Discriminator Variance Regularization for Wasserstein GAN

Jeongik Cho

jeongik.jo.01@gmail.com

Abstract

In Wasserstein GAN, it is important to regularize the discriminator to have a not big Lipschitz constant. In this paper, I introduce discriminator variance regularization to regularize the discriminator of Wasserstein GAN to have a small Lipschitz constant. Discriminator variance regularization simply regularizes the variance of the discriminator's output to be small when input is real data distribution or generated data distribution. Intuitively, a low variance of discriminator output implies that the discriminator is more likely to have a low Lipschitz constant. Discriminator variance regularization does not explicitly regularize the Lipschitz constant of discriminator through differentiation on discriminator but lowers the probability that the Lipschitz constant of the discriminator is high. Discriminator variance regularization is used in Wasserstein GAN with R1 regularization, which reduces the vibration of GAN. Discriminator variance regularization requires very little additional computing.

1. Introduction

In Wasserstein GAN (WGAN) [1], it is important to regularize the discriminator to have a small Lipschitz constant. Several gradient penalty methods [2, 3, 4] were proposed to regularize the Lipschitz constant of the discriminator through differentiation on the discriminator.

In this paper, I introduce discriminator variance regularization (DV regularization) for WGAN to regularize the discriminator to have a small Lipschitz constant. Discriminator variance regularization simply regularizes the variance of the discriminator's output to be low when input is real data distribution or generated data distribution. DV regularization does not explicitly regularize the Lipschitz constant of discriminator through differentiation on discriminator but lowers the probability that the Lipschitz constant of the discriminator is high. Also, DV regularization is used together with R1 regularization [8] to prevent vibration of GAN. DV regularization requires very little additional computing.

2. Discriminator variance regularization

Assuming discriminator input and output are closed sets, the discriminator of WGAN can be considered a Lipschitz-continuous function. The problem is that the Lipschitz constant of the discriminator may be very large. The large Lipschitz constant of the discriminator causes the gradient to explode and prevents the WGAN from being trained. When training WGAN without regularization terms such as weight clipping or gradient penalty, discriminator output distribution for real data distribution or generated data distribution has an extremely large variance, and the WGAN is hard to be trained.

Intuitively, extremely high discriminator output variance indicates that the discriminator has a large Lipschitz constant. On the other hand, intuitively, if the variance of discriminator output is low, the discriminator would have a small Lipschitz constant. For example, if the discriminator output is constant (variance is zero), the Lipschitz constant of the discriminator is zero. More specifically, the low variance of discriminator output distribution indicates a low probability that the Lipschitz constant of the discriminator is high. Therefore, DV regularization that regularizes variance of discriminator output lowers the probability that the Lipschitz constant of the discriminator is high. In fact, other gradient penalty methods that explicitly regularize the Lipschitz constant of the discriminator are also basically probabilistic methods because training the model is based on Monte Carlo simulation. Therefore, lowering the probability that the Lipschitz constant of the discriminator is high is not illogical.

DV regularization regularizes variance of discriminator output when input is real data distribution or generated data distribution. DV regularization is defined as follows.

$$L_{dv} = var\big(D(X)\big) + var\Big(D\big(G(Z)\big)\Big)$$

DV regularization uses batch distribution to approximate the variance of adversarial values. DV regularization loss for each batch is defined as follows.

$$L_{dv} = sum((a_r - \overline{a_r})^{\circ 2}) + sum\Big(\big(a_g - \overline{a_g}\big)^{\circ 2}\Big)$$

The following table explains the terms used in the above equations.

b	Batch size
X	Real data random variable
x	b real data
Z	Latent random variable
z	b latent codes
D	Discriminator
G	Generator
a_r	Adversarial values of real data (i.e., $D(x)$). b-dimensional vector.
a_g	Adversarial values of generated data (i.e., $D\big(G(z)\big)$). b-dimensional vector
$\overline{a_r}$	Mean of a_r
$\overline{a_g}$	Mean of a_g
$vec^{\circ 2}$	Element-wise square of example vector vec
sum	A function that calculates the sum of the input vector
var	A function that calculates the variance of a random variable
L_{dv}	DV regularization loss
λ_{dv}	DV regularization loss weight

Simply, DV regularization loss is a sum of the variance of real adversarial values a_r and fake adversarial values a_g. DV regularization multiplied by λ_{dv} and added to discriminator loss.

R1 regularization on the discriminator makes the training stable when the WGAN almost converges. However, R1 regularization alone does not make the discriminator satisfy the Lipschitz condition, so the WGAN is hard to be trained. Therefore, R1 regularization should be used together with an additional regularization term to make the GAN converges stably.

3. Experiment results

I used StyleGAN2 [5] architecture with a reduced filter size of convolution for the experiment. The model was trained to generate the CelebA dataset [7] resized to 128×128 resolution. Following hyperparameters were used for model training.

$$optimizer = Adam\begin{pmatrix} learning\ rate = 0.001 \\ beta_1 = 0.0 \\ beta_2 = 0.99 \end{pmatrix}$$

$$learning\ rate\ decay\ per\ epoch = 2\%$$

$$latent\ vector\ dimension = 512$$

$$batch\ size = b = 32$$

$$epoch = 50$$

Note that the mapper of the generator was trained with $\times 0.01$ learning rate, same as StyleGAN2. The following figure shows the performance of various regularization methods in WGAN. FID [6] was used for model performance evaluation. The lower the FID, the better generative performance.

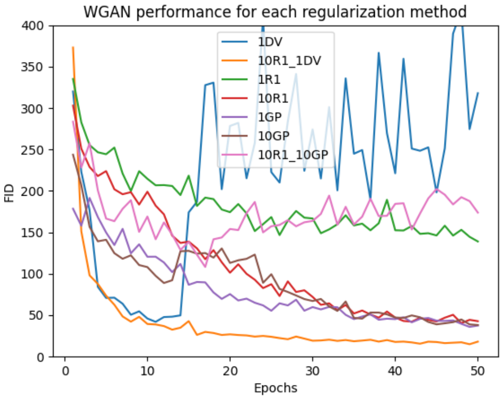

Figure 1. WGAN performance for each regularization method

In Fig. 1, "GP" represents the gradient penalty method of [2]. "R1" represents the R1 regularization method of [8]. The number in front of each method represents the loss weight. "DV" represents DV regularization. Two losses connected by an underbar mean that the two losses are added. For example, "10R1_1DV" represents that $10 \times L_{r1} + 1 \times L_{dv}$ was used for WGAN discriminator regularization. The original paper introduced R1 regularization [8] used $\gamma/2$ for R1 regularization weight, so based on the original paper, γ is $\lambda_{r1} \times 2$.

In Fig. 1, one can see the performance of DV regularization with R1 regularization is the best. Also, in the DV regularization without R1 regularization (1DV in Fig. 1), mode collapse occurred at epoch 14.

The following figure shows generated image with R1 regularization and DV regularization.

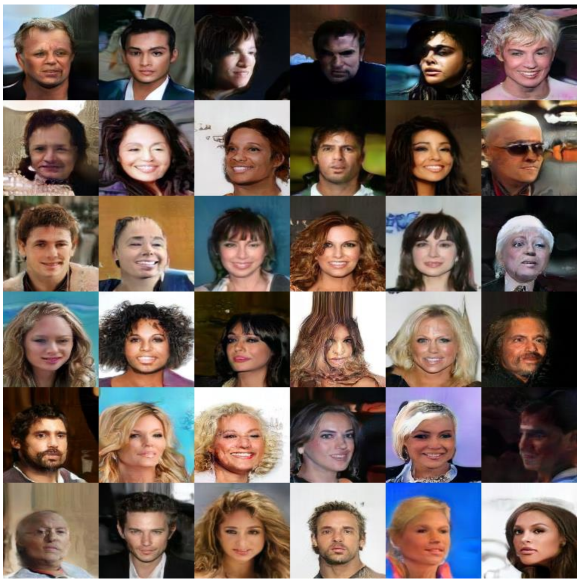

Figure 2. Generated images with 10R1_1DV

I also trained the 10R1_1DV model with larger filter sizes. The FID of the model was 12.676382. The following figure shows the example of generated images with the thicker model.

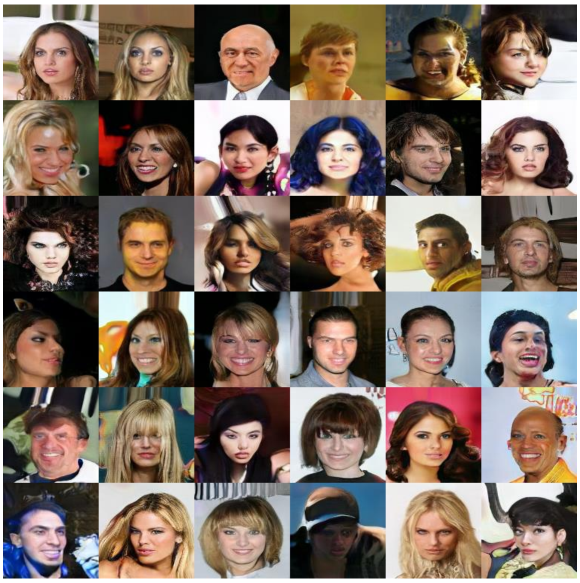

Figure 3. Generated images with 10R1_1DV, thicker model

4. Conclusion

In this paper, I introduced discriminator variance regularization for Wasserstein GAN to regularize the discriminator of Wasserstein GAN to have a small Lipschitz constant. Discriminator variance regularization does not explicitly regularize the Lipschitz constant of discriminator through differentiation on discriminator but lowers the probability that the Lipschitz constant of the discriminator is high. Discriminator variance regularization with R1 regularization boosts Wasserstein GAN training.

5. References

[1] Martin Arjovsky, Soumith Chintala, Léon Bottou, “Wasserstein GAN,” https://arxiv.org/abs/1701.07875

[2] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, "Improved Training of Wasserstein GANs," https://arxiv.org/abs/1704.00028

[3] Henning Petzka, Asja Fischer, Denis Lukovnicov, “On the regularization of Wasserstein GANs,” https://arxiv.org/abs/1709.08894

[4] Zhiming Zhou, Jiadong Liang, Yuxuan Song, Lantao Yu, Hongwei Wang, Weinan Zhang, Yong Yu, Zhihua Zhang, “Lipschitz Generative Adversarial Nets,” https://arxiv.org/abs/1902.05687

[5] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila, “Analyzing and Improving the Image Quality of StyleGAN,” https://arxiv.org/abs/1912.04958

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter, “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium,” https://arxiv.org/abs/1706.08500

[7] Ziwei Liu, Ping Luo, Xiaogang Wang, Xiaoou Tang, "Large-scale CelebFaces Attributes (CelebA) Dataset," https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

[8] Lars Mescheder, Andreas Geiger, Sebastian Nowozin, "Which Training Methods for GANs do actually Converge?," https://arxiv.org/abs/1801.04406v4