Generative Adversarial Networks

Reading: I. Goodfellow et al., Generative adversarial networks, NIPS 2014
Adversarial Networks

Generative Model

Real world

Discriminative Model

real or fake?

From https://www.cs.colorado.edu/~mozer/Teaching/syllabi/DeepLearningFall2017/lectures/gan.pptx
Training Procedure: Basic Idea

- G tries to fool D

- D tries not to be fooled

- Models are trained simultaneously
  - As G gets better, D has a more challenging task
  - As D gets better, G has a more challenging task

- Ultimately, we don’t care about the D
  - Its role is to force G to work harder
Generative Adversarial Networks

Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

Solution: Sample from a simple distribution, e.g. random noise. Learn transformation to training distribution.

Q: What can we use to represent this complex transformation?

A: A neural network!

Output: Sample from training distribution

Input: Random noise

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014
GAN Training Algorithm

**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.

for number of training iterations do
    for $k$ steps do
        • Sample minibatch of $m$ noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
        • Sample minibatch of $m$ examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
        • Update the discriminator by ascending its stochastic gradient:
          $$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D(x^{(i)}) + \log \left( 1 - D\left(G\left(z^{(i)}\right)\right) \right) \right].$$
    end for

    • Sample minibatch of $m$ noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
    • Update the generator by descending its stochastic gradient:
      $$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left( 1 - D\left(G\left(z^{(i)}\right)\right) \right).$$
end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.
From Goodfellow et al., 2014
3.5 Years of Progress on Faces


(Brundage et al, 2018)

Kerras et al, PROGRESSIVE GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION

Figure 5: 1024 x 1024 images generated using the CELEBA-HQ dataset. See Appendix F for a larger set of results, and the accompanying video for latent space interpolations.
Vector arithmetic
Cycle GANs
(Zhu et al., 2017; arXiv:1703.10593v2 [cs.CV])

Given two image collections, algorithm learns to translate an image from one collection to the other.

From https://www.cs.colorado.edu/~mozer/Teaching/syllabi/DeepLearningFall2017/lectures/gan.pptx
Photos to paintings

From https://www.cs.colorado.edu/~mozer/Teaching/syllabi/DeepLearningFall2017/lectures/gan.pptx
Figure 2: Training a conditional GAN to map edges → photo. The discriminator, $D$, learns to classify between fake (synthesized by the generator) and real \{edge, photo\} tuples. The generator, $G$, learns to fool the discriminator. Unlike an unconditional GAN, both the generator and discriminator observe the input edge map.
https://phillipi.github.io/pix2pix/

Image-to-Image Translation
Image-to-Image Translation with Conditional Adversarial Networks
Phillip Isola Jun-Yan Zhu Tinghui Zhou Alexei A. Efros
Figure 15: Example results of our method on day→night, compared to ground truth.
Figure 16: Example results of our method on automatically detected edges→handbags, compared to ground truth.
Figure 17: Example results of our method on automatically detected edges→shoes, compared to ground truth.
Figure 19: Example results on photo inpainting, compared to [43], on the Paris StreetView dataset [14]. This experiment demonstrates that the U-net architecture can be effective even when the predicted pixels are not geometrically aligned with the information in the input – the information used to fill in the central hole has to be found in the periphery of these photos.
Generative Adversarial Text to Image Synthesis

Reed et al.

This flower has small, round violet petals with a dark purple center

\[ \varphi \rightarrow \varphi(t) \]

This flower has small, round violet petals with a dark purple center

\[ z \sim \mathcal{N}(0, 1) \]

\[ \hat{x} := G(z, \varphi(t)) \]

\[ \varphi \rightarrow \varphi(t) \]

\[ D(\hat{x}, \varphi(t)) \]

**Figure 2.** Our text-conditional convolutional GAN architecture. Text encoding \( \varphi(t) \) is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.
Algorithm 1 GAN-CLS training algorithm with step size $\alpha$, using minibatch SGD for simplicity.

1: **Input:** minibatch images $x$, matching text $t$, mis-matching $\hat{t}$, number of training batch steps $S$

2: **for** $n = 1$ **to** $S$ **do**

3: $h \leftarrow \varphi(t)$ \{Encode matching text description\}

4: $\hat{h} \leftarrow \varphi(\hat{t})$ \{Encode mis-matching text description\}

5: $z \sim \mathcal{N}(0, 1)^Z$ \{Draw sample of random noise\}

6: $\hat{x} \leftarrow G(z, h)$ \{Forward through generator\}

7: $s_r \leftarrow D(x, h)$ \{real image, right text\}

8: $s_w \leftarrow D(x, \hat{h})$ \{real image, wrong text\}

9: $s_f \leftarrow D(\hat{x}, h)$ \{fake image, right text\}

10: $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$

11: $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ \{Update discriminator\}

12: $\mathcal{L}_G \leftarrow \log(s_f)$

13: $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ \{Update generator\}

14: **end for**