The Experiment Report of Machine Learning

SCHOOL: SHIEN-MING WU SCHOOL OF INTELLIGENT ENGINEERING

SUBJECT: The super robot Everest class

Author:

Xilang Zeng

Supervisor:

Mingkui Tan

Student ID:

202130461984

Grade:

Undergraduate

2024-3-28

Experiment 1: Linear Regression and Stochastic Gradient Descent

***Abstract*—This experiment involves using linear regression with the Housing dataset from LIBSVM Data. The parameters of the linear regression are optimized using both the closed-form solution and the stochastic gradient descent method for training. Then, the trained linear regression model's prediction accuracy is validated using a validation set. Additionally, an application experiment was conducted using a student grades dataset, where the grades from all courses were used to predict the GPA in a machine learning course.**

I. INTRODUCTION

To predict the Housing data from LIBSVM Data, this experiment utilizes a linear regression model, experimenting with and comparing two methods: the closed-form solution and stochastic gradient descent. The effectiveness of the two methods and the impact of different parameters are compared.

Since linear regression has a closed-form solution, it's clear that the stochastic gradient descent method does not hold an advantage in terms of computational speed.

To test the predictive capabilities of linear regression, an experiment was also conducted using a student grades dataset. This experiment aimed at predicting the GPA in a machine learning course based on the grades from various courses. Due to the lack of nonlinearity in linear regression, there are substantial differences in its predictive performance across different datasets.

II. METHODS AND THEORY

A. Reading the experimental data

The Housing dataset stored in LIBSVM format is loaded using the Python library sklearn, and it is divided into training and validation sets through the sklearn library. The validation set accounts for 10% of the original dataset, and the data is randomized before the split. After reading the data, preprocessing is performed to convert the sparse matrix into a dense array format, and the target vectors y_train and y_valid are transformed from one-dimensional arrays into two-dimensional arrays.

B. Splitting into training and validation sets

The training set is the dataset used to train the model. The model learns and establishes rules for prediction or classification from this part of the data. The training set constitutes a major portion of the entire dataset. The validation set helps us understand the model's performance on unknown data and is used to avoid overfitting of the model. It usually constitutes a smaller proportion of the entire dataset. In this experiment, we divided the dataset into 90% for the training set and 10% for the validation set. Before the division, we also randomized the order of the dataset's data.

C. Choosing the loss function

In machine learning, common loss functions include the zero-one loss function, absolute loss function, and mean squared error (MSE) loss function, among others. The mean squared error loss function is widely used in machine learning and statistics, particularly suitable for regression problems. It evaluates a model's performance by calculating the average of the squares of the differences between the model's predictions and the actual values. The MSE loss function not only quantifies the accuracy of the model's predictions but also imposes higher penalties for larger errors, helping the model to fit the data more precisely. Therefore, the mean squared error loss function is adopted in experiments involving linear regression.

the mathematical expression for the MSE loss function is as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

Where n is the number of samples, y_i is the actual value of the i^{th} sample. $\hat{y_i}$ is the predicted value for the i^{th} sample, MSE represents the mean of the squared differences between actual and predicted values.

For linear regression, the formula for the mean squared error (MSE) loss function is

$$\mathcal{L}(w) = \frac{1}{2}\sum_{i=1}^{n}(y_i - w^T x_i)^2$$

D. Computing model parameters

Since linear regression has a closed-form solution, we can calculate the parameters of linear regression through two methods: closed-form solution or gradient descent method. The closed-form solution for linear regression is

$$w^* = (X^T X)^{-1} X^T y$$

The closed-form solution allows for the direct computation of the optimal parameters for the linear regression model.

Gradient descent is another method for calculating optimal parameters. The negative direction of the gradient is the direction in which the function decreases most rapidly at that point. Since linear regression aims to minimize the loss function, we need to obtain the gradient of the loss function at each point. Then, we update the parameters in the negative direction of the gradient, thereby optimizing the loss function. The gradient of the loss function for linear regression is

$$X^T(Xw - y)$$

Therefore, the update rule for the parameters is

$$w \coloneqq w - \eta X^T(Xw - y)$$

Stochastic Gradient Descent (SGD) is a method used for optimization, with the main goal of minimizing or maximizing an objective function. Unlike traditional gradient descent algorithms that use all samples to calculate the gradient for parameter updates, SGD selects only a random sample to calculate the gradient and update parameters at each step. This approach significantly reduces computational load, allowing the algorithm to converge more quickly. However, because it updates using only one sample at a time, the convergence process of SGD can be noisier or more oscillatory.

E. Validating with the validation set

For the part about validation on the validation set, we calculate the loss value using the samples and their labels from the validation set, thereby determining the effectiveness of this round of gradient descent. The samples from the validation set are used only for validation and are not used to update the parameters.

F. Printing the loss curve

Use Python's Matplotlib library to plot how the training and validation loss change with the number of iterations. Through the graph, we can visually see how the loss on the training and validation sets changes with the number of iterations, which is very helpful for tuning and evaluating the model.

III. EXPERIMENT

A. Dataset

The LIBSVM Data dataset is a widely used machine learning dataset that contains a variety of data types, such as data for classification problems, regression problems, and other machine learning tasks. This experiment uses the scaled version of the Housing dataset within it, which includes 506 samples, each with 13 attributes. In the experiment predicting student GPA, the dataset used is the student grade dataset, which includes 126 training samples and 54 test samples. Each sample has 28 attributes and 1 label (machine learning course GPA).

B. Implementation

First, we read the Housing dataset from LIBSVM Data and split it into training and test sets. Then, using the closed-form solution of linear regression, we calculate the optimal values of the model parameters and compute the loss function. The model's parameters are initialized through a normal distribution. Through multiple calculations using the closed-form solution of linear regression, the mean of the loss function we obtained is shown in Table 1.

TABLE 1
Loss Value of Closed-Form Solution

Train Loss	25.17
Valid Loss	17.06

Next, we experiment with the method of stochastic gradient descent. The model's parameters are also initialized through a normal distribution. Stochastic gradient descent has two hyperparameters that need to be adjusted (learning rate and number of iterations). We tested the results brought by different learning rates and numbers of iterations, and the loss curve is shown in Figure 1, 2, and 3.

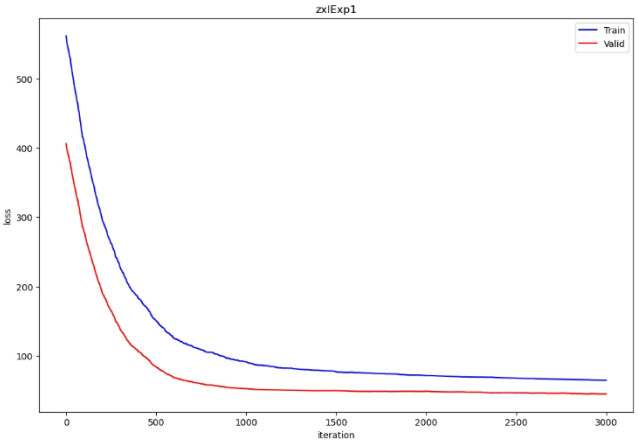

Figure. 1. Loss curve (learning rate 0.0005, iterations 3000).

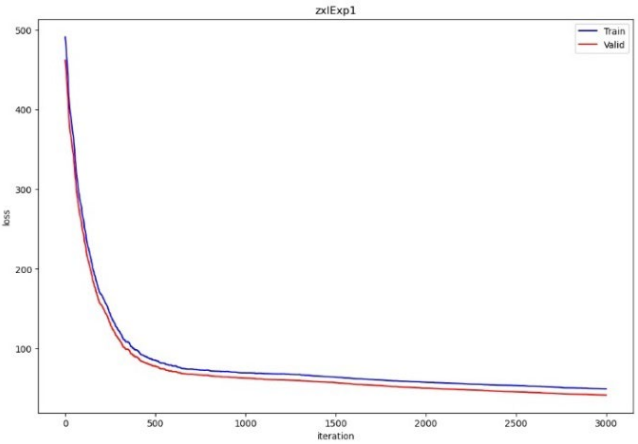

Figure. 2. Loss curve (learning rate 0.001, iterations 3000).

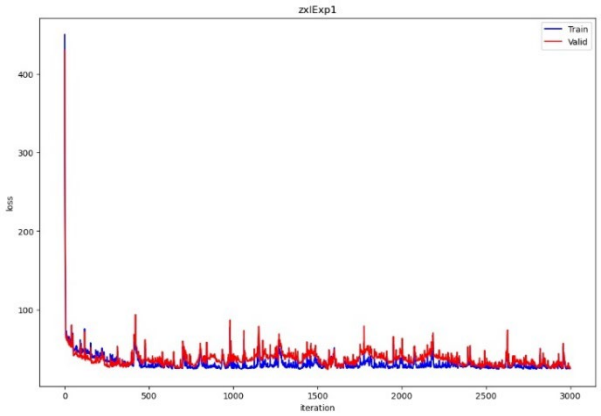

Figure. 3. Loss curve (learning rate 0.05, iterations 3000).

We can see that as the learning rate increases, the loss function initially decreases more rapidly. However, when the learning rate is too high, it causes oscillations in the loss function, making it difficult to converge to the minimum value.

Then, we conduct an experiment on student machine learning GPA prediction. The dataset has been pre-divided into a training set and a test set. We first read the dataset's data, filling in zeros for parts without scores. Then, the training set is put into a stochastic gradient descent function to train the model parameters, and the test set is used to calculate the loss value under the current parameters. Then, print the loss curve, as shown in Figures 4, 5, and 6.

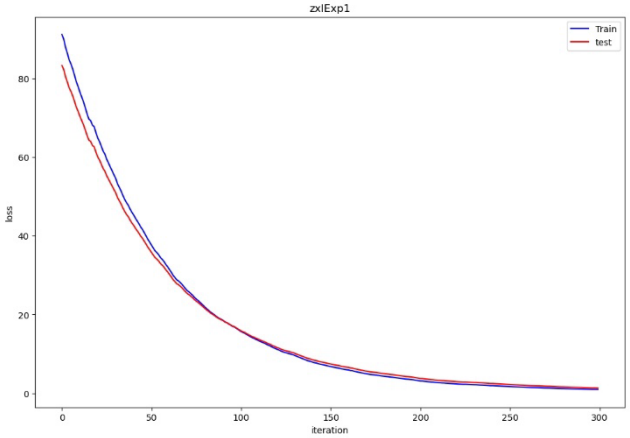

Figure. 4. Loss curve (learning rate 0.0005, iterations 300).

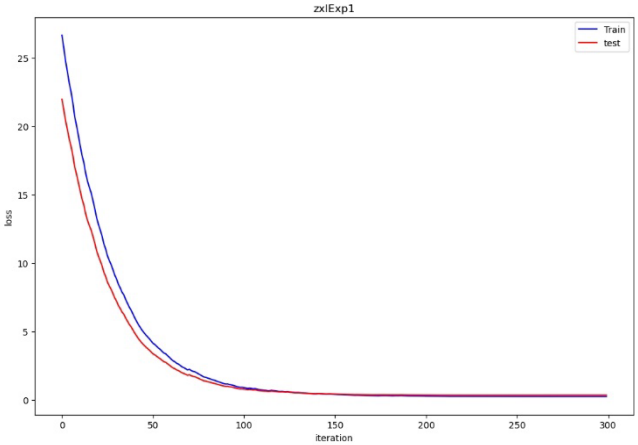

Figure. 5. Loss curve (learning rate 0.001, iterations 300).

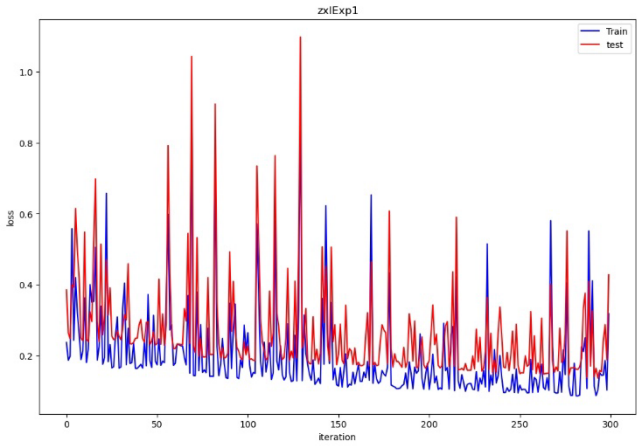

Figure. 6. Loss curve (learning rate 0.05, iterations 300).

It's observed that with an increase in the learning rate, the loss function initially experiences a faster decline. Nevertheless, an excessively high learning rate leads to fluctuations within the loss function, thereby hindering its ability to settle at the minimum value.

Furthermore, we also found that using different initialization methods for the model parameters (such as zero initialization, random initialization, or normal distribution initialization) has a slight impact on the convergence speed.

We can also find that linear regression has different fitting capabilities for different datasets. For the Housing dataset from LIBSVM Data, the loss value will not decrease further around 20~30, whereas for the dataset of student machine learning GPA, the loss value can be reduced to below 1, indicating that the expressive power of linear regression is relatively average.

IV. CONCLUSION

Overall, linear regression can fit different data to solve problems such as prediction or classification. The optimal solution of the linear regression model can be obtained through either a closed-form solution or the method of stochastic gradient descent. With the gradient descent method, different learning rates lead to different learning outcomes; increasing the learning rate can accelerate the convergence rate, but too high a learning rate will cause the loss value to oscillate. The expressive capacity of the linear regression model is limited and cannot adequately complete the prediction tasks in all situations.