Generative Adversarial Networks

listenzcc

April 20, 2020

Abstract

Not done yet.

Contents

1 Real and Fake samples

Generative Adversarial Networks (GAN) can generate *FAKE* samples using adversarial learning algorithm. GAN has been widely used for extending new samples or stylize a given sample. The aim is to make the fake samples following the same distribution with *REAL* samples.

GAN has two parts, *Generator* (G) and *Discriminator* (D). The generator is to generate *FAKE* samples, the discriminator is to detect them. Thus, G and D are adversarial. The question is how GAN works.

2 Discriminator

A *Discriminator* is a two-classes classifier $D(x)$ that gives 1 for *REAL* sample and 0 for *FAKE* one. Thus, the loss function of a discriminator is

$$-\mathbb{E}_{x\sim P_r}[\log D(x)] - \mathbb{E}_{x\sim P_g}[\log(1 - D(x))] \tag{1}$$

where P_r and P_g are the probability of an image x belongs to *REAL* and *FAKE* distributions.

To a given generator, the loss caused by an image is

$$\mathcal{L}(x) = -P_r(x)\log D(x) - P_g(x)\log(1 - D(x)) \tag{2}$$

Lemma 2.1. *The optimal discriminator of a given image x is*

$$D^*(x) = \frac{P_r(x)}{P_r(x) + P_g(x)}$$

Proof. Note the loss function as $\mathcal{L}$, we have

$$\frac{\partial}{\partial D}\mathcal{L} = -\frac{P_r(x)}{D(x)} + \frac{P_g(x)}{1-D(x)}$$
$$\frac{\partial}{\partial^2 D}\mathcal{L}^2 = \frac{P_r(x)}{D^2(x)} - \frac{P_g(x)}{(1-D(x))^2}$$

One solution that minimizes the $\mathcal{L}$ is

$$D^*(x) = \frac{P_r(x)}{P_r(x) + P_g(x)}$$

when $P_r(x) \leqslant P_g(x)$, we have $\frac{\partial}{\partial^2 D}\mathcal{L}^2 \geqslant 0$. Which guarantees that $D^*(x)$ is the minimization solution.

Hence proved. □

3 Generator

The loss function of generator can be like

$$\mathbb{E}_{x\sim P_g}[\log(1 - D(x))] \tag{3}$$

the aim is to deceive the discriminator makes $D(x) = 1$.

The (3) can be rewritten as following by adding an *irrelevant* factor

$$\mathbb{E}_{x\sim P_r}[\log D(x)] + \mathbb{E}_{x\sim P_g}[\log(1 - D(x))] \tag{4}$$

it is easy to see that (3) and (4) are equal. We can also see that (4) is the reverse of (1).

Lemma 3.1. *Under optimal discriminator $D^*(x)$ The (4) can be written as*

$$2JS(P_r \| P_g) - 2\log 2$$

Proof. Start by defining two measurements, KL divergence and JS divergence.

$$KL(P_1 \| P_2) = \mathbb{E}_{x\sim P_1} \log \frac{P_1}{P_2}$$
$$2JS(P_1 \| P_2) = KL(P_1 \| \frac{P_1 + P_2}{2}) + KL(P_2 \| \frac{P_1 + P_2}{2})$$

Re-write (4) under $D^*(x)$

$$\mathbb{E}_{x\sim P_r} \log \frac{2P_r(x)}{P_r(x) + P_g(x)} + \mathbb{E}_{x\sim P_g} \log \frac{2P_g(x)}{P_r(x) + P_g(x)} - 2\log 2$$

Hence proved. □

Use Lemma 3.1 we can conclude that under optimized discriminator, the training of the generator equals to minimize the JS divergence between P_r and P_g.

3.1 Gradient vanishing in generator

In high dimensional space, where the support set of the data *MANIFOLD* is smaller than the space. The JS divergence is 0 at almost for all the images. It results that the metric is almost 0

$$\int P_r(x)P_g(x)dx \approx 0 \tag{5}$$

it shows that either P_r or P_g is 0 for almost every image in the space. As a result, the gradient of (4) is vanishing under D^*.

3.2 Log D trick

One solution to gradient vanishing is $\log D$ trick. It changes the (3) into

$$\mathbb{E}_{x\sim P_g}[-\log D(x)] \tag{6}$$

3.3 Instable of generator