

# RBE 474X

## Project 1

Colin Balfour

School of Robotics Engineering  
Worcester Polytechnic Institute  
Worcester, Massachusetts 01609-2247

Khang Luu

Worcester Polytechnic Institute  
Worcester, Massachusetts 01609-2247

Thanh Nguyen

Worcester Polytechnic Institute  
Worcester, Massachusetts 01609-2247

### I. PART 1

For this part, we are implementing custom layers for a multi-layer perceptron.

#### A. Results

All implementation yielded the same results as with the Torch's built-in layers. The output is in the jupyter notebook.

### II. PART 2

#### A. Implementation

We are training the MLP using the custom layers implemented in Part 1.

#### B. Results

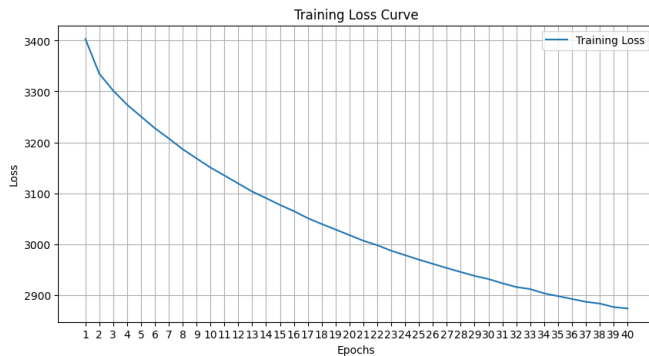


Fig. 1. Training loss curve for part 2

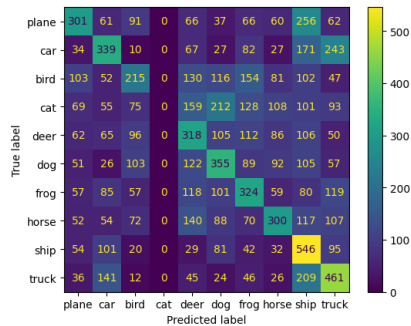


Fig. 2. Confusion matrix for part 2, epoch 1

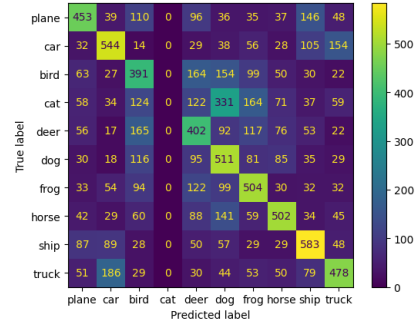


Fig. 3. Confusion matrix for part 2, epoch 39

### III. PART 3

#### A. Implementation

We used Pytorch's built-in CNN layers to train a CNN model.

#### B. Results

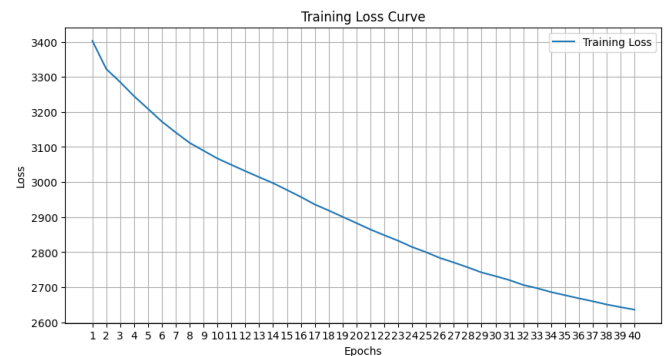


Fig. 4. Training loss curve for part 3

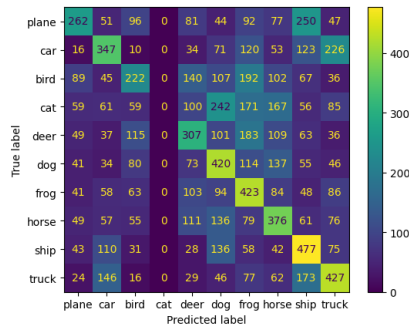


Fig. 5. Confusion matrix for part 3, epoch 1

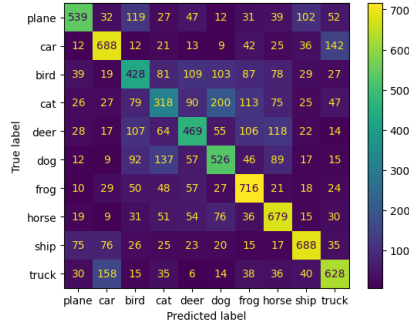


Fig. 6. Confusion matrix for part 3, epoch 39

#### IV. PART 4

- A.
- B.
- C.

#### V. PART 5

- A.
- B.
- C.

#### VI. ACCURACY COMPARISON BETWEEN MLP, CNN (TORCH LAYERS) AND CNN (CUSTOM LAYERS)

On training with the same dataset and an epoch of 40, CNN (torch layers) performed consistently better than MLP (56.79% vs. 43.68%)

With these inputs and hyperparameters, training the MLP took 8m 36s while the CNN with torch layers took 7m 28s.