GAN

Tengji Zhang

March 17, 2018

Generative Adversarial Networks - A Brief Summary

Generative Adversarial Networks, most commonly known with its abbreviation GAN, is one of the most powerful model in the field of Deep Learning. It is introduced by computer scientist Ian Goodfellow and his colleagus in University of Montreal in the year 2014. [1] As its name might suggest, this model is invented primarily to generate new data points that are similar to, or in more technical terms, have similar probabilistic distributions as, the sample data provided for the training of the models. To understand the concept of the model, we here do assume a basic level of understanding of the subject of deep learning.

[1] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Joshua (2014). "Generative Adversarial Networks". arXiv: 1406.2661

GAN in essence is a very specific kind of neural network architecture that mainly comprises of two parts, the discriminator and the generator. Here is a visualization of its architecture.[2]

[2] DL4J. "GAN: A Beginner's Guide to Generative Adversarial Networks". Accessed on March 17th, 2018. Website

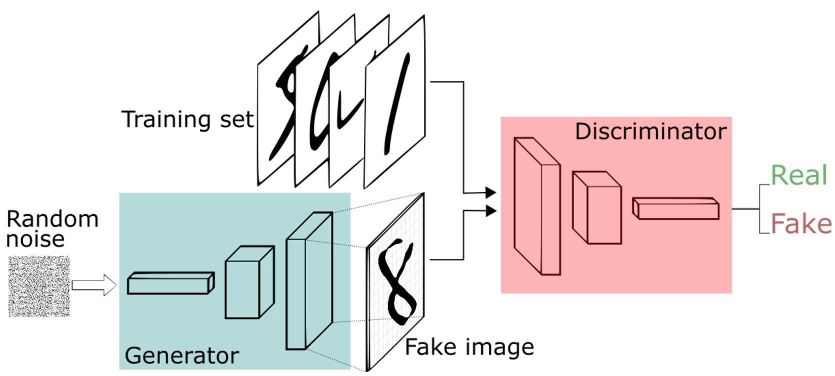

Figure 1: GAN

Remember that the role of the generator is to generate new data samples as similar to the data set we have as possible. For example, we can have data set that have nothing but images of puppies, we want the generator know how to generate new images of puppies that look very realistic. The role of the discriminator, however, is the detective; it needs to tell whether a given image is genuine images similar to the data set, or synthesized images produced by the generator. The input of the generator is simply a random noise signal and its output, the synthesized data point, will then be passed as the input to the descriminator, who, also supplied with the genuine data from the training data set, will try to tell if the given input data is genuine or generated and output its judgement as the result.

The innovations in GAN's architecture can only be fully appreciated once we delve into the details on how it is trained with the data set. Even though the generator is supposed to generate data similar to the ones in the data set, it will not be supplied a single data point directly from the data set in the training process. At the beginning, it is outputting nothing more than randomly transformed random noise signals. On the other end, the discriminator will from the very beginning be supplied both the data from the data set and the ones randomly generated by the generators. However, it will not be told which comes from which; once it gives its judgement, it will then be told correct or false by the model trainer, and then tune its parameters to give more accurate judgements. The generator will also use the accuracy of the judgement of the generator as training data; however, it will be trained in such a way that it aims to minimize the accuracy of the discriminator. So even though the generator and the discriminator are in one single set of neural network, the GAN, they have totally opposite metrics to optiimze for. In plain English, they are competing with each other; the generator is trying to fool the discriminator with its fake generated data, while the discriminator is trying not to be fooled. Hence comes the keyword "adversarial" in its name.

Now we know the lost functionwe are trying to optimize for generator and discriminator, but how are we going to train them? Obviously we cannot train them in one pass just like we train other models. That means we are going to dump all the data into the discriminator, ask the generator to come up with equal amount of generated data, let the discriminator judges, and then optimize the parameters of both with the results. There is only one round of competition between the two models and that is not enough, no matter how much data we suuply in the data set. Instead, we are going to break the entire data set into smaller batches, and then go through the above steps, training both the generator and the discriminator in each batch of data. Here is the detailed algorithm from the paper of Ian Goodfellow[3]:

[3] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Joshua (2014). "Generative Adversarial Networks". arXiv: 1406.2661

Training GAN

The rest of the implementation will be similar to other neural network architectures and we will thus omit them here.

Researchers and developers alike have done significant amount of experimentations on GAN with all kind of data sets, though mostly limited to the categories of objects where there are huge amount of relatively homogeneous pictures available without copyright in the internet. Whether the researchers can achive good result also directly depends on the size and image quality of the data set. Unsurprisingly, so far we have seen the best results of generated pictures with photos of celebrities. Here is a stunning result from a group of researchers from Nvidia[4]:

Generated Celebrities Photos

The results are virtually indistinguishable from real photos.

[4] Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen. "Progressive growing of gans for improved quality, stability, and variation." arXiv:1710.10196 (2017).