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

Implementation Track, 3 team members

1 Introduction

Generative model is an important aspect of machine learning because it aims to solve two significant problems: data generalization and data efficiency(7). Among different generative models, generative adversarial network (GAN) has been a popular research direction since it was first published in 2014(4). GAN models are currently widely used in image and audio synthesizing, text generation and image synthesizing(7; 8; 13). 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(4; 2). Two main problems exist: (1) The loss function cannot correctly indicate its training process (See figure 1) (2) The variety of generation objects are not guaranteed, which is also called mode collapse(2; 4). Later, these two problems were eased by Wassertein GAN to a great extent.

In this project, we will first explain in detail why Wasserstein GAN achive 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(12). To demonstrate our understanding, we will test our model on CelebA and Anime Avatars datasets, which is not included by the original paper. 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 DCGAN, which is coherent with the original paper(2).

The project is meaningful, as Wassertein GAN is one of the landmarks that improves GAN interpretability. Though GAN is a well elaborate model having marvellous performance, the mathematics behind it is hard to interpret(4). Unlike other generative models, GAN does not find a specific function which maps from a random variable to a given distribution(7), 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 mdoel on CelebA and Anime Avatars datasets. Besides, we will use Fréchet Inception Distance (FID) as a metric to evaluate the model performance, which was not provided by the original paper. We believe that a quantitative metric can help understand performance difference between models.

2 Related Works

2.1 Generative model

Generative model aims to find the conditional probability of a known distribution X from a random distribution Y, $P(X|Y = y)$(11). In the image generation task, the model can be interpreted as: We want to construct a function that maps noise (random variable) to a recognizable image (known distribution). To achieve this, the machine learning method applies data to train a model. It has a loss function to penalize the generation quality by comparing the generated distribution with original known distribution.

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

2.2 Generative adversarial network

Generative adversarial network (GAN) is a generative mdoel that introduces two neural networks: generator and discriminator(4). The generator is trained to generate images that can deceive the discriminator. The training obejective function is given as follow:

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

where $\mathbb{P}_r$ represents the known distribution, and $\mathbb{P}_\theta$ represents tha random distribution.

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

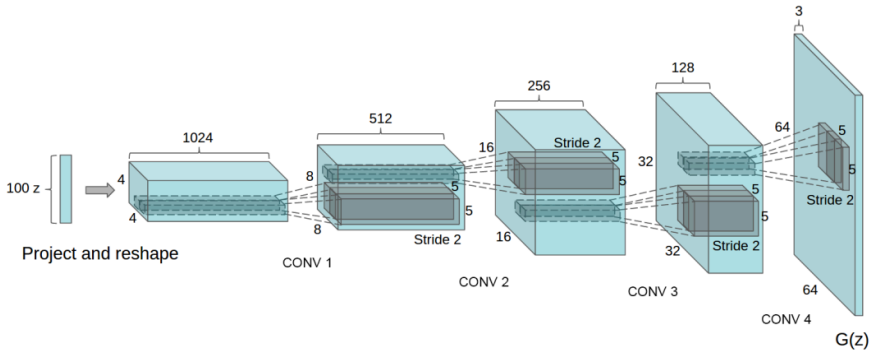

Figure 1: cited from DCGAN paper(10). The structure of DCGAN generator, featuring a deep convolutional network with leaky reLu and tanh activation.

2.4 Kullback-Leibler and Jensen-Shannon Divergence

Kullback-Leibler divergence (KLD) is defined as:(9)

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

Jensen-Shannon Divergence (JSD) is defined as:(9)

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

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

We can change the loss funciton 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 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 image and generated image into a deep convolutional

classifier, extracting and comparing their feature map (output of hidden layer with the following equation):

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

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

3 Method

3.1 Problem of GAN loss function

for the loss function $L(D, g, r)$, when $P_g(x) = 0, P_r(x) \neq 0$ or $P_g(x) \neq 0, P_r(x) = 0$, $L(D, g, r) = \log 2$, 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.

3.2 Wasserstein loss

Wasserstein distance is defined as follow:

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

where $\Pi(P_r, P_g)$ 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, $\inf_{\gamma\sim\Pi(P_r,P_g)}$ cannot be directly calculated. Thus, the autor offered an alternative expression:(2)

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

where f is Lipschitz continuous, and K is the Lipschitz constant (for detailed proof, refer to the WGAN paper(2)).

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

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

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 $K \cdot W(P_r, P_g)$ is the Wasserstein loss.

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.

4 Experiments

4.1 Experiment Process

In the first part of the project, a general background of the WGAN method will be provided. We will analyze the main advantages of WGAN over traditional GAN in the aspects of reducing training difficulty and avoiding mode collapse.

The body of the project will focus on verifying the effectiveness of the WGAN model. We would first examine our model on the LSUN-Bedrooms dataset. Moreover, we will extends the model on various new datasets and test the accuracy of our results. In the process, we will mainly use PyTorch to train the datasets and apply our own methods to conclude the applications and advantages of WGAN.

4.2 Dataset

The datasets we chose to use in the project includes LSUN-Bedrooms, CelebA and Anime Avatars, which can be obtained from public repositories on Github and other public resources. All these datasets contain a large enough number of elements, so that the experiment will be effective in distinguishing the outcomes of WGAN with traditional DCGAN methods. Besides, CelebA and Anime Avatars share similarities in face recognition. By evaluating and comparing the results for these datasets, we could figure out the effectiveness of WGAN on both real-world and simulated data.

4.3 Evaluation

We will use DCGAN as our baseline, and FID as our measurement metric. Experiments will be performed on LSUN-Bedrooms, CelebA and Anime Avatars. We will also perform the same preprocessing and set a certain random seed for fairness and reproducibility.

5 Project Plan

5.1 Work Division

- Author1 will write the code for WGAN and run a toy model on it for testing.
- Author2 will train WGAN and DCGAN with LSUN-Bedrooms, CelebA and Anime Avatars.
- Author3 will run FID on the result and compare the loss and the generation quality with the WGAN paper.

5.2 Experience

1. As we are doing a deep learning project, we will start early, especially downloading the huge datastes.

2. To make training faster, we may apply data parallel and use multiple GPUs.

3. We will be running DCGAN in the same time as running WGAN to save time.

5.3 Challenge

In this project, we will run on 3 models: WGAN, DCGAN and FID. Though we will only implement WGAN, but problems may arise when utilizing existing codes. Therefore, we should have a thorough understanding of all three models, which may be time-consuming.

5.4 Milestone

Write WGAN code (due 12, Nov) $\rightarrow$ Train WGAN and DCGAN (due 21, Nov) $\rightarrow$ Run FID and analysis result (due 28, Nov) $\rightarrow$ Write report (due 10, Dec)

5.5 Contingency plans

If time is urgent, we will give up the reproducibility and set cuda deterministic to accelerate training. If time is still not enough, we will not test our results with FID. Instead, we will just offer several image samples and give a subjective one by blinding scoring.

References

[1] Borji, Ali. "Pros and cons of gan evaluation measures." Computer Vision and Image Understanding 179 (2019): 41-65.

[2] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. International Conference on Machine Learning, ICML, 2017.

[3] Jiezhang Cao, Langyuan Mo, Yifan Zhang, Kui Jia, Chunhua Shen, and Mingkui Tan. Multi-marginal wasserstein gan. In Advances in Neural Information Processing Systems (NIPS), 2019.

[4] 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 (NIPS), 2014.

[5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Gunter Klambauer, and Sepp Í Hochreiter. GANs trained by a two time-scale update rule converge to a nash equilibrium. arXiv preprint arXiv:1706.08500, 2017.

[6] Adler Jonas and Lunz Sebastian. Banach Wasserstein GAN. Advances in neural information processing systems (NIPS), 2014.

[7] Durk P Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. Advances in neural information processing systems (NIPS), 2018.

[8] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. Advances in Neural Information Processing Systems, 2020.

[9] Menéndez, M. L., et al. "The jensen-shannon divergence." Journal of the Franklin Institute 334.2 (1997): 307-318.

[10] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. International Conference on Learning Representations (ICLR), 2016.

[11] Jebara, Tony. Machine Learning: Discriminative and Generative. The Springer International Series in Engineering and Computer Science. Kluwer Academic (Springer). ISBN 978-1-4020-7647-3, 2004.

[12] 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.

[13] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. CoRR, abs/1703.10593, 2017.