CS4210 Summer 2023 Project Assignment 1

Total points: 100 + (with 10 extra-points from optional task 4)

Due date: Friday, 07/21/2023

Purposes:

1. Warm up your Python programming skills.
2. Understand the key concepts of machine learning
3. Master the training loop based on gradient descent optimization
4. Get familiar with linear regression and use Scikit-learn library.

Task Description:

In this assignment, you will use linear regression to study diabetes data, which has 10 baseline variables (age, sex, body mass index, average blood pressure, and six blood serum measurements) were obtained for each of 442 diabetes patients, and the response of interest (a quantitative measure of disease progression one year after baseline).

An iPython notebook is provided ("Assignment_1.ipynb"), where some of the early steps that prepare the training data and validation data have been implemented for you. (Note: due to the simplicity of this linear regression example, we just simply treat testing data as validation data.)

Please to implement the following tasks 1-4:

- **Task 1: Use** LinearRegression() in Scikit-learn library.
- **Task 2: Implement** analytical solution (based on closed form of the optimal solution given in slides) to perform linear regression.
- **Task 3: Implement** basic gradient descent to perform linear regression. Please tune the parameters to get close to the accuracy of the linear regression model from scikit-learn library. Also, **Use** matplotlib to plot the learning curves showing how training error and validation errors along iterations.
- **(Optional)Task 4: Implement** stochastic gradient descent method to perform linear regression. Please tune the parameters to get close to the accuracy of the linear regression model from scikit-learn library. Also, **Use** matplotlib to plot the learning curves showing how training error and validation errors along ***batches***.

In each of the tasks above, please show

- the resulting weights (intercept and coefficients)
- the resulting error $\ell(w) = \frac{1}{2N}\sum_{i=1}^{N}[t^{(i)} - y(x^{(i)})]^2$ on training data and validation data, respectively. (Please note the difference between our loss function and the one in Scikit-learn)

What to Submit?

1. A completed iPython notebook for tasks 1-3 (and optional task 4 if completed) (Note: properly comment your programs)
2. Please zip them into a file (yourname_assignment1.zip) and submit the zipped file in Canvas