華南理工大學

South China University of Technology

The Experiment Report of *Deep Learning*

School: School of Software Engineering

Subject: Software Engineering

Author:
Qichen Huang

Supervisor:
Mingkui Tan

Student ID:
201920142806

Grade:
Graduate

November 9, 2019

Linear Regression and Stochastic Gradient Descent

***Abstract*—Linear Regression is to find a linear function that can predict a continuous value properly. We carry out this experiment to reveal the theory and implementation details of linear regression. The experiment is conducted via Close-Form Solution and Stochastic Gradient descent. Both of them produce promising results.**

I. Introduction

REGRESSION is one of the most common problem in Machine Learning, which is to make up a function mapping from specific features to a continuous value, named label. Its linear situation brings out a certain version, called Linear Regression. In this experiment, we attempt to figure out the theory of Linear Regression and reveal details of its implementation. Our motivation is to 1) understand Linear Regression and its implementation, both close-form solution and stochastic gradient descent solution, 2) conduct some experiments under small scale dataset, 3) experience the process of optimization, like adjusting parameters. We conduct experiments on Housing Data in LIBSVM, with close-form and stochastic gradient descent. Both of them give out promising result.

II. Methods and Theory

The target of Linear Regression is to find a linear function $\hat{y_i} = \boldsymbol{x}_i^T \boldsymbol{w} + b$ so that for every single feature data $\boldsymbol{x}_i$, its predicted value $\hat{y_i}$ gets as close to the ground truth value y_i as possible, in which $\boldsymbol{x}_i$ and $\boldsymbol{w}$ are both column vectors. Thus the only effort is to search proper value for parameter $\boldsymbol{w}$ and b. To conveniently solve this problem, we define a loss function $\mathcal{L}_i = \frac{1}{2}(y_i - \hat{y_i})^2$ for i th feature data to calculate the difference between y_i and $\hat{y_i}$. As a result, the total loss function is $\mathcal{L} = \frac{1}{2}\sum_i (y_i - \hat{y_i})^2$. With matrix representation, it can be written as $\mathcal{L} = \frac{1}{2}(Y - \hat{Y})^T(Y - \hat{Y})$ where Y and $\hat{Y}$ are batch ground truth and batch predicted value separately, and both of them are column vectors. After that, the parameter searching problem is converted to minimization problem of loss function. In this experiment, the minimization problem is solved with close-form solution and stochastic gradient descent.

A. Close-Form Solution

As the loss function of Linear Regression is convex, the minimum of function can be reached by setting its derivative to zero. For convenience of calculation, we absorb parameter b into $\boldsymbol{w}$ and append 1 to the end of every feature data $\boldsymbol{x}_i$. Then the batch predicted value $\hat{Y}$ can be computed as $\hat{Y} = X\boldsymbol{w}$, where $X = \{\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, ..., \boldsymbol{x}_n^T\}^T$ is batch feature data. After setting derivative of loss function $-X^TY + X^TX\boldsymbol{w}$ to zero, we can get $w = (X^TX)^{-1}X^TY$.

B. Stochastic Gradient Descent

Gradient, vector of partial derivative, points in the direction of greatest increase of a function. In other words, negative gradient points to the greatest decrease of function. That is to say, if we update parameters of loss function towards negative gradient direction, the loss function will get to minimum. This is the idea behind Gradient Descent. However, both Close-Form Solution and Gradient Descent face the problem in which computation involving the whole batch of data might use out of memory and fails to output final result. As an alternative Stochastic Gradient Descent considers updating parameters with gradient computed from only single data. For data $(\boldsymbol{x}_i, y_i)$, we can compute gradient as $\mathcal{G}_i = (\boldsymbol{w}^T\boldsymbol{x}_i - y_i)\boldsymbol{x}_i$. Then $\boldsymbol{w}$ is updated as $\boldsymbol{w} = \boldsymbol{w} - \eta\mathcal{G}$, where η is the learning rate.

III. Experiments

A. Dataset

The experiments use scaled Housing Data in LIBSVM which includes 506 samples and each sample has 13 features. These samples are further divided into training set, $\frac{2}{3}$ of total, and validation set, $\frac{1}{3}$ of total.

B. Implementation

1) Experiment Step: For Close-Form Solution, the experiment steps are as follows:

1) Load the experiment data.
2) Divide dataset into training set and validation set.
3) Select a loss function
4) Get the formula of the Close-Form Solution.
5) Get the value of parameter *W* by the Closed-Form Solution, and update the parameter *W*.
6) Get the *loss*, *loss_train* under the training set and *loss_val* by validating under validation set.
7) Output the value *loss*, *loss_train* and *loss_val*.

For Stochastic Gradient Descent, the experiment steps are as follows:

1) Load the experiment data.
2) Divide dataset into training set and validation set.
3) Initialize linear model parameters.
4) Choose loss function and derivative.
5) Calculate gradient *G* toward loss function from each sample.
6) Denote the opposite direction of gradient *G* as *D*.
7) Update model:$W_t = W_{t-1} + \eta D$. η is learning rate, a hyper-parameter that we can adjust
8) Get the *loss_train* under the training set and *loss_val* by validating under validation set.
9) Repeat step 5 to 8 for several times, and output the value of *loss_train* as well as *loss_val*.

2) Initialization: In Close-Form Solution, no parameters need to be initialized. While in Stochastic Gradient Descent, $\boldsymbol{w}$ is initialized randomly. Learning rate and training step are initialized to 0.01 and 10000 separately.

3) result: The final result of experiments are presented in Table I and training process of Stochastic Gradient Descent is depicted in Fig. 1.

TABLE I: Experiment Result

Loss	training loss	validation loss
Close-Form Solution	11.492508	10.362012
Stochastic Gradient Descent	11.786695	10.534759

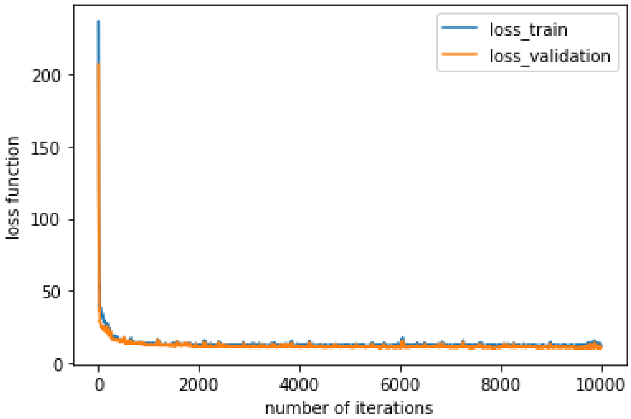

Fig. 1: Training Process of SGD

IV. Conclusion

In this experiment, we explored linear regression problem with Close-Form Solution and Stochastic Gradient Descent. Because the loss function of linear function is convex, Close-Form Solution reaches the minimum, which is better than Stochastic Gradient Descent. Although Stochastic Gradient Descent can perfectly handle out-of-memory problem, it also results in fluctuation during training process, slowing down the optimization.