Image Generation Using Different Models Of Generative Adversarial Network

Ahmad Al-qerem
Computer Information Systems Dept.
Zarqa University
Jordan
ahmad_qerm@zu.edu.jo

Yasmeen Shaher Alsalman
Computer Science Dept.
Princess Sumaya University for Technology
Jordan
yas20188012@std.psut.edu.jo

Khalid Mansour
Computer Science Dept.
Zarqa University
Jordan
kmansour@zu.edu.jo

***Abstract*—Generative adversarial networks (GANs) can be used in modeling highly complex distributions for real world data, especially images. This paper compares between two different models of the Generative Adversarial Networks: the Multi-Agent Diverse Generative Adversarial Networks (MAD-GAN) which consists of multi-generator and one discriminator and the Generative Multi-Adversarial Networks (GMAN) that has multiple discriminators and one generator. The results show that both MAD-GAN and GMAN outperformed the DCGAN. In addition, MAD-GAN performs better than GMAN when avoiding mode collapse or when the dataset contains many different modes.**

***Index Terms*— Deep Learning, Generative Adversarial Networks (GAN), image completion**

I. INTRODUCTION

Since its introduction, Generative Adversarial Networks (GANs) received the attention of the artificial intelligence research community due its important applications especially in the field of image generation [11]. The main idea behind GANs models is to mimic any distribution with high dimensional data and generate a similar one including images, music, speech, etc.

Generative adversarial networks (GANs) are considered to be an example of generative models. Any model that uses a training set which its sample represents a particular distribution p_{data}, and learns to mimic that distribution and also has p_{model} that represents the resulted probability for that distribution, is also called a generative model.

GANs uses the deep learning algorithm to generate a high quality fake images. Constructing a generator along with a discriminator algorithm, where the discriminator uses the formulation p(y|x) where y is the correct label for the feature x. On the other hand, the generator cares about the probability of having feature x given label y p(x|y).

GAN can be used in many different applications such as generate new human poses, generate examples for image datasets, text-to-image translation and others. GANs use deep neural net architectures that can be used in training and sampling a high dimensional probability distributions, and provide predictions on inputs with missing data.

In this paper, two different models of generative adversarial networks are compared: the Generative Multi Adversarial Networks (GMAN) and the Multi-Agent Diverse Generative Adversarial Networks (MAD-GAN). Both models are tested using the MNIST dataset. The experimental results show that MAD-GAN performs better than GMAN when avoiding mode collapse or when the dataset contains many different modes. Moreover, GMAN performed better than GAN on the MNIST dataset.

The rest of the paper is organized as follows: Section 2 reviews the related work. Section 3 introduces the generative adversarial networks approach. Sections 4 and 5 present GMAN and MAD-GAN respectively. Section 6 shows the experimental results while section 7 concludes the paper.

II. RELATED WORK

A. Generative Adversarial Network

Many researchers proposed new models to enhance the outcome of GANs [2], others used GANs in a creative way [5]. The work in [4] expanded GANS to generate image in a specific location. In [2], authors developed a simple and effective GAN architecture that translates human-written descriptions to bird and flower images. In order to generate images from the text the problem have been divided into two sub-problems: first, learn a text feature representation and second, use these features to synthesize a compelling image. A deep convolutional generative adversarial network (DC-GAN) has been trained on text features encoded by a hybrid character-level convolutional- recurrent neural network. Both the generator and discriminator perform feedforward inference conditioned on the text feature.

A new model for GAN called, the Generative Adversarial What-Where Network (GAWWN) is proposed in [4]. It generates images given instructions of what to draw in which

978-1-7281-3010-1/19/$31.00 ©2019 IEEE

location, the image generation are conditioned on both text descriptions and object locations. Locations can be accurately controlled by either a boundary box or a set of part key points, the model learned the content and location key points of the generated image from the Caltech-USCD birds (CUB) and MPII Human Pose dataset (MHP) datasets

On the other hand, a new learning model called the auto-painter model which can generate automatically compatible sketch colors based on conditional Generative Adversarial Networks (cGANs) is proposed in [5]. Not only the new model is able to paint hand drawing sketches with proper colors, but also it allows users to choose their preferred colors so that painted sketches can be automatically generated.

B. Image completion

The work in [6] compares between two algorithms used in image Inpainting, the Exemplar-Based Inpainting (EBI) and Deep Convolutional Generative Adversarial Nets (DCGAN), EBI run faster and it can be applied to larger types of images but it h been noticed that it is heavily dependent on the surrounding pixel information, thus it has a low quality output image if the masked area is large ,while DCGAN produced a high quality output image but it take longer the training phase.

The work in [3] introduced a novel statistics of similar patches offset approach for image completion and when matching these similar patches in the image, a completed image is obtained.

III. THE GAN APPROACH

Generative Adversarial Networks (GANs) were introduced by lan Goodfellow[1] and other researchers in 2014. GAN can learn to mimic any distribution of data and generate a similar one including images, music, speech, prose, etc.

GAN is a minimax game between two parties: a generator and a discriminator, the generator (G) learns to create a new sample similar to the real data and the discriminator (D) learns to determine whether the data is real or it has been generated by the generator. They contest against each other during training phase until the generator can produce high quality fake data that the discriminator can't tell if it is real or fake data, figure 1 shows the architecture of GANs.

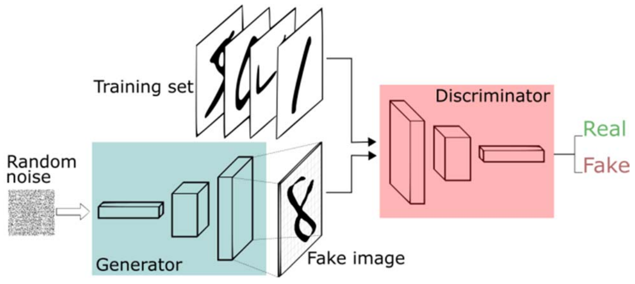

Figure 1: (GAN) architecture shows that the Generator uses a prior noise to generate images and the Discriminator tries to figure out if the generated image is real or fake .

Generally GANs are a structured probabilistic model using the variable z *to* obtain variable x. The generator function takes z as input, where P(z) is the distribution of the prior noise z, and generates x. The discriminator function takes x as input and outputs the probability that x comes from the distribution of the data.

We train D to maximize the likelihood of getting the correct label to both G training examples and samples from data. On the other hand G will be trained to minimize that probability, log(1 − D(G(z))).
It can be written as the following formula (1).

$$min_G max_D V(D,G) = E_{x\sim pdata(x)}[log\ D(x)] + E_{z\sim pz(z)}\ [log(1 - D(G(z)))] \qquad (1)$$

Where x is the real data, z is a prior noise, and we need to obtain the probability of having the sample in the data (px(x)) and not having a prior noise (1-pz(z)) .

Equation 1 works fine theoretically but it doesn't perform well in practice, since the discriminator minimizes the cross-entropy while the generator maximizes the same cross-entropy, because the discriminator will be able to detect the generated images with high confidence, thus the generator gradient will vanish.

For solving this problem Ian Goodfellow in [7] suggests to use the maximum likelihood game, which is the same of minimizing the KL divergence between the data and the model.

$$J(G) = -½\ E_z \exp(σ\text{-}1\ (D(G(z))) \qquad (2)$$

The cost function for the generator is shown in equation 2 after it has been updated, where σ is the logistic sigmoid function.

IV. Generative Multi-Adversarial Networks.

It has been noticed that GAN is difficult to train thus many researchers improved the training techniques and its heuristics. A new framework which has multiple discriminator for improving the training and generating process is proposed in [8].

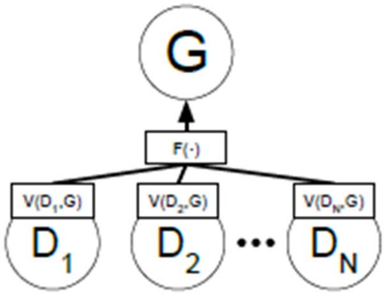

Figure 2: (GMAN) architecture shows that the G uses an aggregating feedback from set of discrimination.

Generative multi adversarial networks (GMAN) were introduced by Ishan Durugkar and others have extended the GANs to multiple discriminators. Two approaches have been designed for this framework: the formidable design and the forgiving teacher.

Generally, the objective of G has been reformulated in figure 1 as min_Gmax F(V(D_1,G), . . . , V(D_N,G)) for various choices of F, and each D tries to maximize its own V(D_i,G).

A. *Formidable Adversary.*

In the formidable design, multi-discriminator variants try to perform a better approximation for max_DV(D,G), providing the generator with a harsh criticism.

The formidable adversary maximize $F_g(v_i)$ with F equal to max, from N identical discriminator, which is in other word optimizing V using stochastic gradient ascent and then presenting $max_{i\in\{1,\dots,N\}}$V (D_i,G) as the loss to the generator, thus the generator is required to minimize the max, and that forces G to produce high quality samples which have to fool all N discriminators where each represents a different max.

B. *Forgiving teacher*

The formidable design focus on enhancing the discriminator with the objective of presenting the generator with a better approximation of max_{Di}V(D,G). on the other hand the forgiving teacher asks, "Ismax_{Di}V(D,G) too harsh?".

The information in the gradient derived from negative feedback, which is from formidable adversary, dictates only where pG(x) is to be driven, not specifically where pG(x) is to be increased, thus the generator needs a permissive discriminator for getting positive feedback. The classical pythagorean means parameterized by λ and the original Minimax objective can be used for softening the max operator. There are three pythagorean means: the arithmetic mean (AM), geometric mean (GM) and harmonic mean (HM) where (Min< HM < GM < AM < Max).

$$\text{AMsoft}(V,\lambda)=\sum_i^N \quad w_i\, V_i \tag{3}$$

$$\text{GMsoft}(V,\lambda)=-exp\ (\sum_i^N \quad w_i\ log\ (-V_i)) \tag{4}$$

$$\text{Msoft}(V,\lambda)=(\sum_i^N \quad w_i\, V_i^{\ -1})^{-1} \tag{5}$$

where $w_i = e^{v_i\lambda}/\sum_j e^{v_i\lambda}$ with $\lambda \geq 0, V_i < 0$.

It can be noticed that the AM(V,0) is equivalent to training the generator using the following formula (6)

$$1/N\ \sum_i^N \quad E_{\times\sim pg(z)}[log(z)] \tag{6}$$

where z = $\prod_i^N \quad (1 - D_i(x))$, the generator gradient (derivative =0) minimize when z=1 according to equation (6), and z=1 means that $D_i = 0\ \forall i$, if so then the gradient of G will vanish because all the discriminators agree that the generated image is fake but this is unlikely to happen if N is large, thus the generator needs only to fool at least one discriminator to get a useful feedback.

At the beginning of the training λ , is set to zero or closer to zero for using the mean and because of hardness of finding $max_{Di}V(D_i, G)$, after that we can increase λ for pushing the generator to generate a high quality images [8].

V. Multi-Agent Diverse Generative Adversarial Networks.

MAD-GAN were introduced by Arnab Ghosh and other authors [9] to solve the mode collapse problem ,which is one of the GANs problem that happens when only a few modes of data are generated, by enforcing the generator to capture different modes.

Generally MAD-GAN consists of multiple generators and one discriminator, two different objectives have been proposed for MAD-GAN; the similarity based competing objective, where the generator objective is not only to produce real looking images but each generator should generate images that are different from other generators, the second objective is the generator identification based objective, where the generator objective has not been changed but the discriminator objective should also detect the generator that generate the specific image.

The architecture depends on the objective that MAD-GAN uses; if similarity based competing objective is used then, for the generator to generate distinct images, the weights of each generator is shared in the initial layers with all other generators and only the final layer have different weights. On the other hand when using the generator identification based objective, the change in the discriminator's output is K+1 classes where K is the number of generators and K+1 determines if it is real or fake image.

In order to enforce diversity of modes in similarity based competing objective, suppose that there is two different modes in the training set and there is two generators, one generator trains from one mode, and the second generator generates images from the other mode, to do so the following formula have been used (7)

$$D(G_i(z)) \geq D(G_j(z)) + \Delta\,(\phi G_i(z), \phi(G_j(z))), \forall j \in K \setminus i \quad (7)$$

Where $\Delta\,(..,..)$ takes specific similarity function and $\phi(G_i(z))$ mapes the generated images to a feature space.

On the other hand the way for enforcing the generative identification objective, the discriminator maximizes the negative cross entropy of the dirac delta function for the discriminator.

$$Max \quad {}_{\theta d}E_{x\sim p}H(\delta.D(x)) \quad (8)$$

where H is the negative cross entropy and $\delta \in \{0,1\}^{\ K+1}$ is the dirac delta function, also $j \in \{1,.....,K\}$, if the $\delta(j) = 1$ then the sample belongs to the jth generator, otherwise $\delta(K+1) = 1$t then the discriminator considers it as a real looking image . The main idea is that the discriminator learns to push different generators to different modes [8].

VI. Experimental Results

In this Section an experiment have been conducted on MNIST dataset using deep convolutional GAN (DCGAN) were the number of iterations was 25000, and each sample contain 500 observations, the Generator latent variable z uses the uniform distribution$Z \sim U(-1,1)^{\ 100}$, the convolutional neural network has been used for both generator and discriminator, the relu function is also used. Figure 3 showed the output of the first iteration.

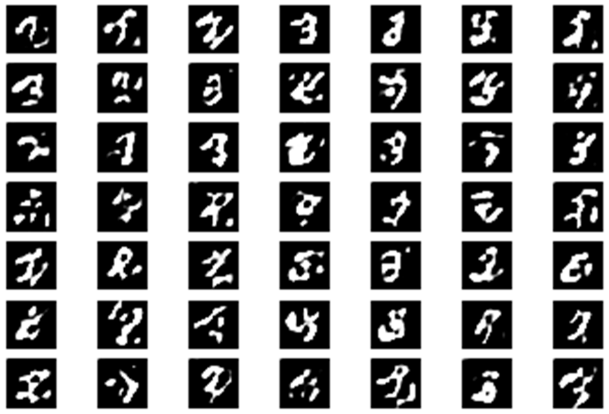

Fig 3: DCGAN

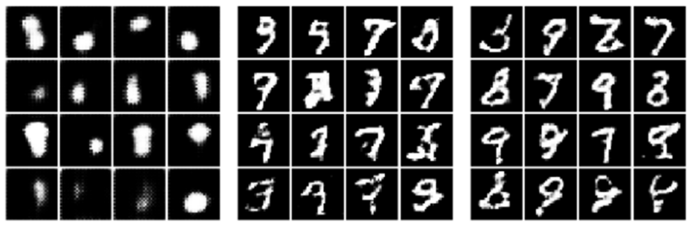

Fig 4: MAD-GAN

While figure 4 shows the results after running the code provided by the author of [9] on the same MNIST dataset, it can be noticed that the output of the first few iteration was improved significantly, when using more than one generator.

On the other hand after running the code that has been provided by [8] on the same dataset, the results shown in figure 5 was two time faster when using more than one discriminator in terms of outputting a readable digit than the original GAN

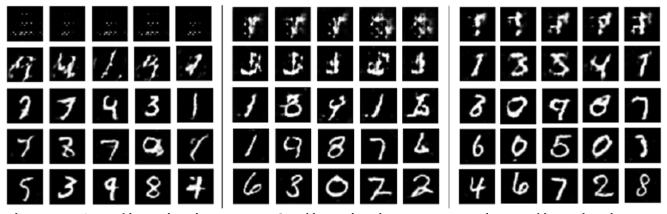

Fig 5: 1- discriminator, 2-discriminator and 5 discriminator GMAN

It can be noticed that both GMAN and MAD-GAN performs better than DCGAN. In general, multiple generator models like the Mixture Generative Adversarial Nets (MGAN) [10] perform better than GMAN in terms of covering different modes. They also have higher inception score when using the CIFAR-10 dataset as shown in Table I.

TABLE I: Inception score for MGAN and GMAN.

	MGAN	GMAN
Score	8.33±0.10	6.001 ± 0.194

VII. Conclusion

This paper compares between different models of Generative Adversarial network: MAD-GAN and GMAN. It can be noticed that MAD-GAN performs better than GMAN when avoiding mode collapse, or if the dataset contains many different modes, especially when using the similarity based objective. On the other hand GMAN, proved that it performs better that the original GAN on MNIST dataset. In addition, multiple generator models like the Mixture Generative Adversarial Nets (MGAN) perform better than GMAN in terms of covering different modes. They also have higher inception score when using the CIFAR-10 dataset.

References

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

[2]S. Reed, Z. Akata, X.Yan, L.Logeswaran R. Schiele, H.Lee" Generative Adversarial Text to Image Synthesis", 2016

[3] He, Kaiming, and Jian Sun. "Image completion approaches using the statistics of similar patches." IEEE transactions on pattern analysis and machine intelligence 36.12 (2014): 2423-2435.
////////
[4] Reed, Scott E., et al. "Learning what and where to draw." Advances in Neural Information Processing Systems. 2016.

[5]Liu, Yifan, et al. "Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks." arXiv preprint arXiv:1705.01908 (2017).

[6] Huang, Yoshida " Evaluations of Image Completion Algorithms: Exemplar-Based Inpainting vs. Deep Convolutional GAN "

[7] Goodfellow, Ian. "NIPS 2016 tutorial: Generative adversarial networks." arXiv preprint arXiv:1701.00160 (2016).

[8] Durugkar, Ishan, Ian Gemp, and Sridhar Mahadevan. "Generative multi-adversarial networks." arXiv preprint arXiv:1611.01673 (2016).

[9] Ghosh, Arnab, et al. "Multi-agent diverse generative adversarial networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[10]Hoang, Quan, et al. "MGAN: Training generative adversarial nets with multiple generators." (2018).

[11] Delbrouck J., Vanderplaetse B., Dupont S., Can adversarial training learn image captioning?, 33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, Canada. 2019.