Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 5
AIDML 2024

Fast and High-resolution Image Generation Based on Improved DCGAN

Yixuan Xin *

School of Materials Science and Engineering, Wuhan University of Technology, Wuhan, Hubei, 430070, China

* Corresponding Author Email: xinyx@whut.edu.cn

Abstract. Among generative models, Generative Adversarial Network (GAN) has been sought after by researchers since its proposal. The researching fields of image generation, style transformation, data augmentation, super-resolution, image restoration, and image transformation have shined because of GAN. Deep Convolution Generative Adversarial Network (DCGAN), as an early neural network to improve GAN, solved the problem of unstableness during training. It can be easily scaled to deal with larger datasets and more sophisticated tasks, so various image generation and manipulation tasks can be tackled by this powerful tool. Nevertheless, it still has certain problems. This research investigates the hyperparameters, label smoothing and improved model's effect on the quality and speed of image generation, and finally selects the appropriate hyperparameters and label smoothing to cooperate with the improved model to quickly generate clearer images with DCGAN in the case of few samples and few number of trainings. This work can bring some ideas for saving computational resources and data for training.

Keywords: Deep Convolution Generative Adversarial Networks; label smoothing; image generation.

1. Introduction

In 2014, Ian J. Goodfellow and colleagues introduced the concept of Generative Adversarial Networks (GAN) [1]. The idea of pitting two neural network models against each other has been presented by scholars previously, such as J. Schmidhuber who mentioned a reinforcement learning algorithm with both internal and external feedback [2] as well as an adaptive predictor that uses other units to predict the current unit, where each unit will minimize predictability as an adversary [3]. As the origination of generative adversarial network models, GAN has triggered research and discussions among scholars all over the world, and people have applied it to image generation [4], style transformation [5], data augmentation [6], super-resolution [7], image restoration [8], image transformation [9], and so on.

In 2016, the convolutional neural network was successfully added to the GAN to achieve unsupervised learning by Alec Radford et al. [10], which performed well on image classification tasks. They removed the pooling layer with a convolution with stride in the GAN model, as well as the layer that was completely connected. The activation function, which is ReLU, was used in the generator except for the last layer using Tanh activation. The discriminator utilized the LeakyReLU activation function, and both were normalized by Batch Normalization. The emergence of DCGAN makes the training of GAN more stable and improves the image generation's quality.

Researchers' constant objective, when it comes to the field of image generation, is to improve the efficiency and quality of images that are produced. J. D. Curtó et al [11] designed High-Resolution Deep Convolutional Generative Adversarial Networks (H-DCGAN) based on DCGAN inspired by Self-normalizing Neural Networks (SNNs). The activation functions of the generator and discriminator in the original DCGAN were modified to be Scaled Exponential Linear Unit (SELU) functions, which greatly improved the convergence speed of the model. Kun Sun et al [12] introduced residual neural networks into the generator of DCGAN to realize high-resolution generation of medical images. Yiheng Song et al [13] aimproved 1-D DCGAN and utilized this model for generating minority classes in the radar HRRP data, which solved the problem of imbalance in the training samples and realized the data augmentation.

Based on the above work, it is found that there is less work directed at improving the quality of generated images with DCGAN, and related work generally prefers the use of GANs to achieve image super-resolution as well as some other image generation applications in the specific fields. This research aims to look into the impact of the network's learning rate, label smoothing and adjustment of the network model on the realism and efficiency when it comes to the generation of headshot images of anime characters based on the DCGAN model proposed by Alec Radford et al. In Section 2, the DCGAN network model and the related mathematical principles, the improved DCGAN network model and the experimental design, including the experimental parameters and environment configuration settings, are discussed. In Section 3, the generation effect and generation efficiency of different experiments are compared. Finally, the related results are discussed and analyzed, and the direction of improvement is proposed.

2. Method

2.1. Mathematical Principles of DCGAN

The math behind DCGAN is derived from GAN with some improvements to the network model. The mechanism of GAN can be divided into two parts. In the generator part, a randomly generated noise z is fed into the generator network and is converted into a fake sample $G(z)$ after the generator neural network. The other part is the discriminator, whose input can be either a real sample or a fake sample, discriminate the likelihood that the input sample is from the real one. The output of the discriminator, $D\left(G(z)\right)$, is the probability that it will conclude that the fake sample is part of the real distribution if it receives the fake sample $G(z)$ as input. On the other hand, If the discriminator receives the genuine sample x as input, the output will be $D(x)$, which indicates the probability that it believes the input is from the real distribution.

When it comes to application, people add labels to the input samples, e.g., 1 for real and 0 for fake. This converts the GAN from unsupervised learning to supervised learning.

Equation (1) denotes the loss equation of GAN, which is shown as follows.

$$V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[\log\left(1 - D\left(G(\boldsymbol{z})\right)\right)\right] \tag{1}$$

The initial term represents the discriminator's forecast on actual data, specified as the mathematical expectation $\mathbb{E}$ of the logarithmic value of the output $D(x)$ of the discriminator when the input is obtained by sampling from a distribution of real data, and the second term is the prediction on fake data, specified as the mathematical expectation $\mathbb{E}$ of the output $D\left(G(z)\right)$ of the discriminator when the input is sampled from a fake sample.

The discriminator wants to discriminate all fake samples as fake, so it wants $D\left(G(z)\right)$ to be as small as possible, while ensuring that all true samples are discriminated as true, i.e., $D(x)$ is as large as possible. For the generator, it wants the samples it generates to be real enough to deceive the discriminator, which means a larger $D\left(G(z)\right)$ as possible.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[\log\left(1 - D\left(G(\boldsymbol{z})\right)\right)\right] \tag{2}$$

It is evident from the foregoing analysis that the discriminator wants the whole equation to be as large as possible, while the generator must allow $\log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$ to get progressively smaller, i.e., the whole equation is as small as possible. This creates a confrontation in the equation (2).

2.2. Model Structure of DCGAN

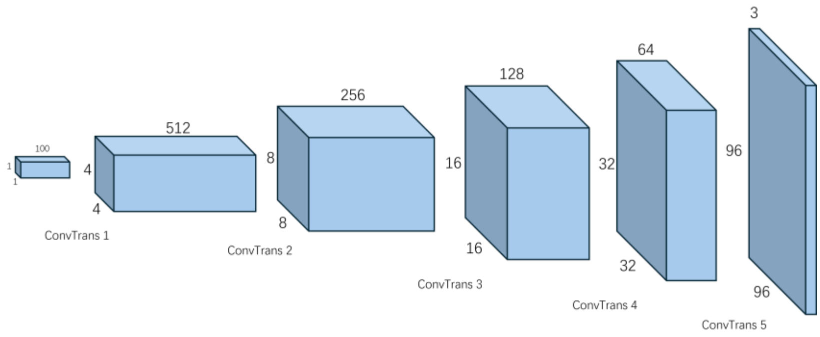


Fig. 1 The model structure of generator in DCGAN (Figure Credits: Original).

The generator model diagram in DCGAN is displayed in Fig. 1. Starting from a random noise, it undergoes a 5-layer deconvolution operation, Batch Normalization and ReLU activation come after the first four layers of deconvolution, while Tanh function alone is used to activate the final layer.

The network structure of the discriminator, on the other hand, is the reverse of the above model, with the inverse convolution changed to convolution, and the remaining layers' size and channel count stay constant. In the first 4 layers, the normalization function is removed. In the final layer, a sigmoid function takes the place of the activation function, which is LeakyReLU.

2.3. Improved DCGAN

2.3.1. Learning Rate

The first point to address the problem of DCGAN network modification is the selection of network hyperparameters. One of the most crucial hyperparameters is learning rate, and even tiny changes in it may lead to fundamental changes in the training process.

When training DCGAN, it is like a teacher teaching a student. The discriminator is equivalent to the teacher and the generator is equivalent to the student. The whole training process is similar that the teacher teaches the student to write an essay, and after the student's essay is read by the teacher, the teacher will tell the student how to revise it by comparing the gap between it and the excellent essay. The training process ends when the student polishes his essay that the teacher thinks is as good as the best one. In this process, only when the teacher's level is high enough, i.e., when the teacher has read a lot of excellent essays and has a lot of experience, his students will be able to write equally good essays based on the teacher's feedback.

The aforementioned research suggests that throughout the training phase, the discriminator's learning rate can be set somewhat higher than the generator's, which ensures that the loss of the discriminator is already quite low when the generator is learning, thus guaranteeing the efficient learning of the generator.

2.3.2. Label Smoothing

In numerous tasks, including speech recognition, machine translation, and picture classification, label smoothing has been utilized to increase the precision of deep learning network models [14].

The real image's label can be adjusted to a lower value, like 0.9, if it is currently set to 1. By avoiding overfitting the model, this technique keeps the discriminator from being overly confident in its classification labels, or, to put it another way, from depending too much on a small number of characteristics to decide whether a picture is true or false.

2.3.3. Model Modification

With the idea of improving other models, Fig. 2 shows that the generator of DCGAN is modified from the original inverse convolution to three parts. The initial stage consists of a downsampling process with five layers, each layer operates almost the same way as the generator of the original

DCGAN with a slight change of activation from LeakyReLU to ReLU. The second part is a 9-layer residual neural network, and each layer of the residual network is similar to downsampling, except that another layer of convolution and Batch Normaliztion are added after activation. The difference is that after the output of the last downsampling layer is handled by the first residual network layer, it is combined with the original output from the residual network and transmitted to the second layer of residual network. Additionally, each successive layer's output of the residual network is mixed with the first layer's output to serve as the subsequent layer's input. The third part is a 5-layer upsampling, where each layer includes deconvolution, Batch Normalization and ReLU activation.

The difference between the modified discriminator and the one in the original DCGAN is that after 4 layers of convolution, normalization, and activation, the last layer is repeated again after convolution, which is equivalent to training the discriminator twice, and then average pooling is used to complete the training.

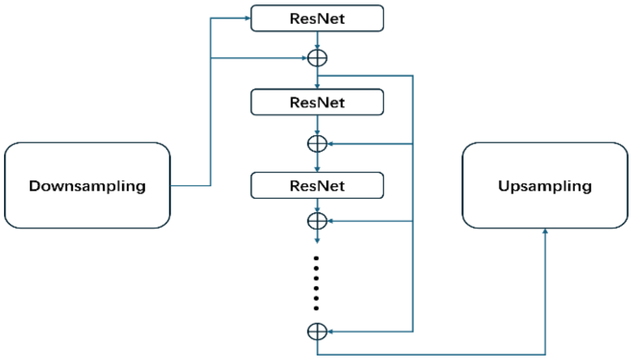


Fig. 2 The generator of the model after modification (Figure Credits: Original).

2.4. Design of Experiments

Anime Face is selected as the dataset for training, which contains 10000 images, each of it is 96*96 colorful images [15]. Some of the images are exhibited in Fig. 3.

Fig. 3 Representative examples from Anime Face dataset [15].

This experiment was conducted under Ubuntu 22.04 operating system using Python 3.10, deep learning framework PyTorch 2.1.0. Computing resources CPU is Intel Xeon Platinum 8255C @ 2.50GHz, GPU is RTX 2080 Ti with 43GB of RAM and 11GB of video memory.

Before determining the experimental parameters, the learning rate was divided into 5 segments from the interval 0.001 to 0.00001, and from 0.01 to 0.001. Ten experiments were conducted to compare the loss function descent curves of the two models, the learning rate was set to 0.0064 to achieve the best convergence of the loss function in Fig. 4(a).

Following the analysis in Section 2.3.1, the following experiments in Table 1 were designed to determine the appropriate dual learning rate as well as label smoothing. Firstly, the learning rate distribution of experiments 1, 2 and 3 in Table 1 are compared, and then the learning rate distribution is determined according to the convergence effect of the loss function. Based on this, three different smoothing labels are designed to determine the optimal value. Note that in Table 1, as for the distribution of learning rate in the table, the left represents the discriminator, the right denotes the generator, and label smoothing for real image labels - fake image labels.

Table 1. Three Scheme comparing.

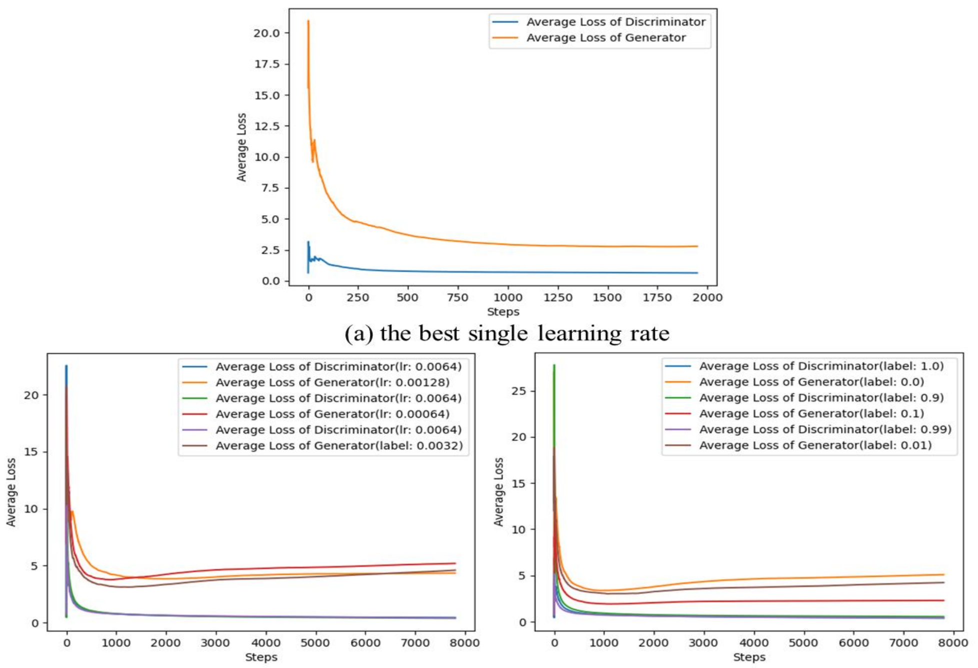
Experiment	Distribution of learning rate	Label smoothing
Experiment 1	0.0064-0.0032	1-0
		1-0
Experiment 2	**0.0064-0.00128**	**0.9-0.1**
		0.99-0.01
Experiment 3	0.0064-0.00064	1-0

Fig. 4(b) illustrates that the loss function curves of discriminator almost overlap when the learning rate is set as discriminator-generator: 0.0064-0.00128, but the generator's loss function has the curve closest to convergence when its learning rate is 0.00128. In Fig. 4(c), the discriminator's loss function curves almost overlap, it is clear to see that the generator loss function is reduced best than the other generator curves when the smoothing label is 0.9-0.1. This determines the generator learning rate to be 0.00128, the discriminator to be 0.0064, and the real image-fake image label to be 0.9-0.1.

Parameters of the experiment are shown in Table 2. Note that in Table 2, the value of batch size was adjusted to 24 in Experiment 8 due to the lack of memory and in Experiments 6, 7, 8, the generator learning rate was 0.00128 and the discriminator was 0.0064.

Table 2. Parameters of the experiments.

Parameters	Value	Explanation
batch size	128	Batch size
image size	96	size of images
nz	100	size of input vector
ngf	64	intermediate channels of generator
ndf	64	intermediate channels of discriminator
epoch	100	epoch for training
Dlr	0.0002	discriminator’s learning rate
Glr	0.0002	generator’s learning rate
Bata1	0.5	Adam momentum 1
Bata2	0.999	Adam momentum 2



(a) the best single learning rate

(b) different distributions of learning rate (c) different distributions of label smoothing

Fig. 4 Learning rate and label smoothing experimental results (Figure Credits: Original).

Based on the above analysis and in response to the improvement ideas in Section 2.3, the experiment shown in Table 3 was designed. Note that in Table 3, as for the distribution of learning rate in the table, the left represents the discriminator, the right denotes the generator, and label smoothing for real image labels - fake image labels.

Table 3. Experimental program.

Experiment	Model	Learning rate	Label smoothing
Experiment 5	DCGAN	0.0002-0.0002	1.0-0.0
Experiment 6	DCGAN	0.0064-0.00128	1.0-0.0
Experiment 7	DCGAN	0.0064-0.00128	1.0-0.0
Experiment 8	**Modified DCGAN**	**0.0064-0.00128**	**0.9-0.1**

3. Results

The outcome of the aforementioned trials is categorized individually according to epochs, specifically in 1, 5, 10 and 20.

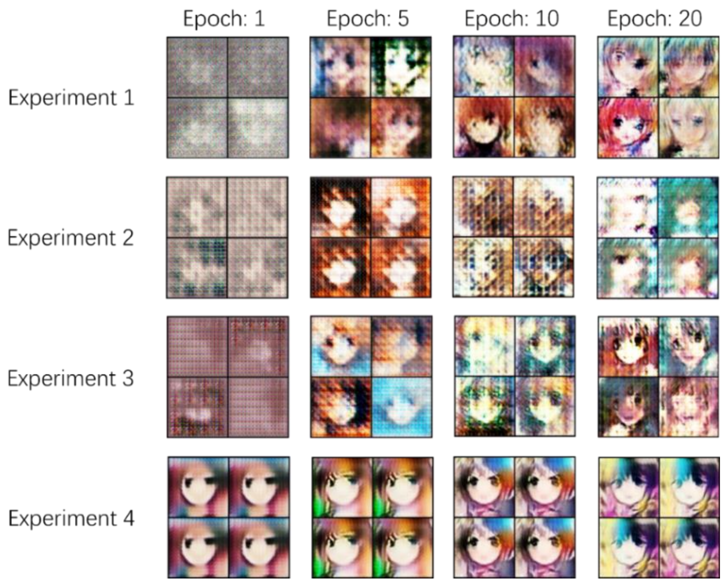


Fig. 5 Results of the experiments (Figure Credits: Original).

The experimental results in Fig. 5 demonstrate clearly that with fewer samples and fewer training times, the original DCGAN-generated images are very blurry, the color scheme is not fully revealed, and the outlines of the anime heads are not clearly depicted and generated quickly. After modifying the model, the rough outline of the anime face in Experiment 7 can be recognized very clearly after the first epoch of training, and in the subsequent comparisons, the eyes, hair, and face outlines reflect the excellent results that are incomparable to the aforementioned model.

4. Conclusion

In this paper, with the goal of DCGAN to quickly generate better quality images with fewer samples and fewer training times, a sequence of experiments was devised to explore the effects of learning rate, label smoothing, and model modification on the quality of the generation as well as the speed of the generation. Eventually, it is found that the modified model DCGAN is better in terms of generation quality and generation speed when the discriminator and generator use different learning speeds, 0.0064 and 0.00128, respectively, and the smoothing label is 0.9-0.1 for real image-fake image.

In future work, it can be considered to further improve the image generation quality of DCGAN on this basis, to realize that the produced photos exhibit superior quality compared to the original dataset, even to the extent that the naked eye cannot distinguish whether it is a generated image or not.

References

[1] Goodfellow, Ian, 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, 2014, 27: 1-9.

[2] Schmidhuber, Jürgen. Making the world differentiable: on using self-supervised fully recurrent neural networks for dynamic reinforcement learning and planning in non-stationary environments. Inst. für Informatik, 1990, 126.

[3] Schmidhuber, Jürgen. Learning factorial codes by predictability minimization. Neural computation, 1992, 4(6): 863-879.

[4] Liu, Bingqi, Jiwei Lv, Xinyue Fan, Jie Luo, and Tianyi Zou. Application of an improved dcgan for image generation. Mobile Information Systems, 2022: 1-14.

[5] Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, 2017: 2223-2232.

[6] Zhao, Shengyu, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. Advances in neural information processing systems, 2020, 33: 7559-7570.

[7] Ledig, Christian, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017: 4681-4690.

[8] Liu, Guilin, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In Proceedings of the European conference on computer vision, 2018: 85-100.

[9] Choi, Yunjey, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2018: 8789-8797.

[10] Radford, Alec, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. ArXiv Preprint, 2015: 1511.06434.

[11] Curtó, Joachim D., Irene C. Zarza, Fernando De La Torre, Irwin King, and Michael R. Lyu. High-resolution deep convolutional generative adversarial networks. ArXiv Preprint, 2017: 1711.06491.

[12] Sun, Kun, Liangqiong Qu, Chunfeng Lian, Yongsheng Pan, Dan Hu, Bingqing Xia, Xinyue Li, Weimin Chai, Fuhua Yan, and Dinggang Shen. High-Resolution Breast MRI Reconstruction Using a Deep Convolutional Generative Adversarial Network. Journal of Magnetic Resonance Imaging, 2020, 52(6): 1852-1858.

[13] Song, Yiheng, Yang Li, Yanhua Wang, and Cheng Hu. Data augmentation for imbalanced HRRP recognition using deep convolutional generative adversarial network. IEEE Access, 2020, 8: 201686-201695.

[14] Müller, Rafael, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help?. Advances in neural information processing systems, 2019, 32: 1-10.

[15] A collection of high-quality anime faces. URL: https://github.com/Mckinsey666/Anime-Face-Dataset. Last Accessed: 20204/03/14