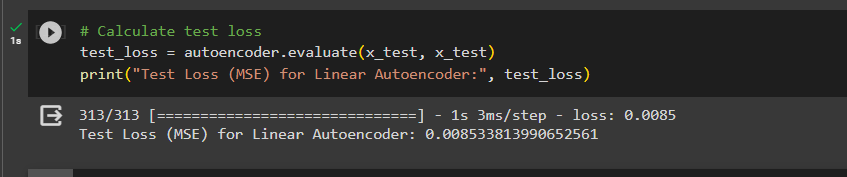
# DL lab 7 -Autoencoders

## Upload the Autoencoder (AE) jupyter notebook file (i.e., lab\_7\_AE\_FFNN.ipynb) to google colab root directory.

### Write the code implementation to calculate the loss (Mean Squared Error) for the test dataset.



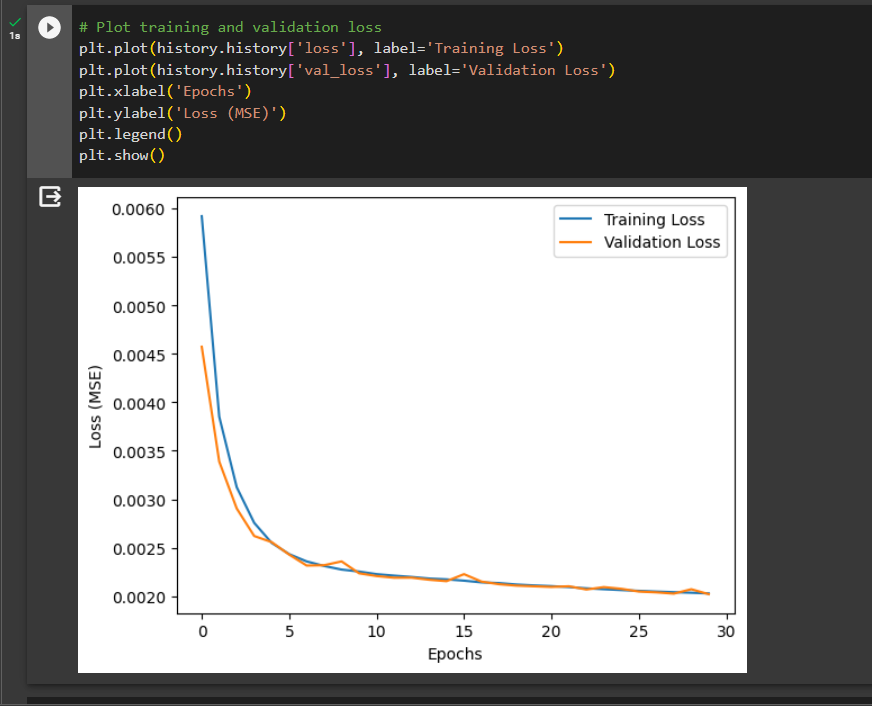
### Write the code implementation to plot the train and validation loss against number of epochs.

## When above AE is used without activation functions, it is called a linear AE. Explain the relationship between linear AE and principal component analysis (PCA). Write the answer in a word file.

Principal Component Analysis (PCA) and a linear autoencoder without activation functions both seek to minimise the dimensionality of data while preserving key information. They both employ linear transformations, and they both strive to reduce variance. The main distinction is that although PCA is a statistical method that computes principal components indirectly from data, linear autoencoders are learned from data through training and provide more flexibility.

## Upload the Vanilla CNN AE jupyter notebook file (i.e., lab\_7\_AE\_Vanilla\_CNN.ipynb) to google colab root directory.

### Write the code implementation to calculate the loss (Mean Squared Error) for the test dataset.



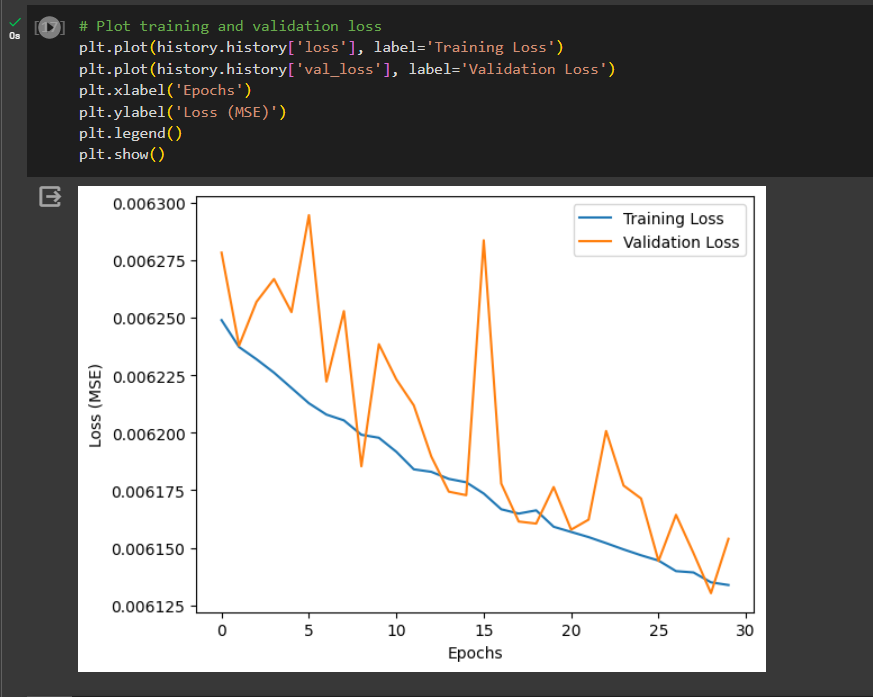
### Write the code implementation to plot the train and validation loss against number of epochs.

## Observe the model performance improvements between the above two models and give reasons for the observed improvements.

For image data like Fashion MNIST, the CNN-based autoencoder performs better than the linear autoencoder because CNNs are excellent at collecting complicated spatial patterns, hierarchical features, and non-linear relationships, which are essential for image analysis. Even while it is easy to understand and straightforward, the linear autoencoder has trouble capturing these complex properties, which results in performance variations.

## Upload the Image De-noising AE jupyter notebook file (i.e., lab\_7\_AE\_CNN\_Image\_Denoising.ipynb) to google colab root directory.

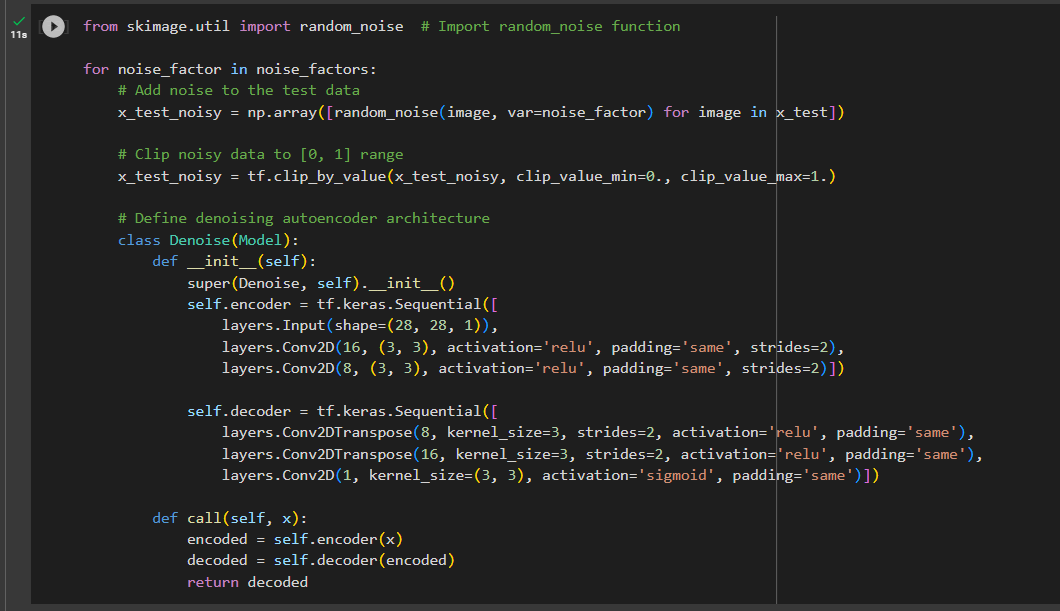
### Write the code implementation to calculate the loss (Mean Squared Error) for the test dataset.



### Write the code implementation to plot the train and validation loss against number of epochs.

### Experiment with “noise\_factor” value and use the best value you find in the final implementation. (Pay attention to how this value affect the images by observing the noise added images in the code.)





## Observe the model performance improvements between the Image De-noising AE and the Vanilla CNN AE.

### Explain the reasons for the observed improvements.

It's vital to keep in mind that while comparing the performance of the Image De-noising Autoencoder (AE) with the Vanilla CNN AE, the Image De-noising AE is intended to not only compress and reconstruct images, but also remove noise from them. The performance improvements and their causes are noted below:

**Performance Improvements of the Image De-noising AE:**

Noise Reduction: The Image De-noising AE successfully reduces noise from input images, which is the most obvious benefit. The fundamental function of the Vanilla CNN AE, in contrast, is reconstruction without noise removal.

Enhanced Image Quality: Because noise is removed, images that are reconstructed using the Image De-noising AE typically have better visual quality and less artefacts. In contrast, the Vanilla CNN AE may replicate noisy patterns, resulting in poorer image quality.

Image De-noising AE is more resistant to noisy input data in terms of noise resistance. It can better manage changes brought on by various degrees of noise in the input images.

**Reasons for the Observed Improvements:**

Noise Modelling: The De-noising of Images The purpose of AE's explicit training is to model and reduce image noise. It gains the ability to recognise noise patterns during training and suppress them while maintaining crucial image properties. The noise reduction produced by this specialised training is superior.

Loss Function: The Image De-noising AE frequently employs a loss function that penalises the variation between the reconstructed image and the clean image (i.e., one that is noise-free). This loss motivates the model to generate output that closely mimics clean photos, resulting in higher-quality images.

Regularisation: By including noise during training, the Image De-noising AE is encouraged to learn reliable representations. The model is effectively regularised, becoming less prone to overfitting and more capable of handling noisy input during inference.

Application Focus: The Image De-noising AE was made with one particular application—removing noise from images—in mind. Due to its narrow focus, it can perform exceptionally well in situations when noise reduction is essential, including when photographing or denoising images for medical imaging.

Data augmentation: The Image De-noising AE develops noise tolerance by introducing controlled noise to the training data. Better generalisation to real-world noisy images may result from this.

## Explain the differences between AE and Variational AE (VAE).

When learning a compressed representation of the input data for reconstruction, autoencoders (AE) concentrate on learning a deterministic representation, whereas variational autoencoders (VAE) try to model a probabilistic latent space distribution. For probabilistic generative modelling, VAEs are a good fit since they introduce stochasticity, regularisation through KL divergence, and the capacity to produce new data samples.