|  |  |
| --- | --- |
| **Ex No: 5**  **Date: 14th September 2024** | **Implementation of Simple , Deep Auto Encoder , CNN and Denoising Auto Encoder** |

**Objective:**

To Build **Simple, Deep Auto Encoder, CNN and Denoising Auto Encoder** using tensorflow and keras and the datasets – fashion mnist and mnist.

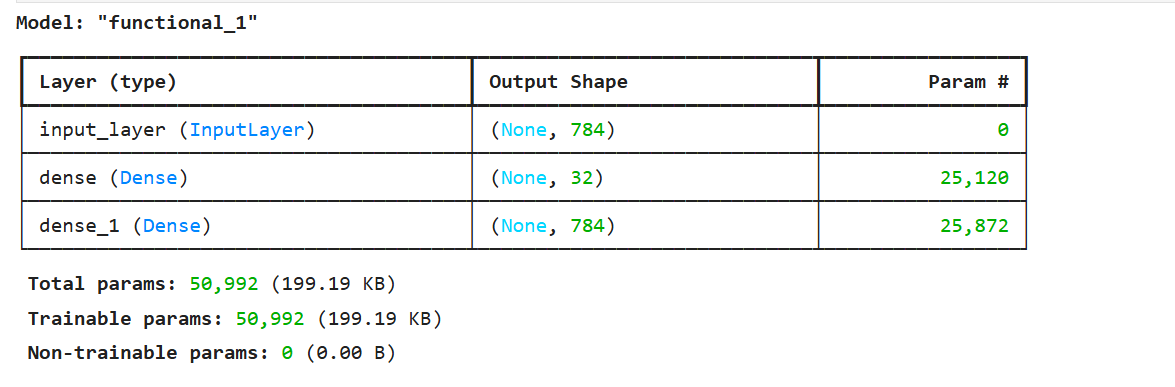
**Descriptions:**

* **Simple Auto Encoder:**

A **Simple Auto encoder** is a type of artificial neural network used for unsupervised learning, primarily for dimensionality reduction or feature learning.

It takes the input as image in the single dimension vector form. The dimension of the image could be reduced using Encoder. The network attempts to reconstruct the input data from the compressed latent space using Decoder. The output of the Simple auto encoder is also a image same as the input.

**Model:**



**1.1 A Simple Auto Encoder**

**Building the parts of algorithm of Simple Auto Encoder**

The main steps for building a Simple Auto encoder are:

1. Define the model structure
2. Building the Encoder
3. Building the Decoder
4. Training the Model by using the epochs
5. Plotting the graph between Model Accuracy and number of epochs
6. Plotting the graph between Model loss and number of epochs

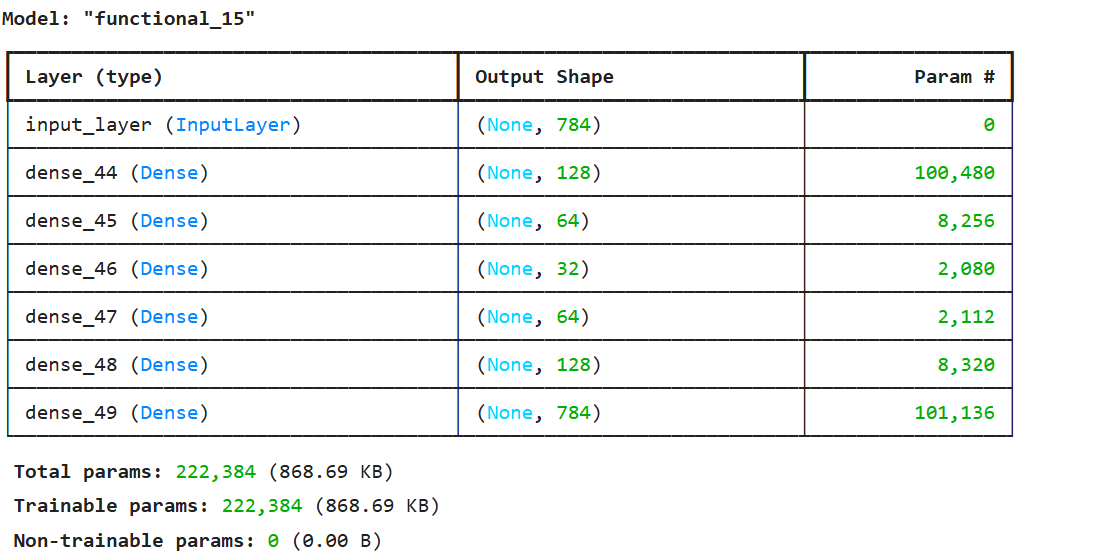
* **Deep Auto Encoder:**

**Description:**

A **Deep Auto encoder** is an extension of the basic auto encoder, where the encoder and decoder consist of multiple layers of neural networks, rather than just a single layer in each.

It takes the input as image in the single dimension vector form. The dimension of the image could be reduced using Encoder and using only Dense layers. The network attempts to reconstruct the input data from the compressed latent space using Decoder and using only Dense layers. The output of the Deep auto encoder is also image same as the input.

**Model:**



**1.2 A Deep Auto Encoder**

**Building the parts of algorithm of DeepAuto Encoder**

The main steps for building a Deep Auto encoder are:

1. Define the model structure
2. Building the Encoder
3. Building the Decoder
4. Training the Model by using the epochs
5. Plotting the graph between Model Accuracy and number of epochs
6. Plotting the graph between Model loss and number of epochs

* **Convolutional Auto Encoder**

**Description:**

A **CNN Auto encoder** is a type of auto encoder that utilizes **Convolutional Neural Networks (CNNs)** for encoding and decoding the input data. It is particularly effective for image-related tasks because CNNs are inherently designed to capture spatial hierarchies and local patterns within images, such as edges, textures, and shapes. CNN auto encoders are often used for image compression, denoising, and feature extraction.

The components of the CNN Auto encoder are:

**Encoder (Convolutional Layers)**:

* The encoder in a CNN auto encoder is composed of **convolutional layers** that learn to capture spatial hierarchies in the input data
* This process helps in extracting local patterns and features from the input, leading to a compressed representation of the image.

**Latent Space (Bottleneck)**:

* The latent space is the compressed, lower-dimensional representation of the input image. The size of this latent space is much smaller than the original image, allowing for data compression.

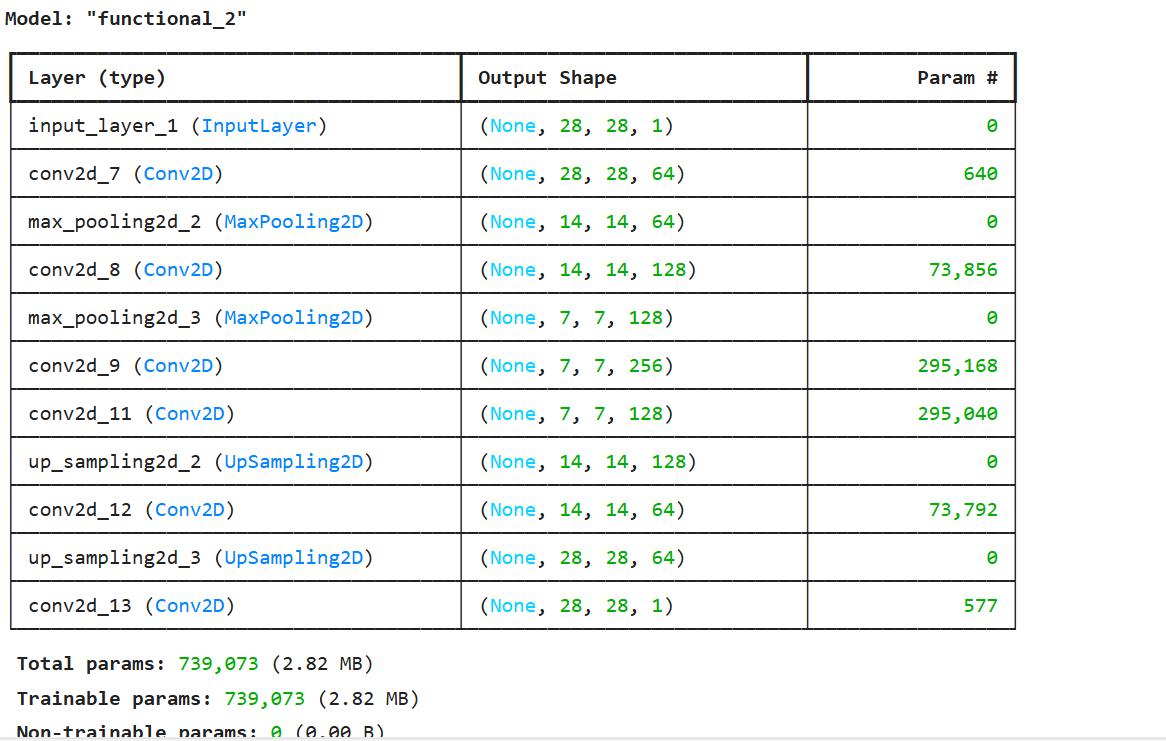
**Decoder (Transpose Convolutional Layers)**:

* The decoder is composed of **transpose convolution layers** (also called **deconvolutional layers**), which perform the reverse of the encoder.
* The goal of the decoder is to reconstruct the input image as accurately as possible from the latent space.

**Output Layer**:

* The final output should closely resemble the input image. The CNN autoencoder minimizes the reconstruction error, often using a loss function like **mean squared error (MSE)** or **binary cross-entropy** for binary images.

**Model:**



**1.3 A CNN Auto Encoder**

**Building the parts of algorithm of CNN Auto Encoder:**

The main steps for building a CNNAuto Encoder are:

1. Define the model structure
2. Building the Encoder
3. Building the Bottle Neck
4. Building the Decoder
5. Training the Model by using the epochs
6. Plotting the graph between Model loss and number of epochs

* **Denoising Auto encoder (DAE):**

**Descriptions:**

A **Denoising Auto encoder (DAE)** is a type of auto encoder specifically designed to remove noise from input data. Unlike standard auto encoders, which aim to reconstruct their inputs, denoising auto encoders are trained to reconstruct the original, noise-free version of the input from a corrupted version.

 **Corruption of Input Data**:

* During training, the input data is intentionally corrupted by adding noise (e.g., Gaussian noise, masking out random portions of the input, or adding salt-and-pepper noise). This creates a noisy version of the input.
* The original clean data is kept as the target output for reconstruction.

 **Training Objective**:

* The objective is to train the autoencoder to map the noisy input back to the clean, original data. In other words, the network learns to "denoise" the input by minimizing the difference between the output and the clean input data.

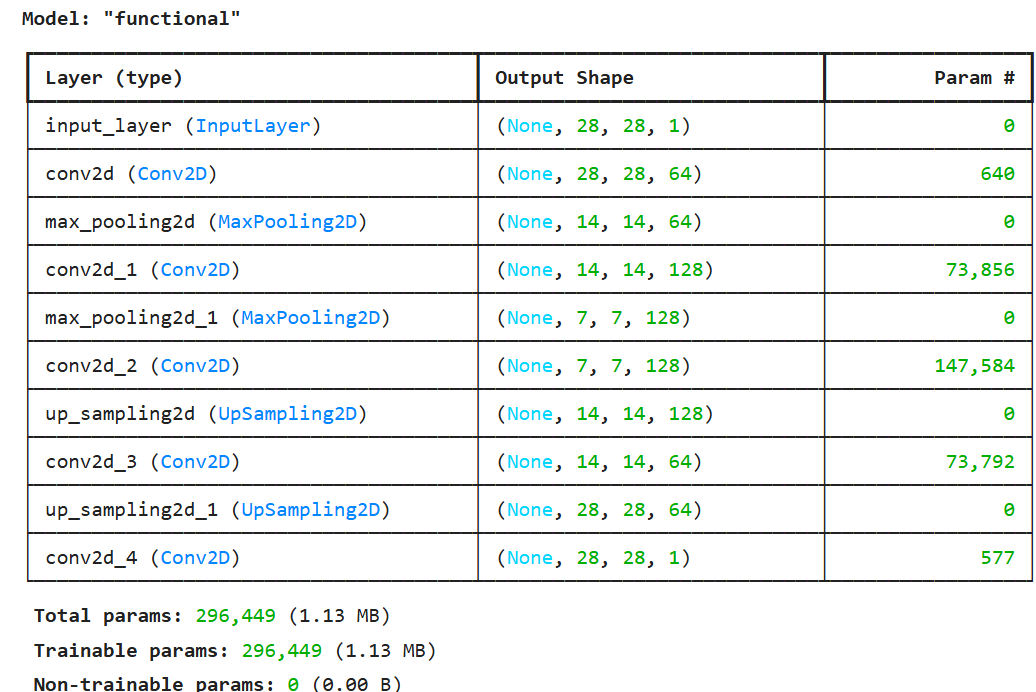
 **Encoder**:

* The encoder network compresses the noisy input into a smaller, more abstract latent representation. This latent representation contains the essential features of the data, even though the input is noisy.

 **Decoder**:

* The decoder reconstructs the original clean data from the latent space. This part of the network attempts to undo the corruption introduced to the input.
* The output of the decoder is the "denoised" version of the noisy input.

**Model:**

****

**1.4 A Denoising Auto encoder using tensorflow and Keras**

**Building the parts of the algorithm of Denoising Auto Encoder**

The main steps for building a Denoising Auto Encoder are:

1. Define the model structure
2. Introducing some noise to the images using noisy factor
3. Building the Encoder
4. Building the Decoder
5. Training the Model by using the epochs

**GitHub Link:**

* + 1. [**https://github.com/jihanrjbtech22/DeepLearningLab/blob/main/Lab5a\_Lab5b\_Simple\_and\_Deep\_Auto\_encoder\_Jihan.ipynb**](https://github.com/jihanrjbtech22/DeepLearningLab/blob/main/Lab5a_Lab5b_Simple_and_Deep_Auto_encoder_Jihan.ipynb)
    2. [**https://github.com/jihanrjbtech22/DeepLearningLab/blob/main/Lab5c\_CNN\_Autoencoder\_Jihan.ipynb**](https://github.com/jihanrjbtech22/DeepLearningLab/blob/main/Lab5c_CNN_Autoencoder_Jihan.ipynb)
    3. [**https://github.com/jihanrjbtech22/DeepLearningLab/blob/main/Lab5d\_Denoising\_Autoencoder\_Jihan.ipynb**](https://github.com/jihanrjbtech22/DeepLearningLab/blob/main/Lab5d_Denoising_Autoencoder_Jihan.ipynb)