PolyU COMP4423 Computer Vision- Group Project

GAN-Based image generation task implementation

Yuan Yunchen 21100038d

1, Abstract

In this project, We are going to develop a back-end algorithm for fashion image generation. The Generative Adversarial Network is used as the underlying framework. Based on this, different model structures are proposed to address the problem, including base GAN structure, DCGAN, and WGAN.

Among them, the base model uses a simple neural network structure and adopts the trivial learning paradigm. The optimized model is based on Deep Convolutional Generative Adversarial Networks, which not only significantly enhances the performance, but also extends the functionality of control label image generation. Finally, we use the Weighted Generative Adversarial Networks (wGAN) structure and modify the loss function to optimize the image quality.

1. Initial analysis

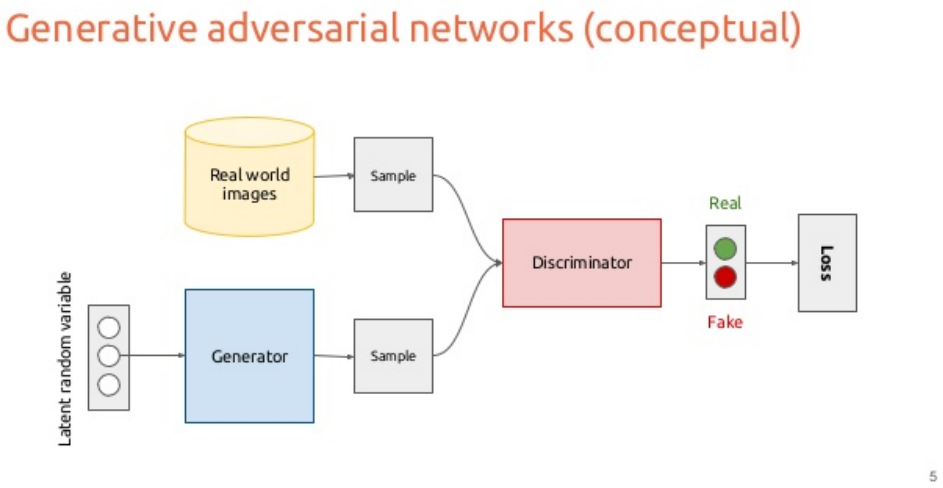
2.1, fashion image Dataset visualization ---- Task 1

2.2, GAN framework introduction

We adopt the Generative Adversarial Network's framework to develop the image generation model. It is the underlying framework of all the model structures proposed to address the problem. Therefore, it is necessary to briefly introduce the basic logic of the Generative Adversarial Network.

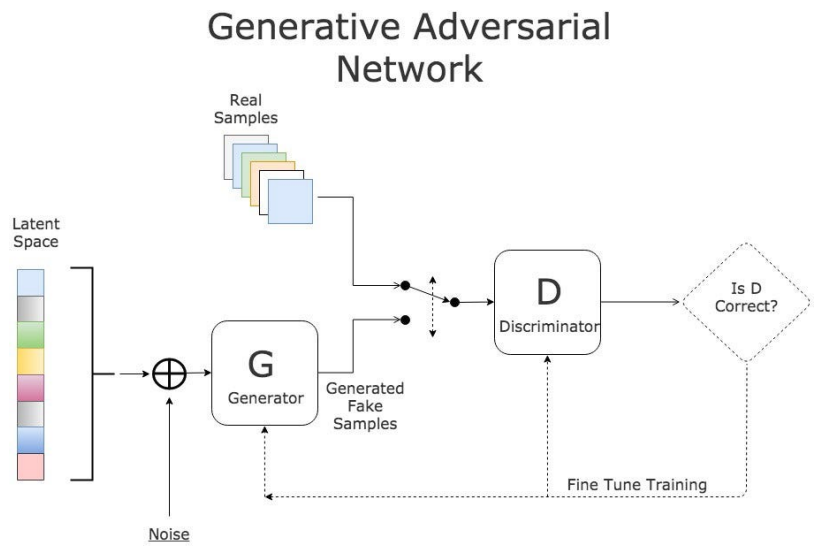
(1) architecture

The GAN model architecture involves two sub-models: a generator model for generating new examples and a discriminator model for classifying whether generated examples are real, from the domain, or fake, generated by the generator model.



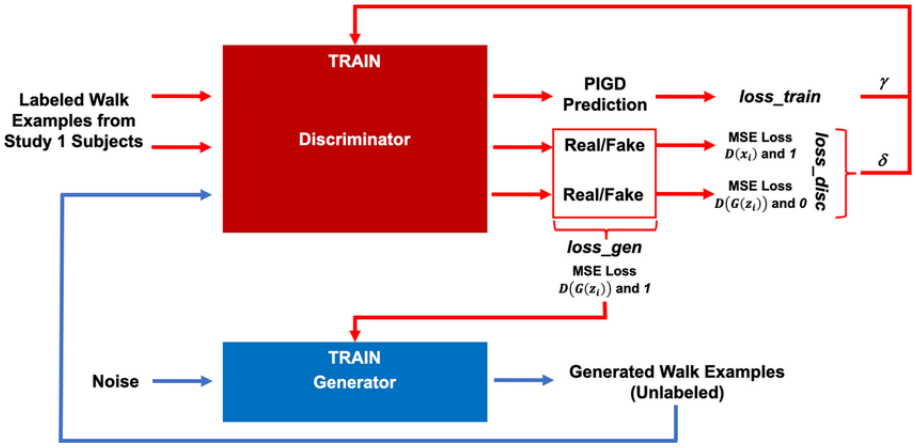
The generator is trained to improve the quality of the image generated, and the discriminator is used to help the generator do the training.

(2) Learning paradigm



Source: https://ai.plainenglish.io/review-cgan-conditional-gan-gan-78dd42eee41

The generator model and discriminator model are trained simultaneously. The discriminator uses the generated image and real-world data to perform training based on the classification task. And the generator models' loss function is based on the judgment of the discriminator. In such a way, the two models are adversarial to each other and enhance the accuracy during the process.



https://www.researchgate.net/figure/GAN-training-paradigm-The-discriminator-predicted-a-PIGD-score-as-well-as-a-real-fake_fig2_363408707

The pseudo-code for the model training process is as follows:

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**
  for k steps **do**
    • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
    • Update the discriminator by ascending its stochastic gradient:

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

  end for
  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  • Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Source: From the paper： Generative Adversarial Nets

2.3, Model design strategy analysis

We have designed different model structures to perform the task, but they adopt the same framework as what is discussed above. Although the framework is the same, there are still many difference:

a) model structure of the generator

b) model structure of the discriminator
c) learning paradigm
(learning rage optimization, loss function, "adversarial" paradigm)
d) expand functionality implementation

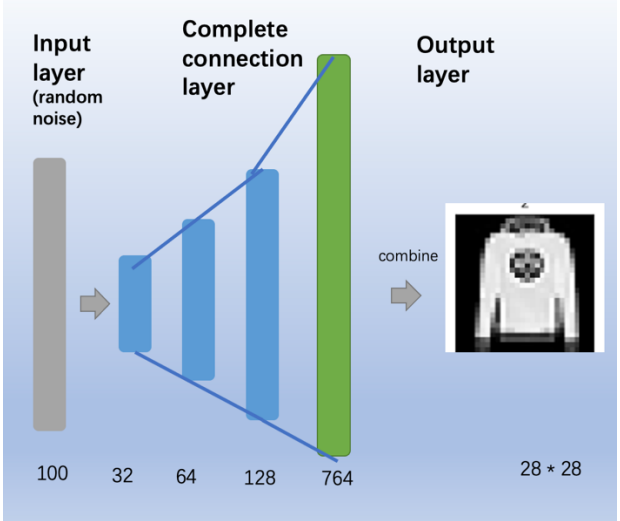
These factors are also the main factors to consider when optimizing the model

3. Basic GAN ----Task 2: base model

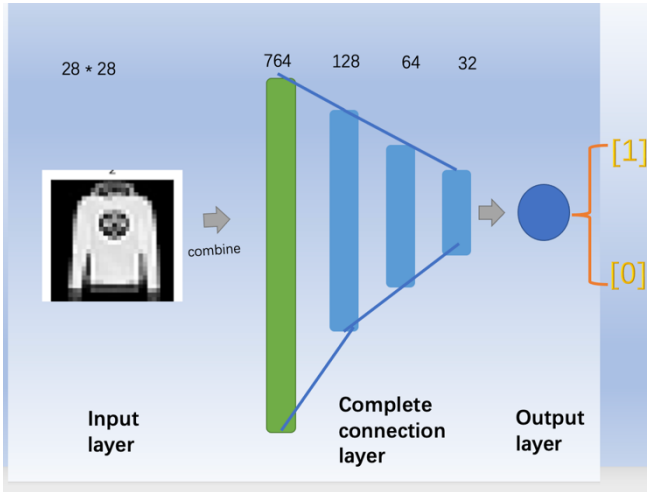
3.1, model analysis

(1) model structure

generator model.



Descriptor model



Both the generative model and the discriminative model adopt the basic neural network structure. They are basically symmetric in network structure.
The input layer and the output layer are set according to the problem. The input of the generated model is a random vector with 100 dimensions, and the vector parameters conform to uniform distribution in the interval of -1 to 1. All parameters are equivalent to each other. No special parameter is set to express information.
The output of the differential model uses a network node. The output ranges from 0 to 1, indicating the probability that the picture is a picture in the real data set.
The middle layer adopts three fully connected layers, and the number of parameters in each layer keeps the gradient rising or falling roughly.

(2) learning paradigm

a) Training process:

```
for each epoch do:
    for each batch do:
        train generator {
```

generate random samples → use generator to generate fakeimage → use discriminator to classify → get loss function → update parameter based on gradient descent

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

}

train discriminator {

generate m fake image by discriminator + get m real image from dataset → use discriminator to classify → get loss function based on label and result → update parameters

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

}

end for

end for

b) learning rate:

fixed learning rate 0.002

c) loss function properties:

The loss function uses sigmoid activation function uses the given logits to compute binary cross entropy between the target and the output.

$$p_{ij} = sigmoid(X_{ij}) = \frac{1}{1 + e^{-X_{ij}}}$$

$$L_{ij} = -[Y_{ij} * ln(p_{ij}) + (1 - Y_{ij}) ln(1 - p_{ij})]$$

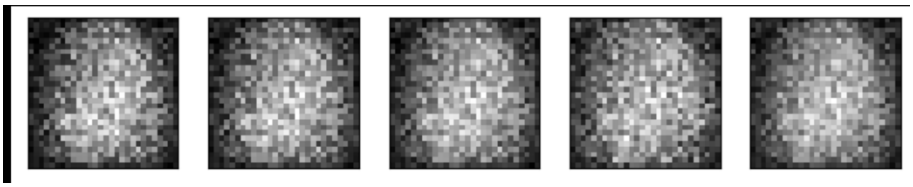
(implemented by nn.BCELogitsLoss() function in tensorflow.)

3.2 performance ---- Task 3: model evaluation

(1) generated image

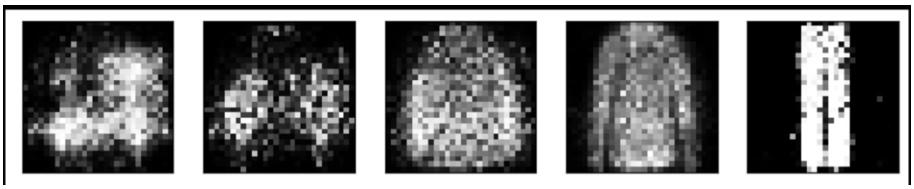
---- Human-judged Image generation performance analysis

Epoch 0:



The initialized model will generate the random noise image.

Epoch 40:
The outline of the cloth can be recognized, although some details are not precise enough.



Epoch 60:
Most images become clearer. From the observation, The performance is better than epoch 40.

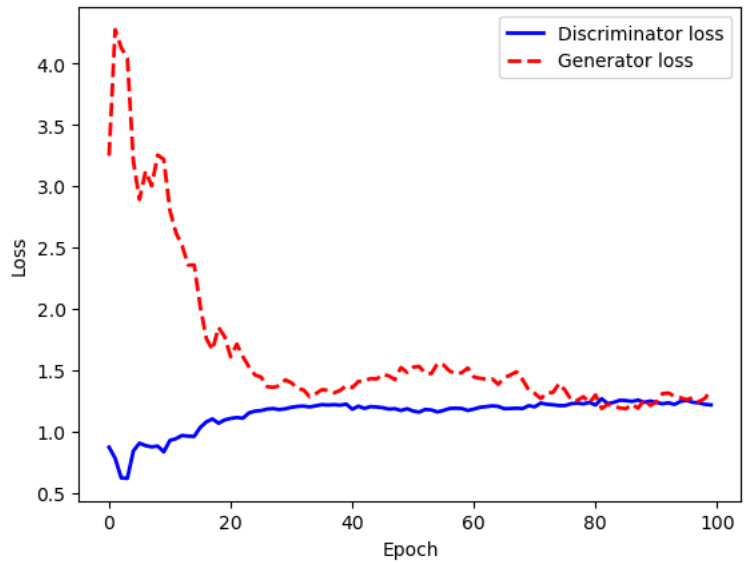
Epoch 100:
Compared with the previous epoch results, the shape of clothes becomes more blurred, the pixel blocks are more dense, and the differentiation is reduced

From the human-judgment analysis, the best learning performance happens in the interval of the training. After epoch 100, the performance decreased instead.

(2) learning loss
---- quantitively performance analysis

Now, let's take a quantitative and more systematic approach to analyzing model performance. (The learning loss curve of model generation and model differentiation during model training is analyzed. Through the analysis of the learning loss curve, we can approximate where the model reaches the optimal performance (the balance point between the generation model and the differentiation model loss, and the boundary of

the model overfitting). In addition, combined with the actual observation situation, we can analyze whether the model size and model structure match the problem, which lays the foundation for further optimization of the model



We can see that before epoch 20, the loss for generating models decreases rapidly, while the loss for differentiating models increases in a roughly linear fashion. This trend is consistent with expectations, indicating that the model is rapidly undergoing training. However, approximately after epoch 40 ~ 60, the loss of distinguishing model becomes convergent, and the loss of generating model also becomes convergent after fluctuations. It indicates that the fitting ability of the model gradually reaches the limit and it is difficult to further optimize the accuracy. The trend of loss curve is roughly consistent with the actual observation results

3.3 evaluation

According to previous observation and loss curve analysis, the main problem of the model is insufficient fitting ability. This is influenced by several factors:

1. Model structure:

(1) The model of the basic model is too simple and, which does not fully play the role of the large data set size.

(2) The model has a single structure, with only three fully connected layers, without corresponding design and model structure optimization based on the problem situation

2. learning paradigm

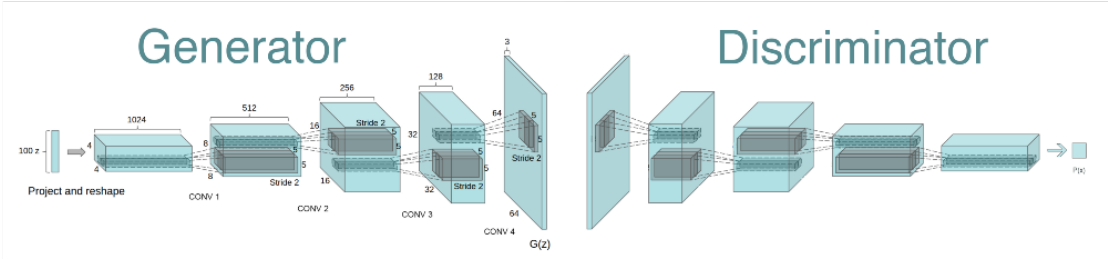
(1) The learning rate is fixed, and error occurs to pivot to the extreme point after training to a certain degree

(2) The generation model and the differentiation model adopt the same iteration times, which makes the differentiation model have limited ability to adjust according to the changes of the generation model. The synchronicity of the two models is not good

2. DCGAN ---- Task 4: model optimization

According to the preliminary judgment, the main problem lies in the model structure. The following model improvement is mainly in the field of model structure.
According to previous experience, the deep convolutional neural network has a strong feature extraction function and has outstanding performance in classification and recognition tasks. Can you use the same structure for the image generation task to improve the complexity model structure and the model's adaptivity?

3.1, model optimization



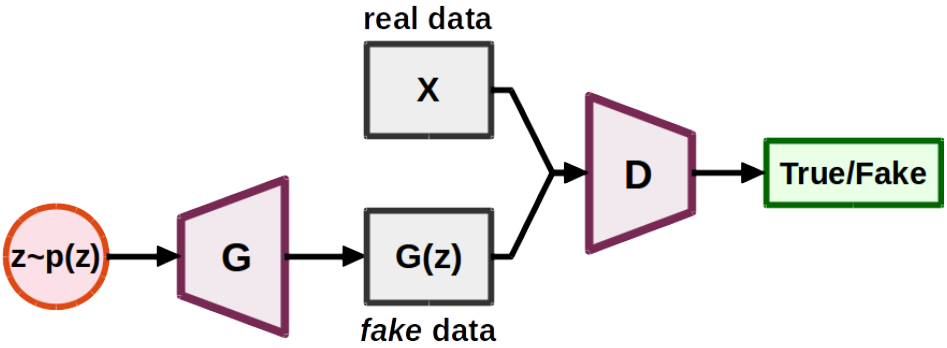
Source: From the paper "Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks"
DCGAN uses convolutional and convolutional-transpose layers in the generator and discriminator, respectively. Here the discriminator consists of strided convolution layers, batch normalization layers, and LeakyRelu as activation function. It takes a 3x64x64 input image. The generator consists of convolutional-transpose layers, batch normalization layers, and ReLU activations. The output will be a 3x64x64 RGB image. (introduction of paper)

The full connection layer is removed from DCGAN model. The disadvantage of full connection is that there are too many parameters, and overfitting is easy to occur if there are too many parameters. Removing the full connection layer can make the model more stable. In the adjusted model structure, convolutional layer stacking is used for feature extraction and deep image retrieval for feature integration.
In addition, a variety of activation functions are used in the model: ReLU is used for each layer in the generator, and Tanh is used for the output. LeakyReLU is used at all levels of the discriminator.

3.2, Model expansion---- Advanced Task: Controllable Label image generation

On the basis of the optimized model, we expand the model function and achieve the controllable label image generation.

The concept of cGAN model architecture is introduced into the adjusted model, and the input layer and loss function are adjusted, so that the model can generate pictures based on a specific label.



Source: https://salu133445.github.io/dan/background.html

Compared to the basic GAN structure, the difference is that conditions are added to the input of both the generator and discriminator.
In the input to the generator, add conditions to the random noise.
In the input of the discriminator, conditions are added to both the true data and the fake data generated by the generator.

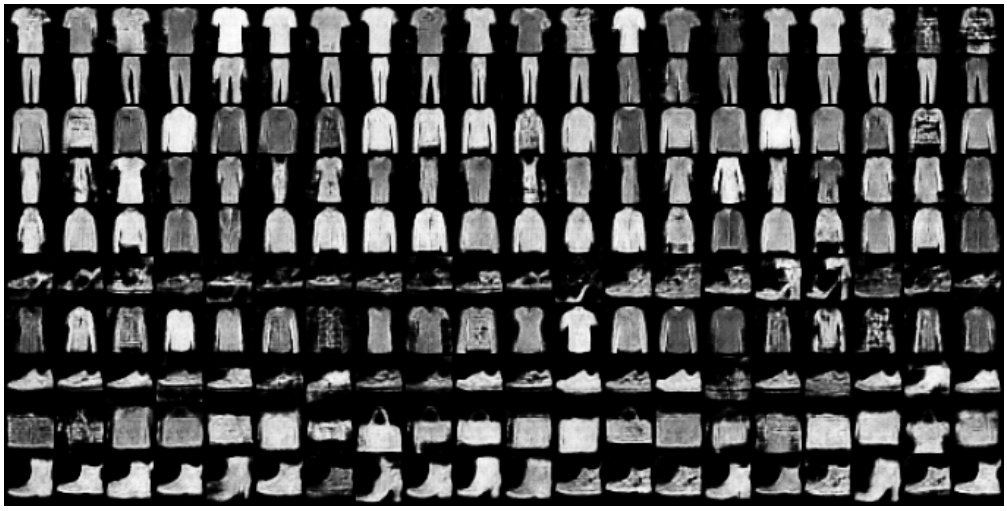
CGAN of loss function generator and the discriminant is corresponding with conditions, namely the type of (x | y) :

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim pdata(x)}[\log D(x|y)] + \mathbb{E}_{z\sim pdata(z)}[\log(1 - D(G(z|y)))]$$

3.3, Performance Analysis

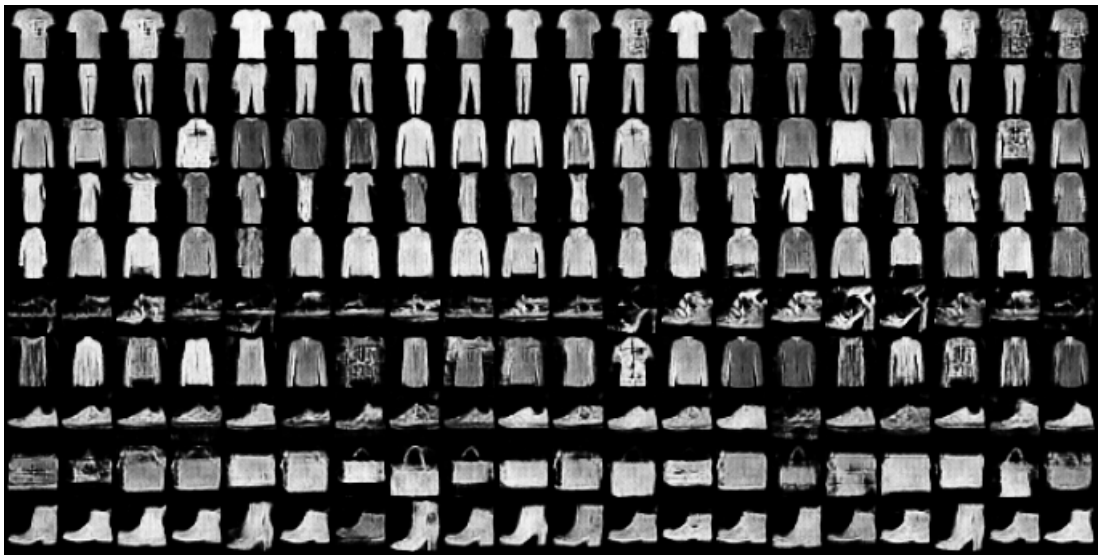
(1) Human judgment of image quality

Epoch 20:



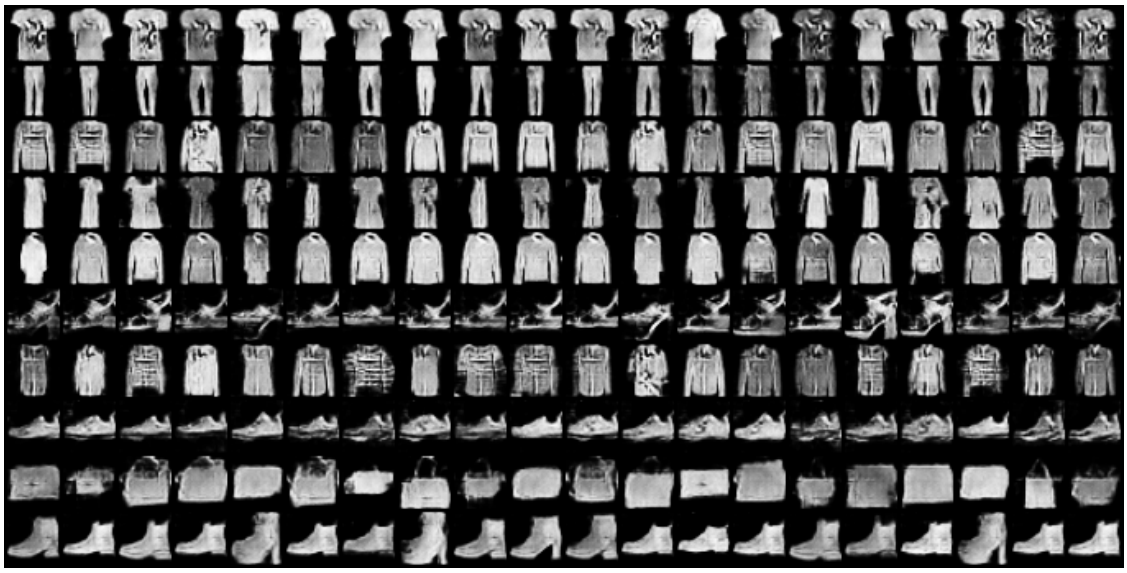
The training of the model is faster than that of GAN model. At epoch 20, the precision of the pictures generated has been higher, and the categories can be directly identified by human eyes. However, some details are still lacking in precision

Epoch 60:



Compared with epoch 20, it has higher precision, better image quality, and more complete and optimized details

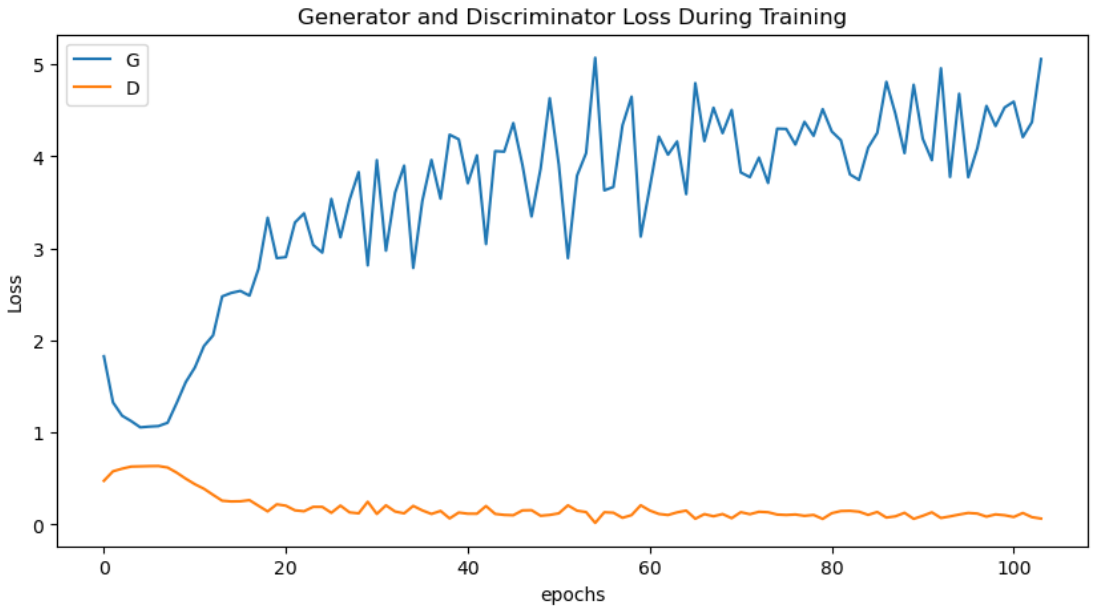
Epoch 100:

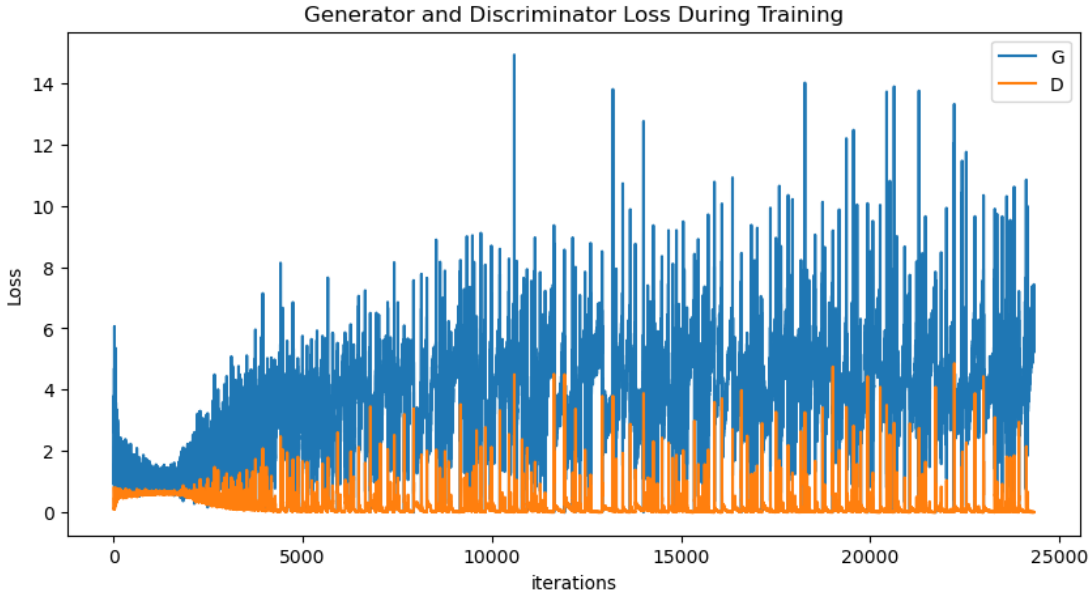


Epoch 100 produces images with fuller pixels than before. However, the differentiation of pictures under the same category has decreased significantly, and some pictures under the label appear abnormal. It indicates that the model has overfitting phenomenon.

(3) Learning loss curve analysis

log is not used to adjust the loss calculation here, so the trend reflected in the end is inconsistent with expectations (the experimental results show that the loss of generator keeps increasing, while the loss of discriminator keeps decreasing). However, it is still possible to analyze the training stage of the model and judge the occurrence of overfitting phenomenon from the perspective of change rate





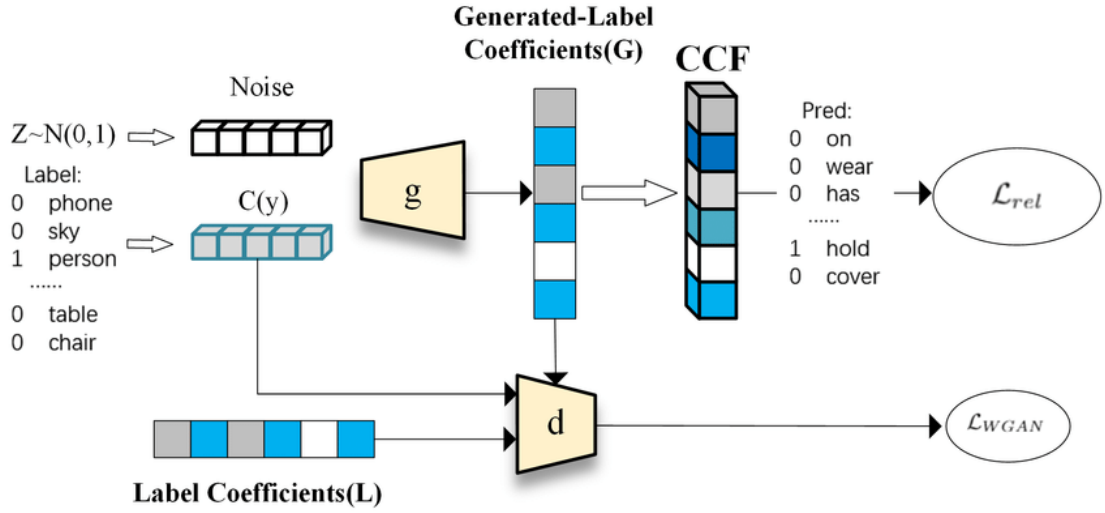
It can be seen that at the position around epoch 60, the loss curves of the generation model and the differentiation model both enter the convergence stage. It can be concluded that the optimal performance of the model occurs in the process of model training. Roughly between epoch 40 and epoch 60 of the model, there is a position where the accuracy of the training model to generate the model reaches a reasonable balance

3. wGAN ---- loss function optimization

One of the problems in the learning process of the previous two GAN models is that the training is unstable, and the discriminator converges shortly after the training, which fails to reach the expected mutual game. The purpose of WGAN is to solve the problem of GAN training instability. To address this problem, we tried to use the method from wGAN to further optimize the DCGAN model.

4.1, model analysis

The wGAN model uses a new loss function based on Wasserstein distance.



Source: https://www.researchgate.net/figure/The-structure-of-our-conditional-WGAN-Our-WGAN-minimizes-the-relationship-predication_fig3_339196399

By removing log from the loss function and Sigmoid from the last layer of the discriminator, you get a fraction in the general sense, not the probability of the original GAN discriminator output.

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.

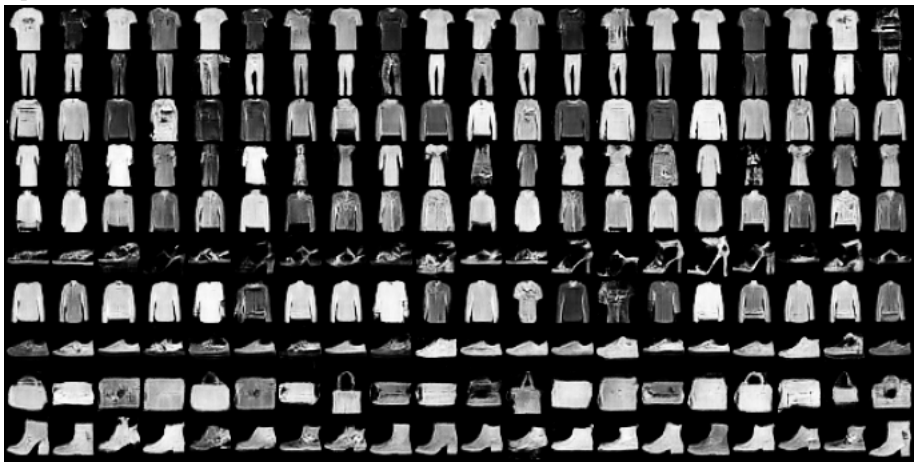
Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

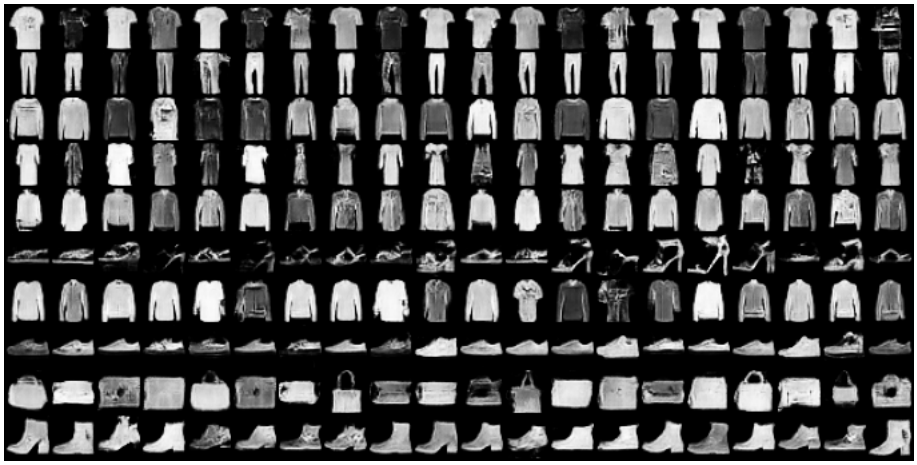
Source: https://arxiv.org/pdf/1701.07875.pdf

4.2, Performance Analysis

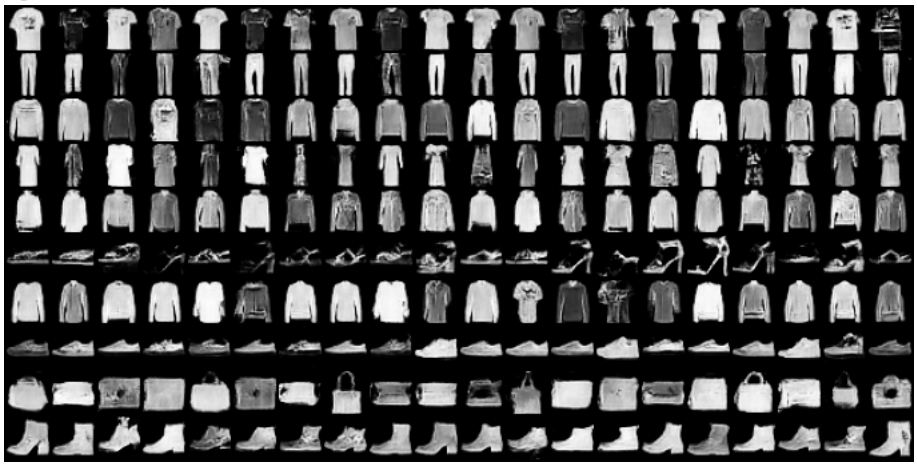
Epoch 20



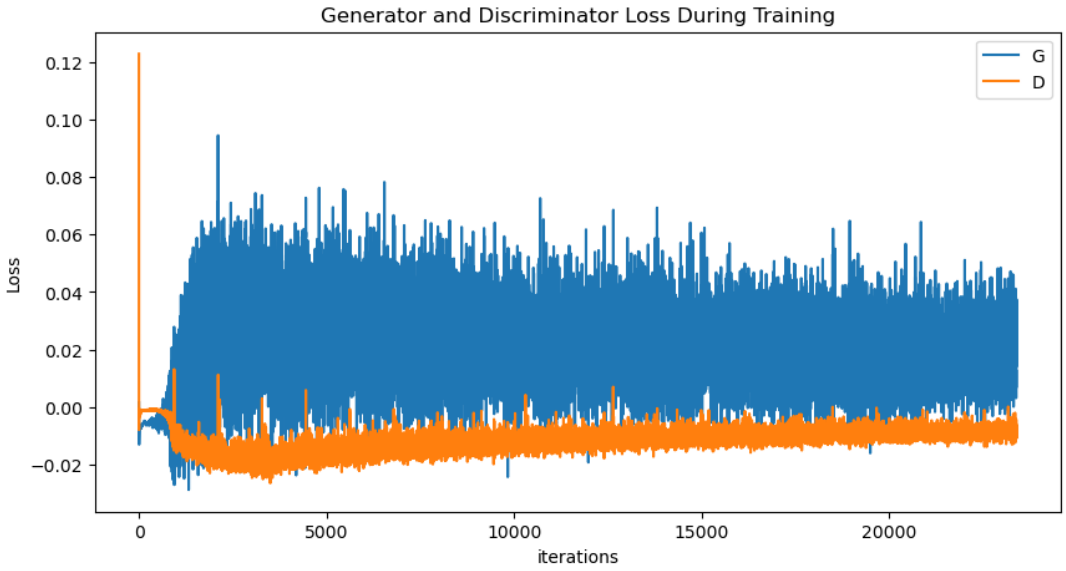
Epoch 60

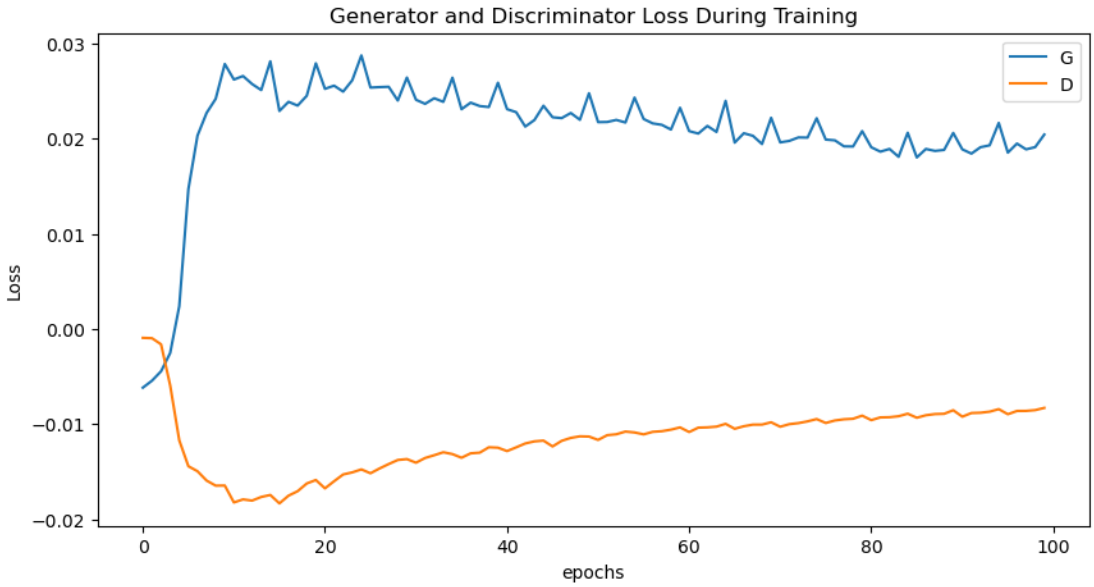


Epoch 100:



Learning loss curse:





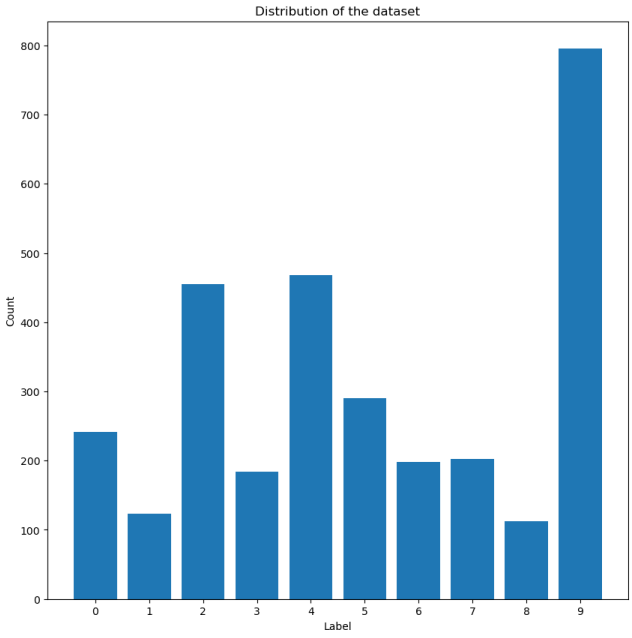
It can be seen that the learning loss curve of the wGAN model is more stable, and the growth trend is consistent with expectations. Moreover, there is no obvious convergence of loss curve and overfitting of the model.

4. ClothingMini Dataset task implementation ---- Model application context extension

In order to test the effect of the model, we further extend the application scenario of the model to 128*128, and generate clothes pictures of color pictures. In this way, we can further analyze the adaptability of the model to domain set and its effectiveness to different image generation tasks

5.1, Dataset visualization

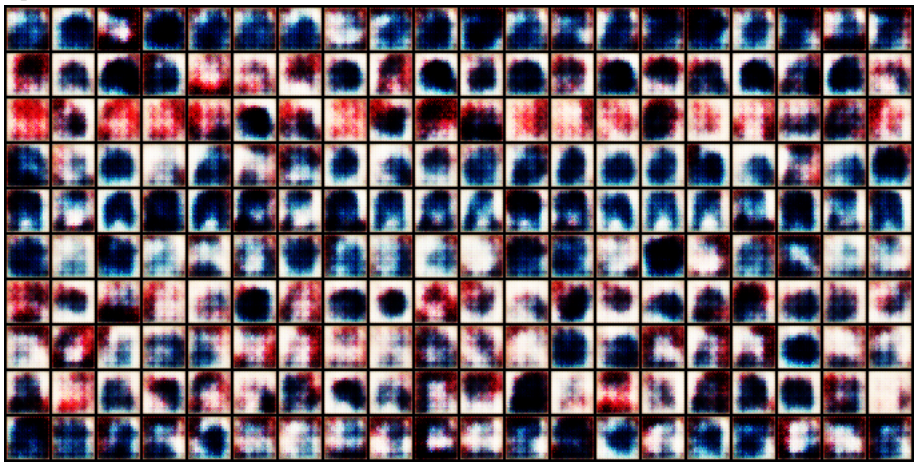
The new data set contains 9 classes. The pictures of the data set and the statistics of the number of each category are as follows



5.2, performance analysis.

(1) image generation

Epoch 5:



Epoch 50:

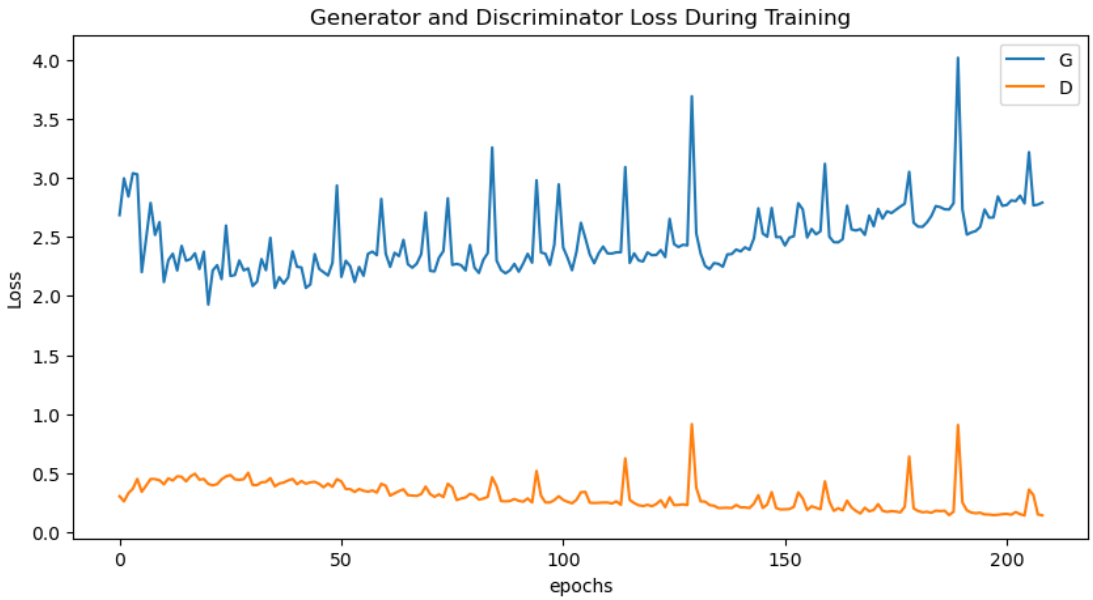
Epoch 100:

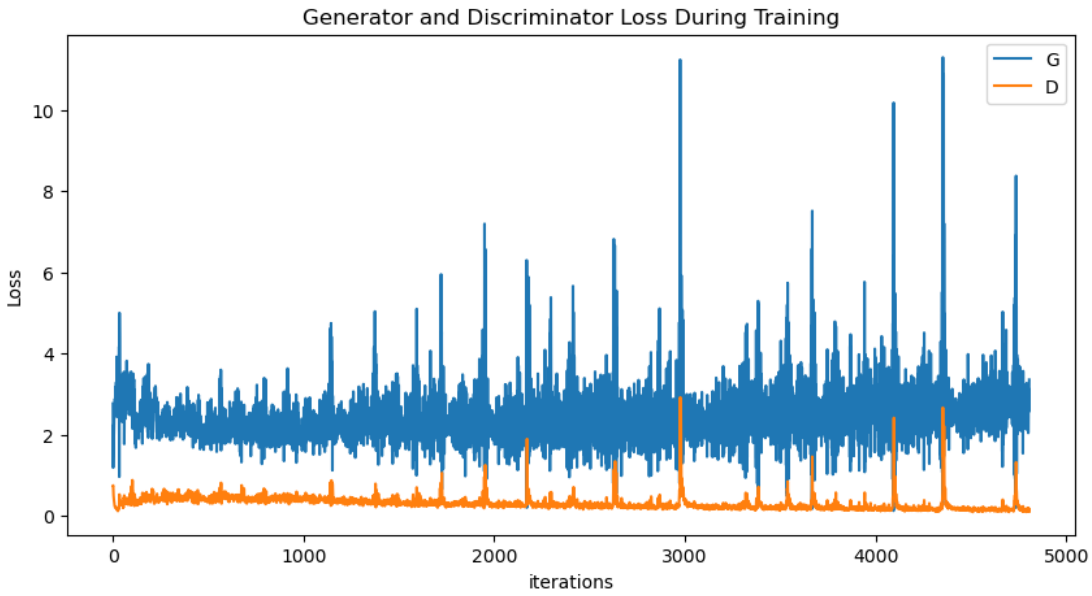
Epoch 150:

Epoch 200:

According to the intuitive observation, the model still performs well in the situation of image generation of color pictures. When epoch 200 is reached, the model has been able to accurately judge the clothing type with good picture quality, and the accuracy of the model is still increasing continuously. What is more interesting is that with the increase of picture scale and the application of color pictures, we can observe more intuitively, and then analyze the model performance more accurately

(2) Learning loss curve





The loss curve of the model has no obvious response to the accuracy. Convergence appears roughly

5. Reference

[1] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs Created Equal? A Large-Scale Study," 2017, doi: 10.48550/arxiv.1711.10337.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," 2015, doi: 10.48550/arxiv.1511.06434.

[3] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," 2014, doi: 10.48550/arxiv.1411.1784.

[4] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, “Generative adversarial networks” 2014, arXiv:1406.2661

[5] P. Kancharla and S. S. Channappayya, "Improving the Visual Quality of Generative Adversarial Network (GAN)-Generated Images Using the Multi-Scale Structural Similarity Index," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018, pp. 3908-3912, doi: 10.1109/ICIP.2018.8451296.

[6] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” 2017, doi: 10.48550/arxiv.1701.07875.