EE 4065 – Embedded Digital Image Processing
Homework 4 Report

Kaan Atalay (ID: 150720057)

December 26, 2025

Overview

This report presents two handwritten digit recognition applications following Sections 10.9 and 11.8 of the course textbook. The MNIST offline dataset is used for training and evaluation. Both models are lightweight and suitable as baselines for microcontroller deployment.

1 Dataset

- **Source**: MNIST IDX files placed under `data/MNIST-dataset/`.
- **Preprocessing**: Flatten to 784 features, normalize pixels to $[0, 1]$; no augmentation.
- **Split**: Train 55k / Val 5k (held-out from train set) / Test 10k.

2 Methodology for Q1 (Section 10.9)

- **Model**: Softmax regression ($784 \rightarrow 10$) implemented in numpy.
- **Training**: Cross-entropy via batch gradient descent; 5 epochs; batch size 256; learning rate 0.1; Glorot-like random init.
- **Export**: Parameters saved to `models/q1_softmax.npz`.
- **Evaluation**: Accuracy on train/val/test.

3 Methodology for Q2 (Section 11.8)

- **Model**: Single-hidden-layer MLP ($784 \rightarrow 128$ ReLU $\rightarrow 10$) in numpy.
- **Training**: Cross-entropy with batch gradient descent; 5 epochs; batch size 256; learning rate 0.05.
- **Export**: Parameters saved to `models/q2_mlp.npz`.
- **Evaluation**: Accuracy on train/val/test.

4 Results

- **Q1 softmax**: Test accuracy 0.9117; validation 0.9256 after 5 epochs.
- **Q2 MLP**: Test accuracy 0.9163; validation 0.9334 after 5 epochs.
- **Observation**: The shallow MLP slightly outperforms softmax due to nonlinear hidden layer while remaining lightweight.

5 Deployment Notes

- Target STM32 part number, available flash/RAM (to be specified). Both models are small: Q1 parameters $\approx$ 7.8k floats; Q2 $\approx$ 100k floats.
- Convert weights to fixed-point (e.g., 8-bit) for MCU; map matrix multiplies to CMSIS-NN or TFLite Micro kernels.
- Measure latency on target board or approximate cycles using MAC counts; ensure RAM fits activations (Q2 hidden layer 128 floats).

6 Conclusion

Summarize comparative performance of Q1 and Q2 approaches, discuss trade-offs for embedded deployment, and outline potential improvements (pruning, further quantization, or optimized kernels).

Repository and Submission

- GitHub repository: *(link to be added after push).*
- Contents: code, models, report PDF, and sample outputs.
- Submission: share repo link via email to the instructor.