Name: Kade Boltjes

Email: kboltjes@purdue.edu

Github Repo: https://github.com/Kboltjes/CS390Lab4

Resources Used:

- https://keras.io/api/layers/core_layers/dense/
- https://keras.io/api/layers/reshaping_layers/reshape/
- https://keras.io/api/layers/convolution_layers/convolution2d_transpose/

Parts Completed

I completed all the parts that were not extra credit. I was able to generate legible F-MNIST records using a convnet for the GAN while having the option to train the generator and discriminator at different ratios by adjusting the global variable GENERATOR_TO_DISCRIMINATOR_TRAIN_RATIO. I also generated plots showing the losses for both the generator and discriminator.

Questions

Describe the discriminator and generator.

The discriminator classifies whether an input image is fake or real. The generator takes in random noise and transforms it into an object that is the same shape as an image. Then every epoch, the discriminator is trained using the output from the generator and actual F-MNIST images. The true labels we feed it for those are fake and real respectively. We then train the GAN by using random data fed into the generator, which is subsequently fed into the discriminator.

Why do we sometimes need to train the discriminator and generator different amounts?

If the discriminator learns at a faster rate, its accuracy could improve much faster than the generator, so the gap between them will grow larger. When that happens, the generator can’t catch up, so the GAN’s outputs are bad.

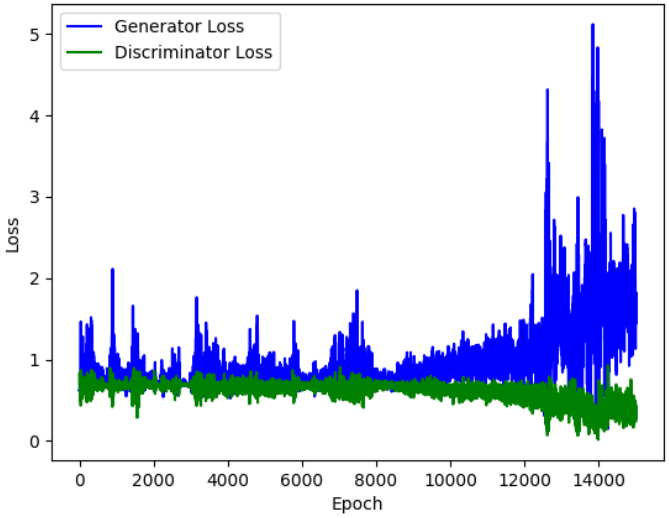
Hyperparameters

Epochs	15000
Batch Size	32
Optimizer	Adam with learning rate 0.0002
Loss	Binary Crossentropy

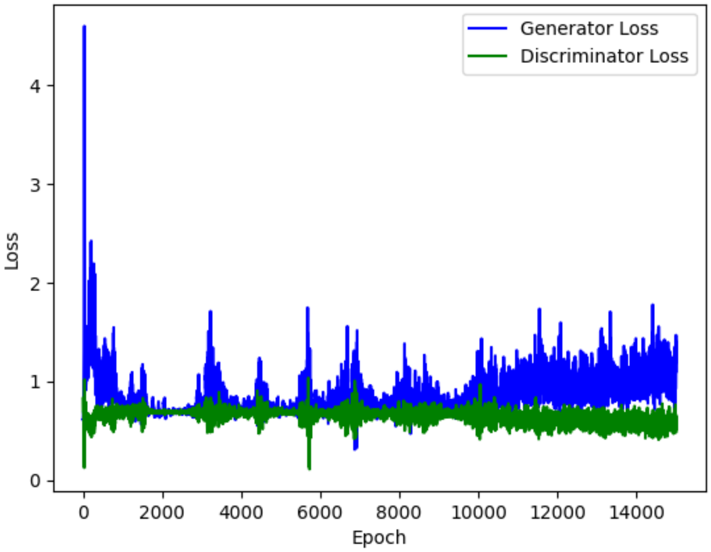
Discriminator			Generator		
Layer	**# of Neurons or filters**	**Activation Function**	**Layer**	**# of Neurons or filters**	**Activation Function**
Conv2D (3x3)	32	Sigmoid	Conv2DTranspose (3x3)	128	Elu
Conv2D (3x3)	64	Relu	BatchNormalization momentum of 0.8	-	-
MaxPooling2D (2x2)	-	-	Conv2DTranspose (5x5)	64	Elu
Dropout of 0.2	-	-	BatchNormalization momentum of 0.8	-	-
Conv2D (3x3)	48	Relu	Conv2DTranspose (5x5)	64	Elu
Conv2D (3x3)	64	Relu	BatchNormalization momentum of 0.8	-	-
Dropout of 0.2	-	-	Conv2DTranspose (3x3)	48	Elu
Dense	128	Relu	BatchNormalization momentum of 0.8	-	-
Dense	64	Relu	Conv2DTranspose (5x5)	48	Elu
Dense	1	Sigmoid	BatchNormalization momentum of 0.8	-	-
			Dense	512	Elu
			BatchNormalization momentum of 0.8	-	-
			Dense	1024	Elu
			BatchNormalization momentum of 0.8	-	-
			Dense	784	Tanh

Loss Plots

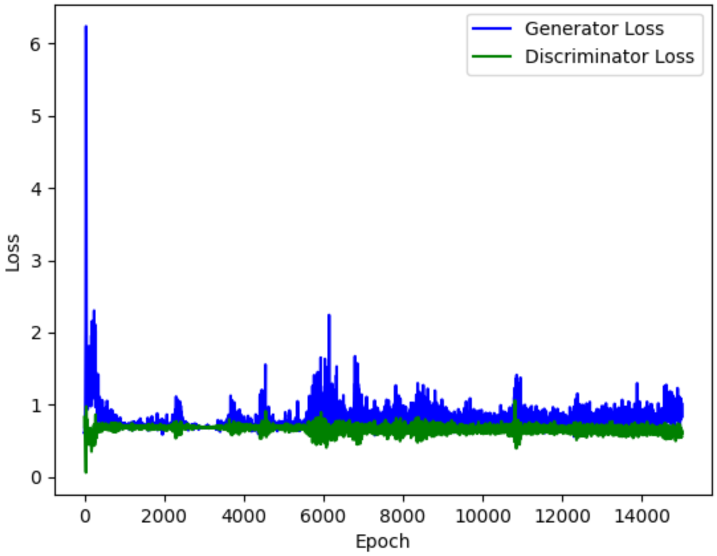
F-MNIST Coat



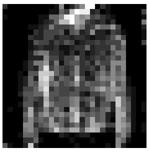
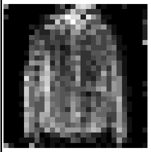
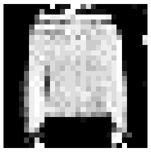
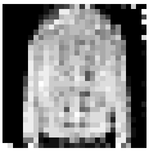
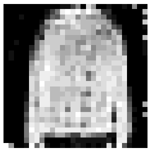
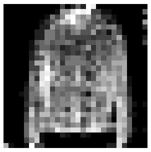
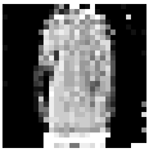
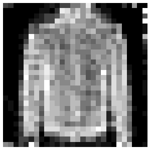
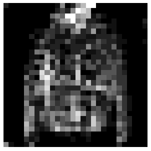
F-MNIST Sneaker

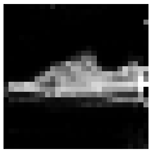
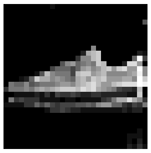
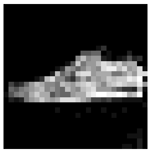
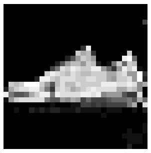
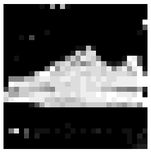
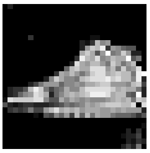
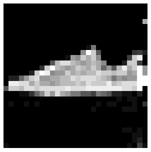
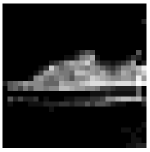
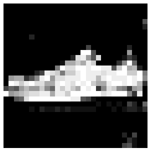


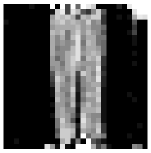
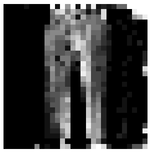
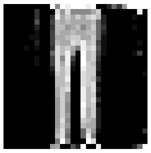
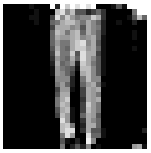
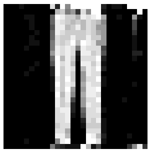
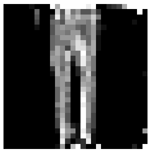
F-MNIST Trouser



Generated Images

<table>
<tr><th colspan="10">Coat</th></tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td> </td>
<td></td>
<td></td>
</tr>
</table>

<table>
<tr><th colspan="10">Sneaker</th></tr>
<tr>
<td></td>
<td></td>
<td></td>
<td> </td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</table>

<table>
<tr><th colspan="10">Trouser</th></tr>
<tr>
<td> </td>
<td></td>
<td></td>
<td> </td>
<td></td>
<td></td>
<td> </td>
<td></td>
<td> </td>
<td></td>
</tr>
</table>