Machine Learning

Homework 2

Faculty of New Sciences & Technologies
University of Tehran
Fall 2023

Part 1

General Homework Policies

1. Due date of this homework is on *Saturday 19 Aban 402 (10 Nov. 2023)*, so you need to submit it before the due date[midnight 19 Aban], otherwise you won't get the total score! We consider the following policy for the late homework,
   - Homework is worth full credit at the beginning of class on the due date.
   - It is worth half credit for the next 48 hours.
   - It is worth zero credit after that.
2. You are welcome to collaborate, cooperate, and consult with your classmates provided that you write-up the solutions independently.
3. Don't plagiarize! Write everything in your own words, and properly cite every outside source you use. Taking credit for work as well as ideas that are not your own is plagiarism. Students who plagiarize will not get any score and they will be introduced in the class.
4. Please create reference for all sources(books, papers, websites) which you use.
5. Please create a cover letter for your report which simply is the Homework#, title of the course, your name, surname, and student number.
6. You may post questions asking for clarifications and alternate perspectives on concepts on piazza or in the class.
7. Submit your final file of assignment to Elearn website with name [ML hw# Surname] which # indicates number of the homework.

Part 2

Questions

1 Regression Model

In this question, you will implement linear regression, and polynomial regression, and learn their parameters using Stochastic Gradient Descent algorithm(SGD) and direct matrix formula. You'll use the Weather in Szeged 2006-2016, with $n = 96453$ samples and $d = 6$ features, where the dataset is given on Elearn website. The features are composed of Apparent Temperature (C), Humidity, Wind, Speed (km/h), Wind Bearing (degrees), Visibility (km), Pressure (millibars). The target is Temperature (C). You are asked to train some models to predict the Temperature (C) based on these features. Also, you should add a constant 1 column to the features matrix for the bias term after standardization of the data(train and test data as well) . On all of parts of this exercise, use 80 percent of the data for training and 20 percent for test of the model and return your results for 10 times.

1. Do some exploratory data analysis to give the readers a viewpoint of the dataset, such as scatter plot and a descriptive display for each feature. Also, it is recommended to standardize the data and then add 1's column to the features matrix.

2. (10 points) Implement a direct approach (normal equations) to obtain the linear regression solution and also save your required running time.

3. (20 points) Implement a mini-batch stochastic gradient descent approach to obtain the linear regression solution (see Algorithm 1). Use batch size from these values $b = \{10, 20, 30, 40, 50, 100\}$. Here, each step corresponds to updating with a randomly chosen mini-batch of data. This entails iterating one or more times over the dataset in order (assuming it is random, with i.i.d. samples). Each iteration over the dataset is called an epoch.

   The update has the form $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha_t \mathbf{g}_t$, where $\mathbf{g}_t$ is the mini-batch gradient on iteration t. On the stepsize choice, we use the inverse of an accumulating sum of gradient norms: $\alpha_t = (1 + \bar{g}_t)^{-1}$ where

   $$\bar{g}_t = \bar{g}_{t-1} + \frac{1}{d+1} \sum_{j=0}^{d} |g_{t,j}|$$

   with $g_{t,j}$ the j-th entry in the vector (the array) $\mathbf{g}_t$ and $\bar{g}_0 = 1$ where t starts at 1. Initialize the weights to zero and set the default number of epochs to 100. You should report the error

Algorithm 1 Stochastic Gradient Descent for objective function $J(\mathbf{w}) = \frac{1}{n} J_n(\mathbf{w})$

Fix iteration parameters: number of epochs $= 10^4$ and mini-batch size $b = 10$
$\mathbf{w} \leftarrow$ random vector in $\mathbb{R}^d$
for $p = 1, \ldots,$ number of epochs **do**
    Shuffle data points from $1, \ldots, n$
    for $k = 1, \ldots, n$ **do**
        $\mathbf{g} \leftarrow \nabla J_k(\mathbf{w})$ # For linear regression $\nabla J_k(\mathbf{w}) = (\mathbf{x}_k^\top \mathbf{w} - y_k)\mathbf{x}_k$
        # For convergence, the step-size α_t needs to decrease with time, such as $\alpha_t = \frac{1}{p}$
        $\alpha \leftarrow \frac{1}{p}$
        $\mathbf{w} \leftarrow \mathbf{w} - \alpha \mathbf{g}$
return $\mathbf{w}$

obtained by your linear regression model for each batch size. Which of them is attained the better result on test data?

4. (20 points) Implement polynomial regression with a $p = 2$ degree polynomial, using the same approaches as in parts 2 and 3. This means using a mini-batch stochastic gradient descent approach with the given step size approach. As a hint, consider calling the your previous algorithm you wrote in the first part, within in the polynomial case, to avoid code duplication and to re-use an already clean algorithm. Then, report the error obtained by your polynomial regression model.

5. (Optional, 20 points) Implement a general M-th order polynomial regression model which takes M the order of the model. Can you provide the best polynomial regression model for this dataset?

6. (10 points) Use AdaGrad algorithm on the stepsize approach on the linear regression model, see the details on GradDescent Ruder. Report the error obtained by your linear regression model trained with AdaGrad.

7. (optional, 10 points) Use Momentum algorithm GradDescent Ruder on the SGD with the same number of epochs the linear regression model. Report the error and the speed of convergence by using this method in the regression cost function.

8. (10 points) You can find some packages in python such as statmodel and sklearn to do regression modeling(see chapter 4 of Geron's Book). Compare your results on linear regression model (part i) with some written tools and packages in python. Finally, tell us about your pros and cons of your written functions versus the built-in ones(running time and accuracy of results) on part 4.