CS4210 Fall 2023 Project Assignment 2

Total points: 100

Due date: Tuesday, 10/30/2023

Purposes:

1. Understand the key concepts of machine learning.
2. Get familiar with logistic regression and use Scikit-learn library.
3. Master the training loop based on gradient descent optimization.
4. Know how to implement a differentiable loss function and its corresponding gradient.

Task Description:

In this assignment, you'll use logistic regression for a binary classification task using a diabetes dataset. This dataset contains information on 768 diabetes patients, including 8 baseline variables: pregnancies, glucose levels, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, and age. The goal is to predict whether a patient is positive (1) or negative (0) for diabetes.

For your convenience, we have included the diabetes dataset named 'diabetes2.csv' in the zipped folder provided for this assignment. To proceed, please upload 'diabetes2.csv' to your Google Drive and ensure that it is located in the specified directory on your Google Drive: 'MyDrive/Colab Notebooks/datasets/diabetes2.csv'.

Please implement the following tasks,

- **(10 pts) Task 1: Preprocess the data,** such as feature scaling, using Scikit-learn's pipeline of transforms.
- **(10 pts) Task 2: Prepare 3 datasets:** training dataset, validation dataset, and testing dataset. A recommended split ratio is 60% training data, 20% validation data, and 20% testing data.
- **(10 pts) Task 3: Use** Scikit-learn's LogisticRegression() model to perform logistic regression.
    - **print** the log-loss errors of the trained model on the training, validation, and testing datasets using the log-loss function from sklearn.metrics.
    - **display** the confusion matrix for the trained model's predictions on the testing dataset..
- **Task 4: Implement** stochastic gradient descent method for logistic regression. Note that the log-loss function from sklearn.metrics is **NOT** allowed to use in this Task 4. In particular,
    - **(20 pts) implement** the cross-entropy loss $\ell(w)$, using the following formula provided.

      $\ell(w) = -\frac{1}{m}\sum_{i=1}^{m}\left[t^{(i)} log\left(y(x^{(i)})\right) + (1-t^{(i)}) log\left(1-y(x^{(i)})\right)\right]$,

      where
        - m is the batch size,
        - $x^{(i)}$ is the ith data sample of the batch and $t^{(i)}$ is the corresponding label

- $y(\cdot)$ is the model function of logistic regression that involves the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$.

- **(20 pts) implement** the gradient $\nabla_w \ell(w)$, using the following formula provided

$$\nabla_w \ell(w) = -\frac{1}{m} X^T \left(t - \sigma(Xw)\right)$$

- **(20 pts) use** matplotlib to plot learning curves that show the training error and validation errors across batches.
- **(5 pts)** tune the parameters to achieve results close to that of the logistic regression model from scikit-learn.
- **(5 pts) print** the trained model's cross-entropy errors on training dataset, validation dataset, and testing dataset, respectively.

Note that in Task 4, please use **matrix/vector operations** to evaluate the above **cross-entropy** and **gradient**, rather than a *for* loop.

What to Submit (on Canvas)?

1. A iPython notebook that contains your codes. A template can be found in the zipped folder of this assignment. Notes:
    1. non-executable programs result in a grade of zero.
    2. regular Python program file with ".py" is not acceptable.
    3. properly comment your programs.
    4. name your file using the following format: "*yourLastName_yourFirstName_assignment2.ipynb"* and submit it on Canvas