

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**   **周宇琛**

**Student ID 201530613997**

**E-mail 1057433699@qq.com**

**Tutor**   **谭明奎**

**Date submitted** **2017. 12 .7**

**1. Topic:** linear regression,linear classification and gradient **descent**

**2. Time:** 2017/12/2-2017/12/7

**3. Reporter:**周宇琛

**4. Purposes:**

(1)、Futher understand the principle of linear regression and gradient descent

(2)、Practice on small data sets

(3)、Understand the process od optimization and reference

**5. Data sets and data analysis:**

**(1)、Linear regression:** the Housing data in LIBSVM Data contains 506 samples, each with 13 attributes. Use the scaled version and divide it into training set and verification set.

**(2)、Linear classification:** the australian data in LIBSVM Data contains 690 samples, each with 14 attributes. Use the scaled version and divide it into training set and verification set.

**6. Experimental steps:**

**Linear Regression and Gradient Descent:**

(1)、Load the experiment data. You can use load\_svmlight\_file function in sklearn library.

(2)、Devide dataset. You should divide dataset into training set and validation set using train\_test\_split function. Test set is not required in this experiment.

(3)、Initialize linear model parameters. You can choose to set all parameter into zero, initialize it randomly or with normal distribution.

(4)、Choose loss function and derivation: Find more detail in PPT.

(5)、Calculate gradient G toward loss function from all samples.

(6)、Denote the opposite direction of gradient G as D.

(7)、Update model: Wt = Wt-1 + ηD. η is learning rate, a hyper-parameter that we can adjust.

(8)、Get the loss Ltrain under the training set and Lvalidation by validating under validation set.

(9)、Repeate step 5 to 8 for several times, and drawing graph of Ltrain as well as Lvalidation with the number of iterations.

**Linear Classification and Gradient Descent:**

(1)、Load the experiment data.

(2)、Divide dataset into training set and validation set.

(3)、Initialize SVM model parameters. You can choose to set all parameter into zero, initialize it randomly or with normal distribution.

(4)、Choose loss function and derivation: Find more detail in PPT.

(5)、Calculate gradient G toward loss function from all samples.

(6)、Denote the opposite direction of gradient G as D.

(7)、Update model: Wt = Wt-1 + ηD . η is learning rate, a hyper-parameter that we can adjust.

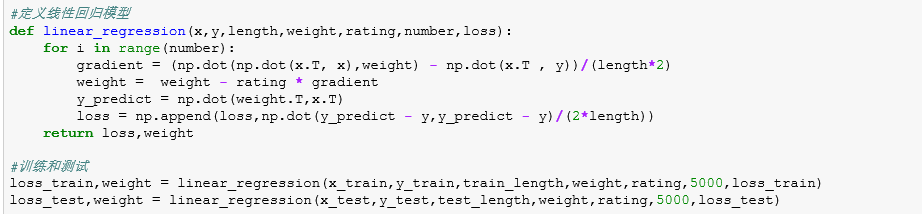
(8)、Select the appropriate threshold, mark the sample whose predict scores greater than the threshold as positive, on the contrary as negative. Get the loss Ltrain under the trainin set and Lvalidation by validating under validation set.

(9)、Repeate step 5 to 8 for several times, and drawing graph of Ltrain as well as Lvalidation with the number of iterations.

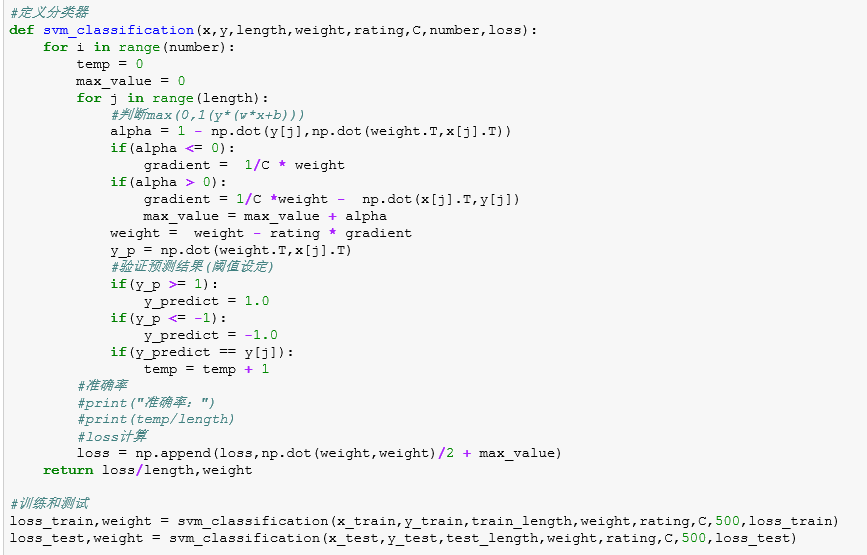
**7. Code:**

(Fill in the contents of 8-12 respectively for linear regression and linear classification)

**Linear Regression and Gradient Descent:**



**Linear Classification and Gradient Descent:**



**8. Selection of validation (hold-out, cross-validation, k-folds cross-validation, etc.):**

**Linear Regression and Gradient Descent:**

Use the hold-out

**Linear Classification and Gradient Descent:**

Use the hold-out

**9. The initialization method of model parameters:**

**Linear Regression and Gradient Descent:**

All parameter is 1

**Linear Classification and Gradient Descent:**

All parameter is 1

**10. The selected loss function and its derivatives:**

**Linear Regression and Gradient Descent:**

**Loss function:**

**Derivative:**

**Linear Classification and Gradient Descent:**

**Loss function:**

**Derivative:**

**11. Experimental results and curve:**

**Linear Regression and Gradient Descent:**

## Hyper-parameter selection (η, epoch, etc.):

Learning rate:0.01

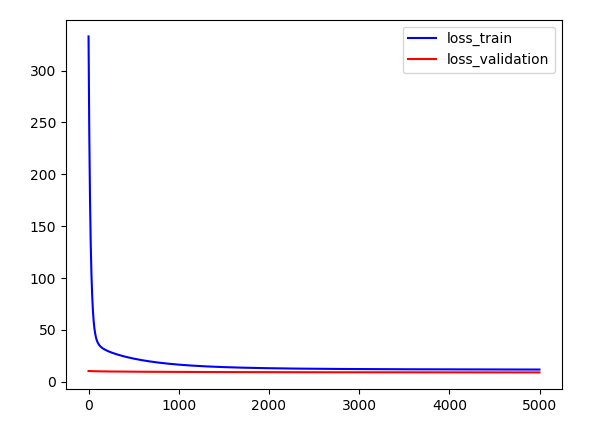
## Assessment Results (based on selected validation):

In the test sets, the loss would be between 9 and 11

## Predicted Results (Best Results):

The loss between the forecast result and the actual value does not exceed 11

## Loss curve:



**Linear Classification and Gradient Descent:**

## Hyper-parameter selection (η, epoch, etc.):

Learning rate:0.01

C:1000

## Assessment Results (based on selected validation):

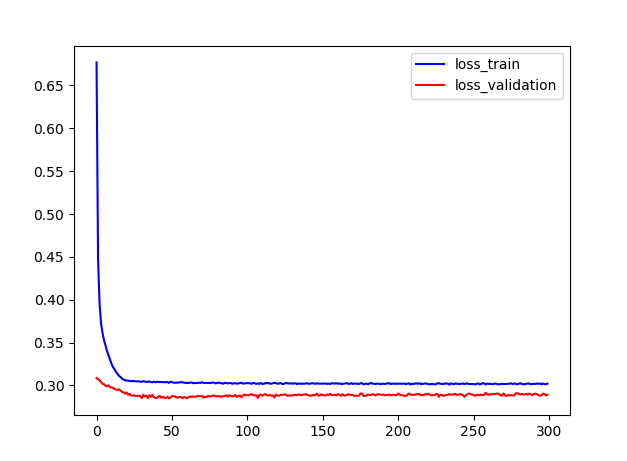
In the test sets, the loss would be between 0.25 and 0.3

## Predicted Results (Best Results):

The loss between the forecast result and the actual value does not exceed 0.3

The accuracy is about 87%

## Loss curve:



**12. Results analysis:**

**Linear Regression and Gradient Descent:**

The training set gradually converged after two thousandths times iter and the loss value close to 11

The test set gradually converged after one hundred times iter and the loss value close to 9

**Linear Classification and Gradient Descent:**

The training set gradually converged after forty times iter and the loss value close to 0.3

The training set gradually converged after thirty times iter and the loss value close to 0.25

The accuracy is about 87%

**13. Similarities and differences between linear regression and linear classification:**

**Different:**

**(1)、The loss function and the value of gradient are different**

**(2)、Linear Classification need other constants C**

**(3)、linear regression is to regression, and linear classification is to classification**

**The same:**

**(1)、All use the gradient descent to decrease gradient**

**(2)、All are the linear problem**

**14. Summary:**

**The experiment is not difficult, the experiment can be based on class PPT described in the encoding, loss function and gradient formulas are given. The main difficulty is the parameter adjustment. Parameters are too large, easy to oscillate, the parameters are too small, will spend more time to learn. The experiment is still quite big gains, on the one hand is learning python, on the one hand is a linear regression and linear classification of the deeper understanding of the final one is the** **Parameter adjustment.**