Understanding Generative Adversarial Nets from First Principles

Weili Nie[†] , Wanjia Liu[*] and Ankit B. Patel[†‡]

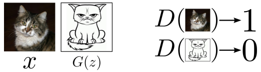
† Department of Electrical and Computer Engineering, Rice University
* Department of Computer Science, Rice University
‡ Department of Neuroscience, Baylor College of Medicine

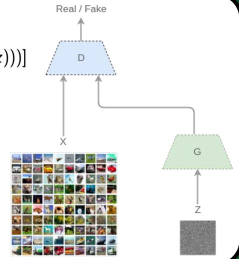
MOTIVATION

- Generative Adversarial Nets (GANs) have succeeded at image generation, text-to-image translation, image style transfer, etc.
- GAN training involves many issues: exploding or vanishing gradient, mode collapse and (somehow) poor visual quality
- The reason why GANs produce realistic images remains unclear

GAN MODEL

$$\min_G \max_D \mathbb{E}_{x\sim real}\left[\log D(x)\right] + \mathbb{E}_{z\sim noise}\left[\log\left(1 - D(G(z))\right)\right]$$





- G: generator neural network
- D: discriminator neural network
- A two-player game

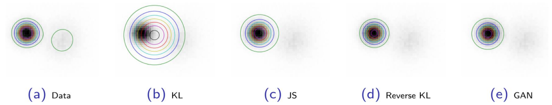
UNREALISTIC ASSUMPTIONS

- Six assumptions needed to derive GAN training objectives

Assumptions	Training Issues	Possible Solutions
A1 (Non-degenerate data)	vanishing gradients exploding gradients	IPMs[4] new training algos
A2 (D objective: cross-entropy)	vanishing gradients	other scoring rules IPMs
A3 (Unlimited D capacity)	poor visual quality vanishing gradients	new training algos new net architecture
A4 (Unlimited G capacity)	poor visual quality mode collapse	new training algos new net architecture
A5 (G objective: JS divergence)	mode collapse vanishing gradients poor visual quality	other divergences IPMs
A6 (Infinite training data)	visual quality	new metrics more data

OBJECTIVE FUNCTIONS - A2 & A5

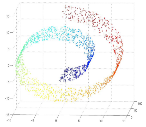
- **D objective** - Use other proper scoring rules: hingle loss, square-error loss, etc.
- **G objective** - Use other f-divergence, MMD, IPMs, etc.



Different behaviors of minimizing KL, JS, Reverse KL and GAN under model mismatch

DEGENERATE DATA - A1

- Both real and generated data lie in lower-dimensional manifold
- f-div $KL(\mathbb{P}_r||\mathbb{P}_g) = KL(\mathbb{P}_g||\mathbb{P}_r) = \infty$, and $JS(\mathbb{P}_r||\mathbb{P}_g) = \log 2$
- KL and reverse KL cause exploding gradient issues; JS causes vanishing gradient issues
- Wasserstein metric is theoretically resonable but still hard to train in practice



Data manifold

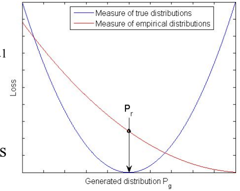
SAMPLE COMPLEXITY - A6

- If m << exp(d), GAN training generalizes poorly
- For example, given real and generated data distributions i.i.d. Gaussians, with high probability[1]

$$JS(\mathbb{P}_r||\mathbb{P}_g) = 0,\ JS(\hat{\mathbb{P}}_r||\hat{\mathbb{P}}_g) = \log 2$$

$$W(\mathbb{P}_r, \mathbb{P}_g) = 0,\ W(\hat{\mathbb{P}}_r, \hat{\mathbb{P}}_g) \geq \sqrt{2} - \sqrt{\frac{10}{d}}$$

- This may serve as a bottleneck of improving GANs performance



[1] S. Arora, et al. "Generalization and equilibrium in generative adversarial nets (GANs)." arXiv preprint arXiv:1703.00573.

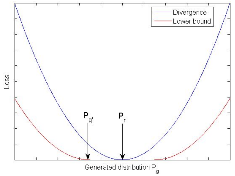
EXPRESSIVE POWER - A3 & A4

- Discriminator cannot closely approximate f-divergences, e.g.

$$L_G(\phi) < JS(\mathbb{P}_r||\mathbb{P}_g)$$

- Generator cannot make fake data close to real data but may still fool an imperfect discriminator

$$\exists\, \mathbb{P}_{g'} \neq \mathbb{P}_r,\ L_G(\phi') = 0$$



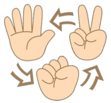
- Better to interpret GANs training as a two-player game rather than minimizing a divergence

A TWO-PLAYER GAME

- Nash Equilibrium in GANs (proved):

Lemma 1. *Assume discriminator and generator both have sufficient expressive power, Nash equilibrium exists in vanilla GAN and is characterized by* $\mathbb{P}_{g_{\phi^*}} = \mathbb{P}_r$, $D_{\theta^*}(x) = \frac{1}{2}$.

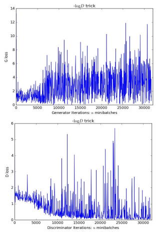
- Non-convergence exists in GANs:
  - (1) Alternatively train G and D to their own optimal
  - (2) G tries to cover the most likely mode
  - (3) D tries to assign lowest value to G output
- Non-convergence results in mode-collapse



EXPERIMENTAL RESULTS

Generated samples

Oscillating behaviors match well with our analysis: Nonconvergence exists if training G and D to be near optimal



Training curves