Data Augmentation Based on Wasserstein Generative Adversarial Nets Under Few Samples

Yuchen Jiang¹,a, Bin Zhu¹,b, Qi Ma¹,c

¹State Key Laboratory of Pulsed Power Laser Technology, National University of Defense Technology, Hefei, Anhui, China

¹jyc2647118@126.com, bzhubineei@163.com, c905303927@qq.com

Abstract. Aiming at the problem of low accuracy of image classification under the condition of few samples, an improved method based on Wasserstein Generative Adversarial Nets is proposed. The small data sets are augmented by generating target samples through minimax game between the generator and the discriminator. Firstly, label information is introduced into the network to save the cost of manual labeling for the generated images, and then classifier loss is introduced to further improve the quality of generated images, and the classification accuracy is improved after expanding the data set.

1. Introduction
In visual processing and image classification, the performance of convolutional neural network is superior, but the good performance of convolutional neural network depends on the support of large-scale image data sets. The size of training sets plays a vital role in the performance of CNN.

However, in the real world, due to the influence of natural factors and the limitation of data recording conditions, it is usually unrealistic to obtain large-scale labeled data sets, often with a small number of labeled data samples. In the field of data augmentation, Generative Adversarial Nets (GAN) [1] are widely used in various data augmentation problems because of its good generating ability. Yang Yinan et al. improved the original GAN in document [2] and expanded the harmonics of power system. Chen Wenbin et al. used Conditional generative adversarial nets [3] and integrated Gauss mixture model to enhance the data of dense fog weather situation in document [4]. In document [5], Xing Enxu et al. combined transfer learning with generative adversarial nets, trained in large sample Chinese character font dataset, and the weight obtained was transferred to small sample Chinese character font dataset to generate a large number of small sample Chinese character fonts.

After data augmentation, a commonly used evaluation index is used for classification tasks. The accuracy of classification is used to judge the quality of generated samples. Starting from the evaluation index of data augmentation, this paper distinguishes it from the generator which is not related to classifier in the literature mentioned above. The loss of classifier classification is added to the generator, and the discriminator and classifier jointly guide the generator to generate samples for classification.

2. Related Work
Generative Adversarial Nets. In 2014, Goodfellow et al. first proposed the concept of Generative Adversarial Nets, and set up a generator G and a discriminator D, in which the generator is responsible for generating data close to the real data, while the discriminator tries to distinguish the real data from
the pseudo data created by the generator. The input of generator G is an arbitrary random variable. In his original paper, Goodfellow mapped the noise signal \( p_z(z) \) as input variable to the data space \( G(z, \theta_g) \) of the generator, and obtained the probability distribution \( p_g \) of the generator. Thus, the generator G is represented by the parameters \( \theta_g \) in the multi-layer perceptron. The input of discriminator D is real data or generated data from generated model G. A multi-layer perceptron \( D(x, \theta_d) \) is also defined and a scalar is output. \( D(x) \) denotes the probability that x comes from real data instead of generating data.

![Fig. 1: Basic Structure Framework of GAN](image)

In GAN, training D maximizes the probability of assigning correct labels to both real data and generated data, while training G minimizes \( \log(1 - D(G(z))) \). Its optimization function is shown in equation (1):

\[
\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]
\]

Updating discriminator D by reducing the gradient:

\[
\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))]
\]

Updating Generator G by reducing the gradient:

\[
\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [1 - \log(1 - D(G(z^{(i)})))]
\]

Finally, they reach Nash equilibrium, and the output of discriminator is 0.5. That is to say, discriminator D can not judge whether the input is from real data or from generated data. However, under the approximate optimal discriminator in the original GAN, the loss of the minimization generator is equivalent to minimizing the JS divergence between the real data distribution and the generated data distribution, which eventually leads to the gradient of the generator being zero and the gradient disappearing.

Wasserstein Generative Adversarial Nets. Martin Arjovsky et al. proposed Wasserstein Generative Adversarial Nets (WGAN) in document [6]. This model uses Wasserstein distance (also known as Earth-mover, EM distance) instead of Jensen-Shannon (JS) divergence to evaluate the distance between actual samples and generated samples. The advantage of EM distance over JS divergence is that it can measure the distance between two non-coincidence parts, and it further enhances the stability of training compared with the original GAN. Gulrajani et al. further improved WGAN in document [7], and Wasserstein Generative Adversarial Nets-Gradient Penalty (WGAN-GP) model was proposed. The weight clipping used in WGAN to ensure Lipschitz restriction was adjusted to gradient penalty.

3. Conditional Wasserstein Generative Adversarial Nets

This paper improves the WGAN-GP model. There are two main works: The first is to use the idea of conditional generative adversarial nets for reference, and add conditional information into the input of generator and discriminator in order to control the type of generated samples and reduce the cost of labeling generated samples manually. The second is to add a classifier and work together with the discriminator to generator to control the generation of samples.
The discriminator loss function adds label information on the basis of WGAN-GP, as shown in Equation (4):

$$L = D_w(\tilde{x}|y) - D_w(x|y) + \lambda \left( \| \nabla_x D_w(\hat{x}|y) \|_2^2 - 1 \right)^2$$

(4)

Where $\tilde{x}$ is the generated image of the generator, $x$ is the real image, $y$ is the label, and $\hat{x}$ is the interpolated image of the generated image and the real image. The value is shown in Equation (5):

$$\hat{x} = \varepsilon x + (1-\varepsilon)\tilde{x}$$

(5)

Two classifier losses are added to the generator loss function:

$$L = -D_w(\tilde{x}|y) - y \log(C(\hat{x})) - y \log(C(x))$$

(6)

**Discriminator structure.** The discriminator uses four layers of convolution. The size of convolution kernel is 3 and the step size is 2. The number of channels increases gradually. After the convolution layer, Batch normalization is used and activated by Leaky ReLU. Then dropout regularization is used. The structure diagram is as follows:

**Generator structure.** In the generator, the input layer is Dense layer, using two Upsampling layers and three convolution layers. The size of convolution kernel is 4 and the step size is 1. The number of channels decreases gradually. After the convolution layer, Batch normalization is used and activated by ReLU. The activation function of the output layer is tanh. Its structure sketch is as follows:
Input
Dense (128*7*7)
Reshape (7,7,128)
Upsampling
Conv(128,k=4)
Batchnormalization
ReLU
Upsampling
Conv(64,k=4)
Batchnormalization
ReLU
Conv(1,k=4)
tanh

Fig. 4: Generator structure

Classifier structure. The classifier uses one layer of convolution, the size of convolution kernel is 5, the step size is 1, and the number of channels is 32. Then Maxpooling and Dropout regularization are used. And two Dense layers are used. The output activation function is Softmax. The structure diagram is as follows:

Input
Conv(32,k=5)
Maxpooling
Dropout(0.2)
Flatten
Dense(128)
ReLU
Dense(10)
Softmax

Fig. 5: Classifier structure

4. Experiments
The experimental data set used in this paper is a subset of the MNIST database of handwritten digits. The MNIST data set consists of 60,000 training samples and 10,000 test samples. Each sample is a 28*28 pixel gray handwritten digital image. In the experiment, a total of 500 pictures were randomly selected from 10 types of the original MNIST training set, and 200 pictures were randomly selected in the original MNIST test set, 100 as validation set and 100 as test set.

During the experiment, three data sets were selected as comparative experiments. The first training data set selected the subset of 500 pictures. The second training data set added augmented pictures to the subset, and the augmented model chose CWGAN-GP, but did not introduce classifier loss. 500 pictures after 20,000 epochs of training were selected to augment, a total of 1000 images. The third training data set also added augmented pictures to the subset, but the augmented model used CWGAN-GP combined with classifier loss. 500 images after 20,000 epochs of training are selected to augment the original data set, a total of 1000 images. The results of using the same classifier to classify the validation set and the test set are shown in the following table:

| Data set | Val | Test |
|----------|-----|------|
| 1        | 83% | 82%  |
| 2        | 86% | 85%  |
| 3        | 87% | 88%  |

It can be seen that the proposed CWGAN-GP+Classifier model improves the accuracy of the classifier under the condition of fewer samples, and performs better than CWGAN-GP. It also shows that adding classification loss to the generative adversarial nets can improve the ability of sample generation.
5. Conclusion
Aiming at the problem of low classification accuracy under the condition of few samples, this paper improves the WGAN-GP model. Firstly, label information is added as a condition to reduce the cost of manual labeling after image generation. Then classifier loss is introduced into generator loss function. The proposed CWGAN-GP+Classifier model achieves better results in the subset of MNIST dataset than CWGAN-GP model, and has certain reference significance for the classification task of few samples.

References
[1] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C].Advances in neural information processing systems. 2014: 2672-2680.
[2] Yang Yinan, Qi Linhai, et al. Research on Small Sample Data Generation Technology Based on Generative Adversarial Nets [J]. Power construction. 2019, 40(05):75-81.
[3] Mirza M, Osindero S. Conditional generative adversarial nets [J]. arXiv preprint arXiv:1411.1784, 2014.
[4] Chen Wenbin, Guan Zhengxiong, et al. Data Augmentation Method Based on Conditional Generative Adversarial Nets [J]. Computer application. 2018, 38(11):259-265.
[5] Xin Enxu, Wu Xiaoyong, et al. Double-Layer Generative Adversarial Nets Based on Transfer Learning [J]. Computer Engineering and Application, 2019,55(15):38-48.
[6] Arjovsky M, Chintala S, Bottou L. Wasserstein gan[J]. arXiv preprint arXiv:1701.07875, 2017.
[7] Gulrajani I, Ahmed F, Arjovsky M, et al. Improved training of wasserstein gans[C].Advances in neural information processing systems. 2017: 5767-5777.