MALIS Project 3:
Adaboost

In this last project, you will be reworking the problem studied in project 2 using multi-class Adaboost. An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

You will extend the standard Adaboost algorithm to solve the digit classification problem, from project 2, using a multi-class algorithm, rather than a one-vs-all approach.

Objectives

By executing this project, you will be able to:

1. Learn to read, understand and implement a research paper.
2. Gain understanding on the principles of ensemble methods and, more specifically, the Adaboost algorithm.
3. Learn to build complex solutions from previously implemented code.
4. Improve proficiency in the usage of machine learning libraries and on programming.

Part I – Revisiting the perceptron algorithm

Task 1. You will use a perceptron as the weak learner in Adaboost. Retrieve the code you used in project 2. Debug it, if necessary, making sure it works properly.

Part II – Implementing multi-class Adaboost

Task 2. Read the paper Multi-class Adaboost by Zhu et al (link).

Task 3. You will need to implement Algorithm 2 (page 351), which corresponds to the multi-class adaboost algorithm.

For this project, you are given complete freedom on how to structure your code. However, keep in mind the clarity, modularity and cleaningless of your code will be evaluated.

Task 4. Document your code.

Part II – Using multi-class Adaboost:

Task 5. You will now use your coded multi-class Adaboost to classify digits from the UCI ML handwritten digits dataset. You will use the perceptron you coded in project 2 as the weak learner.

Create a jupyter notebook denoted `experiment.ipynb`, where you will run your experiments. . Load the dataset and using the load_digits function in scikit-learn.

Task 6. You are expected to train, do model selection and evaluation of the final solution. Document all the steps clearly. Any additional experiments you perform should also be documented. Report all your results.

Report:

You need to prepare a 1000 words (max) report explaining your solution. How does your method work? Provide a brief description of your understanding of multi-class Adaboost. Include information about how your model was trained, the results obtained in the validation set and the strategy you used to choose a model (if any).

Contributions: Please include a section that describes what each team member worked on and contributed to the project. This is to make sure team members are carrying a fair share of the work for projects. This section does not count towards the total word count.

Deliverables:

Upload a zip file containing the report, all code files and any instructions required to run your code.

Important:

- Failing to submit a report leads to a mark of zero (0).
- Failing to run your code leads to a mark of zero (0).
- If ChatGPT is used, failing to report it and explaining its use leads to a mark of zero (0).