DS340 Final Project Paper

Authors: Farid Karimli, Raul-Fikrat Azizli

**Introduction**

The concept of generating fake images was very interesting to us. We considered the ability to create images complicated, requiring specific ‘intelligence’ to be carried out. So, we set out to develop a series of generative adversarial networks that generate images of digits and articles of clothing, to learn more about the mechanics of neural networks, their capabilities and applications.

**Methods**

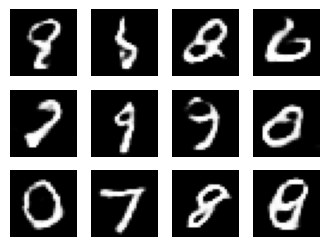
We developed, designed and trained 7 GANs, each with their own unique architecture that generate images of handwritten digits and articles of clothing. We used the MNIST and Fashion MNIST datasets. Three of our GANs were trained on MNIST, and the remaining 4 on Fashion MNIST. We used the Tensorflow GAN tutorial as a base, and changed the hyperparameters for each version. We experimented with the number of layers, parameters of the Conv2d\_transpose layers (kernel size, stride size, number of units). We also tweaked the discriminator for some of them, adding more convolutional layers. Our first big change was increasing the dimension of the input noise into the generator from 100 to 200, which significantly improved the generated images.

The label for a real image is 1, and 0 otherwise. Thus, our loss function for the discriminator was essentially comparing its predictions on real images to an array of ones, and its predictions on the fake images to an array of zeros. The loss function for the generator is comparing the discriminator’s predictions on *fake* images to an array of *ones,* in order to ‘trick’ it.

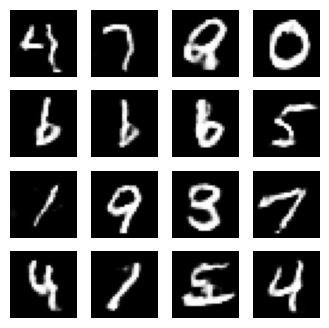
**Results**

All of the images are outputs of the GAN after training for 100 epochs

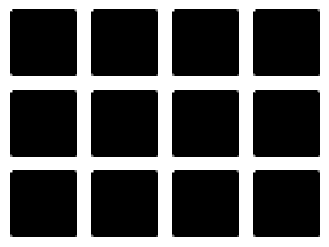
MNIST GAN Version 1 - Uses bias in all layers, kernel size of 5 and strides increasing from (1,1) to (2,2) down the network



MNIST GAN Version 2 - one more dense layer (the only one that uses bias), kernel size of 4.



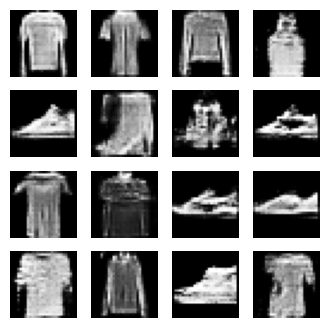
MNIST GAN Version 3 - sigmoid activation in the last layer



Fashion MNIST GAN Version 1

- GAN’S Conv2D\_Transpose strides go down from (2,2) to (1,1), kernel size of 5

- Discriminator has one more convolutional layer with 16 units



Fashion MNIST GAN Version 2

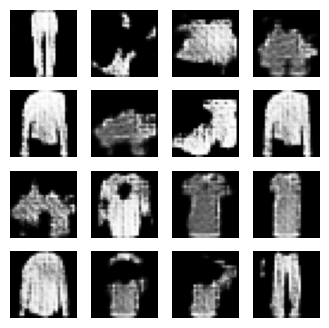
- GAN’S Conv2D\_Transpose kernel size go down from 5 to 3 down the network

- GAN has 2 dense layers, 1 uses bias



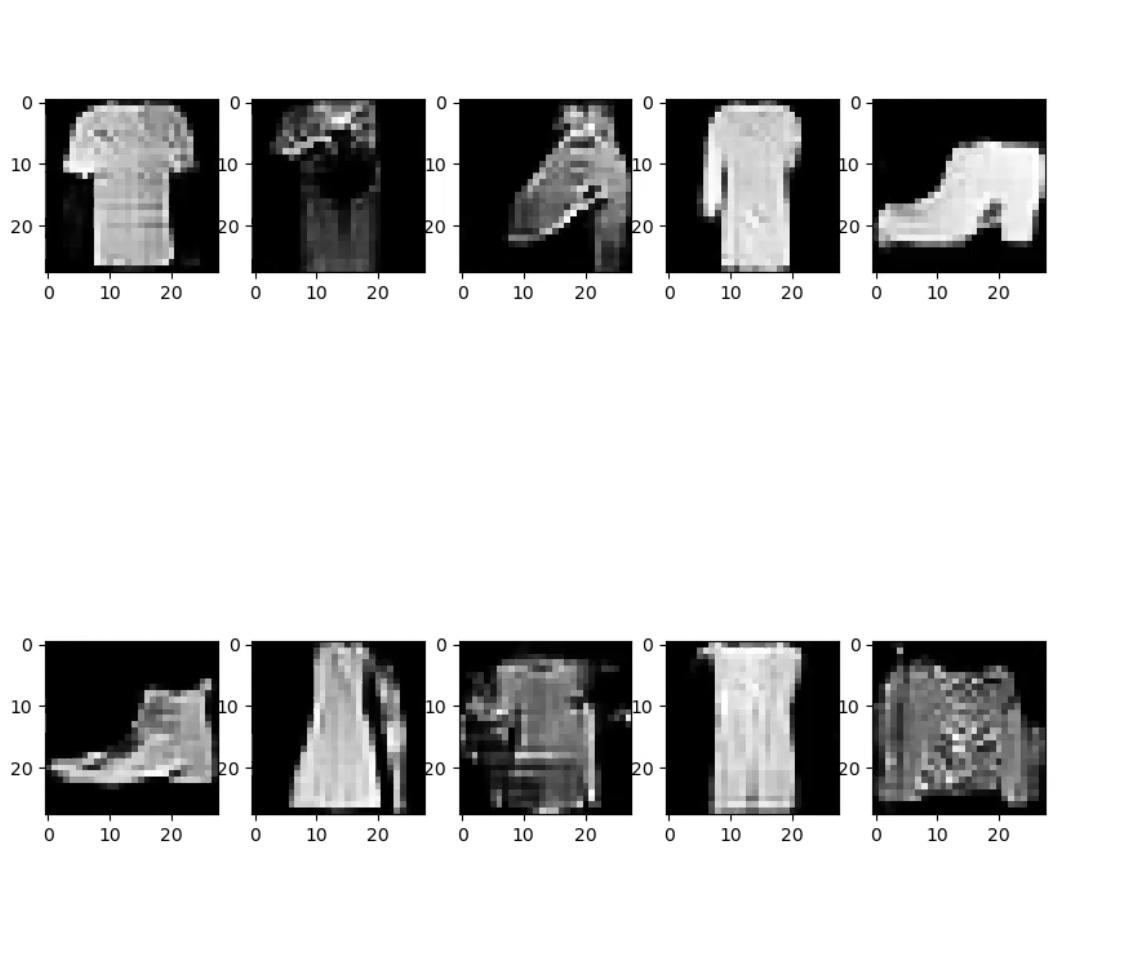
Fashion GAN Version 3

* has another Dense layer that uses bias, and uses ReLU instead of LeakyReLU
* has another Conv2d\_transpose layer with 128 units



Fashion GAN Version 4

* Generator only has 2 upscaling layers, going from 64 units to 1, with activations of relu and tanh respectively. Kernel size is 5, strides are a shape of (2,2).
* The dimensions of the noise input into the generator is 100 instead of 200
* We went with a much simpler architecture for this one.



The way we evaluate our model is through the eye test, since comparing the losses of the generator and the discriminator is not reliable. In nearly all of the versions of both GANs (except V3 of the MNIST GAN), most of the objects in the images are fairly recognizable, and reasonably similar to the real images.

**Conclusion**

We learned a lot from this project. Firstly, we learn about the practice of generating data using neural networks, which is very different from the traditional neural networks that we’ve been used to in our very short time learning about them. We learned about new neural architectures, and their implementations in Tensorflow. By extension, we also got acquainted with new layers, different training workflows, and implementing custom loss functions in Tensorflow.

Finally, we learned how to manage our training time effectively and improve efficiency, namely first training for a small amount of time during experimentation, then longer once we see promising results. Also, using multiple accounts to train different functions at the same time proved to be very effective.