Task 7 : Generative Adversarial Networks

Sanjeev Narayanan

26th March 2018

Generative adversarial networks (GANs) are deep neural network architectures that compete aginst each other. Introduced by *Goodfellow*, et all, GANs are an alternative to Variationaal Autoencoders for learning latent spaces of images. They enable the generation of fairly realistic synthetic images by forcing the generated images to be statistically almost indistinguishable from real ones.

It is important to understand discriminative algorithms (DAs) to grasp the concept of GANs [1]. As the name itself suggests, DAs try to classify input data; that is, given the features of a data instance, they predict a label or category to which that data belongs. GANs can be thought of an opposite mechanism to DAs. While DAs map features to labels, GANs predict features given a certain label. A simple spam filter example can be used to distinguish between the two approaches.

- Given an email, a DA could predict whether the message is spam or not.
- What a GAN does is, assuming this email is spam, how likely are these features?

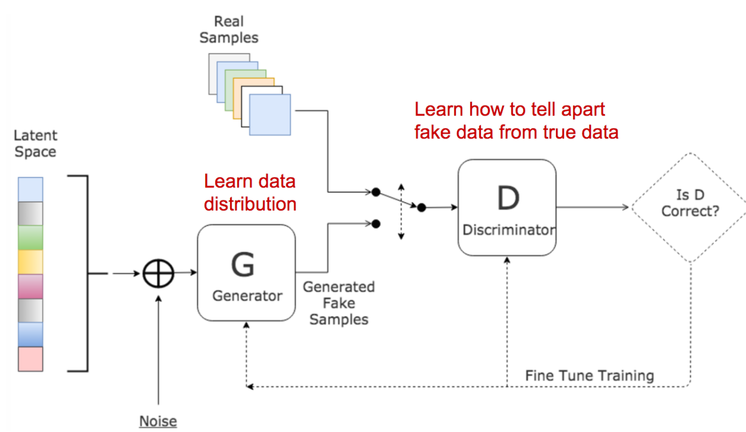

Figure 1: Generative Adversarial Network

Consider the discriminator to be a function D that takes x as input and uses $\theta^{(D)}$ as parameters. The generator is defined by a function G that takes z as input and uses $\theta^{(G)}$ as parameters. The training process consists of simultaneous Stochastic Gradient Descent.On each step, two minibatches are sampled: a minibatch of x values from the dataset and a minibatch of z values drawn from the model's prior over latent variables. Then two gradient steps are made simultaneously: one updating $\theta^{(D)}$ to reduce $J^{(D)}(\theta^{(D)}, \theta^{(G)})$ and one updating $\theta^{(G)}$ to reduce $J^{(G)}(\theta^{(G)}, \theta^{(D)})$. Usually, Adam optimizer is used for both the cases. Without the right hyperparameters, network architecture, and training procedure, the discriminator can overpower the generator, or vice-versa. In one common failure mode, the discriminator overpowers the generator, classifying generated images as fake with absolute certainty. When the discriminator responds with absolute certainty, it leaves no gradient for the generator to descend. This is partly why discriminators are built to produce unscaled output rather than passing its output through a sigmoid function that would push its evaluation toward either 0 or 1. In another common failure mode, known as mode collapse, the generator discovers and exploits some weakness in the discriminator. This is recognized when the GAN generates many very similar images regardless of variation in the generator input z. Mode collapse can sometimes be corrected by "strengthening" the discriminator in some way—for instance, by adjusting its training rate or by reconfiguring its layers [2].

The main formulation of GAN is given by the minimax cost function. Here, costs are requires to be same in magnitude for both generator and discriminator [3].

$$J^{(G)} = -J^{(D)} \tag{1}$$

Since $J^{(G)}$ is tied to $J^{(D)}$, let

$$V(\theta^{(D)}, \theta^{(G)}) = -J^{(D)}(\theta^{(D)}, \theta^{(G)}) \tag{2}$$

The minimax game is mostly of interest because it is easily amenable to theoretical analysis.

$$\theta^{(G)*} = argmin_{\theta^{(G)}} max_{\theta(D)} V(\theta^{(D)}, \theta^{(G)}) \tag{3}$$

Learning in this approach resembles minimizing the Jensen-Shannon divergence between the data and the model distribution, and that it converges to its equilibrium if both, generator's and discriminator's policies can be updated directly in function space.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[2] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[3] Jon Bruner and Adit Deshpande. Generative adversarial networks for beginners.