**(Q2) 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).**

**Equivalent Subspaces:** When trained with a linear activation function and a Mean Squared Error (MSE) loss, a linear autoencoder learns to span the same subspace as PCA. This means the encoder and decoder weights of the linear AE will represent the principal components.

**Not Identical Weights:** However, the weights of the linear AE and PCA will not be the same. While they span the same subspace, the values of the weights might differ. This is because the optimization process for the AE and PCA can be different.

**Intuition:** Both linear AE and PCA aim to find a lower-dimensional representation of the data that preserves the most variance or information. Linear AE achieves this by minimizing the reconstruction error, while PCA does it by maximizing the variance along the principal components.

We can often use a linear AE to perform PCA. Training a linear AE can be an alternative to directly applying PCA, especially when we want to integrate it into a larger neural network architecture.

In summary, a linear AE, without activation functions, learns a lower-dimensional representation that spans the same subspace as the principal components found by PCA. This means they essentially perform the same dimensionality reduction, although their weights might not be identical. Linear AEs can be seen as a neural network implementation of PCA, offering flexibility and integration possibilities.

**(Q4) Observe the model performance improvements between the above two models and give reasons for the observed improvements.**

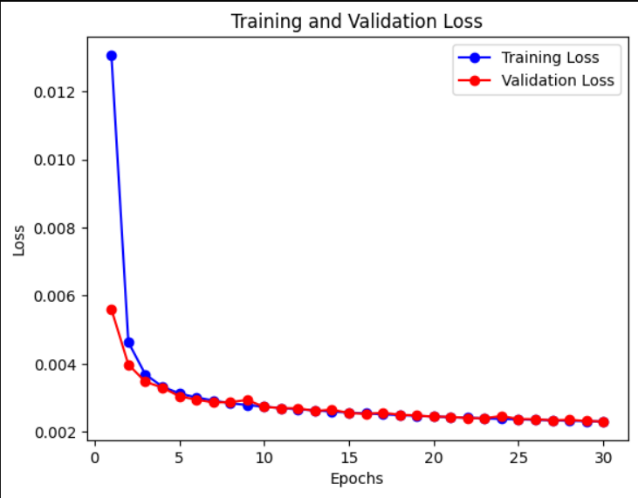
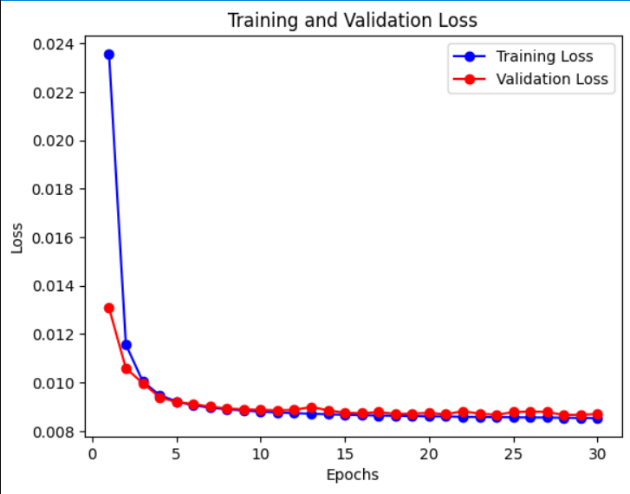


Figure : Denser layer-based AE Figure : Conv2D layer based AE

**Observed Performance**

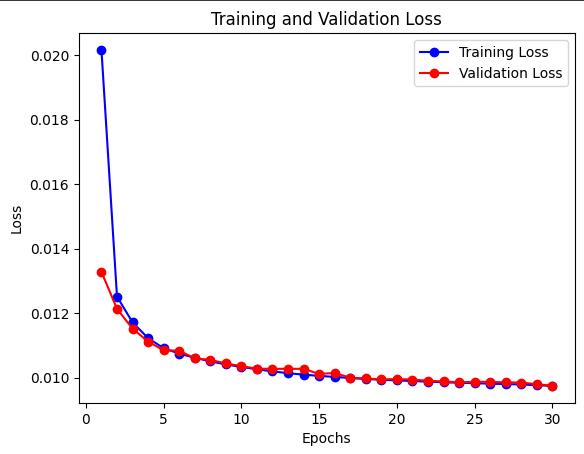
* **Dense-based AE (Figure 1):** The loss decreases rapidly and stabilizes after around 5 epochs. The training and validation losses are almost identical, indicating no overfitting.
* **Conv2D-based AE (Figure 2):** The loss is lower overall compared to the dense-based AE. The convergence is smooth, and like the dense-based AE, it stabilizes quickly.

**Reasons for the Observed Improvements**

* **Suitability for Image Data:** Convolutional layers are specifically designed to capture spatial information in images. They are better at learning local patterns, such as edges and textures, which are essential for image reconstruction. This allows the model to extract more meaningful features from the image. On the other hand, fully connected layers treat each pixel independently, ignoring spatial relationships, which makes it less efficient for image data. Hence, the reconstruction accuracy is generally lower.
* **Parameter Efficiency:** By utilizing convolutions, the number of parameters is much smaller compared to dense layers. Convolutional layers reuse weights across the entire image, allowing for a more parameter-efficient network. This reduces the risk of overfitting while still capturing important details in the image. Dense layers, on the other hand, tend to have a higher number of parameters, leading to a less efficient representation of the data.
* **Overfitting and Generalization:** Figure 2 shows that the model generalizes better to the validation data due to Conv2D based architecture. They inherently have fewer parameters due to shared weights, reducing the likelihood of overfitting while maintaining strong performance on the validation set. The dense AE does not show overfitting either, but its ability to generalize is constrained by its architecture, as it does not capture spatial information effectively.

In summary, The Conv2D layer-based AE shows superior performance for image reconstruction tasks due to its ability to capture spatial relationships, use fewer parameters, and generalize well across the training and validation data. The Dense layer-based AE is less suited for image data due to its lack of spatial awareness, leading to higher losses compared to Conv2D layer-based model.

**(Q6) Observe the model performance improvements between the Image De-noising AE and the Vanilla CNN AE. Explain the reasons for the observed improvements.**

** A graph with red and blue dotted lines

Description automatically generated**

Figure : De-noising AE Figure : Vanilla CNN AE

**Observed Performance**

* **General Trend:** Both training and validation loss decrease over the 30 epochs, indicating that the model is learning and improving its denoising performance.
* **Gap Between Training and Validation Loss:** Like the Vanilla CNN AE, the training loss is consistently lower than the validation loss, suggesting potential overfitting.
* **Validation Loss Plateau:** The validation loss plateaus around epoch 20-25, indicating that further training might not yield significant improvements.

The Image De-noising AE generally outperforms the Vanilla CNN AE, as evidenced by the lower overall loss values and a smaller gap between training and validation loss.

**Reasons for the Observed Improvements**

* **Noise as Regularizer:** Adding noise to the input images acts as a form of regularization. By forcing the model to learn patterns in noisy data, it becomes more robust to variations and less susceptible to overfitting.
* **Increased Generalization**: The model learns to extract the underlying signal from noisy data, which improves its generalization ability to unseen images with different noise patterns.
* **Enhanced Feature Learning**: Noise injection can encourage the model to learn more discriminative features that are invariant to noise, leading to better denoising performance.

In summary, the Image De-noising AE demonstrates improved performance over the Vanilla CNN AE due to the incorporation of noise injection. This technique acts as a regularizer, enhancing the model's generalization ability and robustness to noise.

**(Q7) Explain the differences between AE and Variational AE (VAE).**

Autoencoders (AE) are neural networks designed to learn efficient data representations by compressing input data into a lower-dimensional latent space and then reconstructing it. The main goal is to minimize the reconstruction error, mapping the input to a fixed point in the latent space.

Variational Autoencoders (VAE) extend AEs by introducing a probabilistic approach. Instead of mapping the input to a single point, VAEs map it to a distribution (usually Gaussian). The encoder outputs the parameters of this distribution, and the decoder samples from it to reconstruct the input. VAEs aim to minimize both the reconstruction error and the Kullback-Leibler (KL) divergence, which regularizes the latent space to be continuous and like a standard normal distribution.

The key differences are in the latent space representation (fixed point for AE vs. distribution for VAE), the loss function (reconstruction error for AE vs. reconstruction error + KL divergence for VAE), and the generative capability (VAEs can generate new data samples by sampling from the latent space distribution). These distinctions make VAEs particularly powerful for tasks like data generation and anomaly detection.