MARMARA UNIVERSITY ENGINEERING FACULTY

EE 4065

Introduction to Embedded Digital Image Processing

Homework 4 Report

NAME:	*Baran*	*Muhammet Yücel*
SURNAME:	*ORMAN*	*ÇELİK*
NUMBER:	*150721063*	*150721024*

1. QUESTION

Handwritten Digit Recognition with Single Neuron (Section 10.9)

2.1 Method and Data Preparation

In this section, a logistic regression (single neuron) model was developed to detect whether an image contains the digit '0' or not.

- Dataset: MNIST training and test images were read from IDX format and converted into Numpy arrays.
- Feature Extraction: Instead of using raw pixel data (28x28), 7 Hu Moments were calculated for each image using OpenCV's 'cv2.HuMoments' function. These moments provide shape-based features invariant to scale and rotation.
- Standardization: The training dataset statistics (mean and standard deviation) were used to standardize all input features, improving convergence speed.
- Labeling: The dataset labels were converted into a binary format: '0' (class 0) and 'non-zero' (class 1).

2.2 Model Architecture and Training

A single neuron with a Sigmoid activation function was used to output a probability between 0 and 1.

- Optimizer: Adam (Learning rate: 0.001)
- Loss Function: Binary Crossentropy
- Class Weights: To handle class imbalance (fewer '0's than non-zeros), class '0' was assigned a weight of 8.
- Training Duration: 50 Epochs.

2.3 Results

The trained model was evaluated on the test dataset. The resulting Confusion Matrix is presented in Figure 1.

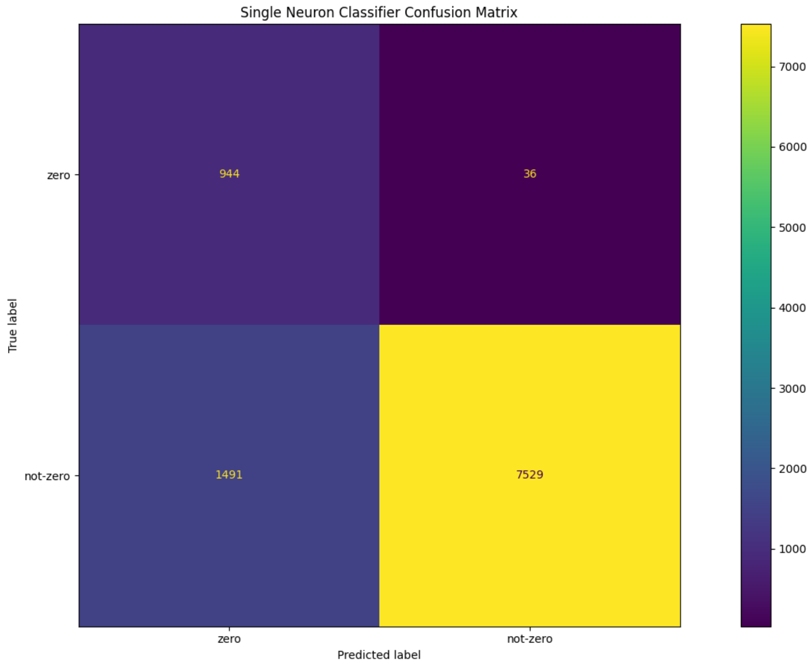

Figure 1 Single Neuron Train Results

According to the matrix:
- The model correctly classified 944 instances of the digit '0'.
- Only 36 instances of '0' were missed (False Negatives).
- For the 'non-zero' class, 7529 instances were correctly classified.

These results demonstrate that a simple single-neuron structure using only 7 Hu moment features is highly effective for this binary classification task.

2. QUESTION

Handwritten Digit Recognition with MLP (Section 11.8)

2.4 Method

In the second stage, the problem was expanded to classify all 10 digits (0-9) in the MNIST dataset.

- Feature Extraction: Similar to the first question, 7 Hu Moments were extracted and standardized for each image.
- Dataset: The original labels (0-9) were preserved for multi-class classification.

2.5 Model Architecture

A 3-layer Multi-Layer Perceptron (MLP) was constructed to improve classification performance:

1. Input Layer: 7 neurons (for Hu moments).
2. Hidden Layer 1: 100 neurons, ReLU activation.
3. Hidden Layer 2: 100 neurons, ReLU activation.
4. Output Layer: 10 neurons (one for each digit), Softmax activation.

2.6 Training Details

- Optimizer: Adam (Learning rate: 0.001).
- Loss Function: Sparse Categorical Crossentropy.
- Callbacks: EarlyStopping was used to prevent overfitting, and ModelCheckpoint was used to save the best model weights.
- Training: Set to a maximum of 1000 epochs, but stopped early upon convergence.

2.7 Results

The performance on the test set is visualized in the Confusion Matrix (Figure 2).

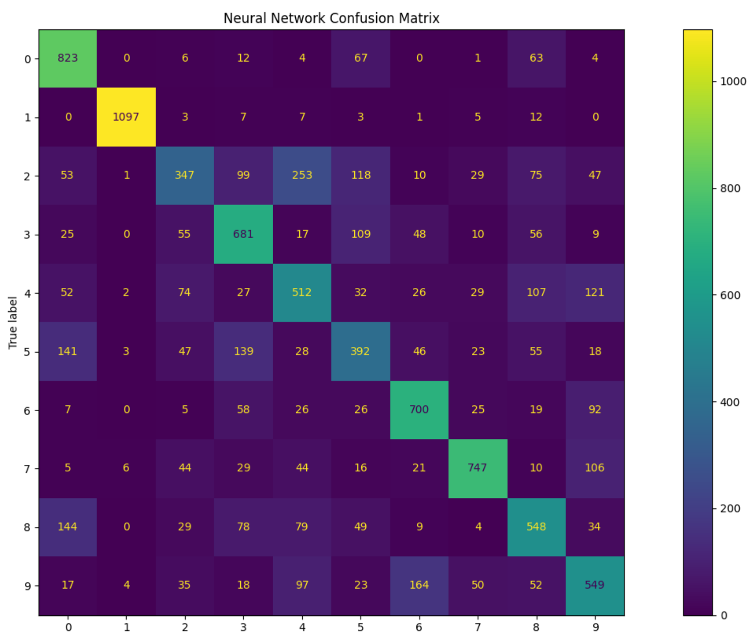

Figure 2 MLP(Multi-Layer Perceptron) Train Results

The results indicate:

- High values on the diagonal show the model is generally successful across all classes.
- For example, the digit '1' was correctly predicted 1097 times.
- Despite using a very small feature set (only 7 inputs), the MLP architecture successfully solved the multi-class problem.

3. DISCUSSION AND CONCLUSION

This homework explored two fundamental machine learning approaches suitable for embedded systems.

1. Efficiency: Using Hu moments (7 inputs) instead of raw pixels (784 inputs) reduced the input size by approximately 99%. This reduction is critical for microcontroller-based systems (like STM32) with limited memory and processing power.
2. Model Comparison:
- The Single Neuron approach offers high accuracy and low computational cost for specific binary tasks (e.g., detecting '0').
- The MLP approach effectively handles complex multi-class problems but requires more computational resources due to the hidden layers.

The results confirm that the methods described in the textbook are well-suited for embedded machine learning applications.