# EE6132 Assignment No. 3: Report

-Akash Reddy A, EE17B001

October 15, 2019

## Introductory Notes

**PyTorch** has been leveraged to construct all the models in this assignment and answer the sub-questions as well. `torchvision.datasets` and `torchvision.transforms` have been used to extract the MNIST datasets directly from the website and convert them into Tensors. The code has been written into a Google Colab notebook, linked here.

## 1 Generative Adversarial Networks

Minibatches of 64 are used. K = 10 is used. Activations are ReLU for generator and tanh for discriminator.

We observe from plots plotted during training that the generated outputs and distribution have random mean and variance in the beginning of training, while they move towards (and oscillate in a damped manner around) the expected mean and variance of the data distribution.

Additionally, the value of the discriminator outputs initially are meaningless, as the untrained discriminator does not do a good job in the beginning. As we train, the discriminator starts to perform better and so does the generator, so the outputs of the discriminator start moving towards 0.5. At the end of training, the discriminator outputs tend to 0.5.

**All of this is visible in the plots which are plotted in the next page. In the subsequent figure, the final mean and standard deviation of the generated histogram are calculated.** They are close to $\mu = 2.0$ and $\sigma = 0.2$, as is required.

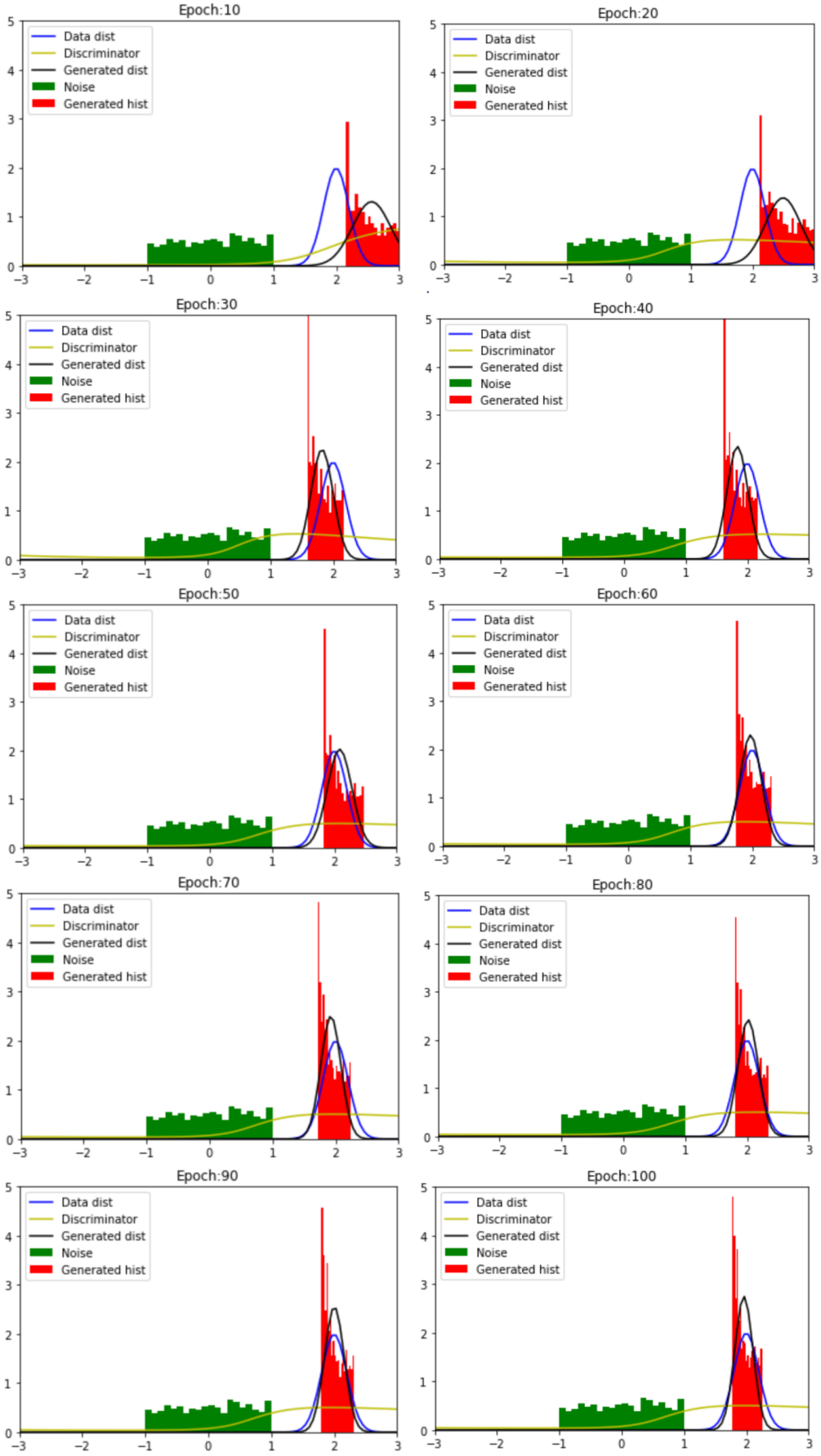


Figure 1: Plots plotted once every 10 epochs

```
outputs = Gen(inputs)
out_mu = torch.mean(outputs).item()
out_sigma = torch.std(outputs).item()
print("Generated dist mean = "+str(out_mu))
print("Generated dist std. deviation = "+str(out_sigma))
```

```
Generated dist mean = 1.9614616632461548
Generated dist std. deviation = 0.145040825009346
```

Figure 2: The mean and standard deviation of the final generator outputs

**********