

Build Your Own Autoencoder

First I tried a model with 8 layers which reduced picture dimension slowly (256,128,64,32,16,8,4,2) but after testing it realized it's not good enough.

I decided to use a model with two hidden layers as I have seen in the internet , using sigmoid activation function for both layers first layer reduce the dim to 50 and second to 2.

We can see the loss decrease quickly :

Step 0: Minibatch Loss: 0.366440

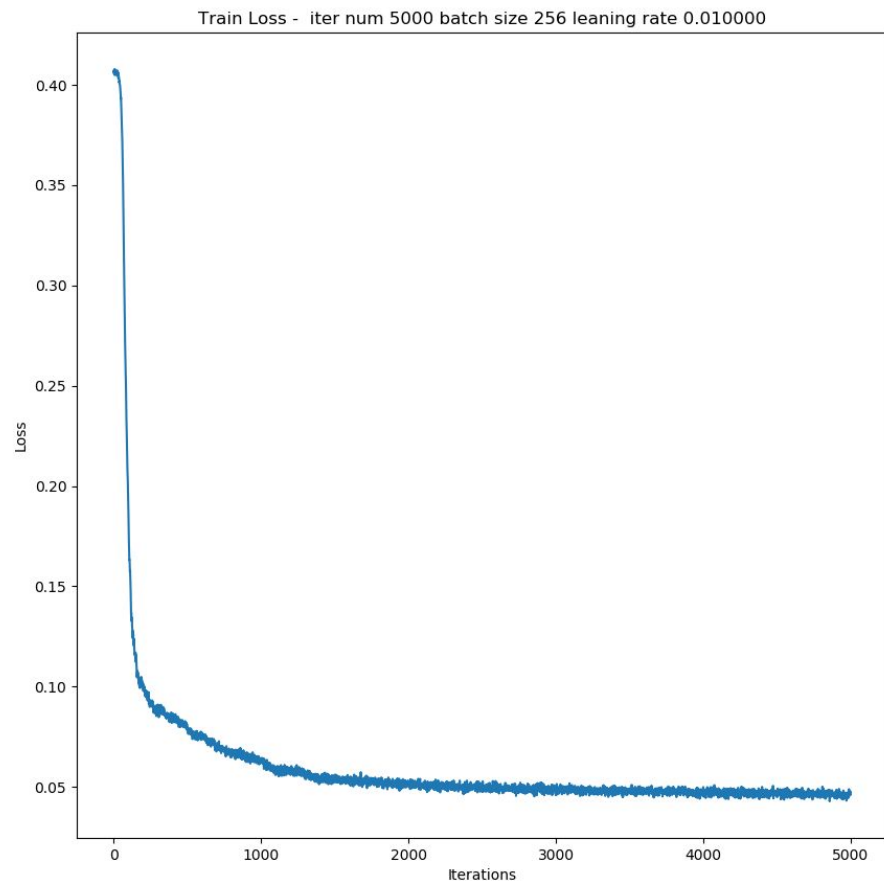
Step 1: Minibatch Loss: 0.366783

Step 1000: Minibatch Loss: 0.058132

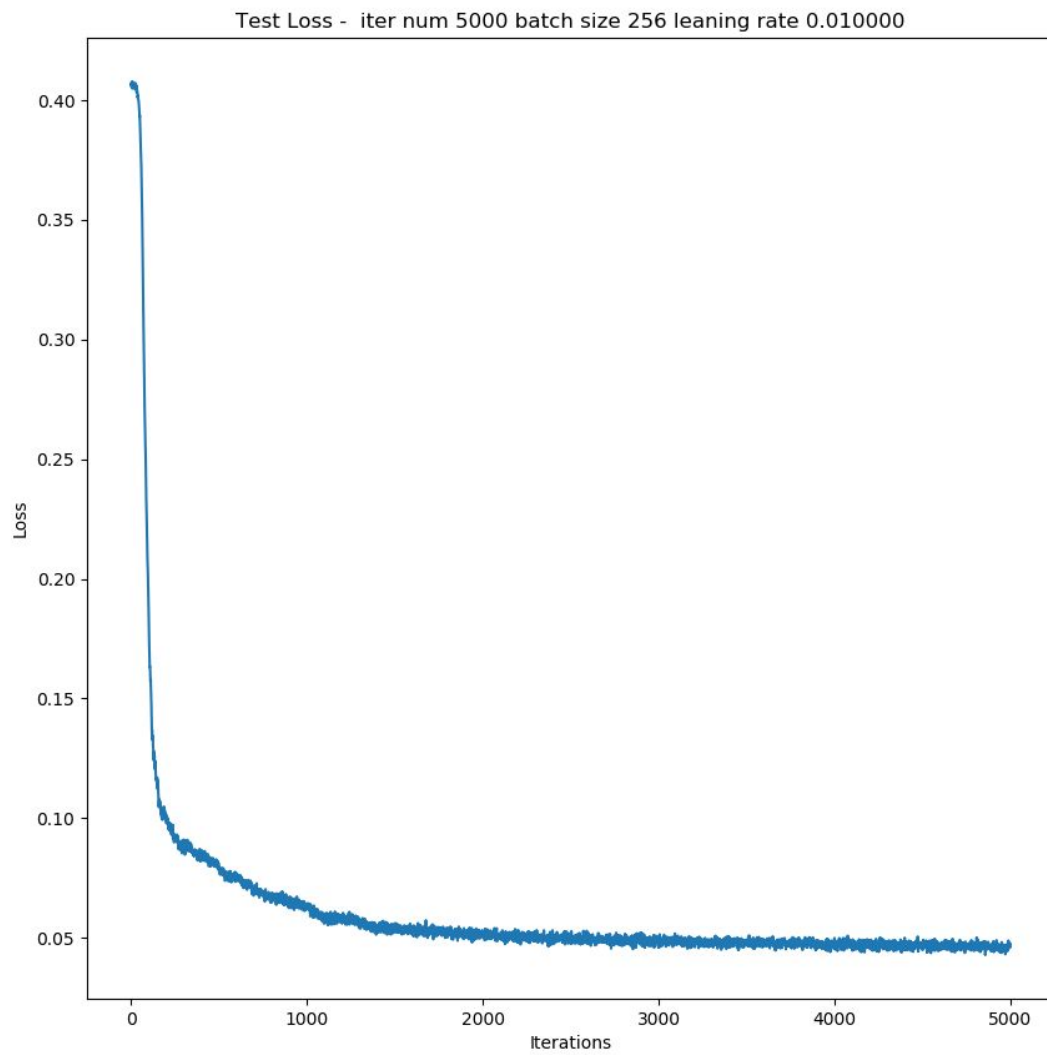
Step 2000: Minibatch Loss: 0.053850

Step 5000: Minibatch Loss: 0.049253

Step 9000: Minibatch Loss: 0.045571

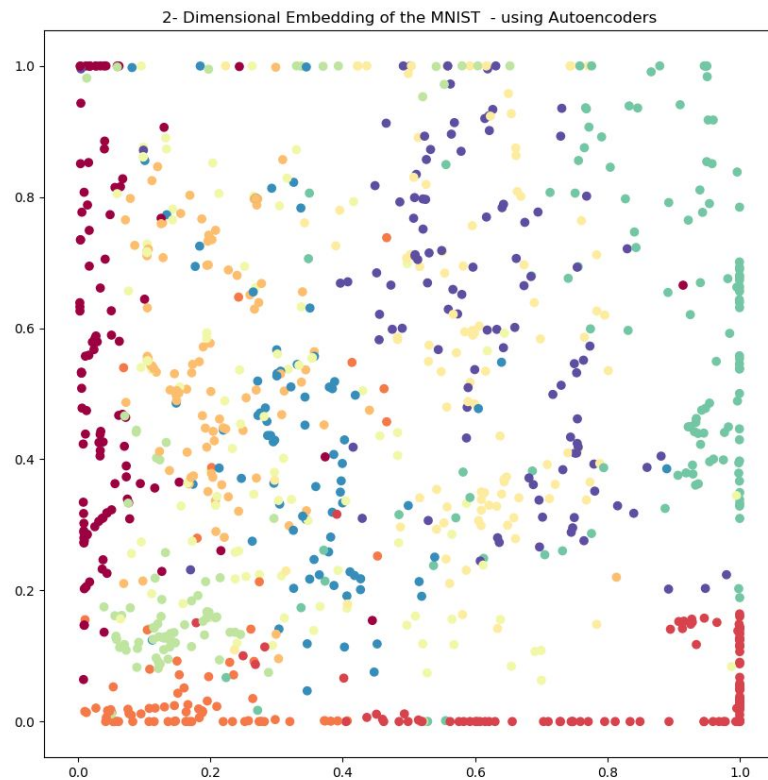


Test my autoencoder -



Plot of the 2 dimensional data :

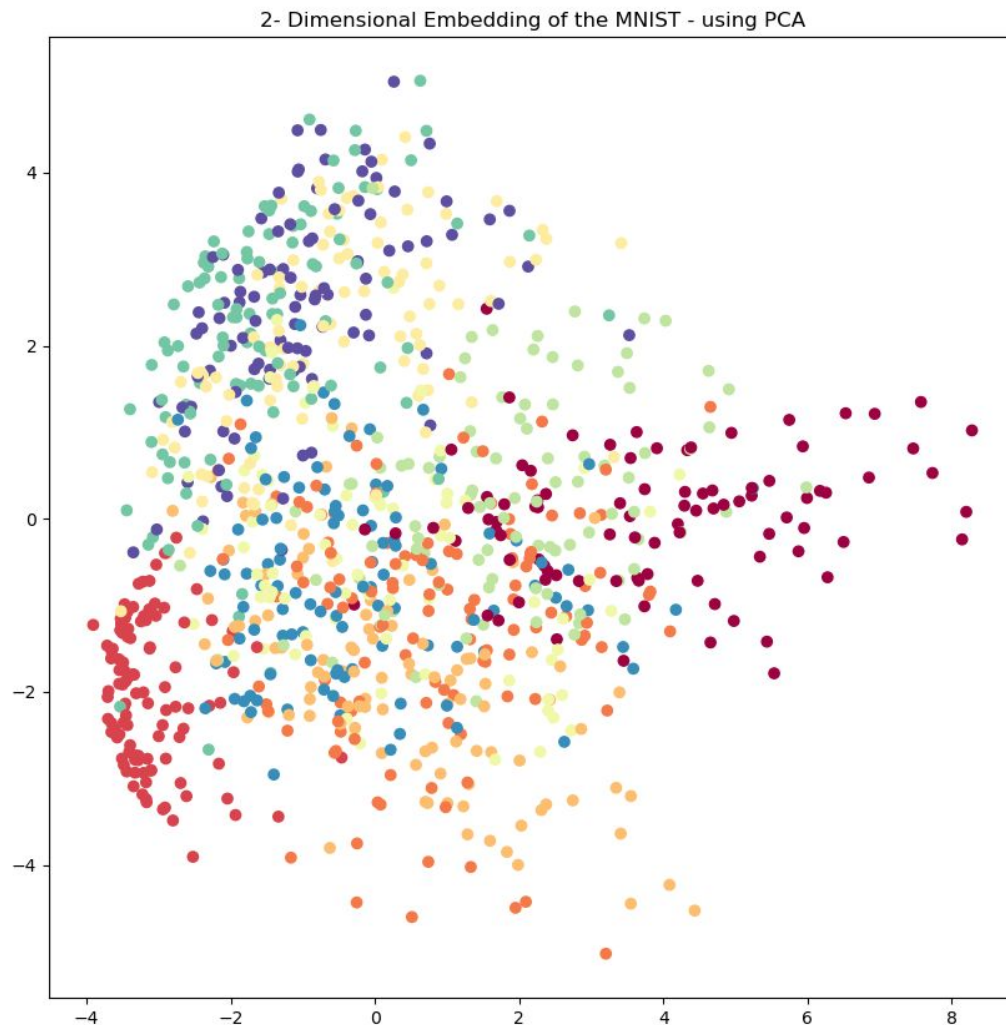
We can see a structure but not clear as in the exercise we did for dimension reduction, since the decoder is still having some hard time in some of the cases, for example differentiate between 2 and 6 (when written in a certain way), and it fits the fact we still have a small loss after 5,000 iterations.



Comparison to PCA

We can see that both of the models generally give us good results reducing the data dimension. PCA is a linear method which uses the most distinguishable information in the data and set them as the new data axis, in the plot we see that indeed the data is clustered into groups but many of the groups have a big overlap.

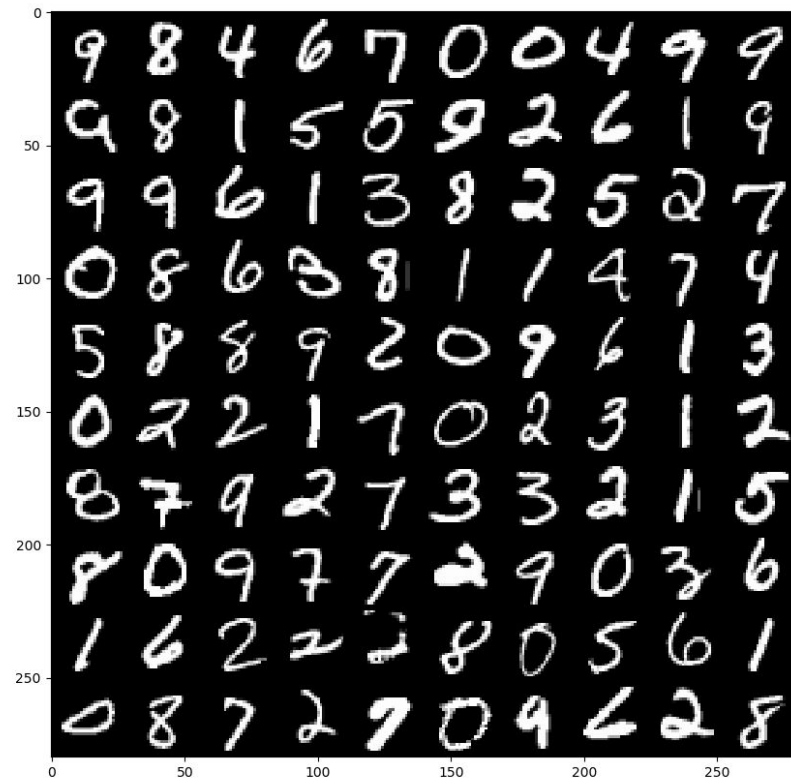
Using autoencoders we get a better result of separation, less overlap and more data clustered at corners. We still see that some of the digits overlap in the middle but overall it looks better than the PCA result.



Reducing the dimension to 32

Original images

-->



Reconstructed images

-->



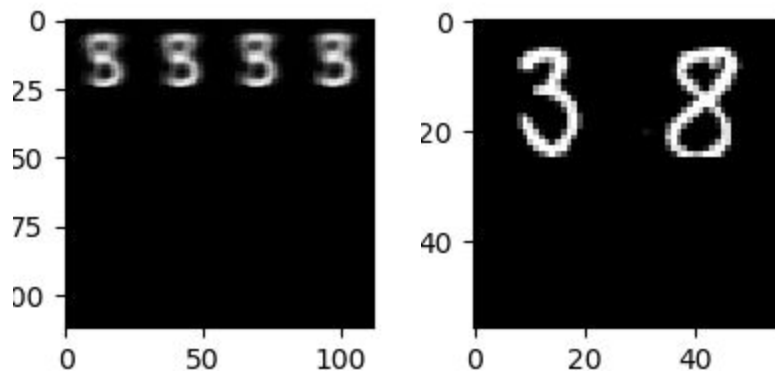
Mean reconstruction error after reducing to 32 dimensions

PCA - 1.680802286e-11

Autoencoders - 0.000445086743831

Latent Space Interpolation

I tried to finish this task but couldn't get it right and didn't have enough time due to the test. I managed to reduce the dimension to 2 using the encoder (as in the previews questions) and manually wrote the interpolation function. But when sending it to the decoder didn't give me good results even after 10,000 iterations



Trying the same reconstruction on the pca latent space :