GitHub repository link [https://github.com/](https://github.com/ThaminduSulakshana/DL-Lab-IT21206078/tree/b4bef0d728fab344f7ab9fcf97365b5562c2ef8b/lab-6)IT21343520/lab-7

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.
2. 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 without activation functions and principal component analysis (PCA) both aim to reduce the dimensionality of data while maintaining important information. They both use linear transformations and aim to reduce variance. The fundamental difference is that although linear autoencoders learn from data through training and offer more flexibility, PCA is a statistical method that indirectly computes principal components from data.**

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.

* **The CNN-based autoencoder performs better than the linear autoencoder for image data like Fashion MNIST because CNNs are superior at gathering intricate spatial patterns, hierarchical features, and non-linear correlations, all of which are crucial for image analysis. The linear autoencoder has problems capturing these complicated features even though it is simple to grasp resulting in performance variations.**

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.

* **Noise Handling**: By adding noise to the input images, the Image De-noising AE is trained to focus on essential features for reconstruction, making the model more robust and less prone to overfitting.
* **Generalization**: The noise acts as a regularizer, allowing the model to generalize better to unseen data by learning to filter out irrelevant details.
* **Feature Learning**: The Image De-noising AE improves feature learning by forcing the model to reconstruct clean images from noisy inputs, enhancing its ability to capture important spatial and hierarchical patterns.

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

* **Autoencoders (AE) focus on learning a deterministic representation while learning a compressed representation of the input data for reconstruction, whereas variational autoencoders (VAE) attempt to model a probabilistic latent space distribution. VAEs are a strong fit for probabilistic generative modelling because they add stochasticity, regularization through KL divergence, and the ability to generate new data samples.**

**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.