MiniProject 3: Modified MNIST

COMP 551, McGill University
Professor: William L. Hamilton

February 26, 2019

Please read this entire document before beginning the assignment.

Preamble

- This mini-project is **due on March 18th at 11:59pm.** Late work will be automatically subject to a 20% penalty and can be submitted up to 5 days after the deadline. No submissions will accepted after this 5 day period.

- This mini-project is to completed in groups of three. All members of a group will recieve the same grade. It is not expected that all team members will contribute equally to all components. However every team member should make integral contributions to the project.

- You will submit your assignment on MyCourses as a group, and you will also submit to a Kaggle competition. **You must register in the Kaggle competition using the email that you are associated with on MyCourses (i.e., @mail.mcgill.ca for McGill students). You can register for the competition at:** `https://www.kaggle.com/t/1161924b38704e528eb203c8ac568d49`. As with MiniProject 1 and 2, you must register your group on MyCourses and any group member can submit. **You must also form teams on Kaggle and you must use your MyCourses group name as your team name on Kaggle. All Kaggle submissions must be associated with a valid team registered on MyCourses.**

- You are free to use any Python library or utility for this project (as well as bash or sh scripts). All your code must be compatible with Python 3.

Background

In this mini-project the goal is to perform an image analysis prediction challenge. The task is based upon the MNIST dataset (`https://en.wikipedia.org/wiki/MNIST_database`). The original MNIST contains handwritten numeric digits from 0-9 and the goal is to classify which digit is present in an image.

Here, you will be working with a Modified MNIST dataset that we have constructed. In this modified dataset, the images contain more than one digit and the goal is find which number occupies the most space in the image. Each example is represented as a 64×64 matrix of pixel intensity values (i.e., the images are grey-scale not color). Examples of this task are shown in Figure 1. **Note that this is a supervised classification task: Every image has an associated label (i.e., the digit that occupies the most space) and your goal is to predict this label.**
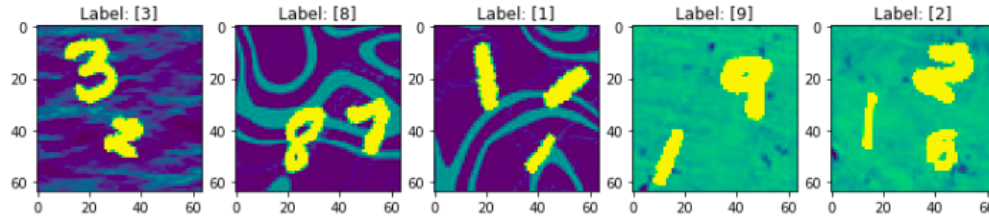
Figure 1: Example images and their associated labels. Note that the images are grey-scale (i.e., there is only one scalar value per pixel), but these images were generated using matplotlib.pyplot.imshow function, using a colormap for improved readability.

Task

You must design and validate a supervised classification model to perform the Modified MNIST prediction task. There are no restrictions on your model, except that it should be written in Python. As with the previous mini-projects, you must write a report about your approach, so you should develop a coherent validation pipeline and ideally provide justification/motivation for your design decisions. **You are free to develop a single model or to use an ensemble; there are no hard restrictions.**

Deliverables

You must submit two separate files to MyCourses (**using the exact filenames and file types outlined below**):

1. **code.zip**: A collection of .py, .ipynb, and other supporting code files, which must work with Python version 3. **It must be possible for the TAs to reproduce all the results in your report and your Kaggle leaderboard submissions using your submitted code.** Please submit a README detailing the packages you used and providing instructions to replicate your results.

2. **writeup.pdf**: Your (max 5-page) project write-up as a pdf (details below).

Project write-up

Your team must submit a project write-up that is a maximum of five pages (single-spaced, 10pt font or larger; extra pages for references/bibliographical content and appendices can be used). We highly recommend that students use LaTeX to complete their write-ups and use the bibtex feature for citations. **You are free to structure the report how you see fit**; below are general guidelines and recommendations, **but this is only a suggested structure and you may deviate from it as you see fit.**

Abstract (100-250 words) Summarize the project task and your most important findings.

Introduction (5+ sentences) Summarize the project task, the dataset, and your most important findings. This should be similar to the abstract but more detailed.

Related work (4+ sentences) Summarize previous relevant literature.

Dataset and setup (3+ sentences) Very briefly describe the dataset/task and any basic data pre-processing methods. **Note:** You do not need to explicitly verify that the data satisfies the i.i.d. assumption (or any of the other formal assumptions for linear classification).

Proposed approach (7+ sentences) Briefly describe your model (or the different models you developed, if there was more than one), providing citations as necessary. **If you use or build upon an existing model based on previously published work, it is essential that you properly cite and acknowledge this previous work.** Include any decisions about training/validation split, regularization strategies, any optimization tricks, setting hyper-parameters, etc. It is not necessary to provide detailed derivations for the model(s) you use, but you should provide at least few sentences of background (and motivation) for each model.

Results (7+ sentences, possibly with figures or tables) Provide results for your approach (e.g., accuracy on the validation set, runtime). You should report your leaderboard test set accuracy in this section, but most of your results should be on your validation set (or from cross validation).

Discussion and Conclusion (3+ sentences) Summarize the key takeaways from the project and possibly directions for future investigation.

Statement of Contributions (1-3 sentences) State the breakdown of the workload.

Evaluation

The mini-project is out of 100 points, and the evaluation breakdown is as follows:

- Performance (50 points)
  - The performance of your models will be evaluated on the Kaggle competition. Your grade will be computed based on your performance on a **held-out test set**. The grade computation is a linear interpolation between the performance of a random baseline and the 10th best group in the class, with the top-10 groups all receiving a perfect score.
  - Thus, if we let X denote your accuracy on the held-out test set, B denote the accuracy of the random baseline, and T denote the accuracy of the 10th best group in the class, the scoring function is:
  $$\text{points} = \max(\frac{50 * (X - B)}{T - B}, 50)$$
  - In addition to the above, the top-3 performing groups will receive a bonus of 5 points.
- Quality of write-up and proposed methodology (50 points)
  - Is your proposed methodology technically sound?
  - Does your report clearly explain your models, experimental set-up, results, figures (e.g., don't forget axis labels and captions on figures, don't forget to explain figures in the text).
  - Is your report well-organized and coherent?
  - Is your report clear and free of grammatical errors and typos?

– Did you go beyond the bare minimum requirements for the write-up (e.g., by including a discussion of related work in the introduction)?

– Does your report include adequate citations?

Final remarks

You are expected to display initiative, creativity, scientific rigour, critical thinking, and good communication skills. You don't need to restrict yourself to the requirements listed above - feel free to go beyond, and explore further.

You can discuss methods and technical issues with members of other teams, but you cannot share any code or data with other teams. Any team found to cheat (e.g. use external information, use resources without proper references) on either the code, predictions or written report will receive a score of 0 for all components of the project.

Rules specific to the Kaggle competition:

- Don't cheat! You must submit code that can reproduce the numbers of your leaderboard solution.

- Do not make more than one team for your group (e.g., to make more submissions). You will receive a grade of 0 for intentionally creating new groups with the purpose of making more Kaggle submissions.