EE456 Computer Assignment 3

Due: 2025-Nov-11 at 23:59

Overview

In this assignment, you will implement two neural network models on the MNIST handwritten digits dataset. The first part requires you to train a Multilayer Perceptron (MLP), while the second part requires you to design and implement a Convolutional Neural Network (CNN). This assignment emphasizes correct handling of data pipelines, training and evaluation loops, and hyperparameter tuning.

Learning Objectives

- Practice organizing Python code in a clear and reproducible way.
- Load and preprocess image data with `torchvision`.
- Train neural networks with PyTorch and understand training dynamics.
- Explore the effect of hyperparameters such as learning rate, batch size, optimizer, number of layers, and hidden sizes.
- Compare MLPs and CNNs in terms of accuracy and generalization on image classification.
- Visualize sample images, training curves, and error cases.

Dataset

You must use the MNIST dataset, which is available directly from `torchvision.datasets`. The dataset consists of 70,000 grayscale images of handwritten digits (28×28 pixels). It is split into 60,000 training images and 10,000 test images.

Classes correspond to the digits:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Part A: MLP

- Use the provided `mlp.py` file.
- **Task 1**: Directly run the provided `mlp.py` code. The program will display 10 sample images (digits 0–9). Modify the code so that the first 9 output images show your student ID, and make sure the last digit of your ID is set to 0. Show this images in your report.
- **Task 2**: After displaying the images, the program performs one round of training. Is this a good result? Explain why or why not.
- **Task 3**: Modify the number of epochs to 15, then run the program again. By increasing the number of training epochs, observe the results. Did this improve the performance? Why or why not?
- **Task 4**: On the basis of Task 3, modify the batch size to 32, then run the program again. Does the result improve? Has the training speed changed? What about 64?

- **Task 5**: On the basis of Task 4, modify the batch size to 64 and the number of epochs to 30. Set the learning rate to 0.5 and run the program again. Did the result become better or worse? Then repeat the experiment with learning rates 0.1, 0.01, and 0.001. Compare the outcomes. What do you observe when comparing the results across different learning rates?
- **Task 6**: What is the structure of this neural network? What is the best accuracy you have obtained so far?
- **Task 7**: Double the number of neurons in all hidden layers, and keep `epoch=10` and `learning rate=0.01`. Run the program again. Compared with the results you obtained in Task 5 at the same learning rate, did the performance improve? Provide an explanation.
- **Task 8**: Add one more hidden layer to the network. You may define the number of neurons in each layer and choose a set of hyperparameters by yourself, until you achieve a better result. Report the hyperparameters you used, the network structure (layer sizes), and the final performance you obtained.

Part B: CNN

- **Task 1**: Complete the `StudentCNN` class exactly as specified below (in this order).
  - **Block 1**: `Conv2d(1, 32, kernel_size=3, padding=1)` → `ReLU` → `Conv2d(32, 64, 3, padding=1)` → `ReLU` → `MaxPool2d(2)`
  - **Block 2**: `Conv2d(64, 128, 3, padding=1)` → `ReLU` → `MaxPool2d(2)`
  - **Head**: `Flatten` → `Linear(128*7*7, 256)` → `ReLU` → `Linear(256, 10)`

  Notes: Input is MNIST `(N, 1, 28, 28)`. After two `MaxPool2d(2)`, the feature map size is `(128, 7, 7)`.
- **Task 2**: Tune hyperparameters (e.g., learning rate, epochs, batch size, `USE_BN`, `DROPOUT_P`) to achieve at least **98% testing accuracy**. Briefly describe which settings worked.

Deliverables

You must submit a report named `studentname_report.pdf`.

- For each task, clearly mark the section in your report with the corresponding task number.
- After each code modification and program execution, include a screenshot of the training output.
- Provide answers to the questions for each task. Explanations should be concise and connected to the observed results.

Submission

Submit your code `studentname_MLP.py`, `studentname_CNN.py` and your report `studentname_report.pdf`.