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.

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.

Both aim to find a lower-dimensional representation of data while preserving as much information as possible.

* Both use linear transformations to reduce data dimensionality.
* They try to minimize reconstruction error.
* The lower-dimensional representation found by a linear autoencoder is very similar to the principal components found by PCA.
* The main difference is in how they're implemented:
  + PCA uses direct mathematical calculations.
  + Linear autoencoders learn through training, like other neural networks.

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.
3. 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.)
4. Observe the model performance improvements between the Image De-noising AE and the Vanilla CNN AE.
   * Explain the reasons for the observed improvements.

the differences in performance between the two models and explain the reasons for improvements in the CNN model, which could include better spatial feature learning due to the use of convolutional layers.

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

Git hub link

<https://github.com/Mithun-Dilshan/Deep-Learning-Lab/tree/main/Lab%2007>

**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.