华南理工大学

South China University of Technology

The Experiment Report of *Machine Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Biquan Wang

Supervisor:
Qingyao Wu

Student ID:
201821038853

Grade:
Graduate

December 31, 2018

Linear Regression and Stochastic Gradient Descent

***Abstract*—Linear regression is one of the simplest and most important model of machine learning. This report try two show the process of closed-form solution and gradient descent of linear regression. Futher more, we try to prove the validity of gradient descent and show the loss with different learning rate.**

I. Introduction

LINEAR regression is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables. It is one of the simplest and important model of machine learning. Stochastic gradient descent is one of the most popular algorithms of optimization and was often used as black-box optimizer. This method is widely used in neural networks.

This report aims to show the process of closed-form solution and stochastic gradient descent of linear regression. It also contains the result of closed-form solution, the compare between the loss of closed-fromr solution, full-batch gradient descent, and mini-batch stochastic gradient descent, which can prove the validity of MSGD and FGD. Futher more, we provided the result of FGD and MSGD.

II. Methods and Theory

In this section, we will give a complete introduction to the experiment, which contains the theory and function of closed-form solution, the losss function and stohcastic gradient descent of linear regression.

The regularized least square regression:

$$\boldsymbol{L}_D(\boldsymbol{w}) = \frac{\lambda}{2}||\boldsymbol{w}||_2^2 + \frac{1}{2}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x_i};\boldsymbol{w}))^2$$

$$\boldsymbol{L}_D(\boldsymbol{w}) = \frac{\lambda}{2}||\boldsymbol{w}||_2^2 + \frac{1}{2}||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||_2^2 \tag{1}$$

Here, $\frac{1}{2}||\boldsymbol{w}||_2^2$ is called Regularizer, λ is called trade-off parameter or regularization parameter. Find minimizer of least squared loss:

$$\boldsymbol{w}^* = arg\,min_{\boldsymbol{w}}\boldsymbol{L}_D(\boldsymbol{W})$$

First-order condition of the optimal solution:

$$\frac{\partial \boldsymbol{L}(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0$$

For the least regression problem, we have:

$$\frac{\partial \boldsymbol{L}(\boldsymbol{w})}{\partial \boldsymbol{w}} = \lambda\boldsymbol{w} - \boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} \tag{2}$$

$$\lambda\boldsymbol{w} - \boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = 0$$

$$(\lambda\boldsymbol{I} + \boldsymbol{X}^T\boldsymbol{X})\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{y}$$

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-}1\boldsymbol{X}^T\boldsymbol{y}$$

We get the **closed-form solution** of linear regression:

$$\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-}1\boldsymbol{X}^T\boldsymbol{y} \tag{3}$$

Gradient descent:

- Identify a set of hupotheses $f(\boldsymbol{x};\boldsymbol{w})$
- Define a loss criterion $\boldsymbol{L}_D$
- Pick the best $\boldsymbol{w}^*$ by minimizing a loss function $\boldsymbol{L}_D(\boldsymbol{w})$

$$arg\,min_{\boldsymbol{w}}\boldsymbol{L}_D(\boldsymbol{W})$$

By Taylor expansion, when $\eta \to 0$:

$$\boldsymbol{L}_D(\boldsymbol{w} + \eta\boldsymbol{d}) = \boldsymbol{L}_D(\boldsymbol{w}) + [\frac{\partial \boldsymbol{L}_D(\boldsymbol{w})}{\partial \boldsymbol{w}}]^T\eta\boldsymbol{d} + o(\eta\boldsymbol{d})$$

$$\boldsymbol{L}_D(\boldsymbol{w} + \eta\boldsymbol{d}) = \boldsymbol{L}_D(\boldsymbol{w}) + \eta[\frac{\partial \boldsymbol{L}_D(\boldsymbol{w})}{\partial \boldsymbol{w}}]^T\boldsymbol{d}$$

We have:

$$\boldsymbol{L}_D(\boldsymbol{w'}) = \boldsymbol{L}_D(\boldsymbol{w} + \eta\boldsymbol{d}) \leq \boldsymbol{L}_D(\boldsymbol{w}) \quad (\eta \neq 0\,\&\,\eta > 0) \tag{4}$$

We use $d = -\frac{\partial \boldsymbol{L}_D(\boldsymbol{w})}{\partial \boldsymbol{w}}$ as the direction of optimization. Update parameters with rate η:

$$\boldsymbol{w'} \to \boldsymbol{w} - \eta\frac{\partial \boldsymbol{L}_D(\boldsymbol{w})}{\partial \boldsymbol{w}} \tag{5}$$

In this experiment, we used **Mini-batch Stochastic Gradient Descent**, which means pick a part of data from train set randomly, and use this part of data to calculate the gradient and update the parameters.

III. Experiments

This part contains two experiments, the first one is closed-form solution of linear regression, and the second part is linear regression with gradient descent.

A. Dataset

Experiment uses the scaled edition of Housing in LIBSVM Data, including 506 samples and each sample has 13 features.

B. Experiment steps

The steps of closed-form solution are as the following:

1. Use load_svmlight_file function in sklearn library to load the Housing scaled data.
2. Divide dataset into training set and validation set using train_test_split function.In these experiments, we set the validation size as 0.25.
3. Use closed-form solution (function (3)) to calculate the parameters $\boldsymbol{w}$ directly.
4. Use loss function (fucntion (1)) to calculate the loss on train set and validation set.

The steps of gradient descent solution are as the following:

1. Use load_svmlight_file function in sklearn library to load the Housing scaled data.
2. Divide dataset into training set and validation set using train_test_split function.In these experiments, we set the validation size as 0.25.
3. Initialize linear model parameters. Set all parameter into zero, initialize it randomly or with normal distribution.
4. Get a mini-batch data from train set randomly.
5. Update the parameters $\boldsymbol{w}$ with function (5)
6. Calculate the loss of train set and validation set with function (1).
7. Repeate setp 4-6 for n times, and return the losses of train set and validation set.

C. Experiment result

TABLE I
CLOSED-FORM SOLUTION RESULT

Type	Loss of train set	Loss of valid set
Closed-form	3.648082165263596	3.5673788625276366
FGD epoch200	3.5083401597660315	4.233630428514219
MSGD epoch200	3.777113540188303	4.439968690213303
FGD epoch2000	3.449777991591353	3.651855651969987
MSGD epoch2000	3.454142930228637	3.653706861808154

Table.1 shows the result of closed-form solution, and compare with the loss of FGD and the loss of MSGD . We can find out from the table, with integration number increase, the difference of the loss of valid set and train set decreased, and both of them approch to closed-form loss.

The next part show the result of the experiment we have done for linear regression with gradient descent. All the figures, use plenty=0.5, epoch=200,batch size=200.

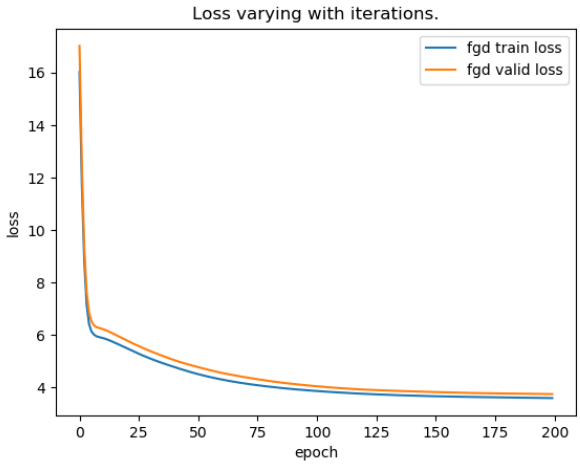

Fig. 1. Full batch gradient descent of linear regression

Fig.1 shows the result of Full batch gradient descent with learning rate 0.001.

Fig.2 compares different result of different learning rate of full match gradient descent. We can find out the loss descent more quick with the learn rate get bigger.

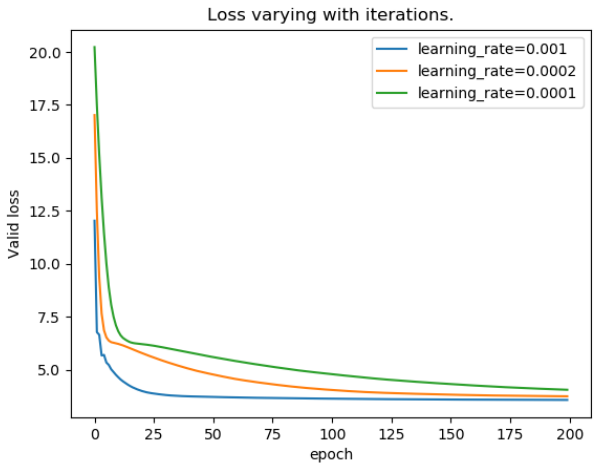

Fig. 2. Full batch gradient descent of linear regression

Fig. 3. Mini-batch stochastic gradient descent of linear regression

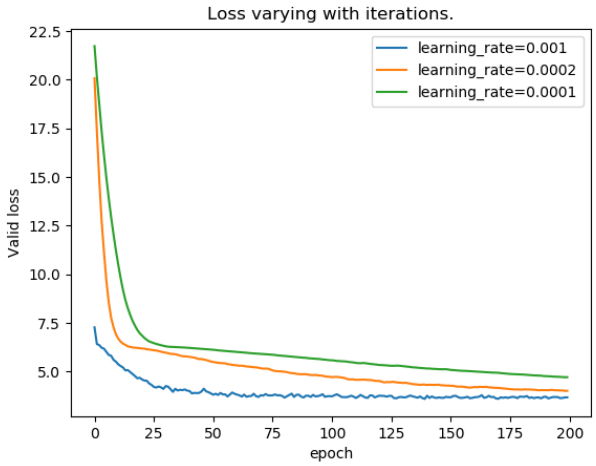

Fig. 4. Different learning rate MSGD of linear regression

Fig.3 shows the valid loss and train loss of Mini-batch stochastic gradient descent with learning rate 0.0002.

Fig4. compares the result of Mini-batch stochastic gradient descent with different learning rate. When learn_rate=0.001, the loss descent quickly and starting to shake after 25 epoch.

IV. Conclusion

In this report, we mentioned two method of linear regression, Closed-form solution and Gradient descent. Closed-form solution can get the parameter $\boldsymbol{w}$ directly, but with the data set increase, this this method can not be relize. Mini-batch stochastic gradient descent is a solution of this situation, Table.1 shows this method can approch the result with the number of iterations increases.