

**Ex No: 6.1****Lab 6.1: Linear (simple) autoencoders****Date: 10-09-2024****Objective:**

This Lab is to demonstrate building a simple autoencoder using TensorFlow and train it on the MNIST dataset to encode and reconstruct images. Autoencoders are a type of neural network used to learn efficient representations (encodings) of data, typically for dimensionality reduction or compression. The model takes an image as input, encodes it into a compressed form, and then decodes it back into the original image format.

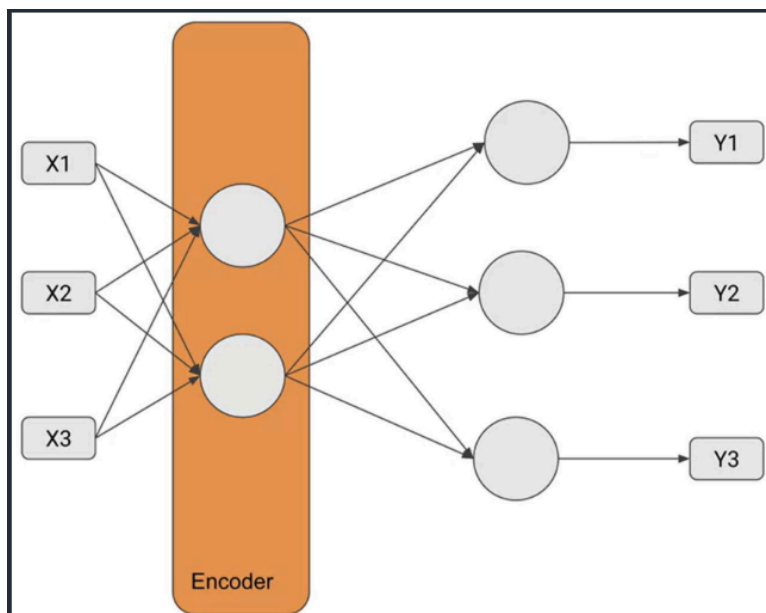
**Descriptions:**

An autoencoder consists of two main components:

**Encoder:** This compresses the input image into a lower-dimensional representation.

**Decoder:** This reconstructs the original image from the compressed encoding.

In this implementation, we use the MNIST dataset, which consists of 28x28 grayscale images of handwritten digits. The input is normalized and flattened to a 784-element vector. The autoencoder then reduces the dimensionality to a 32-unit vector (encoding), and the decoder reconstructs the 784-element vector from this encoding.



USN NUMBER:1RVU22CSE018

NAME:Akshay B

### **Steps to Build the Model:**

We use TensorFlow and TensorFlow Datasets are imported to build the model and load the MNIST dataset.

#### **Load the Dataset:**

The MNIST dataset is loaded using `tfds.load()`.

The images are normalized and flattened to fit the input shape (784 elements).

#### **Data Preprocessing:**

Each image is normalized by dividing the pixel values by 255.

The images are flattened from a 28x28 shape to a 784-dimensional vector.

The train and test datasets are batched and shuffled.

#### **Autoencoder Model:**

The encoder uses a Dense layer with 32 units and ReLU activation to compress the input data.

The decoder uses a Dense layer with 784 units and a sigmoid activation function to reconstruct the input.

The model uses binary cross-entropy as the loss function and Adam optimizer for training.

#### **Compile the Model:**

The encoder and decoder are combined into the autoencoder model, and the model is compiled using the adam optimizer and `binary_crossentropy` loss.

Train the Model:

The model is trained using the preprocessed training dataset for 50 epochs with a batch size of 128.

#### **Evaluate the Model:**

After training, a batch of test images is passed through the encoder and decoder.

The encoded (compressed) representations and the reconstructed images are displayed.

Visualize the Results:

The input images, encoded (compressed) images, and reconstructed (decoded) images are displayed using Matplotlib.

#### **GitHubLink:**

USN NUMBER:1RVU22CSE018

NAME:Akshay B

**<https://github.com/Akb-25/Foundations-of-Deep-Learning/blob/main/Sep%209th.ipynb>**