

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**   **张傲翔**

**Student ID 201530613580**

**E-mail iszhangax@gmail.com**

**Tutor**   **谭明奎**

**Date submitted** **2017.12 .8**

1. **Topic:** Linear Regression, Linear Classification and Gradient Descent

**2. Time:** 2017.12.2

**3. Reporter:**张傲翔

**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:**

Linear Regression uses [Housing](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html#housing) in [LIBSVM Data](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/), including 506 samples and each sample has 13 features. The data set is divided into training set, and validation set. Training set is 67% of the data set and validation set is 33%.

Linear classification uses [australian](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html" \l "australian" \t "_blank) in [LIBSVM Data](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/), including 690 samples and each sample has 14 features. The data set is divided into training set, and validation set. Training set is 67% of the data set and validation set is 33%.

**6. Experimental steps:**

*Linear Regression and Gradient Descent*

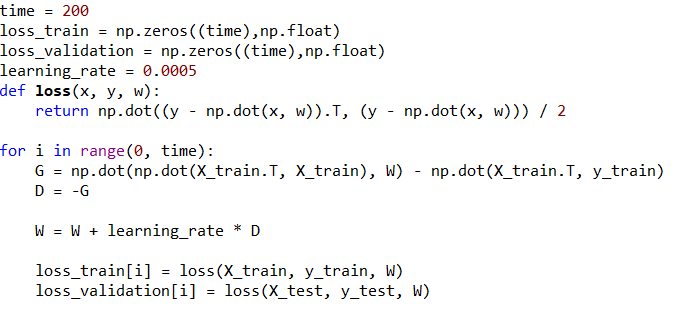
1. Load the experiment data. You can use load\_svmlight\_file function in sklearn library.
2. Devide dataset. You should divide dataset into training set and validation set using train\_test\_split function. Test set is not required in this experiment.
3. Initialize linear model parameters. You can choose 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. 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*

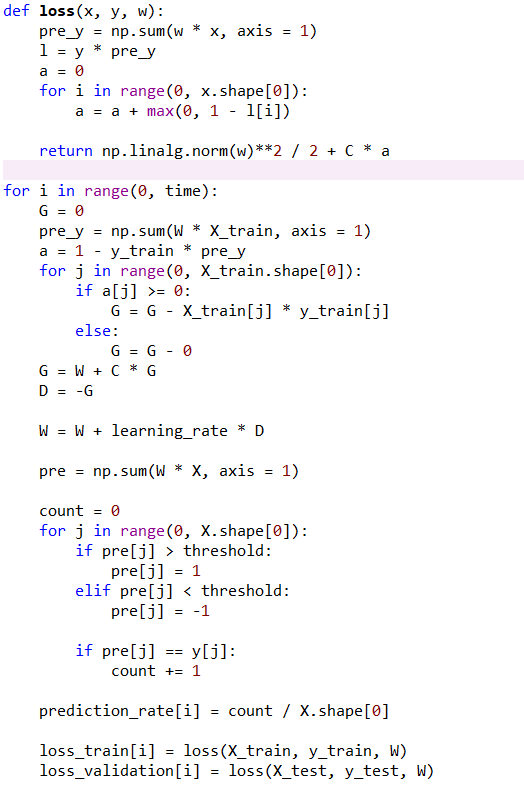
1. Load the experiment data.
2. Divide dataset into training set and validation set.
3. Initialize SVM model parameters. You can choose to set all parameter into zero, initialize it randomly or with normal distribution.
4. Choose loss function and derivation: Find more detail in PPT.
5. Calculate gradient G toward loss function from all samples.
6. Denote the opposite direction of gradient G as D.
7. Update model: Wt = Wt-1 +ηD. 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 train 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:**

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

Linear Classification and Gradient Descent

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

1. Initialize linear model parameters by setting all parameters into zero
2. Initialize SVM model parameters by setting all parameters into zero

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

1. Linear Regression and Gradient Descent

Least squared loss:

LD(W) = =

1. Linear Classification and Gradient Descent

Hinge loss:

**11. Experimental results and curve:**

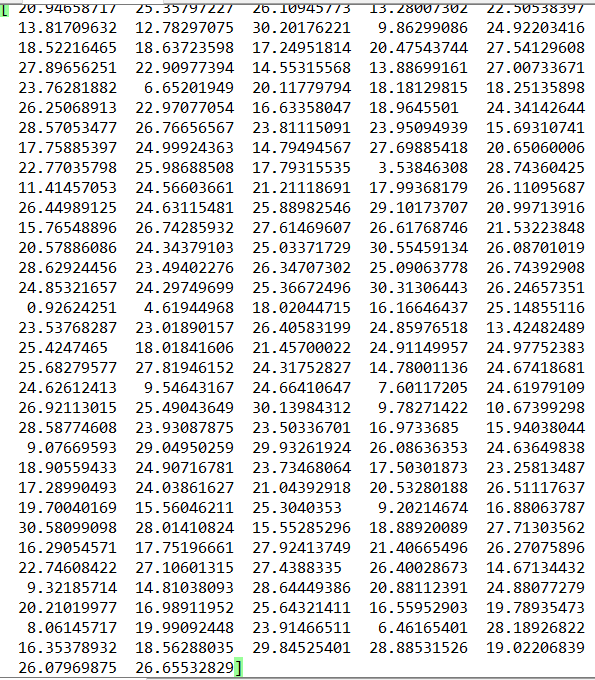
1. Linear Regression and Gradient Descent

## Hyper-parameter selection (η, epoch, etc.): η= 0.00003

## Assessment Results (based on selected validation):

## Predicted Results (Best Results):

## the prediction of X\_test:



## Loss curve:

1. Linear Classification and Gradient Descent

## Hyper-parameter selection (η, epoch, etc.): η= 0.0001 Threshold = -0.5

## Assessment Results (based on selected validation):

## Predicted Results (Best Results):right rate = 0.855072

## Loss curve:

**12. Results analysis:**

1. Linear Regression and Gradient Descent

As the times of iterations increases, the loss of train set and validation set decrease. After enough iterations, the loss of train set and validation become stable. However, since the model is obtained from the training set, the loss of the training set is less than the loss of the test set. The result of the test set predicted by the model with the least loss is similar to the original data

1. Linear Classification and Gradient Descent

As the times of iterations increases,, the loss of train set and validation set decrease. After enough iterations, the loss of train set and validation become stable. The accuracy of the classification results tends to be stable with the increase of the times of iterations, and finally the optimal result is obtained.

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

**Similarities:** First of all, linear regression and linear classification are linear models, and secondly, iteratively using a gradient descent method.

**Differences:** Linear regression is based on the linear model and the known data. The linear classification is based on the linear model and is classified by the threshold. The final result is two data sets.

**14. Summary:**

Through this experiment has been a lot of gains. Although this experiment was the simplest of four experiments in machine learning, it took a long time to complete the experiment. First of all, through the experimental submission process to understand the experimental method of git; secondly in the process of completing the linear regression and linear classification deepen the understanding of the knowledge has been learned; initially mastered the machine learning method.

Each parameter in the forecast affects the outcome of the forecast and needs to be adjusted to get good results

In this experiment also encountered a lot of difficulties, through these difficulties found that learning process should be added to their own understanding, the direct application of existing knowledge is difficult to achieve good results, and due to knowledge of the more likely to appear error.