A Critical Review of Generative Adversarial Networks based on Stability Criteria

Vishal M. Chudasama* and Kishor P. Upla**
*-**Electronics Engineering Department, Sardar Vallabhbhai National Institute of Technology, Surat, India
{vishalchudasama2188, kishorupla}@gmail.com

Abstract: In the machine learning field, the traditional deep learning models are mostly of discriminative type in which their goal is to discover a map from input layers to output layers. Also, these models require large amount of annotated data for training. On the other hand, deep generative models (DGMs) provide a new way to learn features effectively from the sample data which do not require the labeled data. Among the many DGMs, generative adversarial networks (GANs) are the emerging models for both semi-supervised and unsupervised learning. GANs use a pair of discriminator and generator networks which are used in competitive process to learn the effective features. However, the implementation of GANs suffers against the challenging problem of stability of training. This paper discusses the review and challenges of the implementation of GANs. We review different GAN models such as deep convolutional GAN (DCGAN), Wasserstein GAN (WGAN), WGAN with gradient penalty (WGAN-GP) and boundary equilibrium GAN (BEGAN) which improve the stability of the its training. The improvement in terms of stability of these GANs is evaluated by conducting the different experiments on the common database of Fashion-MNIST. Additionally, the mode collapse problem of GAN is tackled using unrolled GAN which is also reviewed and discussed.

Keywords: GAN, DCGAN, WGAN, WGAN-GP, BEGAN, Unrolled GAN, Fashion-MNIST.

Introduction

In the machine learning, deep learning models extract the features from the data samples effectively by using a deep architecture which is designed based on non-linear transformations. These deep learning models are mostly discriminative in nature where the main goal is to discover a map from inputs to outputs. However, these models have several limitations: i) they require vast amounts of annotated data; ii) they fail drastically when given inputs are not similar to the inputs of training set. Among many types of deep models, deep generative models (DGMs) are powerful for unsupervised and semi-supervised learning in which instead of discriminating the inputs, they try to replicate the (hidden) statistical process within the data without relying on external labels. They start by generating "hallucinations" that become more realistic and plausible as the learning process evolves. DGMs learn the abstract representations from unlabeled data and perform a wide range of tasks, including density estimation, data generation and missing value estimation. The most common deep neural network based generative models are variational auto-encoders (VAEs) [1] and generative adversarial networks (GANs) [2], [3]. Training of these models are based on Bayesian deep learning [4], variational approximations [5], Monte Carlo Markov chain (MCMC) estimation [6], or the old faithful stochastic gradient estimation (SGD) [7].

Literature shows that the GANs are one of the most promising areas of research in unsupervised learning. Many models based on GAN have been proposed by the researcher in order to improve its performance [8]–[15]. At the same time, as deep learning became popular, the need for huge amounts of data has risen. GANs tend to fill these gaps as they propose a generative model for natural images that evolves to generate more and more realistic looking data, because of the coupling with an adversarial network.

In practice, training in GAN is very challenging task. The relative model capacities of the generator and discriminator must be carefully balanced in order for the generator to effectively learn. In this paper, we discusses the review and challenges of the implementation of GANs. We review the different GAN models such as deep convolutional GAN (DCGAN) [8], Wasserstein GAN (WGAN) [12], WGAN with gradient penalty (WGAN-GP) [13] and boundary equilibrium GAN (BEGAN) [15] which improve the stability of the GANs training. These models are implemented on the common database of Fashion-MNIST and evaluated based on stability criteria. In addition to that, the mode collapse problem of GAN is reduced by using unrolled GAN [10] which is also reviewed and discussed.

Background of GAN

GAN was proposed by Ian Goodfellow at the university of Montreal in the year of 2014. The main idea behind the GANs is to have two competing neural network models. Goodfellow et al. [2] use simple multi-layer perceptrons (MLP) for both the networks. In these two networks, one takes noise as input and generates the data samples (called the generator (G) network)

and the other network (called the discriminator (D)) receives samples from both the generator and the training data and its task is to distinguish between the two data coming from two sources.

Let, p_{data} is a target distribution which is required by generator model G to learn by approximating it with the model distribution, p_{model}. G is associated with a noise prior p_z, from which G draws samples z, and create fake sample $G(z;\ \theta_G)$. Here, θ_G are the model parameters. The discriminator $D(z;\ \theta_D)$ takes x and $G(z)$ as input, and returns a binary judgment as to whether the given input is from p_{data} or p_{model}. θ_D are the model parameters of discriminator network D. The goal of GANs is to train the generator network $G(z;\ \theta_G)$ which produces samples from the model distribution, $p_{model}(x)$, by transforming vectors of noise z as $x = G(z;\ \theta_G)$. The training signal for G is provided by a discriminator network $D(z;\ \theta_D)$ which is trained to distinguish samples from the generator distribution $p_{model}(x)$ from real data. In other words, generator network $G(z;\ \theta_G)$ is trained to fool the discriminator to accept its outputs as being real. These two networks play a continuous game, where the generator learns to produce more and more realistic samples, and the discriminator is learning to get better and better at distinguishing generated data from real data. These two networks are trained simultaneously, and the hope is that the competition will drive the generated samples to be indistinguishable from real data.

The GAN learning problem is to find the optimal parameters θ_G^* for a generator function $G(z;\ \theta_G)$ in a minimax objective as [2], [3],

$$\begin{aligned}\theta_G^* &= \arg min_{\theta_G} max_{\theta_D} f(\theta_G, \theta_D) \\ &= \arg min_{\theta_G} f(\theta_G, \theta_D^*(\theta_G)) \qquad (1)\\ \theta_D^*(\theta_G) &= \arg\ max_{\theta_D} f(\theta_G, \theta_D) \qquad (2)\end{aligned}$$

where minimax loss function f commonly chosen as,

$$f(\theta_G, \theta_D) = E_{x\sim p_{data}}[\log(D(x;\theta_D))] + E_{z\sim p_{model}}[\log(1-(D(G(z;\theta_G);\theta_D)]. \qquad (3)$$

Here, $x \in X$ is the data variable, $z \in Z$ is the latent variable, the discriminator $D(\cdot;\theta_D): X \to [0,1]$ outputs the estimated probability that a sample x comes from the data distribution, $G(\cdot;\theta_G): Z \to X$ transforms a sample in the latent space into a sample in the data space. Here, function of D is to minimize the value of $D(G(z))$ as it is generated sample and at the same time it also functions to increase the value of $D(x)$ to indicate it as the real data. For optimization of equation (1), Goodfellow et al. [2] propose an algorithm in which gradient descent on θ_G and ascent on θ_D are used in alternating way. This process continued until p_{model} draws closer to p_{data} and eventually reaches an equilibrium point where D can no longer classify samples as real or generated samples (i.e., $D(x) = D(G(z)) = 1/2$). Finally, the optimal solution $\theta^* = (\theta_G^*,\ \theta_D^*)$ is a fixed point of these iterative learning dynamics.

Challenges of GANs

One of the challenges in standard GAN [2] is the nonconvergence between two networks. This is because of the term $\log(1 - D(G(z)))$ in in equation (3) which rapidly saturates in the early stage of training, where D easily rejects $G(z)$ because G generates fake data of poor quality and they continuously differ from the real data. Therefore, rather than evaluating how bad fake data, evaluate how good they are by setting G's goal to maximizing $\log(D(G(z)))$.

Another most important challenge of standard GAN [2] is its stability in training. Despite the theoretical existence of unique solutions, GANs training is challenging and often unstable due to many reasons [8], [9]. One way to improve the stability of training is to assess the empirical "symptoms" which are observed during the training of GANs. These symptoms include:

- Difficulties in the simultaneous convergence of both models [8];
- The generative model may create "mode collapsing" which means to generate very similar samples for different inputs [9];
- The discriminator loss converging quickly to zero [16], providing no reliable path for gradient updates to the generator;
- More powerful discriminator would fail the generator to train effectively [9], [15];
- Difficulties in controlling the diversity of the generated samples [15];
- Difficulties in balancing convergence between discriminator and generator as the discriminator wins easily at the beginning of the training;
- Less impressive results of generating samples for wider variety of visual worlds [11]

Discussion on Stability Improvement of GANS

There are various methods proposed in the literature to overcome the stability issues of GANs. In this section, we discuss few important methods which can improves the stability of GANs and also solve the difficulties of GAN's training.

Deep convolutional GAN (DCGAN) [8]

Radford et al. [8] proposed a DCGAN model which was the first major improvements in the training of GANs for image generation. DCGAN uses a standard convolutional neural network (CNN) components, such as deconvolutional layers, fully

connected layers, etc. Authors propose a set of guidelines which are used in the construction and training of model. Following are the guidelines to stable the training of DCGAN [8]:

- Replace any pooling layers with strided convolution in discriminator and fractional-strided convolution in generator network.
- Use batchnorm in both generator as well as discriminator networks.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation function in generator for all layers except for the output, which uses tanh.
- Use LeakyReLU activation in the discriminator for all layers.

By using above guidelines, Redford et al. [8] obtain accuracy upto 82.8% with error rate of 22.48% on CIFAR10 dataset in image classification. The performance of DCGAN is still poor when compared to CNN based model [17] and also authors found that the generator collapse sometimes due to its large learning rate. In order to improve stability, lower learning rate is required.

Improved techniques for training of GANs [9]

In order to improve the stability of GAN's training, Salimans et al. [9] propose heuristic approaches. They suggest some small changes to the GANs training scheme that lead to visually improved results. The first technique is about feature matching which changes the generator objective slightly in order to increase the amount of information available. The new objective is written as,

$$\| E_{x\sim p_{data}} f(x) - E_{z\sim p_{model}} f\big(G(z)\big) \|_2^2, \tag{4}$$

where f is some intermediate layer in D. The discriminator is still trained to distinguish between real and fake samples, but now the generator is trained to match the discriminator's fake sample expected intermediate activations (features) with the real samples expected intermediate activations.

The second trick is for preventing mode collapse problem in GANs which produces the same samples for different inputs. Authors use mini-batch discrimination in order to overcome mode collapse problem in which an extra input is added to the discriminator. This extra input behaves as a feature that encodes the distance between a given sample in a mini-batch and the other samples. By using this concept, the discriminator can easily tell if the generator is producing the same outputs.

A third approach is heuristic averaging which adds penalty term to the network parameters if they deviate from historical average values. This can help to converge to an equilibrium condition which may not be possible with normal gradient descent. The fourth technique is virtual batch normalization. Here, authors normalize each example with respect to the examples in a reference batch, which was picked once at the start of the training. This reduces the dependency of one sample on the other samples in the mini-batch and improves optimization of neural network.

Finally, one-sided label smoothing technique makes the target for the discriminator 0.9 instead of 1 and smoothing the discriminators classification boundary. This technique prevent discriminator for being more powerful than generator so that generator would train more effectively.

All above techniques are incorporated by Salimans et al. [9] in their work and perform experiments using semi-supervised training. They generated the samples from different databases such as MNIST, CIFAR-10 and ImageNet and obtain the error rate of 14.87% on CIFAR10 database for a given number of labeled samples. This error rate value is lower than that of using standard GANs. Additionally, authors also compare the samples generated by the generator during semisupervised training obtained using feature matching and minibatch discrimination techniques. Authors achieve improvement in the visual quality of the generated samples using mini-batch discrimination. Furthermore, authors also claim that by using their heuristic approaches, the stability as well as image quality are improved over the ordinary GAN.

Unrolled GANs [10]

In [10], Metz et al. demonstrate the issue related to mode collapse and also discuss how to improve the stability of GANs. Authors propose a method which unroll the discriminator for several steps, i.e., discriminator updates on the current generator for several steps, and then using the "unrolled" discriminators to update the generator using the normal minimax objective. As compare to GAN, in the loss function (equation (3)), Metz et al. [10] introduce a surrogate objective function $f_K(\theta_G;\theta_D)$ for training the generator which more closely resembles the true generator objective $f(\theta_G;\ \theta_D^*(\theta_G)$. In unrolled GAN, a local optimum of the discriminator parameters θ_D^* can be expressed as the fixed point of an iterative optimization procedure,

$$\theta_D^{k+1} = \theta_D^k + \eta^k \frac{df(\theta_G,\theta_D^k)}{d\theta_D^k} \tag{5}$$

where, η^k is the learning rate schedule. By unrolling for K steps, authors create a surrogate objective for the update of the generator as,

$$f_K(\theta_G;\theta_D) = f(\theta_G;\ \theta_D^K(\theta_G;\theta_D) \tag{6}$$

This surrogate loss term captures how the discriminator would react to a change in the generator. It reduces the tendency of the generator to engage in mode collapse. However, drawbacks of this approach are the increased training time which increases linearly with the number of unrolling steps and a more complicated gradient calculation.
In their experiments, Metz et al. use the convolutional neural network as discriminator and the recurrent neural network as generator. Due to this, the resulting model has more complex in power balance. Authors also observe that without unrolling, the model quickly collapses to a single mode and rotates around the data distribution. When running with unrolling steps the generator disperses and appears to cover the whole data distribution.

Wasserstein GAN (WGAN) [12]
The standard GAN [2] uses a Jensen-Shannon (JS) divergence to optimize the loss function. However, JS divergence does not provide enough information when the discrepancy is too large and also it is not differentiable to every point in space which makes the gradients in GANs vanish over most of the time. In order to solve above issues, Arjovsky et al. [12] suggest to use Wasserstein distance (also known as Earth mover distance) which is differentiable nearly everywhere in the space. They propose Wasserstein GAN (WGAN) which uses an alternative loss function which is derived from an approximation of Wasserstein distance. The intuition behind the Wasserstein distance is as the probability distributions are defined by how much mass they can put on each point. WGAN use this distance in the loss function. However, computing Wasserstein distance exactly is intractable. Hence, authors tried to compute it approximately by using Kantorovich-Rubinstein duality form. A result from Kantorovich-Rubinstein duality is equivalent to

$$W(p_{data}, p_{model}) = sup_{\|f\|_L<1}[E_{x\sim p_{data}} f(x) - E_{z\sim p_{model}} f(x)] \quad (7)$$

where the supremum is taken over all 1-Lipschitz functions.
Unlike the standard GAN's cost function [2], the WGAN is more likely to provide gradients that are useful for updating the generator. The cost function derived for the WGAN relies on the discriminator, which is referred as the "critic"; practically, this may be implemented by simply clipping the parameters of the discriminator.
In standard GAN [2], gradients vanish over most of the space while in WGAN, the weight clamping gives a reasonably nice gradient over everything. The WGAN samples are in more details and hence they do not mode collapse as much as standard GAN [2]. WGAN method [12] is experimented for image generation. The target distribution is learned for the LSUN-Bedrooms dataset [18] which is a collection of natural images of indoor bedrooms. Authors have demonstrated the performance of WGAN approach with different generator architectures and observe that Wasserstein estimate correlates well with the visual quality of the generated samples. Authors also reported that the training of WGAN becomes unstable at times when one uses a momentum based optimizer such as Adam optimizer on the critic.

WGAN with gradient penalty (WGAN-GP) [13]
Basically, WGAN designs to stabilize the training process of GANs. However, it generates the low-quality samples and also more often fails to converge due to the use of weight clipping which arises due to Lipschitz constraint in discriminator. This weight clipping adversely reduces the capacity of the discriminator model and forces it to learn simpler functions. Gulrajani et al. [13] propose an improved method for training the discriminator of WGAN by penalizing the norm of discriminator gradients with respect to data samples during training, rather than performing parameter clipping. To circumvent tractability issues, authors enforce a soft version of the constraint with a penalty on the gradient norm for random samples$\hat{x} \sim p_{\hat{x}}$. The new objective loss function is

$$L = E_{z\sim p_{model}}\big[D\big(G(z)\big)\big] - E_{x\sim p_{data}}[D(x)] + \lambda E_{\hat{x}\sim p_{\hat{x}}}[(\| \nabla_{\hat{x}} D(\hat{x}) \|_2 - 1)^2]. \quad (8)$$

One advantage over weight clipping is in terms of improvement achieved in the speed of training and quality of sample generated by the generator. By adapting penalty term to the standard GAN's objective function [2], it may stabilize the training and encourages the discriminator to learn smoother decision boundaries. It performs better than standard WGAN and enables stable training of a wide variety of GAN architectures with almost no hyperparameter tuning, such as 101-layer ResNets.
To demonstrate the stability of training process, Gulrajani et al. [13] trained the proposed GAN with the different architectures on LSUN bedrooms dataset. Based on the results authors conclude that the critics trained with weight clipping fail to capture higher moments of the data distribution while WGAN with gradient penalty can capture the same in effective way. Also, they found that their method improve the training speed and sample quality when compared to WGAN with weight clipping [12]. However, convergence rate is slow when compared to DCGAN, but performance is more stable at the point of convergence when compared to DCGAN.

Boundary equilibrium GAN (BEGAN) [15]
Many tricks have been applied in order to improve the training of GANs [8], [9]. However, there are still many difficulties which are unaddressed for the practical implementation of GANs. These include selection of correct hyper-parameters,

controlling the image diversity of the generated samples since discriminator wins too easily at the beginning of training [3], balancing the convergence of the discriminator and generator. To overcome these issues, David et al. [15] propose a new model based on equilibrium called boundary equilibrium GAN (BEGAN) which is paired with a loss derived from a Wasserstein distance for training the auto-encoder based GANs. This method balances the power of the discriminator against the generator during training.

In the BEGAN framework, the discriminator is an auto-encoder. The main idea behind the BEGAN is to match the distributions of the reconstruction losses of generated sample distribution with the data distributions. The real loss is then derived from a Wasserstein distance between the reconstruction losses of real and generated data. In training, *D* reconstructs real images more better and the weights of *D* are updated so that the reconstruction loss of real images is minimized. Additionally, *D* simultaneously increases the reconstruction loss of generated images and *G* works adversarially to that by minimizing the reconstruction loss of generated images. David *et al.* also introduce a convergence measure *M* which is an indicator to measure network's performance. It is also used to control learning rate. This measure can be used to determine when the network has reached its final state or if the model has collapsed.

BEGAN [15] is trained on CelebA dataset [19] using Adam optimizer with an initial learning rate of 0.0001 with decaying factor of 2. Authors compare the effect of varying parameter gamma γ on sample generation. Here, authors observe that for low value of gamma γ, the faces look overly uniform and variety increases with different values of γ. The convergence measure M_{global} of the BEGAN model which correlates well with image fidelity. Here, model converges quickly. To test the robustness of the equilibrium balancing technique, author perform an experiment advantaging the discriminator over the generator, and vice versa and they observe that by maintaining the equilibrium the model remained stable and converged to meaningful results.

Results and Discussions

In this section, we discuss the experimental results obtained using the implementation of different GANs. As discussed earlier, the stability of GANs is improved with different GAN models such as WGAN [12], WGAN-GP [13] and BEGAN [15] better than the GAN models proposed in DCGAN [8] and the method proposed in [9]. Due to this, here we have included the experimental results obtained using WGAN [12], WGANGP [13] and BEGAN [15] only. In addition to that, we also obtain the experimental results using unrolled GAN [10] which prevents the mode collapse problem effectively in standard GAN [2] and same are discussed. All these experiments were performed on a machine with Intel i7 6850k CPU, 64GB RAM and NVIDIA GeForce Titax X Pascal GPU. We trained all the architectures on Fashion-MNIST dataset [20] which is a new dataset comprising of 28×28 grayscale images of 70; 000 fashion products from 10 different categories and of 7; 000 images per category. In our experiments, we use Adam optimizer with learning rate of 0:0002 which is decaying by the factor of 2. We set the batch size of 64 and iterate upto 25 epochs where 1 epoch is equal to 1093 iterations.

Fig. 1 shows the image samples generated during the different training periods for WGAN [12], WGAN-GP [13] and BEGAN [15] methods. One can observe that the quality of images generated with standard GAN model is poor. This is mainly because of the poor convergence of discriminator and generator distribution losses. The samples generated by WGAN [12] model are also not better.

This is because of the use of Adam optimizer in critic as mentioned by the authors. It is interesting to observe that the images generated with WGAN-GP [13] and BEGAN [15] are better when compared to the same with GAN [2] and WGAN [12]. However, one can observe that convergence of BEGAN [15] is faster than that of the WGANGP [13]. Due to this, better image samples are generated with BEGAN [15] when compared to that with WGAN-GP [13] (see Fig. 1).

Furthermore, we have also trained unrolled GAN [10] on a 2D mixture of 8 Gaussians of 0.02 standard deviation and means are arranged equally spaced in a circle to understand the mode collapse problem. Both, discriminator as well as generator networks are optimized using Adam optimizer with learning rate of 1e-4 and 1e-3, respectively. The results are compared with standard GAN [2] (unrolling step = 0) which are displayed in the first row of Fig. 3. Here, a heatmap of the generator distributions after increasing number of training steps is showed. The green dots arranged around circle indicate target data distribution. By looking at top row results (i.e., standard GAN [2]), one can see that the generator never converges to a fix distribution and it assigns mass probability to a few fixed points only. However, in the case of unrolling step 5 (i.e., unrolloed GAN [10]), generator quickly spreads out and converges to the target distribution. This results in minimization of mode collapse problem.

Conclusion

GANs have made breakthroughs to unsupervised and semi-supervised learning in the area of deep learning. The practical implementation of GANs becomes very challenging mainly due to stability issue in training phase. We have discuss the standard GAN and its challenges. In order to solve those challenges, we have discuss the other well known GANs model which can mitigate stability problem to improve the training at some extent.

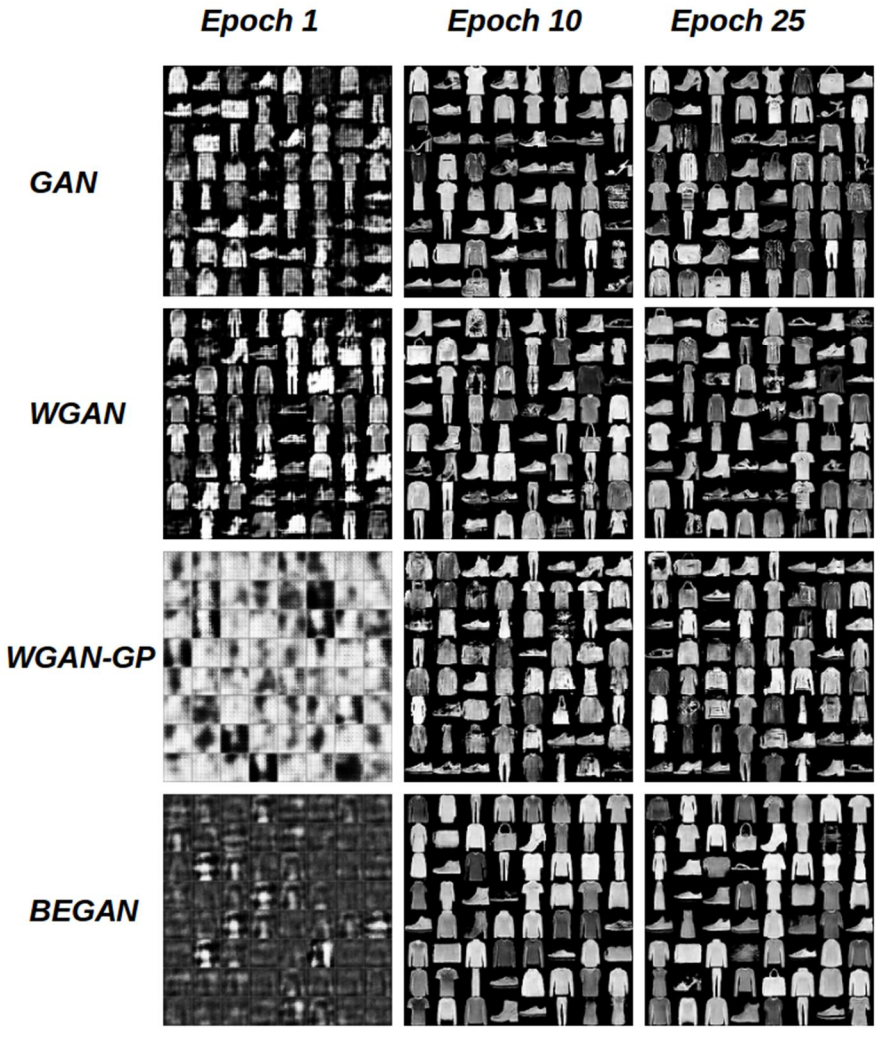

Fig. 1: Generated image samples during different training periods for GAN [2], WGAN [12], WGAN-GP [13] and BEGAN [15] model obtained using Fashion-MNIST dataset

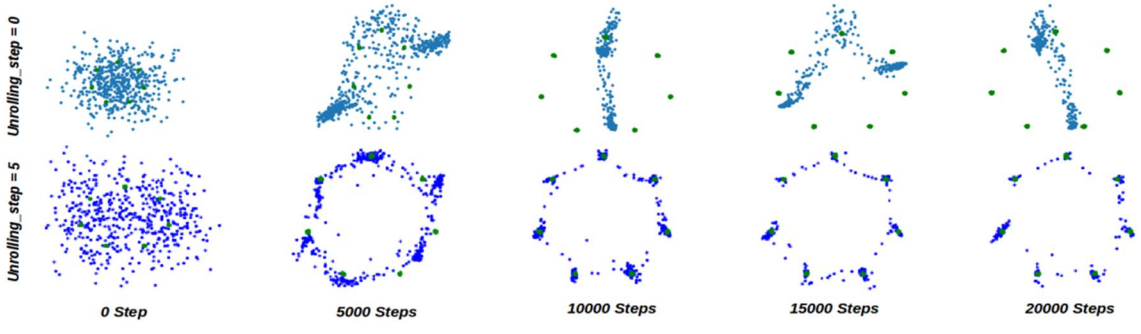

Fig. 3: Behaviour of mode collapse in GAN [2] and unrolled GAN [10]

DCGAN was the first model that use the concepts of CNN which improves the training of GAN. Salimans et al. [9] suggest few techniques to GANs architecture that improves the training stability. As mode collapse is the major issue which occurs due to instability of GANs. Luke et al. [10] introduced unrolled GAN that removes this mode collapse problem but this method required more training time and complicated gradient calculations. While in WGAN, Arjovsky et al. [12] claim that their model never running into mode collapse. By using Wasserstein distance loss function, WGAN model improve the stability of its model. However, it still generates low-quality samples due to the use of weight clipping. Gulrajani et al. [13]

propose new model called WGAN with gradient penalty that overcome the problems of WGAN. In prior GAN models, controlling the image quality and balancing the convergence between generator and discriminator are difficult. David et al. [15] propose a BEGAN model by using the equilibrium term which can balance the power of the discriminator against the generator and therefore model remains stable and achieves meaningful results.
We implemented the above GAN models observe that WGAN [12], WGAN-GP [13] and BEGAN [15] models are better to stabilize the training of GANs. We experiments on these three models on common dataset and compare the results. Here, we found that results obtained using WGANGP are better than GANs and WGAN. However, due to the equilibrium terms in BEGAN, it can result in improved sample quality with fast convergence. Furthermore, our experiments on unrolled GAN [10] conclude that the main issue of mode collapse can be reduced by adding unrolling steps to discriminator to update the generator weights.

Acknowledgment

Authors are gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] A. Kucukelbir, R. Ranganath, A. Gelman, and D. Blei, “Automatic variational inference in stan,” in Advances in neural information processing systems, 2015, pp. 568–576.
[2] 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, 2014, pp. 2672–2680.
[3] I. Goodfellow, “Nips 2016 tutorial: Generative adversarial networks,” arXiv preprint arXiv:1701.00160, 2016.
[4] Y. Sun, X. Wang, and X. Tang, “Deep learning face representation from predicting 10,000 classes,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1891–1898.
[5] J. Grimmer, “An introduction to bayesian inference via variational approximations,” Political Analysis, vol. 19, no. 1, pp. 32–47, 2010.
[6] W. R. Gilks, S. Richardson, and D. Spiegelhalter, Markov chain Monte Carlo in practice. CRC press, 1995.
[7] M. C. Fu, “Stochastic gradient estimation,” in Handbook of simulation optimization. Springer, 2015, pp. 105–147.
[8] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” arXiv preprint arXiv:1511.06434, 2015.
[9] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in Advances in Neural Information Processing Systems, 2016, pp. 2234–2242.
[10] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, “Unrolled generative adversarial networks,” arXiv preprint arXiv:1611.02163, 2016.
[11] D. Warde-Farley and Y. Bengio, “Improving generative adversarial networks with denoising feature matching,” 2016.
[12] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein gan,” arXiv preprint arXiv:1701.07875, 2017.
[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved training of wasserstein gans,” arXiv preprint arXiv:1704.00028, 2017.
[14] J. Zhao, M. Mathieu, and Y. LeCun, “Energy-based generative adversarial network,” arXiv preprint arXiv:1609.03126, 2016.
[15] D. Berthelot, T. Schumm, and L. Metz, “Began: Boundary equilibrium generative adversarial networks,” arXiv preprint arXiv:1703.10717, 2017.
[16] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” arXiv preprint arXiv:1701.04862, 2017.
[17] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. A. Riedmiller, and T. Brox, “Discriminative unsupervised feature learning with exemplar convolutional neural networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, pp. 1734–1747, 2016.
[18] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, “Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop,” arXiv preprint arXiv:1506.03365, 2015.
[19] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3730–3738.
[20] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” CoRR, vol. abs/1708.07747, 2017.