Lab Four: Evaluation and Multi-Layer Perceptron

Due Mar 25 by 11:59pm **Points** 10 **Submitting** a file upload

Lab Assignment Four: Evaluation and Multi-Layer Perceptron

In this lab, you will compare the performance of multi-layer perceptrons programmed in scikit-learn and via your own implementation.

This report is worth 10% of the final grade. Please upload a report (**one per team**) with all code used, visualizations, and text in a rendered Jupyter notebook. Any visualizations that cannot be embedded in the notebook, please provide screenshots of the output. The results should be reproducible using your report. Please carefully describe every assumption and every step in your report.

Dataset Selection

Select a dataset identically to the way you selected for lab one or lab three (table data or image data). You are not required to use the same dataset that you used in the past, but you are encouraged. You must identify a classification task from the dataset that contains **three or more classes to predict**. That is, it cannot be a binary classification; it must be multi-class prediction.

Grading Rubric

- Preparation (**15 points tota**l)
  - [**5 points**] (*mostly the same processes as from previous lab*) Explain the task and what business-case or use-case it is designed to solve (or designed to investigate). Detail exactly what the task is and what parties would be interested in the results. How well would your prediction algorithm need to perform to be considered useful by interested parties?
  - [**10 points**] (*mostly the same processes as from lab one*) Define and prepare your class variables. Use proper variable representations (int, float, one-hot, etc.). Use pre-processing methods (as needed) for dimensionality reduction, scaling, etc. Remove variables that are not needed/useful for the analysis. **Describe the final dataset that is used for classification** (include a description of any newly formed variables you created).
- Evaluation (**30 points tota**l)
  - [**15 points**] Choose and explain what metric(s) you will use to evaluate your algorithm's generalization performance. You should give a **detailed argument for why this (these) metric(s) are appropriate** on your data. That is, why does the metric evaluate performance in terms of the business case you argued for. Please note: rarely is accuracy the best evaluation metric to use. Think deeply about an appropriate measure of performance.
  - [**15 points**] Choose the method you will use for dividing your data into training and testing (*i.e.*, are you using Stratified 10-fold cross validation? Why?). **Explain why your chosen method is**

appropriate or use more than one method as appropriate. For example, if you are using time series data then you should be using continuous training and testing sets across time. Convince the reader that your cross validation method is a realistic mirroring of how an algorithm would be used in practice.

    - **Important**: You should use your chosen evaluation criteria and chosen method for dividing train/test data throughout the report. For example, arguing that f-score is the best evaluation method, but then using accuracy in a grid search will be **regarded as a conceptual error** and graded accordingly.

- Modeling (**45 points total**)
  - All modeling code must be written using only the instructor's template code. Do NOT copy this implementation form anywhere online. Not fully implementing the back propagation algorithm will result in a zero for the entire assignment.
  - [**20 points**] Create a custom implementation of the multi-layer perceptron. Start with the implementation given to you in the course. Update the MLP class to:
    - When instantiated, use a selectable phi function for the initial layer: either sigmoid or linear
    - Use a selectable cost function when instantiated: either quadratic or cross entropy
    - Add support for any number of hidden layers (user customizable).
  - [**15 points**] Tune the hyper-parameters of your MLP model (phi function, objective function, and number of layers). While tuning hyper-parameters, analyze the results using your chosen metric(s) of evaluation. **Visualize the evaluation metric(s) versus the hyper-parameters**. Conclude what combination of parameters are best.
  - [**10 points**] Visualize the magnitude of the gradients in each layer of the neural network versus the training iteration. Do the gradients stay consistent in each layer?
- Exceptional Work (**10 points total**)
  - You have free reign to provide additional analyses.
  - One idea (**required for 7000 level students**): Implement two more phi functions: ReLU and SiLU (also called Swish). Compare their performance to the linear and sigmoid phi functions.