FINAL PROJECT - SPRING 2020
FUNDAMENTALS OF MACHINE LEARNING

Handwritten Digits Classification

Project Due: April 24, 2020, 11:59 pm (report and code)

In the final project, you will develop a method to classify handwritten digits. You can implement this yourselves or using a package/library. You can use any packages that come as a default option with Anaconda or Pytorch. *I need to be able to run your implementation on my machine. So, be sure to get approval from me for any special packages! If I cannot run the code, you will lose a significant number of points.*

You may work in groups of no more than 4 individuals.

Project Report: You should write a report that includes the sections listed below. Your report should follow the IEEE transactions format (single spaced, double column). Focus your report on your training and testing strategies for the contest and any unique implementations. Templates for the IEEE transactions format can be found here: http://www.ieee.org/publications standards/publications/authors/author templates.html

The maximum number of pages for the report is 4. If there are any pages beyond page 4, they will discarded and not read or graded. It should be written with correct English grammar and spelling. Be precise - use pseudocode or equations to be precise.

Abstract A summary description of the contents of the report and your findings.

Introduction Overview of your experiment and a literature review. For the literature review, include any references to any relevant papers for your experiment. So, whatever you decide to do, search the ACM and IEEE (or other) literature for relevant papers to read and refer to.

Implementation Describe and outline any specific implementation details for your project. A reader should be able to recreate your implementation and experiments from your project report. How will you identify digits that were not in the training data?

Experiments Carefully describe your experiments with the training data set and any other small toy data sets you constructed. Include a description of what the goal each individual experiment is and what your findings are.

This is the bulk of what you will be graded on - if your experimental design is not sound or your experiments do not make sense, you will lose points.

Conclusions Describe any conclusions or things you learned from the project. Your conclusions must follow from what you did. Do not copy something out of a paper or say something that has no experimental support in the Experiments section.

References Listing of all references in IEEE bibliography format.

Submission Details: April 24 at 11:59pm turn in your project report and your code (in a zip folder) in our Canvas page. Be sure your code contains the following files: **train.py** (includes a *function* that will run your training code on an input data set X and desired output vector Y. Any parameter settings must be easy to find and modify), **test.py** (includes a *function* that will run your testing code on an input data set X. Note: Your test.py code should already be trained and have parameters set! Any parameter settings must be easy to find and modify. It should return a vector with the class label associated with each input data point in X) and a concise **README.txt** file that clearly illustrates how to run your code. Your classification accuracy on a small test data set will factor into your project grade.

Grade Details: Your grade will be determined using the following breakdown:

- 25% Implementation
  - Turn in code that runs correctly and easily on my machine. This requires a very clear README and easy to modify parameter settings. This also requires clearly listing what packages/libraries are needed to run your code - and checking with me before the due date to ensure I have those libraries.
  - Turn in code that follows the submission requirements described above
- 25% Accuracy on "easy" blind test data set
  - The "easy" test set is composed of hand-written digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).
  - Your code should produce the labels '0' for '0', '1' for '1', '2' for '2', etc..
  - Full points on this component will be obtained if you correctly classify 90% of the blind test data or have a classification accuracy rate greater than the average classification accuracy rate of the class (whichever is lower).
- 50% Project report
  - This component will be graded based on the requirements outlined above.

Extra Credit Contest: Your goal is to train your system to distinguish between handwritten digits. The teams with the best classification accuracy on the "hard" data set will get extra credit. The "hard" data set consists of the following characters: '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' and 'unknown'. There will be test data points from classes that do not appear in the training data. So, you will want to come up with a way to identify when a test point class is "unknown" or was not in the training data. The label you should return for this case is -1.

Please have your code output a class label that matches the class value in the provided training data. These should be: 0,1,2,3,4,5,6,7,8,9 and -1 respectively.