Programming Assignment - 4
EE5179

Name: C Sneha Sree
Roll No. : EE21S049

A. Hyperparameters

a. Number of epochs = 10
b. Batch Size = 64
c. Learning Rate = 1e-3
d. Optimizer - Adam

B. Models

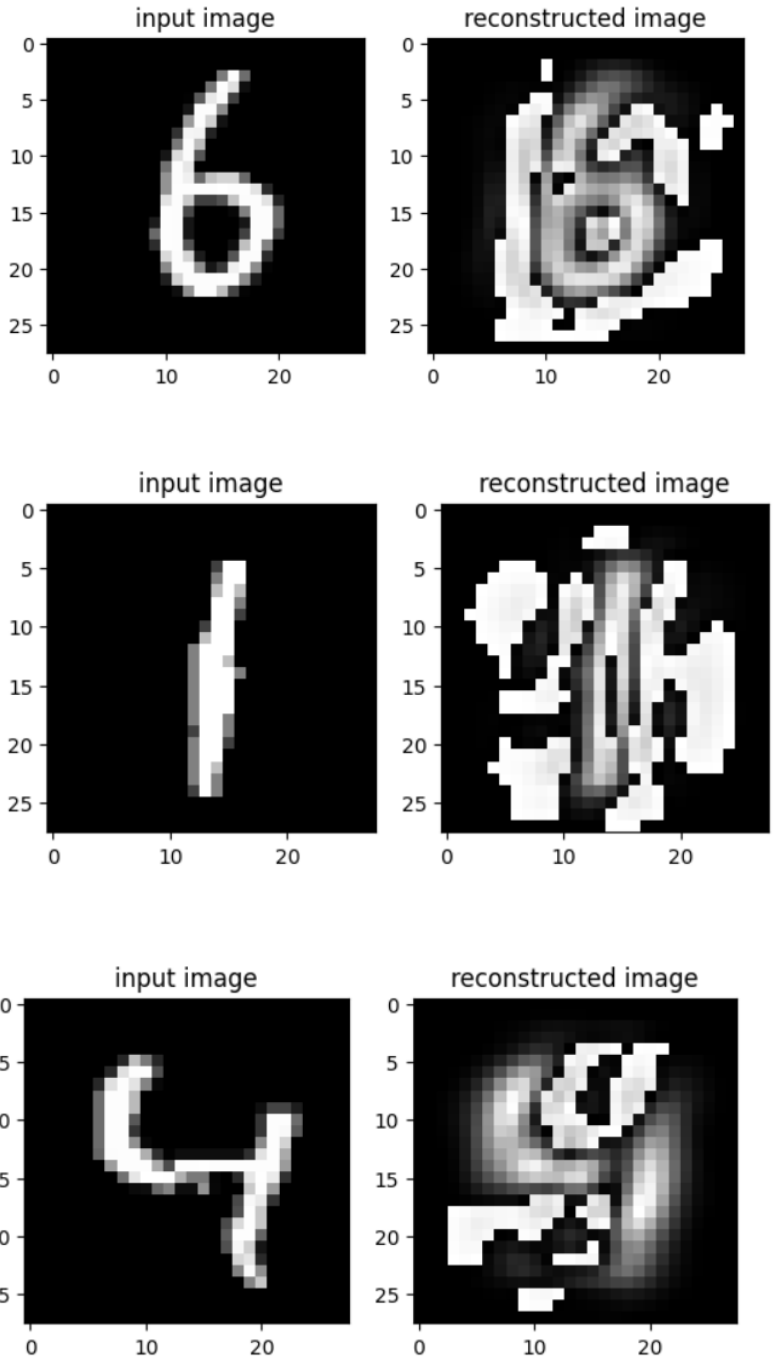
a. PCA
b. Vanilla Autoencoder
c. Autoencoder with one hidden layer
    i. Hidden_layer size - 64, 128, 256
d. Covolutional Autoencoders:
    i. Convolutional Autoencoder with Unpooling
    ii. Convolutional Autoencoder with deconvolution
    iii. Convolutional Autoencoder with both deconvolution and Unpooling

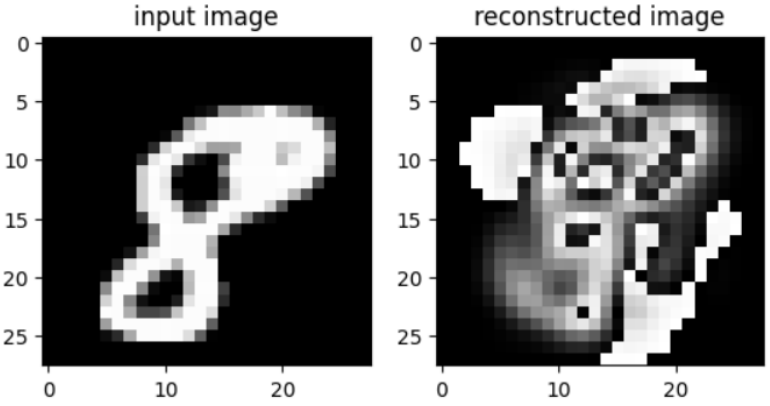
C. Functions : Following are the functions defined for this assignment

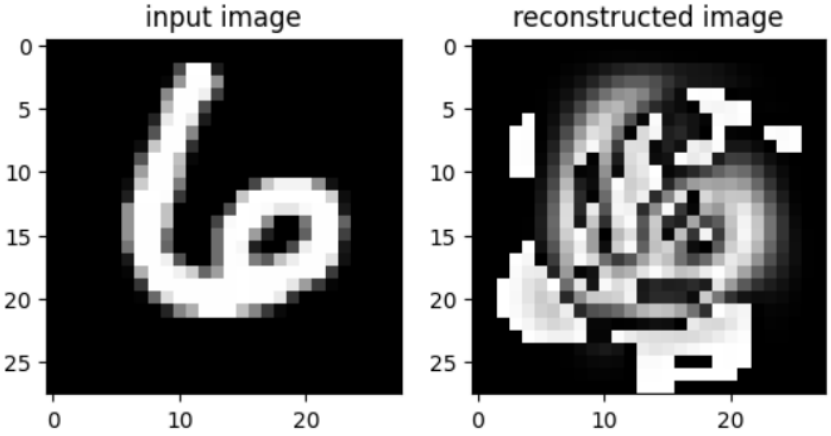
a. *pca_reconstructed_images* : function to plot the reconstructed images obtained using PCA
b. *train:* function to train different models
c. *test:* function to test different models
d. *train_test:* Generalized function to train and test models for the given number of epochs.
e. *plot_reconstructed-image:* function to plot the reconstructed image obtained using different models
f. *plot_losses;* function to plot training and test losses obtained using different models.
g. *encoder_decoder_filters_plots:* function to visualize the encoder and decoder learned filter weights
h. *avg_hl_activations:* function to calculate the average activation of neurons
i. *visualize_activations*: function to visualize the output of activations
j. *add_noise:* function to add random noise of different noise levels to a given input image

1. Comparing PCA and Autoencoders

a. PCA : PCA is performed on MNIST dataset using the first 30 principal components and following are some of the reconstructed images obtained for randomly selected input images.







PCA MSE reconstruction error : 0.018121636925097294

b. Autoencoder : Architecture used for Autoencoder is as follows

• Encoder

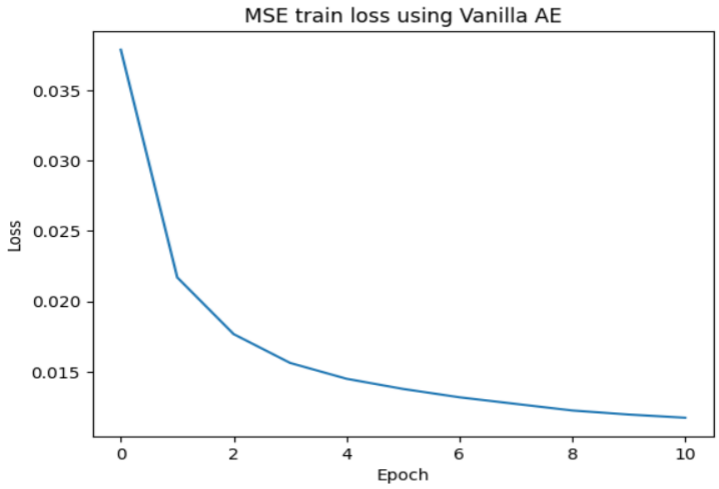
input (784)
fc (512)
fc (256)
fc (128)
fc (30)

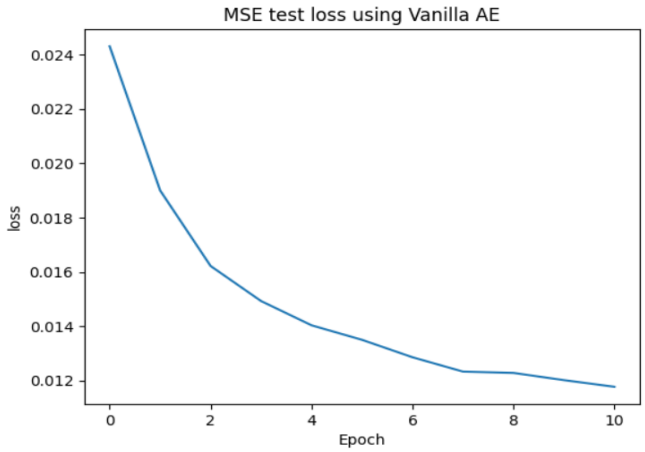
• Decoder

fc (128)
fc (256)
fc (784)

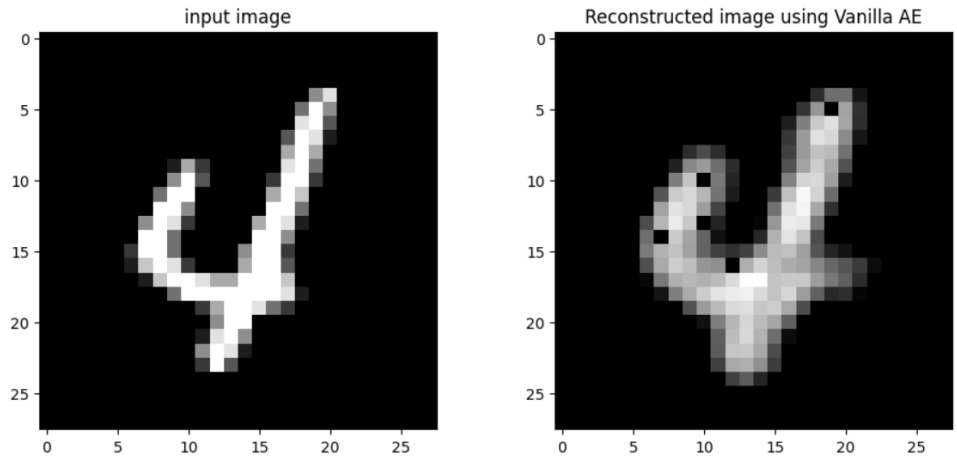
ReLU is used as the activation function

I. Training and Test loss plots :





II. Reconstructed image using this autoencoder for a random MNIST input :



III. MSE Reconstruction error for Vanilla AE is 0.011768510565161705

Observations:

1. As PCA is reconstructing the image after projecting the data into a 30 dimensional space having high variance from a 784 dimensional space, the reconstructed images are not so smooth but we can visually still make out the digit.
2. Reconstructed images using Autoencoder are much better compared to PCA as the contours of the input image are almost there.
3. The reconstruction error of PCA is more than that of the Autoencoder
4. As Autoencoder uses nonlinear activations the performance of Autoencoder is much better than PCA

2. Experimenting with hidden units of varying sizes

Architecture:

Encoder:

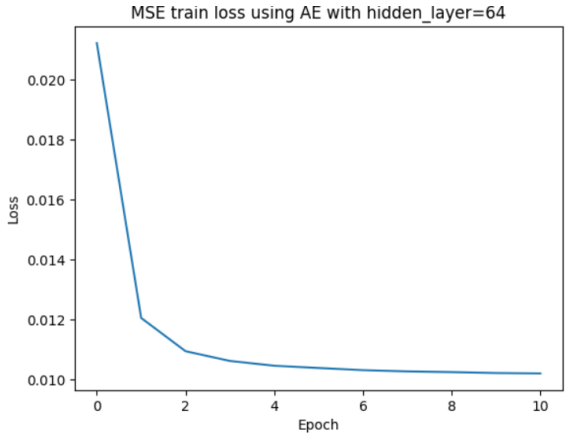
input (784)
fc (hidden_dimension)
ReLU
encoded

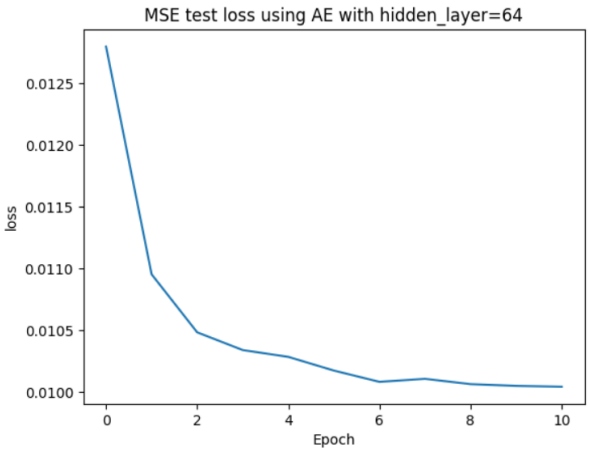
Decoder:

encoded
fc(hidden_dimension)
ReLU
decoded(784)
ReLU
reconstructed_image(784)

I. Training and Test loss plots for different hidden layer size

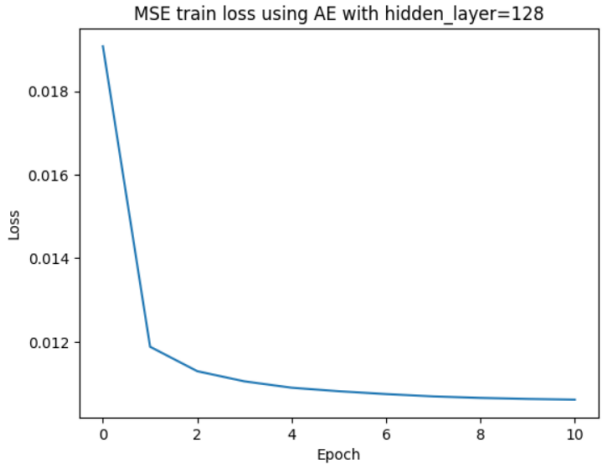
A. h=64

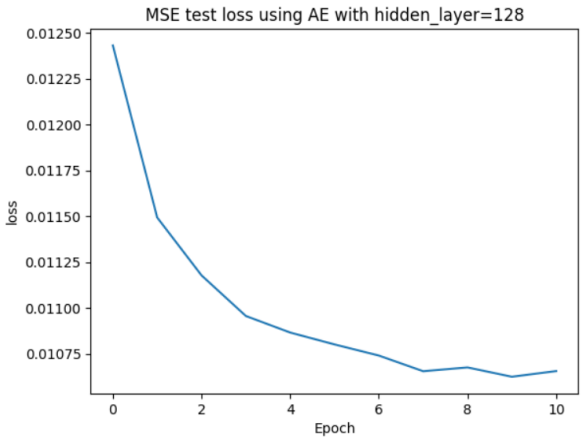




MSE Reconstruction error for AE with hidden layer size 64 is 0.10811023414134979

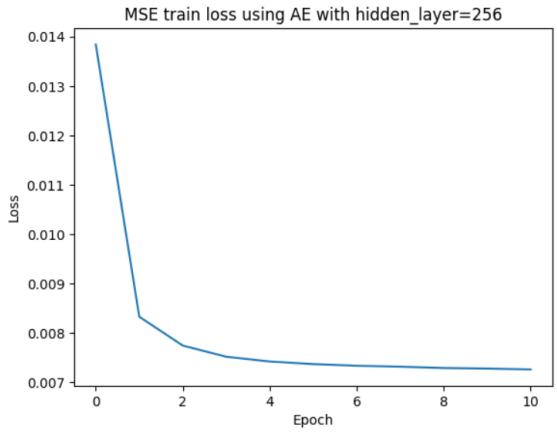
B. h=128

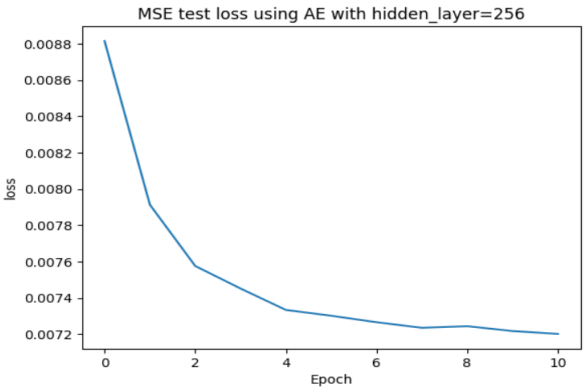




MSE Reconstruction error for AE with hidden layer size 128 is 0.1090042516589164

C. h=256



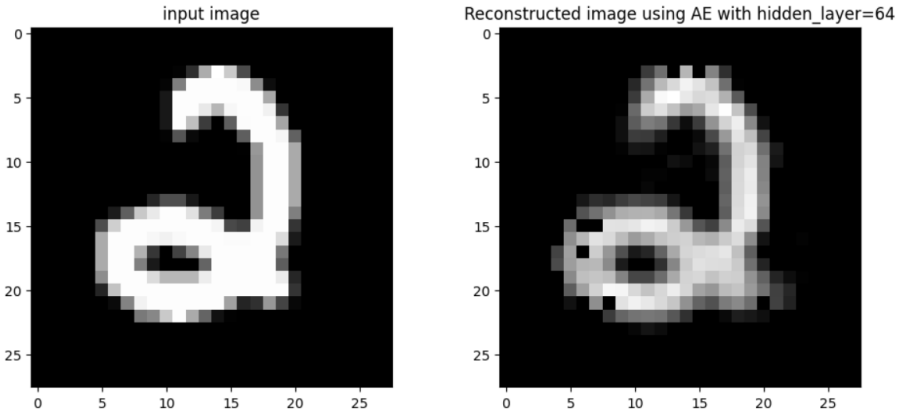


MSE Reconstruction error for AE with hidden layer size 256 is 0.10745961964130402

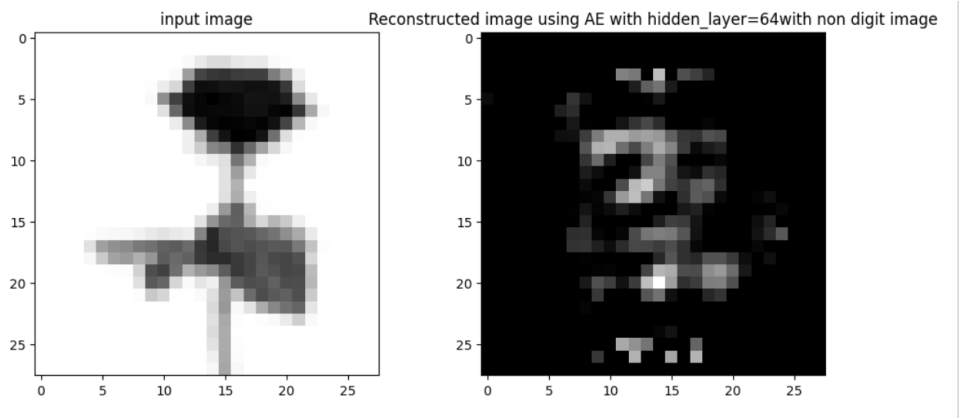
II. Reconstructed Images:

a. h=64

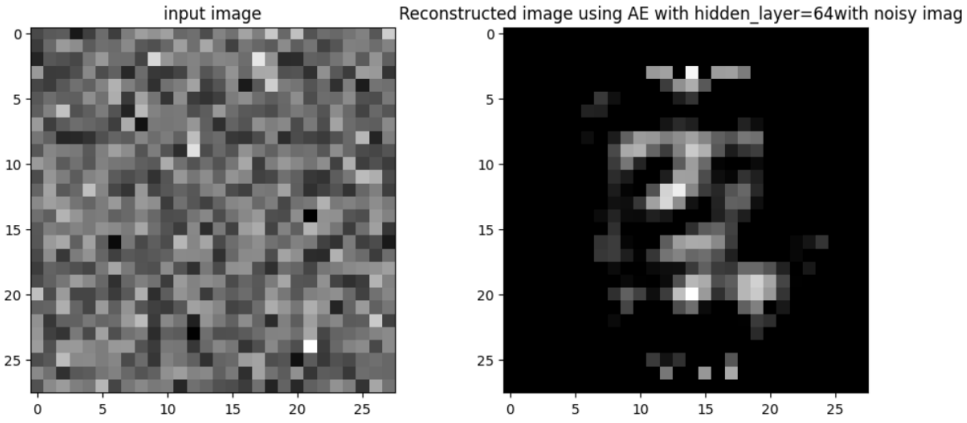
i. MNIST digit image



ii. Non-digit image

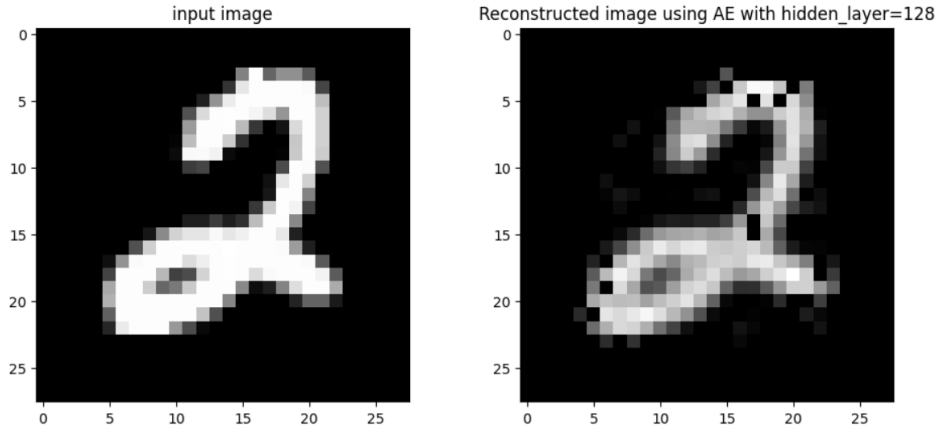


iii. Noisy image

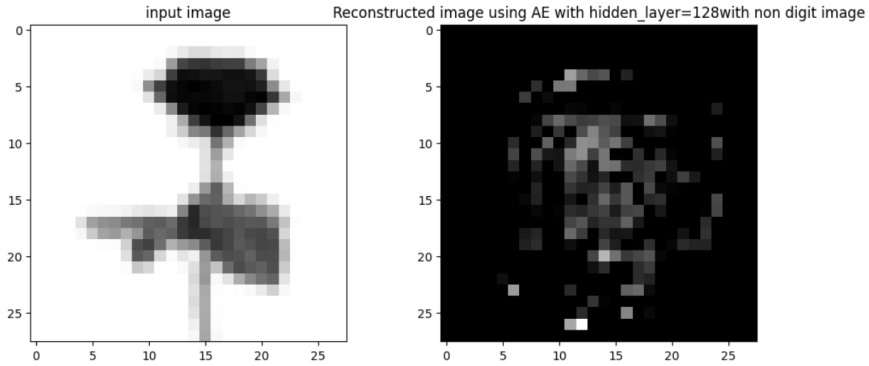


b. h=128

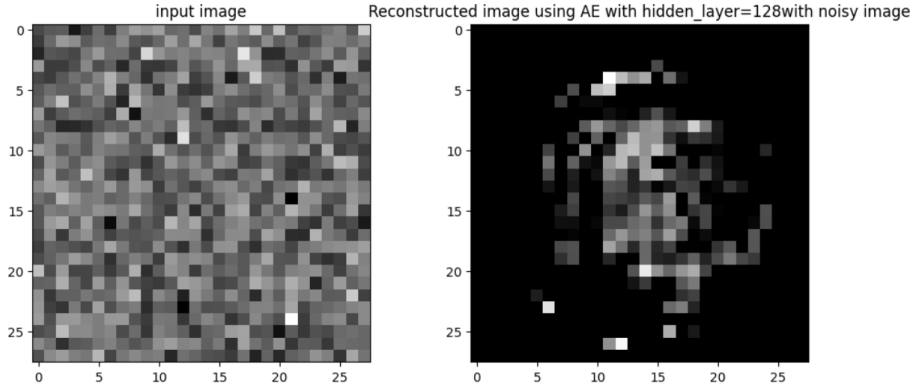
i. MNIST digit image



ii. Non-digit image

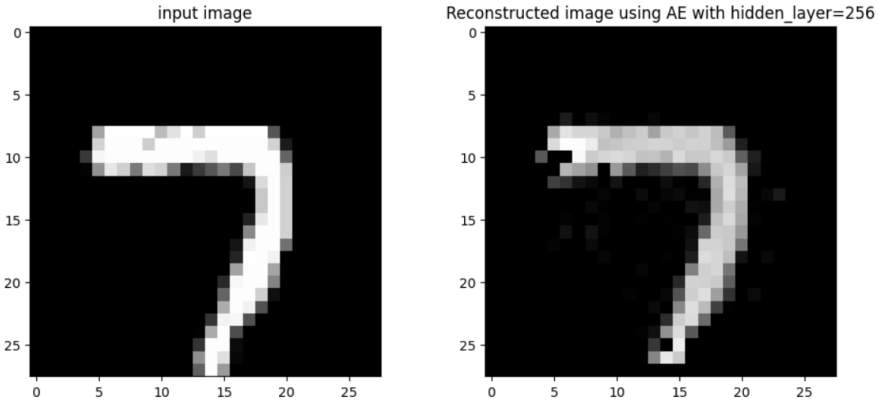


iii. Noisy image

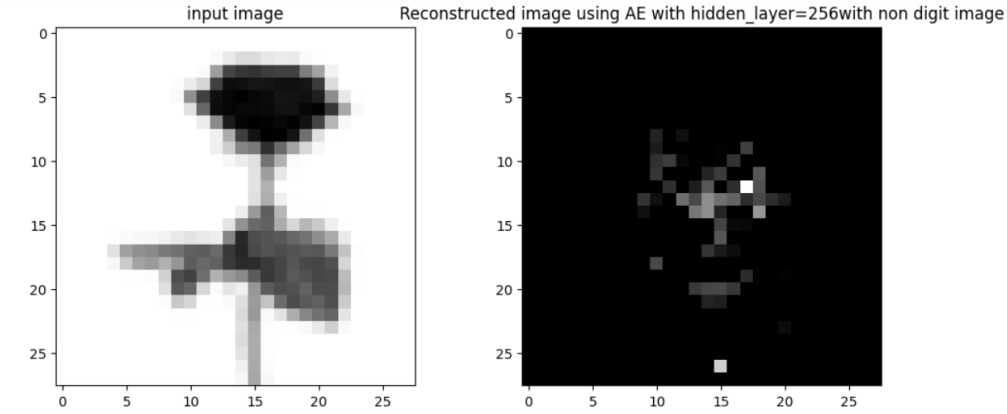


c. h =256

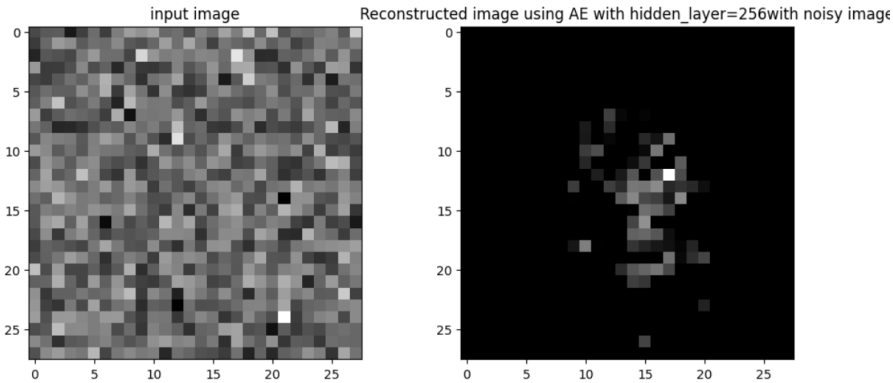
i. MNIST digit image



ii. Non-digit image



iii. Noisy image



Observations:

1. As the size of the hidden layer increases the AE performs better, which can be observed from the reconstructed image for the input MNIST digit images for all three models.
2. For both the non digit image and noisy image as input the model output is just noise as the model is trained on the MNIST dataset but not on any random images.
3. The MSE reconstruction error has not been improved much as the size of hidden layer increases but visually the reconstructed image of AE with hidden size = 256 produced better results.

3. Sparse Autoencoders:

Architecture:

Encoder:

Linear(784,961)
ReLU()

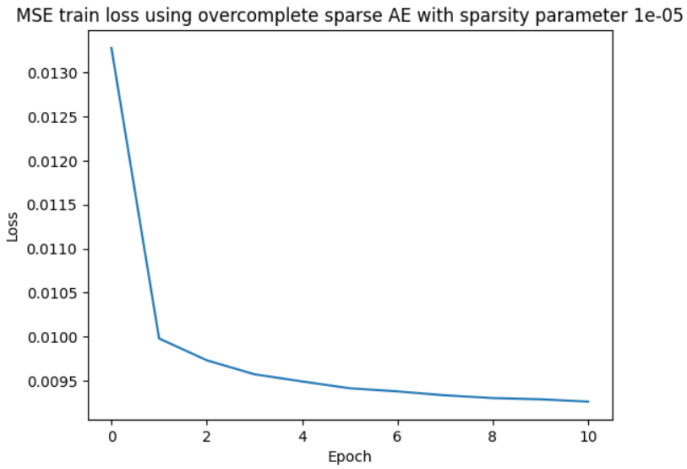
Decoder:

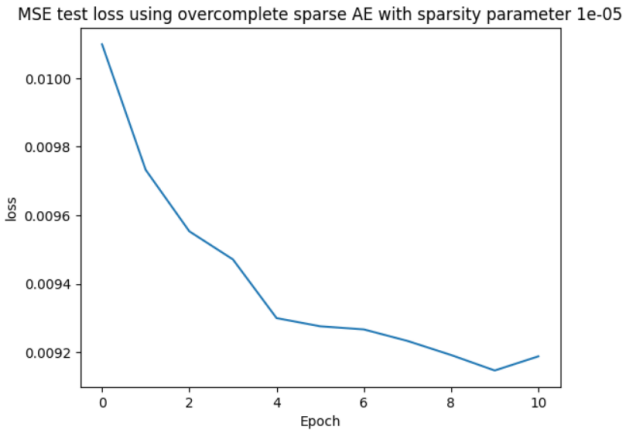
Linear(961,784)
ReLU()

The hidden layer size is chosen as 961 so that the designed AE is an overcomplete AE
An L1 regularization with different regularization parameters has been added to the loss.

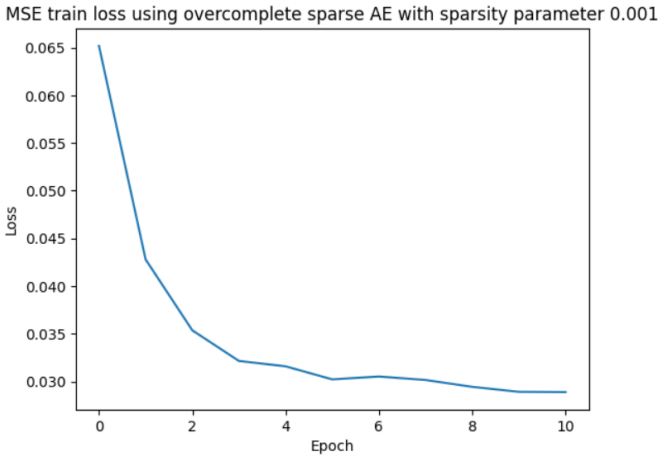
I. Training and Test loss plots:

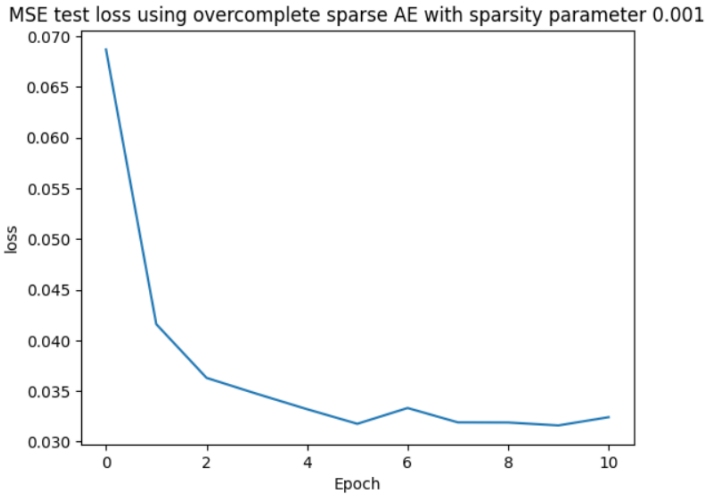
a. Lambda_reg = 1e-5



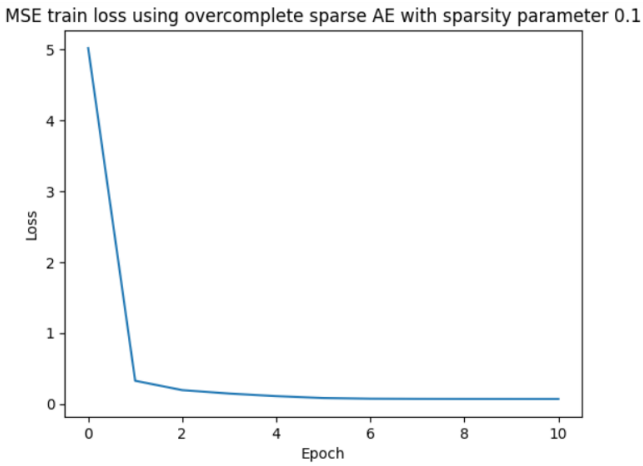


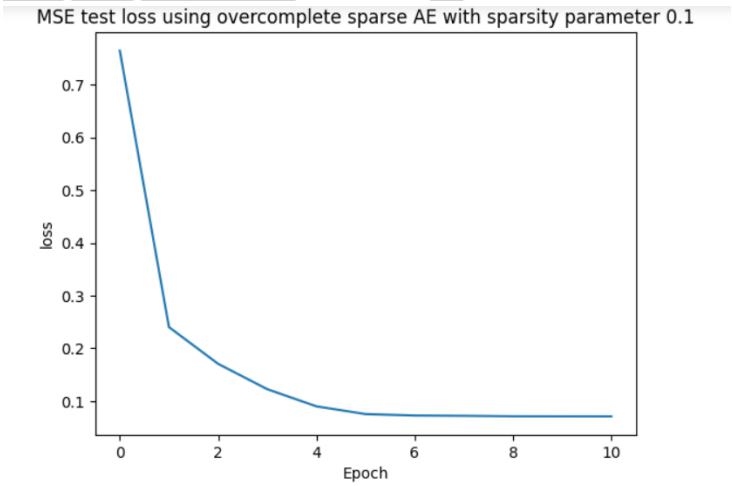
b. Lambda_reg = 1e-3





c. Lambda_reg = 1e-1

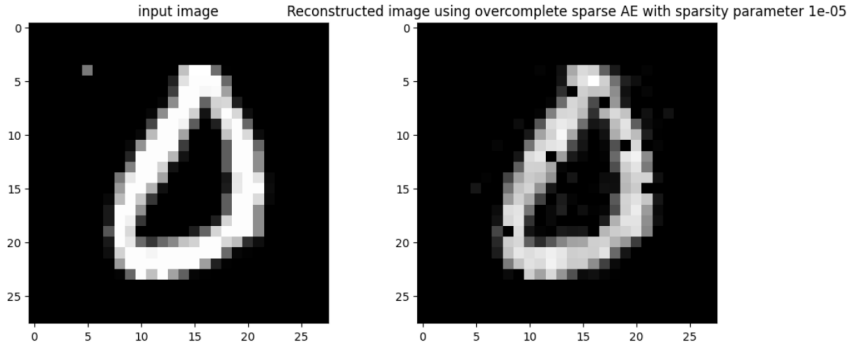




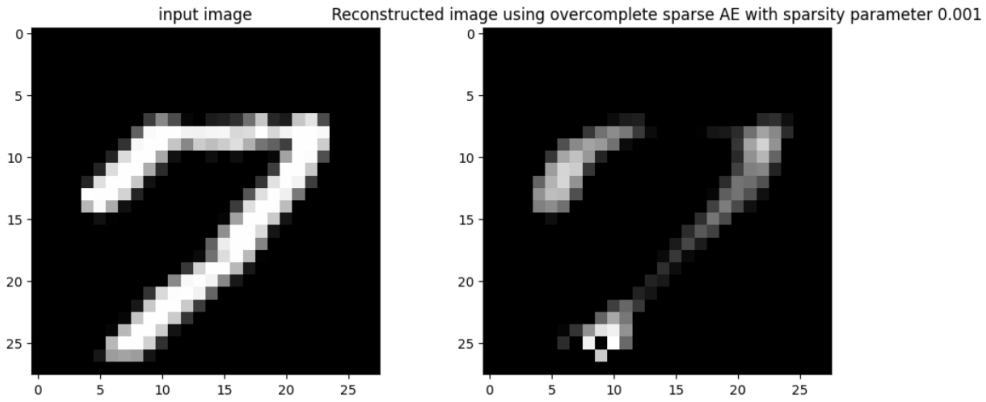
II. Reconstructed images:

Reconstructed images for a random MNIST digit input using sparse AE with different sparsity parameters.

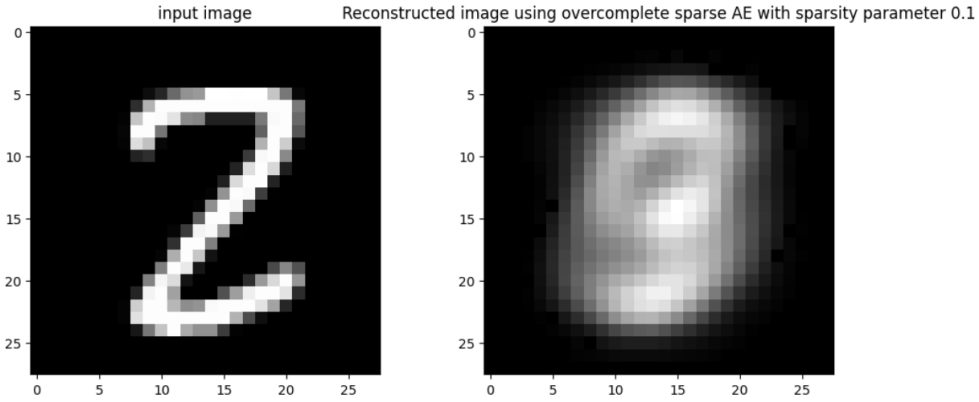
a. Lambda_reg = 1e-5



b. Lambda_reg = 1e-3

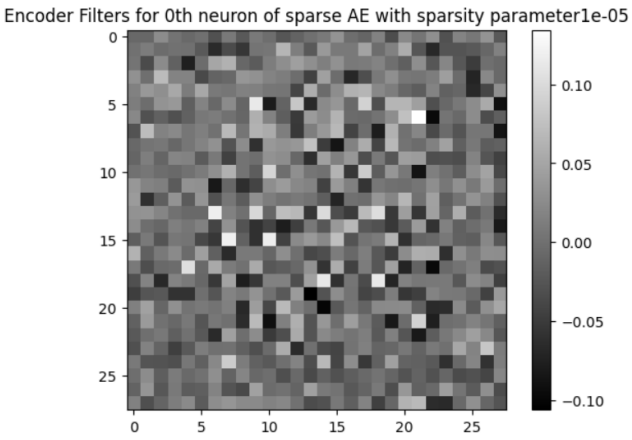


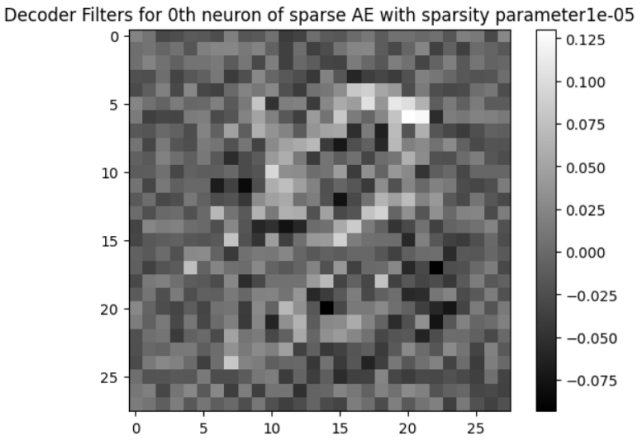
c. Lambda_reg = 1e-1



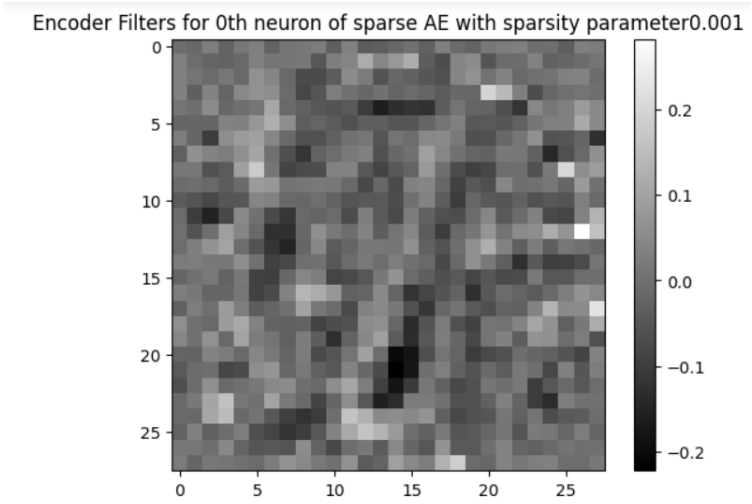
III. Visual Representation of learned filters of 0th neuron

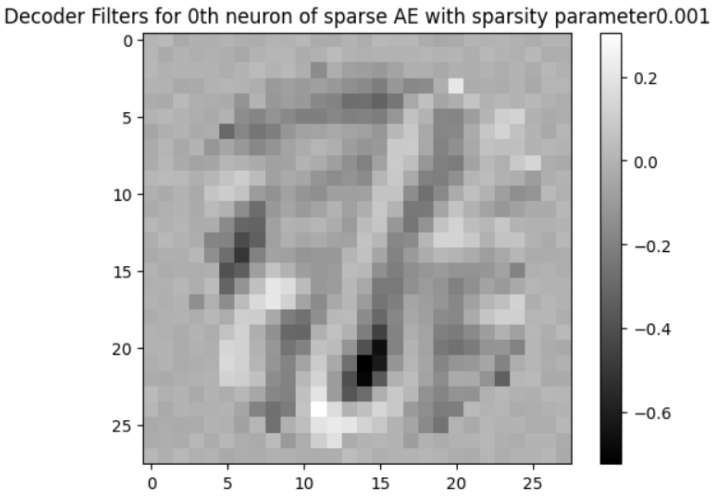
a. Lambda_reg = 1e-5



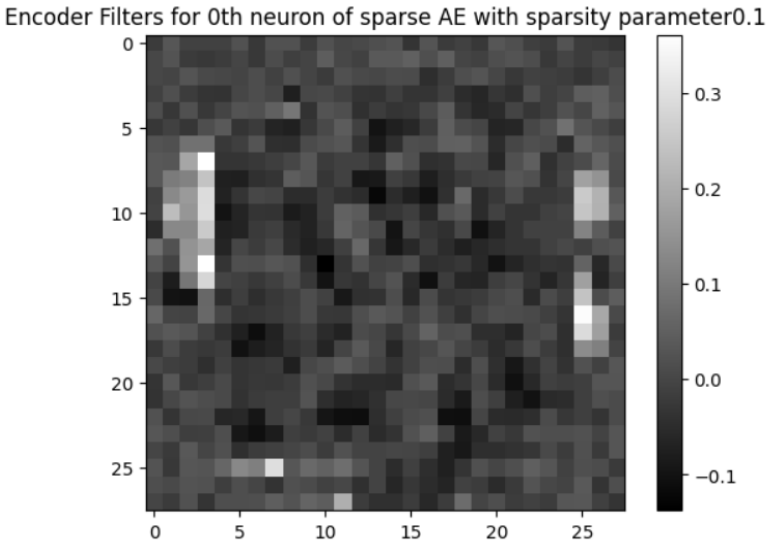


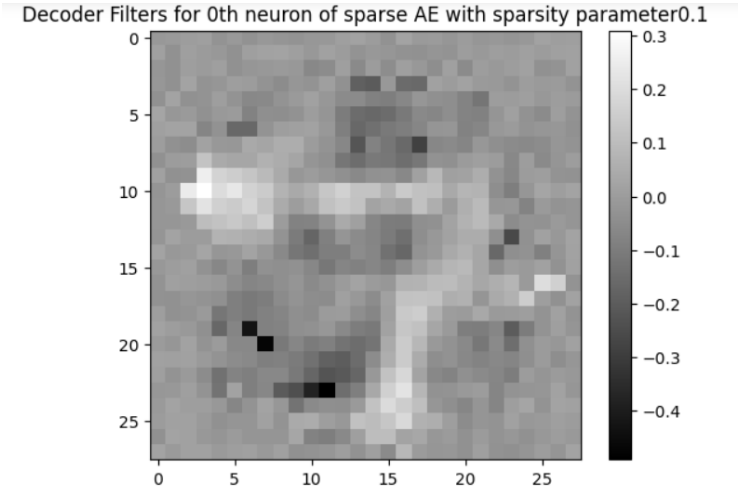
b. Lambda_reg = 1e-3





c. Lambda_reg = 1e-1



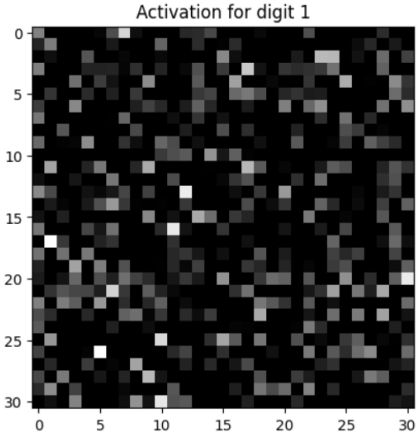


IV. Average Activation

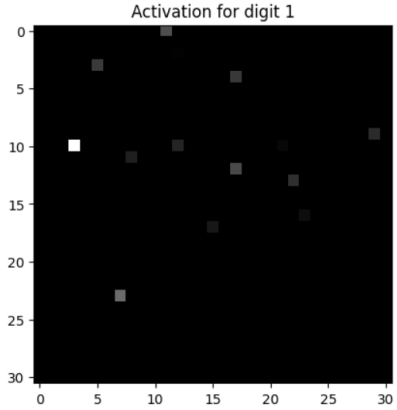
a. The average activation of overcomplete sparse AE with sparsity parameter 1e-05 is 0.17650093831074465
b. The average activation of overcomplete sparse AE with sparsity parameter 0.001 is 0.008566434456593101
c. The average activation of overcomplete sparse AE with sparsity parameter 0.1 is 1.1660724150623433e-06

V. Activations of neurons for a digit with different sparsity parameters:

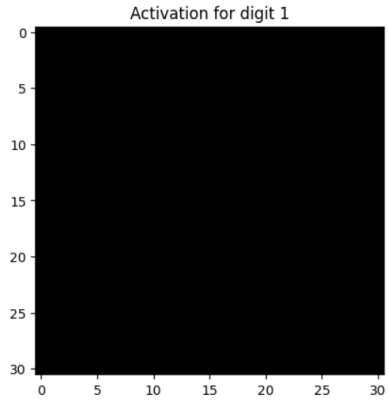
a. Lambda_reg = 1e-5



b. Lambda_reg = 1e-3

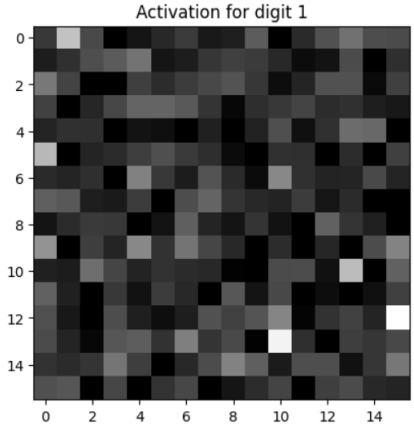


c. Lambda_reg = 1e-1

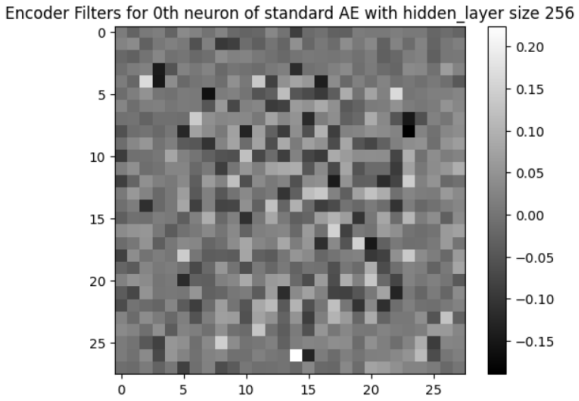


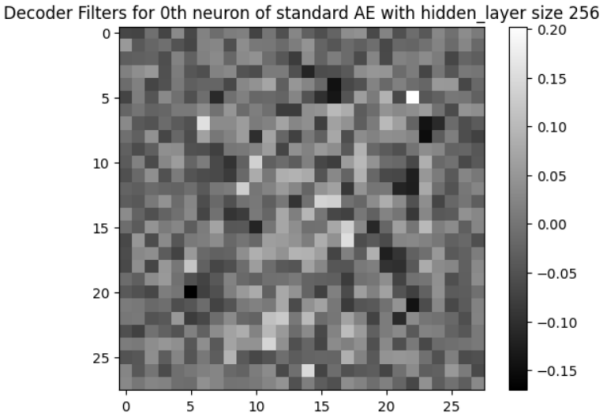
VI. Standard AE with hidden size = 256

a. Activations of neurons for a digit



b. Visual Representation of learned filters of 0th neuron





The average activation of Standard AE with hidden_layer=256 is 0.07100810338357452

Observations:

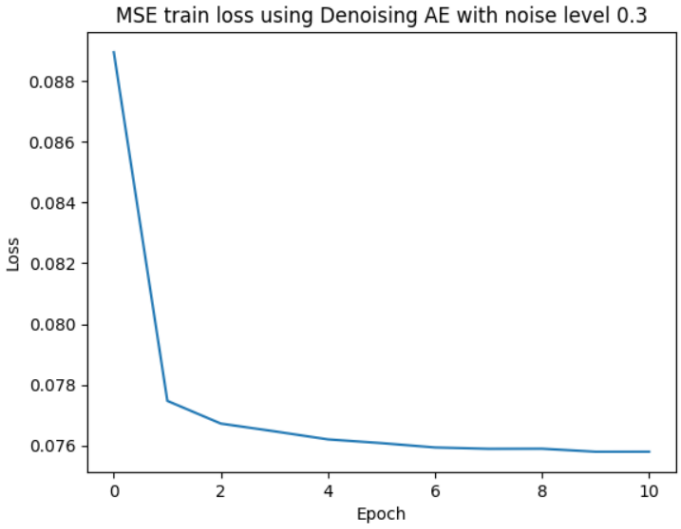
1. As the regularization increases the performance decreases i.e. images are not reconstructed properly
2. Reconstruction is better using standard AE with hidden layer size 256 compared to sparse autoencoders.
3. Losses of regularized sparse AE are more than that of standard AE
4. As the regularization increases fewer number of neurons fire
5. Average activation of neurons of standard AE is more than that of regularized sparse AE
6. Encoder and decoder filters plots look similar to digits which implies that each neuron is looking for features of some digits and they fire when that digit is provided as input whereas for standard AE encoder and decoder filters plots don't give any specific implication.

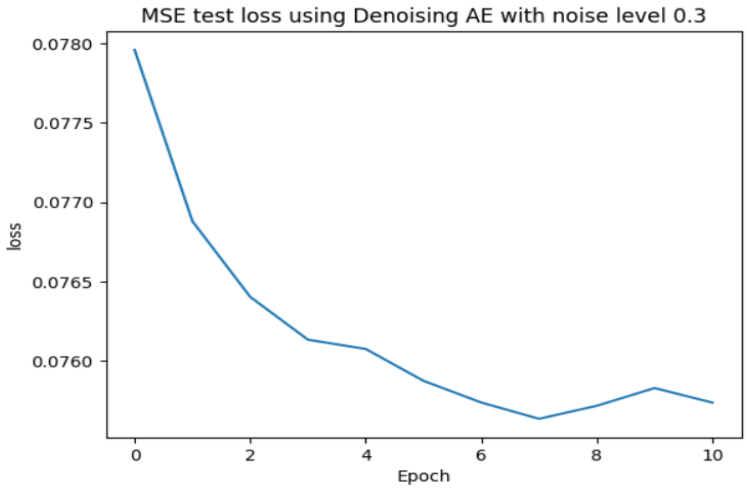
4. Denoising Autoencoders

For the denoising Autoencoder the architecture used is the standard AE with hidden layer size = 256 but the inputs to the encoder are noisy i.e noise is added to the MNIST digit input

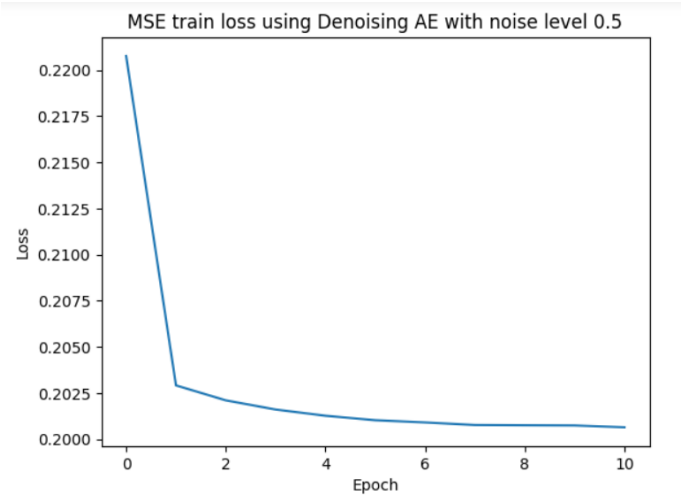
I. Training and Test loss plots for different noise levels

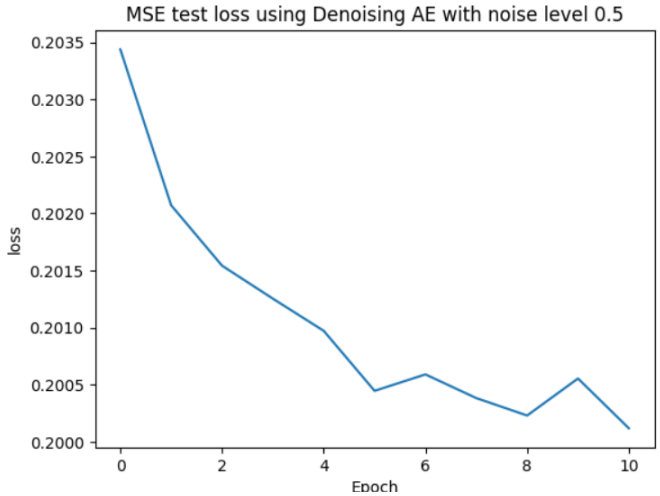
a. Noise level = 0.3



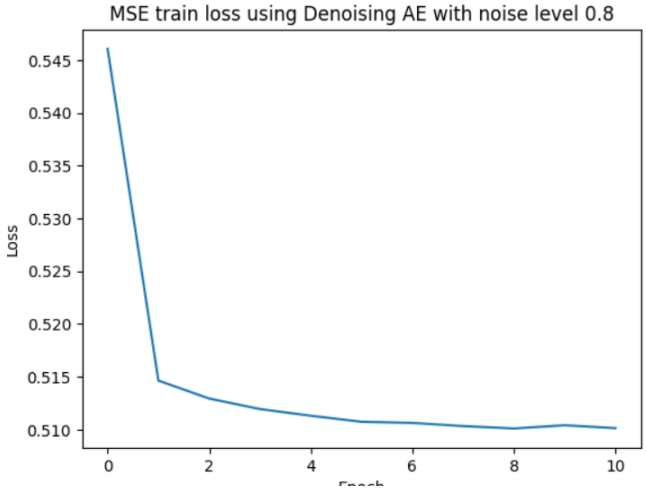


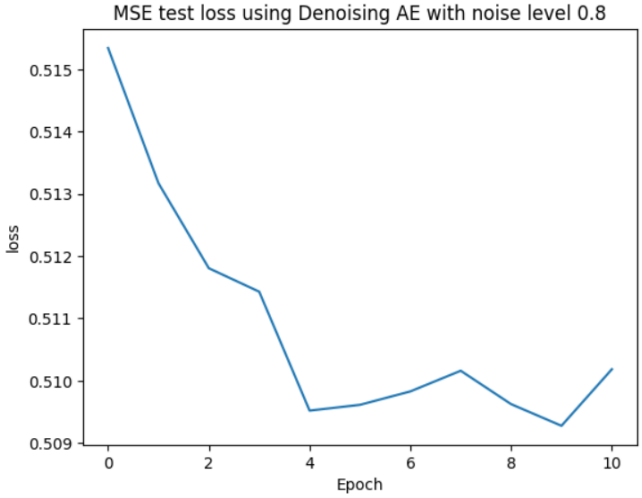
b. Noise level = 0.5



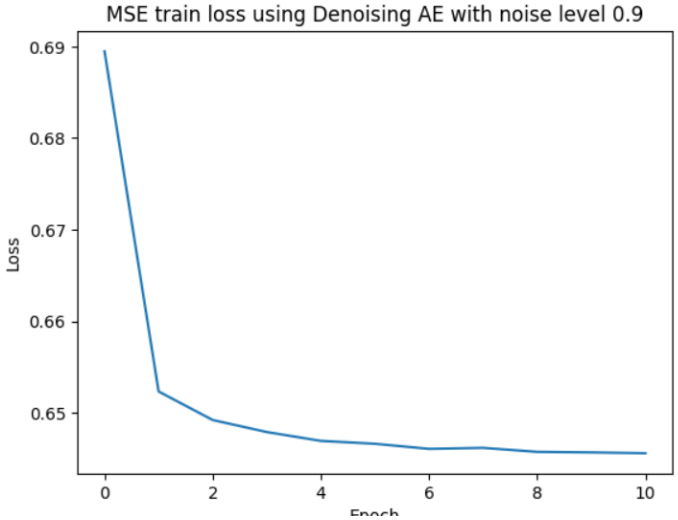


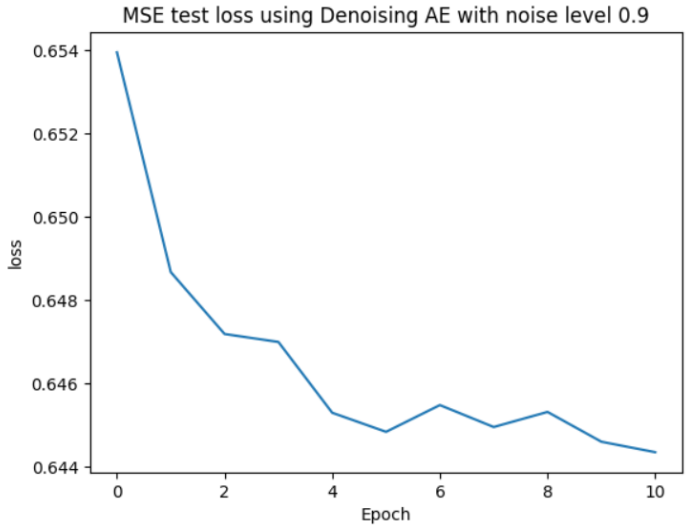
c. Noise level = 0.8





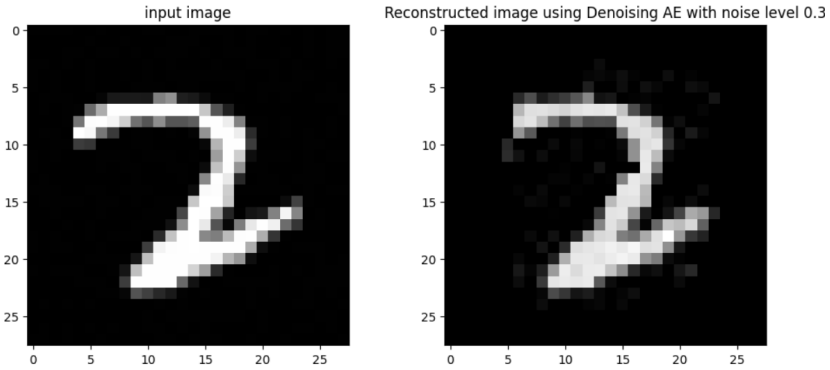
d. Noise level = 0.9



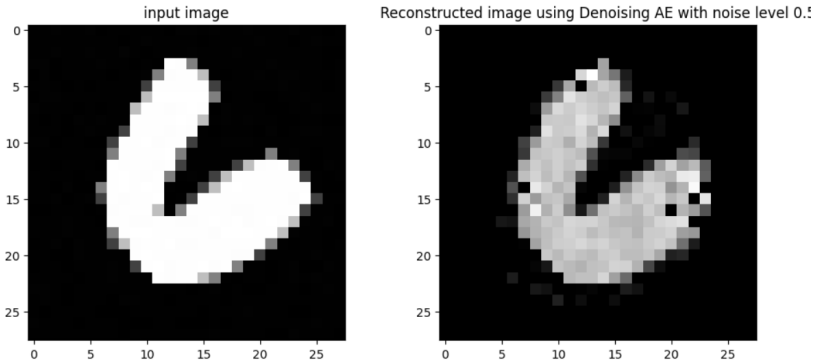


II. Reconstructed images for different noise levels

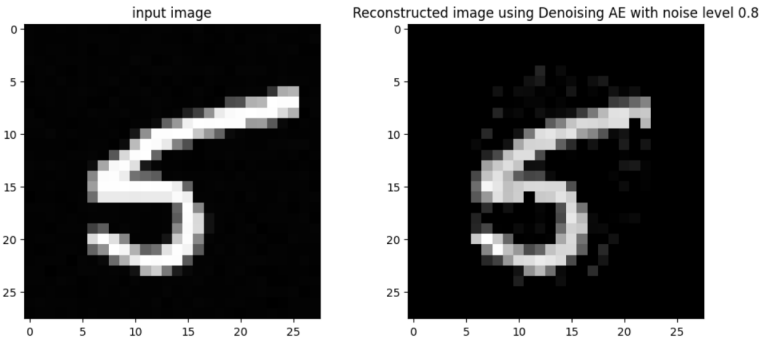
a. Noise level = 0.3



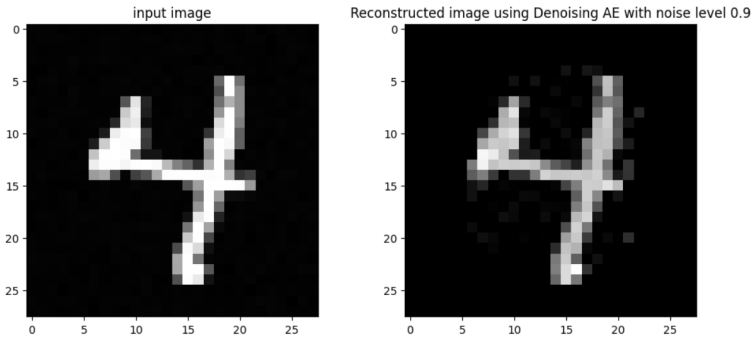
b. Noise level = 0.5



c. Noise level = 0.8

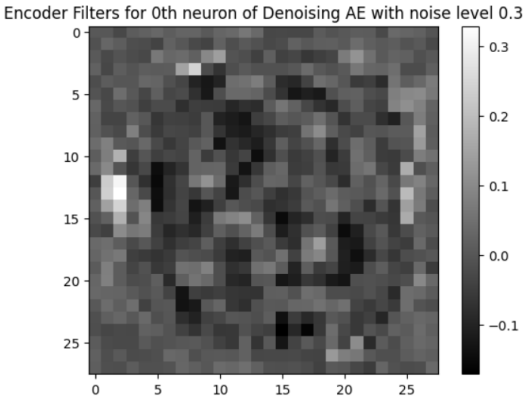


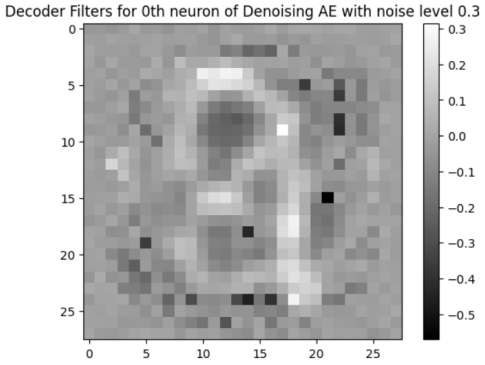
d. Noise level = 0.9



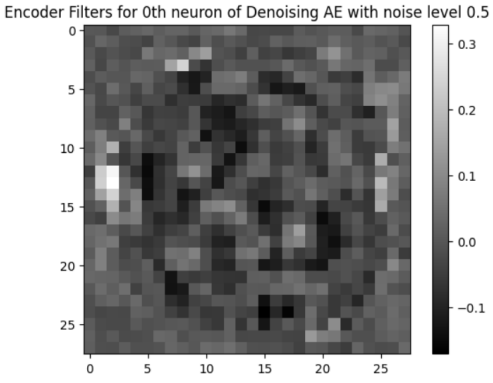
III. Visual Representation of learned filters of 0th neuron

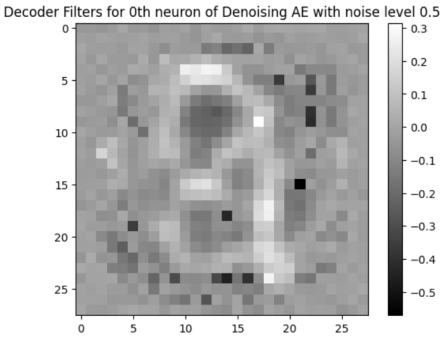
a. Noise level = 0.3



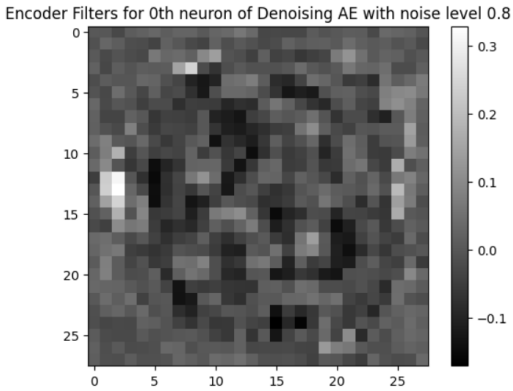


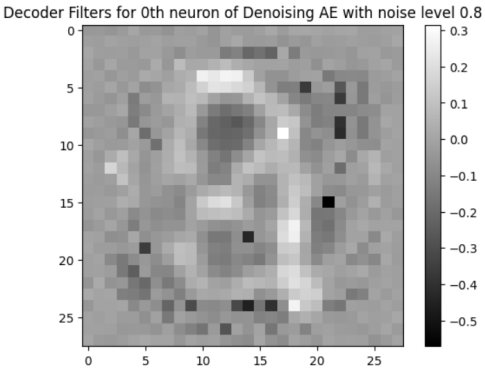
b. Noise level = 0.5



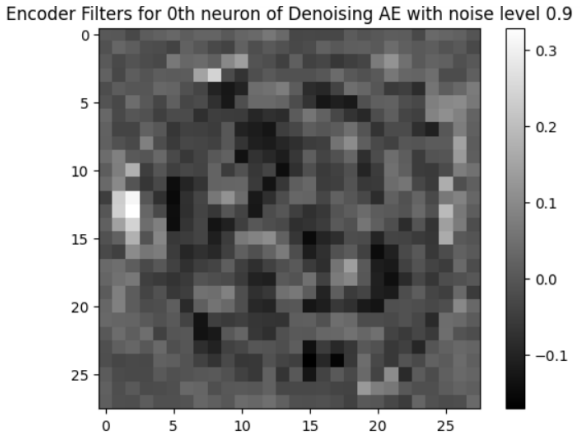


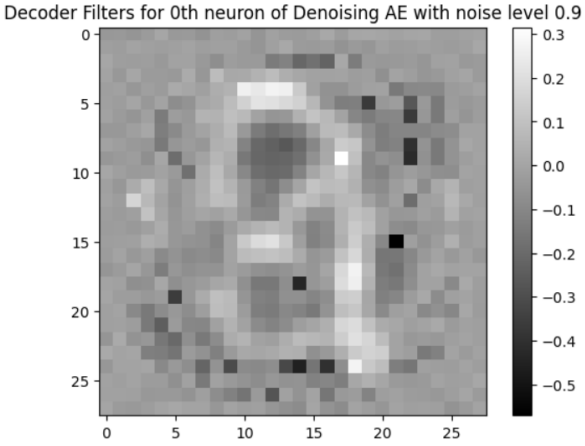
c. Noise level = 0.8





d. Noise level = 0.9





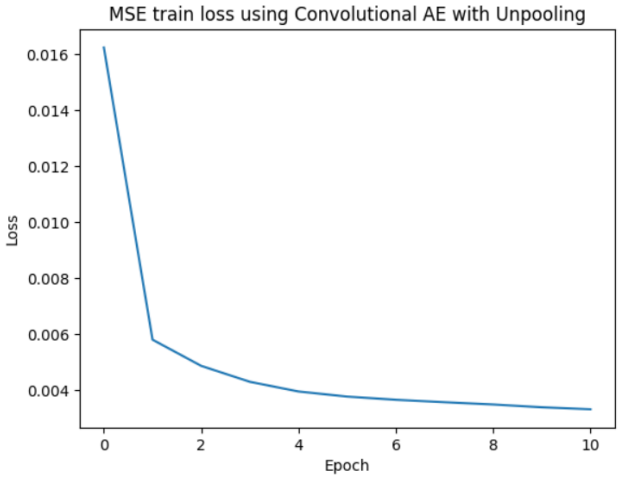
Observations:

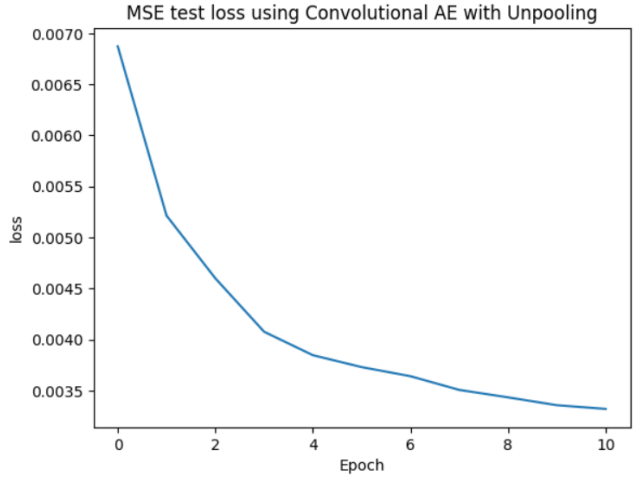
1. As noise level increases reconstruction error increases
2. Encoder and decoder filters resembles digits
3. Denoising AEs are more prone to noise compared to standard AE

5. Convolutional Autoencoders

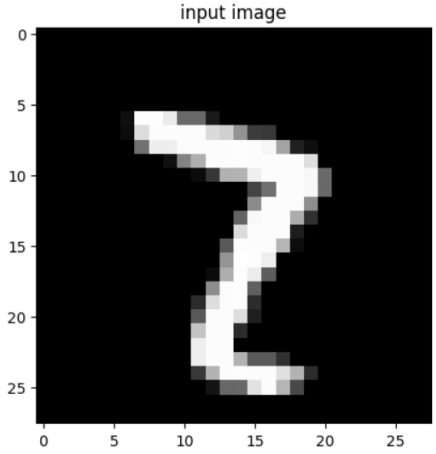
I. Convolutional Autoencoders with Unpooling

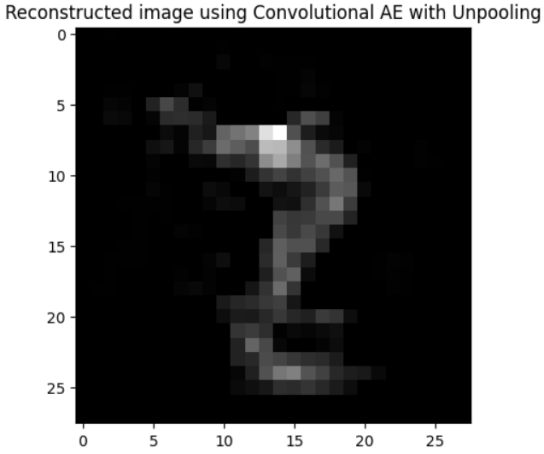
a. Training and Test loss plots





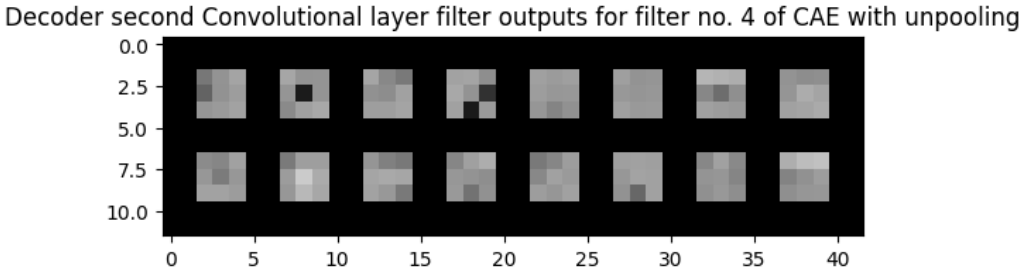
b. Reconstructed image for a given random MNIST input image

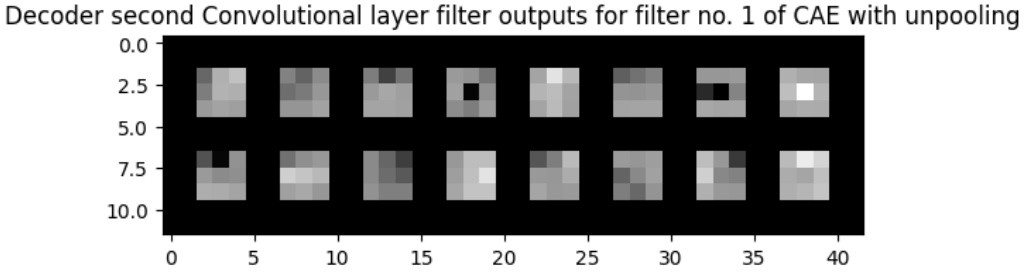


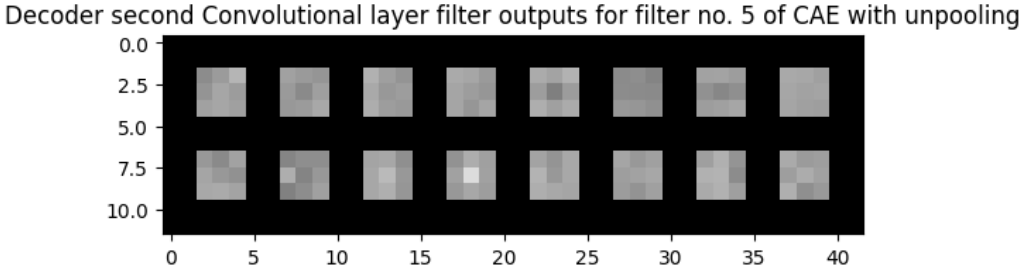


c. Visualizing the decoder weights

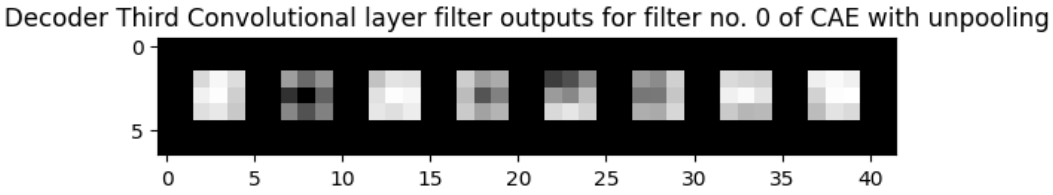
Second convolution layer has 8 - 16x3x3 filters, following are three randomly selected filter outputs out of the 8 filters





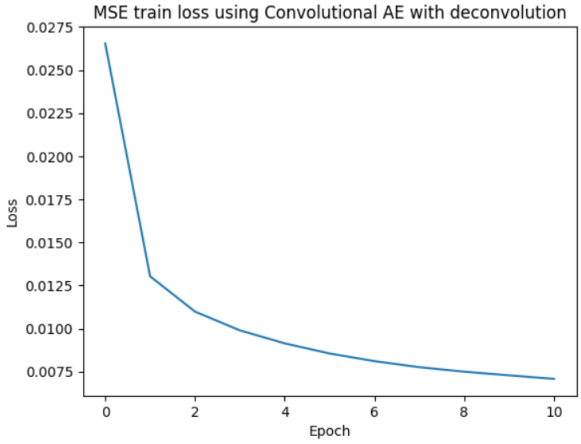


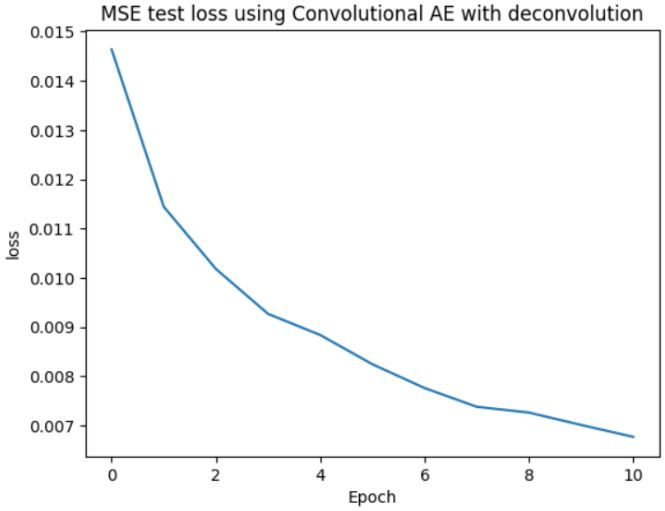
The third convolutional layer has 8-3x3x1 filters, following are those 8 filter outputs.



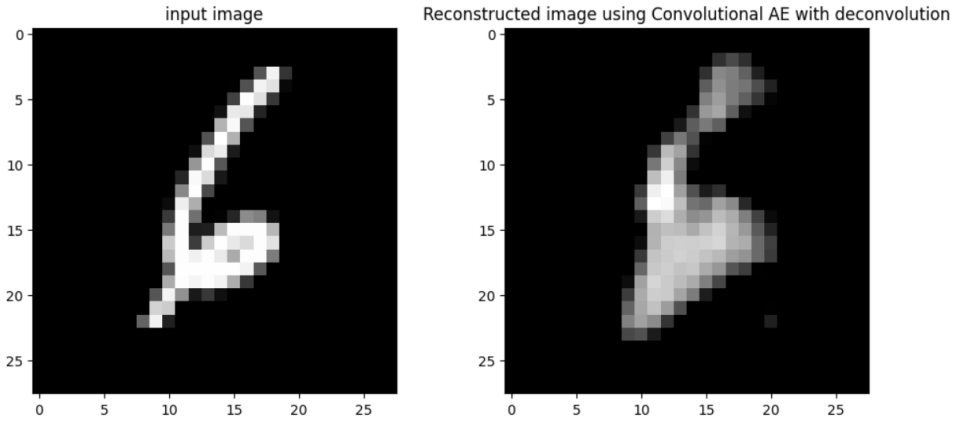
II. Convolutional Autoencoders with deconvolution

a. Training and Test loss plots



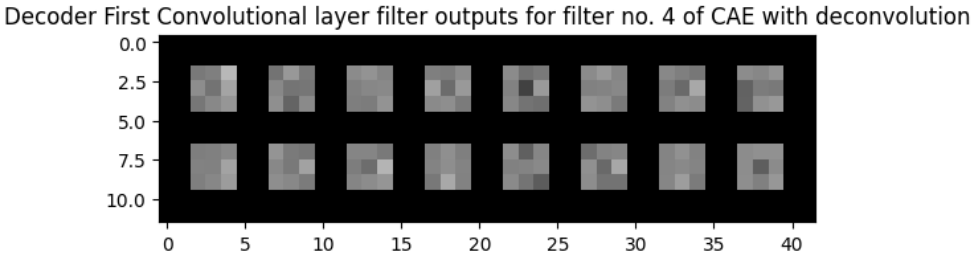


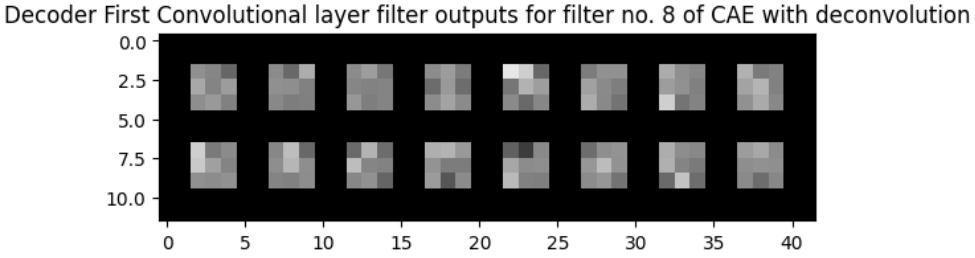
b. Reconstructed image for a given random MNIST input image

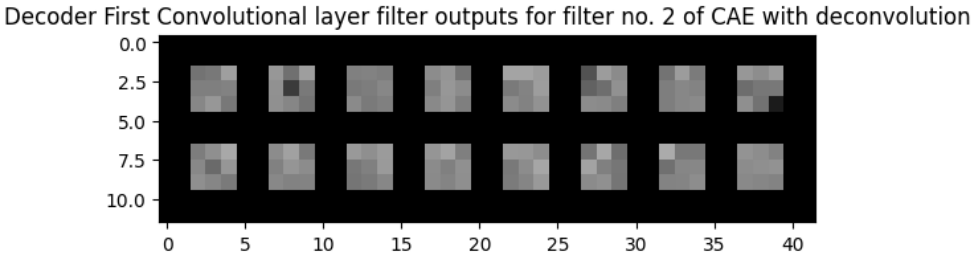


c. Visualizing the decoder weights

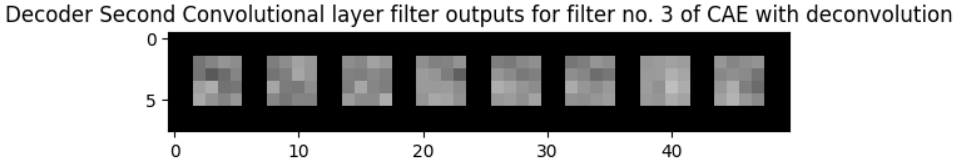
First convolution layer has 8 - 16x3x3 filters, following are three randomly selected filter outputs out of the 8 filters

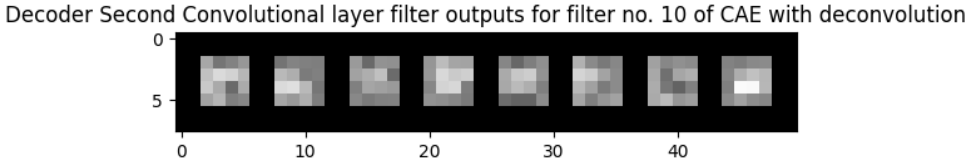


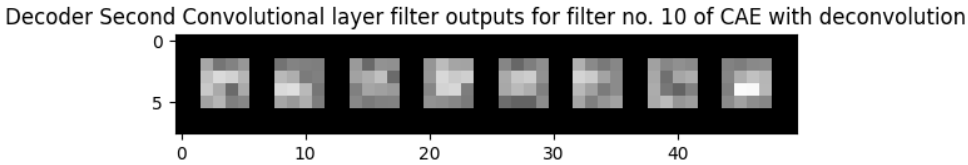




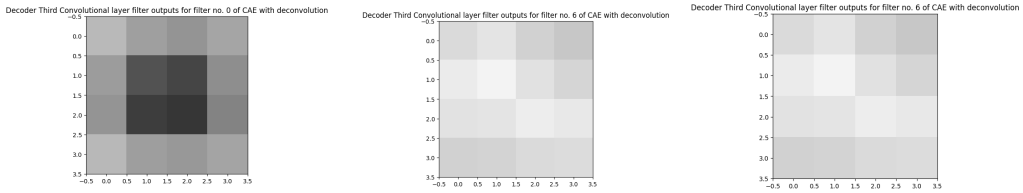
Second convolutional layer has 16 - 8x4x4 filters, following are three randomly selected filter outputs





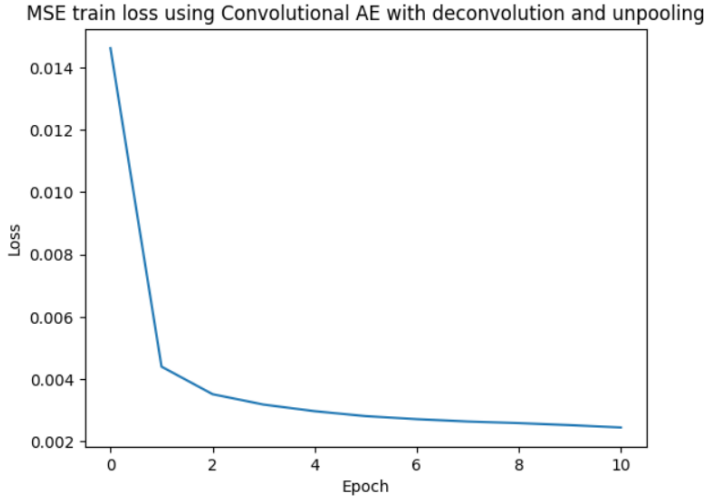


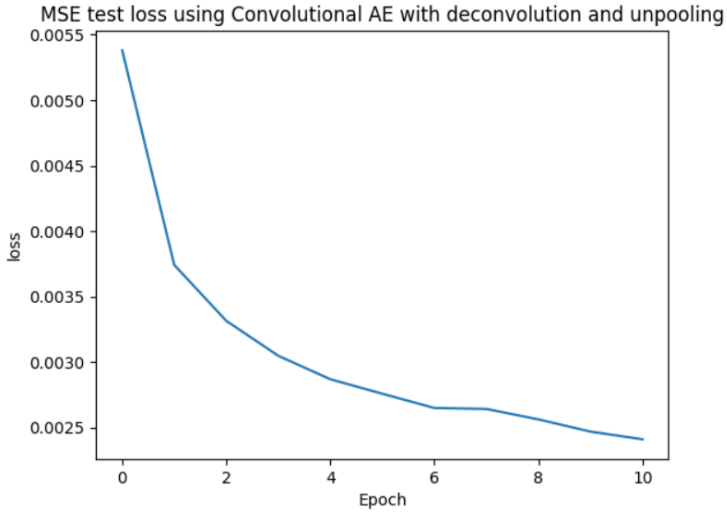
Third convolutional layer has 8- 1x4x4 filters following are three randomly plotted filter outputs.



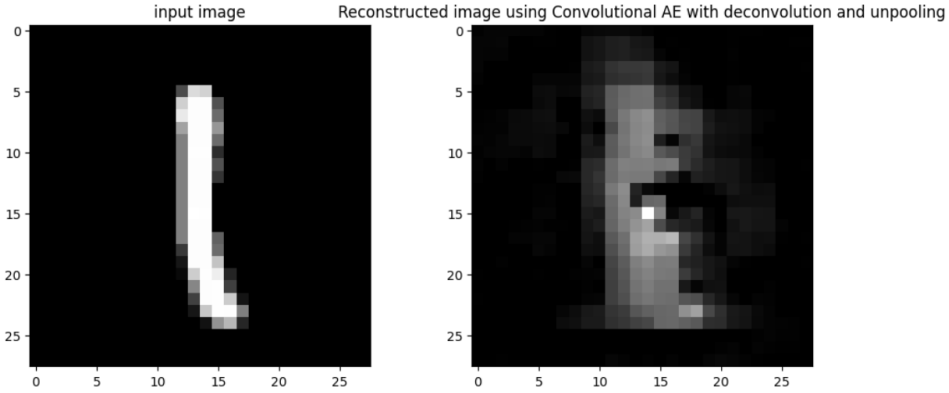
III. Convolutional Autoencoders with deconvolution and Unpooling

a. Training and Test loss plots



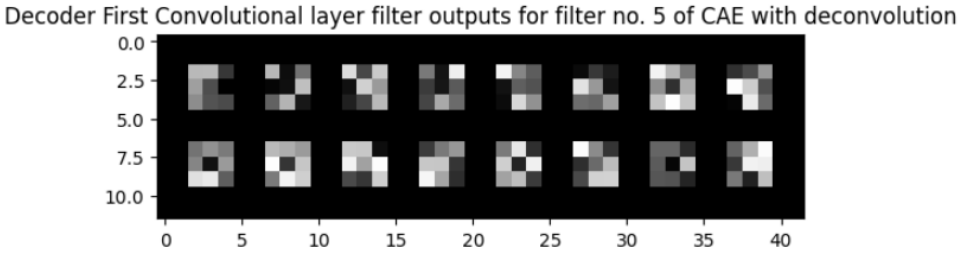


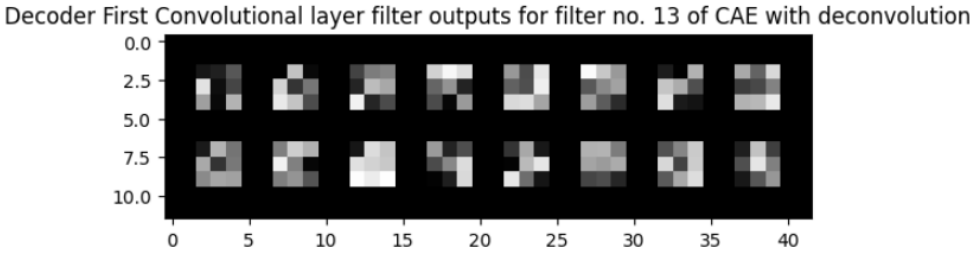
b. Reconstructed image for a given random MNIST input image

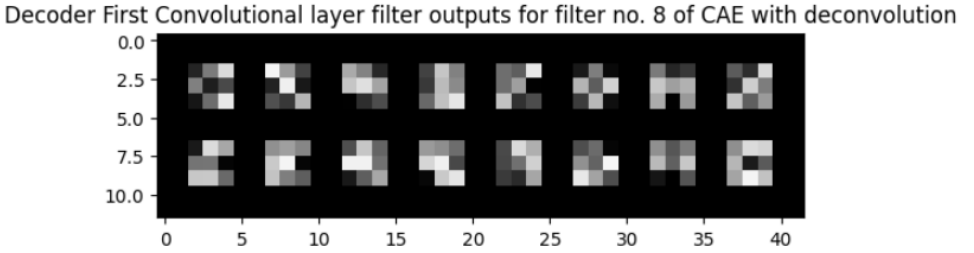


c. Visualizing the decoder weights

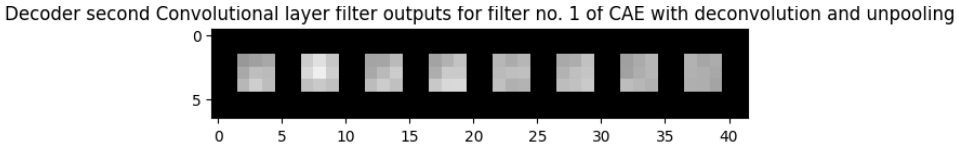
First convolutional layer has 16 - 8x3x3 filters following are three randomly selected filter outputs.

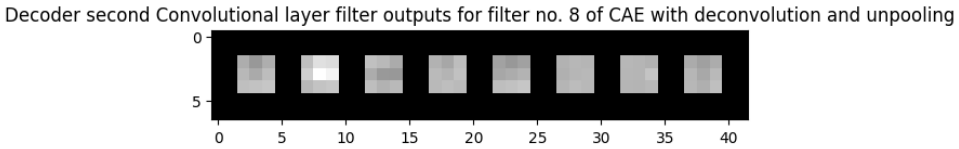


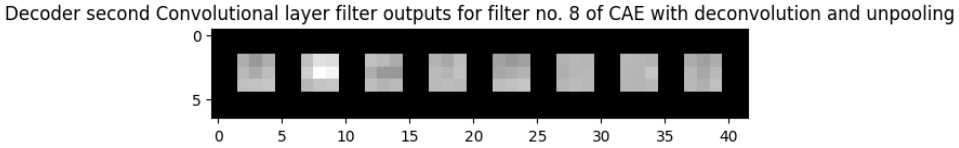


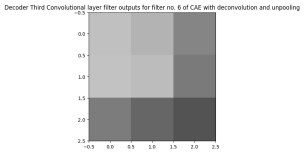


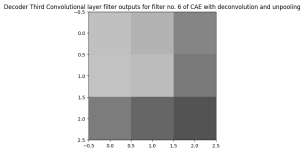
Second convolutional layer has 8 - 1x4x4 filters following are three randomly selected filter outputs.

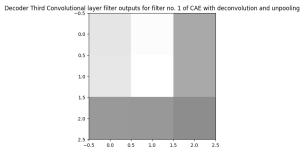












Observations:

1. Performance wise Convolutional AE with unpooling and deconvolution perform much better than the one with only unpooling and one with only deconvolution which can be observed from the reconstruction images.