CPSC 8430-HW3

Siqi Zheng

April 2022

Director: Dr. Feng Luo

Generative Adversarial Networks

The Generative Advertising Network was proposed by Ian Goodfellow and his colleagues in June 2014. In the GAN, two neural networks, Generator and Discriminator, contest with each other, continuous optimization to produce results that are closer to reality.

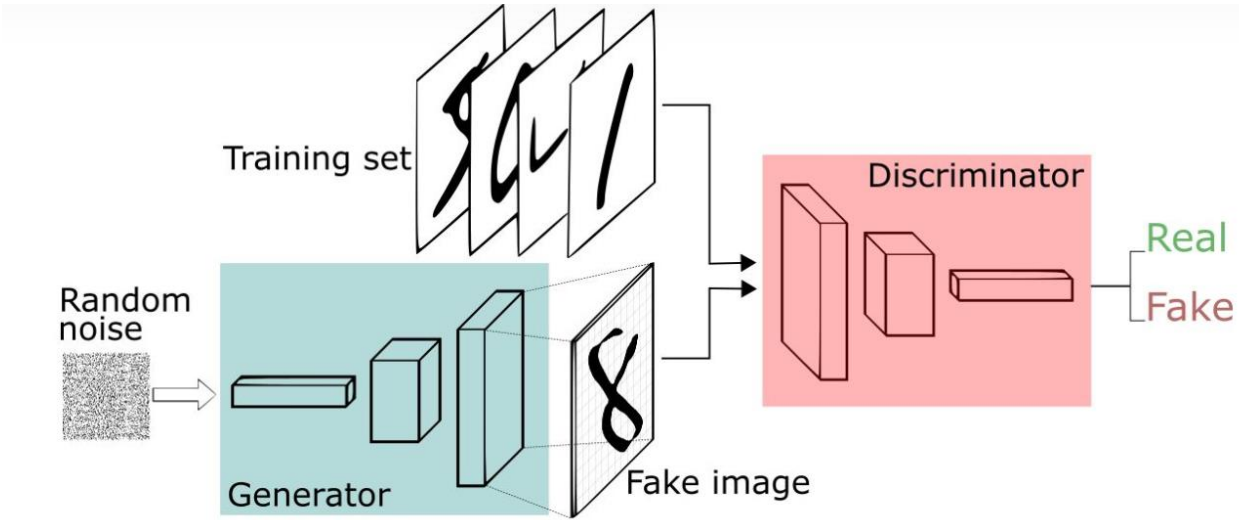

Figure 1. Structure of GAN

Given a dataset, the Generator try to generate a new dataset with the same statistics as the training set. In the meanwhile, the Discriminator try to find the difference between the data generated by the generator and the real data. In the hw3, CIFAR-10 dataset is used as the training set and three technologies, DCGAN, WGAN and ACGAN, are adopted to try to generate the images like the dataset. After that, Frechet Inception Distance (FID) score is used to measure the difference between the generated data and original data and compare three approaches.
To make Generator and Discriminator improve in the training, the following loss function is proposed. In the loss function, it can be found that D (discriminator) tries to maximize the probability it correctly classifies reals and generated data, and G (generator) tries to minimize the possibility that D will correctly predict its output.

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

CIFAR-10 data set

CIFAR-10 data set contains 50000 32× 32 RGB images with 10 different classes. They are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The figure 2 shows the samples of 10 classes of images. CIFAR-10 dataset can be derived directly from the torchvision datasets and put into dataloader for further use.

FID score is used to measure the difference between the real image dataset and generated image dataset. The library is from the GaParmar Github. A 128-images group derived from the CIFAR-10 and a 128-images group derived from the GAN generator are put into "compute_fid" function in this library and compute the FID value.

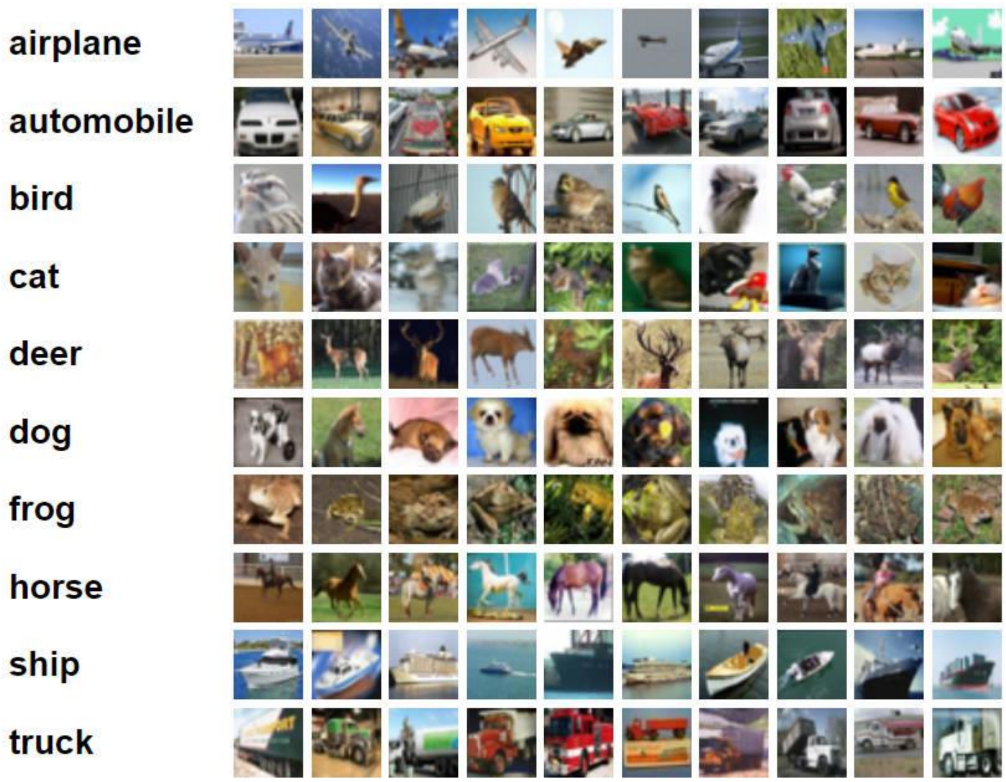

Figure 2. CIFAR-10 data set

DCGAN

Compared to GAN, DCGAN has the same structure of the GAN except the model of the Generator and Discriminator. Convolutional and convolutional transpose layers are added to the Generator and Discriminator to have a better performance. The structure of the G model and D model draw on the structure from the PyTorch official website tutorial.

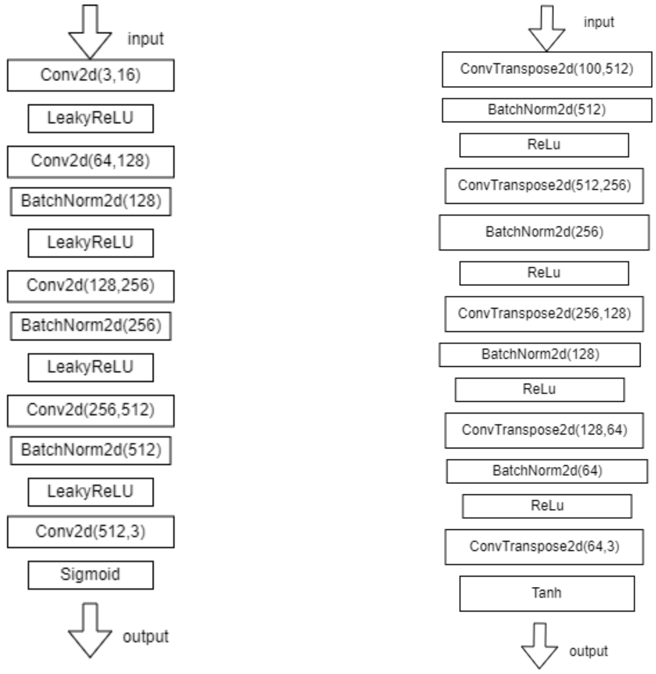

Figure 3. Generator and Discriminator structure (left G, right D)

Due to the limitation of the hardware, the whole trianing process runs 25 peoch. The hyper-parameter and result of the DCGAN using CIFAR-10 training dataset is shown in the Tabel 1.

Tabel 1.
DCGAN hyper parameter and result

Network frame	DCGAN	Optimizer	Adam
Batch size	128	Epoch	25
Learning rate	2e-4	FID score	75.217

The following shows the 10 best figure generated from the DCGAN. From the figure, it can be found that the image has a blurry shape, and it can be speculated that as the training epoch increases, an image similar to the training dataset can eventually be generated.

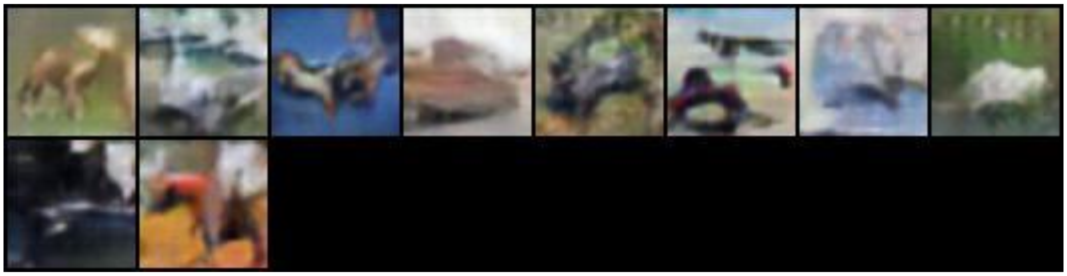

Figure 3. DCGAN fake image

The following shows the set of pictures generated from the first generation, 10th generation and 25th generation.

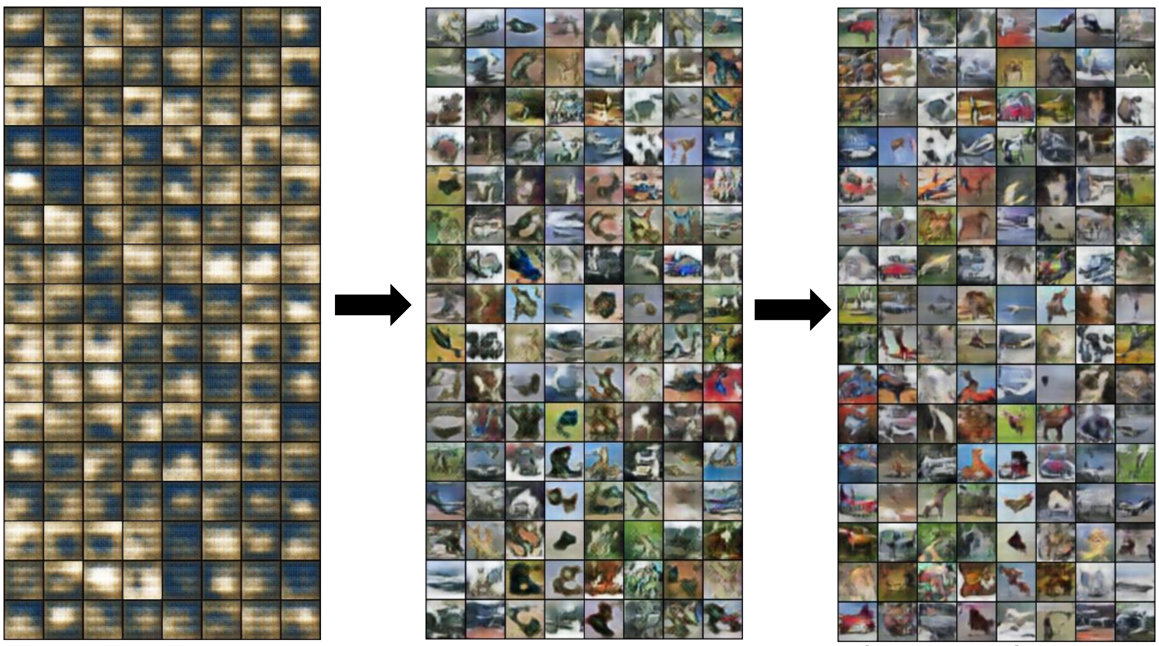

Figure 4. Generator evolution (From left to right: first epoch, 10th epoch, 25th epoch)

WGAN

In the normal GAN, the loss function mentioned in the previous Eqns. It can be proved to minimize the Jensen-Shanon (JS) divergence between the training dataset and generated dataset. However, as the dimension of the data increase, the distribution of training dataset and generated dataset are not overlapped. Which means that JS divergence is always the same value if two distributions don’t overlap, even if some of the generated dataset is closer to the real dataset, there will be no difference. In order to deal with this problem, the Wasserstein GAN is proposed. It measures the difference between two distributions using Earth Mover’s Distance. The following is the WGAN loss function.

$$V(G,D) = max_{D\in 1-Lipschitz}\{\mathrm{E}_{x \sim p_{data}}[D(x)] + \mathrm{E}_{x \sim p_G}[D(x)]\}$$

To enforce the Lipschitz constraint, a gradient penalty is added to the loss function which derive the improved WGAN. The following is the equation of the loss function of WGAN-GP.

$$V(G,D) = max_{D\in 1-Lipschitz}\{E_{x \sim p_{data}}[D(x)] + E_{x \sim p_G}[D(x)]\} - \lambda E_{x \sim p_{penalty}}(\left|\left|\nabla_x D(x)\right|\right| - 1)^2$$

The whole trianing process runs 25 peoch. The hyper-parameter and result of the WGAN using CIFAR-10 training dataset is shown in the Tabel 2.

Tabel 2.
WGAN hyper parameter and result

Network frame	WGAN-GP	Optimizer	Adam
Batch size	128	Epoch	25
Learning rate	2e-4	FID score	200.236

The following shows the 10 best figure generated from the WGAN.

Figure 5. WGAN fake image

ACGAN

A conditional GAN or CGAN, is an extension of the GAN architecture that adds structure to the latent space. In the CGAN, the input of the Generator consists of two parts: a point in the latent space and a class label. The generator knows which class it needs to generate. Similarly, the discriminator is also provided with an image and a class label. It should classify if the image is real or not. In the figure 6, the left side of the figure describe the structure of the CGAN. This process makes the GAN classify the class more reasonable if the dataset has more than one class.

The Auxiliary Classifier GAN (ACGAN), proposed in 2016 by Augustus Odena, is the architecture used in this report. Compared to the CGAN, it only provided with the image as the input to the Discriminator and the Discriminator need to predict the class label of the image. In the figure 6, the right side of the figure describe the structure of the ACGAN. The effect of changing the conditional GAN in this way is both a more stable training process and the ability of the model to generate higher quality images with a larger size than had been previously possible.

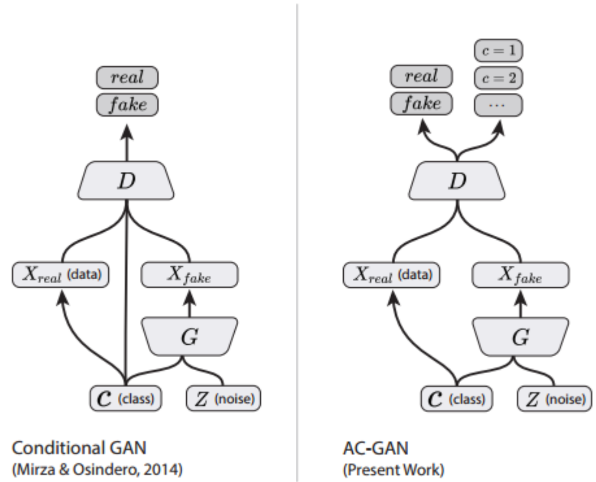

Figure 6. structure of the CGAN and ACGAN

The loss function of the ACGAN has two part: the log likelihood of the correct source and the log likelihood of the correct class. The whole trianing process runs 25 peoch. The hyper-parameter and result of the ACGAN using CIFAR-10 training dataset is shown in the Tabel 2.

$$L_s = E[logP(S = real|X_{real})] + E[logP(S = fake|X_{fake})]$$
$$L_c = E[logP(C = c|X_{real})] + E[logP(C = c|X_{fake})]$$

Tabel 3.

ACGAN hyper parameter and result

Network frame	ACGAN	Optimizer	Adam
Batch size	100	Epoch	25
Learning rate	2e-4	FID score	196.63

The following shows the 10 best figure generated from the ACGAN. Even if it only trained 25 epoch. Some of the figure can be recognized such as the first image is a bird.

Figure 7. 10 best figure generated from the ACGAN

Conclusion

Compared three GAN, DCGAN has the lowest FID score which means its generated dataset is most similart to the real data. Due to the limitation of the hardware, all the experiments run only 25 epoch. So it can’t get the best result. Due to the mature architecture of DCGAN, it is easier to obtain better results. In contrast, the hyperparameters of WGAN and ACGAN require further fine-tuning to achieve desirable results. In all, DCGAN has the best result among these three GAN architectures.