**ICP-6**

**Name: Manu Sharma**

**700762311**

**GitHub Link**: https://github.com/analystmanu/ICP\_6

**Video Link**: https://drive.google.com/file/d/1Ku3KKVMyDXtuYb7afFg7Jx7BIzj6Hnvq/view?usp=drive\_link

**1:**

Autoencoder Architecture:

Input Layer: Takes 784-dimensional vectors (flattened 28x28 images).

Encoded Layer: Compresses input to 32 units.

Hidden Layer: 64 units to capture more complex patterns.

Decoded Layer: Reconstructs the input back to 784 units.

Key Components:

Encoding: Reduces dimensionality from 784 to 32, retaining essential features.

Hidden Layer: Adds complexity to better capture and decode patterns.

Loss Function: Uses binary crossentropy to compare input vs reconstruction.

Optimizer: Adadelta adjusts the learning rate during training.

Data Preparation:

Fashion MNIST images are normalized to [0, 1] and flattened to 784-dimensional vectors.

Training:

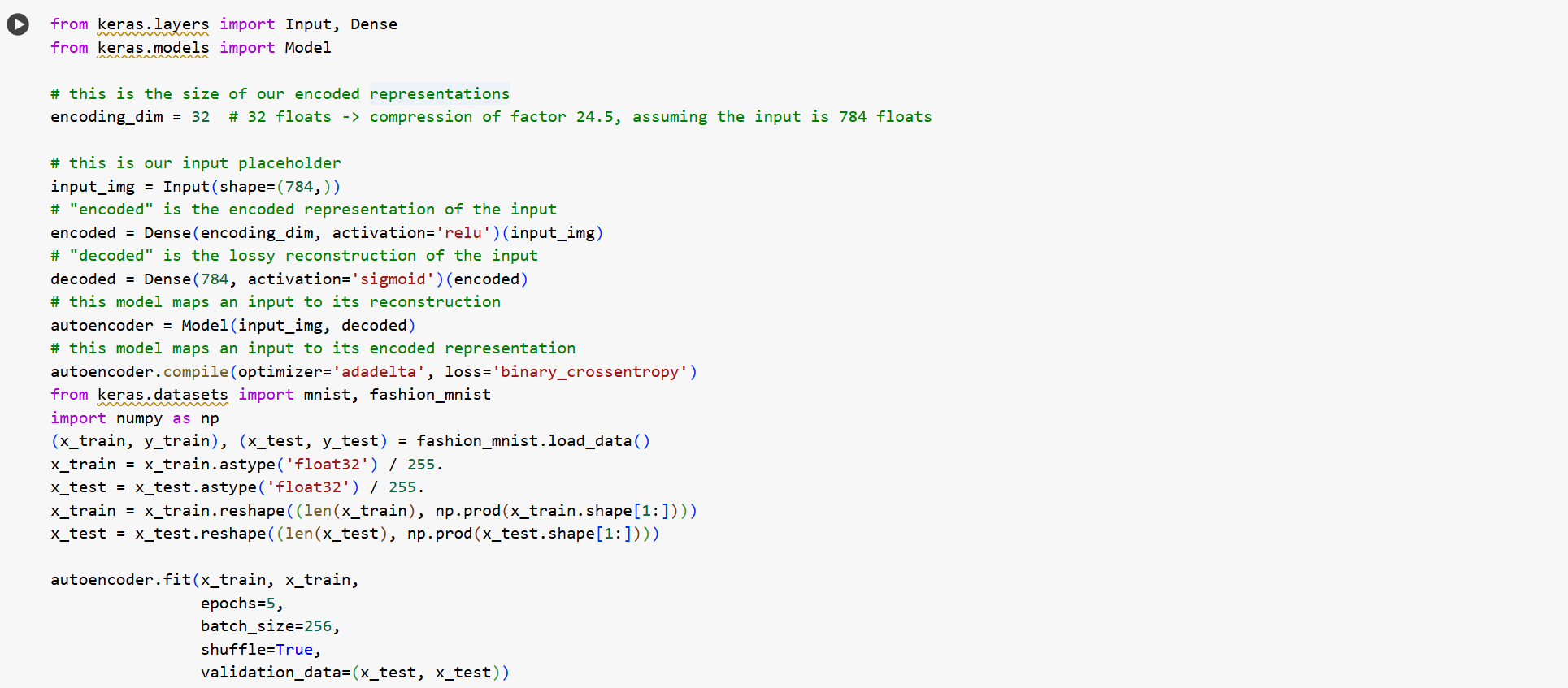
Trained for 5 epochs with a batch size of 256 on Fashion MNIST data to learn compression and reconstruction.

Applications:

Dimensionality Reduction: Reduces data size while preserving key features.

Feature Learning: Helps discover useful features for other tasks.

Denoising: Can be adapted for removing noise from images.



**Output:**



**Question 2:**

**Code:**



**Explanation:**

Model Architecture:

Input Layer: Accepts flattened 28x28 grayscale images (784 pixels).

Encoding Layer: Compresses the input to a 32-dimensional representation.

Hidden Layer: 64 units for more complex feature extraction.

Decoding Layer: Reconstructs the original image from the compressed form.

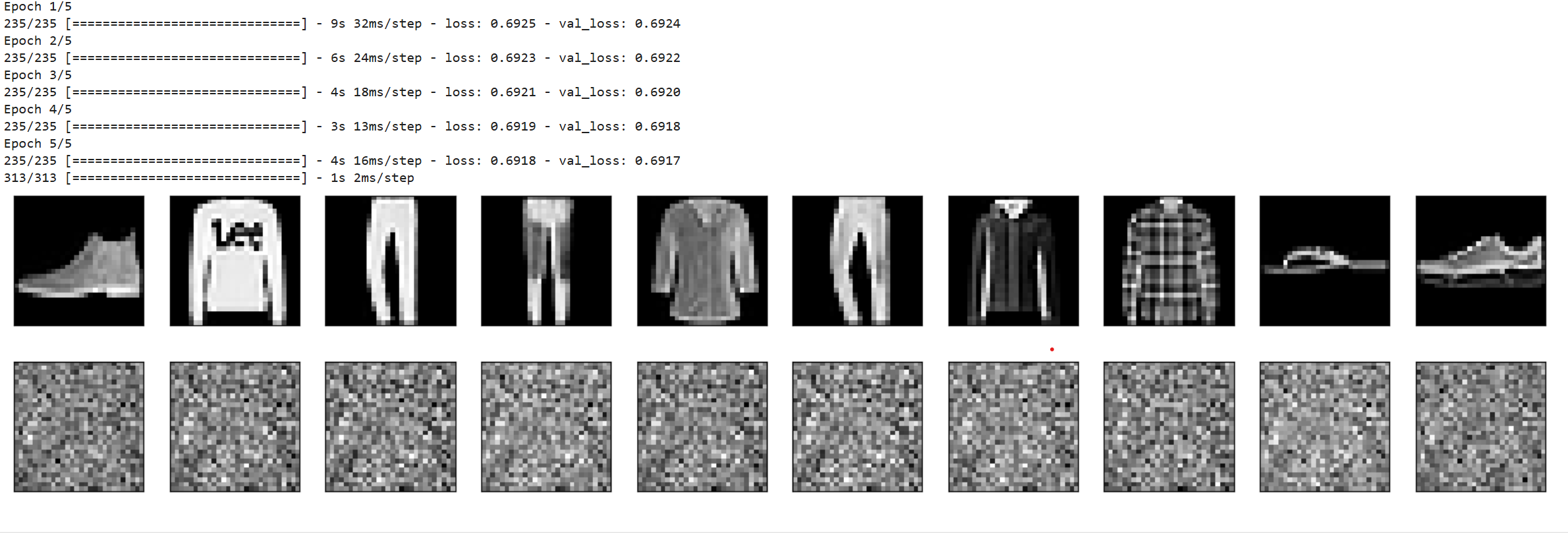
Compilation:The model uses Adadelta optimizer and binary crossentropy loss, ideal for reconstruction tasks.

Data Preparation: Fashion MNIST dataset is loaded, normalized to pixel values between 0 and 1, and reshaped to 784-dimensional vectors.

Training: The model is trained for 5 epochs with a batch size of 256 and validated on the test set.

Visualization: After training, the model predicts and compares the original and reconstructed images for 10 test examples, allowing visual assessment of the reconstruction quality.

**Output:**



**Question 3:**

**Code:**



**Explanation:**

Autoencoder Architecture:A simple neural network with an input layer, one hidden layer (encoded representation), and an output layer (decoded representation). It compresses the input into a lower-dimensional encoded form and reconstructs the output from this encoding.

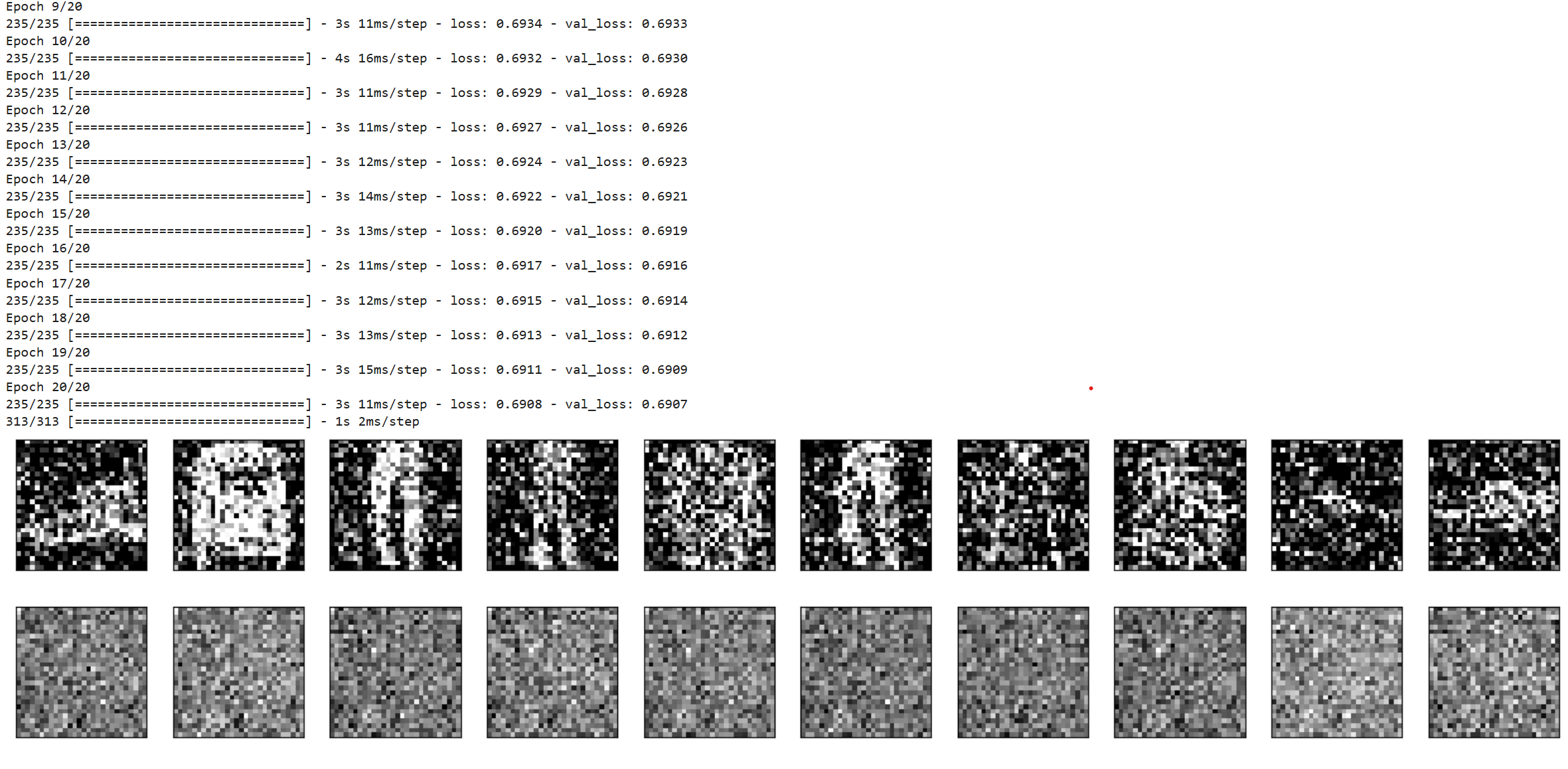
Data Preparation:The Fashion MNIST dataset is loaded, normalized, flattened into 784-dimensional vectors, and scaled to values between 0 and 1.

Noise Addition: Artificial Gaussian noise is added to both the training and test sets, and the pixel values are clipped to stay between 0 and 1.

Training: The model is trained on the noisy images as input, with the clean images as the target for reconstruction. It learns to denoise and recover the original images.

Evaluation and Visualization: After training, the model denoises the test set images. A comparison of the original noisy images and their denoised reconstructions is visualized side by side.

**Output:**



**Question- 4:**

**Code:**



**Explanation:**

Data Preparation: The Fashion MNIST dataset is loaded, normalized, and flattened into 784-dimensional vectors. Labels are one-hot encoded for multi-class classification.

Model Architecture: A simple neural network with an input layer, a hidden layer of 128 neurons with ReLU activation, and an output layer of 10 neurons (one for each class) with softmax activation. This is designed for multi-class classification.

Compilation and Training: The model is compiled with Adam optimizer and categorical crossentropy loss. Accuracy is tracked as a metric. The model is trained for 10 epochs with a batch size of 256, using training and validation data.

Performance Visualization: After training, training and validation accuracy and loss are plotted over epochs, helping visualize the model's learning and generalization ability on the validation set.

**Output:**

