

COMP-598: Applied Machine Learning

Mini-project #3: Difficult digits

Due on November 4, 11:59pm.

Background:

For this project, you will participate in an in-class Kaggle competition on image analysis. The goal is to devise a machine learning algorithm to automatically classify images of hand-written digits (from 0 to 9) represented in cropped image. The dataset for this task is based on the classic MNIST dataset (LeCun, Cortes, Burges: <http://yann.lecun.com/exdb/mnist/>), however each image has been modified using the following transformations:

- Embossing of the digit.
- Rotation (by a random angle, sampled from $[0, 360\text{deg}]$).
- Rescaling (from 28×28 pixels to 48×48 pixels).
- Texture pattern (randomly selected, and overlayed on background)

Examples of the training samples are shown on the right.

This dataset was constructed by researchers from the LISA lab at the University of Montreal (Pierre Luc Carrier, Aaron Courville). It is shared under the understanding that it can only be used for the purposes of this project, and cannot be shared outside of the course.

The competition, including the data, is available here (you can use the same Kaggle account as for the previous project): <https://inclass.kaggle.com/c/difficult-digits>
See the discussion board on *myCourses* for the link to join this (private) competition.

As for previous projects, this one should also be completed in a group of 3. Remember: you must work with different team members on each mini-project.

Requirements:

To participate in the competition, you must submit a list of predicted outputs for the test instances on the Kaggle website.

To solve the problem, you should try the following methods (the 4th one is optional):

- 1) A baseline learner (from lectures 2-5), such as logistic regression. This method must be fully implemented by your team, and corresponding code submitted.
- 2) A fully connected feedforward neural network (from lecture 14), trained by backpropagation, where the network architecture (number of nodes / layers) and learning rate are determined by cross-validation. This method must be fully implemented by your team, and corresponding code submitted.
- 3) A linear SVM (from lectures 12-13). You can use a package of your choice for this method, e.g. *scikit-learn*, *SVMTool*; you must provide appropriate references in your report.



- 4) Any *other* machine learning method of your choice. Existing packages can be used. In addition, you can use supplementary data of your choice to enrich the training set (e.g. the original MNIST dataset); you must provide appropriate references in your report.

Your written report should include results from all methods considered, as per the above categories. For the Kaggle competition, you can submit results from your best performing method, from any of these categories.

You must submit a report describing your approach. It should contain the following:

- Your team name (feature in the title).
- A list of team members (enter them as “authors” on the submission website).
- Main text with the following sections (note that “Related work” is not required this time):
 - o Introduction (overview of approach)
 - o Data pre-processing methods
 - o Feature design/selection methods
 - o Algorithm selection (for each of the 3 categories *baseline/neural net/open*)
 - o Optimization (if required for the algorithm)
 - o Parameter selection method (model order, learning rate, etc.)
 - o Testing and validation (detailed analysis of your results, outside of Kaggle)
 - o Discussion (pros/cons of your approach & methodology).
- When appropriate, use figures, tables and graphs to illustrate your work. Always include captions, axes labels, etc.
- References for any software package or supplementary dataset used.
- Before the references, add the following statement: “We hereby state that all the work presented in this report is that of the authors.” Make sure this statement is truthful!
- Spell-check and proof-read carefully.

The main text of the report should not exceed 6 pages. The appendix and references can be in excess of the 6 pages. The format should be double-column, 10pt font, min. 1” margins. You can use the standard IEEE conference format, e.g. ewh.ieee.org/soc/dei/ceidp/docs/CEIDPFormat.doc. Only acceptable file format is *.pdf*.

Evaluation criteria:

Marks will be attributed based on: 33% for performance on the private test set in the competition; 67% for the report including a clear description of the methods. The code will not be marked, but may be used to validate the other components.

For the competition, the performance grade will be calculated as follows: The top team, according to the score on the private test set, will receive 100%. A Random predictor, entered by the instructor, will score 0%. All other grades will be calculated according to interpolation of the Leaderboard scores between those two extremes.

For the report, the evaluation criteria include:

- Technical soundness of proposed methodology (feature selection, algorithms, optimization, validation plan)
- Clarity of methodology description
- Overall organization and writing.

The same evaluation criteria will be used for peer-reviews and evaluation by TAs and instructor.

Final grades and any late penalties will be attributed per team (i.e. all team members will get the same grade.) You can discuss methods and technical issues with members of others 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 or report will receive a score of 0 for both the report and competition.

Submission instructions:

Predictions on the test set must be submitted on Kaggle:

<https://inclass.kaggle.com/c/difficult-digits>

For the report, we will be using the same online conference management system:

<https://easychair.org/conferences/?conf=comp598>

You should use the same account as for the previous project, but submit to a new track, called “Mini-project #3”. The new report should be submitted as a “New Submission” (one per group), linking other team members as co-authors.

You should submit the code developed for this project as an “Attachment”. Acceptable file formats for this are *.zip*, *.gz*, *.tar*, *.tgz*. Make sure that the code is set up so that we can run it (e.g. include a README file).