Blake Washburn
Dr. Feng Luo
Deep Learning – CPSC 8430
24 April 2021

# Homework 3 Report

The files contained in my repository implement the DCGAN, WGAN, and ACGAN networks. Pytorch was the main library used in these files. These networks are composed of a generator, which generates fake images, and a discriminator, which determines if an image is real or fake. All three networks draw from the CIFAR10 dataset for training. Overall, I was disappointed with the quality of the fake images generated from all networks. At the beginning of the project, I looked online to see how effective other developers were in generating fake images that looked real. I assumed that I would be able to mimic their impressive performance to a tee. However, that proved more difficult than expected. Even after training my network for 120+ epochs, my results were still less than perfect.

To give a precise measurement of the network's performance, a quantity called the FID score was provided to students in a github repository. This library produces an integer value that determines how effective the generator of the network was in creating images that appeared to be real. Unfortunately, I did not have time to add this tool to my files. I spent more time than I should have on the implementation of the networks rather than the addition of this important tool. I thought it would be a simple function call, and as such, I did not think it would be difficult to add to my program after its implementation was complete. However, I realized too late that the FID score consisted of an entire library. I then ran into issues downloading and using the library. As a result, it is not included in my implementation and I cannot provide a numerical value that represents the performance of my networks.

In the following sections, I provide some of the resulting images produced by my networks, and give graphs showing their loss values during training. All networks were trained for 135 epochs.

The resulting images from my DCGAN network are provided in Figure 1. DCGAN proved to be the easiest to implement out of the three networks. However, it was also a more unstable network with wildly varying loss values from epoch to epoch. The loss values regularly fluctuate between 0 and 40. Furthermore, the images produced by DCGAN are the least believable when compared to WGAN and ACGAN. The shapes produced by my DCGAN network are not clear or vibrant and look the least like the real images from the CIFAR10 dataset.

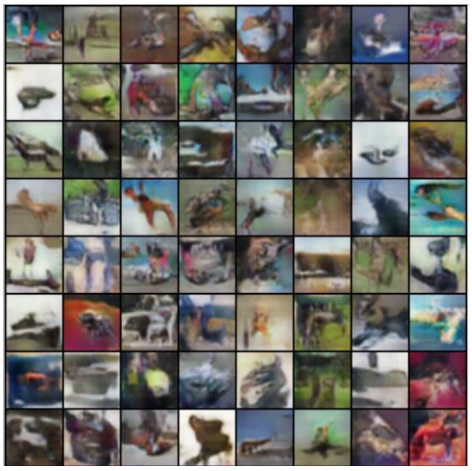

Figure 1: Sample images produced by DCGAN

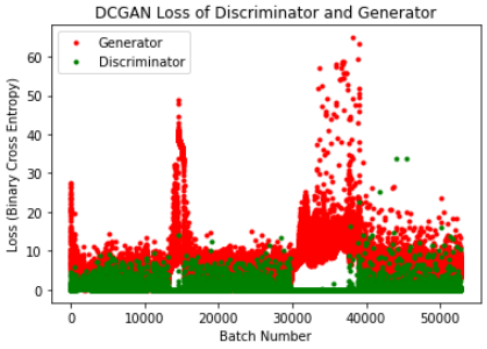


Figure 2: Loss values during DCGAN training

Some sample images from my WGAN network are shown in Figure 3. My WGAN network was much more stable than the DCGAN network, as can be seen in Figure 4. Loss values only fluctuate between 0.5 and -1.5. I intended to go back and improve WGAN by using gradient penalty, but I ran out of time before this could be completed. However, the images produced by WGAN are a bit clearer than DCGAN's images.

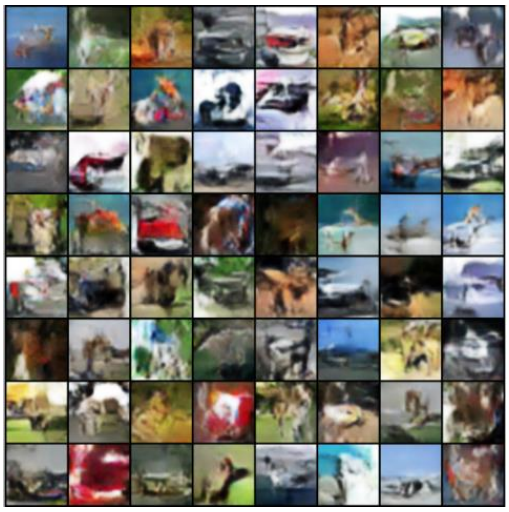

Figure 3: Sample images produced by WGAN

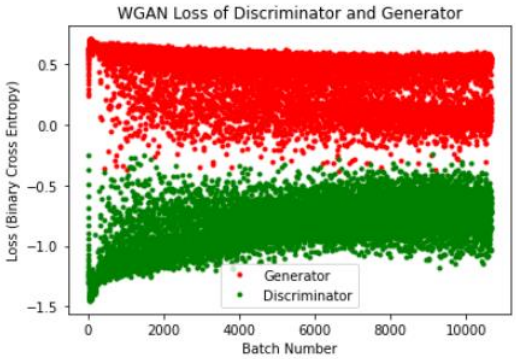


Figure 4: Loss values during WGAN training

The ACGAN network was the most difficult to implement for me personally. The need to keep track of image labels and perform embedding complicated the process quite a bit. Some of the images produced from my ACGAN network are available in Figures 5 and 6. However, I believe that these images are the best produced by any of the networks that I implemented. It is possible in Figures 5 and 6 to distinguish cars, planes, and boats in these images without too much of a stretch of the imagination. If I had to select my best 10 images, I would pick them from the following two figures below.

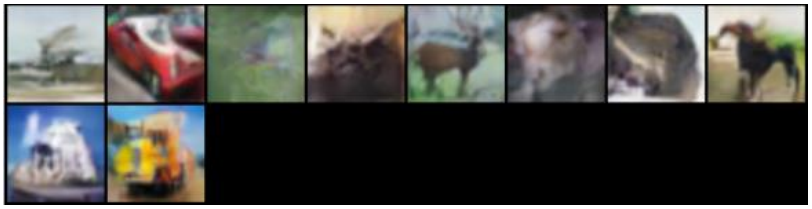

Figure 5: Sample images produced by ACGAN

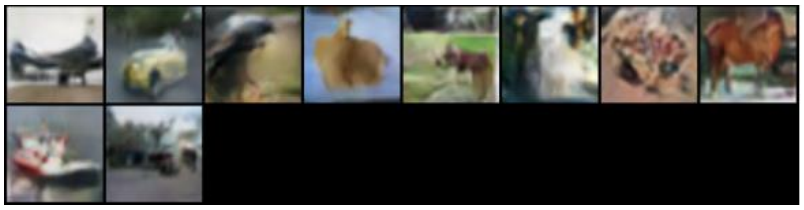

Figure 6: Sample images produced by ACGAN

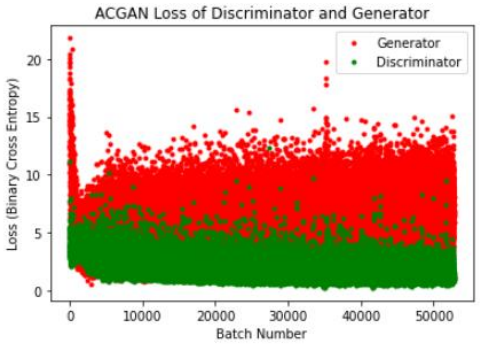


Figure 7: Loss values during ACGAN training