EEL 5840 Fundamentals of Machine Learning
Final Project - Spring 2021

Title: Handwritten Digits Classification

Project Due: Friday, April 23, 2021, 11:59 PM
Group Size: up to 4 individuals
Material Due: final report and code implementation

1. Description

In the final project, you will develop a Machine Learning algorithm to classify handwritten digits. The data set is to be collected by the students enrolled in this course. 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.*

1.1. Data Set

Each group will collect part of the training set that everyone will use to train their machine learning models. In order to collect the data, you will you will handwrite digits on paper, including digits from 0 to 9.

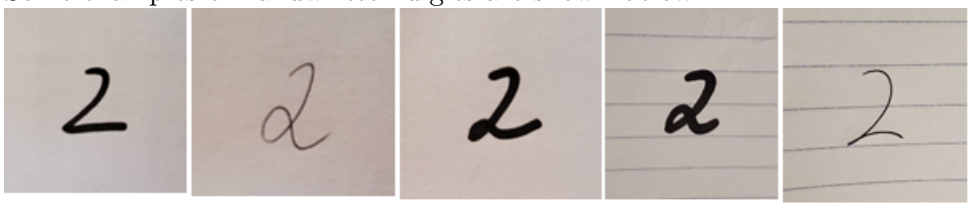
- The total number of digits is 10, which includes the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.

Each student should handwrite and take a picture of a total of **10 trials** per digit, giving a total of 100 images per student. So, for a group with 4 members, there should be a total of 400 images.

- Your handwritten digits should be written on paper. If you use a digital environment to draw digits, points will be deducted.

- I recommend you to use a ball point pen or a sharpie.
- You should take pictures of different handwritten digits (not the same one). Take a picture with your phone, crop it and make sure the digit is centered in the picture. Save this image as a jpg.
- Use the starter code from the *Data Collection* assignment to create numpy arrays will jpg images.

Some examples of handwritten digits are shown below:



I recommend you to save your files using a **coding system**, e.g. **ID-trial-digit**.

- First give a number from 1 to 4 to each team member, this is the ID. Then, for example, when team member with ID 4 is recording hers/his 5th handwriting of digit 8, the file name should read "4-5-8.jpg".

After collecting all the data from all teams, I will be partitioning this data into a **training set** (about 70%) and an ***easy* test set** (about 30%). You will be given the **training set** to fit your model. I will hold the ***easy* test set** until after you submit your code implementation and report. This blind test set will be used for grading.

I will also create a separate ***hard* test set** that will contain images from the classroom but also include other images with digits or other objects outside the provided labels. This test set will be used for **extra credit contest** - see details below.

1.2. 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).

- You can find a (word doc or LaTeX) template for the IEEE transactions format here: `https://www.ieee.org/conferences/publishing/templates.html`

Focus your report on your training and testing strategies for the contest and any unique implementations. The **maximum number** of pages for the report is 4. If there are any pages beyond page 4, they will be discarded and not read or graded. It should be written with correct English grammar and spelling. Be precise - use pseudo-code or equations to be precise.

For full credit consideration, your report should include the following sections:

- *Abstract.* A summary description of the contents of the report and your findings.
- *Introduction.* Overview of your experiment/s and a literature review. For the literature review, include any references to any relevant papers for your experiment/s. 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. If you participate in the extra credit contest, be sure to describe the methodology on how you will identify emotional tones or other sounds that were not in the training data?
- *Experiments.* Carefully describe your experiments with the training data set and any data augmentation set you constructed or existing data sets. Include a description for the goal of each experiment and experimental findings. 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.

When writing the report as a group, I recommend you to use **Google Docs** using your UFL account. This way you can all make synchronous and simultaneous edits in your project report.

1.3. Project Code Implementation

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 can implement your algorithm using Jupyter Notebook or your favorite development environment (e.g. mine is Spyder). Your final code submission should contain 3 files:

- README file - directly editable in your team repository
- train.py or a Notebook with a function "train"
- test.py or a Notebook with a function "test". This function should receive data and labels in the same format as the training data and output an accuracy value and the predicted labels.
- if you compete in the contest, you can create a separate file for testing on the hard test set. This function should receive data and labels in the same format as the training data and output an accuracy value and the predicted labels.

1.4. Submission Details

Turn in your project report and your code on your group GitHub repository on **Friday, April 23 at 11:59 PM**. In Canvas, you should submit your GitHub URL **AND** the project report.

Be sure your repository 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 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.

2. Grading Details

Your grade will be determined using the following breakdown:

- 10% Data Collection
  - You will be graded on data collection. Each person should collect 100 pictures of handwritten digits with the specifications mentioned in section 1.1.
- 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 the held-out blind test set with all 10 labels (digits 0 through 9).
  - Your code should produce the numerical label for each digit. For example, if an image has number 8 then label is 8; if it has the number 0 then label is 0.
  - 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).
- 40% Project report
  - This component will be graded based on the requirements outlined in section 1.2.

2.1. Extra Credit Contest

The goal of this project is to implement an end-to-end system to distinguish between handwritten digits from pictures. The teams with the best classification accuracy on the "hard" data set will get extra credit. The "hard" data set will also have all 10 digits (labels 0, 1, 2, 3, 4, 5, 6, 7, 8, 9) and class unknown (label -1). 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 test function 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.