Project 2 Report

**Model Architecture**

The model architecture consisted of the following sections: Importing Packages, Dataset Preparation, Gan Architecture, Loss and Optimizer, and training loop. First, the *Importing Packages* section consisted of importing the proper libraries and packages for the rest of the code to work. Second, the *Dataset Preparation* section consisted of downloading the Fashion-MNIST dataset, normalizing the values between -1 and 1, and creating DataLoader objects for training the dataset to iterate over batches. Third, the *GAN Architecture* consisted of two classes one for the generator and the discriminator. The generator consisted of four linear FFNN layers and LeakyRelu activation functions ending with tanh. The discriminator consisted of three linear FFNN layers and LeakyRelu activation functions ending with sigmoid. Fourth, The *Loss and Optimizer* consisted of a binary loss entropy function for both the generator and the discriminator and an Adam optimizer. The learning rate was set at .0002. Finally, the training loop, for each batch, is iterated over the data loader doing the following: training the discriminator, training the generator, calculating the average loss for each epoch, visualizing the images, and plotting the loss curves for the generator and discriminator.

**Training Progress**

There were many tweaks made to the model over the course of training:

1. For high-quality results a proper learning rate needed to be used for this model. The first learning rate used was the standard .001, which yielded a loss curve that would first dip and gradually rise. After researching the learning rate for GAN architectures, a new learning rate of .0002 was used as it was mostly standard for these models. This learning rate yields a better loss curve that would plateau after 50 epochs.
2. Implementing batch normalization in the generator was also considered to improve the results of the model. However, the loss curve for the generator began to climb positively and the resulting images were grainier with white spots on the background. The model was reverted and did not use batch normalization. Below are the loss curves and images using batch normalization.

A collage of images of a person

Description automatically generated

A graph of a graph

Description automatically generated

**Challenges**

While building this model the main challenge was writing the code for the training loop. Using the instructions in the assignment outline and conversations with Professor Watson, I obtained a better idea of the necessary components for the loop to work. I began by adding the beta parameters to the optimizer for the generator and discriminator. Then I mapped out the training loop by dividing it into the following sections: training the discriminator, training the generator, calculating the loss, and visualizing generated images. Using the loss code from a class presentation I was able to get output loss for both the generator and discriminator. The rest of the code fell into place afterward.

**Results and Evaluation**

**A collage of images of a person's face

Description automatically generated**

A collage of images of clothes

Description automatically generated

A graph of a graph

Description automatically generated

The images presented primarily feature identifiable fashion items, encompassing both apparel and footwear. Among these, three items are less discernible due to their blurred appearance and ambiguous shapes, making it challenging to determine their identity conclusively. Despite this, the assortment of items showcased lacks uniformity, illustrating a diverse range. The loss curve depicted indicates that the generator's performance starts at a loss value of 1.4, which subsequently decreases to 0.8. In contrast, the discriminator's loss curve exhibits a rise from 0.4 to 0.7. The visuals generated by the generator do not exhibit any indications of mode collapse, suggesting effective variation and learning in the model's output.