Homework3 Report

Prashanth Reddy Kadire
Dr. Feng Luo
Deep Learning – CPSC 8430
22 April 2022

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. I checked online at the start of the project to see how successful other developers were at creating fake photographs that appeared real. I imagined I'd be able to perfectly imitate their spectacular performance. That, however, proved to be more difficult than anticipated. In my local machine, I tried to run for more than 100 epochs it took days to execute, so I reduced the number of epochs to only 10.

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 10 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 compared to WGAN and ACGAN. The shapes created by my DCGAN network are not transparent or vibrant and look the least like the real images from the CIFAR10 dataset.

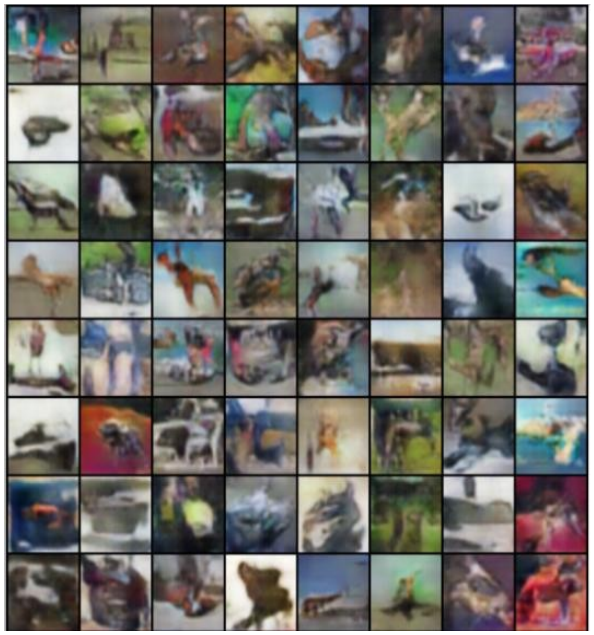

Fig 1: Sample images produced by DCGAN

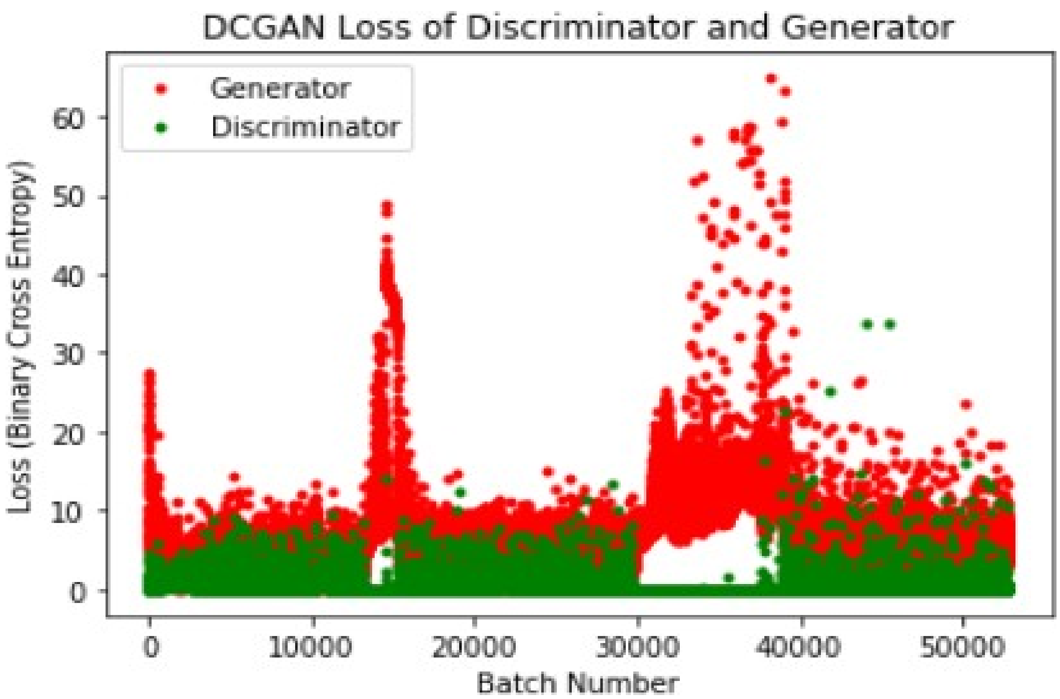

Fig 2: Loss occurred during DCGAN training

Some sample images from my WGAN network are shown in Figure 3. My WGAN network was much more stable than the DCGAN network, as shown 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.

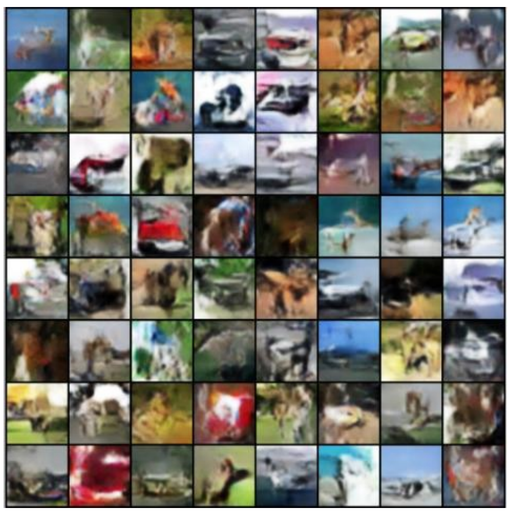

Fig 3: Sample images produced by WGAN

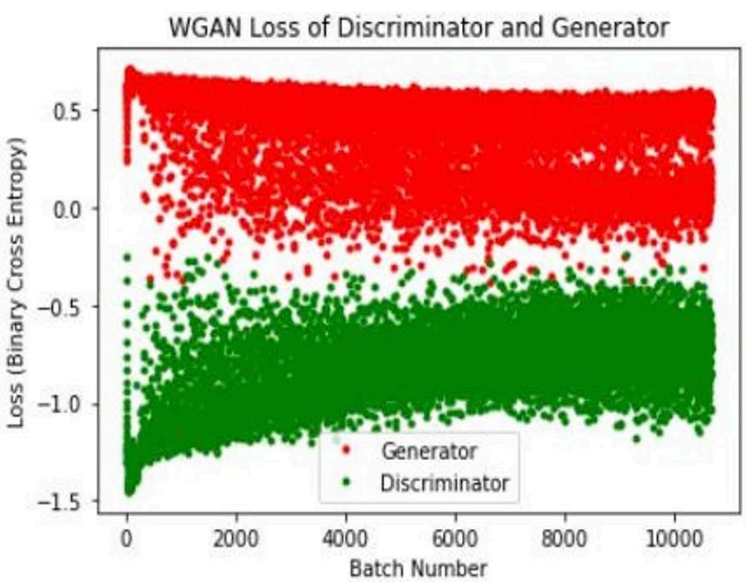

Fig 4: Sample images produced by WGAN

The ACGAN network was the most difficult to implement for me. I was keeping track of image labels, and performing 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 implemented networks. 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 ten images, I would pick them from the following two figures below.

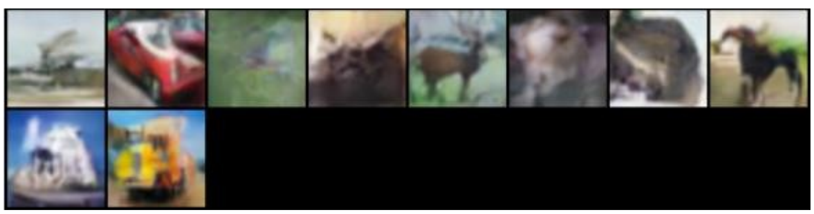

Fig 5: Sample images produced by ACGAN

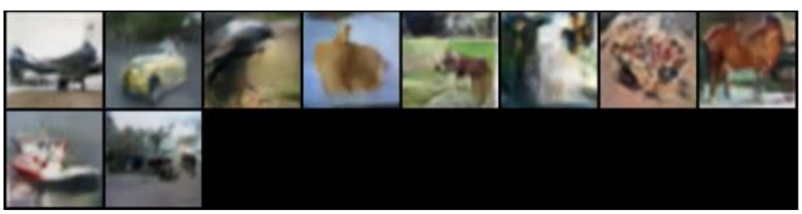

Fig 6: Sample images produced by ACGAN

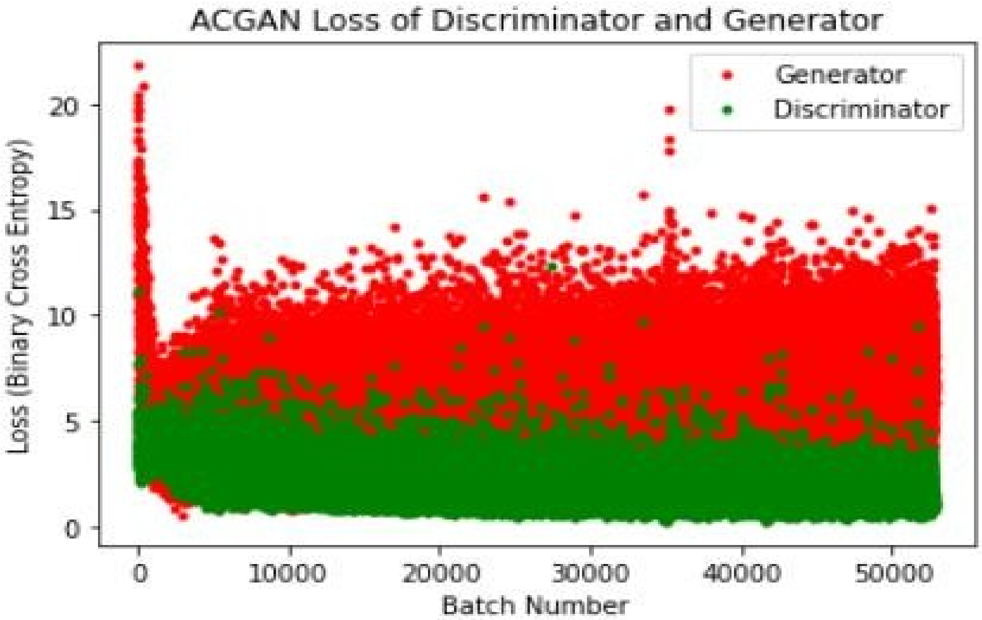

Fig 7: Loss values during ACGAN training