Supervised Contrastive Generative Adversarial Networks

Honglei Gu

The experimental high school attached to Beijing Normal University, Beijing, 100032, China

raymondguhonglei@163.com

Abstract. Generative Adversarial Networks (GANs) is becoming more and more popular, artists use them to find their own inspirations, computer scientists use it for data synthesis, workers use it for machine fault diagnosis and so on. However, GANs are flawed despite its popularity: they are unstable. GANs are based on game theory. In a typical GAN model, the generator and the discriminator are both improved by competing with each other. Therefore, in this highly competitive training process, GANs can easily run into trouble while they move towards the optimal solution. In most cases, the case of such instability arises from the loss function, or in other words, the gradient of the loss function. This research proposed a new set of GAN that replaces its objective function with supcon, or the supervised contrastive loss to solve gradient-related problems. We have also proved that under our model, the GANs are less likely to suffer from these two factors of instability. Finally, we have compared our model and the traditional generative adversarial nets.

Keywords: Generative Adversarial Networks, Contrastive learning, 2C loss, Machine Learning.

1. Introduction

Generative Adversarial Nets are first proposed by Goodfellow et al. [1, 2] and has achieved great success in generating realistic data, images in particular and multiple variants of generative adversarial nets have tried to improve the model by altering its structure and objective function. Throughout the past few years, GANs remains to be one of the most popular fields in deep learning. However, there is one major problem that have been noticed by scholars – the train process of GANs are not stable. This instability could lead to serious mode collapse.

Many scholars have attempted to solve the problem with numerous approaches by proving some aspects of the traditional GANs theoretically and try to apply some additional transformations to stabilize the training process. Nagarajan et al. [3] has proved that the original GANs with gradient based learning is locally stable. Heusel et al [4]. introduced a Two Timescale Update Rule(TTUR) into the traditional GAN training and theoretically prove that it converges to local equilibrium.

Contrastive learning has also gained much attention in learning visual representations with or without supervision. The common idea of contrastive losses is pulling the anchor and a positive sample together in the embedding space, do- ing just the opposite to the positives. Furthermore, Cui. et al. introduced a set of parametric class-wise centers to help contrastive losses deal with noisy datasets. The examples above are just a tiny portion of the scholars attempts. However, we believe that changes in the objective functions are necessary to further stabilize GANs'training. Due to the excellent performances of contrastive losses, we proposed a new contrastive objective function in the GAN model based on the contributions of supcon loss [5] and 2C loss. Our contributions are as follows:

(1) We implemented the supcon loss into the original GAN framework to replace the original objective function as the original GAN objective encourages gradient explosion in the discriminator and thus lead to mode collapse.

(2) We prove that the GANs trained by SupCon loss are less likely to suffer gradient explosion or gradient disappearance. Thus gaining more stability in the training process.

2. Related work

Nagarajan et al. [3] and Heusel et al. [4] has proven their GANs to be locally stable. Arora et al. [6] proved that a generator can deceive the discriminator by recording a set of training sample, thus implying that low capacity discriminators lack the ability to distinguish the lack of diversity, in other words, the generator is unable to learn the target distribution. Except these theoretical re- searches, many practical trick have been used to stabilize the training process of GANs. Radford et al. [7] performed a variety of empirical tricks to help stabilize GANs. Arjovsky and Bottou [8] signals the importance of divergences in GAN training. They also introduced Wasswestein distance in to the GAN model, creating Wasserstein GANs, or WGANs [9]. The traditional form of GANs is based on game theory where a discriminator and a generator compete on the same loss function and can be converted into a minmax problem. The objective function of traditional GANs are as follows:

$$\min_{G}\max_{D} E_{xP_{data}(x)}[log\, D\,(x)] + E_{xP_{x}(z)}[log(\,1 - D(G(x)))]$$

where G is the discriminator, D is the discriminator, P_{data} is the data distribution over the dataset and P_z is the data distribution over the generated samples. This minmax formula can be seperated into two parts, First:

$$E_{xP_{data}(x)}[log\, D\,(x)]$$

This part of the formula tries to maximize the discriminator's ability to distinguish the real samples from the generated samples. Second:

$$E_{xP_{z}(z)}[log(\,1 - D(G(x)))]$$

where G is the discriminator, D is the discriminator, P_{data} is the data distribution over the dataset and P_z is the data distribution over the generated samples. The rest part of the formula tries to maximize the generator's ability to cheat the discriminator. So we could notice that the generator wants its generated samples to be closer to the real images, and the discriminator wanted the sample he classified as real are real samples and the samples it classifies as fake to be the generated samples. The two models compete against each other to finally reach an equilibrium. Besides GANs, supervised contrastive learning has also become a major focus. Khosla et al.[5] introduced the unsupervised contrastive loss in the supervised learning framework and proposed SupCon. In Supcon, the loss is illustrated as follows:

$$L_i = \sum_{z_k \in A(i)} log \frac{exp(\,z_+ \bullet T(x_i))}{\sum_{z_k \in A(i)} exp(\,z_k \bullet T(x_i))}$$

Where

$$w(z_+) = \begin{cases} \alpha, z_+ \in P(i) \\ 1.0, z_+ \in \{c+y\} \end{cases}$$

And

$$z\bullet\ \mathrm{T}(\mathcal{X}_i) = \begin{cases} z\bullet \mathrm{G}(x_i), z \in A(\mathcal{i}) \\ z\bullet \mathrm{F}(\mathcal{X}_i), x \in\ C \end{cases}$$

Where $P\ (i)$ is the distribution of the dataset, $w(z_+)$ is the parametric weight, C is the set of class-wise centers and *tt*() is a multi-layer perceptron and F () is the identity mapping or $F(x) = x$. The loss is scale with a temperatureτ applied to the loss function. Furthermore Kang et al. incorporated the concept of contrastive learning into the GAN framework. Kang also improve the loss into 2C loss so it better fits the GAN's goal. Originally, the SupCon loss is like the three equations above. However, the Supcon loss, along with many other contrastive losses, NT Xent, for example, did not consider any data-to-class representations. Therefore, Kang introduced a set of class embedding function in the 2C loss so it suit the goal of GANs better. The pesudocode for training Kang's ContraGAN with 2C loss along with the 2C loss are as follows:

$$l_{2C}(x_i, y_i; t) = -\,log(\frac{exp(\,l(x_i)^T e(y_i)/t) + \sum_{k=1}^{m} I_{y_k = y_i} \bullet \exp(l(x_i)^T l(x_k)/t)}{exp(\,l(x_i)^T e(y_i)/t) + \sum_{k=1}^{m} I_{k \neq i} \bullet exp(\,l(x_i)^T l(x_k)/t)})$$

Inspired by Kang's work but still consider 2C loss as overly complex, we decided to develop our own set of contrastive GANs. Currently, two of the simplest contrastive losses are Supcon [5] and its dataset imbalance tolerant variant Paco [10]. Although in the traditional supervised learning framework, noise and dataset imbalance are two main factors that may lead to a low model quality. In the generative adversarial networks, where almost half of our samples are generated and the class labels sometimes do not affect the overall quality of the generator and the generated images. In addition, the lack of data-class representation in the loss can be solved by only comparing images from different classes, in other words, the latent feature of the generated images and the latent feature of the original real pictures. In this method, it will bring the generated samples closer to the contrasted real images and the discriminator maximizes the loss function, it pushes the generated samples away from the real images Therefore, our final loss function are as follows:

$$L = -\frac{\sum_{x_+ \in P(i)} \sum_{x_- \in G(z)} exp(\,x_+ \bullet x_-/\tau)}{\sum_{x \in P(i)} \sum_{y \in P(i)} I_{x \neq y}\ exp(\,x \bullet y/\tau)}$$

3. Method

3.1. Loss

We implement Supcon loss to help it gain stability in the dangerous training periods. The main feature of the loss has been mentioned in section two. But allow me to display the the formula again for the gradient analysis section and further analysis that will be illustrated below.

$$L = -\frac{\sum_{x \in P(i)} \sum_{x_- \in G(z)} exp(\,Z_x \bullet Z_y/\tau)}{\sum_{x \in A(i)} \sum_{y \in A(i)} I_{x \neq y}\ exp(\,Z_x \bullet Z_y/\tau)}$$

Where P(i) is the real images and G(z) is the set of generated samples and z_x stands for the feature of x when it passes through the discriminator. The samples that belong to the same class are multiplied together, then added into the loss function and divide a common denometer to normalize the loss. The loss has the following attractive properties: First and the most important is that it prevents the GAN model from suffering gradient catastrophes. we offer a proof to our conclusion in the gradient analysis section. Second, it requires less computational effort to calculate the gradient, thus gaining faster training speed. Third, as elaborated by [5], this loss has great ability to perform hard positive or negative mining.

$$l_{2C}(x_i, y_i; t) = -\,log(\frac{exp(\,l(x_i)^T e(y_i)/t) + \sum_{k=1}^{m} I_{y_k=y_i} \bullet \exp(l(x_i)^T l(x_k)/t)}{exp(\,l(x_i)^T e(y_i)/t) + \sum_{k=1}^{m} I_{k\neq i} \bullet exp(\,l(x_i)^T l(x_k)/t)})$$

3.2. Gradient analysis

The gradient of this loss function can be calculated with the following equations, for the sake of clarity, we define $A(i) = G(z) \cup P(i)$, therefore:

$$L = -\frac{\sum_{x_+\in P(i)} \sum_{x_-\in G(z)} exp(\,x_+ \bullet x_-/\tau)}{\sum_{x\in P(i)} \sum_{y\in P(i)} I_{x\neq y}\, exp(\,x \bullet y/\tau)}$$

$$\begin{aligned}
\frac{\partial L_i}{\partial z_i} &= \frac{1}{|P(i)|} \sum_{p\in P(i)} \frac{\partial}{\partial z_i} \{\frac{z_i \bullet z_p}{\tau} - log \sum_{a\in A(i)} exp(\,z_i \bullet z_a)\} \\
&= \frac{-1}{\tau|P(i)|} \sum_{p\in P(i)} \{z_p - \frac{\sum_{a\in A(i)} z_a\, exp(\,z_i \bullet z_a/\tau)}{\sum_{a\in A(i)} exp(\,z_i \bullet z_a/\tau)}\} \\
&= \frac{-1}{\tau|P(i)|} \sum_{p\in P(i)} \{z_p - \sum_{p'\in P(i)} z_{p'} P_{ip'} + \sum_{n\in G(z)} z_n P_{in}\} \\
&= \frac{-1}{\tau|P(i)|} \sum_{p\in P(i)} \{\sum_{p\in P(i)} z_p - \sum_{p\in P(i)} \sum_{p'\in P(i)} z_{P'} P_{in} + \sum_{n\in G(z)} \sum_{p\in P(i)} z_n P_{in}\} \\
&= \frac{-1}{\tau|P(i)|} \{\sum_{p\in P(i)} z_p - \sum_{p'\in P(i)} |P(i)| z_{p'} P_{ip'} + \sum_{n\in N(i)} |P(i)|\, z_n P_{in}\} \\
&= \frac{1}{\tau} \{\sum_{p\in P(i)} z_p (P_{ip} - \frac{1}{|P(i)|}) - \sum_{n\in G(z)} z_n P_{in}\}
\end{aligned}$$

Furthermore, if the model gradually reaches optimal state, as stated by Goodfellow [1], the distribution of the real data will gradually become equal to the distribution of the generated samples. In other words:

$$P(i) = G(z)$$

Under such circumstances, we can further our analysis on the gradient:

$$\frac{\partial L_i}{\partial z_i} = \frac{1}{\tau} \{\sum_{p\in P(i)} z_p P_{ip} - \sum_{n\in G(z)} z_n P_{in} - \frac{\sum_{p\in P(i)} z_p}{|A(i)|}\}$$

Therefore, this the gradient of this loss will not explode or disappear as the generator reaches optimal.

4. Experiments

This paper mainly conduct research on the CIFAR10 and MNIST dataset.

4.1. Dataset

MNIST [11] (Modified National Institute of Standards and Technology database) is a large hand written digits dataset, it consists of over 60000 images and is widely used as a basic dataset to test a model's quality. CIFAR10 also consists over 60000 images, but different from MNIST, its data were colored and consists of ten classes.

4.2. Stability comparison

As stated before in section1, a major contribution of our GANs is that it helps to improve the stability of the model. Following Arjovskky et al. [9] we construst two models to compare stability One is created by deleting the batch normalization layer in the generator. another omitting all batch normalizations. At the same time, we would like to test the difference between different optimizer, Adam and RMSProp in particular. These combinations meant four different models need to be constructed for comparison [12-19]. We train the models above on CIFAR10 as differences in color pictures are more easily captured. Regular GAN suffer great mode collapse while Contrastive GANs usually perform better than traditional GANs with or without batch normalization layers. Like Arjovsky et al [9] mentioned. We also noted that when generated samples becomes worse, Adam [15] displays a negative cosine with its gradient and thus indicates instability. Similarly, we abandoned Adam and turned to RMSProp [14]. Besides this, we also find out that the images generated by GANs with supcons tend to have a smoother loss curve than the traditional GANs, indicating improved stability.

5. Conclusion

In this research, we created a GAN model based on SupCon loss to improve the stability of the model. By analyzing it gradient, we proved that GANs with this loss are less likely to suffer gradient explosion or gradient disappearance. We also conducted experiments to further prove that our model is more stable.

Acknowledgement

Although academic activities related to machine learning have never been more active nowadays, I wouldn't have realized this field of study if it wasn't for my teacher MaJing. No other people assisted in the completion of this paper, and I completed this paper all by myself.

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2020). Generative adversarial networks. Commu- nications of the ACM, 63(11), 139-144.

[2] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, Least squares generative adversarial networks, in IEEE International Conference on Computer Vision, pp. 2794-2802, 2017.

[3] Nagarajan, V., Kolter, J. Z. (2017). Gradient descent GAN optimization is locally stable. Advances in neural information processing systems, 30.

[4] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30.

[5] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., ... Krishnan, D. (2020). Supervised contrastive learning. Advances in Neural Information Processing Systems, 33, 18661-18673.

[6] Arora, S., Risteski, A., Zhang, Y. (2018, February). Do GANs learn the distribution? some theory and empirics. In International Conference on Learning Representations.

[7] Radford, A., Metz, L., Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

[8] Arjovsky, M., Bottou, L. (2017). Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862.

[9] Arjovsky, M., Chintala, S., Bottou, L. (2017, July). Wasserstein generative adversarial networks. In International conference on machine learning (pp. 214-223). PMLR.

[10] Cui, J., Zhong, Z., Liu, S., Yu, B., Jia, J. (2021). Parametric contrastive learning. In Proceedings of the IEEE/CVF international conference on com- puter vision (pp. 715-724).

[11] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[12] Krizhevsky, A., Hinton, G. (2009). Learning multiple layers of features from tiny images.

[13] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen X. (2016). Improved techniques for training gans. Advances in neural in- formation processing systems, 29.

[14] Tieleman, T., Hinton, G. (2012). Rmsprop: Divide the gradient by a run- ning average of its recent magnitude. coursera: Neural networks for machine learning. COURSERA Neural Networks Mach. Learn.

[15] Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[16] Krizhevsky, A., Hinton, G. (2009). Learning multiple layers of features from tiny images.

[17] Kang, M., Park, J. (2020). Contragan: Contrastive learning for conditional image generation. Advances in Neural Information Processing Systems, 33, 21357-21369.

[18] Chen, T., Kornblith, S., Norouzi, M., Hinton, G. (2020, November). A simple framework for contrastive learning of visual representations. In In- ternational conference on machine learning (pp. 1597-1607). PMLR.

[19] Karras, T., Laine, S., Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4401-4410).