Handwritten Number Recognition with PyTorch

85 Points

CS 360
Spring 2022

1 Objective

In this project you will use neural networks and cross validation to identify handwritten digits from black and white image sources.

2 Groups/Collaboration

You may collaborate with your fellow students.
That being said each student will be responsible for turning in their own code.
The following are things which constitute collaboration:

- Asking a fellow student to explain an approach.
- Asking a fellow student to check a few lines of code for mistakes.
- Discussing potential approaches to solving the problem.
- Distribution of test cases.

Collaboration with a fellow student needs to be documented. This is as simple as mentioning it in an acknowledgements section in your write up. Please detail the extent of the collaboration as well.

The following are things which do NOT constitute collaboration:

- Copying large chunks of code.
- Copying any part of the write up.
- Using ideas/code without any knowledge of the other party.
- Getting code from a fellow student who is not in this class.
- Getting another student to fix your code.

The actions above are outlines of what is and what isn't collaboration. If you don't know if something would be collaboration please email me, but usually if you are wondering if something would be collaboration it probably isn't. If you take any of the above actions, or similar, it will be considered plagiarism and you will receive a 0. Additionally, normal reporting procedures for academic dishonesty will be followed.

3 Project

This project is to be done in the Python programming language using the PyTorch framework. Your program should be able to do the following tasks:

1. Identify which of 5 different numbers a black and white image represents.
2. Create at least 3 different types of neural network structure.
3. Assess which structure of neural network works best to classify these images.

Following the project you will be required to submit a paper at least 1.5 (and no more than 5) pages (excluding acknowledgements) in length (style details below) detailing the performance of your algorithm (how well did it predict who wrote what,) any trouble you had, and surprising results.

3.1 Program Flow

Your program should have the following rough flow:

- Load the training set of examples.
- Setup several neural network models.
- Split the data setup for use in cross validation.
- Perform training and cross validation.
- Determine the best model type to use.
- Train a model of the best model type using all training examples.
- Test the accuracy of that model on the test data.

3.2 Data File Format

Each data file will be in CSV format with ONE image per line and the first element is the class label.
All data can be found in the mnist.zip file on Canvas under the data folder.

4 Write Up Requirements

You write up must have the following style:

- 12pt Time New Roman Font
- 1” margins
- Single spaces after periods
- Single spaced

Your write up should include the following information in the top left corner:

1. Name
2. Class
3. Year/Semester

The contents of your write up are up to you, but should include the following:

- What structures did you go with an why?
- Why do you suppose a particular structure worked better than others?
- Results of your cross validation and final learning accuracy numbers.
- Potential improvements.
- Places where you had trouble with the assignment and how you rectified them.
- Acknowledgements of any person you collaborated with.

5 Extra Credit

If you finish the project early (and it is working) you may contact me about extra credit on the project.

6 Rubric

Item	Points
Code Style and Readability	15
Program Runs	10
Correctly trains the neural networks	20
Correctly determines best model type and has sufficient number of types.	10
Write Up	30

7 Deliverables

All code, along with the write-up in either .docx or .pdf format, should be zipped and submitted to Canvas by the due date (listed on Canvas) using the file naming convention <*your_last_name*>_CS360_project_4.zip