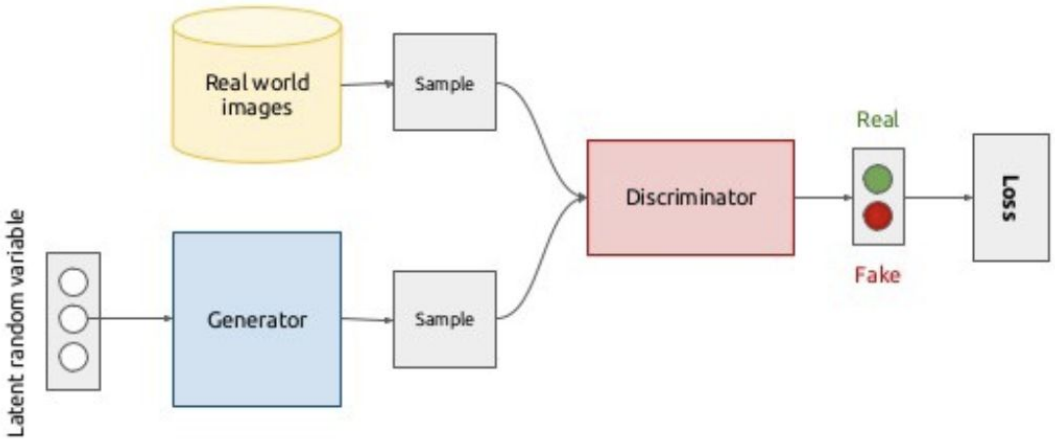
6.S198 Fall 2018 Assignment 5:



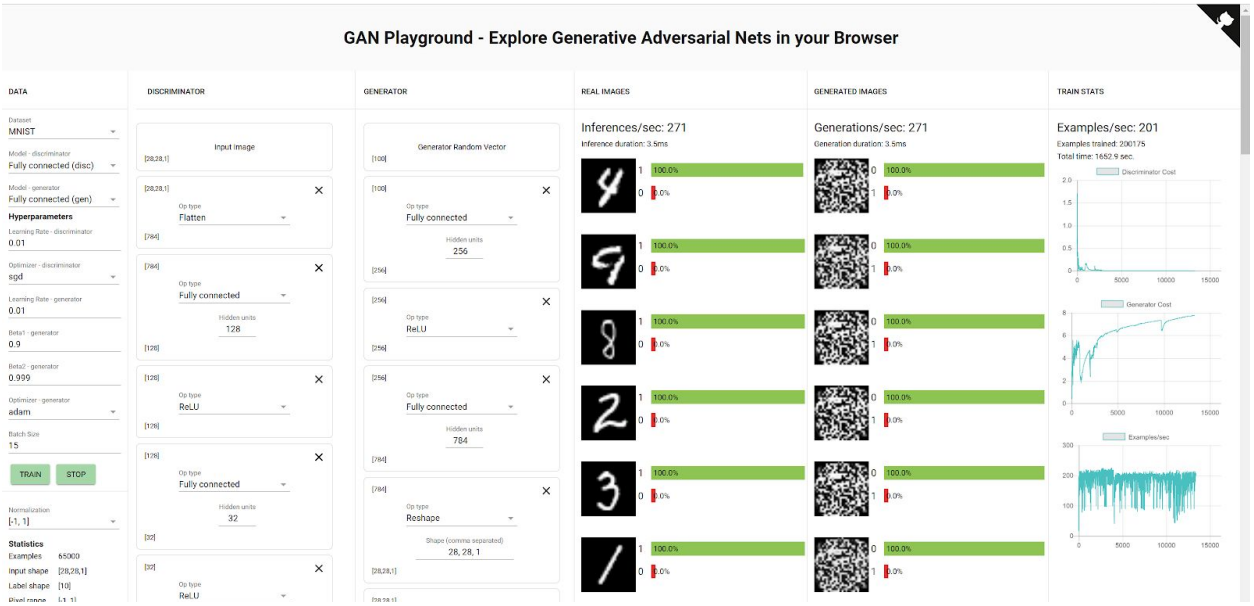
1.2.1 Why does the Generator need the second FC layer to transform the shape [256] output of the first FC layer? Hint: Can a Reshape layer reshape [600] to [20,20,2]? How about [800] to [20,10,4]?

It does not need a second FC layer to shape the 256 output of the first FC layer. The first FC layer can be set to 784 and work with the rest of the system. The output, however, needs to be [784] to be reshaped to [28.28,1]. Furthermore, [600] cannot be reshaped to [20,20,2] as the product of the latter is 800, not 600. Relatedly [800] can be reshaped to [20,10,4] as the product is 800.

Hit Train the train button and let the GAN train. Somewhat reasonable results should appear within 10-15 minutes, and you can expect results like this between iterations 200,000 and 300,000:

1.2.2 Submit screenshots of some results (generated examples, discriminator predictions on real and fake data, and learning curves as in the plots above).

FC Layer



Conv Net

1.2.3 Can you say anything about the performance of the system's default fully connected model versus the performance of the convolutional model?

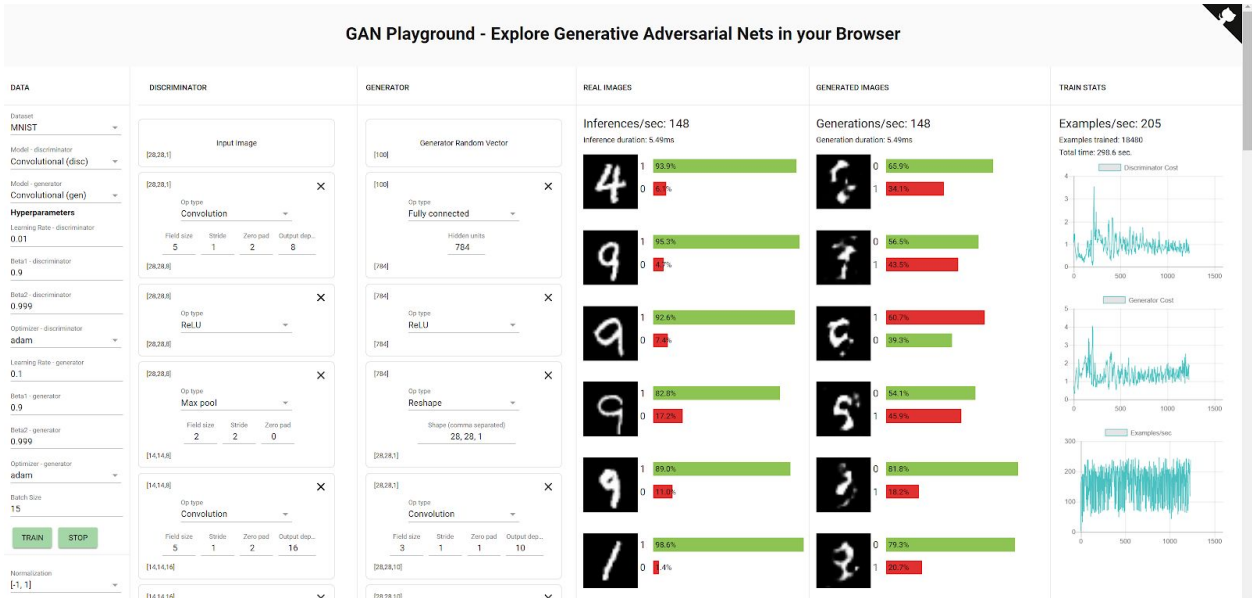
The default fully connected model was unable to product digits after 200,000 - 300,000 iterations were as the convolutional net was much more successful. The result was the convolutional net was much more accurate it is producing a strong generator

Play try some variations of the default models for the Discriminator and Generator. You can add or change layers and play with hyperparameters like learning rate and optimizer (e.g., try adams rather than simple gradient descent -- sgd). One particular direction you might explore is to investigate the effect of changing the learning rates of the Discriminator and Generator, e.g., try 0.1 rather than 0.01.

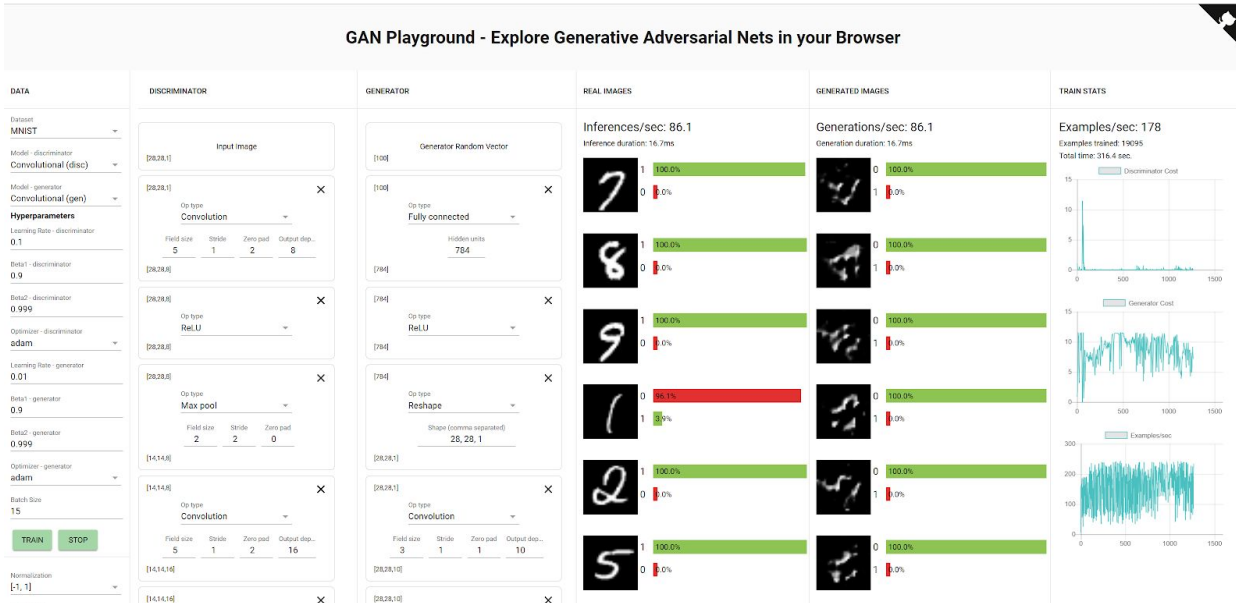
Submit screenshots of your results with at least 3 different configurations (architecture, learning rate, optimizers). At a minimum, try: the provided architecture with generator learning rate greater than discriminator learning rate, same architecture with discriminator learning rate greater than generator learning rate, and one other variant. Feel free to experiment with other layer types as well. No need for all your results to look great. Keep in mind that you'll probably need to run each of your experiments for several minutes or longer, so don't feel the need to got overboard trying lots of variations.

1.3.1. Were any of your models able to generate any reasonable MNIST digits? If so, were any of your models able to generate all of the MNIST digits (0-9)? Did any of your models get stuck at some point generating one or a few digits only (i.e. mode-collapse)?

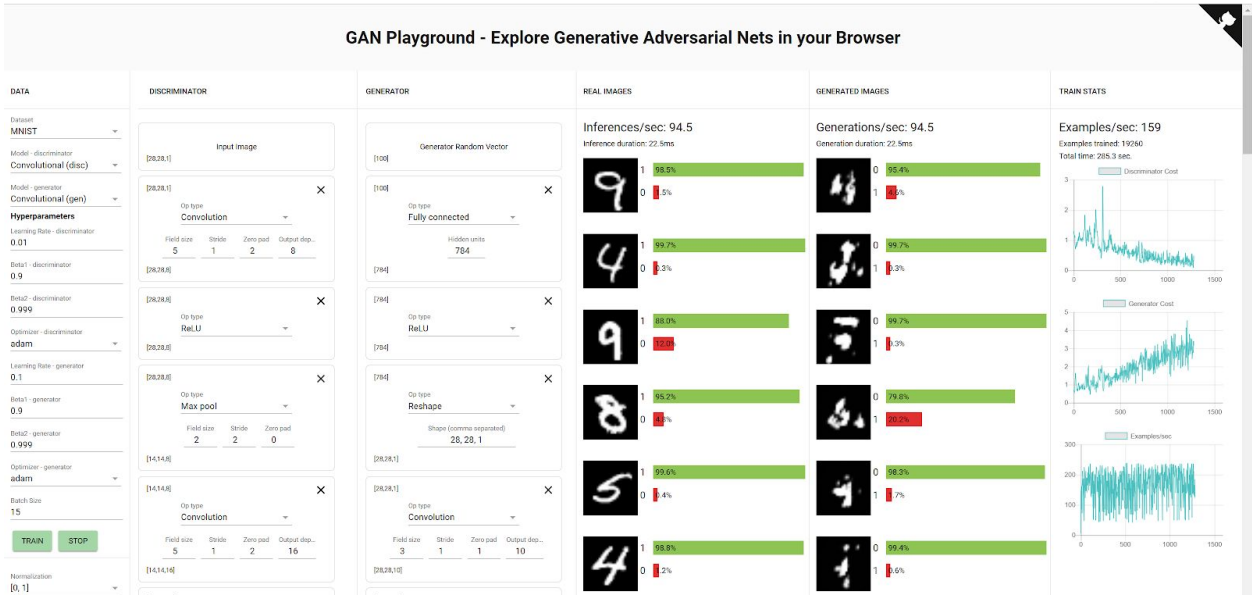
Adam, Discriminator Learning Rate .01,Generator Learning Rate .1



Adam, Discriminator Learning Rate .1, Generator Learning Rate .01



Adam, Discriminator Learning Rate .01, Generator Learning Rate .1, Normalization [0,1]



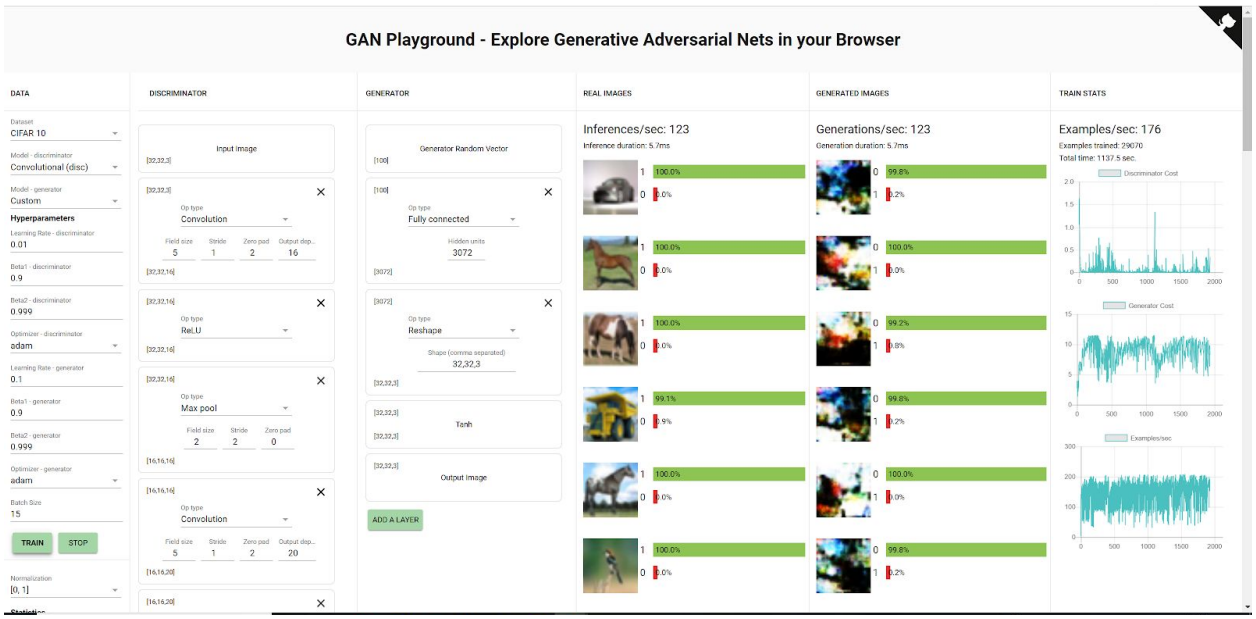
For the models where the generator learning rate was faster, I was able to produce MNIST digits. They tended to be 1,4, and 9s.

1.3.2 What happened when the discriminator learning rate was greater than the generator learning rate? What about when the generator learning rate was greater than the discriminator learning rate?

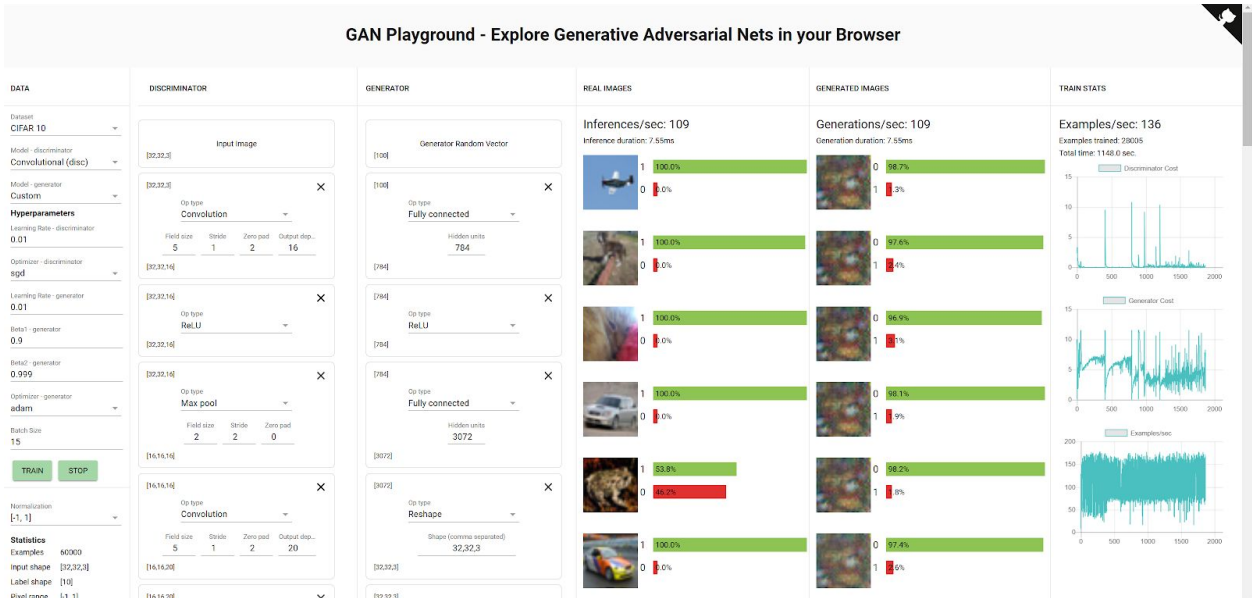
When the discriminator learning rate was greater than the generator learning rate, the discriminator was able to keep the generator from succeeded and the generator produced poor images that the discriminator was able to identify as false. This was in sharp contrast to when the generator learning rate was faster and we were able to produce realistic MNIST data points.

1.3.3 Try building configurations for CIFAR. (This will require a minute or two to loadFirst, try running with only FC layers for ~15 minutes and document your results.

1x FC



2x FC



Use This form to hand in Assignment 5:

Due 10AM Monday, 9/15

https://drive.google.com/open?id=16dUFcUxy_VVt7WIOv8ZJodXc7H-tu6sQf0yYW--i7Qc