International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 5.22 (SJIF-2017), e-ISSN: 2455-2585
Volume4, Issue 7, July-2018

Review on Generative Adversarial Networks

Mr. Jaynil Patel[1], Mr. Shivang Pandya[2], Prof. Vatsal Shah[3]

[1]IT Department, Birla Vishvakarma Mahavidyalaya,
[2]IT Department, A.D. Patel Institute of Technology,
[3]IT Department, Birla Vishvakarma Mahavidyalaya,

Abstract*— In the last few years, a type of generative model known as Generative Adversarial Networks (GANs), has achieved tremendous success mainly in the field of computer vision, image classification, speech and language processing, etc. GANs are the models which are used to produce new samples which have similar data distribution as of the training dataset. In this review paper, we will first introduce the idea behind the GANs, followed by a brief overview of various types of GANs as well as comparing it with different generative models. Then, we will discuss the application range and finally the future work with its associated research frontiers.

Keywords*— GAN, Discriminator, Generator, Nash equilibrium, Convolutional, Recurrent, Laplacian, NLP, Wasserstein, SAGAN, ChemGAN

I. INTRODUCTION

With the growing research in artificial intelligence, the ability of machines to mimic the human nature increases exponentially. But in order to create a general-purpose AI or Artificial General intelligence (AGI), researchers are first needed to bring forth new machine learning algorithms, each of which is well curated for performing a specific task. One such algorithm which was introduced by Goodfellow et al. [1] was Generative Adversarial Networks (GANs). The basic principle of GANs is inspired by two-player zero-sum game, in which the total gains of two players are zero, and each player's gain or loss of utility is exactly balanced by the loss or gain of the utility of another player [2].

GANs comprises of two models. First is a generative model (G), which tries to generate new instances which are completely different from that of the real dataset and are never seen before by the generator but have similar data distribution to real dataset. The generator is given a random noise, which it uses to create new samples obeying the distribution of real data. The second model is a discriminator model (D). The discriminator's job is to look at the distribution which could have come from the original dataset or through the generator, and to estimate the probability that the sample is from the original dataset or through the generator as shown in Fig. 1.

Both the models can be any machine learning model but are generally deep neural networks. The generator learns with the feedback from the discriminator as it does not have any access to the real dataset. On the other hand, the discriminator has access to both the real data samples and the samples drawn by the generator (fake samples). The error for the discriminator is provided by how many times, the discriminator classifies a fake synthesized sample, from the generator, to be from the real dataset. The same can be a measure of error for the generator as well, for instance, how many times the fake sample was classified as fake by the discriminator.

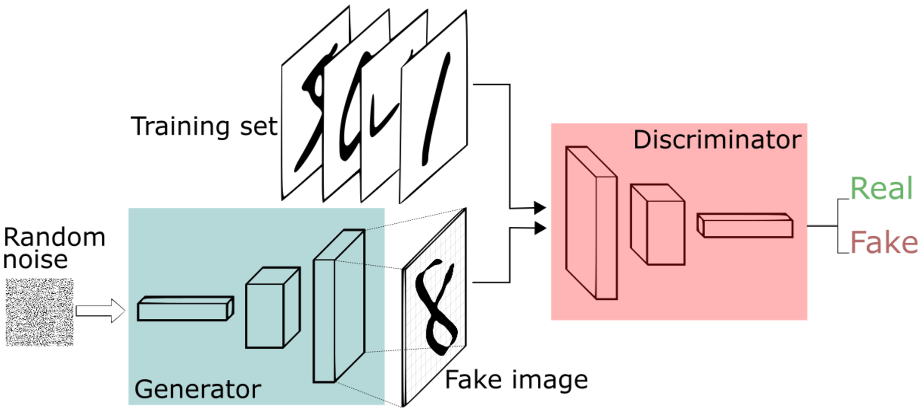

Fig 1. Training process for a GAN involving the discriminator (D) and the generator (G). [3]

In other words, the training procedure for generator is to maximize the probability of discriminator making a mistake. This framework corresponds to a minimax two-player game [1], because these two networks are fighting over one number. One of them wants the number to be high and one of the wants the number to be low. That number is the error rate of the discriminator. So, the discriminator wants a low error rate and the generator wants a high error rate. Thus, the optimization goal is to reach a Nash equilibrium [4]. Moreover, there is a trade-off between the learning rate of both the models. There must be separate learning rates for discriminator and generator so that neither generator or discriminator become way better than the other.

II. Types of GANs

A. Conditional GAN (CGAN)

This is a type of supervised learning GAN, which can be prolonged to a conditional model if both the discriminator and the generator are conditioned to some additional auxiliary information *y*. In the generator the prior input noise $p_z(z)$, and y are combined in joint hidden representation, and the adversarial training framework allows for considerable flexibility in how this hidden representation is composed [5]. This information can consist of class labels or data from other modalities. Its network architecture is multilayer perceptron (MLP). The objective here is similar to a minimax game but conditioned on extra information.

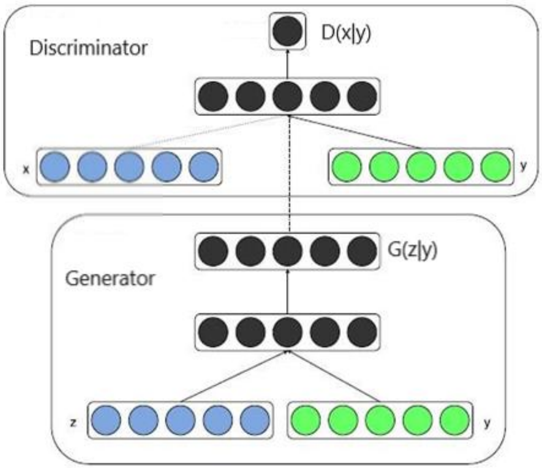

Fig 2. The structure of a Conditional Generative Adversarial Network (CGAN)

B. Deep Convolutional GAN (DCGAN)

This is one of the most popular types of GANs today which are employed for unsupervised learning. In this type of GAN, Convolutional Neural Networks (CNN) are used in place of the MLP with some additional constraints. With the introduction of DCGAN, the gap between the success of CNNs for supervised and unsupervised learning have been reduced [6].

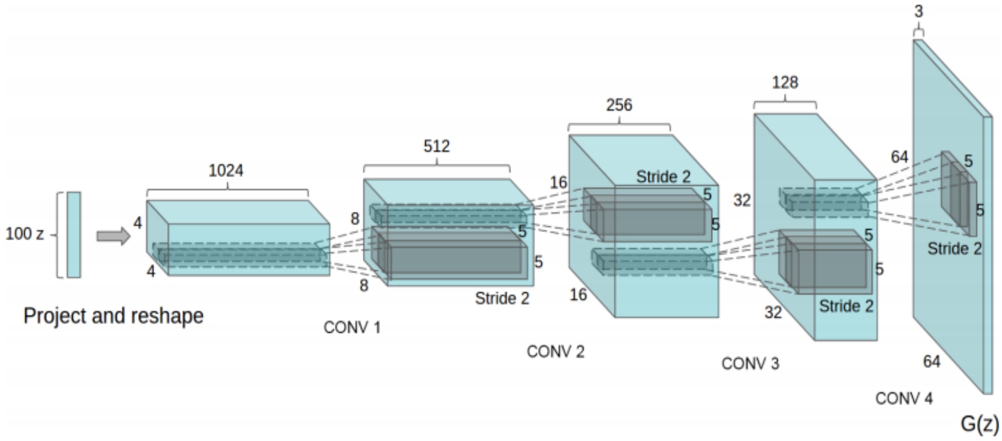

Fig 3. Architecture of the Generator in a DCGAN [6]

The architecture of the discriminator is mostly just the opposite of that of the generator, i.e., it takes in an image and produces 2 numbers (which are just the probabilities of the image being fake or not) [7]. In the discriminator, the forward process consists of the *Conv Transpose* or *Deconv* operation at every layer [7].

C. Laplacian Pyramid GAN (LAPGAN)

This is again a GAN employed for unsupervised learning which Denton et al. (2015) proposed for the generation of images in a coarse-to-fine fashion using cascade of convolutional networks within a Laplacian pyramid framework [8]. This uses multiple Generator-Discriminator networks at various levels of a Laplacian Pyramid. The Laplacian pyramid is built from a Gaussian pyramid using upsampling u(.) and downsampling d(.) functions [9].

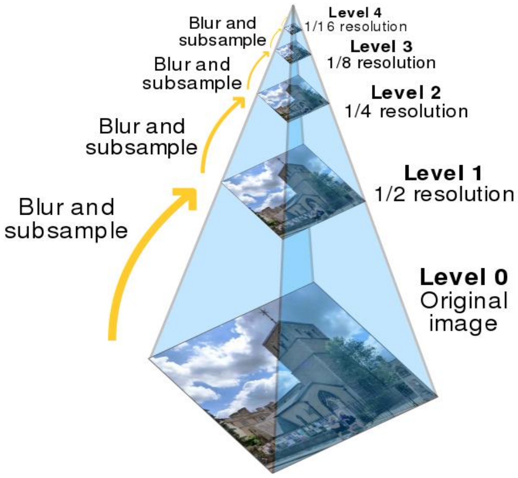

Fig 4. Laplacian Pyramid [7]

The basic working principle in LAPGAN is that the image is first downscaled at each layer of the pyramid until it reaches the last level. Then, in the backward pass through the pyramid, at each level the image assimilates a certain amount of noise produced by DCGAN. The image size is restored using upsampling. Moreover, this method generates image samples in a sequential process. The authors evaluated the performance of the LAPGAN model on three datasets: (i) CIFAR10 (ii) STL10 and (iii) LSUN datasets. This evaluation was done by comparing the log-likelihood, quality of image samples generated and a human evaluation of the samples.

D. Generative Recurrent Adversarial Networks (GRAN)

Im et al. (2016) proposed recurrent generative model showing that unrolling the gradient based optimization yields a recurrent computation that creates images by incrementally adding to a visual "canvas" [10]. The generative recurrent adversarial networks consist of two main entities: an encoder (convolutional neural network) and a decoder. The encoder's job is to extract images of the current canvas which along with the code of reference image is fed to the decoder. The decoder then decides whether to update the canvas or not.

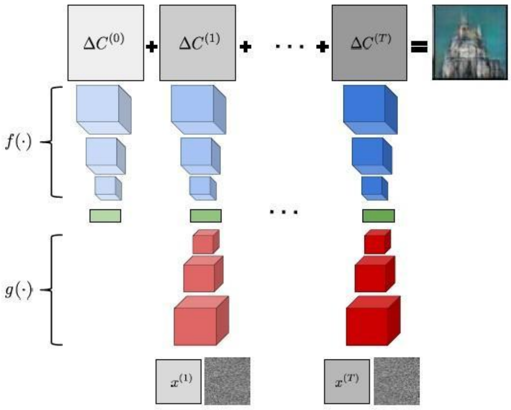

Fig 5. Architecture of GRA. A function f serves as decoder and g serves as encoder. [10]

Some other types of GANs are:

- Photo-Realistic Single Image Super-Resolution GAN (SRGAN)
- Information maximizing GAN (InfoGAN)
- Bidirectional Generative Adversarial Networks (BiGAN)
- Wasserstein GAN (WGAN)
- Cross-domain GAN (DiscoGAN)
- Generative Adversarial Autoencoders Networks (GAAN)
- Self-Attention Generative Adversarial Networks (SAGAN)

TABLE I

COMPARISONS OF SOME TYPES OF GANS [9]

VANILLA GAN	CGAN	DCGAN	GRAN	LAPGAN
Supervised Learning	Supervised Learning	Unsupervised Learning	Supervised Learning	Unsupervised Learning
Follows Multilayer perceptron Structure	Follows Multilayer Perceptron Structure	Uses Convolutional networks with constraints	Uses Recurrent convolutional networks with constraints	Uses Laplacian pyramid with sequential convolutional networks
Stochastic Gradient Descent with *k* steps for D and 1 step for G is used for optimization	Stochastic Gradient Descent with *k* steps for D and 1 step for G is used for optimization	Stochastic Gradient Descent with Adam optimizer for both G and D is used for optimization	Stochastic Gradient Descent updates to both G and D is used for optimization	No updates
G wants high error rate of D, and D wants low error rate	G wants high error rate of D, and D wants low error rate conditioned on extra information	Learn hierarchy of representations from object parts to scenes in both G and D	Generation of images by incremental updates to a “canvas”	Generation of images in grainy-to-fine fashion with sequential CNN at each level of pyramid.

III. Applications of GANs

The most widely used applications of GANs arise from improving the efficiency of the machine by feeding it with more virtually feeding machine more amount of data by adding noise to the existing dataset which aid to deceive a machine into thinking the data is unfamiliar and treats it as distinguished data, thereby increasing the size of the training data.

As the knowledge of GANs has spread across researchers and students, many of them have proposed their own version of GANs which include: adversarial autoencoder[11] which can be used in applications such as semi-supervised classification, disentangling style and content of images, unsupervised clustering, dimensionality reduction and data visualization; image generation and sketch retrieval which can be used to improve translation, rotation and scale[12] Adversarial Feature Learning which help projecting data back into the latent space[13].

Also using GANs is getting popular for image blending, image inpainting, image translation, semantic segmentation, and video prediction and generation. As GANs will get more researched various new applications shall spring out which can extend the potential of the neural network.

IV. Future Work and Research Frontiers

GANs were first proposed only recently in 2014 which indicates that there are still a lot of areas which are unexplored where GANs can contribute. Though it has significantly contributed to the development of generative models, there are a lot of areas where GANs fails. In a Reddit thread, Ian Goodfellow [14] addressed to why GANs are still not applied in Natural Language Processing (NLP). GANs only work where the computed gradients have small yet continuous value. They fail to produce relevant results where the applications are based on discrete values such as NLP. A recent introduction to other types of GANs such as Wasserstein GANs has made progress on text-related applications which still requires a lot of research.

Another important problem in computer vision is image synthesis. The CGAN have been successful in this field. But it struggles to capture geometric or structural patterns that occur consistently in images (for example, dogs are often drawn with realistic fur texture but without clearly defined separate feet) [15]. Hence a new type of GAN known as Self-Attention Generative Network (SAGAN) was introduced in May 2018, which allows attention-driven, long-range dependency modeling for image generation tasks. The paper [15] also mentions that, with the help of SAGAN s, an increase in ~43% Inception Scores and reduction in ~32% Fréchet Inception distance, from the previous best-published scores, was observed on ImageNet dataset.

The combinations of GANs with different architectural models can also be used for text-to-image synthesis. Moreover, GANs are still mostly used only in computer vision related problems but they can also be extended to the field of audio and video domains. GANs have also started to show its presence in drug discovery for healthcare using ChemGAN [16]. The increasing domains of applications leads to challenging problems which are quite complex, thus demanding active research work and developments in generative models.

V. Conclusions

In this paper, we study and review about a developing and popular network, Generative Adversarial Networks. We also investigate about the various types of GANs, their applications, and current future developments in the field.

In our opinion, GANs could be one of the highly potential field in the niche area of neural networks, which is still left unexplored and has not reached to full potential in research development. It could emerge out to be one of the most powerful generative models available.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Advances in Neural Information Processing Systems 27, Montreal, Quebec, Canada, 2014, pp.2672−2680.

[2] Kunfeng Wang, Chao Gou, Yanjie Duan, Yilun Lin, Xinhu Zheng, Fei-Yue Wang “Generative Adversarial Networks: Introduction and Outlook”, Ieee/Caa Journal Of Automatica Sinica, Vol. 4, No. 4, October 2017, pg. 588-598.

[3] “GAN: A Beginner’s Guide to Generative Adversarial Networks”, https://deeplearning4j.org/generative-adversarial-network.

[4] L. J. Ratliff, S. A. Burden, and S. S. Sastry, “Characterization and computation of local Nash equilibria in continuous games,” in Proc. 51st Annu. Allerton Conf. Communication, Control, and Computing (Allerton), Monticello, IL, USA, 2013, pp.917−924.

[5] Mehdi Mirza, Simon Osindero, “Conditional Generative Adversarial Nets”, arXiv:1411.1784v1 [cs.LG] 6 Nov 2014.

[6] Alec Radford, Luke Metz and Soumith Chintala, “Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks”, arXiv:1511.06434v2 [cs.LG] 7 Jan 2016.

[7] Shravan Murali, ”GANs, a modern perspective”, https://medium.com/deep-dimension/gans-a-modern-perspective-83ed64b42f5c.

[8] Denton, Emily L, Chintala, Soumith, Fergus, Rob, et al. Deep generative image models using a Laplacian pyramid of adversarial networks. In Advances in neural information processing systems, pp. 1486–1494, 2015.

[9] Saifuddin Hitawala, "Comparative Study on Generative Adversarial Networks", arXiv:1801.04271v1 [cs.LG] 12 Jan 2018.

[10] Im, Daniel Jiwoong, Kim, Chris Dongjoo, Jiang, Hui, and Memisevic, Roland. Generating images with recurrent adversarial networks. arXiv preprint arXiv:1602.05110, 2016.

[11] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, Brendan Frey, "Adversarial Autoencoders" arXiv:1511.05644 [cs.LG] Nov 2015.

[12] Creswell A., Bharath A.A. (2016) Adversarial Training for Sketch Retrieval. In: Hua G., Jégou H. (eds) Computer Vision – ECCV 2016 Workshops. ECCV 2016. Lecture Notes in Computer Science, vol 9913. Springer, Cham.

[13] Jeff Donahue, Philipp Krähenbühl, Trevor Darrell," Adversarial Feature Learning", arXiv:1605.09782v7 [cs.LG] May 2016

[14] https://www.reddit.com/r/MachineLearning/comments/40ldq6/generative_adversarial_networks_for_text/cyyp0nl/?st=jipxa0hb&sh=84d2d5af

[15] Han Zhang , Ian Goodfellow, Dimitris Metaxas, Augustus Odena, "Self-Attention Generative Adversarial Networks", arXiv:1805.08318v1 [stat.ML] 21 May 2018

[16] Mostapha Benhenda ," ChemGAN challenge for drug discovery: can AI reproduce natural chemical diversity?", arXiv:1708.08227v3 [stat.ML] , Aug 2017