Lab 4: Neural Networks

Mamoru Ota, S7784976

***Abstract*—This report examines neural network architectures through various tasks, focusing on Task 1: Feedforward multi-layer networks (multi-layer perceptrons), Task 2: Autoencoder and Optional task: Neural networks in Keras. For Task 1, we analyze the effect of varying hidden layer sizes on classification accuracy using datasets such as Iris, Wine, and Thyroid. For Task 2, we employ an autoencoder to learn compressed representations of MNIST data, observing its capability to differentiate between handwritten digit classes. For Optional task, we implement an autoencoder on the MNIST dataset using in order to train a neural network for dimensionality reduction and reconstruction of images.**

I. Introduction

Neural networks offer versatile architectures for solving classification and representation learning problems. This report investigates:

- **Task 1:** Evaluating feedforward multi-layer networks (multi-layer perceptrons) for classification accuracy using three datasets (Iris, Wine, and Thyroid).
- **Task 2:** Implementing an autoencoder to encode and visualize compressed representations of MNIST data.
- **Optional Task:** Implementing an autoencoder on the MNIST dataset using Keras to train a neural network for dimensionality reduction and reconstruction of images.

Each task highlights unique aspects of neural network design and performance evaluation.

II. Methods

A. Task 1: Feedforward multi-layer networks (multi-layer perceptrons)

For Task 1, a 2-layer MLP (Multi Layer Perceptron) architecture was employed with the following properties:

- Hidden layer: Sigmoid activation function with varying unit sizes (2, 10, 50).
- Output layer: Softmax activation for multi-class classification.

Datasets included:

- **Iris Dataset:** 4 input features, 3 output classes.
- **Wine Dataset:** 13 input features, 3 output classes.
- **Thyroid Dataset:** 21 input features, 3 output classes.

All datasets were split into training (70%), validation (15%), and testing (15%).

Figure 1 illustrates the example of MLP architecture used in Task 1. The model consists of two layers: the first hidden layer employs a sigmoid activation function, while the output layer uses a softmax activation for multi-class classification. The number of input and output units corresponds to the dataset's features and classes, respectively.

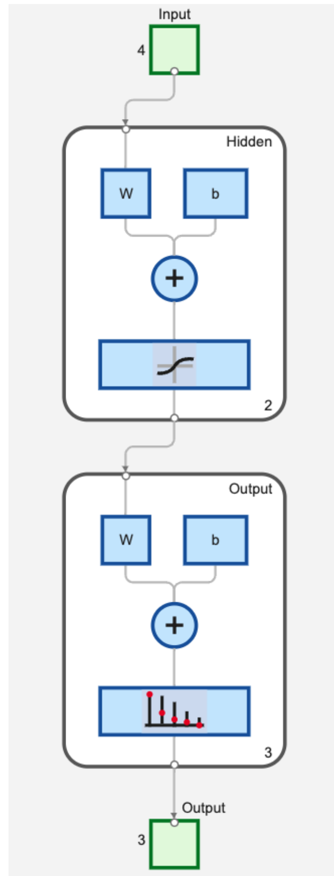

Fig. 1: Example of MLP architecture for Task 1. This model has 4 input units, 2 hidden units, and 3 output units. The hidden layer uses a sigmoid activation function, and the output layer uses a softmax activation.

B. Task 2: Autoencoder

The autoencoder used is a 3-layer neural network:

- Input layer: n_{inputs} dimensions.
- Hidden layer: n_{hidden} units ($n_{hidden} < n_{inputs}$) with sigmoid activation.
- Output layer: n_{inputs} units with linear activation.

The MNIST dataset was normalized to $[0, 1]$, and samples for two classes (e.g., digits 1 and 7) were extracted. The encoded representations were visualized in a 2D space. The architecture and parameters (hidden units, epochs) were specified interactively.

C. Optional task: Neural networks in Keras

The autoencoder architecture consisted of:

- An encoder that compressed the input image into a 32-dimensional latent space.
- A decoder that reconstructed the original image from the latent representation.

The network was trained on the MNIST dataset for 10 epochs using a batch size of 128. The Adam optimizer and Mean Squared Error (MSE) loss function were used during training. After training, the following visualizations were performed:

1) Latent space representation of the digits using t-SNE.
2) Comparison of original images and their reconstructions.

III. Experiments

A. Task 1: Feedforward multi-layer networks (multi-layer perceptrons)

Experiments for Task 1 evaluated test accuracy across datasets and hidden layer sizes. Table I summarizes the findings.

TABLE I: Test Accuracy for Different Hidden Layer Sizes.

Hidden Units	Iris (%)	Wine (%)	Thyroid (%)
2	95.65	92.59	93.70
10	100.00	100.00	94.35
50	100.00	100.00	93.89

B. Task 2: Autoencoder

For Task 2, the autoencoder was trained on MNIST digits, comparing pairs of classes (1 and 7, 1 and 0). The hidden layer had 50 units, and training was conducted over 10 epochs. Figure 2 and Figure 3 show latent spaces learned by the autoencoder for two pairs of MNIST digits. The separation of clusters reflects the similarity or dissimilarity between the selected digits.

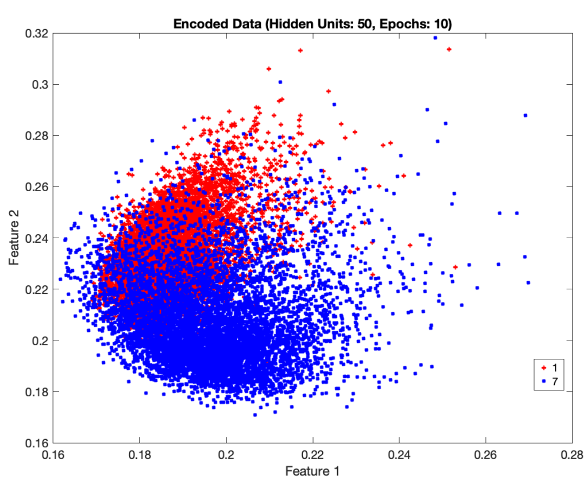

Fig. 2: Encoded representations for MNIST digits 1 and 7.

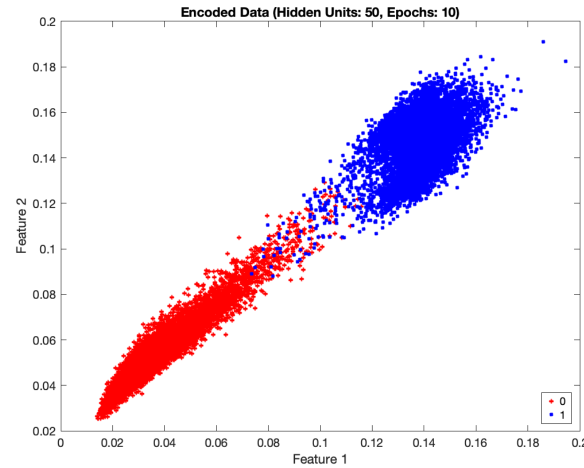

Fig. 3: Encoded representations for MNIST digits 1 and 0.

C. Optional task: Neural networks in Keras

Figure 4 shows the latent space of the MNIST digits visualized using t-SNE. Each point in the scatter plot corresponds to a digit, with colors representing the digit labels. The clusters demonstrate that the encoder effectively captured distinct features of each digit.

Figure 5 compares the original and reconstructed images. The reconstructed images preserved the key features of the original digits, indicating the encoder-decoder pair successfully learned the underlying data distribution.

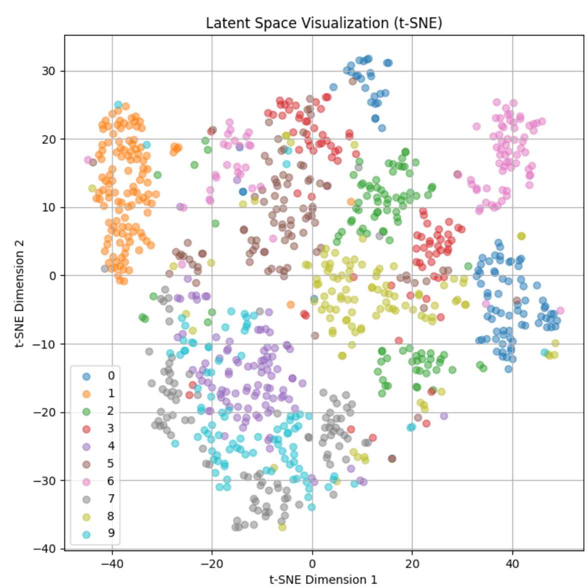

Fig. 4: Latent space visualization of MNIST digits using t-SNE. Each point represents a digit, colored by its label.

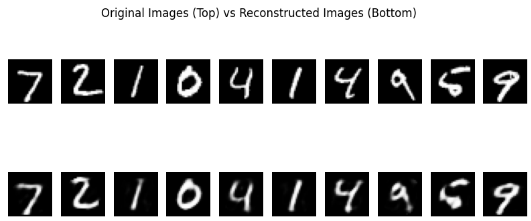

Fig. 5: Original images (top row) and reconstructed images (bottom row) generated by the autoencoder.

IV. Discussion

A. Task 1: Feedforward multi-layer networks (multi-layer perceptrons)

- **Iris Dataset:** Achieving high accuracy with only 2 hidden units highlights the simplicity of this dataset due to its low-dimensional input features.
- **Wine Dataset:** Perfect classification at 10 and 50 hidden units suggests the dataset's moderate complexity is well-suited to networks with modest capacity.
- **Thyroid Dataset:** A plateau in performance at 10 hidden units indicates that additional capacity does not benefit this higher-dimensional dataset, potentially due to overfitting.

B. Task 2: Autoencoder

- Encoded representations of digits 1 and 7 overlap, suggesting their features are more similar.
- Encoded representations of digits 1 and 0 are well-separated, reflecting their distinct features.
- These results demonstrate the ability of autoencoders to learn meaningful compressed representations that capture relationships between input features.

C. Optional task: Neural networks in Keras

The results of Task 3 highlight the effectiveness of autoencoders for dimensionality reduction and feature extraction:

- **Latent space visualization:** The t-SNE plot revealed clear clustering of digits, reflecting the ability of the encoder to extract meaningful features.
- **Image reconstruction:** The reconstructed images closely resembled the original inputs, demonstrating the decoder's capability to reconstruct data from compressed representations.

V. Conclusion

This study evaluated the performance of neural networks. Key findings include:

- For classification tasks (Task 1), simpler datasets like Iris achieved high accuracy with fewer hidden units, while more complex datasets like Wine and Thyroid required additional capacity.
- Autoencoder (Task 2) demonstrated the ability to learn compact, meaningful representations, as evidenced by the clear separation or overlap of MNIST digit classes in the latent space.
- Optional Task further emphasized the use of Keras for implementing and evaluating deep learning models, providing insights into latent representation and reconstruction quality.

Future work could explore deeper architectures, alternative activation functions, and regularization techniques to enhance network performance further.