EE6132 Advanced Topics in Signal Processing - Assignment No 5

Name: Atishay Ganesh
Roll Number: EE17B155

December 2, 2019

1 Abstract

The goal of this assignment is to experiment with Generative Adversarial Networks and use them to learn a mapping from a latent space to a distribution of interest.

2 Assignment

2.1 Implementation Details

We use Pytorch in Python 3.6 for implementing the Generative Adversarial . We use torch as the Deep Learning Framework, and matplotlib to plot the graphs.

2.2 Training a Generative Adversarial Network

We train a GAN to generate data belonging to a Gaussian distribution with $\mu = 2 and \sigma = 0.2$. The Generator and the Discriminator are both fully connected networks with a 2 hidden layers of size 8. Adam with LR of 1e-3 was used for the generator and 1e-2 for the discriminator. for every 10 updates of the discriminator, the generator was updated once. Minibatch size 64 Every 50 minibatches a test is done to see how close the generator samples are to the data distribution. Training is continued till the Discriminator Accuracy becomes 50%. An epoch is considered as 100 minibatches, for the sake of simplicity.

Final Loss after 60 epochs:
Discriminator loss 0.6824
Generator loss 0.7155
The final GIF is included in the zip file along with code for the GAN.

3 Conclusions

We explored Generative Adverserial Networks and how they can be used to create new samples from a distribution of interest.

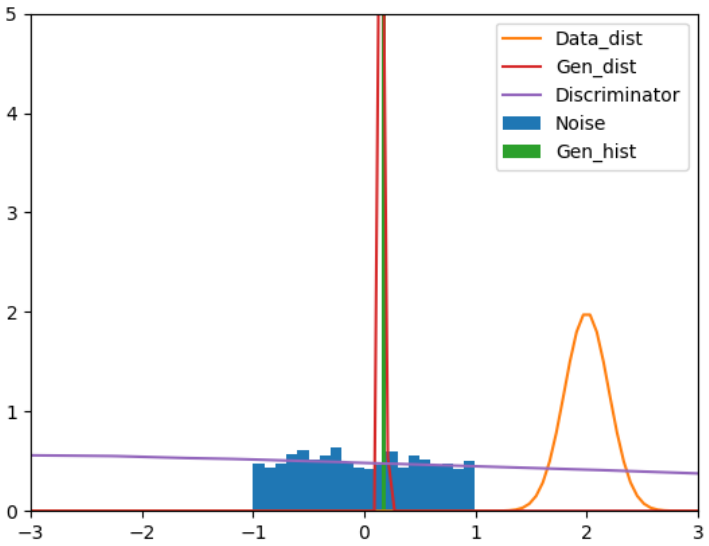

Figure 1: Initially the generated distribution is zero mean with very low variance.

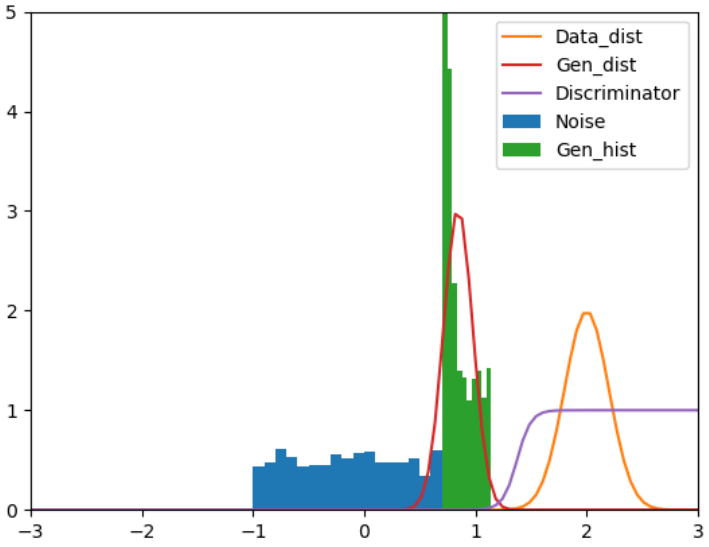

Figure 2: Progress after 10 Epochs.

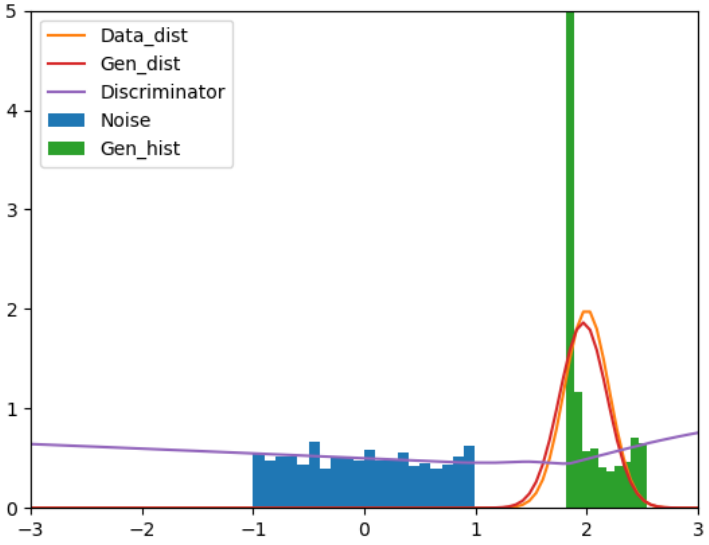

Figure 3: Progress after 20 Epochs.

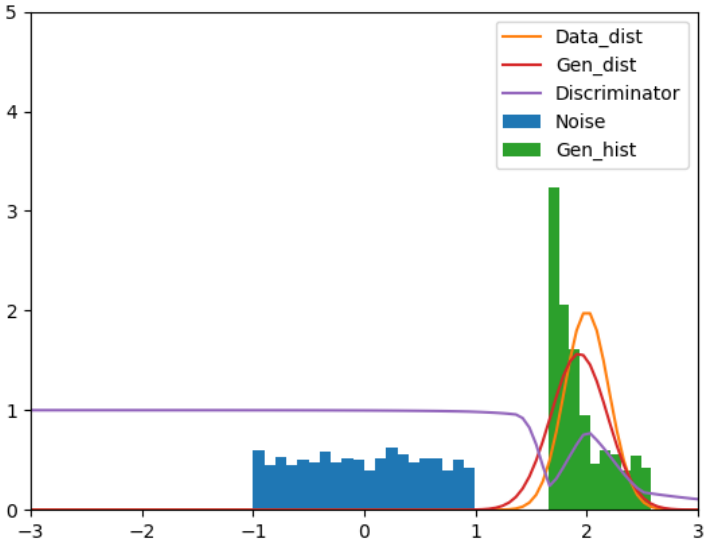

Figure 4: Progress after 30 Epochs.

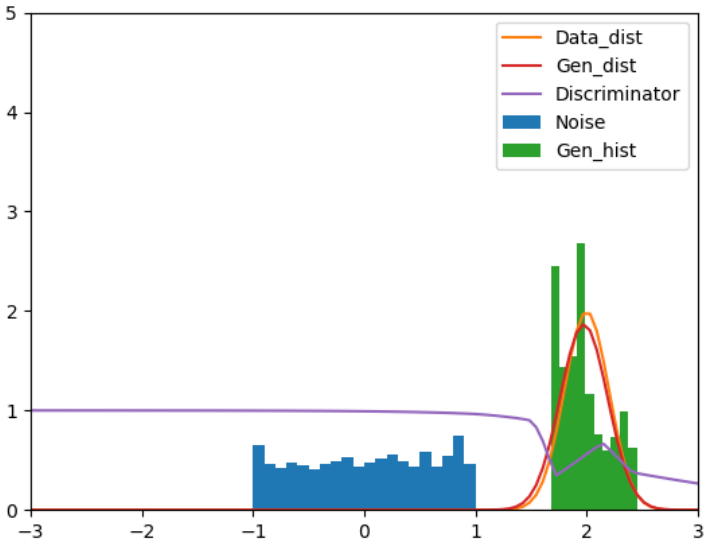

Figure 5: Progress after 40 Epochs.

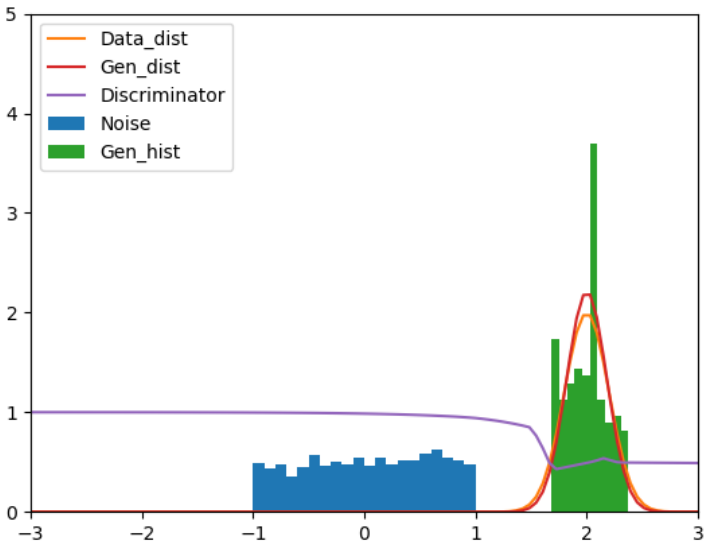

Figure 6: Progress after 50 Epochs.

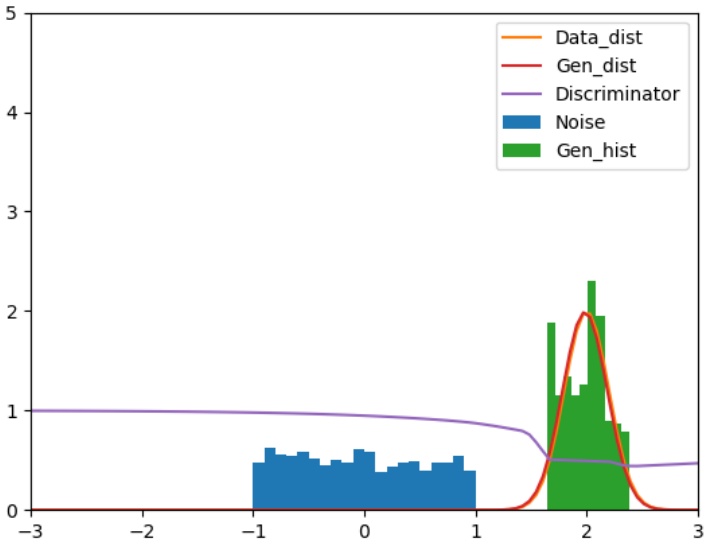

Figure 7: Progress after 60 Epochs.