Improve Wasserstein GAN with Group Normalization

Nan Wu Qianyu Cheng Zhiming Guo
Center of Data Science
New York University
New York City, NY 10018

Abstract

The objective of the project is to combine Group Normalization Wu and He [2018] with WGAN-GP (Wasserstein Generative Adversarial Network with Gradient Penalty)(Gulrajani et al. [2017]) to achieve better performance. In this report, we first review recent literatures that are relevant to WGAN-GP and group normalization. We then implement WGAN-GP with resNet using Group Normalization. In the end, evaluation results are compared and presented.

1 Introduction

Generative Adversarial Networks (GANs,(Goodfellow et al. [2014]) is a promising approach for training a model that synthesizes images. However, it is remarkably difficult to train GANs without good heuristics which may not generalize across different network architectures or application domains. Arjovsky and Bottou formally analyzed some issues and proposed an alternative direction. The corresponding algorithm, namely, Wasserstein GAN (WGAN), showed superior performance over GANs. However, researches find out that there are some problems during image generation. The model is slow to converge and the generated images show low quality. Although the weight clipping mechanism used in WGAN satisfies the constraint of Lipschitz continuity, it limits the capability of whole model. In the improved version with gradient penalty Gulrajani et al. [2017], a momentum-based optimizer is proposed, and it efficiently prevents loss explosion issue.

One thing that comes to our notice is that batch normalization does not apply for discriminator in WGAN-GP. In the theory of Kantorovich-Rubinstein duality, the gradient of each pair is unique. WGAN-GP penalizes the norm of the gradient to individual data input, not the entire batch of data. Batch normalization, however, maps whole batch of inputs for a batch of outputs. Therefore, Layer Normalization is recommended by the author in replace of batch normalization.

Instead of normalizing along the batch dimension, Group Normalization(GN) divides the channels into groups and within each group computes the mean and variance. Therefore, similar to Layer Normalization, this scheme also doesn't introduce correlations between examples in the same batch. Considering that GN shows effectiveness in a wide range of visual tasks, which include image classification (ImageNet), object detection, segmentation (COCO), and video classification (Kinect), we think it should be interesting to investigate on WGAN-GP with GN, especially in the discriminator architecture.

2 Background

In the following sections, we briefly review GAN, WGAN, WGAN-GP and a list of methods for normalization.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

2.1 Generative Adversarial Network Related Models

Generative Adversarial Network, as the name suggests, estimated generative models by using an adversarial process (Goodfellow et al.). Wasserstein GAN (Arjovsky et al.) further developed the model by introducing wasserstein distance as a metric to optimize discriminator. As the model still appears slow to converge and hard to train, Gulrajani et al. improved WGAN with a gradient penalty in loss to achieve better learning result.

Generative Adversarial Network (GAN): GAN's architecture is composed of two deep neural nets: a generative model and a discriminative model. The generative model is called generator, which tries to learn the joint distribution $P(x, y)$. Generator takes in noises and generates samples. The discriminative model that learns a function mapping the input data(x) to output class label(y), which is essentially conditional distribution $P(y|x)$, is called discriminator. It receives samples from generator and real data, and learns to be able to tell which one is which. The discriminator is a standard convolutional network, while the generator is a inverse convolutional network.

Wasserstein Generative Adversarial Network (WGAN) and WGAN-GP: In original GAN, when generator is fixed, optimizing discriminator is equivalent to minimize Jensen–Shannon divergence between distribution of generated samples and real data. WGAN however, replaces Jensen–Shannon (JS) divergence with wasserstein distanceArjovsky and Bottou [2017]. The essential reason is that when there is no overlap between P_r (distribution of training data) and P_g (distribution of generated sample), JS divergence will be fixed as $log(2)$, which will lead to zero gradient. Nevertheless, wasserstein distance takes into account how far two distribution are even if there are no overlap between them. However, WGAN tends to learn a simple mapping function after weight clipping: the weights are almost all either -0.01 or 0.01. To help the weights to reflect more information, Gulrajani et al. added gradient penalty to loss function of discriminator. In particular, they introduced a gradient penalty term by noting that the differentiable discriminator (D($\cdot$)) is 1-Lipschitz if and only if the norm of its gradients is at most 1 everywhere,

$$GP|_{\hat{x}} := \mathbb{E}_{\hat{x}}[(\parallel \Delta_{\hat{x}} D(\hat{x}) \parallel_2 - 1)^2],$$

where x is uniformly sampled from the straight line between a pair of data points sampled from the model P_G and the real P_r, respectively.

2.2 Normalization

Normalization is usually formulated as: $x^{'} = \frac{x-\mu}{\sigma}$, where $x^{'}$ is the normalized new feature, x is the original feature, μ and σ are the mean and standard deviation of all examples given that feature. By transforming all features through normalization, one can achieve faster convergence rate using some optimization techniques at the training stage.

Nowadays, there are three normalization techniques: Batch Norm (BN), Layer Norm (LN), and Instance Norm (IN). It is well-known that BN can accelerate training speed, can reduce the amount by what the hidden unit values shift around (covariance shift), and allows each layer of a network to learn by itself a little bit more independently of other layers.

However, normalizing along the batch dimension increases BN's error rapidly if batch size is smaller. To avoid normalizing along batch dimension, IN and LN were created. IN is just like BN but where each batch element is independent, whereas LN normalizes across the channel dimension rather than the batch dimension. Therefore, it is independent of batch size and avoids going through batch dimension.

Furthermore, GN was created by Wu and He [2018] and has better performance than BN while the batch size is small in image classification, object detection, video classification tasks. It also beats LN and IN in visual recognition. Similar to LN, but GN divides channels in a layer into N groups and normalized the features within every group.

3 Experiment and Evaluation

We conducted experiments on the CIFAR-10 (Krizhevsky & Hinton, 2009) datasets. With the ResNet model described in the following table and another simple two layer convolutional network (Simple-CNN). Group normalization layers were inserted after each convolutional layer in the discriminators.

Generator G(z)			
	Kernel Size	Resample	Output Shape
z	-	-	128
Linear	-	-	$128 \times 4 \times 4$
Residual Block	$[3 \times 3] \times 2$	Up	$128 \times 8 \times 8$
Residual Block	$[3 \times 3] \times 2$	Up	$128 \times 16 \times 16$
Residual Block	$[3 \times 3] \times 2$	Up	$128 \times 32 \times 32$
Conv, tanh	3×3	-	$3 \times 32 \times 32$

Discriminator D(x)			
	Kernel Size	Resample	Output Shape
Residual Block	$[3 \times 3] \times 2$	Down	$128 \times 16 \times 16$
Residual Block	$[3 \times 3] \times 2$	Down	$128 \times 8 \times 8$
Residual Block	$[3 \times 3] \times 2$	-	$128 \times 8 \times 8$
Residual Block	$[3 \times 3] \times 2$	-	$128 \times 8 \times 8$
ReLU, mean pool	-	-	128
Linear	-	-	1

Figure 1: ResNet Architecture used in WGAN-GP for CIFAR-10

And the numbers of groups in each layer were tuned within 64, 32, 16, 8, 4, since the number of channels for each layer is 128.

3.1 Quantitative effect of group normalization with WGAN-GP

The criterion we choose to compare model performances is Inception Score (Salimans et al. [2016]), which is a widely-used automatic assessment of image quality. For both the simple-CNN and ResNet, discriminator with group normalization could achieve better Inception score on CIFAR-10 dataset when it is at some of the best number of groups settings. However, considering about the computational time for well training a WGAN, it is not easy to find a good settings for the group normalization.

Model Settings	IS(Unsupervised)
Simple-CNN	3.261
Simple-CNN (Group norm)	3.324
ResNet (Layer norm)	3.632
ResNet (Group norm)	3.841

Figure 2: Due to the size of the network, results for Simple-CNN is from models trained more than 200000 steps but results for ResNet is from models trained only for 10000 steps, under the same batch size. IS is the short for Inception score.

3.2 Sample quality

In Figure 3, we visualized samples generated from ResNet with layer normalization and group normalization, (here we took the best setting of number of groups in each layer according to our experiment). Even though, due to the low resolution of images in CIFAR 10 (32×32), it is difficult for human to tell the classes of images from both sides, model with group normalization gives raise sharper and more realistic samples.

3.3 Data Augmentation with WGAN

In a work from Antoniou et al. [2017], they applied GANs in the data augmentation task by training an encoder to project an input image down to a lower dimensional manifold and concatenating this representation with a random vector as the input for the generator. Combined with a different training schema of the discriminator, they named this model as Data Augmentation Generative Adversarial

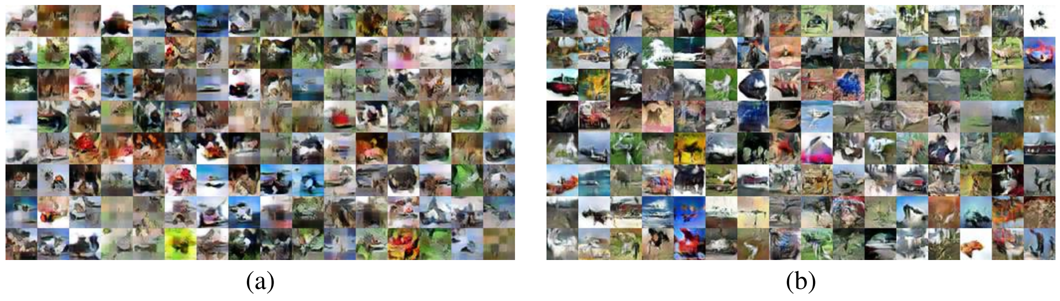

Figure 3: Generated samples by the ResNet model: (a) Generated samples by model with layer normalization and (b) Generated samples by model with group normalization (with the best number of groups in each layer).

Network (DAGAN). We explored the ability of DAGAN in augmenting patch level mammograms and improved its performance by introducing group normalization in the architecture. Patches with a resolution of 256×256 are sampled from a clinically realistic mammography data set which is provided by Medical School of New York University.

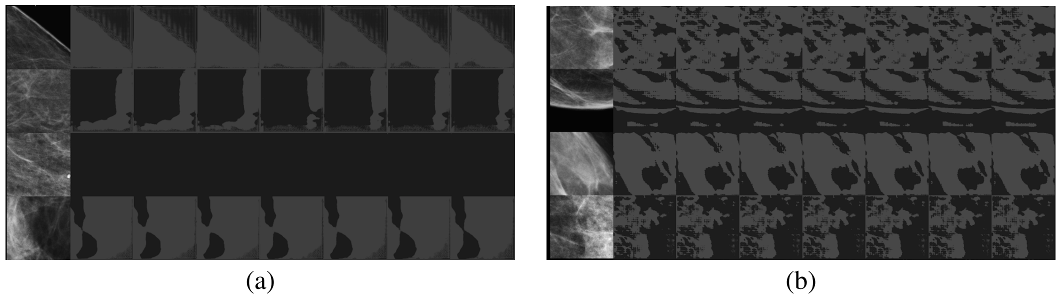

Figure 4: Generated samples by the DAGAN: (a) Generated samples by original DAGAN model and(b) Generated samples by improved DAGAN with group normalization (with the best number of groups in each layer). First column for each group is the original input mammogram patches and others on the same line are the generated samples based on that input patch.

4 Replicate code in PyTorch

We replicated WGAN-GP modelSalimans et al. [2016] in PyTorch[1] based on a the TensorFlow version codes published by the author on GitHub [2]. For the realization of GN, we followed the description in the paper and constructed our own GN layer. We also took a inception score script from GitHub repository[3] as reference to evaluate model performance.

References

Yuxin Wu and Kaiming He. Group normalization. *arXiv preprint arXiv:1803.08494*, 2018.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[1]https://github.com/wooginawunan/wgan_with_group_normalization

[2]https://github.com/igul222/improved_wgan_training

[3]https://github.com/sbarratt/inception-score-pytorch

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. URL `http://arxiv.org/abs/1606.03498`.

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.