Module 8: Portfolio Project

Working with a Generative Adversarial Network: Network Performance

Nolan Byrnes

CSC580 – Applying Machine Learning and Neural Networks - Capstone

Colorado State University – Global Campus

Dr. Isaac Gang

November 3rd, 2023

Module 8: Portfolio Project

I developed and trained a Generative Adversarial Network (GAN) so that the generator can generate images of cats. The dataset I used for this project was the CIFAR10 small images classification dataset, filtered to only include images that had the label of “3”, which represented the cat images of the dataset. This paper provides an analysis of the results of training the developed GAN.

There are challenges when it comes to evaluating the performance of GAN models. Although other deep learning models can train until convergence using a loss function, a GAN generator model “is trained using a second model called a discriminator that learns to classify images as real or generated” (Brownlee, 2019, para. 1). Brownlee explains that due to the lack of an objective loss function, “models must be evaluated using the quality of the generated synthetic images” and there is “no generally agreed upon way of evaluating a given GAN generator model” (Brownlee, 2019, para. 6, 12). Due to this, evaluating the results manually and obtaining user feedback on the images can help assess the performance of the model. For evaluating the Cat GAN, I had the generator create a sample of 49 images of cats after the first epoch, and at every 10 epochs, and evaluated the results myself, as well as requested user feedback from my wife to determine the effectiveness of the trained generator model.

To evaluate the samples generated by the generator model, it is important to observe the original images that the model is being trained on, to help give a baseline of what we could expect out of the trained model. I did notice that with the real images displayed in figure 1 below, I could tell that the images contained cats, but the images did seem to look a little blurry, which can be explained by the images having a height and width of 32 pixels.

The original images also had the cats in various poses, with a variety of backgrounds from being in an outdoor or indoor environment, with some of the images being closeups, containing no background at all. A sample of 49 real images from the training dataset can be observed in figure 1 below.

Figure 1.

Real Samples

A collage of cats

Description automatically generated

Note. This figure displays 49 cat images from the CIFAR10 Dataset that were used during training.

After the first completed epoch, the images after the first epoch were seemly just grey squares throughout each of the 49 generated images, with no variability in the output. When taking a closer look at the images, a grid-like pattern can be observed. Within my model, I used the Conv2DTranspose layer which contains a kernel, and a stride input parameter. GeeksForGeeks explains that the “transposed convolutional layer slides the input over the kernel and performs element-wise multiplication and summation” (GeeksForGeeks, 2023, para. 3). Odena et. al. explains that “deconvolution has uneven overlap when the kernel size (the output window size) is not divisible by the stride (the spacing between points on the top)” (Odena et. al., 2016, para. 2). Due to using Conv2DTranspose layers, with a 3-by-3 kernel and a 2-by-2 stride, the grey grid could have been an artifact of the generator model applying the Conv2DTranspose layers with the initial weights and what it gained from the first epoch. The sample images from the first epoch can be observed in figure 2 below.

Figure 2.

Epoch 1 Results

A group of squares in rows

Description automatically generated

Note. This figure displays 49 sample images that the generator model created after the first epoch.

After 250 epochs, I noticed that the images resembled more like cats, and were no longer grey boxes. I thought that the diversity in the data was high, unlike after the first epoch. I did not see a pattern of the same image being generated. I asked my wife to provide feedback on the images populated, and she commented that the images resembled more like animals, and mentioned that she saw a guinea pig, hamster, and “maybe a cat”, within the images in figure 3.

Figure 3.

Epoch 250 Results

A collage of images of cats

Description automatically generated

Note. This figure displays 49 sample images that the generator model created after 250 epochs.

Observing the results from the 500th epoch, I noticed that the images were resembling cats in different poses. When I asked my wife for user feedback on the images of the 500th epoch, she mentioned that she saw cats in different poses. Due to the cats having a high degree of variability in the poses, I do not think that there is mode collapse at this point, which is where “the generator can only produce a single type of output or a small set of outputs” (Shen, 2022, para. 8).

The images also have a variety of backgrounds, where in some of the images, we could see green, which could resemble grass, whereas others have a black background, or tan walls, resembling painted walls. The images produced by the generator after 500 epochs of training can be viewed in figure 4 below.

Figure 4.

Epoch 500 Results

A collage of squares of different colors

Description automatically generated

Note. This figure displays 49 sample images that the generator model created after 500 epochs.

Conclusion

Evaluating a GAN model can be a challenge due to not having an objective loss function unlike other machine learning models. Some ways to evaluate the performance include inspecting results manually and by receiving user feedback. Observing the real images that the GAN model is important, as it gives you a baseline of what you can expect the generator to create after training. After the first epoch, the images were mostly grey, with a grid-like pattern, which could be due to the nature of the deconvolutional layers that the generator had with the initial weights and what it had gained off the first epoch, with the convolutional layer having an uneven overlap. From the user feedback I obtained, after the 250th epoch, it was observed that some of the images resembled animals, and by the 500th epoch, it was noted that cats could be seen within the images. I noticed that after the 500th epoch, there was a variety of cat poses and backgrounds, so at that point I did not observe signs of mode collapse. After 500 epochs, the generator was able to create images that resembled cats, which was the goal of the generator to perform.

**REFERENCES**

Brownlee, J. (2019, July 12). How to Evaluate Generative Adversarial Networks. Machine Learning Mastery. <https://machinelearningmastery.com/how-to-evaluate-generative-adversarial-networks/>

GeeksforGeeks. (2023, January 24). What is transposed Convolutional layer? <https://www.geeksforgeeks.org/what-is-transposed-convolutional-layer/>

Odena, A., Dumoulin, V., & Olah, C. (2016, October 17). Deconvolution and checkerboard artifacts. Distill. <https://distill.pub/2016/deconv-checkerboard/>

Shen, K. (2023, October 30). Measuring Mode Collapse in GANs Using Weights & Biases. Weights & biases. <https://wandb.ai/authors/DCGAN-ndb-test/reports/Measuring-Mode-Collapse-in-GANs-Using-Weights-Biases--VmlldzoxNzg5MDk>