

Understanding Generative Adversarial Networks Mathematically

Samrat Sahoo

UT Arlington

Advisor: Professor Won Hwa Kim

Email: won.kim@uta.edu

Abstract

Generative Adversarial Networks have become a large deal within the machine learning world. In this paper we take a look at the full mathematical source behind how generative adversarial networks work. We start from the very basics, giving an intuitive understanding and defining the different parts of a generative adversarial networks all the way up to understanding the loss function both mathematically and conceptually.

Keywords: Generative Adversarial Network (GAN), Loss Functions, Machine Learning

1 Defining the Generator

A Generative Adversarial Network is made up of 2 different parts: The discriminator and the generator. Given the generator through the function:

$$G(z) = x \quad (1)$$

The generator function is represented through $G(z)$ where z , the parameter of the generator, is the noise that gets added to a data set. The output variable, x , is representative of the new generated data after the noise has been applied. [2]

2 Defining the Discriminator

The discriminator is the second part of a generative adversarial network. The discriminator can be represented through the function:

$$D((G(z))) \quad (2)$$

The discriminator function then outputs a value based on its prior knowledge after being trained on data sets. The discriminator outputs a scalar value, such that:

$$0 \leq D(G(z)) \leq 1 \quad (3)$$

If $D(G(z))$ yields a value closer to 1 then the generator has done its job and produced an image that does not seem to be synthetic. If $D(G(z))$ is closer to 0 then the discriminator recognizes the image as synthetic data. This process is iterated until the discriminator can no longer differentiate between a synthetic and real image, hence, outputting numbers closer to 1. [2]

3 The Minimax Loss Function

A traditional Generative Adversarial Network consists of 2 loss functions. One loss function is for the discriminator and one for the generator. Combined, they create the Minimax loss function. The Minimax loss function is define as follows:

$$E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))] \quad (4)$$

In this loss function, there are several parts. $D(x)$ is representative of the discriminator's value for real data. E_x is representative of the expected value for all real data. $D(G(z))$ is the discriminator's value for generated data, given a noise z into the generator. E_z is representative of the expected value for all generated data. $E_x[\log(D(x))]$, is needed to better recognize real images better while the $E_z[\log(1 - D(G(z)))]$ is needed to better recognize the generated images. [1]

4 Minimax Loss Function Objective

The ultimate goal of the generator of the GAN is to minimize the Minimax loss function while the discriminator tries to maximize the loss function. This can also be understood intuitively because as $D(G(x))$ approaches 1, the discriminator believes that the generated image is real. At the same time, $E_z[\log(1 - D(G(z)))]$ approaches ∞ , meaning the Minimax loss function is being maximized. On the flip side, if $D(G(z))$ approaches 0, the discriminator recognizes that the generated image is fake. This would cause $E_z[\log(1 - D(G(z)))]$ to reach approach 0 as well, meaning the Minimax loss function is being minimized. [1].

5 The Wasserstein Loss Function

The Wasserstein Loss Function is another loss function that is commonly used with Generative Adversarial Networks. GANs that do use the Wasserstein Loss Function are known as WGAN. The Wasserstein Loss Function has two different parts to it. The discriminator loss function, also known as the critic, is defined as follows:

$$D(x) - D(G(z)) \quad (5)$$

The generator loss function is defined as follows:

$$D(G(z)) \quad (6)$$

6 Understanding Wasserstein Loss

It is easiest to understand the Wasserstein Loss function when looked at separately. Looking solely as the critic function, the discriminator aims to maximize the critic value by minimizing $D(G(z))$. On the other hand, the generator aims to maximize the $D(G(z))$, therefore causing this adversarial nature. [3]

7 Concluding Remarks

By understanding the mathematical reasoning behind Generative Adversarial Networks, a more intuitive understanding of their functionality is reached. Throughout this paper, we not only looked at the greatest simplifications of Generative Adversarial Networks but also the worked our way to understanding the basis of the loss functions both mathematically and conceptually. Through this approach, the backend of the Generative Adversarial Networks is more fully understood.

8 References

- [1] Blerta, Lindqvist. "Minimax Defense against Gradient-based Adversarial Attacks" Rauf Izmailov, v1, 4 February 2020, 1. arxiv, arXiv:2002.01256v1
- [2] Ian, Goodfellow. "Generative Adversarial Networks." , Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, v1, 10 June 2014, 1. arxiv, arXiv:1406.2661v1.
- [3] Martin, Arjovsky. "Wasserstein GAN" Soumith Chintala , Léon Bottou, v3, 26 January 2018, 1. arxiv, arXiv:1701.07875v3