

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**  **Lin Guoshi**

**Student ID 2015306122127**

**E-mail 1509282534@qq.com**

**Tutor**   **Tan Mingkui**

**Date submitted** **2017.12 .08**

**1. Topic:**

**Linear Regression, Linear Classification and Gradient**

**2. Time: 2017.12 .02**

**3. Reporter: Lin Guoshi**

**4. Purposes:**

1. Further understand of linear regression and gradient descent.

2. Conduct some experiments under small scale dataset.

3. Realize the process of optimization and adjusting parameters.

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

**Datasets:**

Linear Regression uses Housing in LIBSVM Data, including 506 samples and each sample has 13 features. You are expected to download scaled edition. After downloading, you are supposed to divide it into training set, validation set.

Linear classification uses australian in LIBSVM Data, including 690 samples and each sample has 14 features. You are expected to download scaled edition. After downloading, you are supposed to divide it into training set, validation set.

**Analysis:**

The two experiments are both based on small and scaled datasets. It’s not difficult to handled the dataset. The key point of the experiments may be the computation of the Loss and Gradient, as well as the selection of the hyper-parameter.

**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.

5.Choose loss function and derivation: Find more detail in PPT.

6.Calculate gradient toward loss function from all samples.

7.Denote the opposite direction of gradient G as D .

8.Update model: Wt=Wt-1+ηD ηis learning rate, a hyper-parameter that we can adjust.

9.Get the loss Ltrain under the training set and Lvalidation by validating under validation set.

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.

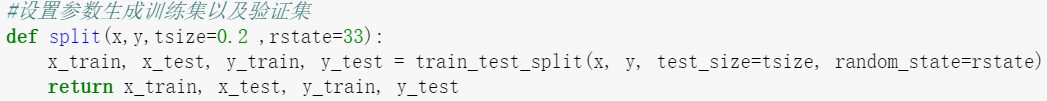
Repeate step 5 to 8 for several times, and drawing graph of as well as with the number of iterations.

**7. Code:**

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

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

**hold-out:choose the parameter and split the validation set and training set.**

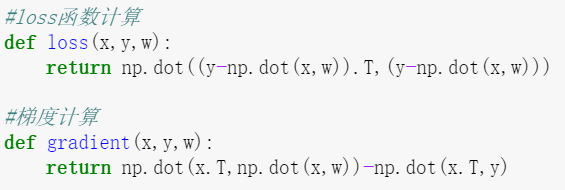


**9. The initialization method of model parameters:**

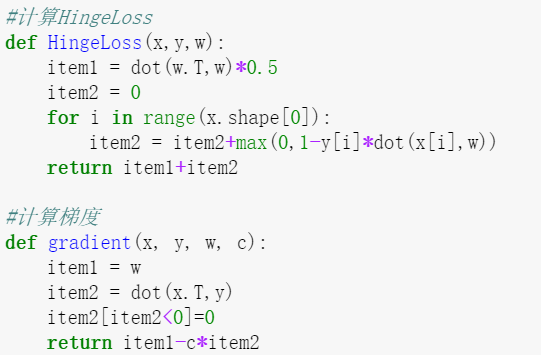
Set all the model parameters into zero.

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

**Linear regression:**



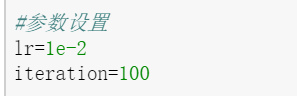
**Linear classification:**



**11. Experimental results and curve:**

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

**Linear regression:**



**Linear classification:**



## Assessment Results (based on selected validation):

**Linear regression:**

**Validation Size=0.2 ,random state=33**

**Linear classification:**

**Validation Size=0.2 ,random state=33**

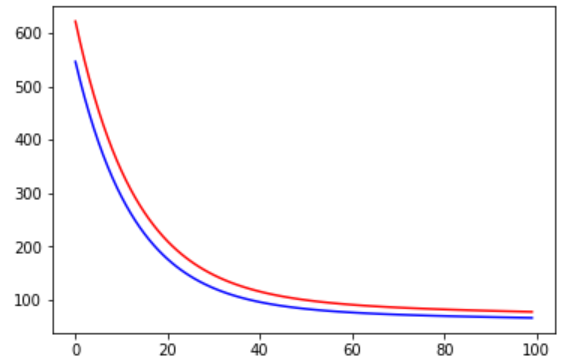
## Predicted Results (Best Results):

**Linear classification:**

Accuracy = 0.85

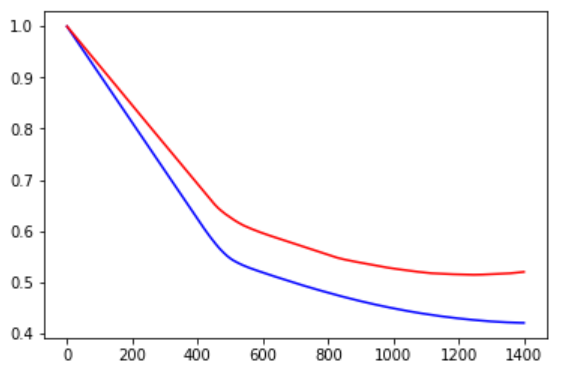
## Loss curve:

**Linear regression:**



The red line for validation loss and the blue line for training loss

**Linear classification:**



The red line for validation loss and the blue line for training loss

**12. Results analysis:**

By selecting proper parameters, both the training loss and validation loss of the linear regression and the linear classification can be decrease with the optimization method of gradient descent.And the learning rate and the epochs may be the key of whether the experiments can achieve success or not.

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

**Similarities:**

Linear regression and linear classification can work well on linear datasets and the two models are easy to implement but effective.

**Differences:**

Linear regression is choose to deal with the regression problem that predicts a continuous value while linear classification is choose to classify the samples into two or more classes.

**14. Summary:**

1. Methods for validation split is very important.

2. How to choose proper hyper-parameters is the key point of Machine Learning.

3. Gradient descent is a simple but effective method for optimization but it needs to be careful while selecting the right learning rate.