**ASSIGNMENT** - **8**

# **KARTHIK BATHULA**

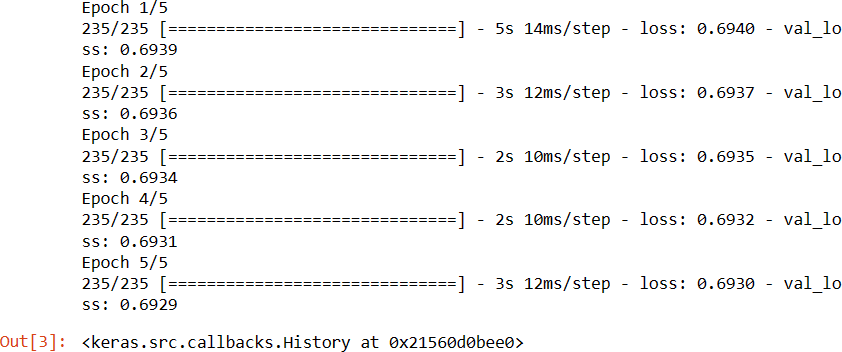
# **700743520**

**GIT-HUB LINK:**

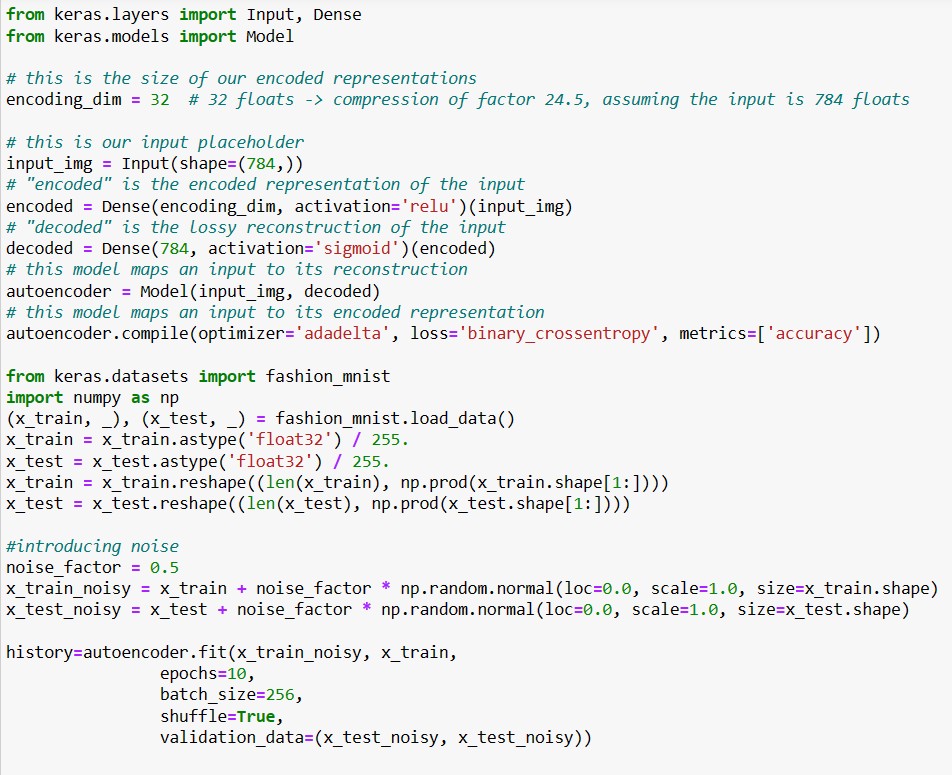
<https://github.com/Karthikbathula25/neuralnetwork_assignment8_INP#neuralnetwork_assignment8_inp>

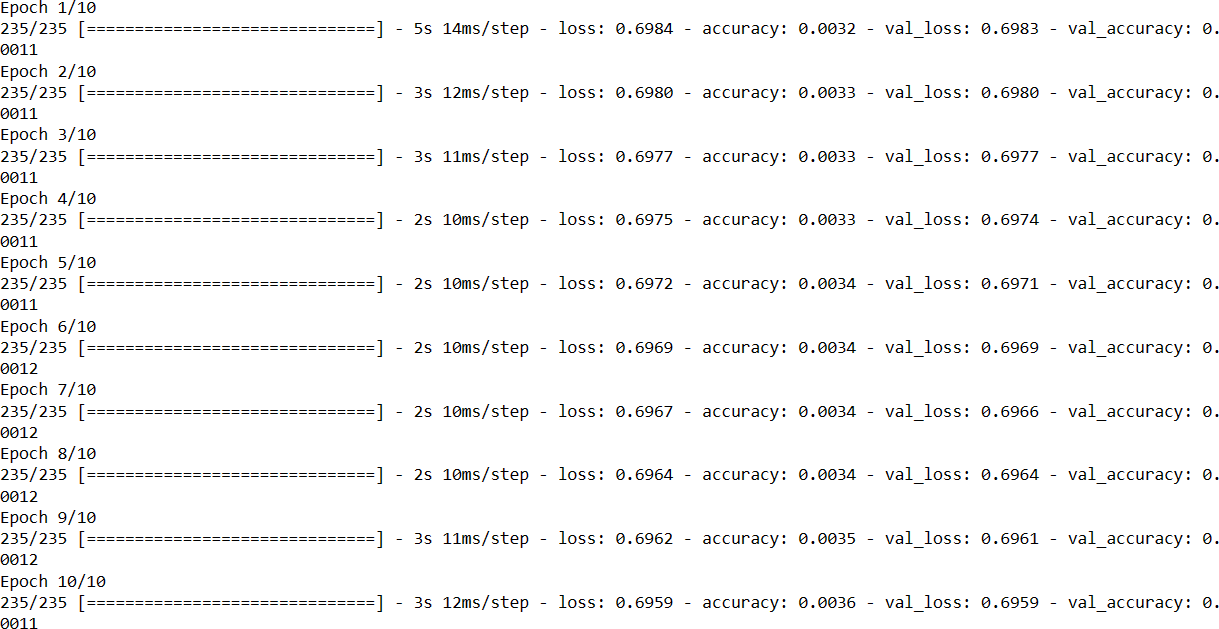
In-class programming:

* First step, we are importing the files as shown below.

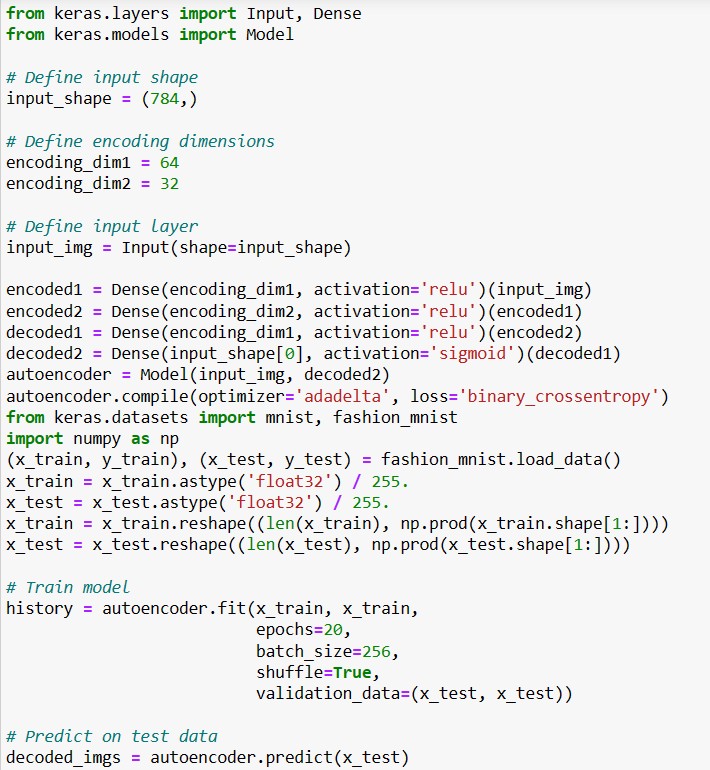


# Denoising Autoencoder





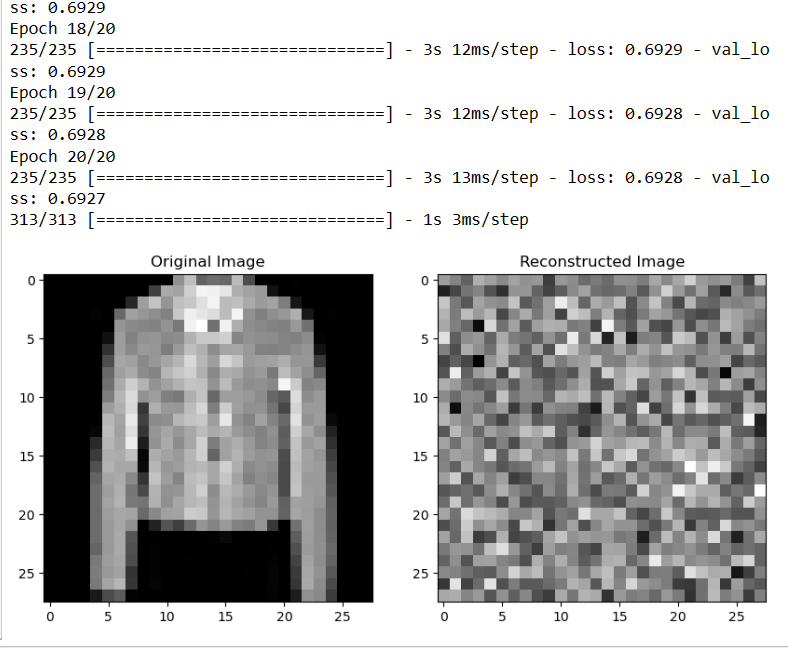
1. Add one more hidden layer to the autoencoder.



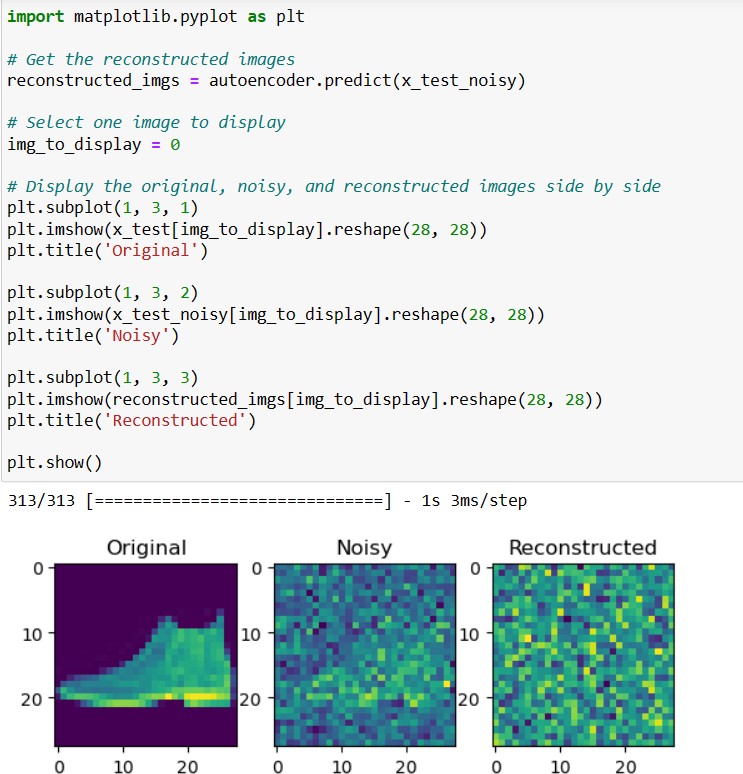
* + The code defines a two-layer autoencoder using Keras for compressing and reconstructing Fashion MNIST images.
  + It loads, preprocesses, and trains the autoencoder to minimize the difference between input and reconstructed images.
  + The code visualizes the quality of reconstruction for a selected test image to assess the autoencoder's performance.



Output:



1. Do the prediction on the test data and then visualize one of the reconstructed version of that test data. Also, visualize the same test data before reconstruction using Matplotlib.



1. plot loss and accuracy using the history object.
   * The code creates a two-part plot using Matplotlib. The first subplot displays training and validation loss over training epochs, and the second subplot displays training and validation accuracy.
   * It uses data stored in history. History to plot these values. The plt.legend() function adds labels to distinguish training and validation data. Finally, plt.show() displays the plot, aiding in visualizing the model's training progress, including loss and accuracy trends.

