Proceedings of CONF-MLA 2024 Workshop: Semantic Communication Based Complexity Scalable Image Transmission System for Resource Constrained Devices
DOI: 10.54254/2755-2721/83/2024GLG0064

Enhancing WGAN performance by architectural and optimizer variations for image generation

Liuding Wang

School of Computer Science and Technology, East China Normal University, Shanghai, 200241, China

10215102462@stu.ecnu.edu.cn

Abstract. Generative model has opened up the area of image generation and has become a hot topic in recent years. Among the most famous generative models, Generative Adversarial Network (GAN) is outstanding among them, offering extensive avenues for exploration. The Wasserstein GAN (WGAN), as one of the GANs, introduces an innovative framework for training GANs based on the Earth Mover's (Wasserstein) distance, providing a steadier training process. The experiment tried various modifications to WGAN, including changing the optimizers and the network architecture. Specifically, this work tried replacing the original Root Mean Square Prop (RMSprop) with another optimizers. Also, this work tried to add residual blocks to the network structure. These modifications provided interesting results, providing supplementary validation of the original WGAN structure, and providing some possibilities of optimization. According to the results, it could be found that the results of some modifications are very positive. However, some of the changes presented very unsatisfactory results, which gave us some insight.

Keywords: Image generation, Generative adversarial network, Optimizer.

1. Introduction

Recently, the field of image generation has grown rapidly. Among them, Generative Adversarial Network (GAN) is unique because of its generation mode and excellent generation effect, and widely used for projects like image generation and editing, data enhancement, text generation and translation [1,2]. GANs, first presented in 2014 by Ian Goodfellow and associates, are designed to provide artificial data that can be distinguished from real data [3]. The architecture of Gans is based on an adversarial framework consisting of two neural networks: a Generator and a Discriminator. The two networks compete with one another during training and work together to enhance performance. In more detail, the task of Generator(G) is to generate synthetic data by using a random noise vector and generate data that is as realistic as possible to confuse the discriminator and make it judge the generated images as real. The Discriminator(D)'s role is to receive input from the real data set and the data generated by the generator, and accurately distinguish between the actual data and the generated data [4].

The generator adjusts its weight through Backpropagation and gradient descent, so that the generated images can fool the discriminator to the maximum extent possible. During the training process, generators and discriminators are trained alternately. Typically, the generator is fixed first and updates are made to the discriminator weights. Then, fix the discriminator and update the generator's weights

DOI: 10.54254/2755-2721/83/2024GLG0064

GAN may encounter convergence problems in practical training [5]. The generator can get into a mode crash and only generate a limited sample of data. Also, Gradient vanishing or gradient explosion may occur.

Wasserstein GANs (WGANs) are an important improvement over traditional GANs and are designed to solve common problems encountered during the training process of Gans, for example, mode collapse, training instability, and convergence difficulties [6]. The main contribution of WGAN is the introduction of The Earth Mover's Distance (EM distance), commonly referred to as the Wasserstein Distance, which replaces the Jensen Shannon divergence (JS divergence) used by traditional GANs as a loss function [7].

Also, WGAN ensures that the calculation of Wasserstein distance is reasonable by introducing Lipschitz continuity constraint into the loss function. To achieve this, WGAN uses the weight clipping, ensuring the weight of the discriminator is in a proper range. The weight clipping improves the stability of the training process.

This work made some changes to WGAN to try to achieve better results by replacing the optimizer used in WGAN into others, like Adam which is commonly used. Also, this paper tried to change the network structure of WGAN by adding residual blocks. These changes show interesting results.

2. Method

2.1. Principle of WGAN

The two primary parts of a GAN are a discriminator and a generator. The discriminator's job is to distinguish between true and false data, whereas the generator is to produce fake data that mimics the distribution of real data. The generator and discriminator engage in a competitive game with one another to optimize their individual loss functions, resulting in the generator generating realistic data. In standard GANs, the loss function depends on JS divergence. Sometimes training is unstable and pattern breakdown occurs [6].

In WGAN, the difference between the generated data distribution and the real data distribution is measured using the EM distance.

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in (\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||] \quad (1)$$

, where $(\mathbb{P}_r, \mathbb{P}_g)$ denotes the set of all joint distributions $(x, y)\sim\gamma$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_g$. Intuitively, $(x, y)\sim\gamma$ indicates how much "mass" must be transported from x to y in order to transform the distributions $\mathbb{P}_r$ into the distribution $\mathbb{P}_g$. And the EM distance then is the "cost" of the optimal transport plan.

In WGAN, the discriminator is called Critic, and its task is no longer to provide the probability that the sample is accurate, but to output a real number that evaluates the Wasserstein distance between the generated image distribution and the actual image distribution. So, the discriminator attempts to maximize the distance between the two expectations, while the generator attempts to reduce the generated sample's discriminator score. Also, to ensure that the Critic meets the Lipschitz condition, WGAN uses weight clipping, limiting the weight to a fixed range after updating the weight of critic.

2.2. Model architecture

2.2.1. Residual Network (ResNet). ResNet is a deep convolutional neural network structure proposed by He Kaming et al. ResNet's primary goal is to address the issue of gradient disappearance in deep neural networks through the Residual Block [8]. Thus, the network can deepen the number of network layers while maintaining efficient learning. A typical ResNet structure consists of multiple residuals stacked on top of each other. In this experiment, this work replaced some of the convolution blocks in generator and discriminator in the original WGAN structure with ResBlocks, adding residual connections to the structure. Residual networks can make the network structure deeper and more

DOI: 10.54254/2755-2721/83/2024GLG0064

complex, and thus have stronger ability of extracting feature. The author tried to explore how adding a residual network to WGAN's architecture would affect its performance.

2.2.2. Densely Connected Convolutional Networks (DenseNet) is a kind of deep neural network designed by Gao Huang et al. It enhances the efficiency of gradient propagation by introducing dense connections into the network, thus improving the performance and trainability of the network [9]. In DenseNet, each layer receives feature maps directly from all the preceding layers as input, not just the result of the previous layer. The dense connection allows the features of the previous layer to be shared by the later layers, enhancing the training efficiency and performance of the model.

2.3. Optimization algorithm

This work conducts experiments on an improved WGAN algorithm. The target distribution to learn is the Large-Scale Scene Understanding (LSUN) dataset, a collection of images widely used in computer vision research, the same dataset as used in WGAN.

2.3.1. Adam. Adam's main concept is to dynamically adjust each parameter's learning rate by calculating the first-order momentum (i.e. the mean of the gradient) as well as the second-order momentum (i.e. the mean of the square of the gradient). This adjustment allows Adam to use different learning rates in different dimensions, thus finding the optimal solution faster during training.

2.3.2. SGD. This approach is a variation of the conventional gradient descent method that speeds up the training process and increase the randomness of parameter by updating model parameters with only a portion of the samples at each iteration. Its advantages include low computational cost, low memory requirements, and suitable for processing large data sets.

3. Results

3.1. Dataset

This work used the traditional data set cifar-10. The dataset consist of 60,000 pieces of 32x32 pixel color images, divided into 10 categories. This dataset is widely used in image generation and has the characteristics of diverse categories and standard format [11].

3.2. Performance Comparison

This paper changed some of the configuration in WGAN, especially different optimizers. Firstly, this work tried different optimizers to get better results, as demonstrated in Figure 1, 2,3,4,5 and 6. Secondly, this work tried to replace the convolutional neural network with ResNet, as shown in Figure 7 and Figure 8.

As can be seen from Critic loss, Adam is more effective in distinguishing between real images and generated images. However, the data generated by the generator is more easily recognized by the discriminator as fake data. Also, in SGD, the resulting pictures has noticeable blurs.

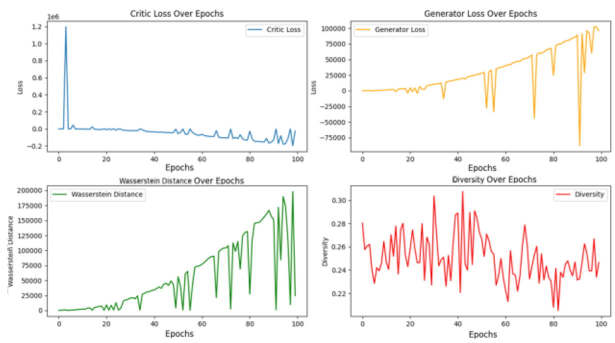

Figure 1. Performance using RMSprop optimizer (Figure Credits: Original).

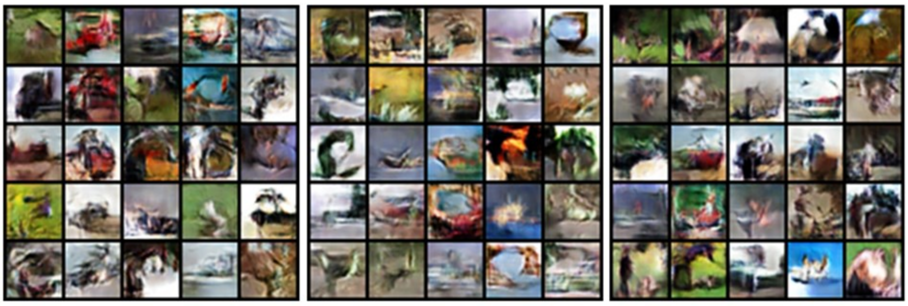

Figure 2. Representative generated images using RMSprop optimizer (Figure Credits: Original).

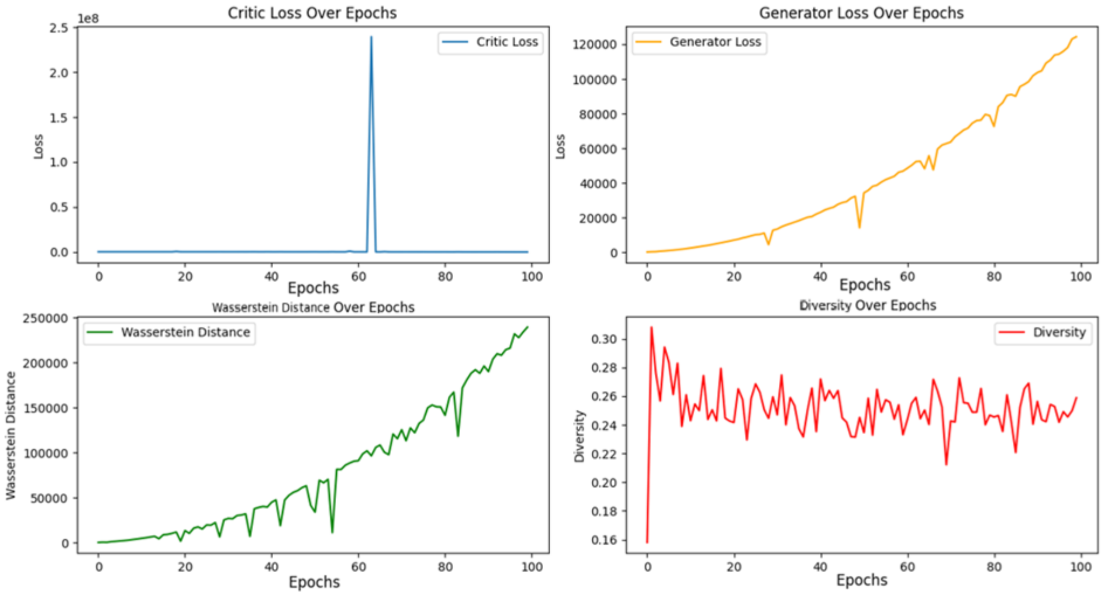

Figure 3. Performance using Adam optimizer (Figure Credits: Original).

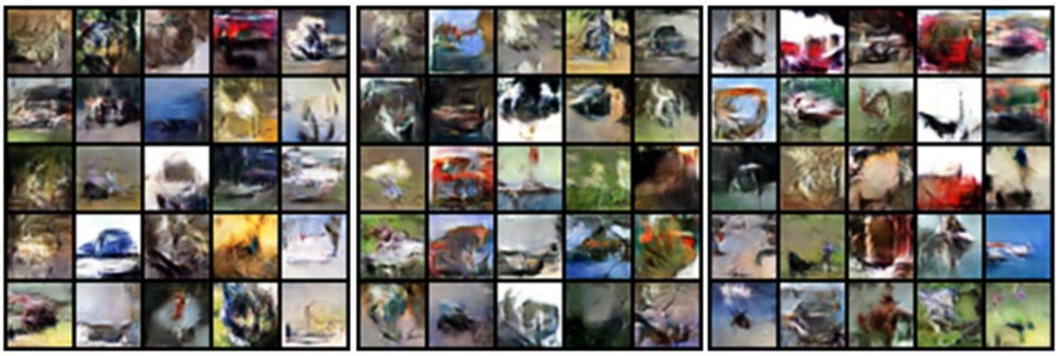

Figure 4. Representative generated images using Adam optimizer (Figure Credits: Original).

DOI: 10.54254/2755-2721/83/2024GLG0064

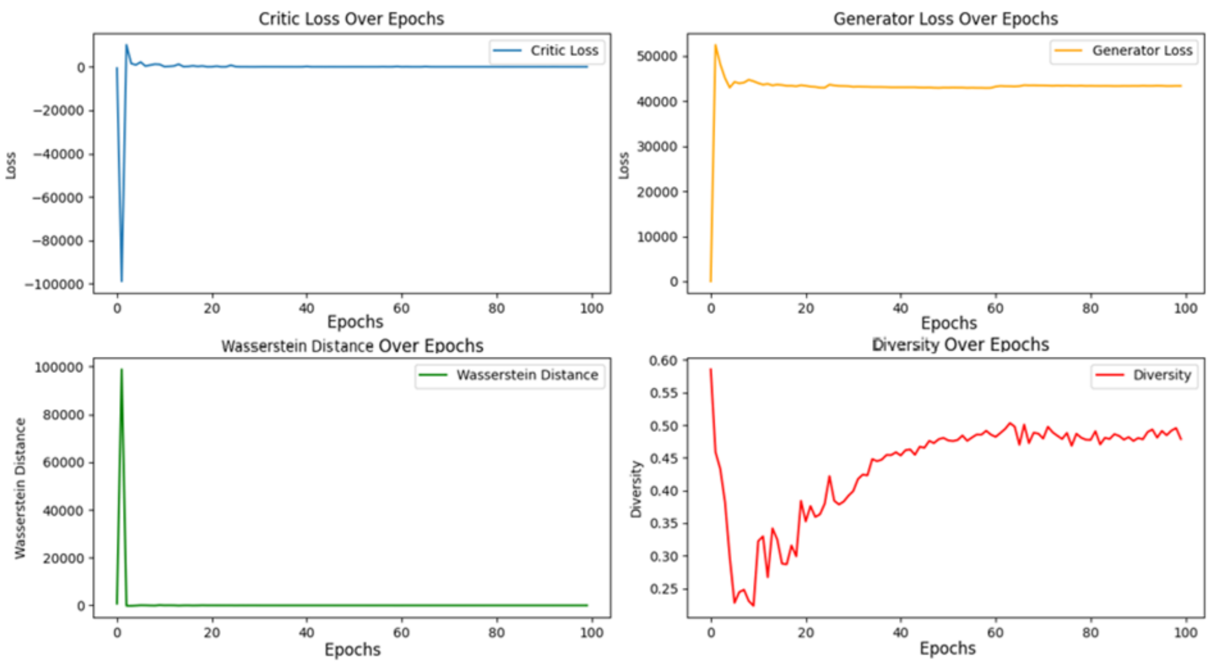

Figure 5. Performance using SGD optimizer (Figure Credits: Original).

Figure 6. Representative generated images using SGD optimizer (Figure Credits: Original).

Figure 7. Representative generated images using ResNet (Figure Credits: Original).

DOI: 10.54254/2755-2721/83/2024GLG0064

Figure 8. Representative generated images using DenseNet (Figure Credits: Original).

4. Discussion

In order to compare the influence of various changes on WGAN, this work uses four dimensions of 100epoch, including critic loss, generator loss, Wasserstein distance, diversity.

Among them, Critic Loss represents the model's judgment on sample generation and actual samples. The lower critic loss means the discriminator is more effective at telling apart real images from generated ones. Generator Loss reflects the difference between the generator's sample as well as the actual sample. The lower Generator Loss represents that the sample generated by the generator is closer to the real image. The Wasserstein Distance is the measurement of the separation between generator and discriminator from WGAN. The Diversity reflects the variety of generated samples. The higher the value, the greater the diversity of generated images. Moreover, the diversity is estimated by calculating the variance of the generated image in the feature space.

In this work, optimizers achieve various performance. For RMSprop, just like it said in the original WGAN, the effect of RMS is pretty standard. For Adam, the use of Adam brings some interesting changes compared to the original RMSprop. According to Critic Loss, RMSprop is not as good as Adam, which lets the discriminator do a better job of distinguishing between real and generated ones. When it comes to generator loss, the sample generated by RMSprop is closer to the real ones. Also, in terms of Wasserstein Distance, Adam performed less well than RMSprop. The samples generated by the generator showed higher diversity in the Adam case. For SGD, the effect of SGD does not seem to be that ideal. After a relatively big up-and-down, the critic loss, the generator loss and the Wasserstein distances all collapse, showing an abnormal level of stability. Various reasons may have contributed to this result. Perhaps SGD introduces large gradient noise by updating model parameters using only a small number of samples at a time, which may result in unstable updates.

As for models, ResNet works just fine, the output picture shows a good effect without collapse. Although the various values of ResNet fluctuated greatly, the final results were satisfactory. The ResNet can be used as the direction of improvement. DenseNet is doing just fine. Critic loss and generator loss are relatively stable. However, DenseNet's Wasserstein distance is large, which means that the pictures it produces are not as realistic as WGAN's. It could be seen from the generated picture that after 50 steps, the picture shows a relatively monotonous color and the effect becomes worse.

After trying a variety of modifications, it could be found that some modifications would not achieve the desired effect because of the incompatibility with the model and other reasons. In some cases, the phenomenon of rapid attenuation of learning rate and gradient explosion occurs, resulting in poor quality of the generated images. It is expected to continue to try to make modifications to the WGAN structure, adjust parameters, adopt new network structures or optimize the code so that the changes blend better with the original WGAN structure.

5. Conclusion

On the whole, this paper has achieved constructive results. Based on WGAN, some changes are made to see if they lead to better results. All experiments are based on LSUN in order to be consistent with

DOI: 10.54254/2755-2721/83/2024GLG0064

WGAN. This work used control variables and comparison methods. First, this work kept the network architecture unchanged and tried a variety of different optimizers. This paper tried the usual optimizers like Adam and SGD. Some of them have shown good results, such as Adam. However, some optimizers show less than satisfactory results, there have even been serious problems, such as gradient explosions. These results provide us with important references. And this work tried to change the network architecture, adding ResNet and DenseNet into the generator and discriminator. WGAN with ResNet has shown promising results, while the images generated by DenseNet are of poor quality. To conclude, experiments show the potential and possibilities of WGAN. These attempts gave us ideas and experience to improve WGAN. In the future, it is expected to expand the thinking further and try to improve WGAN even more. If possible, this work could extend WGAN to other generation tasks, such as text generation, and explore the effects of various components on the performance of WGAN.

References

[1] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. IEEE signal processing magazine, 35(1), 53-65.

[2] Gui, J., Sun, Z., Wen, Y., Tao, D., & Ye, J. (2021). A review on generative adversarial networks: Algorithms, theory, and applications. IEEE transactions on knowledge and data engineering, 35(4), 3313-3332.

[3] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., et, al. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139-144.

[4] Aggarwal, A., Mittal, M., & Battineni, G. (2021). Generative adversarial network: An overview of theory and applications. International Journal of Information Management Data Insights, 1(1), 100004.

[5] Durall, R., Chatzimichailidis, A., Labus, P., & Keuper, J. (2020). Combating mode collapse in gan training: An empirical analysis using hessian eigenvalues. arXiv preprint arXiv:2012.09673.

[6] Martin A., Soumith C., Léon B. (2017). Wasserstein GAN. arXiv preprint arXiv: 1701.07875.

[7] Rüschendorf, L. (1985). The Wasserstein distance and approximation theorems. Probability Theory and Related Fields, 70(1), 117-129.

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770-778.

[9] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 4700-4708.

[10] Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., & Xiao, J. (2015). Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365.

[11] Cifar-10 Dataset. URL: https://www.cs.toronto.edu/~kriz/cifar.html. Last Accessed:2024/08/23.