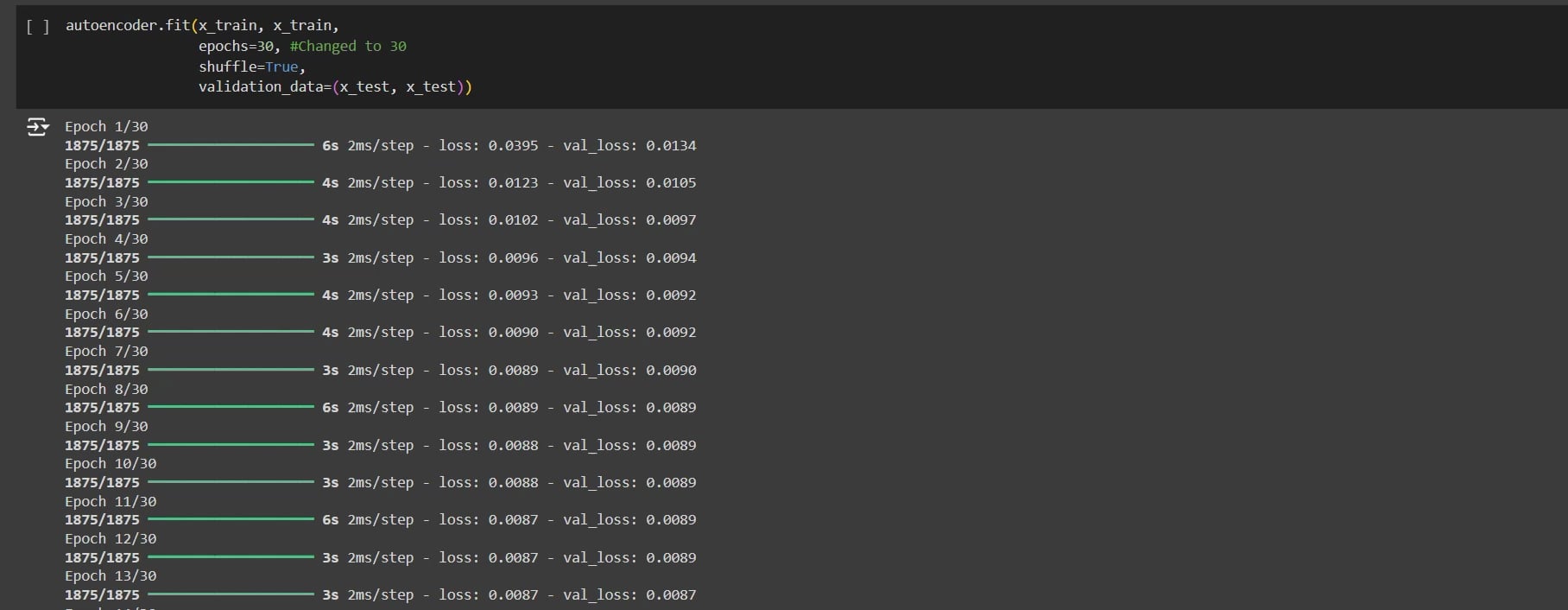
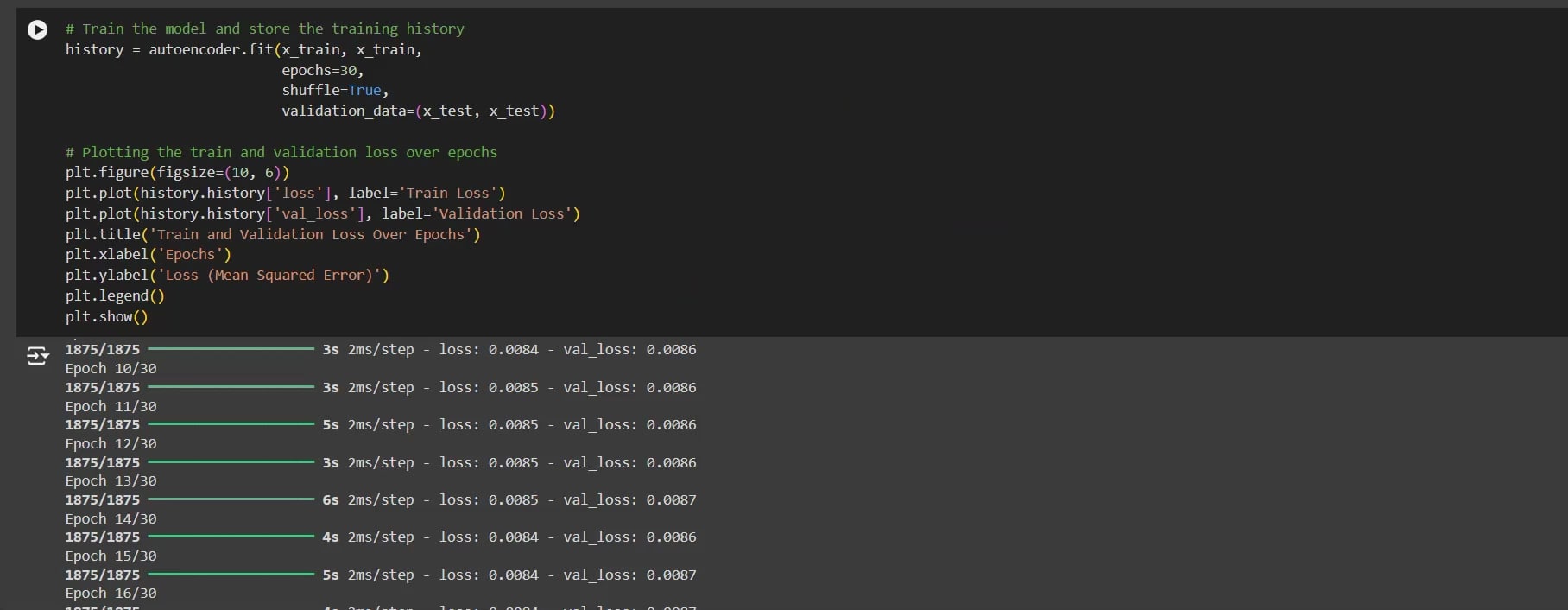
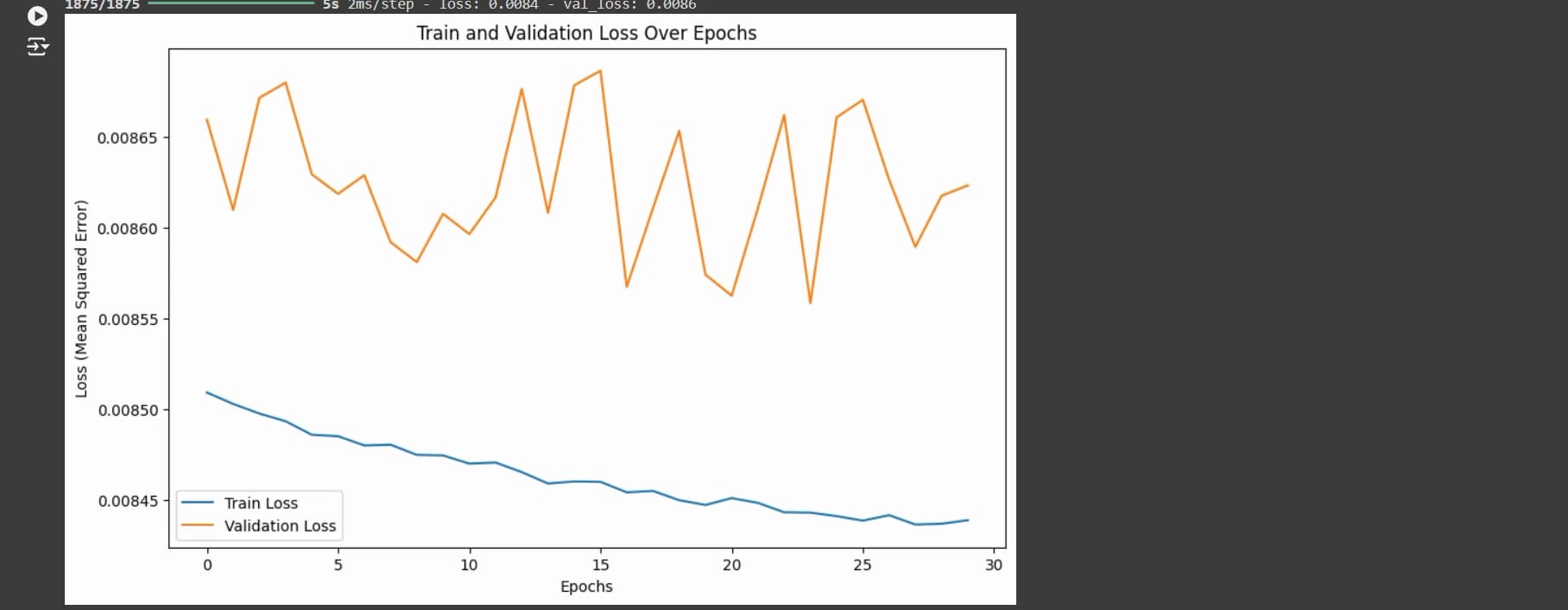
DL lab 7 -Autoencoders

1. Upload the Autoencoder (AE) jupyter notebook file (i.e., lab\_7\_AE\_FFNN.ipynb) to google colab root directory.
   * In this code, an image reconstruction is done using dense layers-based AE.
   * Fashion MNIST dataset is used for this task (also for the subsequent tasks as well).
   * Run the above code and understand it.
   * Train the model with 30 epochs.



* + 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.
  + 
  + 

1. 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.

A **Linear Autoencoder (AE)**, when used without activation functions, is mathematically equivalent to **Principal Component Analysis (PCA)**. Both techniques aim to reduce the dimensionality of data by finding a lower-dimensional representation that captures the most important information. In PCA, this is done by projecting the data onto principal components, which maximize the variance in the data. Similarly, a linear AE encodes the data into a latent space and reconstructs it, minimizing the reconstruction error. The key difference is that PCA is strictly a linear statistical method, whereas AEs can be extended to non-linear cases by incorporating activation functions.

1. Upload the Vanilla CNN AE jupyter notebook file (i.e., lab\_7\_AE\_Vanilla\_CNN.ipynb) to google colab root directory.
   * In this code, instead of dense layers, 2D CNN layers are used.
   * Task in the same as before with the same Fashion MNIST dataset.
   * Run the above code and understand it.
   * Train the model with 30 epochs.
   * 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.
2. Observe the model performance improvements between the above two models and give reasons for the observed improvements.

**Better Feature Extraction** - Convolutional layers are more effective in learning spatial hierarchies in image data. They capture local patterns like edges, textures, and shapes, leading to better reconstruction compared to fully connected layers, which flatten the data and lose spatial relationships.

**Improved Generalization** - By focusing on local regions of the input (using filters), CNN-based autoencoders tend to generalize better, improving performance on unseen data.

1. Upload the Image De-noising AE jupyter notebook file (i.e., lab\_7\_AE\_CNN\_Image\_Denoising.ipynb) to google colab root directory.
   * In this code, noise is first added to the images before the reconstruction.
   * This is a method to overcome the overfitting that happens in AEs.
   * Run the above code and understand it.
   * Train the model with 30 epochs.
   * 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.)
2. Observe the model performance improvements between the Image De-noising AE and the Vanilla CNN AE.
   * Explain the reasons for the observed improvements.

The Image De-noising Autoencoder (AE) outperforms the Vanilla CNN AE primarily due to its specialized training to remove noise from images.

**Reasons for Improvements:**

1. **Noise Handling:** The De-noising AE effectively filters out irrelevant noise, leading to clearer reconstructions.
2. **Feature Sensitivity:** It learns to identify and preserve essential features while ignoring noise, enhancing output quality.
3. **Improved Generalization:** Training with noise reduces overfitting, allowing better generalization on unseen data.
4. **Regularization:** Adding noise encourages the model to learn patterns rather than memorizing the training set, improving robustness.
5. **Optimal Noise Factor:** Fine-tuning the noise level enhances feature extraction, balancing learning and noise handling.

Overall, the De-noising AE's focus on noise reduction and feature preservation makes it more effective in practical applications.

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

AEs primarily focus on data compression and reconstruction, using a deterministic latent space to encode input data. In contrast, VAEs aim to generate new data samples by employing a probabilistic latent space characterized by a Gaussian distribution. The loss function in AEs typically relies on reconstruction loss, while VAEs incorporate both reconstruction loss and a Kullback-Leibler divergence term to regularize the latent space. Additionally, AEs produce direct reconstructions of the input, whereas VAEs can generate new samples by sampling from the learned latent distribution. Overall, while both models share similarities in their architecture, VAEs offer enhanced generative capabilities, making them suitable for tasks like data synthesis and anomaly detection.

**Submission.**

Download the final modified notebook files (all 3 jupyter notebooks). Add these notebooks and the word file to a new zip file. Upload this zip file to the courseweb submission link. The file name should be your registration number.