

Stabilizing Generative Adversarial Networks: A Survey

Based on <https://arxiv.org/pdf/1910.00927.pdf>

Generative Adversarial Networks (GANs) are generative model that are very popular since they can reproduce real-world data. Nevertheless, training GANs is still a challenging problem, due to non-convergence, exploding or vanishing gradients, and mode collapse. Recently, many methods have been proposed to improve stabilizing the GAN training process. Here you will see existing GAN training stabilization methods with their advantages and Disadvantages.

Generative Adversarial Networks (GAN) is a generative model trained by deep learning methods, such as convolutional neural networks. Generative model could be seen as an unsupervised task that discovers patterns in input data which can be used to generate new examples from the original dataset. The GAN method was originally described in Ian Good fellow paper, titled “Generative Adversarial Networks.” (2014)

The Generative Adversarial Networks model has two sub-models:

- Generator (G) model to generate new samples
- Discriminator (D) model to classify whether generated samples are real or fake

It’s like a game. The generator creates new samples. And the discriminator tries to differentiate samples between the input data and from the generator. As a result of such confrontational training, GANs can generate realistic images.

GANs accomplish state-of-the-art results in the generation of images, NLP, time-series and other domains. But despite good results, the training of GAN has proven to be unstable.

The main causes of this instability are:

1. Convergence

In the original paper Goodfellow (2014) showed that at $D = 1/2$ the loss function :

$$L(D, G) = \mathbb{E}_{x \sim p_r} \log[D(x)] + \mathbb{E}_{z \sim p_z} \log[1 - D(G(x))] \quad (1)$$

relates to a Nash equilibrium (NE) in which neither actor can improve their score unilaterally (Nash, 1951). In practice, achieving this equilibrium/convergence is not straightforward since discriminator and generator are neural networks. The optimization relates to the parameter space of these networks, instead of direct learning the probability density function. Moreover, the non-convex-concave nature of the game makes it very hard for the gradient descent ascent (GDA) to converge, regularly resulting in diverging, oscillating, or cyclic behavior (Mertikopoulos, 2017; Salimans, 2016; Goodfellow, 2015) and proneness to convergence to a local Nash equilibrium (NE) in the parameter space of the neural networks.

Also, the local NE can be far away from the global NE, preventing the discriminator being optimal in this case. Lastly, training the discriminator to optimal result for every iteration of the generator is computationally expensive.

2. Exploding or Vanishing Gradients

According to the theory of Nash equilibrium (NE) that Goodfellow supported, a discriminator near to optimality would create meaningful response to the generator, thus causing in an improved generator. But, practically the cost of the well trained discriminator approaches to 0.

Such a perfect discriminator squeezes the loss function to 0, making gradients close to zero, which gives small response to the generator, slow down or totally stopping the learning.

Later, as the discriminator gets better at differentiating real and generated samples, the updates to the generator get worse.

When discriminator is so perfect, its gradients will reach zero on nearly every data point, giving no feedback to the generator. To fix the vanishing gradients, Goodfellow (2014) suggested another loss function for the generator - $\log D(G(z))$, which is named as non-saturating (NS) loss.

Though this loss variation improves the vanishing gradients problem, change to the loss still producing less stable and oscillating training.

3. Mode Collapse

Mode collapse occurs when the generator maps various inputs to the same output.

It means that the generator creates samples of low diversity (Huszar, 2016; Theis, 2015).

It is a problem for training, because generator with no ability to create various samples is not valuable for training purposes.

It is a difficult problem, since the reason of mode collapse is going deeply in the concepts of GANs.

There are mainly five GAN Variants and Stabilization Techniques currently: architecture, loss function, game theory, multi-agent, and gradient-based.

Modified Architectures

Different architectures of GANs have been shown to improve the stability of training in GANs.

However, the scale at which these methods achieve the improvements of the training differs widely.

A current solid model is BigGAN, which improves the training in all of the aspects.

However, it is computationally expensive, and improvement leads to increase in complexity of the model.

Different architectures of GANs is a developing and active field, offering small performance enhancements, particularly for application-specific GANs.

The nonconvex-concave model of the GAN game is a task that will require improvements outside of changes in architecture.

Modified Loss Functions

Different loss functions of GANs are very heavily studied after the architecture variations at this time. Some variations of the loss function overcome the vanishing gradient problem (Arjovsky, 2017; Jolicoeur-Martineau, 2019; Miyato, 2018; Mao, 2016; Qi, 2018).

Moreover, compare to architecture types, they propose more generalization properties. Methods like WGAN or SN-GAN being commonly applied and proposing opportunities for further research (Pfau and Vinyals, 2017).

Also, the quality of computational complexity and generated data is not proportional to some of the various architecture of GANs such as BigGAN (Wang et al., 2019; Brock et al., 2018).

Nevertheless, variations of loss function mostly work only with certain restrictions (Arjovsky, 2017; Gulrajani, 2017), and appropriate hyperparameter tuning often lead to produce superior results (Lucic, 2018). The stabilization of the training process is still a challenge.

Game Theory for GANs

Currently, variations of the game theory of GANs are limited. Proposed methods very restrictive, therefore not so often applicable. Still, work such as (Arora et al., 2017) offers outstanding theoretical analysis for dealing with problems in the training process and releases an opportunity for further research.

Multi-Agent GANs

At this time, works related to multi-agent GANs often revolves the simple approach of hindering mode collapse by the introduction of various discriminators and generators. While the results are encouraging, the computational costs present a significant problem. And it is not obvious how many of generators or discriminators should be taken. Lastly, various methods in multi-agent learning works are not connected to stabilizing GAN training (Papoudakis, 2019; Loweal, 2017; He, 2016).

Modified Gradient Optimization

The original work for GANs with gradient-based variations demonstrate improvements in convergence and avoiding mode collapse in contrast to GANs employing gradient descent ascent (GDA). Though these results could be achieved with appropriate tuning of hyperparameters.

Gradient-based methods stay mostly unexplored area with promising research opportunities. With GDA's limitations in general games, gradient variants offer an extra edge, and could be combined with other methods.

Here are other GAN methods which are not related to prior categories.

First is the Evolutionary GAN (Wang) that adjusts the GAN to the environment through a chain of operations on population of discriminators and generators population.

AdaGAN (Tolstikhin, 2017) offers a boosting method to iteratively report the problem of missing modes. A deeply theoretical method was proposed by Unterthiner (2018) who suggests a method and a proof to escape local equilibria by applying energy-based potential fields. Another method to escape mode collapse suggests marginalizing the weights of the discriminator and generator presented in the Bayesian GAN (Saatchi and Wilson, 2017).

General Stabilization Heuristics

Many studies on GANs (Lucic, 2018) has presented that optimization of hyperparameters tends to achieve better results compared to specific methods like modified architectures and loss functions. Additionally, other works in this area (Fedus, 2018; Kurach, 2018) have confirmed the effect of regularization and normalization methods on various loss function and GAN architecture.

Conclusion

While we have seen an increase of published works providing methods to overcome GAN instability of training, many of these studies have limitation to provide only one training problem and often limitation of rigorous theoretical explanation.