Weekly Report(July 12th, 2019)

Jianghao Lin
Shanghai Jiao Tong University
chiangel.ljh@gmail.com

Abstract

I read the paper *Generative Adversarial Networks* written by Ian Goodfellow in 2014 and figure out the basic idea and mathematical derivations of GAN.

1 Basic idea of GAN

Generative adversarial nets are based on a game theoretic scenario where the generator network directly produces images x from a random noise z and another discriminator network tries to distinguish the generated image from the real ones.

Ideally, the distribution of the generated data will be the same as the real data image and the accuracy of the discriminator network will be 0.5 because it can not tell an image is actually fake or not.

2 Mathematical derivations

2.1 How to measure the divergence ?

2.1.1 KL divergence

KL divergence is also called relative entropy.

For discrete probability distributions P and Q defined on the same probability space, the KL divergence between P and Q is defined to be:

$$KL(P||Q) = \sum_x P(x)log(\frac{P(x)}{Q(x)}) \tag{1}$$

For continuous random variables, it is defined to be:

$$KL(P||Q) = \int_{-\infty}^{\infty} p(x)log(\frac{p(x)}{q(x)})dx \tag{2}$$

We can have the inequation because the log function is a convex function:

$$\begin{aligned} KL(P||Q) &= \int_{-\infty}^{\infty} p(x) log(\frac{p(x)}{q(x)}) dx \\ &= -E(log\frac{P(x)}{Q(x)}) \\ &\geq -log(E(\frac{P(x)}{Q(x)})) \\ &= -log(P(x) \times \frac{P(x)}{Q(x)}) = 0 \end{aligned} \tag{3}$$

Therefore, KL divergence is in range of $[0, 1]$. KL divergence is equal to 0 if and only if two distribution is exactly the same. And the smaller the KL divergence is, the more similar two distributions are.

2.1.2 JS divergence

It is obvious that KL divergence is asymmetric, which means that $KL(P||Q) \neq KL(Q||P)$. So we introduce the JS divergence:

$$JS(P||Q) = \frac{1}{2} KL(P||M) + \frac{1}{2} KL(Q||M) \tag{4}$$

where $M = \frac{1}{2}(P + Q)$. The range of JS divergence is $[0, log2]$. JS divergence is obviously symmetric and the smaller the JS divergence is, the more similar two distributions are.

2.2 Definitions in GANs

- **Generator G** - G is a function that takes a vector z and outputs x. So given a prior distribution $P_z(z)$, then a probability distribution $P_G(x)$ is defined by function G.
- **Discriminator D** - D is a function (actually a binary classification) takes x as inputs and outputs a scalar in range of $[0, 1]$ to tell the input x is faked or not.
- **V(G, D)** - The function V(G, D) is defined to be:

$$V(G, D) = E_{x \sim p_{data}}[log(D(x))] + E_{z \sim p_z}[log(1 - D(G(z)))] \tag{5}$$

  Defining V(G, D) like this allows the following missions:

  1. Given a G, $\max_D V(G, D)$ evaluates the difference between p_G and p_{data}.
  2. $\min_G \max_D V(G, D)$ picks a generator that minimize the difference.

2.3 Solve $\arg\max_D V(G, D)$

Given a G, thus distribution p_G is determined. We have

$$\begin{aligned} V &= E_{x \sim p_{data}}[log(D(x))] + E_{z \sim p_z}[log(1 - D(G(z)))] \\ &= E_{x \sim p_{data}}[log(D(x))] + E_{x \sim p_G}[log(1 - D(x))] \\ &= \int_x p_{data}(x) log(D(x)) dx + \int_x p_G(x) log(1 - D(x)) dx \\ &= \int_x [p_{data}(x) log(D(x)) + p_G(x) log(1 - D(x))] dx \end{aligned} \tag{6}$$

Because p_{data} and p_G are both fixed, for each input x, we can easily compute the result that maximizes $V(G, D)$:

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_G(x)} \tag{7}$$

2.4 Why $\max_D V(G, D)$ evaluates the difference of distributions?

$$\begin{aligned}
&\max_D V(G, D) \\
&= V(G, D^*) \\
&= E_{x\sim p_{data}}[log\frac{p_{data}(x)}{p_{data}(x) + p_G(x)}] + E_{x\sim p_G}[log\frac{p_G(x)}{p_{data}(x) + p_G(x)}] \\
&= \int_x p_{data}(x) log\frac{p_{data}(x) \times \frac{1}{2}}{(p_{data}(x) + p_G(x)) \times \frac{1}{2}} dx \\
&\quad + \int_x p_G(x) log\frac{p_G(x)}{\frac{1}{2}(p_{data}(x) + p_G(x)) \times \frac{1}{2}} dx \\
&= -2log2 + \int_x p_{data}(x) log\frac{p_{data}(x)}{\frac{1}{2}(p_{data}(x) + p_G(x))} dx \\
&\quad + \int_x p_G(x) log\frac{p_G(x)}{\frac{1}{2}(p_{data}(x) + p_G(x))} dx \\
&= -2log2 + KL(p_{data}||\frac{p_{data} + p_G}{2} + KL(p_G||\frac{p_{data} + p_G}{2}) \\
&= -2log2 + JS(p_{data}||p_G)
\end{aligned} \tag{8}$$

Therefore, we can see that $V(G, D^*)$ is the sum of a JS divergence and a constant $-2log2$. So it can evaluate the difference of distributions. And the smaller the $V(G, D^*)$ is, the more similar p_{data} and p_G are.

2.5 Final target - the generator

Our final aim is to achieve a generator G that have the same distribution p_G as p_{data}, which means the smallest divergence. So our optimal objective is

$$G^* = \arg\min_G \max_D V(G, D) \tag{9}$$

3 Tips for implementing a GAN

1. Due to the difficulty of computing the real expectation of distributions, we use the arithmetic mean of samples $\{x(1), ..., x^{(m)}\}$ and $\{z^{(1)}, ..., z^{(m)}\}$ to approximate it:

$$\frac{1}{m}\sum_{i=1}^{m}[log(D(x^{(i)})) + log(1 - D(G(z^{(i)})))] \tag{10}$$

2. Update the generator by descending the following stochastic gradient can be inefficient:

$$\nabla_{\theta_G}\frac{1}{m}\sum_{i=1}^{m} log(1 - D(G(z^{(i)}))) \tag{11}$$

   Because the equation indicates that the more real the generated image z is, the smaller the gradient is. That is, the gradient will be almost 0 at the begin of training and increase significantly when the generator is able to generate a good enough image! This will make

the training inefficient and uneasy to converge. So we usually use the gradient below as an alternative. More real the image, less the gradient.

$$\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} log(D(G(z^{(i)}))) \tag{12}$$

4 Works to do next week

Next week, I will start training a GAN using the lab's GPU server. I have already obtain my account but I am not familiar with the operations to work remotely using a terminal. So I may spend some time to learn how to use the GPU server.