|  |  |
| --- | --- |
| **Ex No: 6.1**  **Date: 10-9-2024** | **MNIST Autoencoder** |

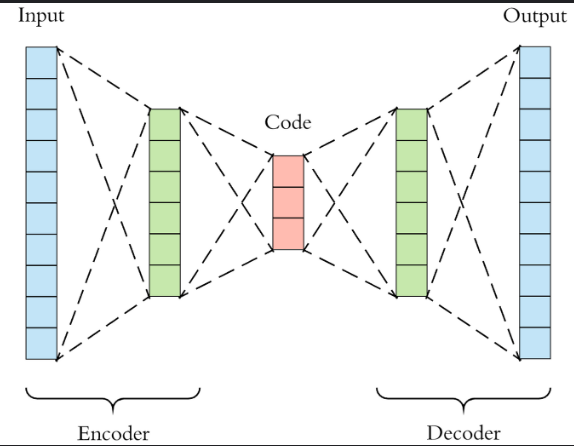
**Objective:**

Implement a simple autoencoder for the MNIST dataset that compresses images to a 32-dimensional latent space and reconstructs the original images from this compressed representation. Evaluate the model's performance using binary crossentropy loss and visualize the encoder's output.

**Descriptions:**

This lab involves constructing a basic autoencoder for the MNIST dataset, aiming to compress 28x28 pixel images into a 32-dimensional latent space and reconstruct them. The dataset is preprocessed by normalizing pixel values and reshaping images into 784-dimensional vectors. The autoencoder's architecture includes an encoder that compresses inputs and a decoder that reconstructs them. The model is compiled with the Adam optimizer and binary crossentropy loss, and trained for 50 epochs. The encoder's output will be visualized to explore the latent space representation. Additionally, the performance of the autoencoder will be evaluated based on the reconstruction quality of the test images, providing insights into the model's ability to generalize.

**Model:**

****

**Building the parts of algorithm**

1) Data Preparation:

* Load the MNIST dataset using TensorFlow Datasets (TFDS).
* Normalize pixel values and reshape images to a flat vector of size 784.
* Shuffle and batch the training data, and batch the test data.

2) Autoencoder Architecture:

* Encoder: A dense layer that compresses the input from 784 dimensions to 32 dimensions.
* Decoder: A dense layer that reconstructs the input from the 32-dimensional latent space back to 784 dimensions.

3) Model Setup:

* Define the input shape of the model (784-dimensional vector).
* Construct the autoencoder model by connecting the encoder and decoder.
* Create a separate encoder model for visualization purposes.

4) Model Compilation:

* Compile the autoencoder model with the Adam optimizer and binary crossentropy loss function.

5)Model Training:

* Train the autoencoder model using the training dataset for a specified number of epochs.

6) Evaluation and Visualization:

* Monitor the training process and evaluate the model’s performance.
* Visualize the encoded latent space representation and reconstructed images to assess the model’s effectiveness.

**GitHub Link:**

**https://github.com/amruthaa-m/DL-Lab1/tree/main/Unit-2/lab6.2**