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.

**Answer:**

A Linear Autoencoder (AE) is a neural network with no activation functions, and it performs dimensionality reduction similarly to Principal Component Analysis (PCA).   
Both methods attempt to capture the essential structure of the data by reducing its dimensionality, but they achieve this in slightly different ways.

**1.LinearAE**

* In a Linear AE, both the encoder and decoder are linear transformations, without the use of non-linear activation functions such as ReLU or Sigmoid.
* The network tries to minimize the reconstruction error between the input data and its reconstruction, often using Mean Squared Error (MSE) as the loss function.
* Without non-linear activation functions, the linear transformation in an AE resembles a matrix factorization approach, like PCA.

**2.PCA**

* PCA is a statistical method that transforms data by projecting it onto a set of orthogonal basis vectors called principal components.
* PCA seeks to capture the directions (components) that explain the most variance in the data and performs a linear transformation of the data.
* PCA does not involve iterative training like neural networks; it can be solved by performing an eigenvalue decomposition of the data covariance matrix.

**3.Similarity**

When no activation function is used in AE, the linear transformations in the encoder and decoder can be interpreted similarly to PCA’s principal component projections. Both methods linearly project data into lower-dimensional spaces and aim to capture the essential information.  
  
 **4. Differences**  
PCA has a closed-form solution based on eigenvalue decomposition, whereas linear AE is optimized through gradient-based methods. Furthermore, AE can be extended to non-linear cases by introducing activation functions, while PCA is strictly linear.

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.

**Answers:**

**1. Improved Performance in CNN AE**

* **Reason**: **Convolutional layers** are more effective at processing image data compared to fully connected (dense) layers. Convolutional layers in CNNs are designed to detect spatial hierarchies in images by capturing local patterns, such as edges, textures, and shapes, and then progressively learning more complex features. This makes CNNs naturally suited for image-based tasks like the reconstruction of Fashion MNIST images, where spatial relationships between pixels are crucial.

In contrast, **dense layers** treat each pixel independently, which limits their ability to capture spatial patterns as effectively as CNN layers do.

**2. Reduction in Parameters**

* **Reason**: CNNs use **parameter sharing** (where the same filters are applied across the entire image) and **local connectivity** (where filters only operate on small regions of the input), leading to fewer parameters to learn compared to fully connected layers. This results in a more efficient and robust model that is less prone to overfitting.

A dense autoencoder, on the other hand, has a fully connected architecture, leading to a significantly larger number of parameters. As a result, the model can be more prone to overfitting, especially when trained on smaller datasets or when the dimensionality of the input is large.

**3. Better Feature Extraction**

* **Reason**: CNNs can capture **localized patterns** and learn hierarchical representations of the input data. In image data, this allows the CNN AE to focus on critical features such as edges, corners, and textures, and build more complex patterns in deeper layers. This leads to a better representation of the data in the latent space, which enhances the reconstruction accuracy.

In contrast, the Dense AE captures global information but may not efficiently extract these local patterns, resulting in less accurate reconstructions.

**4. Handling of Image Translation and Distortions**

* **Reason**: CNN layers are relatively **invariant to small translations or distortions** in the input images. This makes CNN AEs more robust in image processing tasks, as they can still capture important features even when images have slight variations. Dense layers lack this property because each pixel is treated independently without considering neighboring pixels.

This robustness allows CNN AEs to generalize better, particularly when dealing with noisy or distorted images, leading to more reliable reconstructions.

**5. Better Scaling to Larger Datasets**

* **Reason**: CNNs scale well to larger datasets and higher-dimensional input images due to their ability to efficiently capture local spatial information. Dense AEs may struggle with larger input sizes or datasets due to the large number of parameters that need to be learned, which can lead to overfitting or convergence issues.

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

**Answers:**

**6.** When comparing the performance of the **Image De-noising Autoencoder (AE)** with the **Vanilla CNN Autoencoder (AE)**, you will likely observe that the **Image De-noising AE** performs better in certain ways, especially in reducing overfitting and handling noisy inputs. Here’s why these improvements occur:

**Observed Improvements:**

1. **Better Generalization**: The Image De-noising AE typically generalizes better to new data compared to the Vanilla CNN AE. This is because the Image De-noising AE is trained to reconstruct clean images from noisy inputs, which inherently makes it more robust to variations in the data.
2. **Robustness to Noisy Inputs**: Image De-noising AEs are explicitly designed to handle noisy data, which means they can reconstruct images even when the input has some form of noise or corruption. In contrast, a Vanilla CNN AE is typically not trained to handle noisy data and may struggle when the input data has variations or noise.
3. **Reduced Overfitting**: By adding noise to the input during training, the Image De-noising AE regularizes itself, which helps to prevent overfitting. The model learns to ignore the noise and focus on the essential features of the input data. The Vanilla CNN AE may tend to overfit to the training data, especially if the dataset is small or noisy, since it tries to memorize every detail of the input without any form of regularization.
4. **Improved Reconstruction Quality**: Due to its robustness and ability to denoise, the Image De-noising AE is often better at reconstructing the underlying structure of an image even when the input data is corrupted. The Vanilla CNN AE might struggle with such reconstructions if there is any form of noise or distortion in the input data.

**Reasons for the Observed Improvements:**

* **Noise Injection as Regularization**: Adding noise during training acts as a regularizer, forcing the model to focus on the most important features of the data while ignoring the noise. This leads to better generalization and reduces overfitting.
* **Handling Noisy Inputs**: Since the Image De-noising AE is explicitly trained to remove noise from inputs, it is more robust to variations in the data, leading to more accurate reconstructions compared to the Vanilla CNN AE, which is not designed to handle noise.
* **Better Feature Learning**: The Image De-noising AE learns to map noisy inputs to clean outputs, which helps the model focus on learning essential patterns in the data, such as edges, shapes, and textures, leading to better performance, particularly when input images have noise or distortions.

**7. Differences Between Autoencoder (AE) and Variational Autoencoder (VAE)**

Autoencoders (AEs) and Variational Autoencoders (VAEs) are both types of neural networks used for unsupervised learning tasks, especially for data compression, dimensionality reduction, and generative tasks. However, they have fundamental differences in how they work and their purposes:

**1. Deterministic vs. Probabilistic Approach:**

* **AE**: A standard autoencoder is a deterministic model. It learns to map an input to a compressed latent representation and then reconstruct the original input from that latent space. This process is deterministic, meaning that the same input will always produce the same output.
* **VAE**: A variational autoencoder is a **probabilistic model**. Instead of mapping the input to a fixed latent space, it maps the input to a **probability distribution** in the latent space. The decoder samples from this distribution to reconstruct the input. This introduces variability and randomness into the reconstruction process, which makes VAEs useful for generative tasks.

**2. Latent Space:**

* **AE**: The latent space in a standard autoencoder is not structured. It simply compresses the data into a lower-dimensional space without any constraint on how the data is distributed in this space. As a result, the latent space of an AE may not be continuous, and interpolation between points in this space might not produce meaningful results.
* **VAE**: The latent space in a VAE is **regularized** and structured. VAEs introduce a constraint during training that forces the latent space to follow a known distribution (typically a Gaussian distribution). This makes the latent space more continuous, meaning that interpolations between points in the latent space produce meaningful results. This is particularly important for generating new, plausible data points.

**3. KL Divergence Regularization:**

* **AE**: A standard AE does not involve any regularization in its latent space. The loss function typically consists of just the reconstruction loss (e.g., Mean Squared Error or Binary Crossentropy).
* **VAE**: In a VAE, the loss function consists of two components: the **reconstruction loss** and the **Kullback-Leibler (KL) divergence**. The KL divergence ensures that the learned latent representation follows a desired distribution (usually a Gaussian). This regularization encourages the model to learn a smooth, continuous latent space that is well-suited for generating new data points.

**4. Generative Capabilities:**

* **AE**: Standard autoencoders are primarily focused on reconstruction tasks (e.g., image denoising, dimensionality reduction). While they can compress and reconstruct data, they are not designed for generating new data points in a controlled manner.
* **VAE**: VAEs are **generative models**. Because the latent space is continuous and well-structured, you can sample from the latent distribution to generate new, unseen data. This makes VAEs particularly useful for tasks like generating new images, synthesizing data, and more.

**5. Reconstruction Process:**

* **AE**: The reconstruction process in a standard autoencoder is straightforward—the encoder compresses the input, and the decoder reconstructs it deterministically.
* **VAE**: In a VAE, the encoder outputs two vectors: the **mean** and **variance** of the latent space distribution. During reconstruction, the decoder samples from this distribution, introducing some randomness into the output. This makes the output of a VAE more flexible and capable of generating new variations of the input data.

**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.