Student ID : 20194293

Name : Go, Kyeong Ryeol

[AI 502] Generative Adversarial Nets

1. Paper Summary

Deep generative models have had less of an impact, due to the difficulty of approximating many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies, and due to difficulty of leveraging the benefits of piecewise linear units in the generative context. Therefore, the author proposed a new generative model named as 'Generative Adversarial Nets (GAN)' with estimation procedure through the minimax two-player game.

Like GAN, Variational Auto-Encoder (VAE) pair a differentiable generator network with a second neural network, but unlike GAN, this is a recognition model that performs approximate inference. As a result, VAE requires differentiation through the hidden units, and thus cannot have discrete latent variables while GAN require differentiation through the visible units, and thus cannot model discrete data.

Following is the objective function of GAN. Here, the generative model $\mathrm{G}(\mathrm{z};\theta_\mathrm{g})$ is pitted against its adversary which is the discriminative model $\mathrm{D}(\mathrm{x};\theta_\mathrm{d})$ that learns to determine whether a sample is from the model distribution or the data distribution. (Both models are comprised of multilayer perceptron.)

$$\min_{\mathrm{G}}\max_{D} V(D,G) = \mathbb{E}_{x\sim p_{data(x)}}[\log D(x)] + \mathbb{E}_{\mathrm{z}\sim \mathrm{p_z(z)}}\left[\log\left(1-D\left(G(z)\right)\right)\right]$$

Competition in this game drives both teams to improve their methods until the counterfeits are indistinguishable from the genuine articles. Theoretically, the design of the network is justified as it is shown that for a fixed generator, the optimal discriminator is $\mathrm{D}^*_\mathrm{G}(x) = \frac{\mathrm{p_{data(x)}}}{\mathrm{p_{data(x)}}+p_g(x)'}$ so that the virtual training criteria $\mathrm{C(G)} \coloneqq \max_{\mathrm{D}} V(D,G)$ can be expressed as following where the optimum is achieved at value $-\log 4$ when $\mathrm{p_g} = p_{data}$.

$$\mathrm{C(G)} = -\log 4 + KL(p_{data}||\frac{p_{data}+p_g}{2}) + KL(p_g||\frac{p_{data}+p_g}{2}) = -\log 4 + 2\cdot \mathrm{JSD}(\mathrm{p_{data}}||p_g)$$

In practice, we must implement the game using an iterative, numerical approach. Optimizing D to completion in the inner loop of training is computationally prohibitive, and on finite datasets would result in overfitting. Instead, we alternate between k steps of optimizing D and one step of optimizing G. This results in D being maintained near its optimal solution, so long as G changes slowly enough. Moreover, rather than training G to minimize $\log\left(1-\mathrm{D}\left(\mathrm{G}(\mathrm{z})\right)\right)$, we can train G to maximize $\log \mathrm{D}\left(\mathrm{G}(z)\right)$. This objective function results in the same fixed point of the dynamics of G and D but provides much stronger gradients early in learning.

2. Discussion

The main advantages of GAN are that only backprop is used to obtain gradients and that no inference is needed during training. Also, wide variety of functions can be incorporated into the model and practically intriguing, it can represent very sharp, even degenerate distributions. However, the disadvantages of GAN network are that there is no explicit representation of $\mathrm{p_g(x)}$ and that the discriminator must be synchronized well with the generator during training. In particular, in order to avoid "The Helvetica Scenario" where the generator collapses too many values of z to the same value of x to have enough diversity to model $\mathrm{p_{data}}$, the generator must not be trained too much without updating the discriminator. What may encourage the model to be trained in a stable manner? Furthermore, how can we change the model structure so that GAN can perform inference on latent representation of data?