CSE473s: Computational Intelligence – Fall 2025

Milestone 2 :

Name	Code
Sara Saber Samuel	2101138
Clara Ashraf Younan	2100932
Bishoy Tarek Soliman	2101067
Mina Ezzat Ragheb	2101363
Mark Matta Guirguis	2100372

Part 2 — Autoencoder and Latent Space Classification

1. Introduction

In this part of the project, an autoencoder neural network is implemented and trained on the MNIST handwritten digits dataset using a custom neural network library developed from scratch.

The objective is to learn compact latent representations of images and evaluate their usefulness for downstream classification tasks.

2. Dataset Description

The MNIST dataset consists of grayscale images of handwritten digits from 0 to 9.
Each image has a resolution of 28×28 pixels and is flattened into a 784-dimensional input vector.
Pixel values are normalized to the range [0, 1] before training.

The dataset is split into:

- 60,000 training samples
- 10,000 testing samples

3. Autoencoder Architecture

The autoencoder is composed of two main parts:

Encoder

The encoder compresses the input data into a low-dimensional latent space:

- Input layer: 784 neurons
- Hidden layer: 256 neurons with ReLU activation
- Latent layer: 64 neurons with ReLU activation

Decoder

The decoder reconstructs the original input from the latent representation:

- Hidden layer: 256 neurons with ReLU activation
- Output layer: 784 neurons with Sigmoid activation

The Sigmoid activation in the output layer ensures reconstructed pixel values remain in the normalized range [0, 1].

4. Training Procedure

The autoencoder is trained using:

- Loss function: Mean Squared Error (MSE)
- Optimization method: Stochastic Gradient Descent (SGD)
- Learning rate: 0.1
- Number of epochs: 20

During training, the model minimizes the reconstruction error between the input images and their reconstructed outputs.
The training loss decreases steadily, indicating successful learning of image representations.

5. Reconstruction Results

To evaluate reconstruction quality, original MNIST test images are compared with their reconstructed counterparts.
The reconstructed images preserve the overall structure and digit shapes, demonstrating that the autoencoder successfully captures meaningful features of the data.

6. Latent Space Feature Extraction

After training, the encoder portion of the autoencoder is isolated to generate latent feature vectors.
Each input image is mapped to a 64-dimensional latent representation.
These latent features serve as compressed and informative representations of the original images.

7. Classification Using Latent Features

A Support Vector Machine (SVM) classifier with an RBF kernel is trained using the latent representations produced by the encoder.
The classifier is evaluated on the test set and achieves significantly higher accuracy compared to an SVM trained directly on raw pixel data.

To assess performance, the following metrics are used:

- Classification accuracy
- Confusion matrix
- Precision, recall, and F1-score

8. TensorFlow Comparison

For validation, a reference autoencoder with the same architecture is implemented using TensorFlow/Keras.
Latent features extracted from the TensorFlow model are also classified using an SVM.
The classification performance is comparable to the from-scratch implementation, confirming the correctness and effectiveness of the custom neural network library.

9. Conclusion

In this part of the project, an autoencoder was successfully implemented from scratch and trained on the MNIST dataset.

The learned latent representations proved effective for classification tasks, outperforming raw pixel-based approaches.

The comparison with TensorFlow further validates the accuracy and reliability of the custom implementation.

This work demonstrates a solid understanding of unsupervised learning, feature extraction, and neural network training principles.