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.

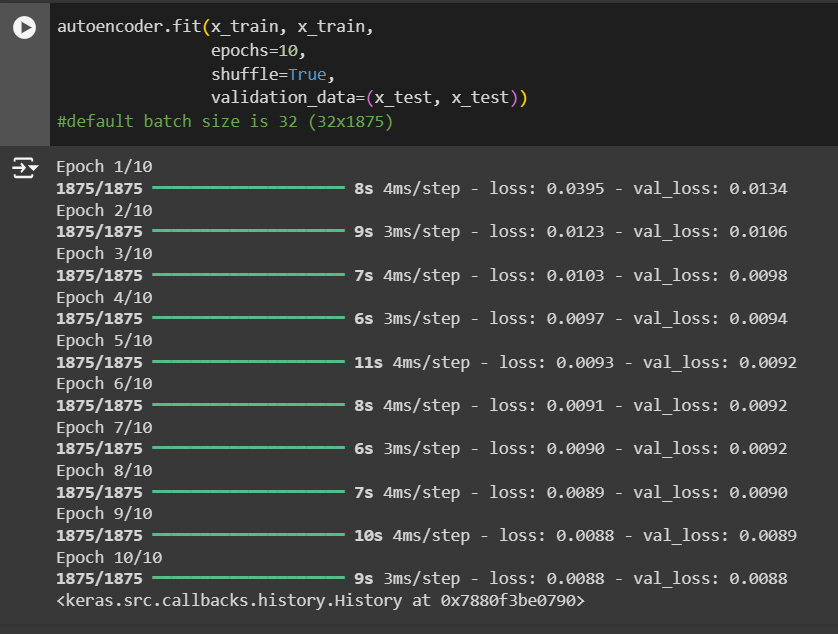


Figure 1: Training with the 10 epoch



Figure 2: Summary of the autoencoder when epochs are 10

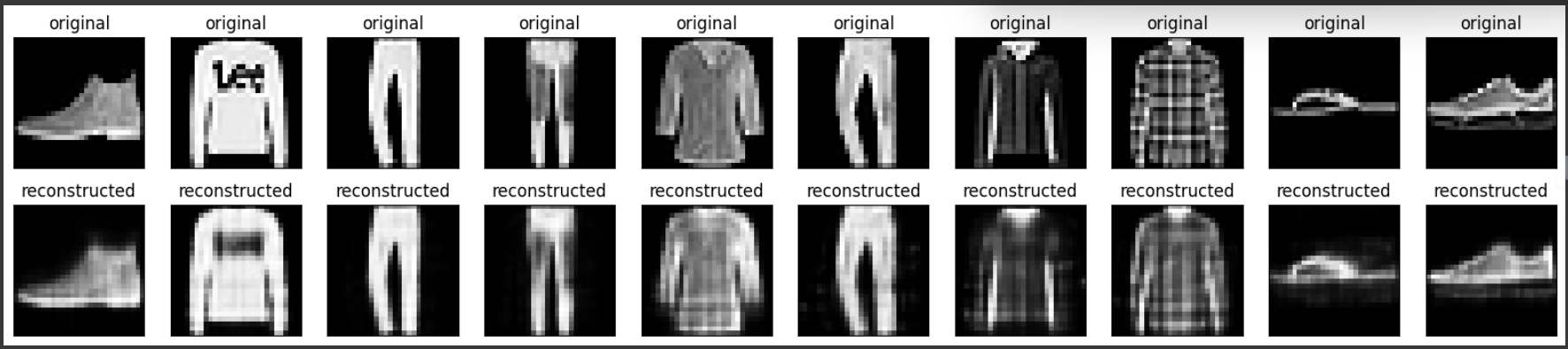


Figure 3: comparison of the construction and reconstruction, when the epochs are 10

A screenshot of a computer

Description automatically generatedChanging epochs to 30

A screenshot of a computer program

Description automatically generatedFigure 5: Training results when epochs are 30

Figure 4: Summary of the autoencoder when epochs are 30

A collage of different clothing

Description automatically generated

Figure 6: Comparison of the original image and reconstructed image when epochs are 30

* + A screenshot of a computer program

    Description automatically generatedWrite the code implementation to calculate the loss (Mean Squared Error) for the test dataset.

Figure 7: Mean squared Error

* + A screenshot of a graph

    Description automatically generatedWrite the code implementation to plot the train and validation loss against number of epochs.

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.

A **linear autoencoder without activation functions** is effectively performing a task similar to **PCA**. Both methods find the best lower-dimensional representation of the data that minimizes reconstruction error, and under certain conditions, the learned subspace of a linear AE will match the principal components found by PCA. The key difference is that PCA is a purely linear method, while AEs can be extended with non-linear transformations to capture more complex patterns in the 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.

A graph with blue lines and red lines

Description automatically generated **Model Performance Improvements Between Dense AE and CNN AE:**

1. **Better Feature Extraction**: CNN AE captures spatial hierarchies and local patterns (e.g., edges, textures) better than Dense AE, which processes input features independently. This makes CNNs more effective for image data.
2. **Parameter Efficiency**: CNNs share weights across spatial locations, reducing the number of parameters. This makes CNNs more efficient and less prone to overfitting compared to Dense AEs.
3. **Improved Generalization**: CNN AE generalizes better to unseen data, leading to lower Mean Squared Error (MSE) and improved reconstructions on the test set compared to Dense AE.

Overall, CNN AE tends to outperform Dense AE in tasks involving image data due to its ability to capture spatial information efficiently.

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.

**Summary of Image De-noising Autoencoder Improvements Over Vanilla CNN AE:**

1. **Improved Generalization with Noise**: Adding noise helps the model learn to filter out irrelevant features, making the Autoencoder more robust and less prone to overfitting. This enhances the generalization ability of the model on unseen data.
2. **Learning to Denoise**: The model learns to focus on reconstructing the clean version of the image, which trains it to detect and preserve essential features while discarding noise.
3. **Regularization through Noise**: By adding noise during training, the model avoids memorizing the training set and instead focuses on learning patterns, improving performance and robustness in handling noisy or distorted data.
4. **Performance with Optimal Noise Factor**: Experimenting with the “noise\_factor” value allows fine-tuning of the model, as too much noise can hinder learning, while the right amount encourages better feature extraction and performance.

Overall, the Image De-noising AE outperforms Vanilla CNN AE due to its enhanced ability to generalize and handle noisy data effectively.

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

**AE:** Learns deterministic mappings and focuses on minimizing reconstruction error without modeling the data distribution. It is not generative by design.

**VAE:** Learns probabilistic latent space representations, allowing it to generate new data points by sampling from learned distributions. It incorporates a regularization term (KL divergence) for smoother latent space structure.

**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.