Challenges, Corresponding Solutions, and Applications of Generative Adversarial Networks

Hongbo Zhong*

Alberta University, Edmonton, Canada

*Corresponding author: hongbozhong@outlook.com

Abstract. Generative adversarial networks (GANs) have become a popular deep learning framework in the field of artificial intelligence. Researchers have developed various types of GANs, such as Conditional GANs (CGANs), Mode Dropping GANs (MDGANs), and Wasserstein GANs (WGANs), which have been applied to many different fields. Despite the success of GANs in various tasks, there are still some challenges that need to be addressed. For example, model collapse, non-convergence, and diminished gradient could hinder the training of GANs. In this paper, the authors first introduce the basics of GANs before highlighting a few common problems associated with GANs and their corresponding solutions. For instance, they discuss techniques such as mini-batch discrimination and batch normalization that can help mitigate model collapse and diminish gradient problems. Additionally, they cover methods that can improve the convergence rate and stability of GANs, such as using alternative loss functions and incorporating regularization algorithms. Finally, this paper briefly introduces some recent applications of GANs in image generation, video prediction, and other areas. Despite the challenges that still exist, GANs have shown great promise in various fields, and with further research, GANs have the potential to become more robust and efficient models for generating high-quality synthetic data.

Keywords: Artificial intelligence, Generative Adversarial Network, Corresponding solutions.

1. Introduction

Generative Adversarial Network (GAN) is a machine learning model used to generate data, consisting of two neural networks: the generator and the discriminator. The generator receives a random noise vector and gradually adjusts its parameters through backpropagation to generate fake data similar to the training data. Meanwhile, the discriminator also gradually adjusts its parameters through backpropagation to distinguish real data from the generated fake data and provide classification results. This cycle iterates to improve the generator's ability to generate fake data and the discriminator's accuracy in classification, ultimately reaching a dynamic equilibrium state [1].

GAN models have shown powerful generating capabilities in the fields of image and video generation. For example, using GAN technology, we can generate realistic images, videos, audios, and even various styles of non-image data such as text. In addition, GAN has been applied in areas such as image restoration, data augmentation, and super-resolution, demonstrating broad application prospects.

However, GAN models are challenging to train and require avoiding issues such as model collapse. For example, when the generator is too powerful, the discriminator may fail to distinguish between real and fake data. Furthermore, the model may experience problems such as gradient vanishing or exploding. Therefore, in practical applications, it is necessary to design and adjust reasonably according to specific situations in order to fully utilize the advantages of the GAN model. In this review, we restrict our focus on model collapse, non-convergence and evaluation metrics and limited data.

2. GAN Architecture

It has no explicit hypothesis and estimates the distribution of the data directly. And compared to earlier implicit density models, GANs do not need to use ancestor sampling or a Markov chain, which

has a negative effect on performance and can narrow the range of applications. As shown in Figure 1.

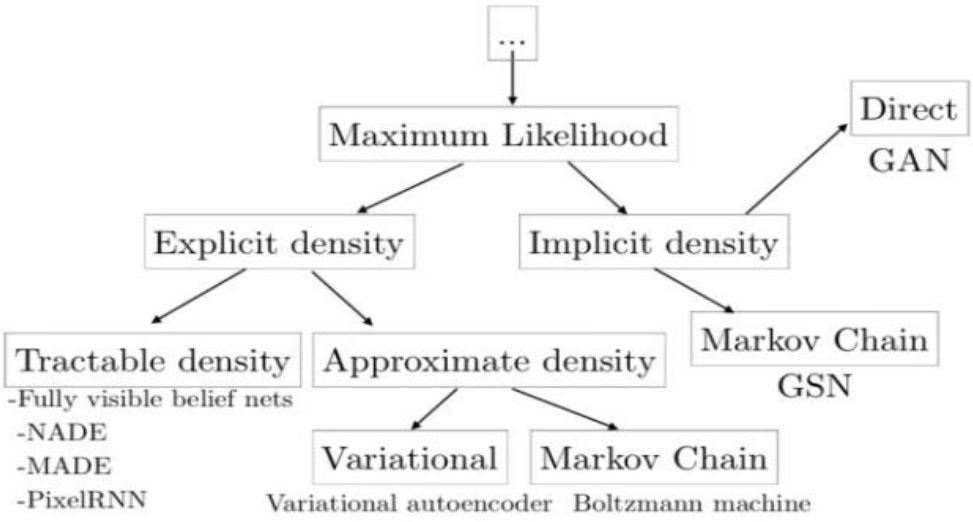

Figure 1. Relationship structure diagram.

The basic generator takes a input noise variable z then map it to data space as G(z,θ1) where G is a multi-layer neural network with parameters θ1. And D(x, θ2) is also a multi-layer neural network that represents the probability that x is the true data sample instead of generated data by G. Then D is trained to maximize the probability of predicting the correct label of G(z) and at the same time G is trained to minimize what D tries to maximize like a minimax two-player game.
The adversarial idea could be formulated as below:

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

Where the G(z), D(x) is the abbreviation of G(z, θ1) and D(x, θ2). And the $P_{data}(x)$, $P_z(x)$ represents real data distribution and noise distribution respectively.

3. Challenges

3.1. Non-convergence and Instability

The non-convergence and instability of the original GAN occurs due to the loss function (1). From formula (1), we could infer the discriminator aims to minimize the loss:

$$-E_{x\sim P_{data}(x)}[log\ D\ (x)] - E_{x\sim P_g(x)}[log(1 - D(x))] \quad (2)$$

Where $P_g(x)$ is the generated data space by generator G.

And for a specific sample x, it is possible for it to come from both real data space and generated data space. Then what it would contribute to the loss in (2) is:

$$L = -P_{data}(x)\ log\ D\ (x) - P_g(x)\ log(1 - D(x)) \quad (3)$$

Then we have the best discriminator:

$$\frac{dL}{dD(x)} = -\frac{P_{data}(x)}{D(x)} + \frac{P_g(x)}{1 - D(x)} = 0$$

$$=> D^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)} \quad (4)$$

Substitute formula (4) into formula(2) and we have

$$E_{x\sim P_{data}(x)} \frac{2P_{data}(x)}{P_{data}(x)+P_g(x)} + E_{x\sim P_g(x)} \frac{2P_{data}(x)}{P_{data}(x)+P_g(x)} - 2\,log\,2 \tag{5}$$

And we could transform (5) to the form of Jensen-Shannon divergence, then it could be written as

$$2JS(P_{data}||P_g) - 2\,log\,2 \tag{6}$$

Until now, we could conclude that assuming the best generator, the loss in GAN is equivalent to minimizing the JS divergence of the real distribution P_{data} and the generated distribution P_g. But the divergence indicates a problem since if Pr and Pg does not overlap nearly, then this divergence would be a constant log2 which could result in 0 gradient.

Finally, according to analysis above, we have figured out why original GAN is not stable and sometimes the non-convergence occurs. And some variants of GAN have similar problems due to their loss functions.

3.2. Model Collapse

Theoretically Mode collapse problem occurs in GANS when the max-min solution is not in accordance with the min-max solution. The result is that G takes different noises but persist to produce the same sample.

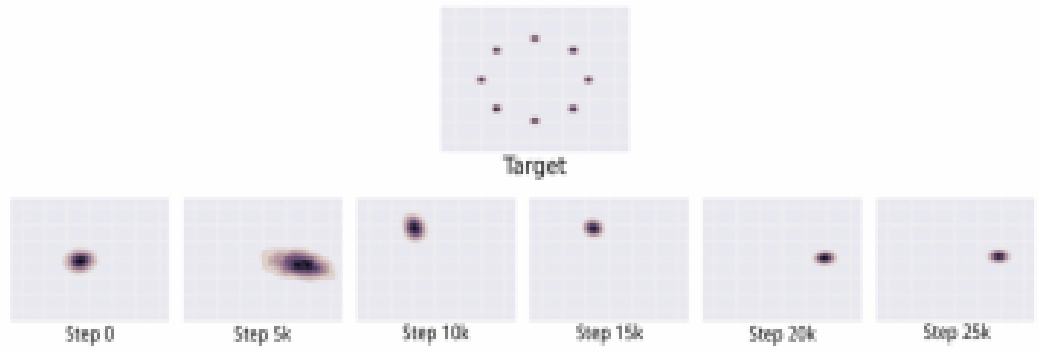

Figure 2. Example diagram

As the figure 2 shows, the results keeping cycling between different modes and have difficult in producing other target points.

In general, the former one is actually the same problem in section 3.4 which we would discuss later and the latter one is usually caused by the insufficient penalty for lack of diversity in loss function.

3.3. Evaluation Metrics

Although GANS have been used extensively in many applications of both supervised and unsupervised learning, the performance evaluation process still depends on visual evaluation through human eyes. Visual examination could miss discrete features, cost a large amount of time and is probably biased. But in order to designing better models, the development of appropriate evaluation metrics is required so that right conclusions could be drawn after training.

3.4. Limited Data

Even though the various variants of GANs that are currently innovative and used in many areas, the success of GANS is highly dependent on large amounts of training data which are difficult or inconvenient to collect. This problem is becoming increasingly common with the development of GANs.

4. Solutions

4.1. Wasserstein Distance

Since as the analysis in sections 3.1 and 3.2, we know the loss function is the key problem of non-convergence and model collapse, the solution to them is definitely a new loss function.
Wasserstein distance [2] is defined as below to overcome instability, non-convergence and model collapse

$$W(P_{data}, P_g) = inf_{\gamma \sim \prod(P_{data}, P_g)} E_{(x,y)\sim\gamma}\left[||x-y||\right] \tag{7}$$

Where $\prod(P_{data}(x),\ P_z(x)\ )$ is the the joint probability function of $P_{data}(x)\ P_z(x)$.For every true sample x and generated sample x in the joint probability function, we could get the expectation of the distance ||x-y||. And then the Wasserstein distance is expressed as the lower bound of this expectation.

This distance would provide meaningful gradient even if the Pr, Pg do not overlap at all which could improve the stability and ensure the model to converge. But we could not use Wasserstein distance directly as loss function since it is impossible to calculate it directly. But it could be transformed with Lipschitz constant to the below two losses for generator and discriminator respectively [2].

$$-E_{x\sim P_g}[f_w(x)] \tag{8}$$

$$E_{x\sim P_g}[f_w(x)] - E_{x\sim P_{data}}[f_w(x)] \tag{9}$$

Besides, while fitting the Wasserstein distance, the task becomes a regression task instead of classification task, so we do not use sigmoid function anymore.

4.2. Gradient Penalty

While using Wassesrtein distance, it uses wight clipping to implement Lipschitz limit indirectly. But according to statement in [3], it would cause two problems

The discriminator would tend to learn a very simple map function and that is a huge waste for a deep learning networks.

If the clipping threshold is not set properly, the gradient would easily disappear or booms.

To solve these two issues, it is effective not to add Lipschitz in whole space. Instead it is enough to grip the area where generated sample gather, area where real data gather and the area between these two areas[4].

To be specific,

$$x_r \sim P_{data}, x_g \sim P_g, \alpha \sim Uniform[0,1]$$

$$x^* = \alpha x_r + (1-\alpha)x_g$$

$$L(D) = E_{x\sim P_g}[f_w(x)] - E_{x\sim P_{data}}[f_w(x)] - \lambda E_{x^*\sim P_x}[||\nabla_{x^*} D(x^*)||_2 - 1]^2 \tag{10}$$

L(D) is the new loss of the discriminator and then the gradient penalty would obviously increase the training speed.

4.3. Inception Score

Inception score is proposed to evaluate the effective of GAN model in 2016. It is defined as:

$$exp(\ E_x KL(p(y|x)||p(y))) \tag{11}$$

Where the p(y|x) is the conditional distribution by applying inception model to every image.p(y|x)should have low entropy if its corresponding Images contain meaningful objects.

4.4. Regularization

Apart from usual data augmentation, a new method could be integrated with augmentation to improve the performance: regularization on the basis of limited data [5,6].

$$\max_{D} V_D, V_D = E_{x\sim\Gamma}[f_D(D(x))] + E_{x\sim p_x}[f_G(D(G(z)))] \tag{11}$$

$$\min_{G} L_G, L_G = E_{x\sim p_x}[g_G(D(G(z)))] \tag{12}$$

The way of calculating the αR and αF are provided in the supplementary document [7,8]. Then the identical objective LG described in the (2) above is used to train the generator while minimizing the regularized objective LD for the discriminator [9]:

$$\min_{D} L_D, L_D = -V_D + \lambda R_{LC}(D) \tag{13}$$

And it is empirically demonstrated that this regularization method with appropriate weight λ efficiently improves the generalization under limited training data [10].

5. Applications

5.1. Image Processing and Computer Vision

5.1.1 Super-resolution

The first powerful super-resolution GAN is SRGAN[11] proposed in 2017. It can produce an image with an image with 4 times higher resolution than the original input image. Soon after, ESRGAN [12] came out as an upgrade of SRGAN that increases the reality predicted by the discriminator. Additionally, TGAN [13], deep tensor generative adversarial networks, is also innovated to generate large and high-quality images.

5.1.2. Image Generation

LR-GAN was introduced in 2017 as a adversarial model for generating images that can take into account both scene structure and context [14]. Unlike previous methods that attempt to generate images without considering that images are two-dimensional projections of a three-dimensional visual world containing many structures, this unsupervised model explicitly encodes that structure.. It combines foregrounds on the background recursively for generating more realistic images and outperforms DCGAN. The ability in image generation could be used in many fields such as anime character generation, face aging, 3D image transformation [15]. The discriminative output plays as a reward when the reinforcement learning reward is passed back to the intermediate stages by Monte Carlo search. It has shows superiority both on synthetic data experiments and real-world scenarios such as poems, speech language and music generation. GANs have also been applied to medical fields such as drug discovery Chem-GAN [16], medical image processing for measuring organ functions [17], dental restorations [18].

6. Conclusion

Generative Adversarial Networks (GANs) have become a significant breakthrough in the field of deep learning and artificial intelligence. The most significant significance of GANs is their ability to generate realistic data from scratch, which can be used for various purposes like data augmentation, image restoration, and synthesis. GANs work by making two neural networks compete against each other; one generates fake data while the other tries to discern whether it is real or fake. Through this process, both networks learn from each other, allowing the generator network to continually improve its output to become more realistic, while the discriminator network also improves its ability to tell apart the real from the fake data.One of the major applications of GANs is in image and video generation. With their ability to create realistic images,GANs have become an essential tool for

graphic designers, game developers, and other artists working with digital media. Additionally, GANs have been used to generate synthetic data for machine learning models, which can help overcome data scarcity issues, leading to more robust models. Moreover, GANs have the potential to revolutionize many industries, such as healthcare and finance. For instance, medical professionals might be able to use GANs to generate synthetic medical images or 3D models to better understand complex diseases, while financial institutions could use GANs to simulate market scenarios and predict risks.In conclusion, GANs have immense potential for innovation and impact in different industries, including entertainment, healthcare, and finance. As GANs continue to evolve, their applications are likely to become more widespread across diverse fields, leading to significant advancements in artificial intelligence and machine learning.

References

[1] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. arXiv preprint arXiv:1406.2661.

[2] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 214-223). PMLR.

[3] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of Wasserstein GANs. arXiv preprint arXiv:1704.00028.

[4] Goodfellow, I. J. (2016). NIPS 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160.

[5] Saxena, D., & Cao, J. (2022). Generative adversarial networks (GANs): Challenges, solutions, and future directions. ACM Computing Surveys (CSUR), 54(3), 63.

[6]

[7] Tseng, H. Y., Jiang, L., Liu, C., Yang, M. H., & Yang, W. (2021). Regularizing generative adversarial networks under limited data.

[8] Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., & Aila, T. (2020). Training generative adversarial networks with limited data. In Advances in Neural Information Processing Systems (pp. 7124-7135).

[9] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training GANs.

[10] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the inception architecture for computer vision. arXiv preprint arXiv:1512.00567.

[11] Laine, S., & Aila, T. (2017). Temporal ensembling for semi-supervised learning. In International Conference on Learning Representations.

[12] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W., Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[13] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., & Loy, C. C. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV).

[14] Ding, Z., Liu, X. Y., Yin, M., Liu, W., & Kong, L. (2019). Tgan: Deep tensor generative adversarial nets for large image generation. arXiv preprint arXiv:1901.09953.

[15] Yang, J., Kannan, A., Batra, D., & Parikh, D. (2017). LR-GAN: Layered Recursive Generative Adversarial Networks for Image Generation.

[16] Yu, L., Zhang, W., Wang, J., & Yu, Y. (2017). SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient.

[17] Benhenda, M. (2017). ChemGAN challenge for drug discovery: Can AI reproduce natural chemical diversity? arXiv preprint arXiv:1708.08227.

[18] Dai, W., Doyle, J., Liang, X., Zhang, H., Dong, N., Li, Y., & Xing, E. P. (2017). Scan: Structure correcting adversarial network for chest xrays organ segmentation. arXiv preprint arXiv:1703.08770.

[19] Hwang, J. J., Azernikov, S., Efros, A. A., & Yu, S. X. (2018). Learning beyond human expertise with generative models for dental restorations. arXiv preprint arXiv:1804.00064.