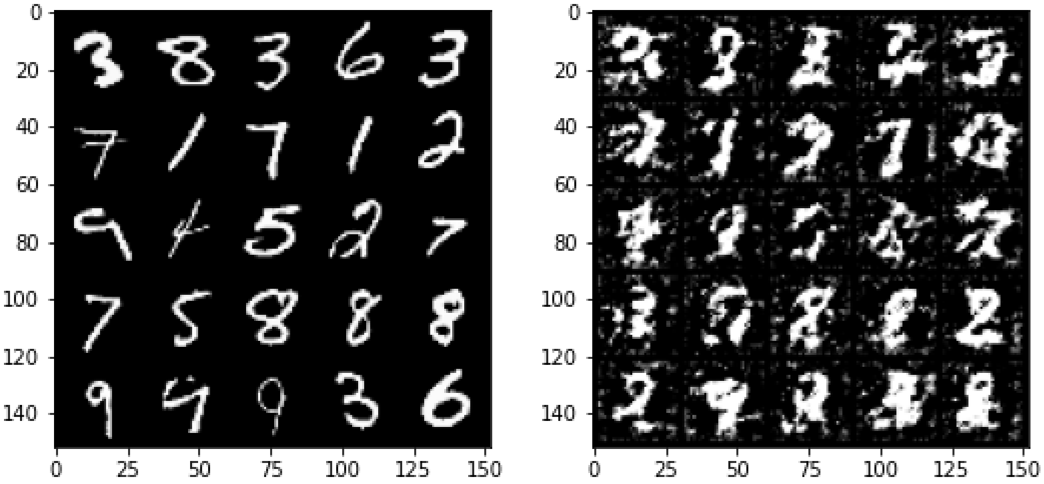
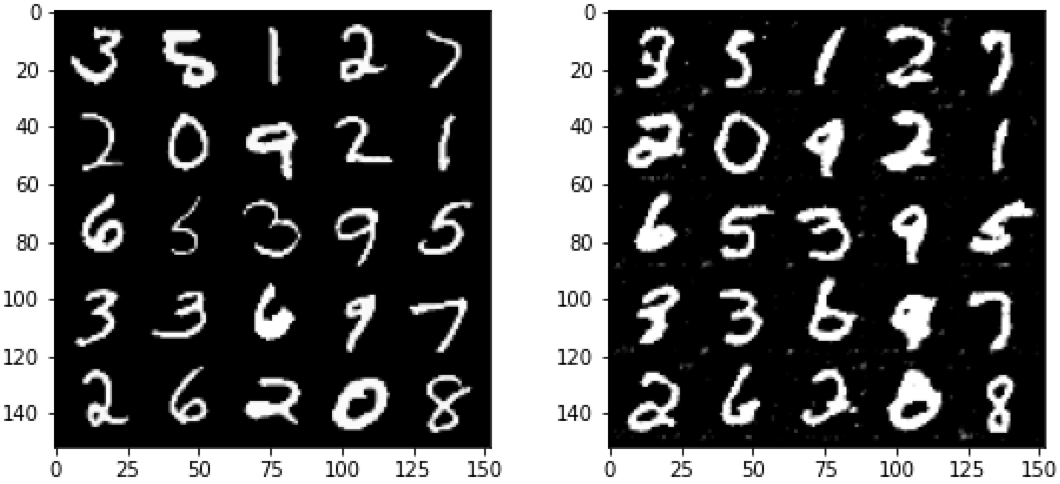
ECE 792 Midterm

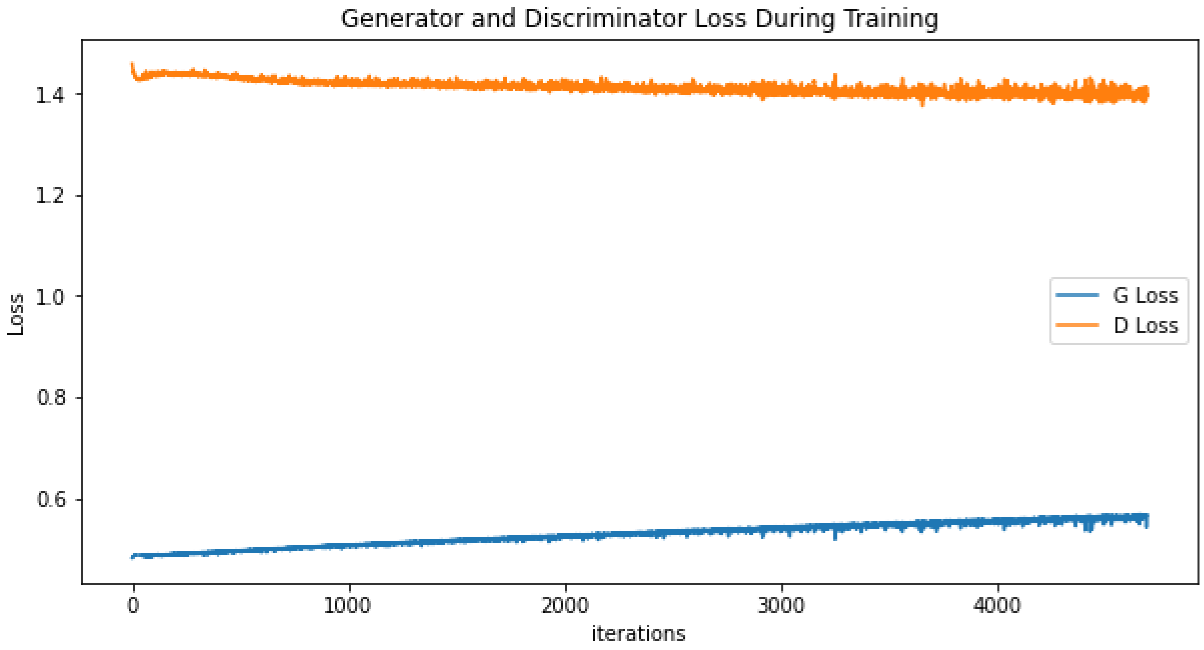
Problem 3:

For epoch 1, (real and fake image):



For epoch 10 (real and fake image) :





Why was batch-normalization used in the last layers of the Discriminator and Generator?

Batch normalization helps stabilize the training process by normalizing inputs at each layer where it is applied. This can make the training process more stable and prevent the model from getting stuck in a poor local minimum.

In our case in the discriminator, batch normalization can help to normalize the output of the last layer before it is fed into the sigmoid activation function, which can help to prevent vanishing gradients and improve the stability of the model.

In the generator, batch normalization can help to normalize the output of the last transposed convolution layer before it is fed into the ReLU activation function. This can help to prevent the generator from producing extreme pixel values, which can make the generated images look unrealistic. Additionally, batch normalization can help to smooth out the distribution of the feature maps in the generator, which can improve the overall quality of the generated images.

Why was backpropagation done only on Discriminator and not on Generator in Step-1?

In step-1, backpropagation is only done on the Discriminator and not on the Generator because the goal of step-1 is to train the Discriminator to distinguish between real and fake images.

In a conditional GAN (cGAN), the Generator takes in additional information in the form of class labels to generate images. This means that during the training process, the Generator's weights are updated not just to generate realistic images but also to generate images that belong to a specific class. Therefore, updating the weights of the Generator involves updating the weights for both the image generation and class prediction tasks, making the optimization process more complex.

To simplify the optimization process in cGANs, backpropagation is done only on the Discriminator in Step-1 of the training process. This means that during this step, the Discriminator's weights are updated to better distinguish between real and fake images, but the Generator's weights are not updated. The purpose of this is to ensure that the Discriminator is able to provide meaningful feedback to the Generator on how to improve its generated images. Once the Discriminator is optimized, backpropagation is done on the Generator in Step-2 to update its weights to generate better images that are more likely to fool the Discriminator.