Assignment 1: Getting Started with Machine Learning

COMP 551, Fall 2024, McGill University
Contact TAs: Huiliang Zhang and Shubham Vashisth

Please read this entire document before beginning the assignment.

Preamble

- This assignment is **due on September 30th at 11:59pm (EST, Montreal Time).** There is a penalty of 2^k percent for k days of delay, which means your grade will be scaled to be out of $100 - 2^k$. No submission will be accepted after 6 days of delay.
- **This assignment is to be completed in groups**. All members of a group will receive the same grade except when a group member is not responding or contributing to the assignment. If this is the case and there are major conflicts, please reach out to the Head TA for help and flag this in the submitted report. Please note that it is not expected that all team members will contribute equally. However every team member should make integral contributions to the assignment, be aware of the content of the submission and learn the full solution submitted.
- You will submit your assignment on MyCourses as a group. You must register your group on MyCourses and any group member can submit. See MyCourses for details.
- We recommend to use **Overleaf** for writing your report and **Google colab** for coding and running the experiments. The latter also gives access to the required computational resources. Both platforms enable remote collaborations.
- You should use Python for this assignment. You are free to use libraries with general utilities, such as numpy, pandas, scipy, and matplotlib for Python unless stated otherwise in the description of the task. In particular, in most cases you should implement the models and evaluation functions yourself, which means **you should not use pre-existing implementations of the algorithms or functions as found in scikit-learn, and other packages**. The description will specify this on a per-case basis.

Background

In this assignment you will implement two ML models—Linear regression, Logistic regression (with gradient descent)—and provide analysis on these two models on two distinct datasets. The goal is to get started with programming for Machine Learning and learn how these two commonly used models work.

Task 1: Acquire, preprocess, and analyze the data

Your first task is to acquire the data, analyze it, and clean it (if necessary). We will use two fixed datasets in this assignment, outlined below.

- **Dataset 1: Infrared Thermography Temperature** (regression)
  Link: `https://archive.ics.uci.edu/dataset/925/infrared+thermography+temperature+dataset`
  Column to predict: `aveOralM`

- **Dataset 2: CDC Diabetes Health Indicators** (classification)
  Link: `https://archive.ics.uci.edu/dataset/891/cdc+diabetes+health+indicators`
  Column to classify: `target`

The essential subtasks for this part of the assignment are:

1. Load the datasets into NumPy or Pandas objects in Python.
2. Prepare the dataset, including handle categorical features, null values (if any), scale features, etc. For this step, you may use pandas, numpy, and scalers from sklearn.preprocessing.
3. Compute basic statistics on the data to understand it better. For example, what are the distributions of the positive vs. negative classes (is the dataset balanced?), and what are the distributions of some of the numerical features?

Task 2: Implement the models

You are free to implement these models as you see fit, but you should follow the equations that are presented in the lecture slides, and you must implement the models from scratch (i.e., you **CANNOT** use scikit-learn or any other pre-existing implementations for implementing these models). However, you are free to use the relevant code given on the course website and colab notebooks.

In particular, your two main tasks in the part are to:

1. Implement analytical linear regression solution for Dataset 1.
2. Implement logistic regression with gradient descent for Dataset 2.
3. Implement mini-batch stochastic gradient descent for both linear and logistic regression.

Using the NumPy or Pandas package, however, is allowed and encouraged. Regarding the implementation, we recommend the following approach (but again, you are free to do what you want):

- Implement both models as Python classes. You should use the constructor for the class to initialize the model parameters as attributes, as well as to define other important properties of the model.
- Each of your models classes should have (at least) two functions:
  - Define a `fit` function, which takes the training data (i.e., $\mathbf{X}$ and $\mathbf{Y}$)—as well as other hyperparameters (e.g., learning rate and batch size)—as input. This function should train your model by modifying the model parameters.
  - Define a `predict` function, which takes a set of input points (i.e., $\mathbf{X}$) as input and outputs predictions (i.e., $\hat{\mathbf{y}}$) for these points.

Task 3: Run experiments

The goal of this assignment is to have you be familiar with how to train models.

Split each dataset into training, and test sets. Use test set to estimate performance in all of the experiments after training the model with training set. Evaluate the performance using the corresponding cost function for the classification and regression tasks. You are welcome to perform any experiments and analyses you see fit, **but at a minimum you must complete the following experiments in the order stated below**:

1. Report the performance of linear regression and fully batched logistic regression. For both datasets use a $80-20$ train/test split and report the performance on both training set and test set.
2. Report the weights of each of features in your trained models and discuss how each feature could affect the performance of the models.

3. Sample growing subsets of the training data (20%,30%,...80%). Observe and explain how does size of training data affects the performance for both models. Plot two curves as a function of training size, one for performance in train and one for test.
4. For both linear and logistic regression, try out growing minibatch sizes, e.g., 8, 16, 32, 64, and 128. Compare the convergence speed and final performance of different batch sizes to the fully batched baseline. Which configuration works the best among the ones you tried?
5. Present the performance of both linear and logistic regression with at least three different learning rates (your own choice).
6. Compare analytical linear regression solution with mini-batch stochastic gradient descent based linear regression solution. What do you find?

Note: The above experiments are the minimum requirements that you must complete; however, this assignment is open-ended. For example, what happens when you add momentum to the gradient descent implementation? what if you transform your data with non-linear bases discussed in the class? How about different evaluation metrics for classification and regression problem? Which kind of feature preprocessing improves performance? You do not need to do all of these things, but you should demonstrate creativity, rigour, and an understanding of the course material in how you run your chosen experiments and how you report on them in your write-up.

Deliverables

You must submit two separate files to MyCourses.

1. **code.ipynb or code.zip**: Your data processing, classification and evaluation code. Please keep all your running results in the code.ipynb file or in your code folder. Submit your code results separately along with your final report. Ensure that the **results in your Colab/.ipynb file match those in your report.**
2. **writeup.pdf**: Your (max 5-page, excluding references) write-up as a pdf (details below).

Assignment write-up

Your team must submit a assignment write-up that is a **maximum of** 5 **pages** avaibale template could be find at **Overleaf Assignment Template** (single-spaced, 11pt font or larger; minimum 0.5 inch margins, excluding references (if any). We highly recommend that students use LaTeX to complete your write-ups and you could share it with your groupmates via share project on overleaf. You have some flexibility in how you report your results, and you could follow the structure and minimum requirements listed below:

Abstract ($100-250$ **words)** Summarize the assignment task and your most important findings. For example, include sentences like "In this assignment we investigated the performance of two machine learning models on two benchmark datasets".

Introduction ($5+$ **sentences)** Summarize the assignment task, the two datasets, and your most important findings. This should be similar to the abstract but more detailed. You should include background information and citations to relevant work (e.g., other papers analyzing these datasets).

Datasets ($5+$ **sentences)** Very briefly describe the datasets and how you processed them. Present the exploratory analysis you have done to understand the data, e.g. class distribution. Highlight any possible ethical concerns that might arise when working these kinds of datasets.

Results (7+ sentences, possibly with figures or tables) Describe the results of all the experiments mentioned in **Task 3 i.e. experiments** (at a minimum) as well as any other interesting results you find (Note: demonstrating figures or tables would be an ideal way to report these results.

Originality/creativity (3+ sentences, possibly with figures or tables) Describe what you have done to go beyond the bare minimum requirements and your findings).

Discussion and Conclusion (5+ sentences) Summarize the key takeaways from the assignment and possibly directions for future investigation.

Statement of Contributions (1−3 sentences) State the breakdown of the workload across the team members.

Evaluation

This assignment is out of 100 points, and the evaluation breakdown is as follows:

- Completeness (20 points)
  - Did you submit all the materials?
  - Did you run all the required experiments?
  - Did you follow the guidelines for the assignment write-up?
- Correctness (40 points)
  - Are your models implemented correctly?
  - Are your reported performance close to our solution?
  - Do you observe the correct trends in the experiments (e.g., how performance changes as the minibatch or learning rates changes)?
  - Do you find notable features of the decision boundaries?
- Writing quality (30 points)
  - 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)?
  - Do you effectively present numerical results (e.g., via tables or figures)?
- Originality / creativity (10 points)
  - Did you go beyond the bare minimum requirements for the experiments?
  - **Note:** Simply adding in a random new experiment will not guarantee a high grade on this section! You should be **thoughtful and organized** in your report. That is, the distinctive ideas that you came up with should blend in your whole story. For instance, explaining the triggers behind them would be a great starting point.

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.**

Congratulations on completing your first course assignment! You are now familiar with the two most commonly used ML models. As the class continues, we will see more ML models and their interesting real-world applications.