Loss Function Matters: Probing into the Reason behind Wasserstein GAN's Success

Implementation Track, 3 team members

Abstract

In this project, we implemented Wasserstein in Generative Adversarial Network (WGAN), as well as WGAN with gradient penalty (WGANGP). Wasserstein Generative Adversarial Networks (WGAN) was developed based on the Generative Adversarial Networks (GAN), which is a powerful generative model but still suffering from some defects. It modified the loss function of the original GAN, which made the training more stable and solved the mode collapse problem of the original GAN. The main breaking point of this network is using the Wasserstein Distance instead of KL and JS divergence. However, the weight clipping in WGAN could still cause gradient vanishing or exploding. Therefore, the gradient penalty was introduced to settle it. We regard WGAN as a landmark in the history of GAN, as it makes GAN's loss function much more interpretable. We carried out 4 parts of experiments: quantitatively evaluate the generation quality with Fréchet Inception Distance (FID) score, testing on training stability, evaluating generation diversity, and testing the performance of gradient penalty. The experiments were carried out on three datasets: LSUN Bedroom, CelebA, and Animefaces-Danbooru, where LSUN bedroom is the dataset used in the WGAN paper. Our experiment results verified the claim in WGAN and WGANGP paper. However, with testing the FID score, we thought the generation quality was something that can be improved further.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

1 Introduction

Generative model is an important aspect of machine learning because it aims to solve two significant problems: data generalization and data efficiency[Kingma and Dhariwal, 2018, Kingma and Welling, 2013]. Among different generative models, generative adversarial network (GAN) has been a popular research direction since it was first published in 2014[Goodfellow et al., 2014]. GAN models are currently widely used in image and audio synthesizing, text generation and image synthesizing[Kingma and Dhariwal, 2018, Kong et al., 2020, Zhu et al., 2017]. The separable structure of GAN makes it one of the lightweight models, ensuring its fast synthesizing speed.

However, when GAN was first raised, its training process was unstable due to its loss function[Arjovsky et al., 2017, Goodfellow et al., 2014]. Two main problems exist: (1) The loss function cannot correctly indicate its training process and is very unstable (2) The variety of generation objects are not guaranteed, which is also called mode collapse[Arjovsky et al., 2017, Goodfellow et al., 2014]. Later, these two problems were eased by Wasserstein GAN (WGAN) to a great extent. WGAN has thus become a base model for many later worksCao et al. [2019], Adler and Lunz [2018].

In this project, we will first explain in detail why Wasserstein GAN achieves improvements from the mathematical perspective. Then we will implement the Wasserstein GAN model on different datasets. To verify the effectiveness of our implementation, we will test our model on LSUN-Bedrooms dataset[Yu et al., 2015], which is the dataset used in the original paper. We will use LSUN Bedroom to justify some claims in the WGAN paper. To demonstrate our understanding, we will test our model on CelebA and Animefaces-Danbooru datasets, which is not included in the original paper[Viuts, 2021, Shuo, 2015]. These datasets feature capturing facial features and include a large number of elements, thus it's sufficient to manifest the advantages of Wasserstein GAN model. Both of the test baselines will be deep convolutional generative adversarial network (DCGAN), which is coherent with the original paper[Arjovsky et al., 2017].

The project is meaningful, as Wasserstein GAN is one of the landmarks that improves GAN interpretability. Though GAN is a well-elaborated model having marvelous performance, the mathematics behind it is hard to interpret[Goodfellow et al., 2014]. Unlike other generative models, GAN does not find a specific function that maps from a random variable to a given distribution[Kingma and Dhariwal, 2018], making it hard to rigorously prove why it works. Therefore, deriving a mathematically interpretable loss function can give a clearer picture of GAN model. It can also be a foundation for the following research.

Meanwhile, our project enriches the experiment of the original Wasserstein GAN paper by testing the model on CelebA and Animefaces-Danbooru datasets. Besides justifying the claims in WGAN paper, we carried out two additional experiments. we applied Fréchet Inception Distance (FID) score as a metric to evaluate the generation quality quantitatively, which was not provided by the original paper [Heusel et al., 2017]. FID is one of the most widely used evaluation metrics on GANs, and we believe that a quantitative metric can help understand performance differences between models. During our experiments, we also found that WGAN model failed on CelebA dataset. We solved the problem by applying the gradient penalty to the loss function.

2 Related Works

2.1 Generative Model

Generative model aims to find the conditional probability of a known distribution X from a random distribution Z, $P(X|Z = z)$[Jebara, 2004]. In the view of maximum likelihood, it can be expressed as: Given a known distribution $P_r(x)$ of real data and a generated distribution $P_g(x;\theta)$ ($P_g(x;\theta)$ also contains z), we'd like to find a optimal θ that make the generated distribution be as close as possible to the real distribution.

To transform the random latent variable z into generated distribution, we need to calculate the conditional probability in the following equation:

$$P_g(x;\theta) = \int_z P(z)P_\theta(x|z)dz \tag{1}$$

For simplicity, $P_g(x)$ will be used instead of $P_g(x;\theta)$ in the latter parts.

A challenge lies in that achieving an exact expression conditional probability needs a huge amount of calculating, making it quite time-consuming.

2.2 Generative Adversarial Network

In 2014, the first generative adversarial network(GAN) model was raised[Goodfellow et al., 2014]. To tackle the challenges of the generative model, GAN introduces a neural network discriminator to evaluate the quality of generated samples besides the generator. The discriminator takes a sample as input and outputs a value indicating whether it is fake or not. On one hand, the goal of the discriminator is to classify real samples as real and generated samples as fake. On the other hand, the goal of the generator is to deceive the discriminator i.e. trying to make generated samples classified as real. The training objective function of the discriminator can be defined as follow:

$$L\left(d,g\right) = \mathbb{E}_{x\sim\mathbb{P}_r}[\log d(x)] + \mathbb{E}_{x\sim\mathbb{P}_g}[\log(1-d(x))] \tag{2}$$

where d is the discriminator, and P_g is the distribution created by the generator. And the optimal generator is to solve:

$$g^* = \arg\min_g \max_d L(d,g) \tag{3}$$

In practice, generator and discriminator are alternately trained to obtain the best generator. Both generator and discriminator apply neural networks with fully connected layers and activation layers. Thus, the generator is solved by backpropagation of two neural networks, making a black box to approach $P_\theta(x|z)$. Due to the great performance of neural networks in finding the global minimum, GAN model can have high-quality generation. Meanwhile, as only the generator is needed in the synthesizing stage, the synthesizing speed can be very fast by making a small generator and a large discriminator to ensure its quality.

2.3 Deep convolutional Generative Adversarial Network

Deep convolutional generative adversarial network (DCGAN) is used as the baseline of our project. It changed the fully connected layer in the simple GAN model generator into convolutional layers. It enhances its performance on 2D image synthesizing. The structure of DCGAN is in Figure 1

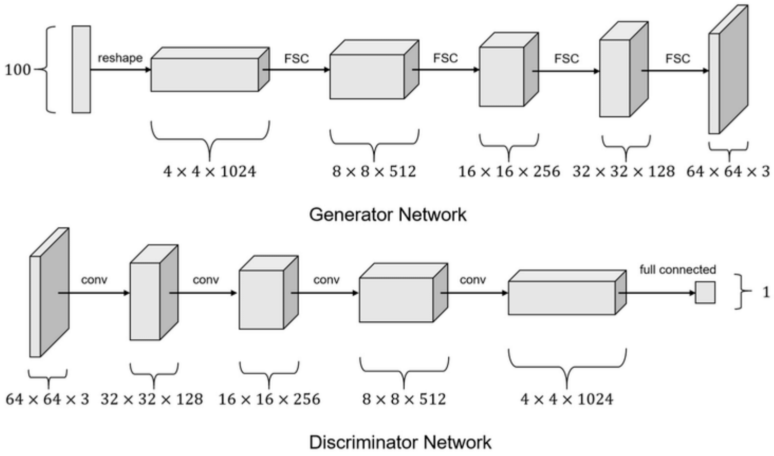

Figure 1: The structure of DCGAN. conv in discriminator is a 2D convolutional layer, and FSC in generator is 2D deconvolutional layer (which acts reversely as convolutional layer).

2.4 Kullback-Leibler and Jensen-Shannon Divergence

Kullback-Leibler divergence (KLD) is defined as:[Menéndez et al., 1997]

$$KL\left(P_1 \| P_2\right) = \mathbb{E}_{x \sim P_1} \log \frac{P_1}{P_2}$$

Jensen-Shannon Divergence (JSD) is defined as:[Menéndez et al., 1997]

$$JS\left(P_1 \| P_2\right) = \frac{1}{2} KL\left(P_1 \| \frac{P_1 + P_2}{2}\right) + \frac{1}{2} KL\left(P_2 \| \frac{P_1 + P_2}{2}\right)$$

Both of them are measurements of the similarity between P_1 and P_2.

We can change the loss function of GAN into this form:

$$\mathbb{E}_{x \sim P_r} \log \frac{P_r(x)}{\frac{1}{2}\left[P_r(x) + P_g(x)\right]} + \mathbb{E}_{x \sim P_g} \log \frac{P_g(x)}{\frac{1}{2}\left[P_r(x) + P_g(x)\right]} - 2 \log 2$$

Plugging in JSD, we have the simple GAN and DCGAN discriminator loss function as follow:

$$L\left(d, g, r\right) = 2JS\left(P_r(x) \| P_g(x)\right) - 2 \log 2$$

2.5 Fréchet Inception Distance

Fréchet Inception Distance (FID) is the evaluation metric we'll be using to quantitatively measure model performance. FID model sends both real images and generated images into a deep convolutional classifier, extracting and comparing their feature map (output of hidden layer with the following equation):

$$\text{FID} = \left\|\mu_g - \mu_r\right\|_2^2 + tr\left(\Sigma_g + \Sigma_r - 2\left(\Sigma_g \Sigma_r\right)^{1/2}\right)$$

where μ is the expectation and Σ is the covariance matrix.

3 Method

3.1 Problem of GAN Loss Function

For the loss function $L\left(D, g, r\right)$, when $P_g(x) = 0, P_r(x) \neq 0$ or $P_g(x) \neq 0, P_r(x) = 0$, $L\left(D, g, r\right) = \log 2$, which is a constant, causing the gradient to vanish, which means if P_r and P_g have no intersection, the loss will be a constant. Meanwhile, as GAN usually models a low dimension distribution (Gaussian variable) to a high dimension distribution (real images), it's likely that there's no intersection between P_r and P_g. And it will make gradient vanish, causing an **unstable training**.

Meanwhile, when $P - g \to 0$ and $P - r \to 1$, $P_g(x) \log \frac{P_g(x)}{P_r(x)} \to 0$, contributing little to the KLD and thus the GAN loss function. But when $P - g \to 1$ and $P - r \to 0$, $P_g(x) \log \frac{P_g(x)}{P_r(x)} \to \infty$, contributes much more to the loss function. This means if the generator tries to create safe samples that approaches the known distribution, it doesn't matter if it creates fake samples. However, if it tries to create samples that are not included in the known distribution, it will receive a severe penalty. These two situations should be treated as the same, and it could lead to lack of generation diversity, which is also called **mode collapse** problem

3.2 Wasserstein Loss

Wasserstein distance is defined as follow:

$$W\left(P_r, P_g\right) = \inf_{\gamma \sim \Pi(P_r, P_g)} \mathbb{E}_{(x,y) \sim \gamma}[\|x - y\|]$$

where $\Pi\left(P_r, P_g\right)$ is the union distribution of P_r and P_g. This distance outperforms JSD, as it can reveal the actual distance between P_r and P_g even if there is no intersection.

However, $\inf_{\gamma\sim\Pi\left(P_r,P_g\right)}$ cannot be directly calculated. Thus, the author offered an alternative expression:[Arjovsky et al., 2017]

$$W\left(P_r, P_g\right) = \frac{1}{K} \sup_{\|f\|_L \leq K} \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{x\sim P_g}[f(x)]$$

where f is Lipschitz continuous, and K is the Lipschitz constant (for detailed proof, refer to the WGAN paper[Arjovsky et al., 2017]).

As we are dealing with discrete values, the equation can be changed into

$$K \cdot W\left(P_r, P_g\right) \approx \max_{w:|f_w|_L \leq K} \mathbb{E}_{x\sim P_r}\left[f_w(x)\right] - \mathbb{E}_{x\sim P_g}\left[f_w(x)\right]$$

To obtain the expression of f_w, we can apply the method of deep learning, where f is a learnable parameter. It is exactly the discriminator in GAN models, and $K \cdot W\left(P_r, P_g\right)$ is the Wasserstein loss. Thus, we can obtain the loss of WGAN discriminator:

$$L_{WGAN:d}\left(P_r, P_g, d\right) = \mathbb{E}_{x\sim P_r}\left[d(x)\right] - \mathbb{E}_{x\sim P_g}\left[d(x)\right]$$

For the generator loss, instead of the rigorous minimax game of the original gan, it's simply defined as part of reversed discriminator loss, which still applies the idea of the minimax game.

$$L_{WGAN:g}\left(P_r, d\right) = -\mathbb{E}_{x\sim P_r}\left[d(x)\right]$$

The Wasserstein loss **makes sense when** $P_g(x) = 0$ **or** $P_r(x) = 0$, as neither the value of $P_g(x) = 0$ nor $P_r(x)$ itself will contribute on the loss function, but their difference $P_g(x) - P_r(x)$ will make the contribution. Therefore, it doesn't matter if $P_g(x) = 0$ or $P_r(x) = 0$, and both the unstable training and mode collapse problems settled.

Practically, WGAN made these modifications from DCGAN:

- Substituting original GAN loss with Wasserstein loss
- Remove the final activation layer with Tanh
- Limiting the norm of discriminator weight less than Lipschitz constant.

3.3 Wasserstein Generative Network with Gradient Penalty

To limit the norm of discriminator weight, the original WGAN applies weight clipping. However, this can easily lead to gradient vanishing or explosion, as it just cut the weight at a certain line with brute force, and this weight can easily make the discriminator undifferentiable and needs to be carefully chosen. To solve this problem, a new method (WGANGP) is raised that instead of doing clipping, a penalty term is added to the discriminator loss[Gulrajani et al., 2017]. Its objective is to make the norm of discriminator output gradient close to 1, at which time the weight of discriminator is Lipschitz continuous with constant 1 (proof is detailed in the original paper).

$$L_{WGANGP:d}\left(P_r, P_g, d\right) = \mathbb{E}_{x\sim P_r}\left[d(x)\right] - \mathbb{E}_{x\sim P_g}\left[d(x)\right] + \lambda\mathbb{E}_{x\sim P_r,P_g}\left[\left(\left\|\nabla_x D(x)\right\|_2 - 1\right)^2\right]$$

λ is set 10 in the original paper. For the gradient penalty, samples from both generated distribution and known distribution are calculated separately. We applied this method to Animefaces-Danbooru and CelebA datasets to test generation quality.

4 Experiments

Our experiments are divided into four parts: generation quality, a test of training stability, a test of mode collapse problem, and performance of gradient penalty. The structure of convolutional layers is the same in all three models, and all of the models share the same generator. For WGAN model, the clipping range of discriminator weight is set [-0.01, 0.01].

4.1 Generation Quality

We tested the generation quality with three datasets: LSUN Bedroom, CelebA, and Animefaces-Danbooru. LSUN bedroom is the dataset in the original WGAN paper, which is a collection of natural images of indoor bedrooms. CelebA is a large human face dataset, and Animefaces-danbooru is an Animefaces-danbooru face dataset[Viuts, 2021, Shuo, 2015, Yu et al., 2015]. All images are generated with 64*64 in size and 3 in channels. Generation quality is quantitatively calculated with FID score, and results are in Table 1. Our baseline model is DCGAN. Samples of CelebA and Animefaces-Danbooru are in figure 2 - 6(more samples are available in the following parts).All samples are randomly chosen without selection to ensure authenticity. No data are provided for WGAN on CelebA is due to the gradient vanishing, which will be discussed in Section 4.4.

FID Score	LSUN Bedroom	CelebA	Animefaces-danbooru
DCGAN	31.2	46.5	43.3
WGAN	29.4		50.2
WGANGP	26.6	42.1	44.0
Ground Truth	1.15	2.56	0.97

Table 1: FID Score of models

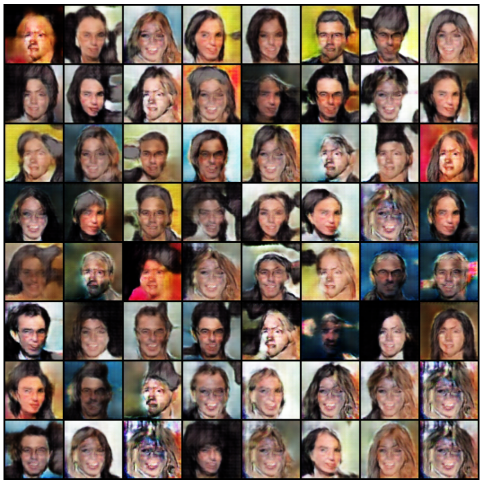

Figure 2: DCGAN on CelebA.

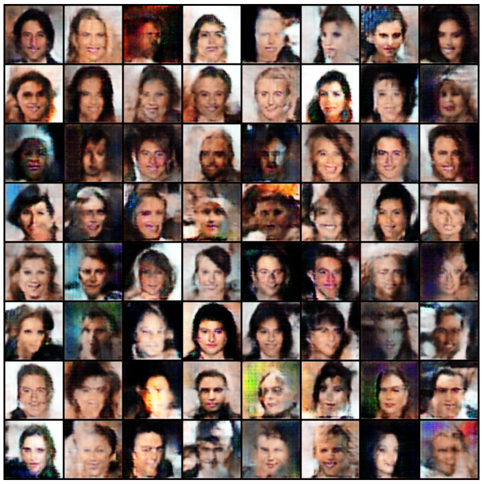

Figure 3: WGAN on CelebA.

Among the three models, WGANGP is slightly better than the other two models, but many of the generated images are still far from reality.

4.2 Training Stability

The stability of training is demonstrated by the discriminator loss function. The dataset we use as an example is LSUN Bedroom, which is the dataset in original WGAN paper[Arjovsky et al., 2017]. We think it can better justify the claim in the paper. See Figure 7, 8.

The loss function decent of WGAN is much more robust than that of DCGAN. Thus it's less likely for WGAN to have an overfitting problem. Meanwhile, it is much easier to find if the loss of WGAN comes to a convergence, which is hard to tell from DCGAN loss. Therefore, we can conclude that the loss of WGAN ensures a more stable training.

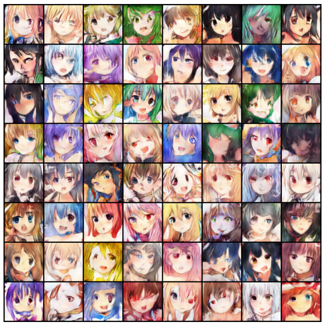

Figure 4: DCGAN on Animefaces-danbooru.

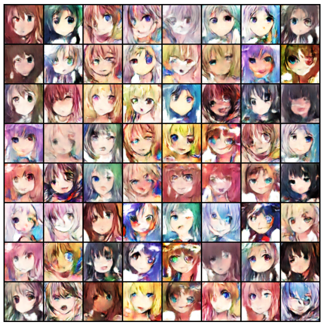

Figure 5: WGAN on Animefaces-danbooru.

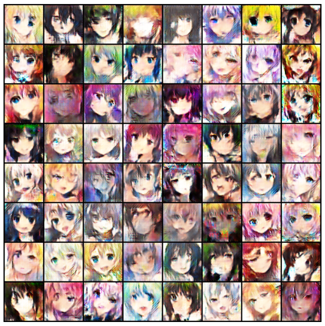

Figure 6: WGANGP on Animefaces-danbooru.

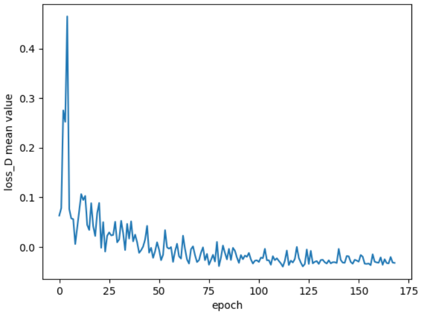

Figure 7: DCGAN discriminator loss.

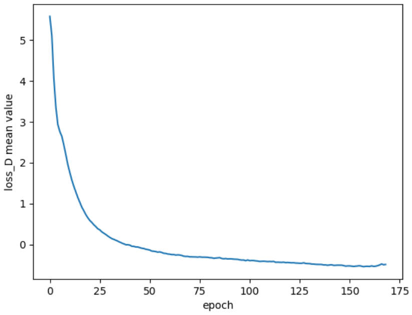

Figure 8: WGAN discriminator loss.

4.3 Mode Collapse

The mode collapse problem of DCGAN can be easily found in the generated samples of LSUN bedroom in Figure 10, where a lot of images look very similar to each other. While in bedrooms generated by WGAN, the problem doesn't exist. Therefore, we can say that the mode collapse problem has been greatly eased by WGAN.

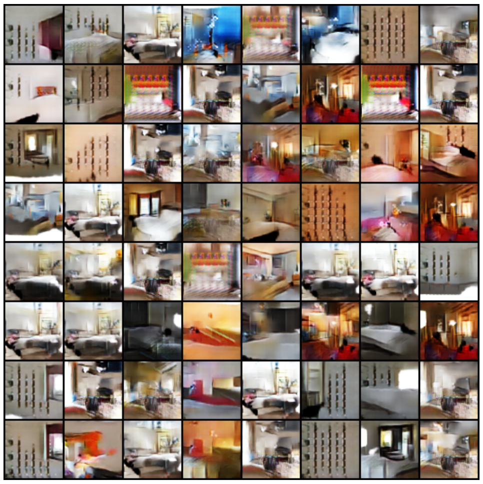

Figure 9: DCGAN on LSUN Bedroom.

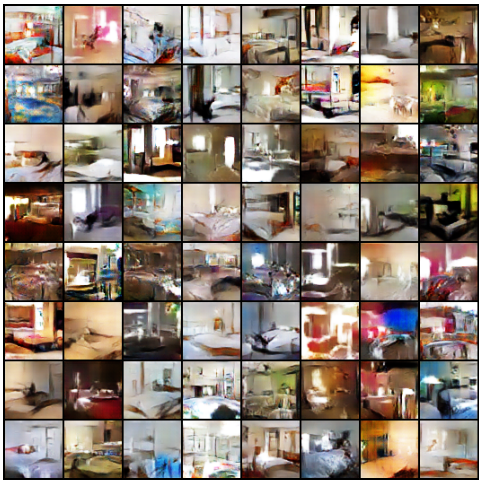

Figure 10: WGAN on LSUN Bedroom.

4.4 Performance of Gradient Penalty

As described in Section 3.3, an inappropriate clipping range can cause gradient vanishing and explosion. We tested WGAN with [-0.01, 0.01] as the clipping range on CelebA, whose loss had converged but wasn't able to generate satisfactory images. However, when we applied the gradient penalty instead, the problem was solved. See Figure 11, 12. Mode collapse problem of GAN also exists in the other datasets, please refer to the samples above.

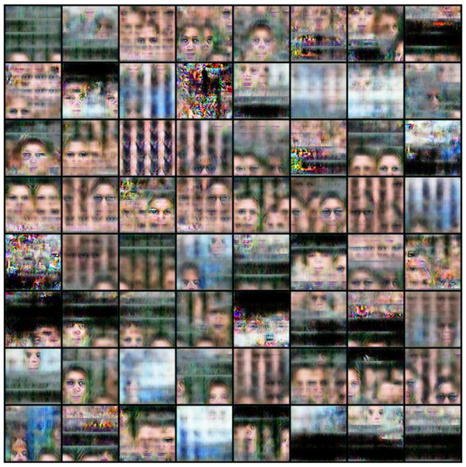

Figure 11: DCGAN (without gradient penalty) on CelebA.

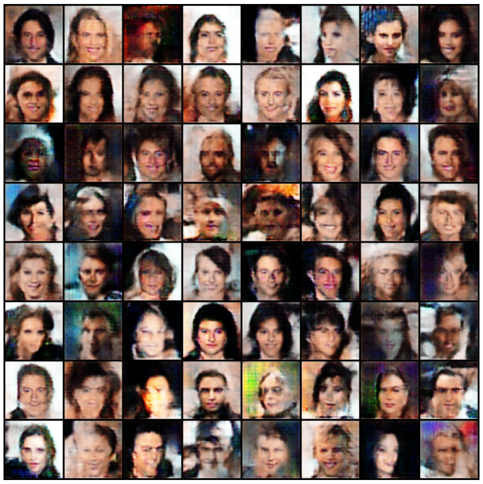

Figure 12: DCGANGP (with gradient penalty) on CelebA.

5 Conclusion

This is a meaningful project to hand-on implement the Wasserstein GAN model. We justified that the model does make great improvements in solving the unstable training and mode collapse problem of the original GAN model because of its unique loss function. We not only tested it with experiments but also learned the mathematical motivation behind the adjusting of the loss function. And motivation is much more important when coming up with a new research idea. With rigorous mathematical proof, we can know that we are on the right track. Meanwhile, when we are doing experiments, we found that the WGAN model failed on CelebA dataset, which prompted us to find the reason behind it, and we solved it by adding a gradient penalty term to the loss. We also learned from this episode that when solving a problem, we should dig deep to find its derivation, which is the clipping weights. Actually, it's not illustrated in the WGAN paper, but can be only found in the code. Carefully reading codes is also an important ability that we learned, as the author always puts positive results and high-level motivation in the paper. But it may have defects, which can be only found in the code and with experiments. Though the WGAN model takes a great stride in the interpretability of GAN, it still cannot generate very high-resolution samples. Thus, we should say that we have implemented a classic paper, but there are still many following up works for us to navigate through. This is a paper with great motivation and significance, but it's not the model with the best performance. But we still regard this project as meaningful as we learned a lot about how to carry out research other than just put previous works together and tune the parameters.

6 Work Distribution

Author 1: proposed the topic of the project, implemented the backbone code of WGAN and DCGAN, wrote the theoretical parts in the proposal and final report.

Author 2: Ran experiments on WGAN and DCGAN, found the clipping problem of WGAN, wrote the experiment part in the final report.

Author 3: solved the clipping problem of WGAN by implementing gradient penalty. Ran experiments on WGANGP, refined the format of the final report.

References

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems*, volume 33, pages 17022–17033, 2020. URL https://proceedings.neurips.cc/paper/2020/file/c5d736809766d46260d816d8dbc9eb44-Paper.pdf.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. doi: 10.1109/ICCV.2017.244.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223, 2017. URL https://proceedings.mlr.press/v70/arjovsky17a.html.

Jiezhang Cao, Langyuan Mo, Yifan Zhang, Kui Jia, Chunhua Shen, and Mingkui Tan. Multi-marginal wasserstein gan. *Advances in Neural Information Processing Systems*, 32:1776–1786, 2019.

Jonas Adler and Sebastian Lunz. Banach wasserstein gan. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/file/91d0dbfd38d950cb716c4dd26c5da08a-Paper.pdf.

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015. URL http://arxiv.org/abs/1506.03365.

Viuts. Animefaces-danbooru, version 1. *Kaggle*, 2021. URL https://www.kaggle.com/lukexng/animefaces-512x512/version/1.

Yang et al. Shuo. From facial parts responses to face detection: A deep learning approach. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

T. Jebara. *Machine Learning: Discriminative and Generative*. Kluwer Academic Publishers, 2004.

M.L. Menéndez, J.A. Pardo, L. Pardo, and M.C. Pardo. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997. doi: https://doi.org/10.1016/S0016-0032(96)00063-4.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

7 Appendix

7.1 Random seed for reproduction

LSUN bedroom: 7840
Animefaces-danbooru: 5635
CelebA: 3303