华南理工大学

South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCUT

SUBJECT: SOFTWARE ENGINEERING

Author:
Yinqi Zhang

Supervisor:
Mingkui Tan

Student ID：201630666394

Grade:
Grade 2016

October 28, 2018

Linear Regression and Stochastic Gradient Descent

Abstract：

This article explain about linear regression, closed-form solution and Stochastic Gradient Descent, and also implement the two algorithms to solve Linear Regression.

Introduction:

When we have a dataset , we want to use the dataset to train a linear function to predict a right result of other data. We can use closed-form solution or Stochastic Gradient Descent to train a linear function.

Method and theory:

1、*closed-form solution of Linear Regression*:

We use analytic techniques to find a rigorous formula. So arbitrary independent variable can use this formula to find the dependent variable, which is the solution of the problem. In Linear Regression program, we have the loss function as:

$$l_D(w) = \frac{1}{2}\sum_{j=0}^{n}(y_i - w^T x_i)^2$$

We change form of the loss function:

$$L_D(w) = \frac{1}{2}\sum_{j=0}^{n}(y_i - w^T x_i)^2 \quad (1)$$

$$= \frac{1}{2}(y - Xw)^T(y - Xw) \quad (2)$$

$$= \frac{1}{2}(y^T - 2w^T X^T y + w^T X^T X w) \quad (3)$$

Let's take the derivation of the loss function:

$$\frac{\partial L_D(\theta)}{\partial w} = \frac{1}{2}(\frac{\partial y^T y}{\partial w} - \frac{\partial 2w^T X^T y}{\partial w}) + \frac{\partial w^T X^T X w}{\partial w}) \quad (4)$$

$$= \frac{1}{2}(-2X^T y + (X^T X + (X^T X)^T)w) \quad (5)$$

$$= -X^T y + X^T X w \quad (6)$$

We make the derivation of the loss function equal to zero:

$$\frac{\partial L_D(\theta)}{\partial w} = -X^T y + X^T X w = 0 \qquad (7)$$

$$\Rightarrow X^T X w = X^T y \qquad (8)$$

$$\Rightarrow w = (X^T X)^{-1} X^T y \qquad (9)$$

So the optimal parameters w^* is:

$$w^* = (X^T X)^{-1} X^T y = arg \min_w L_D(w)$$

That's all the derivation of SGD.

2、*Linear Regression and Stochastic Gradient Descent*

We want to minimize the parameter w^* , we can use Gradient Descent method to decrease the loss of the function.Each time w^* "take a small step" at the negative direction of gradient. We can decrease the parameter until it's small enough.

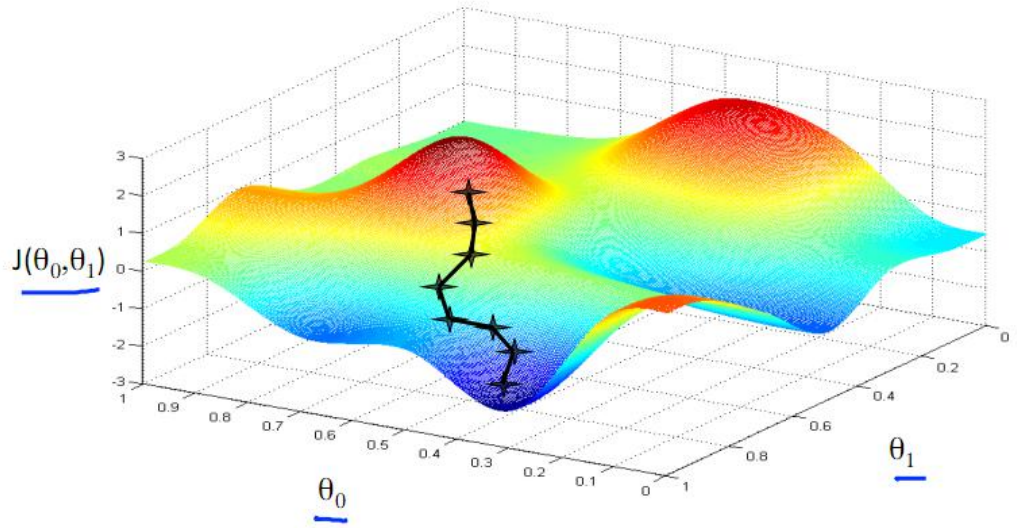


Figure. 1. Each time w^* "take a small step" at the negative direction of gradient

The loss function of the Linear Regression is：

$$L_D(w) = \frac{1}{2}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2$$

To decrease the loss, Each we randomly select a data to decrease w:

$$w_j = w_j + \eta(y^{(i)} - f(x^{(i)}))x_j^{(i)} \qquad (for\ every\ j)$$

That's all the derivation of SGD.

Experiment:

Dataset:

This experiment use the Housing Dataset in LIBSVM Data, including 506 samples and each sample has 13 features.

Implementation:

1、closed-form solution of Linear Regression:

The implementation of the algorithm is:

First we get the dataset from Internet and change the form of the data.

Then we initialize the variable and initialize the parameter using different method:

```
losses_train = []
losses_val = []
w = numpy.zeros((n_features + 1, 1))
# w = numpy.random.random((n_features + 1, 1))
# w = numpy.random.normal(1, 1, size=(n_features + 1, 1))
```

Then update the parameter using the formula, and print it out:

```
a=numpy.dot(X_train.transpose(),X_train)
b=numpy.dot(linalg.pinv(a),X_train.transpose())
w=numpy.dot(b,y_train)
```

Lastly, we calculate the loss of validation set.

2、Linear Regression and Stochastic Gradient Descent

First we get the dataset from Internet and change the form of the data.

Then we divide dataset into training set and validation set.

Then we initialize the variable and initialize the parameter using different method:

```
learning_rate = 0.1
max_epoch = 200
losses_train = []
losses_val = []
# w = numpy.zeros((n_features + 1, 1))    # initialize with zeros
#w = numpy.random.random((n_features + 1, 1))    # initialize with random numbers
w = numpy.random.normal(1, 1, size=(n_features + 1, 1))    # initialize with zero normal distribution
```

Then update the parameter using SGD, and calculate the loss of the training set and validation set:

```
for epoch in range(max_epoch):
    i=random.randint(0,n_train_samples-1)
    diff=numpy.dot(X_train[i],w)-y_train[i]
    w=w-learning_rate*diff*X_train[i].reshape(X_train[i].shape[0],1)

    Y_predict = numpy.dot(X_train, w)
    loss_train = numpy.average(numpy.abs(Y_predict - y_train))
    losses_train.append(loss_train)
    Y_predict = numpy.dot(X_val, w)
    loss_val = numpy.average(numpy.abs(Y_predict - y_val))
    losses_val.append(loss_val)
```

Lastly, print the graph of absolute diff value varing with the number of iterations.

Result：

1、closed-form solution of Linear Regression：

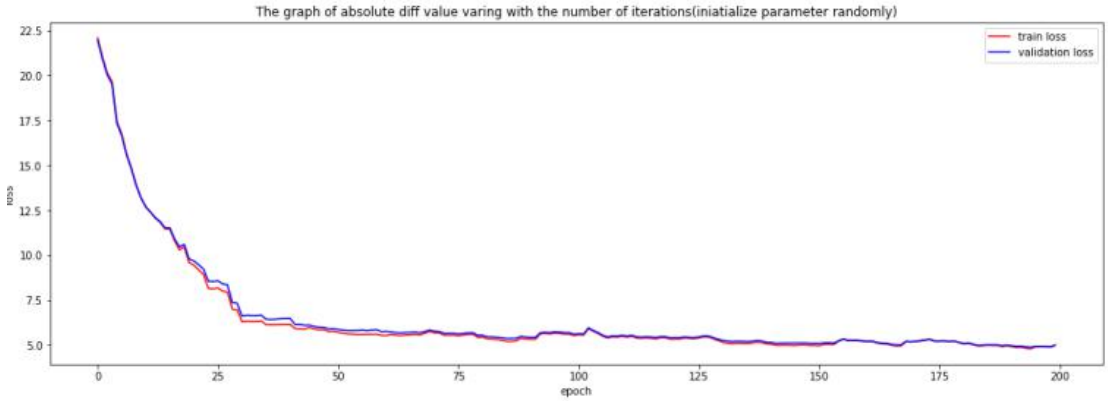
(1) The choice of initialize the paramenter w doesn't have any impact of the result.(set all parameter into zero, initialize it randomly or with normal distribution)

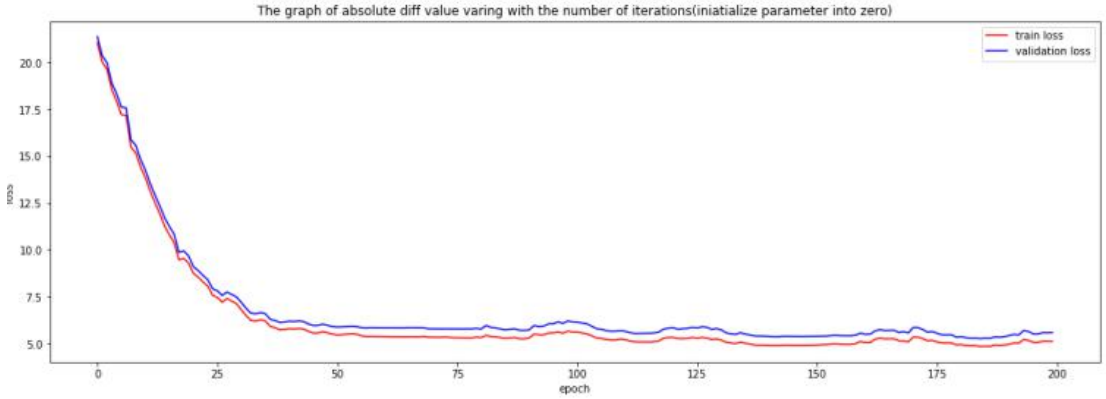
(2) The results in the four time test:

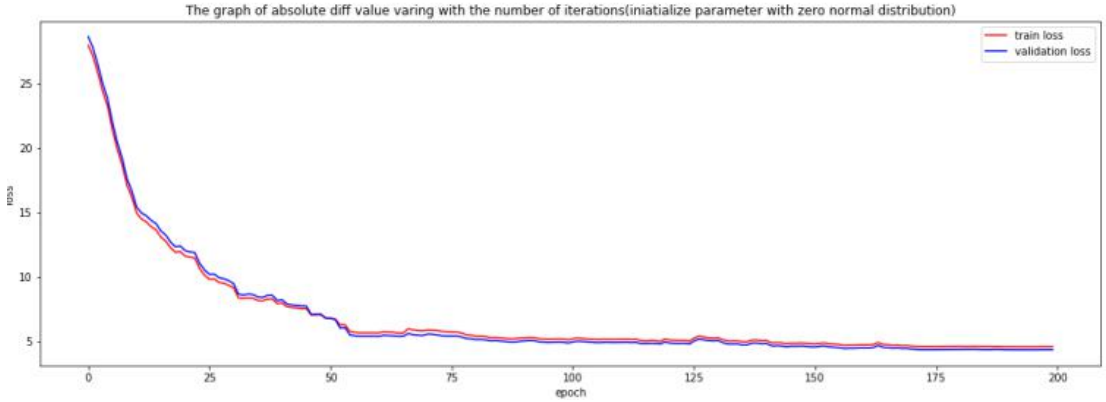
	First_time	Second_time	Third_time	Fifth_time
Loss_train	2.92486031905 37024	3.26143475639 1684	3.44976552866 2935	2.87979043684 3237
Loss_val	3.53847496693 88733	3.44976552866 2935	3.76810180361 72254	3.73697235541 8671

2、Linear Regression and Stochastic Gradient Descent

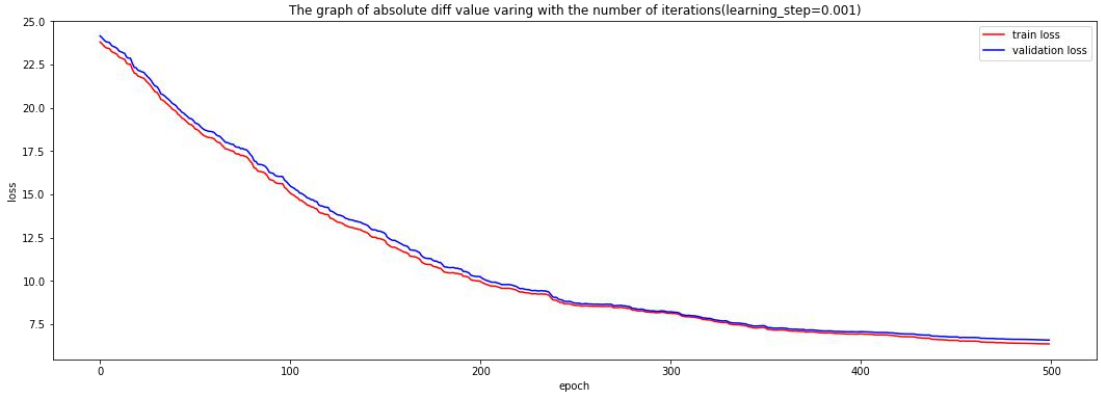
(1)The choice of initialize the paramenter w doesn't have too much impact of the result.(set all parameter into zero, initialize it randomly or with normal distribution)

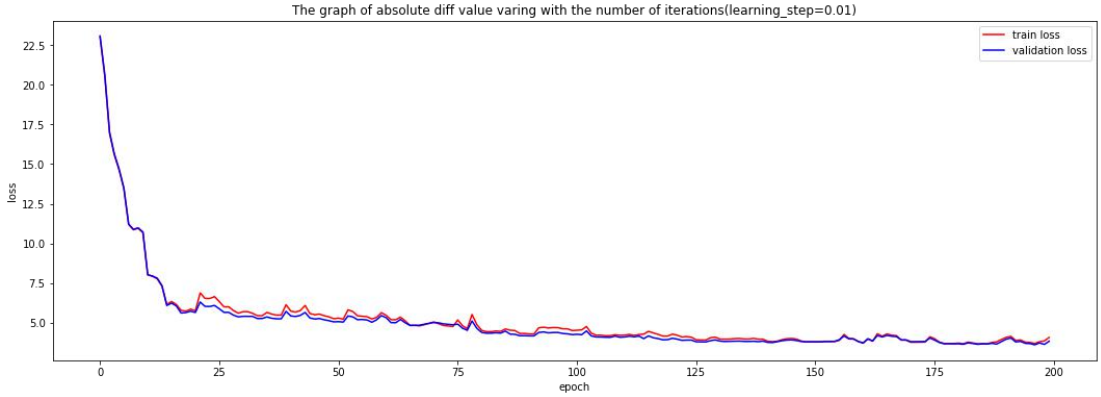


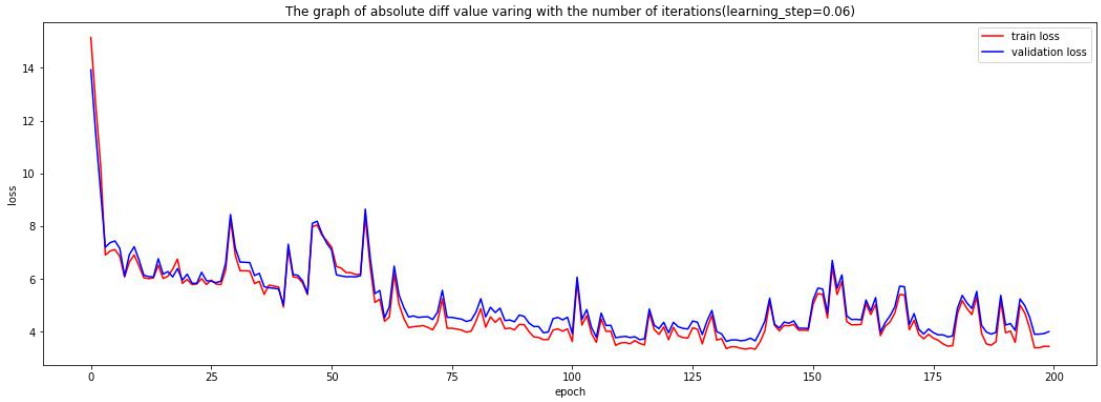




(3) Bigger the learning rate is, the greater the curve fluctuate, the faster the image converges.







Conclusion:

For linear regression problem, there are two methods to solve it——closed-form solution and SGD. For a small dataset, using closed-form solution is a very fast way to get the answer. But for a big dataset, using gradient descent is a better choice.Closed-form solution will only cost more resource.

It’s the first to implement the machine learning algorithm.I had a better understanding about the conception of machine learning, the three steps machine learning concluded.Also, I practice about the linear regression and learning using two methods to solve this kind of problem.And I also experienced the process of change hyperparameters.