DL lab 7 -Autoencoders

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.

Relationship Between Linear AE and PCA

A **Linear Autoencoder** is an autoencoder without any non-linear activation functions in the hidden layers, meaning the encoding and decoding process is purely linear. In such cases, the autoencoder learns to reconstruct the input data through a linear transformation, which can be closely related to Principal Component Analysis (PCA).

**PCA** is a statistical technique used to reduce the dimensionality of a dataset by finding the directions (principal components) along which the variance of the data is maximized. PCA performs a linear transformation of the data and retains the most important features (the components that explain most of the variance).

Key Similarities:

**Dimensionality Reduction:** Both PCA and linear autoencoders aim to reduce the dimensionality of the input data by encoding it into a lower-dimensional space.

**Linear Transformations:** Linear AEs, like PCA, use linear transformations (matrix multiplication) to map the data to a lower-dimensional space and reconstruct it back.

Learning Process: In PCA, the transformation is determined by the eigenvectors of the data covariance matrix, while in linear AEs, the transformation is learned via gradient descent to minimize reconstruction loss (usually MSE).

Key Differences:

**Learning Objective:** PCA explicitly finds directions that maximize variance, whereas linear AEs learn to minimize reconstruction error. Though the learned features in both methods might be similar, their objectives differ.

**Reconstruction:** PCA guarantees that the reconstruction will be optimal in terms of variance retention, while linear AEs do not directly focus on maximizing variance.

Thus, a Linear Autoencoder without non-linear activations effectively performs a task similar to PCA, but with a different optimization process.

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

Performance Comparison:

**Dense AE:**

In the Dense Autoencoder (AE), fully connected layers are used to compress and reconstruct the image data.

Since the image's spatial structure (such as neighboring pixel relationships) is not explicitly preserved, the model might struggle to capture complex spatial dependencies, which can affect reconstruction quality.

**CNN AE:**

In the CNN-based Autoencoder, convolutional layers are employed, which are specifically designed to capture spatial hierarchies in images.

CNN layers preserve the 2D structure of the image, allowing the model to recognize spatial features such as edges, textures, and patterns more effectively.

As a result, CNN autoencoders typically outperform dense autoencoders on image-related tasks, as they can better exploit the local connectivity patterns in images.

**Reasons for Observed Improvements in CNN AE:**

**Spatial Awareness:** CNN layers respect the 2D spatial arrangement of pixels, which allows the model to capture important features such as textures and object shapes. Dense layers, on the other hand, flatten the image, which can lose critical spatial information.

**Parameter Efficiency:** CNN layers use fewer parameters than fully connected layers because they share weights across the input image, focusing only on local areas. This allows CNNs to generalize better, especially for larger image datasets.

**Hierarchical Feature Learning:** Convolutional layers learn hierarchical features—starting from lower-level features like edges and textures, moving to higher-level representations. Dense AEs do not naturally capture this feature hierarchy.

**Reduced Overfitting:** CNN-based autoencoders are less prone to overfitting than dense autoencoders because of their weight-sharing mechanisms and local receptive fields. This leads to better generalization to unseen data, especially when the training dataset is not large.

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.

**Performance Comparison:**

**Vanilla CNN AE:** This model reconstructs the input images without any noise. While it is good at preserving spatial features and reducing dimensionality, it may still overfit if the model learns to memorize the training data instead of generalizing to unseen data.

**Image Denoising AE:** In this model, noise is added to the input images, forcing the autoencoder to learn to remove the noise and reconstruct a clean image. By introducing noise, this approach reduces the risk of overfitting, as the model focuses on capturing the most important features rather than memorizing the exact pixel patterns.

**Reasons for Observed Improvements in Image Denoising AE:**

**Regularization Effect:** Adding noise to the input images acts as a form of regularization, helping to prevent overfitting. The model is forced to focus on essential features rather than memorizing the input data, which leads to better generalization on unseen data.

**Improved Generalization:** Since the model learns to denoise images, it becomes more robust to variations and distortions in the data. This helps it generalize better to new, noisy, or imperfect data compared to the Vanilla CNN AE, which may overfit on clean data.

**Robust Feature Learning:** Denoising AEs are designed to learn robust features that are useful for image reconstruction even when the input is corrupted. The network learns to focus on meaningful patterns and representations that contribute to noise removal, which helps improve reconstruction performance on noisy test data.

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

**Generative Capability:** VAEs can generate new data by sampling from the latent space, whereas traditional AEs are limited to reconstructing the input data.

**Latent Space Structure:** VAEs learn a more structured and continuous latent space, enabling smooth interpolation between different points, which is useful for data generation tasks. In contrast, AEs have an unstructured latent space.

**Loss Function:** AEs focus only on minimizing reconstruction loss, while VAEs also include a regularization term (KL divergence) to ensure that the latent space follows a certain distribution.