Portfolio Project - Program Analysis

Karl Estes

Colorado State University Global

CSC 580: Applying Neural Networks and Machine Learning

Dr. Brian Holbert

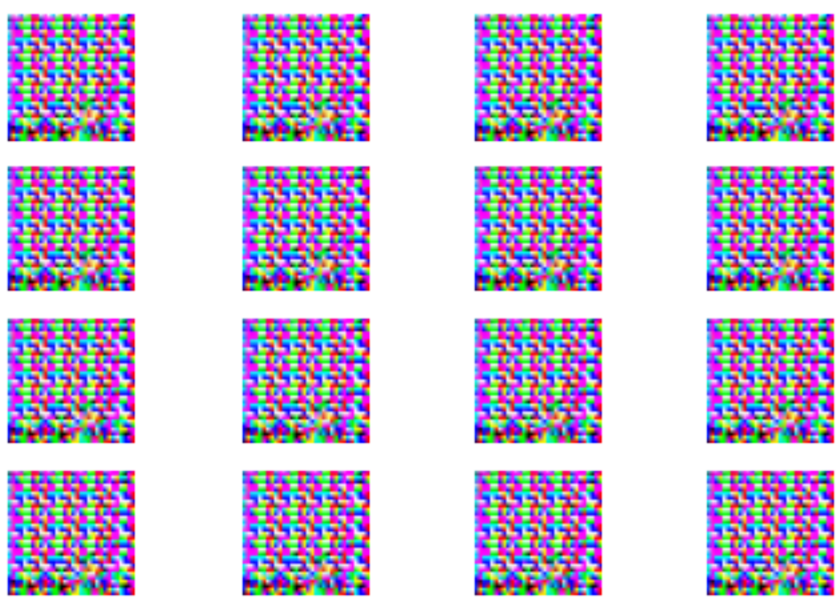
April 10th, 2022

Portfolio Project - Program Analysis

Per the assignment instructions, the provided code was run with slight modification for saving images created by the Generator at various steps in the training process. While the GAN was able to run without errors, repeated attempts resulted in the output of randomly colored pixels, many of which appeared remarkably similar with no form of any sort. It did not matter which of the CIFAR10 classes the network was run on; it continually produced images like the ones in Figure 1.

I could not determine the problem that was causing the training issue, so I began by using code from Brownlee's (2019) tutorial since it was focused on creating and training a GAN for the CIFAR10 dataset. The training algorithm implemented mini batch learning across the entire

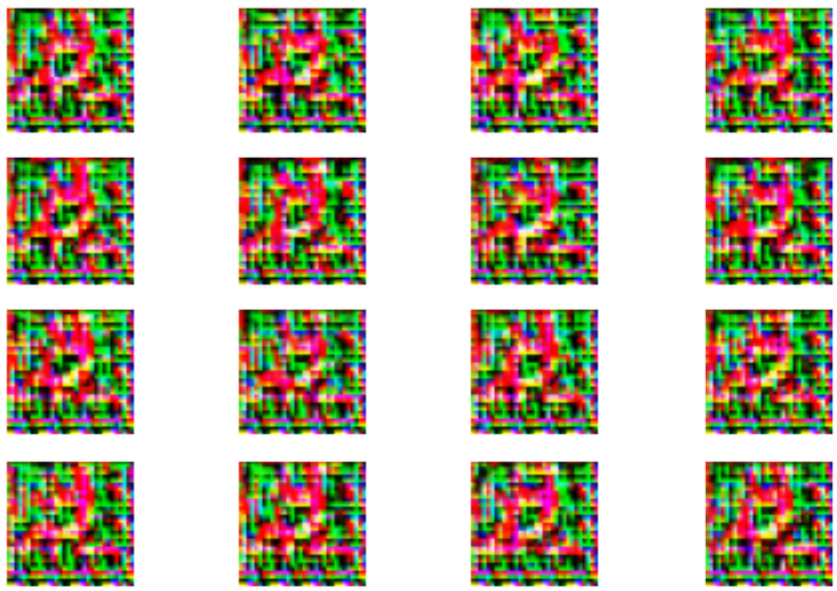
Figure 1 **Original Trained GAN Output**



Note. These 16 images represent a sample output of the GAN after it was trained (15000 epochs) with the code provided in the assignment instructions.

available training dataset, so far fewer total epochs were used than in the originally provided code. While I adopted the outline for the training algorithm from that tutorial, I reused the generator and discriminator architectures from the code provided in the assignment instructions and was met with a similar type of image output (see Figure 2).

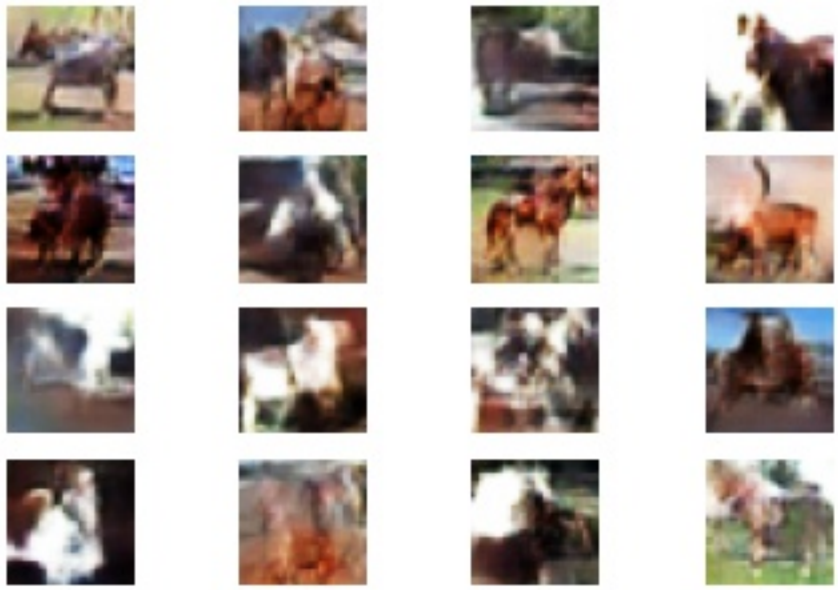
Figure 2 GAN Output with New Training Algorithm



Note. These 16 images represent a sample of the output of the GAN after it was trained with the provided architecture, but with the training algorithm outlined in Brownlee's (2019) tutorial (200 Epochs).

After adjusting both the discriminator and generator architectures, it appeared the batch normalization layers were causing the issue. I am unsure if this was due to the library versions I was running, but removing the batch normalization layers ultimately resulted in better quality images, as seen in Figure 3.

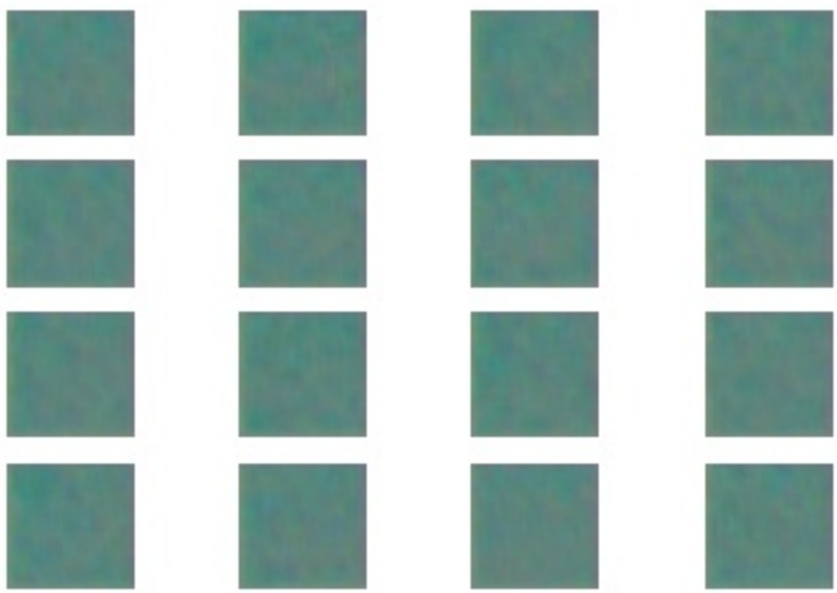
Figure 3 GAN Output with New Architecture



Note. These 16 images represent a sample of the output of the GAN after it was trained with architecture and training algorithm outlined in Brownlee's (2019) tutorial (200 Epochs).

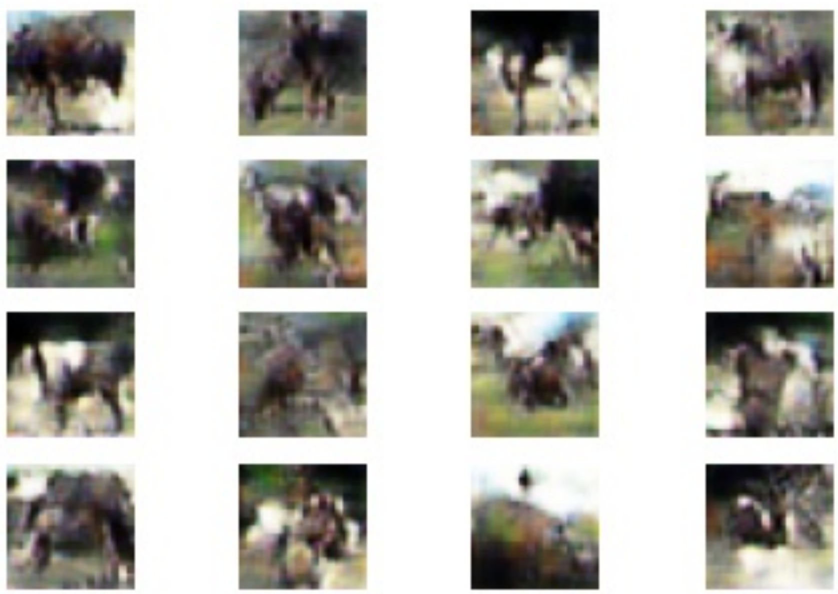
In an attempt to improve image generation quality, I implemented some additional discriminator training steps into the training pipeline. Since the ideal way to train a GAN is to ensure the Discriminator is optimal against the current generator (Creswell et al., 2018), every 10 epochs (under the new training algorithm), I had the discriminator undergo further updates in the event it identified less than 80% of the real or fake images successfully. With that addition, the GAN produced the results seen in Figures 4, 5, and 6. While not perfectly defined horses, a sample of the output images clearly show shapes and colors reminiscent of horses.

Figure 4 Finalized GAN Output - Epoch 001



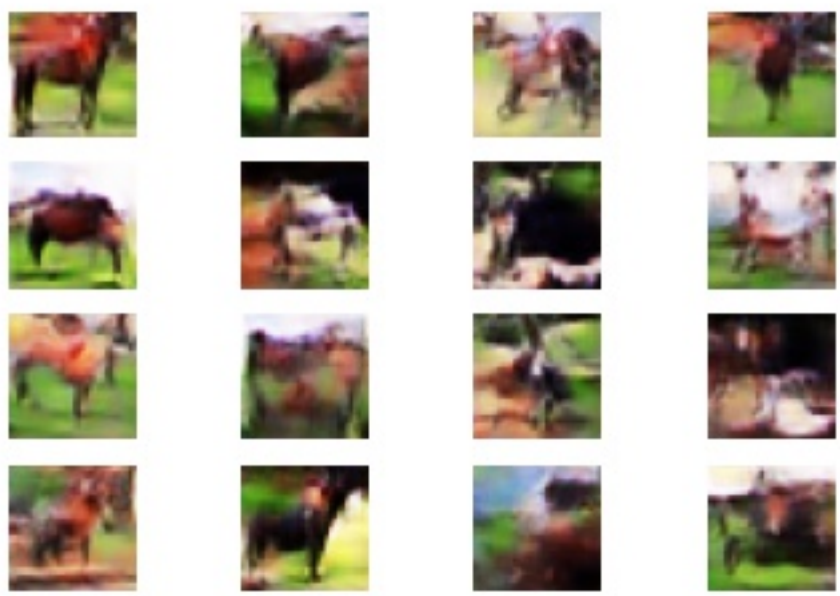
Note. These 16 images represent a sample of the output of the GAN after it was trained with original architecture (minus batch normalization layers) and training algorithm outlined in Brownlee's (2019) tutorial for 1 total epoch.

Figure 5 Finalized GAN Output - Epoch 100



Note. These 16 images represent a sample of the output of the GAN after it was trained with original architecture (minus batch normalization layers) and training algorithm outlined in Brownlee's (2019) tutorial for 100 total epochs.

Figure 6 **Finalized GAN Output - Epoch 200**



Note. These 16 images represent a sample of the output of the GAN after it was trained with original architecture (minus batch normalization layers) and training algorithm outlined in Brownlee's (2019) tutorial for 200 total epochs.

References

Brownlee, J. (2019, July 1). How to develop a GAN to generate CIFAR10 small color photographs. *Machine Learning Mastery*. https://machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-a-cifar-10-small-object-photographs-from-scratch/

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, *35*(1), 53–65. https://doi.org/10.1109/MSP.2017.2765202