Generative Adversarial Nets

Priyanka Mehta
Sudhanshu Srivastava
Generative Adversarial Networks

What: a network that uses two kinds of models working against each other to solve a problem

Why: It can solve complex problems while requiring much less input data, with reduced computational complexity
Background: Discriminative vs Generative Models

- Categorize samples
- Widely used
- Backpropagation & dropout
- Piecewise linear units with “well-behaved gradients”

- Model the distribution that produces the samples
- Less widely used
- “Intractable probabilistic computations”
- Hard to use piecewise linear units

Image source: http://joelouismarino.github.io/blog_posts/blog_VAE.html
Adversarial Nets

Generative model (G) competes with discriminative model (D):
G tries to generate data indistinguishable from training data
D tries to distinguish them
Adversarial Nets
Adversarial Nets Implementation details

- Two Networks, Generator(G), Discriminator(D).
- G takes random noise Z, generates output G(Z).
- Aim: G(Z) must look like samples from the distribution we are interested in.
GAN Implementation details

- For an input $X$, $D$ produces the probability that $X$ is a sample from the distribution.

Image courtesy: Adam Geitgey, *Machine Learning is Fun*
GAN implementation details

- D(X) should be closer to 1 if X is a sample from the distribution, and closer to 0 if X is generated by G.
- In each iteration, take a genuine X, and a generated sample, G(Z).
- Aim: Maximize: $\log(D(X)) + \log(1-D(G(Z)))$
- Achieved by Gradient Ascent- calculate gradient of the above function, ascend along the Gradient (update weights of D according to the gradient).
GAN Implementation Details

- G should be able to fool D, so $D(G(Z))$ should be close to 1.
- Aim: Minimize $\log(1-D(G(Z)))$.
- Achieved by Gradient Descent.

In Game Theory, such a scenario is called a **Min-Max Game**.
Summary of the Math

With enough time and sufficiently complicated D and G, the distribution generated by G converges to the distribution of interest.
GAN paper results
Application 1: Conditional GAN (Mirza et al., 2014)

Natural Question- I don’t want a replica of the entire MNIST dataset, I just want the number 9.

X and Z are the same as previous discussion, Y is the subset of interest.
Figure 2: Generated MNIST digits, each row conditioned on one label
Application 2: Image from Captions (Reed et al., 2016)

this small bird has a pink breast and crown, and black primaries and secondaries.

this magnificent fellow is almost all black with a red crest, and white cheek patch.

the flower has petals that are bright pinkish purple with white stigma

this white and yellow flower have thin white petals and a round yellow stamen
Phi is an autoencoder, which gives a vector of size 128. Z is a vector of size 100.

Image courtesy Reed et al.
StackGAN: Hyper-Realistic Images from Captions (Zhang et al, 2016)

<table>
<thead>
<tr>
<th>Text description</th>
<th>Image 1</th>
<th>Image 2</th>
<th>Image 3</th>
<th>Image 4</th>
<th>Image 5</th>
<th>Image 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>This flower has petals that are white and has pink shading</td>
<td><img src="image1.png" alt="Image 1" /></td>
<td><img src="image2.png" alt="Image 2" /></td>
<td><img src="image3.png" alt="Image 3" /></td>
<td><img src="image4.png" alt="Image 4" /></td>
<td><img src="image5.png" alt="Image 5" /></td>
<td><img src="image6.png" alt="Image 6" /></td>
</tr>
<tr>
<td>This flower has a lot of small purple petals in a dome-like configuration</td>
<td><img src="image7.png" alt="Image 7" /></td>
<td><img src="image8.png" alt="Image 8" /></td>
<td><img src="image9.png" alt="Image 9" /></td>
<td><img src="image10.png" alt="Image 10" /></td>
<td><img src="image11.png" alt="Image 11" /></td>
<td><img src="image12.png" alt="Image 12" /></td>
</tr>
<tr>
<td>This flower has long thin yellow petals and a lot of yellow anthers in the center</td>
<td><img src="image13.png" alt="Image 13" /></td>
<td><img src="image14.png" alt="Image 14" /></td>
<td><img src="image15.png" alt="Image 15" /></td>
<td><img src="image16.png" alt="Image 16" /></td>
<td><img src="image17.png" alt="Image 17" /></td>
<td><img src="image18.png" alt="Image 18" /></td>
</tr>
<tr>
<td>This flower is pink, white, and yellow in color, and has petals that are striped</td>
<td><img src="image19.png" alt="Image 19" /></td>
<td><img src="image20.png" alt="Image 20" /></td>
<td><img src="image21.png" alt="Image 21" /></td>
<td><img src="image22.png" alt="Image 22" /></td>
<td><img src="image23.png" alt="Image 23" /></td>
<td><img src="image24.png" alt="Image 24" /></td>
</tr>
<tr>
<td>This flower is white and yellow in color, with petals that are wavy and smooth</td>
<td><img src="image25.png" alt="Image 25" /></td>
<td><img src="image26.png" alt="Image 26" /></td>
<td><img src="image27.png" alt="Image 27" /></td>
<td><img src="image28.png" alt="Image 28" /></td>
<td><img src="image29.png" alt="Image 29" /></td>
<td><img src="image30.png" alt="Image 30" /></td>
</tr>
<tr>
<td>This flower has upturned petals which are thin and orange with rounded edges</td>
<td><img src="image31.png" alt="Image 31" /></td>
<td><img src="image32.png" alt="Image 32" /></td>
<td><img src="image33.png" alt="Image 33" /></td>
<td><img src="image34.png" alt="Image 34" /></td>
<td><img src="image35.png" alt="Image 35" /></td>
<td><img src="image36.png" alt="Image 36" /></td>
</tr>
<tr>
<td>This flower has petals that are dark pink with white edges and pink stamen</td>
<td><img src="image37.png" alt="Image 37" /></td>
<td><img src="image38.png" alt="Image 38" /></td>
<td><img src="image39.png" alt="Image 39" /></td>
<td><img src="image40.png" alt="Image 40" /></td>
<td><img src="image41.png" alt="Image 41" /></td>
<td><img src="image42.png" alt="Image 42" /></td>
</tr>
</tbody>
</table>
StackGAN (contd.)
Application 3: “Style” Detection

Can we use a GAN to capture categorically and continuously varying details about a visual image?

Like the way digits are written, or the way hair is styled!
InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
Chen et al., 2016

Slightly modified original GAN--instead of just noise for the generative model, adds in “latent code” that encodes salient features

(ie if we know the data is written numbers, we can make sure the “random noise” contains a variable for “digit”, a variable for “stroke angle”, and a variable for “stroke thickness”)

Goal is to learn the latent variables governing the samples in a dataset
Figure 2: Manipulating latent codes on MNIST: In all figures of latent code manipulation, we will use the convention that in each one latent code varies from left to right while the other latent codes and noise are fixed. The different rows correspond to different random samples of fixed latent codes and noise. For instance, in (a), one column contains five samples from the same category in $c_1$, and a row shows the generated images for 10 possible categories in $c_1$ with other noise fixed. In (a), each category in $c_1$ largely corresponds to one digit type; in (b), varying $c_1$ on a GAN trained without information regularization results in non-interpretable variations; in (c), a small value of $c_2$ denotes left leaning digit whereas a high value corresponds to right leaning digit; in (d), $c_3$ smoothly controls the width. We reorder (a) for visualization purpose, as the categorical code is inherently unordered.
Figure 6: **Manipulating latent codes on CelebA:** (a) shows that a categorical code can capture the azimuth of face by discretizing this variation of continuous nature; in (b) a subset of the categorical code is devoted to signal the presence of glasses; (c) shows variation in hair style, roughly ordered from less hair to more hair; (d) shows change in emotion, roughly ordered from stern to happy.
Discussion Question

Applications for this application? What could this generative “style detection” be useful for?
Application: Medical Imaging!

Problems for medical image neural nets:

-old methods: go pixel-by-pixel -> information loss

-newer methods: semantic segmentation, but it needs LOTS of data to train -> medical imaging does not have that luxury
Adversarial Networks for the Detection of Aggressive Prostate Cancer Kohl et al., 2017

Solution: Adversarial nets! (Just like the initial paper)

This paper -> \[
\min_G \max_D \left( \mathbb{E}_{x \sim p_{data}}(x) \left[ \log D(x) \right] + \mathbb{E}_{z \sim p_z}(z) \left[ \log (1 - D(G(z))) \right] \right)
\]

Original paper -> \[
\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_z}(z)[\log(1 - D(G(z)))].
\]
Adversarial Networks for the Detection of Aggressive Prostate Cancer Kohl et al., 2017

S tries to produce segmentations that fool D.
D tries to identify the fakes.
Adversarial Networks for the Detection of Aggressive Prostate Cancer Kohl et al., 2017

MRI dataset: 152 patients with suspicious results

Trained on 188 tumor-diagnosed images and 475 tumor-free-diagnosed images

<table>
<thead>
<tr>
<th>training scheme</th>
<th>cross-entropy $\mathcal{L}_{mce}$</th>
<th>adversarial $\mathcal{L}_S &amp; \mathcal{L}_D$</th>
<th>hybrid $\mathcal{L}_{mce}/2 + \mathcal{L}_S &amp; \mathcal{L}_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>tumor DSC</td>
<td>0.35 ± 0.29</td>
<td>0.41 ± 0.28</td>
<td>0.39 ± 0.29</td>
</tr>
<tr>
<td>tumor sensitivity</td>
<td>0.37 ± 0.33</td>
<td>0.55 ± 0.36</td>
<td>0.49 ± 0.35</td>
</tr>
<tr>
<td>tumor specificity</td>
<td>0.98 ± 0.14</td>
<td>0.98 ± 0.14</td>
<td>0.98 ± 0.14</td>
</tr>
</tbody>
</table>
Adversarial Networks for the Detection of Aggressive Prostate Cancer

Kohl et al., 2017
Discussion Question

Medical use of adversarial networks: is it meaningless unless it can be as good as human doctors/almost no chance of dangerous error? Or is it useful the way it is now?
Other cool existing applications


- **Draw a face from text description** - could help in generating a sketch of a criminal. (Attribute2Image: Conditional Image Generation from Visual Attributes, Yan et al.)

- **Remove snow/rain from an image** - could help Self-Driving Cars by clearing their field of view when the weather is bad. (Image De-raining Using a Conditional Generative Adversarial Network, Zhang et al.)

- **Effects of aging on a given face** - Face Aging With Conditional Generative Adversarial Networks, Antipov et al.
Other cool existing applications

- Music Generation: C-RNN-GAN, Mogren et al
- Interactive Image Generation:
  https://www.youtube.com/watch?v=9c4z6YsBGQ0&feature=youtu.be
Other potential applications

- Hyper-realistic Virtual Conversational Agent using inverse Lip-reading. Can be made to look like any person you want. Will also be useful in the animation industry.
- Translating other languages to ASL.
- Once the training is complete, the Discriminator can be used to detect fake/photoshopped images.
- Demand-based art.
- ‘Virtually’ trying on clothes on e-commerce websites.
Discussion

How does everyone feel about GANs?

Can you think of other uses for them?
Sources
