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.

Autoencoders (AEs) are a type of neural network that aim to compress and reconstruct input data. A Linear Autoencoder is a special case where the activation functions in the hidden layers are removed, leading to a purely linear mapping between input and output. The relationship between Linear Autoencoders and Principal Component Analysis (PCA) becomes evident in this scenario.

**Principal Component Analysis (PCA)**

PCA is a statistical method used to reduce the dimensionality of data while preserving as much variance as possible. It works by finding orthogonal axes (principal components) that explain the variance in the data and projects the data onto these axes. PCA is often used for data compression and noise reduction.

**Linear Autoencoders and PCA**

When an autoencoder is purely linear, its encoder and decoder functions become linear mappings. In this case, the linear autoencoder learns a latent space that captures the most significant components of the input data. In fact, it has been shown that for a linear autoencoder, the learned latent space corresponds to the principal components of the input data, making linear autoencoders equivalent to PCA in terms of the learned features.

The encoder of the linear autoencoder acts like the PCA projection, mapping input data onto the principal components. Similarly, the decoder performs a linear transformation to reconstruct the input, analogous to projecting the data back from the principal component space. Therefore, a linear autoencoder learns the same subspace as PCA but through gradient-based optimization.

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.

**Training vs. Validation Loss Trend**:

* In the first plot, the training loss decreases consistently, but the validation loss shows erratic behavior, with peaks and valleys.
* In the second plot, both the training and validation loss follow a smooth downward trend, showing a better convergence.

**Stability of Validation Loss**:

* In the first plot, the validation loss fluctuates significantly with sharp increases and decreases, indicating instability during the learning process.
* In the second plot, the validation loss gradually decreases, although it shows some minor fluctuations, it appears more stable overall compared to the first plot.

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.

**6.**

**Performance Improvements**:

* **Image Quality**: The denoising autoencoder typically produces clearer images compared to the vanilla CNN autoencoder, especially when the input images are noisy. This is because it is specifically trained to reconstruct clean images from corrupted inputs.
* **Robustness to Noise**: The denoising autoencoder is designed to handle noise, which can lead to better performance on test datasets that contain noise.
* **Lower Loss**: The training loss for the denoising autoencoder may converge to a lower value than that of the vanilla CNN autoencoder due to its specialized training objective.

**Reasons for Improvements**:

**Training Objective**: The denoising autoencoder learns to reconstruct clean images from noisy versions, allowing it to capture essential features more effectively.

**Data Augmentation**: By intentionally adding noise, the model learns to generalize better, making it more robust to variations and improving performance on unseen data.

**Feature Extraction**: The denoising autoencoder often extracts more relevant features that contribute to better reconstructions, especially in challenging conditions.

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

**AE**: The goal is to minimize the reconstruction error (e.g., Mean Squared Error) between the input and the output. It focuses on learning a compressed representation of the data.

**VAE**: Aims to maximize the likelihood of the data while also incorporating a regularization term to ensure the latent space follows a specific distribution (usually Gaussian). This allows for generating new samples.

**AE**: The latent space representation can be irregular and may lead to clustering or overfitting, making it harder to sample new data points.

**VAE**: Forces the latent space to have a continuous and structured distribution, enabling easy sampling and interpolation between data points.

**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.