07</td>
</tr>
<tr>
<td>Feature loss + GAN</td>
<td>0.50</td>
<td>0.10</td>
<td>0.06</td>
</tr>
<tr>
<td>CycleGAN (ours)</td>
<td><strong>0.58</strong></td>
<td><strong>0.22</strong></td>
<td><strong>0.16</strong></td>
</tr>
</tbody>
</table>

**Classification performance of photo → labels**
Cityscapes
Monet Paintings → Photos
Collection Style Transfer
Object Transfiguration

- horse → zebra
- zebra → horse
- apple → orange
- orange → apple
Season transfer

winter Yosemite → summer Yosemite

summer Yosemite → winter Yosemite
Photo Enhancement
Limitations

- Works well for translation tasks involving color and texture changes
- Failed for tasks that require substantial geometric changes to the image, such as cat-to-dog translations because of the generator architecture which is trained to perform appearance changes in the image
Potential Improvement

- CycleGAN lacks the straightforward description of the target domain
  - Adding additional regularization term to enforce similar image content in the source space to also be similar in the target space (Harmonic GAN)
  - Translating both an image and the corresponding set of instance attributes while maintaining the permutation invariance property of the instances (InstaGAN)
  - Disentangling structured information (Cross-domain disentanglement networks)
  - Introducing a semantic content loss to cope with substantial style variation and an edge-promoting adversarial loss for preserving clear edges (CartoonGAN)
Color filling

https://github.com/RikoLi/cyclegan-line2color
Converting Monet into Thomas Kinkade

https://web.eecs.umich.edu/~fouhey//fun/monet/index.html
Face to Ramen
Thank you!