

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**   **徐志范**

**Student ID 201530613269**

**E-mail 1113082025@qq.com**

**Tutor**   **QingYao Wu**

**Date submitted** **2017. 12.04**

## 1.**Topic:**Linear Regression, Linear Classification and gradient Descent

**2. Time:** 2017.12.2

**3. Reporter: 徐志范**

**4. Purposes:**

（1）Further understand of linear regression and gradient descent.

（2）Conduct some experiments under small scale data set.

（3）Realize the process of optimization and adjusting parameters.

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

Linear Regression used scaled edition of Housing in LIBSVM Data,including 506 samples and each sample has 13 features. Then I divided it into training set, validation set.

Linear classification used scaled edition of australian in LIBSVM Data, including 690 samples and each sample has 14 features. Then I divided it into training set, validation set.

**6. Experimental steps:**

***6.1 Linear Regression and Gradient Descent***

(1) Load the experiment data, using load\_svmlight\_file function in sklearn library.

(2) Divide data set, dividing data set into training set and validation set using train\_test\_split function. And test set is not required in this experiment.

(3) Initialize linear model parameters.

(4) Choose loss function and derivation.

(5) Calculate gradient 2017-12-07_142140 toward loss function from all samples.

(6) Denote the opposite direction of gradient 2017-12-07_142444 as 2017-12-07_142516.

(7) Update model: 2017-12-07_142653. 2017-12-07_142724 is learning rate, a hyper-parameter that we can adjust.

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

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

***6.2 Linear Classification and Gradient Descent***

(1) Load the experiment data.

(2) Divide datas et into training set and validation set.

(3) Initialize SVM model parameters.

(4) Choose loss function and derivation.

(5) Calculate gradient 2017-12-07_142140 toward loss function from all samples.

(6) Denote the opposite direction of gradient 2017-12-07_142444 as 2017-12-07_142516.

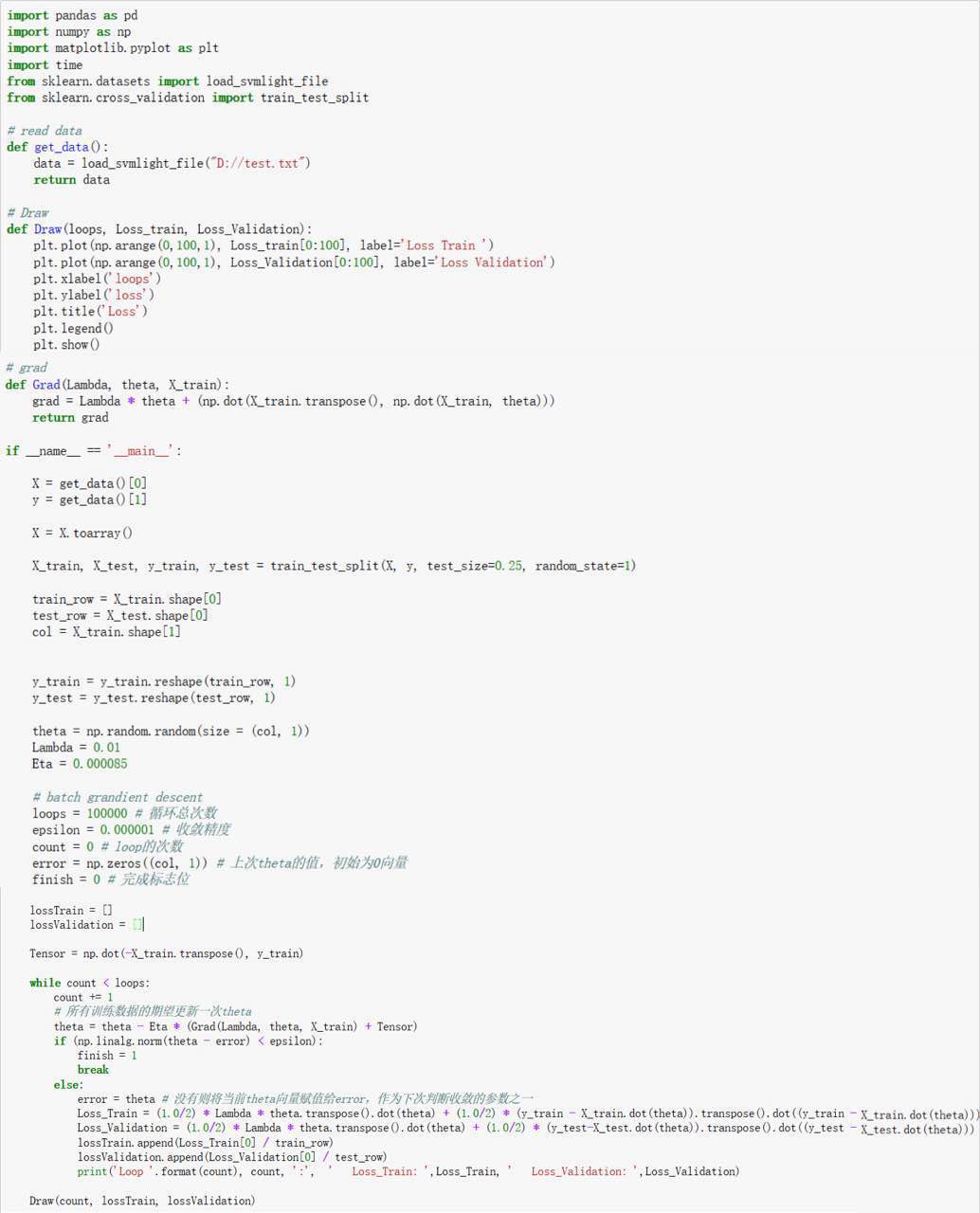
(7) Update model: 2017-12-07_142653. 2017-12-07_142724 is learning rate, a hyper-parameter that we can adjust.

(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  under the trainin set and by validating under validation set.

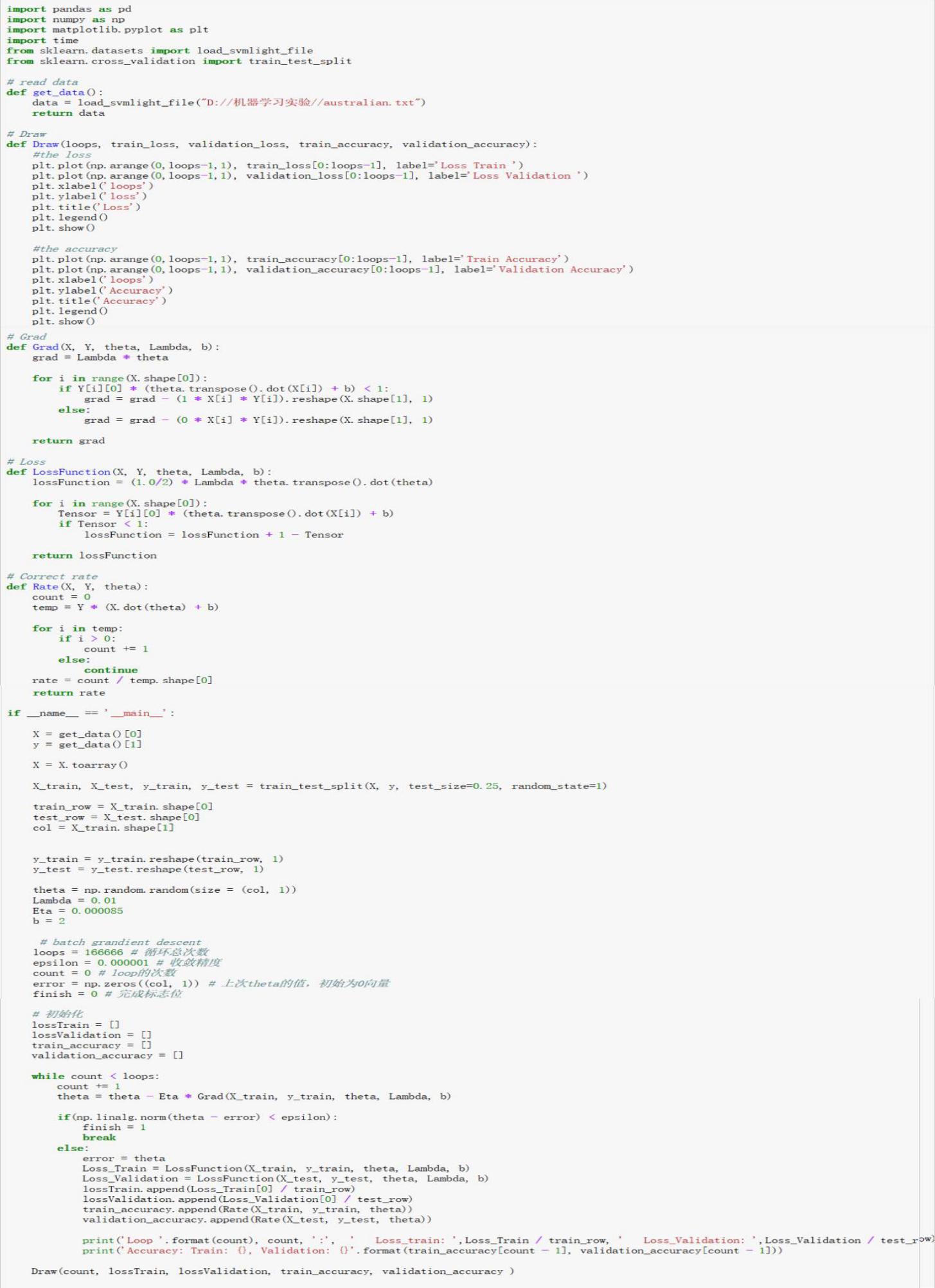
(9) Repeate step 5 to 8 for several times, and drawing graph of  as well as with the number of iterations.

**7. Code:**

***7.1 Linear Regression and Gradient Descent***

****

***7.2 Linear Classification and Gradient Descent***

****

**8. Selection of validation:**

Those two all used the simple cross validation, which involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the testing set).

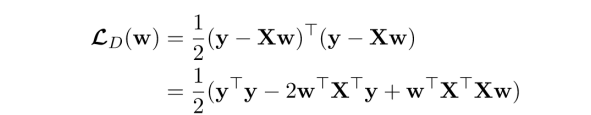
**9. The initialization method of model parameters:**

Initialize it randomly, with theta was initialized randomly between 0 to 1

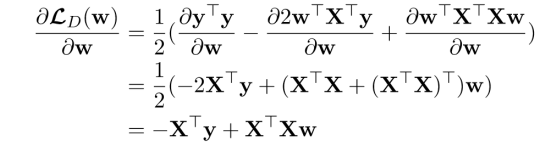
**10. The selected loss function and its derivatives:**

***10.1 Linear Regression and Gradient Descent***

(1) loss function:



(2) loss function’s derivatives:



***10.2 Linear Classification and Gradient Descent***

(1) loss function:



(2) loss function’s derivatives(if  > 0)







**11. Experimental results and curve:**

***11.1 Linear Regression and Gradient Descent***

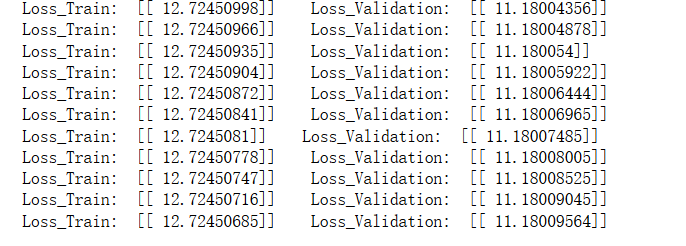
## Hyper-parameter selection :

## Lambda = 0.01, Eta= 0.000085, epsilon = 0.0001

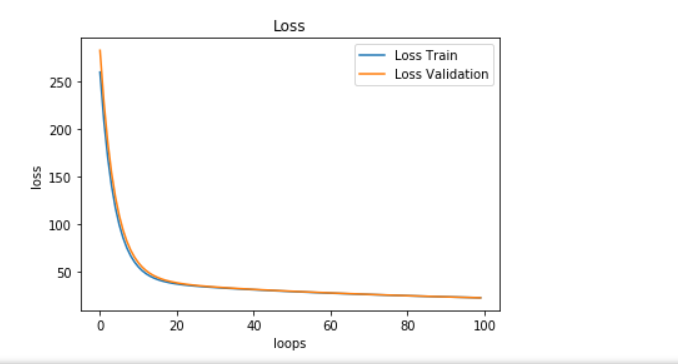
Assessment Results (based on selected validation):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lambda | Eta | epsilon | Loss\_Train | Loss\_Validation |
| 0.01 | 0.000085 | 0.000001 | 12.72431313 | 11.18739025 |
| 0.1 | 0.000085 | 0.000001 | 12.79176497 | 11.38460477 |
| 0.01 | 0.0001 | 0.000001 | 12.72431311 | 11.18745495 |
| 0.01 | 0.000085 | 0.00001 | 12.72431509 | 11.18662457 |
| 0.01 | 0.000085 | 0.0001 | 12.72450685 | 11.18009564 |

Predicted Results (Best Results):



## Loss curve:



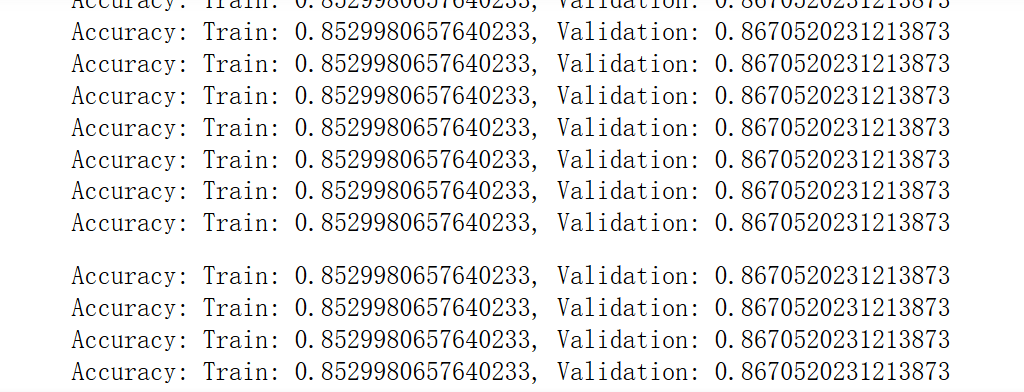
***11.2 Linear Classification and Gradient Descent***

Hyper-parameter selection :η= 0.000085

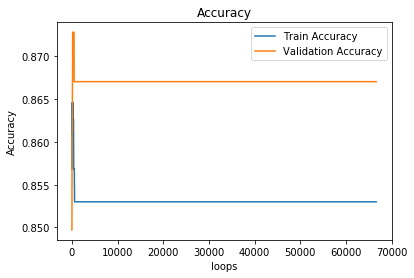
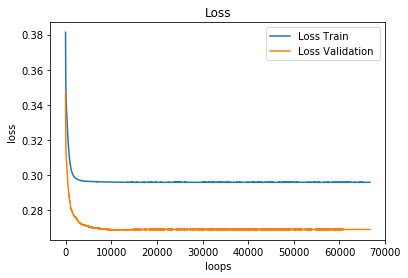
Assessment Results (based on selected validation):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Lambda | Eta | epsilon | Threshold(i) | Train\_Accuracy | Validation\_Accuracy |
| 0.01 | 0.000085 | 0.000001 | 0 | 85.299% | 86.705% |
| O.1 | 0.000085 | 0.000001 | 0 | 85.299% | 86.705% |
| 0.01 | 0.00001 | 0.000001 | 0 | 85.106% | 86.705% |
| O.1 | 0.000085 | 0.00001 | 0.05 | 85.299% | 86.705% |
| O.1 | 0.000095 | 0.000001 | 0 | 85.299% | 86.705% |

Predicted Results (Best Results):



## Loss curve:



**12. Results analysis:**

***12.1 Linear Regression and Gradient Descent***

As the **Assessment Results** that they converge to a same roughly, and the best epsilon is 0.0001. Because the value of the loss function declined more quickly with this value than others with smaller value. And we can see that the training loss is a little smaller than validation loss sometimes, which may be caused by the size of dataset is too small.

***12.2 Linear Classification and Gradient Descent***

We can summarize that the smaller threshold is, within the certain range, the better accuracy will be. The value of Lambda or Eta has less influence on the accuracy. Some of the features of the data set are left in the validation set, so there is a large gap between loss. We could improve the results by using different validation sets.

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

Regression is needed when the response variable is numeric or continuous. For example, the predicted price of a consumer good. Thus regression trees are applicable for prediction type of problems as opposed to classification.

Classification as the name implies is used to separate the data set into classes belonging to the response variable. Usually the response variable has two classes: Yes or No (1 or 0). If the target variable has more than 2 categories, then a variant of the algorithm, called C4.5, is used. For binary splits however, the standard CART procedure is used. Thus classification are used when the response or target variable is categorical in nature.

In general, those two are essentially the same, which is the matching of the model. But the y value of classification problem, also known as label, is more discrete. Moreover, the same Y may correspond to a large number of X. These X have a certain range. So the classification problem is that X in some areas corresponds to only one Y. And the regression problem model is more inclined to X in a small area, even only one X) corresponds to a Y.

1. **Summary:**

## This is my first time to write code by python. It’s rather difficult for me and the first experiment took me a long time. After that, I feel my biggest results is that I start to understand what are the linear regression, linear classification and gradient descent, and know how do they work. At the same time, I have known a little about the differences between linear regression and linear classification. And I realized the effort will be better if the algorithm have a better hyper-parameter. But just as a beginner, there are so much knowledge for me to study with. It is still hard, but I will put my full-heart to study it.