Mathematics behind GAN

Li Jun

2018 c 1 œ 29 F

Contents

Math behind GAN

Definition

GAN is composed of two networks: Descrimitive Network, and Generative Network.

GAN abstract

1. GAN is a framework for estimating generative models via an *adversarial process*

GAN abstract

1. GAN is a framework for estimating generative models via an *adversarial process*
2. simultaneously train two models: A *generative* model G and A *discriminative* model D.

GAN abstract

1. GAN is a framework for estimating generative models via an *adversarial process*
2. simultaneously train two models: A *generative* model G and A *discriminative* model D.
3. This framework corresponds to a minimax two-player game.

Previous Generative Models

1. deep Boltzmann machine

Previous Generative Models

1. deep Boltzmann machine
2. Generative stochastic networks

Previous Generative Models

1. deep Boltzmann machine
2. Generative stochastic networks
3. variational autoencoders(VAEs)

Previous Generative Models

1. deep Boltzmann machine
2. Generative stochastic networks
3. variational autoencoders(VAEs)
4. ...

Math For Basic GAN

Generator

1. data x
2. input noise variables $p_z(z)$
3. mapping to data space as $G(z;\theta_g)$,where G is a differentiable funciton represented by a multilayer perceptron with parameter θ_g.

Math For Basic GAN 2

Discriminator

1. $D(x; \theta_d)$ which is a multilayer perceptron that outputs a single scalar.
2. $D(x)$ represents the probability that x came from data rather than p_g

Math For Basic GAN 3

minimax playgame

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

Math For Basic GAN 4

Optimium D

$$\max_D V(D,G) = E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p_z}[\log(1-D(G(z))]$$
$$= E_{x\sim p_{data}}[\log D(x)] + E_{x\sim p_g}[\log(1-D(x)]$$
$$= \int_x p_{data}(x)[\log D(x)]dx + \int_x p_g(x)\log(1-D(x))dx$$

Math For Basic GAN 5

Optimium D

1. $\max_D V(D,G) = \int_x p_{data}(x)[\log D(x)]dx + \int_x p_g(x)\log(1-D(x))dx$
2. for given x, $p_{data}(x)$ is constant,marked as a
3. for given x, $p_g(x)$ is constant,marked as b
4. $f(D) = a\log D + b\log(1-D)$
5. To find max of $f(D)$, $\frac{\partial f(D)}{\partial D} = 0$
6. We get $D = \frac{a}{a+b}$
7. That is,for given G, $D^\star = \frac{p_{data}(x)}{p_g(x)+p_{data}(x)}$

Math For Basic GAN 6

Optimium D

for given G, $D^{\star} = \frac{p_{data}(x)}{p_g(x)+p_{data}(x)}$

$$\max_{D} V(D^{\star}, G) =$$

$$\int_x p_{data}(x)[\log D^{*}(x)]dx + \int_x p_g(x)\log(1 - D^{*}(x))dx$$

$$=$$

$$\int_x p_{data}(x)[\log \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}]dx + \int_x p_g(x)[\log \frac{p_g(x)}{p_{data}(x)+p_g(x)}]dx$$

$$= \int_x p_{data}(x)[\log \frac{\frac{1}{2}p_{data}(x)}{\frac{1}{2}(p_{data}(x)+p_g(x))}]dx +$$

$$\int_x p_g(x)[\log \frac{\frac{1}{2}p_g(x)}{\frac{1}{2}(p_{data}(x)+p_g(x))}]dx$$

$$= -\log 4 + KL(p_{data}||\frac{1}{2}(p_{data} + p_g)) + KL(p_g||\frac{1}{2}(p_{data} + p_g))$$

Math For Basic GAN 7

Optimum Discriminator $D^{\star}$

$$\max_D V(D,G) =$$

$$-\log 4 + KL(p_{data}||\frac{1}{2}(p_{data}+p_g)) + KL(p_g||\frac{1}{2}(p_{data}+p_g))$$

if $p_g = p_{data}$

then $\max_D V(D,G) = -\log 4$

Math For Basic GAN 8

Distance at Optimum Discriminator

$$\max_D V(D,G) =$$
$$-\log 4 + KL(p_{data}||\tfrac{1}{2}(p_{data}+p_g)) + KL(p_g||\tfrac{1}{2}(p_{data}+p_g))$$
$$\max_D V(D,G) = -\log 4 + 2 * JSD(p_{data}||p_g)$$

Math For Basic GAN 9

Theorem (Optimum Generator G^*)

After we get optimum Discriminator $D^\star$, we want to get optimum Generator by gradient descent to get the minimum value of $\max V(D, G)$, $\rightarrow \min_G \max_D V(G, D)$ *and when* $p_g = p_{data}$, *it can get the minimum.*

$\theta_g \leftarrow \theta_g - \alpha \frac{\partial L(G)}{\partial \theta_g}$

Different Distances

1. Total Variation Distance

$$\delta(P_r, P_g) = \sup_{A \in \Sigma} |P_r(A) - P_g(A)|$$

Different Distances

1. Total Variation Distance

   $$\delta(P_r, P_g) = \sup_{A \in \Sigma} |P_r(A) - P_g(A)|$$

2. Kullback-Leibler (KL) Divergence

   $$KL(P_r || P_g) = \int P_r(x) \log \frac{P_r(x)}{P_g(x)} d\mu(x)$$

Different Distances

1. Total Variation Distance

   $$\delta(P_r, P_g) = \sup_{A \in \Sigma} |P_r(A) - P_g(A)|$$

2. Kullback-Leibler (KL) Divergence

   $$KL(P_r || P_g) = \int P_r(x) \log \frac{P_r(x)}{P_g(x)} d\mu(x)$$

3. Jensen-Shannon (JS) Divergence

   $$JS(P_r, P_g) =$$

   $$KL(P_r || (P_r + P_g)/2) + KL(P_g || (P_r + P_g)/2)$$

Different Distances

1. Total Variation Distance

$$\delta(P_r, P_g) = \sup_{A \in \Sigma} |P_r(A) - P_g(A)|$$

2. Kullback-Leibler (KL) Divergence

$$KL(P_r||P_g) = \int P_r(x) \log \frac{P_r(x)}{P_g(x)} d\mu(x)$$

3. Jensen-Shannon (JS) Divergence

$$JS(P_r, P_g) =$$
$$KL(P_r||(P_r + P_g)/2) + KL(P_g||(P_r + P_g)/2)$$

4. Earth-Mover(EM) distance or Wasserstein-1

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} E_{(x,y)\sim\gamma}[||x - y||]$$

Wassertein GAN

Distance

$W(P_r, P_g) = \sup_{\|f\|_L \le 1} E_{x \sim P_r}[f(x)] - E_{x \sim P_\theta}[f(x)]$

$W(P_r, P_g) = \max_{\|f\|_L \le 1} E_{x \sim P_r}[f(x)] - E_{x \sim P_\theta}[f(x)]$ We should find all 1-Lipschitz functions f: $\chi \to \mathbf{R}$

Lipschitz Function

$\|f(x_1) - f(x_2)\| \le k\|x_1 - x_2\|$ when $k = 1,$ for 1-Lipschitz

That is: Find Max distance$W(P_r, P_g)$, and then optim by gradi

$\nabla_\theta W(P_r, P_\theta) = -E_{z \sim p(z)}[\nabla_\theta f(g_\theta(z))]$ Also, we clamp the weights to a fixed box,say $W = [-0.01, 0.01]$.

Wassertein GAN summary

So, Wassertein GAN just clip the weights? and then assume the neural network learn the f function to fullfil the 1-Lipschitz function!

Improved Wassertein GAN

T

o be continued.

KL divergence

Definition

$KL(p||q) = \sum_{k=1}^{N} p_k \log \frac{p_k}{q_k}$

What's the mean of KL divergence

the divergence (distance) of two distributions.