The Experiment Report of Machine Learning

SCHOOL: Automation Science and Engineering

SUBJECT: automatization

Author:

Zhirui Zhang

Supervisor:

Mingkui Tan

Student ID:

202130462158

Grade:

Undergraduate

2024-3-29

Linear Regression, Linear Classification and Gradient Descent

***Abstract*—This experiment explores the fundamental concepts of linear regression and gradient descent in machine learning, including closed-form solution, gradient descent, and the applications of linear regression. By conducting experiments on small-scale datasets, the experiment offers practical insights into the application of these methods and the iterative process of optimization and parameter tuning. Participants will become familiar with and apply the fundamental algorithms, consolidating their foundational knowledge.**

I. INTRODUCTION

Linear regression serves as a cornerstone in the field of machine learning, providing a simple yet powerful framework for modeling relationships vetween variables. In this experiment, we delve into the intricacies of linear regression and focus on two primary optimization methods: closed-form solution and gradient descent. The closed-form solution, also known as ordinary least squares, allows us to obtain the analytical solution directly from a mathematical perspective. On the other hand, gradient descent offers a more iterative and computationally efficient approach, particularly suitable for large-scale datasets. By conducting experiments on small-scale datasets and continuously adjusting various parameters to achieve optimal convergence performance, we aim to find better methods and parameter combinations. After completing the experiment, participants will have a deeper understanding of linear regression and its optimization, laying the foundation for further exploration of machine learning algorithms and techniques. Please translate the above content in accordance with the standards of a research paper.

II. METHODS AND THEORY

1. Closed-Form Solution

The closed-form solution, also known as Ordinary Least Squares (OLS), is an analytical method used to find the optimal parameters of a linear regression model. It determines the best parameter values by minimizing the sum of squared residuals of the objective function.

For linear regression problems, the commonly used loss function is the Mean Squared Error (MSE).

$$l(\hat{y}_i, y_i) = \frac{1}{2}(\hat{y}_i - y_i)^2$$

Thus the total loss function can be obtained:

$$L_D(w) = \frac{1}{2} \parallel Y - Xw \parallel_2^2$$

In order to minimize the total loss function, we need to derive the total loss function and make the derivative equal to zero.

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} = \lambda \mathbf{w} - \mathbf{X}^{\mathrm{T}}\mathbf{y} + \mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} = 0$$

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

According to the closed-form solution, you can directly substitute the experimental data and find the coefficients.

2. Gradient Descent

The gradient descent algorithm is an iterative optimization method used to find the optimal parameters that minimize the loss function. It optimizes the model by continuously updating the parameters through iterations. The specific steps are as follows:

1) Initialize parameters: Randomly initialize the model parameters w.
2) Compute gradients: Calculate the gradients of the loss function with respect to the model parameters.
3) Update parameters: Update the parameters by taking a step in the opposite direction of the gradients multiplied by a learning rate.
4) Repeat iterations: Repeat steps 2 and 3 until reaching the designated stopping criteria (e.g., maximum number of iterations or convergence of the loss function).

Following the loss function of the closed-form solution of linear regression, we can let the parameters gradually approach the optimal value over many iterations.

$$\mathbf{w}' = \mathbf{w} - \eta \frac{\partial \mathcal{L}_{\mathcal{D}}(\mathbf{w})}{\partial \mathbf{w}}$$

3. Stochastic Gradient Descent(SGD)

Stochastic Gradient Descent (SGD) for linear regression is a variant of the gradient descent algorithm primarily utilized for handling large-scale datasets or a large number of training samples. In contrast to traditional gradient descent algorithms, SGD updates parameters by randomly selecting samples, thereby reducing computational complexity and enhancing efficiency.

III. EXPERIMENT

1. Experimental data

Linear regression used the Housing data from LIBSVM Data, containing 506 samples with 13 attributes each. We split the data into two groups, one for training the model and the other for validating the model.

2. Closed-Form Solution

In the closed-form solution of linear regression we find the analytic solution by means of the mean square error loss function and also measure the loss value of the analytic solution by the mean absolute error loss function and the huber loss function.

TABLE I
DIFFERENT LOS VALUES

	train	valid
Mean squared	19.6268246	31.5948237
Mean absolute	3.08131773	3.6029813
Huber loss	2.6278197	3.12617829

3. Gradient Descent

In the linear regression gradient descent experiment, we explored the effects of the experimental parameters on the experimental results by modifying different experimental parameters.

3.1 Loss Function Comparison

We trained the model using the mean square error loss function and the mean absolute error loss function, respectively. Where the final loss values obtained when using the mean square error loss function are 22.45156393456746 for training loss and 20.956072609533962 for validation loss; and 3.268520114705345 for training loss and 3.1190712385307338 for validation loss when using the mean absolute error loss function. according to the their loss conversion relationship relationship, the mean square error loss function yields a lower equivalent loss.

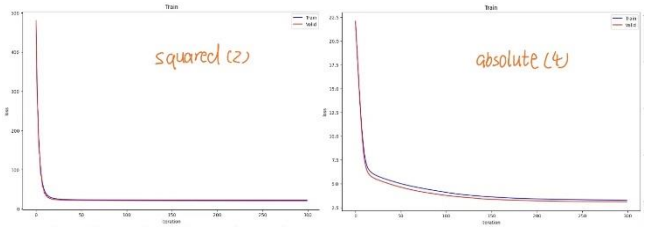


Figure. 1. Mean square error loss and absolute mean loss

3.2 Stochastic gradient and plenary gradient

Theoretically, stochastic gradient descent can reduce computational complexity and improve computational efficiency in the face of models with large amounts of data. However, in this experiment, the data volume is small, and stochastic gradient descent does not play an advantage, but performs less well than full gradient descent.

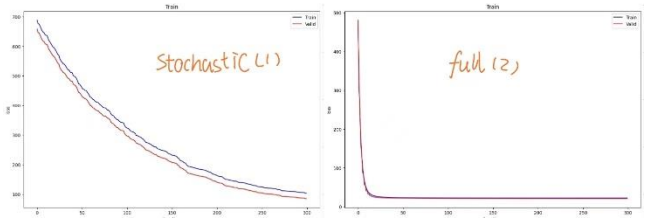


Figure. 2. Stochastic gradient and full gradient

3.3 Adjustment of the number of iterations

The number of training iterations is also an important parameter; the higher the number of iterations, the better the model fit is theoretically. This experiment compares the results of 300 iterations and 500 iterations and verifies that this is indeed the case.

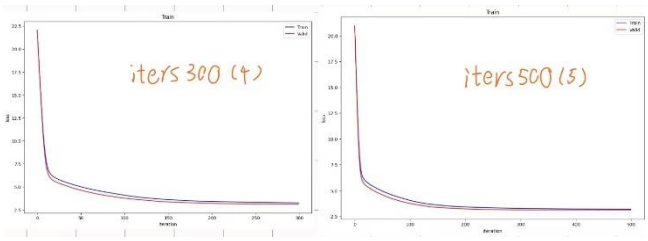
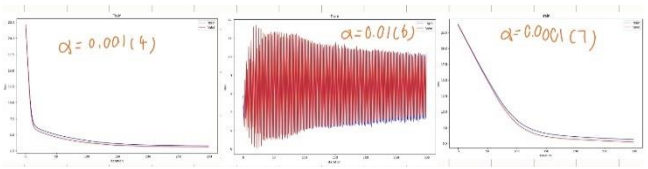


Figure. 3. 300 iterations and 500 iterations

3.4 Comparison of learning rates

We also compared different learning rates. The experimental results illustrate that when the learning rate is too low, the model converges slowly, and when the learning rate is too high, the model tends to cross the optimal point and oscillate the loss values. Therefore, a moderate learning rate is optimal.

Figure. 4. Comparison of learning rates



3.5 Comparison of parameter initialization methods

Commonly used modes of parameter initialization are random initialization, all-zero initialization, and normal distribution initialization. We tested these three models and their final losses on the validation set were 3.1190712385307338, 3.1402969125354345, and 3.1609001107958887 respectively. they do not differ much in small models.

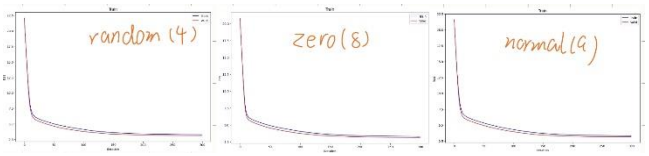


Figure. 5. parameter initialization methods

3.6 Machine learning merit prediction

We apply the defined model to predict students' grade points. Firstly, we import the data and handle any occurrences of NaN values. Then, we separate the labels from the input data for model training. There are two approaches to handling NaN values: the first is to fill the NaN positions with zero data, and the second is to directly ignore data with NaN values. Experimental results indicate that disregarding data with NaN values during training often leads to lower loss as the data with missing values may not accurately represent the true scenario.

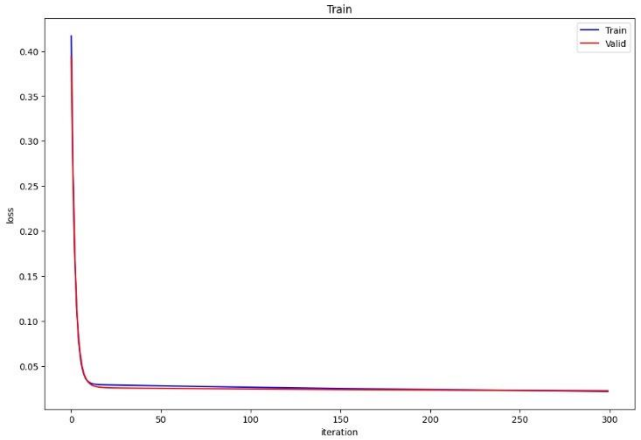


Figure. 6. make up zeros

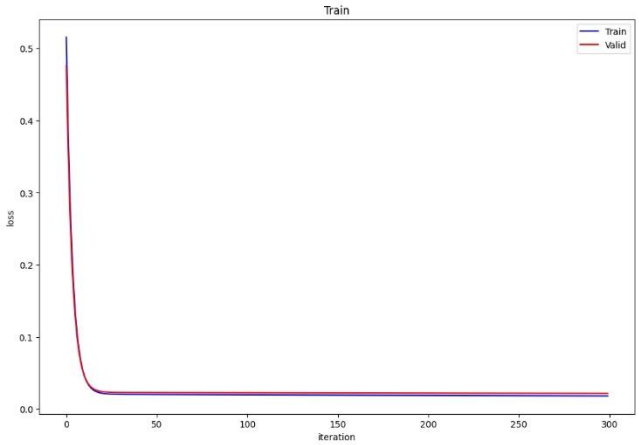


Figure. 6. ignore nan

TABLE II

ZERO OR IGNORE

	train	valid
zeros	0.0216113415	0.022361532
ignore	0.018280207	0.0218754779

IV. CONCLUSION

In the linear regression closed-form solution and gradient descent experiments can initially understand the impact of various parameters on model training, which helps to achieve better training results by adjusting the training parameters in future experiments. We also note that the most accurate solution on the training set is not necessarily good, which may make the model lack generalization ability and underperform on new data.