Homework Assignment 2a

Due: Friday, Mar. 8, 2024, 11:59 p.m. Mountain time

Total marks: 55

Question 1. [55 MARKS]

In this question, you will implement an algorithm to estimate $p(y|x)$, for a batch of data of pairs of (x, y): $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ where both $x_i, y_i \in \mathbb{R}$. We have provided you with a `julia` codebase as a `pluto notebook`, to load data and run an algorithm to estimate the parameters for $p(y|x)$. Please see this document linked here with instructions on how to get started with Julia.

It uses a simplified Kaggle data-set[1], where we use only the weight (x) and height (y) of 10,000 people. The provided data-loader function splits the data-set into a training set and testing set, after centering the variables so that the mean values for random variables x, and y will be zero. The notebook `A2.jl` has two baseline algorithms, for comparison: a random predictor, and a mean predictor. These baselines are sanity checks: your algorithm should be able to outperform random predictions, and the mean value of the target in the training set. Because of randomization in some of the learning approaches, we run each of the algorithms 10 times for different random training/test splits, and report the average error and standard error, to see how well each algorithm averaged across splits.

(a) [10 MARKS] In this assignment we assume we are learning $p(y|x) = \mathcal{N}(\mu = wx, \sigma^2 = 1.0)$ for an unknown weight $w \in \mathbb{R}$. For fun, you can derive the stochastic gradient descent update for the negative log-likelihood for this problem. But, we also provide it for you here. For one randomly sampled (x_i, y_i), the stochastic gradient descent update to w is:

$$w_{t+1} = w_t - \eta\,(x_i w_t - y_i)x_i \tag{1}$$

You will run this iterative update by looping over the entire dataset multiple times. Each pass over the entire dataset is called an *epoch*. At the beginning of each epoch, you should randomize the order of the samples. Then you iterate over the entire dataset, updating with Equation (1). Implement this algorithm to estimate b, by completing the code for `epoch!`.

(b) [10 MARKS] **WRITTEN:** Test `SimpleStochasticRegressor` implemented in (a) by running it, along with algorithms `RandomRegressor` and `MeanRegressor`. We fixed the stepsize for `SimpleStochasticRegressor` to 0.01 and the epochs to 30. Report and compare the mean and standard deviation of the test error (squared errors) for each of these three approaches.

(c) [15 MARKS] Next implement a mini-batch approach to estimate w, in `epoch!`. The idea is similar to stochastic gradient descent, but now you use blocks (or mini-batches) of N_{batch} samples to estimate the gradient for each update. For each epoch, you iterate over the dataset in order. For the first mini-batch update, the update equation is

$$w_{t+1} = w_t - \eta g_t \qquad \text{where } g_t = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} (x_i w_t - y_i)x_i \tag{2}$$

Then the next update uses the next mini-batch of points, $(x_{N_{\text{batch}}+1}, y_{N_{\text{batch}}+1}), \ldots, (x_{2N_{\text{batch}}}, y_{2N_{\text{batch}}})$. In total, for one epoch, you will complete $\lceil n/N_{\text{batch}} \rceil$ updates. Make sure your code is robust to the fact that the number of samples n might not be divisible by the batch size. The very last batch

[1] `https://www.kaggle.com/mustafaali96/weight-height`

could be shorter, and you should normalize that last mini-batch by the number of samples in the mini-batch rather than by N_{batch}.

(d) [10 MARKS] Next you will implement a more sensible strategy to pick stepsizes. You will implement the heuristic for adaptive stepsizes given in section 4.4 of the notes

$$\eta_t = (1 + |g_t|)^{-1} \tag{3}$$

where g_t is the gradient for the weight update $w_{t+1} = w_t - \eta_t g_t$. Implement `epoch!` when `MiniBatchHeuristicRegressor` is used, with the stepsize calculated using the above heuristic.

(e) [10 MARKS] **WRITTEN:** Run the mini-batch approach and the batch approach, both with the adaptive stepsizes given by Equation (3). Again, leave the number of epochs at 30 in the provided code, and leave the stepsizes as set in the code. Report and compare the test error of them with and without using adaptive stepsize selection strategies, along with the already given `Mean` baseline, based on the mean test error after 30 epochs. (No need to comment on the learning curve).

Homework policies:

Your assignment should be submitted as two pdf documents and a .jl notebook, on eClass. **Do not** submit a zip file with all three. One pdf is for the written work, the other pdf is generated from the .jl notebook. The first pdf should contain your answers for questions starting with "**WRITTEN:**". Your answers must be written legibly and scanned or must be typed (e.g., Latex). This .pdf should be named Firstname_LastName_Sol.pdf, For your code, we want you to submit it both as .pdf and .jl. To generate the .pdf format of a Pluto notebook, you can easily click on the circle-triangle icon on the right top corner of the screen, called Export, and then generate the .pdf file of your notebook. The .pdf of your Pluto notebook as Firstname_LastName_Code.pdf while the .jl of your Pluto notebook as Firstname_LastName.jl. All code should be turned in when you submit your assignment.

Because assignments are more for learning, and less for evaluation, grading will be based on coarse bins. **The grading is atypical**. For grades between (1) 81-100, we round-up to 100; (2) 61-80, we round-up to 80; (3) 41-60, we round-up to 60; and (4) **0-40, we round down to 0**. The last bin is to discourage quickly throwing together some answers to get some marks. The goal for the assignments is to help you learn the material, and completing less than 50% of the assignment is ineffective for learning.

We will not accept late assignments. There is no late penalty policy. The assignments must be submitted electronically via eClass on time, by 11:59 pm Mountain time on the due date. There is a grace period of 48 hours when assignments will be accepted. No submissions will be accepted after 48 hours after the deadline, and the assignment will be considered as incomplete if not submitted.

All assignments are individual. All the sources used for the problem solution must be acknowledged, e.g. web sites, books, research papers, personal communication with people, etc. Academic honesty is taken seriously; for detailed information see the University of Alberta Code of Student Behaviour.

Good luck!