On the Convergence and Mode Collapse of GAN

Zhaoyu Zhang
University of Science and Technology of China
zzy95@mail.ustc.edu.cn

Mengyan Li
University of Science and Technology of China
limmy@mail.ustc.edu.cn

Jun Yu*
University of Science and Technology of China
harryjun@ustc.edu.cn

ABSTRACT

Generative adversarial network (GAN) is a powerful generative model. However, it suffers from several problems, such as convergence instability and mode collapse. To overcome these drawbacks, this paper presents a novel architecture of GAN, which consists of one generator and two different discriminators. With the fact that GAN is the analogy of a minimax game, the proposed architecture is as follows. The generator (G) aims to produce realistic-looking samples to fool both of two discriminators. The first discriminator (D_1) rewards high scores for samples from the data distribution, while the second one (D_2) favors samples from the generator conversely. Specifically, the ResBlock and minibatch discrimination (MD) architectures are adopted in D_1 to improve the diversity of the samples. The leaky rectified linear unit (Leaky ReLU) and batch normalization (BN) are replaced by the scaled exponential linear unit (SELU) in D_2 to alleviate the convergence problem. A new loss function that minimizes the KL divergence is designed to better optimize the model. Extensive experiments on CIFAR-10/100 datasets demonstrate that the proposed method can effectively solve the problems of convergence and mode collapse.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; *Image representations*; Unsupervised learning;

KEYWORDS

GAN, convergence, mode collapse.

ACM Reference Format:
Zhaoyu Zhang, Mengyan Li, and Jun Yu. 2018. On the Convergence and Mode Collapse of GAN. In *SIGGRAPH Asia 2018 Technical Briefs (SA '18 Technical Briefs), December 4–7, 2018, Tokyo, Japan.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3283254.3283282

1 INTRODUCTION

Generative adversarial network (GAN) [Goodfellow et al. 2014] is one of the most powerful generative models that can produce very visually appealing samples, however, it is often difficult to train and suffers from the convergence problem. It is because the gradient of the generator disappears when the best discriminator is obtained[Arjovsky and Bottou 2017]. In addition, GAN also suffers from the mode collapse problem [Saatci and Wilson 2017].

Focusing on the convergence and mode collapse problems, this paper presents a novel GAN architecture which consists of one generator (G) and two different discriminators (D_1 and D_2), inspired by D2GAN [Nguyen et al. 2017]. As shown in Fig.1, the discriminator D_1 rewards high scores for data that are sampled from the distribution of real data (p_{data}) and gives low scores for data that are generated from the distribution of generated samples (p_G). Conversely, the discriminator D_2 is in favor of data generated from p_G and despises data sampled from p_{data}. Although the two-discriminator network is similar to D2GAN conceptually, our network is fundamentally different from it in two respects. First, the loss function of D2GAN will make the value of the loss unbounded during training and the values of the two discriminators are not probabilities in $[0, 1]$, while the loss function of our method can avoid these issues. Second, the two discriminators of D2GAN have an identical architecture, while the discriminator D_1 and D_2 are designed with different architectures in our method, to further address the convergence and mode collapse problems. Experiments on CIFAR-10/100 [Krizhevsky et al. 2014] fully demonstrate the effectiveness of our method. The key contributions of this paper are the following:

(1) Improving the convergence problem of GAN. Focusing on the convergence problem of GAN, our proposed architecture introduces the SELU [Klambauer et al. 2017] in D_2 to ensure that the distribution of the generated data (p_G) is not a set of measure 0 in that of the real data (p_{data}). Therefore, in opposition to [Arjovsky and Bottou 2017], the gradient of our network will not disappear when the best discriminator is obtained, and the network can be trained more thoroughly. A new loss function that minimizes the KL divergence is designed to address the convergence problem further. The higher inception scores on CIFAR-10/100 demonstrate that

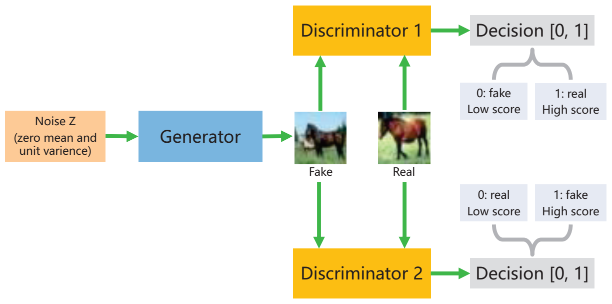

Figure 1: A overview of the architecture of our GAN.

*Corresponding author: Jun Yu

our method addresses the convergence problem of GAN effectively compared with other GANs.

(2) Basically solving the mode collapse problem. In our architecture, two discriminators are used to balance the generator, the minibatch discrimination (MD) [Salimans et al. 2016] architecture is embedded in D_1 to hinder the generator from learning fixed patterns. Compared to other GANs, our method can further increase the diversity of the generated samples. The higher inception scores on CIFAR-10/100 show that our method fixes the mode collapse problem of the GAN.

2 PROPOSED METHOD

In this section, we will present the reason why we using SELU in GAN and prove the convergence of proposed loss function, the detailed architectures of our network can be found in supplementary materials.

2.1 Reason of using SELU

In this section, we explain why SELU can solve the vanishing gradient problem. Based on the loss function of GAN, the best discriminator D^* can be obtained as:

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_G(x)}. \tag{1}$$

From Eq. (1), p_{data} is the distribution of the real data, and p_G is the distribution of the generated data. When p_{data} is equal to p_G, we can conclude that $D^* = 1/2$, which means that the discriminator cannot distinguish the images from synthesized data and real data. Under such circumstances, the optimization of the generator is equal to the optimization of the function $C(G)$, as follows [Goodfellow et al. 2014]:

$$C(G) = -\log 4 + 2JS(p_{data} \parallel p_G). \tag{2}$$

As shown in Eq.(2), the first item of Eq.(2) is a constant that is chosen such that optimizing the generator is equal to optimizing the JS (Jensen-Shannon) divergence [Arjovsky and Bottou 2017] when the best discriminator is obtained.

To explain the JS divergence, KL (Kullback-Leibler) divergence is first introduced as:

$$KL(p_{\text{data}} \parallel p_G) = \underset{x\sim p_{data}}{E} \log \frac{p_{data}}{p_G}. \tag{3}$$

The equation of $JS(p_{data} \parallel p_G)$ is described as:

$$\begin{aligned} JS(p_{data} \parallel p_G) = \frac{1}{2} KL(p_{data} \parallel \frac{p_{data} + p_G}{2}) + \\ \frac{1}{2} KL(p_G \parallel \frac{p_{data} + p_G}{2}). \end{aligned} \tag{4}$$

From Eq.(4), the value of the JS divergence is determined by p_{data} and p_G. If $p_{data} = p_G$, the value of JS divergence in Eq.(4) is 0. Thus, they share the same distribution and the generator is optimal. Furthermore, the value of the JS divergence in Eq.(4) is $log2$ when p_G and p_{data} do not overlap or the overlapping parts can be ignored (as is typically the case). This dynamic means that the two types of distribution are not equivalent, and the gradient of the generator disappears.

Next, the reason why p_G and p_{data} do not usually overlap or the overlapping parts can be ignored is given as follows.

If the supports of p_{data} and p_G are disjoin or lie in low-dimensional manifolds, p_G will be a set of measure 0 in p_{data}. Then, p_G and p_{data} do not overlap, or overlapping parts can be ignored. The proof of this conclusion is given in [Arjovsky and Bottou 2017].

For a vanilla GAN, the input of the generator is a 100-dimensional random noise z. Therefore, p_G will be limited by the random noise z. In a typical case, the distribution of the random noise z is different from that of p_{data} in such a way that the supports of p_{data} and p_G are disjoint or lie in low-dimensional manifolds. p_G and p_{data} do not overlap or the overlapping parts can be ignored. In the situations above, p_G will be a set of measure 0 in p_{data}, and the gradient of the generator disappears [Arjovsky and Bottou 2017]. This scenario is the convergence problem of GAN.

From the perspective of the convergence problem of GAN, the core reason that the gradient of the generator disappears in the GAN is that p_G cannot approach p_{data} more. The distribution of random noise z is different from p_{data}, while p_G depends on p_z. SELU activation function can converge toward zero mean and unit variance. One idea is to enforce the random noise z to be a zero mean and unit variance distribution and to adopt SELU in the discriminator to make the distribution of z the same as that of p_{data}. In this case, p_G and p_{data} will overlap, thus, we can further optimize the generator. The SELU function can converge toward a zero mean and unit variance distribution. By default, the random noise z is set to be a zero mean and a unit variance distribution, while using SELU in the discriminator is done to learn a type of p_{data} from real data with a zero mean and unit variance distribution. In this case, the two distributions p_G and p_{data} will be the same. Thus, the generator will be optimized when the best discriminator D^* is obtained. In other words, which means we solved the convergence problem preliminarily.

2.2 Novel Loss Function

In this section, we will introduce the proposed loss function and prove its convergence ability. The loss function of our network can be formulated as:

$$\begin{aligned} \min_G \max_{D_1,D_2} V(D_1, D_2, G) = & \underset{x\sim p_{data}}{E} [\log D_1(x)] + \underset{z\sim p_z}{E} [\log(-D_1(G(z)))] + \\ & \underset{x\sim p_{data}}{E} [\log(-D_2(x))] + \underset{z\sim p_z}{E} [\log D_2(G(z))], \end{aligned} \tag{5}$$

wherein we have used the different functions between D2GAN and GAN. To prove the convergence of Eq.(5), we first prove the convergence of

$$\begin{aligned} \min_G \max_{D_1,D_2} V(D_1, D_2, G) = & \underset{x\sim p_{data}}{E} [\log D_1(x)] + \underset{z\sim p_z}{E} [\log(1-D_1(G(z)))] + \\ & \underset{x\sim p_{data}}{E} [\log(1-D_2(x))] + \underset{z\sim p_z}{E} [\log D_2(G(z))]. \end{aligned} \tag{6}$$

These two loss functions Eq.(5) and Eq.(6) are almost the same, but the convergence is different. We next analyze the theory in Section 2.2.1.

2.2.1 Theoretical analysis of Eq.(6). Here, we consider the optimization problem with respect to (w.r.t.) discriminators given a fixed generator.

Proposition 1 *Given a fixed G, maximizing $J(G, D_1, D_2)$ yields the following closed-form optimal discriminators $D_1^*(x)$, $D_2^*(x)$:*

$$D_1^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_G(x)} \quad and \quad D_2^*(x) = \frac{p_G(x)}{p_{data}(x) + p_G(x)}$$

Because of the space limits, the detailed proof of proposition 1 is not provided here, which can be found in the supplementary materials.

Next, let $D_1 = D_1^*(x)$, $D_2 = D_2^*(x)$ and the optimal solution G^* for the generator G can be obtained.

Theorem 2. *Given $D_1 = D_1^*(x)$, $D_2 = D_2^*(x)$ at the Nash equilibrium point $\mathcal{J}(D_1^*, D_2^*, G^*)$ for minimax optimization problem of our GAN*:

$$J(D_1^*, D_2^*, G^*) = -4log2,$$

$D_1^* = D_2^* = 1/2\ \forall x$ at $p_{G^*} = p_{data}$.

Liking proposition 1, the detailed proof of theorem 2 can be found in the supplementary materials.

2.2.2 Theoretical analysis of Eq. (5). According to [Goodfellow et al. 2014], the loss function is modified as in Eq.(5). Using this function, the Nash equilibrium point is shown in Theorem 3.

Theorem 3. *Given $D_1 = D_1^*(x)$, $D_2 = D_2^*(x)$ at the Nash equilibrium point $\mathcal{J}(D_1^*, D_2^*, G^*)$ for minimax optimization problem of our GAN*:

$$J(D_1^*, D_2^*, G^*) = 2 \underset{x \sim p_{data}}{E} [\log D_1^*(x)],$$

$D_1^* = D_2^* = 1/2\ \forall x$ at $p_{G^*} = p_{data}$.

Proof. In Eq.(5), the loss function of generator under $D_2^*(x)$ is not changed, and thus, it is still the function with $-2\log 2 + 2JS(p_{data} \parallel p_G)$. We inspect the item $KL(p_G \parallel p_{data})$ and then obtain:

$$\begin{aligned} \mathrm{KL}(p_G \parallel p_{data}) &= \underset{x \sim p_G}{E} \log \frac{p_G(x)/(p_{data}(x) + p_G(x))}{p_{data}(x)/(p_{data}(x) + p_G(x))} \\ &= \underset{x \sim p_G}{E} [\log(1 - D_1^*(x))] - \underset{x \sim p_G}{E} [\log D_1^*(x)]. \end{aligned} \tag{7}$$

From the Theorem 2, it can be formulated as:

$$\underset{x \sim p_{data}}{E} [\log D_1^*(x)] + \underset{x \sim p_G}{E} [\log(1 - D_1^*(x))] = 2JS(p_{data} \| p_G) - 2\log 2. \tag{8}$$

According to Eq.(7) and Eq.(8), minimizing G under D_1^* is equal to minimizing the formula:

$$\begin{aligned} \underset{x \sim p_G}{E} [\log(-D_1^*(x))] = &\underset{x \sim p_{data}}{E} [\log D_1^*(x)] + KL(p_G \| p_{data}) \\ &- 2JS(p_{data} \| p_G) + 2\log 2. \end{aligned} \tag{9}$$

From Eq. (9), the first item on the right is irrelevant to G and it is a constant when we optimize G. So that optimizing G under D_1^* is equal to optimizing the equation with $\mathrm{KL}(p_G \parallel p_{data}) - 2JS(p_{data} \parallel p_G)$. Therefore, the two items are opposite, and they are different to optimize. Hence, we using D_2 to offset the item $-2JS(p_{data} \parallel p_G)$ because the optimization of G under D_2^* will not change, and for the total D_1^* and D_2^*, the Nash equilibrium point is defined as:

$$J(D_1^*, D_2^*, G) = \mathrm{KL}(p_G \parallel p_{data}) + 2 \underset{x \sim p_{data}}{E} [\log D_1^*(x)]. \tag{10}$$

Eq.(10) shows that optimizing Eq.(5) is equal to minimizing the KL divergence between p_G and p_{data}. When $p_G = p_{data}$, Eq.(10) is $2 \underset{x \sim p_{data}}{E} [\log D_1^*(x)]$ and we can obtain the Nash equilibrium point in theory.

The proofs above show that if the loss function is similar to Eq.(6), the discriminator D_2 has less effect, and a loss function similar to Eq.(5) can converge to $KL(p_G \parallel p_{data})$, which means that in optimizing p_G to approach p_{data}, the proposed loss function addresses the disadvantage of negative item $-2JS(p_{data} \parallel p_G)$. Using D_2 balances the negative item $-2JS(p_{data} \parallel p_G)$ compared with the original GAN. This finding means our proposed loss function is effective and solves the convergence problem of the GAN.

3 EXPERIMENTS

We demonstrate the superiority of our proposed network on two datasets (CIFAR-10/100) using our proposed architecture and compare with state-of-the-art GANs. Only part of the experimental results are given here for space limitations, more details can be found in the supplementary materials.

3.1 Evaluation with Inception Scores

Evaluating the quality of the image produced by generative models is notoriously challenging due to the variety of probability criteria and the lack of a perceptually meaningful image similarity metric [Theis et al. 2015]. Even a model can generate plausible images, it is not useful if those images are visually similar. Therefore, to quantify the performance of covering data modes as well as producing high quality samples, we adopt the inception score proposed in [hvy 2017; Salimans et al. 2016], which is computed by:

$$exp(E_x [D_{KL}(p(y|x) \parallel p(y)])$$

where $p(y|x)$ is the conditional label distribution for image x that is estimated using a pretrained inception model, and $p(y)$ is the marginal distribution: $p(y) \approx 1/N \sum_{n=1}^{N} p(y|x_n = G(z_n))$. This score can adequately reflect the variety and visual quality of the images.

3.2 Results on CIFAR-10/100 Dataset

In this section, we will present the results of CIFAR-10 dataset, results of CIFAR-100 dataset can be found in the supplementary materials. The values of the inception scores on our model and the others models are shown in Table 1, all the others GANs are best run results. Some inception scores of the others GANs are different from the original papers in that we use chainer [pfnet research 2017; Tokui et al. 2015] not tensorflow to obtain the inception scores. This is because more and more work of GAN using chainer, using chainer is convenient to evaluate the model. Because GAN will obtain different results in different times, we show the average inception scores computed from 5 times run and the value is 7.47, this value is still higher than other GANs. To further showing our GAN is better, the images generated on CIFAR-10 can be found in Fig.2, it can be seen that the images generated by our GAN is better and more diversity.

3.3 Contrastive Experiments

In this section, results of contrastive experiment on CIFAR-10 dataset are shown in order to validate the effectiveness of our method. The results on Table 2 and Table 3 shows that our method is effective and obtain a good performance. Contrastive Experiments include two parts which are different architectures and different loss functions. Results of compared with different architectures can be found in Table 2, Results of compared with different loss functions can be found in Table 3. From the results we can see that our method obtain a higher inception scores compared with others. which also means the theory and experiment are coincident and our method is better.

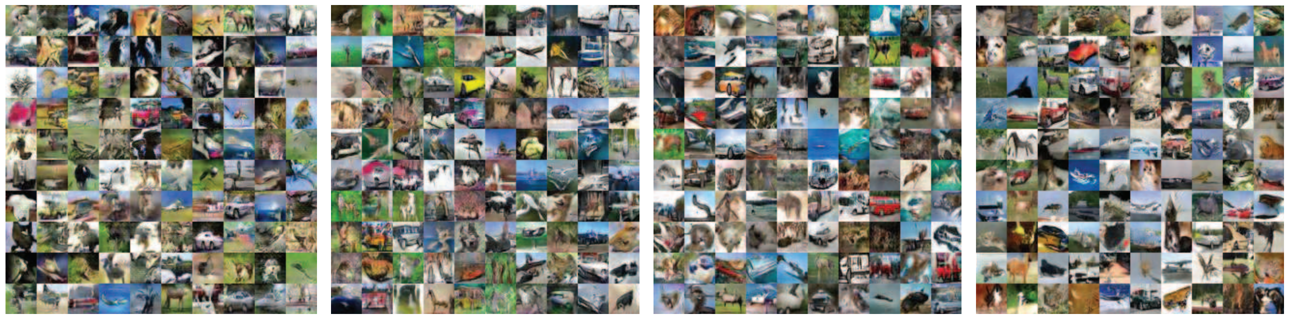

Figure 2: The results of different GANs applied to CIFAR-10 dataset.

Table 1: Inception scores on the CIFAR-10 dataset.

Method	Inception scores (chainer[Tokui et al. 2015])
Real data	12.00
WGAN-GP [Gulrajani et al. 2017]	6.80
DFM [Warde-Farley and Bengio 2016]	7.30
Cramer GAN [Bellemare et al. 2017]	6.40
DRAGAN [Kodali et al. 2018]	7.10
DCGAN [Radford et al. 2016]	6.70
MD [Salimans et al. 2016]	7.00
BEGAN [Berthelot et al. 2017]	5.40
D2GAN [Nguyen et al. 2017]	6.76
Our GAN (best run)	**7.65**
Our GAN (computed from 5 runs)	**7.47**

Table 2: Experiments on different architectures.

Method	Inception scores (chainer[Tokui et al. 2015])
Our GAN (without MD)	6.95
Our GAN (without SELU)	7.36
Our GAN (both using MD)	7.52
Our GAN (both using SELU)	7.06
Our GAN (without ResBlock)	7.32
Our GAN	**7.65**

Table 3: Experiments on different loss function.

Method	Inception scores (chainer[Tokui et al. 2015])
Our GAN (Hingle loss)	7.41
Our GAN (our loss+gp)	6.94
Our GAN (Hingle loss+gp)	6.86
Our GAN	**7.65**

4 CONCLUSIONS

We present a novel architecture of GAN, which consists of one generator and two different discriminators. Extensive experiments demonstrate that our method obtains high inception scores on the CIFAR-10/100 datasets. In the future, we will focus on ImageNet [Deng et al. 2009] to obtain a higher value of the inception score and to train a better GAN model.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China U1736123, 61572450. Anhui Provincial Natural Science Foundation 1708085QF138 and the Fundamental Research Funds for the Central Universities WK2350000002.

REFERENCES

Martin Arjovsky and Léon Bottou. 2017. Towards principled methods for training generative adversarial networks. *In ICLR* (2017).
Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. 2017. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743* (2017).
David Berthelot, Tom Schumm, and Luke Metz. 2017. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717* (2017).
Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 248–255.
Ian 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 NIPS.* 2672–2680.
Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in NIPS.* 5769–5779.
hvy. 2017. chainer-inception-score. *online: https://github.com/hvy/chainer-inception-score* (2017).
Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. In *Advances NIPS.* 972–981.
Naveen Kodali, James Hays, Jacob Abernethy, and Zsolt Kira. 2018. On convergence and stability of gans. (2018).
Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. 2014. The CIFAR-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html* (2014).
Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. 2017. Dual discriminator generative adversarial nets. In *Advances in NIPS.* 2667–2677.
pfnet research. 2017. chainer-GAN-lib. *online: https://github.com/pfnet-research/chainer-gan-lib* (2017).
Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *In ICLR* (2016).
Yunus Saatci and Andrew G Wilson. 2017. Bayesian GAN. In *Advances in NIPS.* 3622–3631.
Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in NIPS.* 2234–2242.
Lucas Theis, Aäron van den Oord, and Matthias Bethge. 2015. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844* (2015).
Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, Vol. 5.
David Warde-Farley and Yoshua Bengio. 2016. Improving generative adversarial networks with denoising feature matching. (2016).