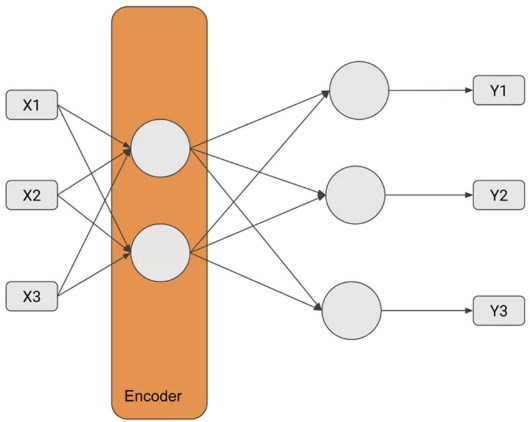
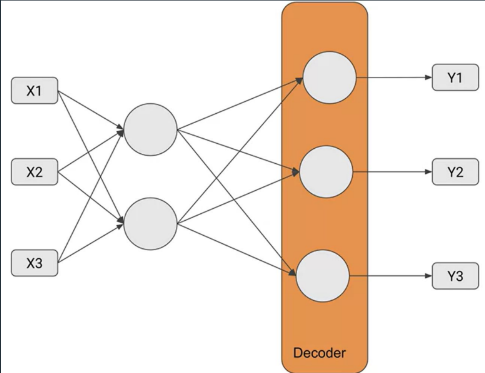
|  |  |
| --- | --- |
| Ex No: 6  Date: 11/09/2024 | MNIST Denoising Autoencoder |

**Objective:**

The main goal of the notebook is to implement a **denoising autoencoder** using convolutional neural networks (CNNs). The autoencoder is trained to remove noise from images of handwritten digits from the **MNIST dataset**. This task demonstrates how autoencoders can learn useful latent representations by reconstructing clean images from noisy inputs.

**Code Explanation for simple\_autoencoder:**

** **

The function simple\_autoencoder(inputs) constructs a two-layer autoencoder:

* **Encoder**: Compresses the input from 784 units to 32 units using a dense layer with ReLU activation.
* **Decoder**: Reconstructs the input back to 784 units using a dense layer with sigmoid activation.

Two models are built:

* **Encoder Model**: Extracts the compressed representation.
* **Autoencoder Model**: Performs the end-to-end encoding and decoding of the input.

For denoising the data:

The function first normalizes the input image by converting it to float32 type and scaling its pixel values from the range [0, 255] to [0, 1]. Next, it adds Gaussian noise to the image by generating random noise with the same shape as the image and scaling it by a noise factor of 0.5, which controls the strength of the noise. This scaled noise is then added to the normalized image to create a noisy version. To ensure the noisy image remains valid, its pixel values are clipped to stay within the range [0, 1]. Finally, the function returns the noisy image as input and the original clean image for target reconstruction.

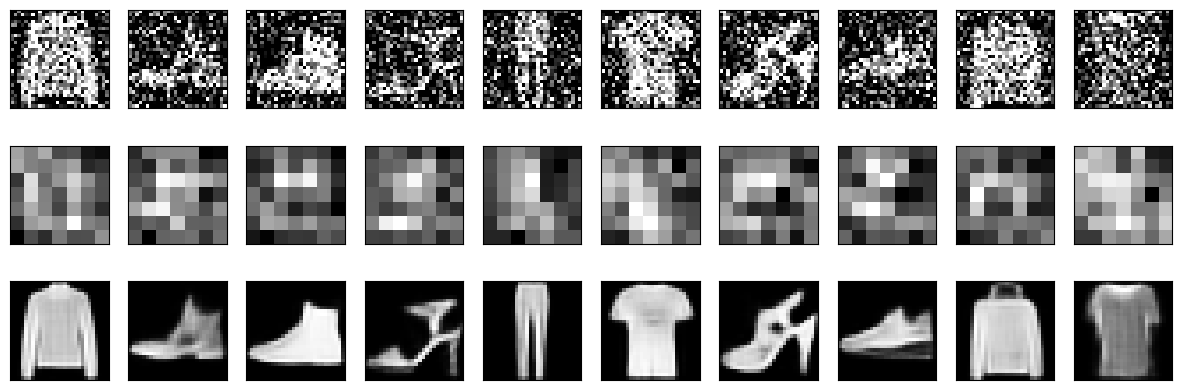
**Results**

The notebook includes code to visualize the performance of the autoencoder by displaying the original images, noisy images, and reconstructed (denoised) images side by side.

The model shows effective denoising, successfully reconstructing the images by removing most of the added noise while preserving the important features of the digits.

The loss curves and visual output indicate the model’s ability to learn the mapping between noisy inputs and clean outputs.

**Result Analysis:**



**Summary:**

The notebook demonstrates the implementation and training of a denoising autoencoder using CNNs and the MNIST dataset. By corrupting the images with Gaussian noise, the autoencoder learns to remove the noise and reconstruct the clean images. The results show that the model effectively learns useful features and performs well in reconstructing clean images, as evidenced by the qualitative and quantitative outputs. This exercise highlights the practical application of autoencoders in tasks requiring dimensionality reduction and data denoising.

**GitHub Link:**

https://github.com/spoorthytorne/fundamentals-of-Deep-learning/tree/main/Lab%206