Understanding the vanilla Generative Adversarial Networks

Linlin Zhao

Institute of Mathematical Modeling of Biology Systems, HHU Dusseldorf

October 25, 2018

Outlines

- **What are generative models**
- What is the Buzzword, GAN?
- Theoretical digestion of GAN
- Challenges and frontiers in GAN by Nima

What are generative models?

Statistically modeling the probability distribution of the input data.

Generative models - Text to image

source: hanzhanggit/StackGAN

Generative models - Image translations

source: Isola et.al 2016

More awesome applications on GitHub: nightrome/really-awesome-gan

Outlines

- What are generative models
- **What is the Buzzword, GAN?**
- Theoretical digestion of GAN
- Challenges and frontiers in GAN by Nima

What is GAN?

Generate vs. discriminate

It is a network! :-)

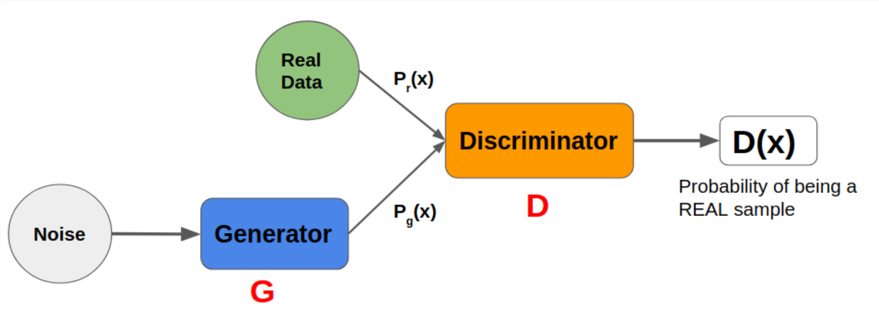
Generative Adversarial Network

In a competitive environment

G tries to generate **fake** samples

D tries to **distinguish** fake and real samples

What is GAN?



- Zero sum game between players G and D
- Training G to maximize the probability of D making a mistake

Mathematical descriptions for GAN

$$J_D = -\left[\mathbb{E}_{x\sim\mathbb{P}_r} \log D(x) + \mathbb{E}_{x\sim\mathbb{P}_g} \log\left(1 - D(x)\right)\right]$$

$$J_G = \mathbb{E}_{x\sim\mathbb{P}_g} \log\left(1 - D(x)\right)$$

$J_D := J_D(\theta_d, \theta_g)$, $J_G := J_G(\theta_g)$

Training algorithm for GAN

Number of steps k applying to train the discriminator

initializing θ_g, θ_d;

for *number of training iterations* **do**

  for *k steps* **do**

  - Sample minibatch $\{z^{(i)}\}_{i=1}^{m}$ from noise $p(z)$
  - Sample minibatch $\{x^{(i)}\}_{i=1}^{m}$ from real data $\mathbb{P}_r$
  - $\theta_d \longleftarrow \theta_d + \alpha \partial J_D / \partial \theta_d$

  end

- Sample minibatch $\{z^{(i)}\}_{i=1}^{m}$ from noise $p(z)$
- $\theta_g \longleftarrow \theta_g + \alpha \partial J_G / \partial \theta_g$

end

Outlines

- What are generative models
- What is the Buzzword, GAN?
- **Theoretical digestion of GAN**
- Challenges and frontiers in GAN by Nima

Fix G, what is the best D?

For an arbitrary sample x

$$J_D(x) = -\mathbb{P}_r(x)\log D(x) + \mathbb{P}_g(x)\log\left(1 - D(x)\right)$$

$$\frac{\partial J_D(x)}{\partial D(x)} = 0 \implies D^*(x) = \frac{\mathbb{P}_r(x)}{\mathbb{P}_r(x) + \mathbb{P}_g(x)}$$

Note that $\mathbb{P}_r(x) = \mathbb{P}_g(x) \implies D^*(x) = 0.5$

With optimal D, what happens to G?

After some math twists, we have

$$\min J_G = \min\{JS\left(\mathbb{P}_r \| \mathbb{P}_g\right)\}$$

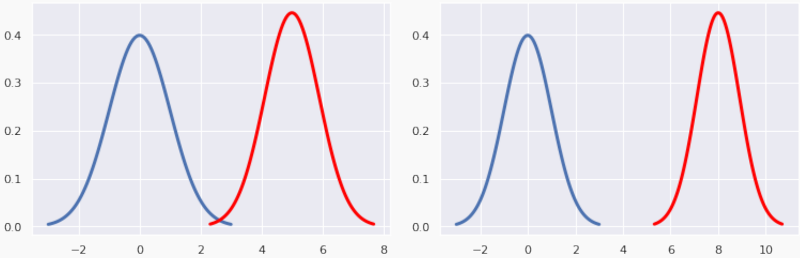
$JS(p_1 \| p_2) := \frac{1}{2}KL(p_1 \| \frac{p_1+p_2}{2}) + \frac{1}{2}KL(p_2 \| \frac{p_1+p_2}{2})$

$KL(p1 \| p_2) := \mathbb{E}_{x \sim p_1} \log \frac{p_1}{p_2}$

Source: Arjovsky and Bottou, 2017

With a perfect discriminator, we are using Jensen-Shanon Divergence as cost function to train the generator!

What is wrong with JS convergence?



$$\max JS(p_1 \| p_2) = \log 2$$

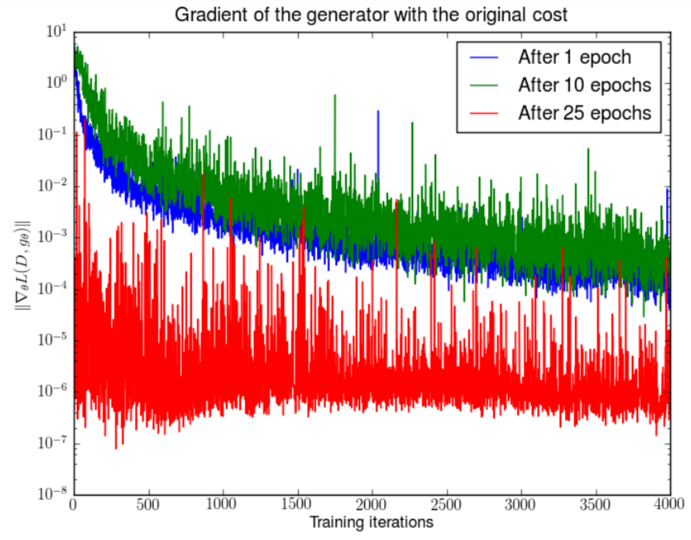
Gradient Vanishing

- With small number of iterations, we'll have an almost perfect D.
- And:

$$P\left(JS(\mathbb{P}_r \| \mathbb{P}_g) = \log 2\right) = 1$$

This means we'll easily have a constant cost function for the G!

Gradient Vanishing



Source: Arjovsky and Bottou, 2017

The "Log D trick"

To avoid gradient vanishing, alternative cost function for G

$$J_G = \mathbb{E}_{x \sim \mathbb{P}_g} \left[- \log \left(D(x) \right) \right]$$

G maximizes the log probability of the discriminator being mistaken, obtaining better gradient signal.

Unstability caused by the "Log D trick"

Similarly, with optimal D,

$$\min J_G = \min \left\{ KL(\mathbb{P}_g \| \mathbb{P}_r) - 2JS(\mathbb{P}_g \| \mathbb{P}_r) \right\}$$

- Minimizing KL divergence drags $\mathbb{P}_g$ and $\mathbb{P}_r$ close.
- Maximizing JS divergence pushes them far apart.

Very counter-intuitive, this objective makes the training very unstable

Mode collapse by the "log D trick"

The $KL(\mathbb{P}_g \| \mathbb{P}_r)$ is asymmetric, which leads to mode collapse.

KL	$P_g(x)$	$P_r(x)$	Intuition
Low	Small	Large	G did not generate a real sample
High	Large	Small	G did generate unreal sample

$$KL(p_g \| p_r) = P_g(x) \log \frac{P_g(x)}{P_r(x)}$$

The large KL value punishes the G for exploring different modes.

The dilemma in training GANs

The discriminator can neither be too good nor too bad!

- If the discriminator behaves badly, the generator does not have accurate feedback and the loss function cannot represent the reality.
- If the discriminator does a great job, we are facing gradient vanishing, unstability and mode collapse.

Outlines

- What are generative models
- What is the Buzzword, GAN?
- Theoretical digestion of GAN
- **Challenges and frontiers in GAN by Nima**