DL Lab 07 – Answer Document

Github repo link - https://github.com/X-pose/DL\_Lab\_07.git

**Question ->**

**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 ->** In terms of goal, both methods aim to preserve important information from the data. But they are different in terms of how they achieve that goal. Specifically, linear AE does it by learning lower dimensional representation of the data while PCA does it by reducing data to a lower-dimensional space by finding the directions (principal components) that capture the most variance.

**Question ->**

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

**Answer ->**

# Performance comparison

## MSE comparison

A close up of text

Description automatically generated

Figure.. MSE of Dense based AE

A close up of text

Description automatically generated

Figure.. MSE of 2D-CNN based AE

## Loss graphs comparison

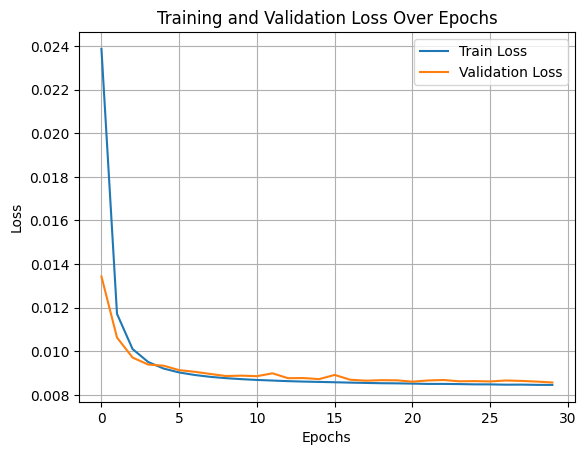


Figure . Dense based AE Loss graph

A graph with a line graph and numbers

Description automatically generated

Figure . 2D-CNN based AE Loss graph

Observations –

1.MSE of 2D-CNN based AE has significantly reduced (0.0016) compared to the dense based AE (0.0085)

2. 2D-CNN based AE displays a more stable learning by having a smoother loss reduction over epochs compared to the dense based AE which has fluctuated losses over epochs.

Reasoning –

2D CNN based AE was able to out-perform the dense based model due its ability to capture local features in images, learn complex spatial patterns and hierarchies through multiple convolutional layers, and its ability to generalize better as shown in the loss graph.

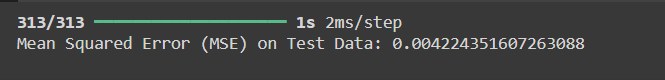
**Question ->**

**5. Experiment with the noise factor.**

**Answer ->**

Observations ->

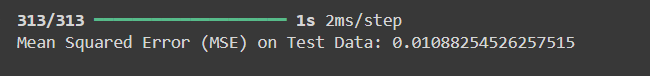
Noise\_factor = 0.2



A graph with blue line and red line

Description automatically generated

Noise\_factor = 0.4



A graph with blue line and red line

Description automatically generated

Noise\_factor = 0.1

A screen shot of a test

Description automatically generated

A graph with a line

Description automatically generated

Conclusion –

As the noise factor increases the model is struggling to learn the patterns and generalize as shown in the (Noise\_factor = 0.4) graph. As the noise factor decreases the model tends to generalize well as shown in the (Noise\_factor = 0.1) graph and by having the lowest MSE among the three. But if the noise factor gets too low, the model starts to overfit as it remembers the noise from the image. Therefore, having noise factor of 0.1 seems like a good choice.

**Question ->**

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

**Answer ->**

By comparing the MSE of both Vanilla CNN AE (0.0016) and Image De-noising AE (0.0023), the slight increase in error for the Image De-noising AE can actually be seen as a sign of better generalization. The addition of noise during training acts as a form of regularization, forcing the model to learn more robust features and avoid overfitting to the training data.

In contrast, the Vanilla CNN AE, while achieving a lower MSE, may overfit to specific patterns in the data without noise, limiting its ability to generalize to unseen or imperfect inputs. The noise added during training with the Image De-noising AE encourages the model to focus on the most important underlying patterns in the data, improving its performance on noisy or real-world inputs.

In summary, while the Vanilla CNN AE achieves a lower error on clean data, the Image De-noising AE demonstrates superior generalization due to noise acting as regularization. This makes it more effective in real-world applications where inputs are often noisy or variable

**Question ->**

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

**Answer ->**

**Autoencoder (AE):**

* An **Autoencoder** is a neural network that takes input data, compresses it into a smaller form (called the "latent space"), and then tries to recreate the original data from this compressed version.
* It has two parts:
  + **Encoder**: Shrinks the data.
  + **Decoder**: Rebuilds the data.

**Variational Autoencoder (VAE):**

* A **Variational Autoencoder** is like an Autoencoder but with a twist. Instead of just compressing the data, it learns how the data is spread out (its distribution).
* It doesn’t just encode the data into a fixed point but into a range (or distribution) of values, allowing it to create new data.

**Key Differences:**

1. **Latent Space**:
   * AE: Encodes the data into single points.
   * VAE: Encodes the data into a range of values (a distribution).
2. **Reconstruction vs. New Data**:
   * AE: Good at recreating input data but not at making new data.
   * VAE: Can generate new data because it learns how the data is distributed.
3. **Goal**:
   * AE: Focuses on making the output look like the input.
   * VAE: Does that too, but also learns the underlying patterns to create new, similar data.