Generative Adversarial Nets

Author: Zhang Xiaozheng Date:2019/11

Unlike some DNN or CNN which is used to do prediction or classification, Generative Model is used to generate data which is just looks like true data. At first , I didn't understand its application, but after see some articles, I find the Generative Model do have many applications, which is:

- When we face the case that our data set have many incomplete data, we need to handle these incomplete data to full fill our training.
- If we have some unclear photo, we can use generative model to generate a clear photo.
- We can use generative model to generate human voice.

So in this passage, the author propose a new theory to generate Generative model which is Generative Adversarial Nets.

To explain it clearly, we can take an example. Imagine that you are a counterfeiter, trying to produce fake currency and use it without detection. While the police is trying to analysis the fake currency in real currency. As for Generative Adversarial Nets, we define a generative model G and a discriminate model D. What G does is just like a counterfeiter and D is the police.

We use ***x*** to represent the real data which obeys distribution p_data(x), ***z*** to represent the noise which obeys distribution p_z(z). G(z|θ_g) mapping z to real data x which obeys distribution p_g(x). And what D(x) outputs is the probability that x is from real data or fake data. The higher D(x) outputs the more likely x is from real data.

So D's goal is to maximize:

$$E_{x \sim p_{data}}[log(D(x))]$$

What's more we also want to minimize D(G(z)), to let the goal all be to maximize some equation. We rewrite our function like:

$$E_{z \sim p_z}[log(1-D(G(z)))]$$

At the same time, what G want to do is to minimize last equation. So we can rewrite our function:

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

It just like G and D are playing minimax game.

The prove is in another PDF.