Zhi Fang Tan

CPSC 881 Homework 3

Introduction

In this assignment, two different generative adversarial network(GAN) models are trained on the CIFAR10 dataset. The chosen GAN architectures are DCGAN and WGAN-GP. GAN model generally consists of two major parts which are the generator and discriminator. The objectives of these two parts of the model are different as the generator is trying to generate an image to be real enough to fool the discriminator whereas the discriminator is trying to sort out the generated images from the real input images.

DCGAN and WGAN-GP network structure

<u>Generator</u>

Noise(100) -> Dense(2*2*512)->reshape(2,2,512)->BN->LeakyRelu->
conv2dTranspose(filter=256,kernel=(5,5),strides=2)->BN->LeakyRelu->
conv2dTranspose(filter=128,kernel=(5,5),strides=2)->BN->LeakyRelu->
conv2dTranspose(filter=64,kernel=(5,5),strides=2)->BN->LeakyRelu->
conv2dTranspose(filter=3,kernel=(5,5),strides=2)->tanh->output

<u>Discriminator</u>

Image(32,32,3)->conv2d(filter=64,kernel=(5,5),strides=2)->LeakyRelu->Dropout(0.5)->
conv2d(filter=128,kernel=(5,5),strides=2)->BN->LeakyRelu->Dropout(0.5)->
conv2d(filter=256,kernel=(5,5),strides=2)->BN->LeakyRelu->
conv2d(filter=512,kernel=(5,5),strides=2)->BN->LeakyRelu->Flatten->Dense(1)->Output

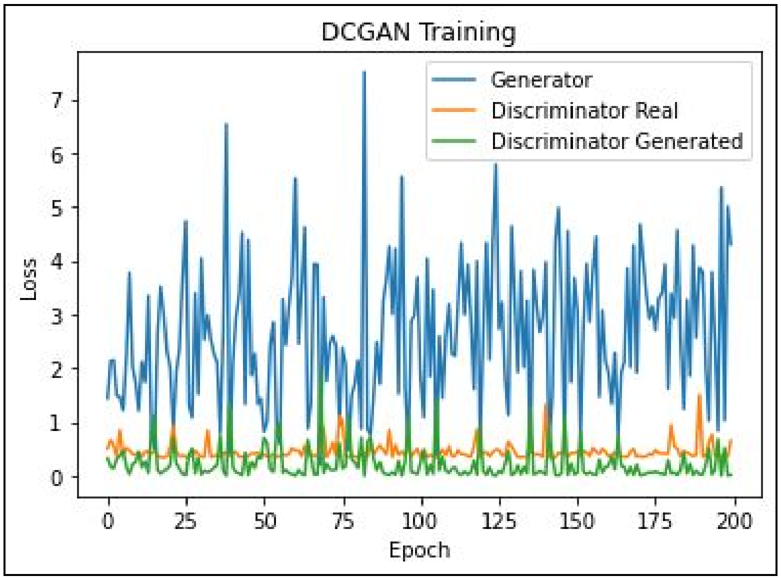
DCGAN, WGAN, and WGAN-GP

WGAN [2] which is based on DCGAN[1] is proposed in 2017 to further improve the stability of the training process. Instead of minimizing the Jenson-Shannon divergence between the real data and generated data probability, the Wasserstein loss is minimized. In terms of the actual implementation, the sigmoid activation function in discriminator is removed and weights of the network are clipped between certain values such as -0.01 and 0.01. The proposed Wasserstein loss is able to measure the distance between probabilities in a better way as JS divergence does not provide great feedback on the distance between probabilities. When two probabilities are not overlapped, the loss JS divergence between them will be log2. Once the probabilities overlap with each other, the JS divergence will immediately jump to zero. This sudden jump sometimes causes instability in the training of the network. Another improved version which is WGAN-GP[3] is used in this assignment. [3] shows that most of the weights actually gathered around -0.01 and 0.01 when the weights are clipped between these values. This could easily lead to vanishing gradient or exploding gradient during training. Therefore, the author proposed a gradient penalty to solve the mentioned problem. Unlike WGAN which uses RMSProp to train the network, WGAN-GP uses Adam optimizer, same as DCGAN. Compared to the original WGAN, WGAN-GP can converge at a faster rate and have better stability in the training process. From [2] and [3], we can see that the main improvement of both WGAN and

WGAN-GP over the original DCGAN is not the performance of the network, but rather the stability of the network. From the results in this assignment, DCGAN has overall results than WGAN-GP which follows the trend from the original paper.

Results

DCGAN Training Curve



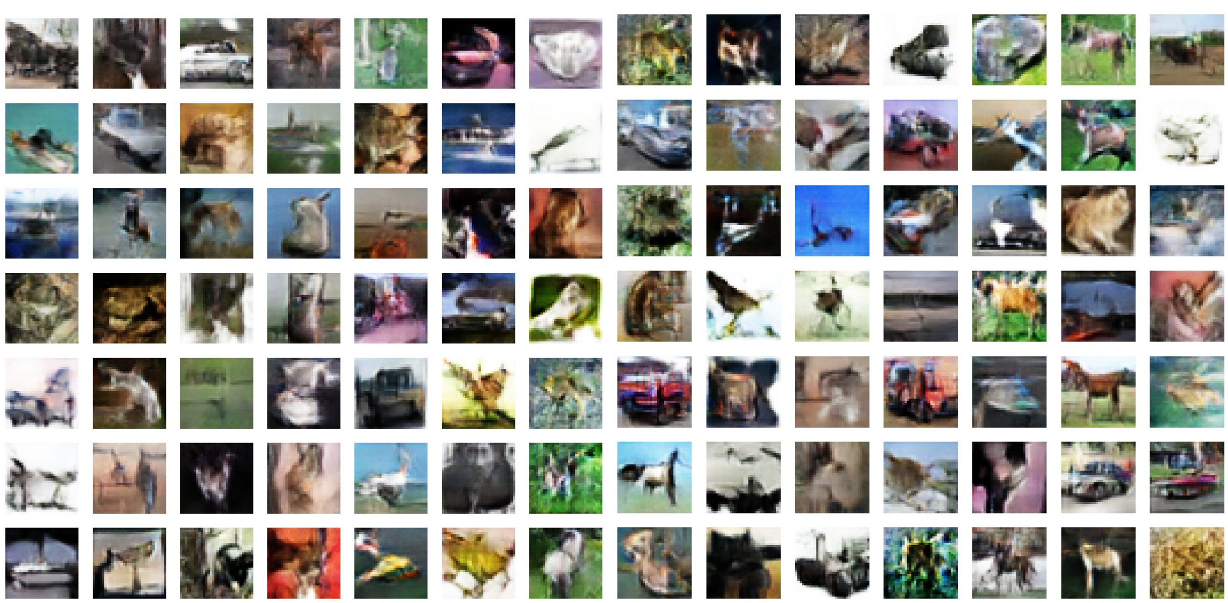
Hyperparameters

Learning Rate: 0.0002

Epoch: 200

Batch Size: 64

DCGAN Generated Output



Zhi Fang Tan

Best DCGAN Images

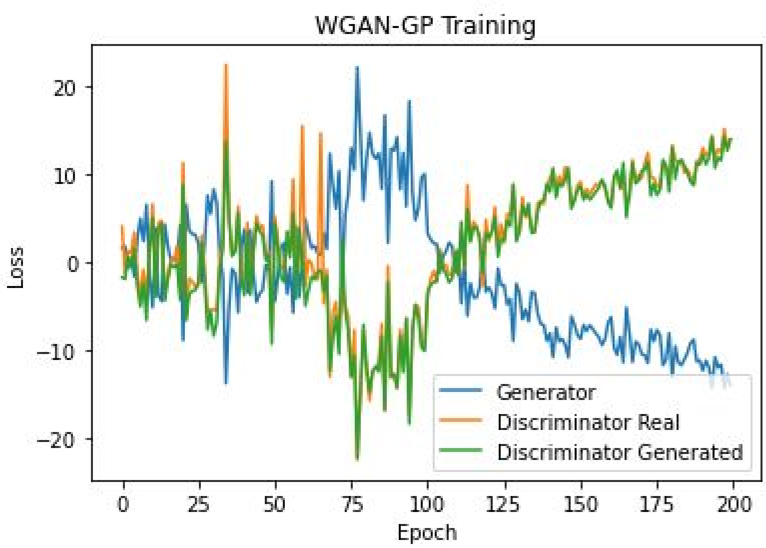
DCGAN IS Score:

Ave: 6.494 Std: 0.329

DCGAN FID Score:

42.229

WGAN-GP Training Curve



<u>Hyperparameters</u>

Learning Rate: 0.0002

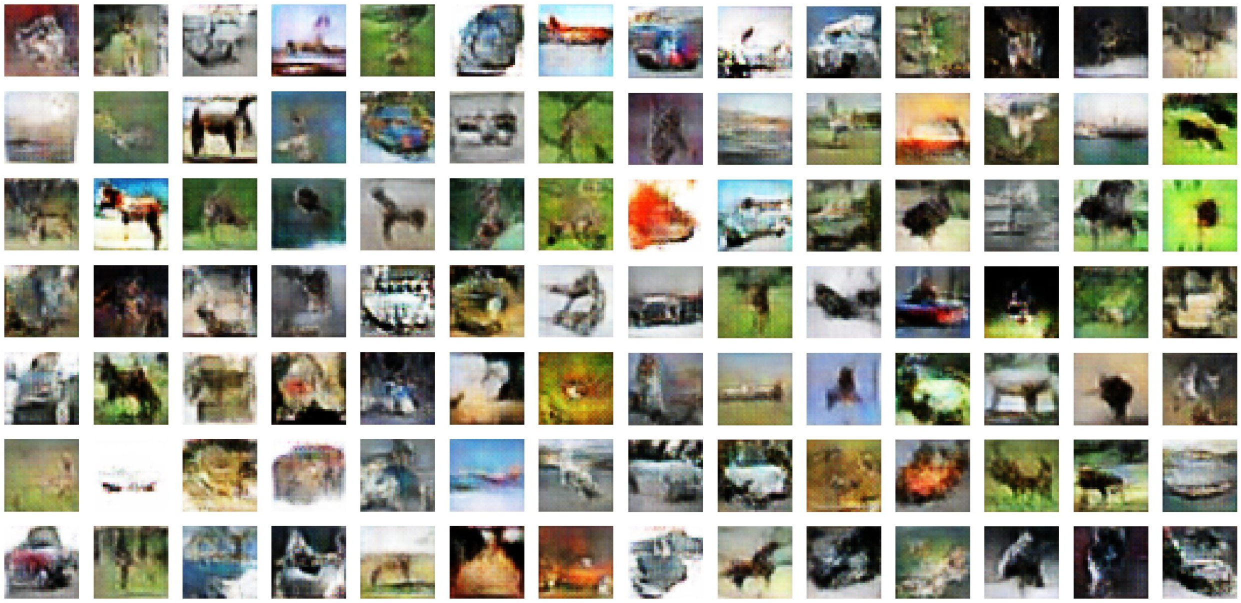
Epoch:200

Batch Size: 64

Times to train critic per batch: 2

Zhi Fang Tan

WGAN-GP Generated Output



Best WGAN-GP Images

WGAN-GP IS Score:
Ave: 4.608 Std: 0.207
WGAN-GP FID Score:
92.718

Conclusion

WGAN-GP seems to need more training iteration and train the critic more times per batch. In this assignment, WGAN-GP model is only trained for 200 epochs and the critic is trained twice for every batch whereas in the original implementation it is trained 5 times per batch due to the extensive amount of time with higher critic training times. Overall, DCGAN is able to produce decent images. However, further tuning and training with longer epoch can be done to further improve the images.

Zhi Fang Tan

Reference

[1] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017.

[3] I. Gulrajani, F. Ahmed, M. Arjvsky, V. Dumoulin, and A.Courville. Improved Training of Wasserstein GANs. arXiv preprint arXiv:1704.00028, 2017.