

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**  Jalil Hassan

**Student ID 201728800097**

**E-mail hassanjalil723@yahoo.com**

**Tutor**   **Prof. Tan**

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

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

**2. Time:** 2017.12.8

**3. Reporter:** Jalil Hassan

**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:**

(1) 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. After downloading, you are supposed to divide it into training set, validation set.   
 (2) 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. After downloading, you are supposed to divide it into training set, validation set.

**6. Experimental steps:**

(1) Linear Regression and Gradient Descent

Load the experiment data.

Divide dataset. Divide dataset into training set and validation set using [train\_test\_split](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html) function. Test set is not required in this experiment.

Initialize linear model parameters. Choose to set all parameter into zero, initialize it randomly or with normal distribution.

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

Calculate gradient toward loss function from all samples.

Denote the opposite direction of gradient as.

Update model. Learning rate is a hyper-parameter that we can adjust.

Get the loss under the training set and by validating under validation set.

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

(2) Linear Classification and Gradient Descent

Load the experiment data.

Divide dataset into training set and validation set.

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

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

Calculate gradient toward loss function from all samples.

Denote the opposite direction of gradient as.

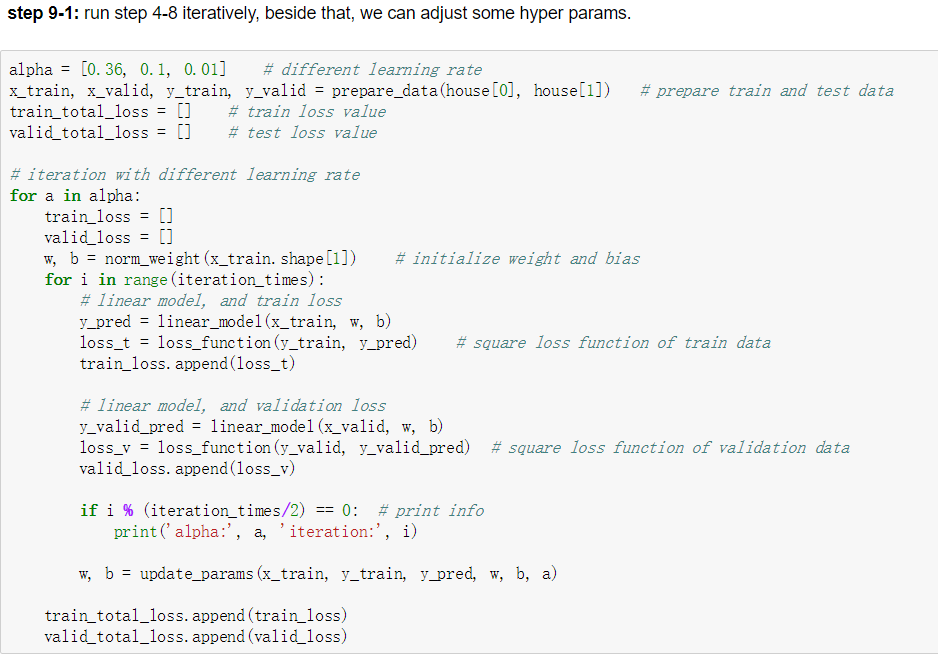
Update model:   is learning rate, a hyper-parameter that we can adjust.

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 training set and by validating under validation set.

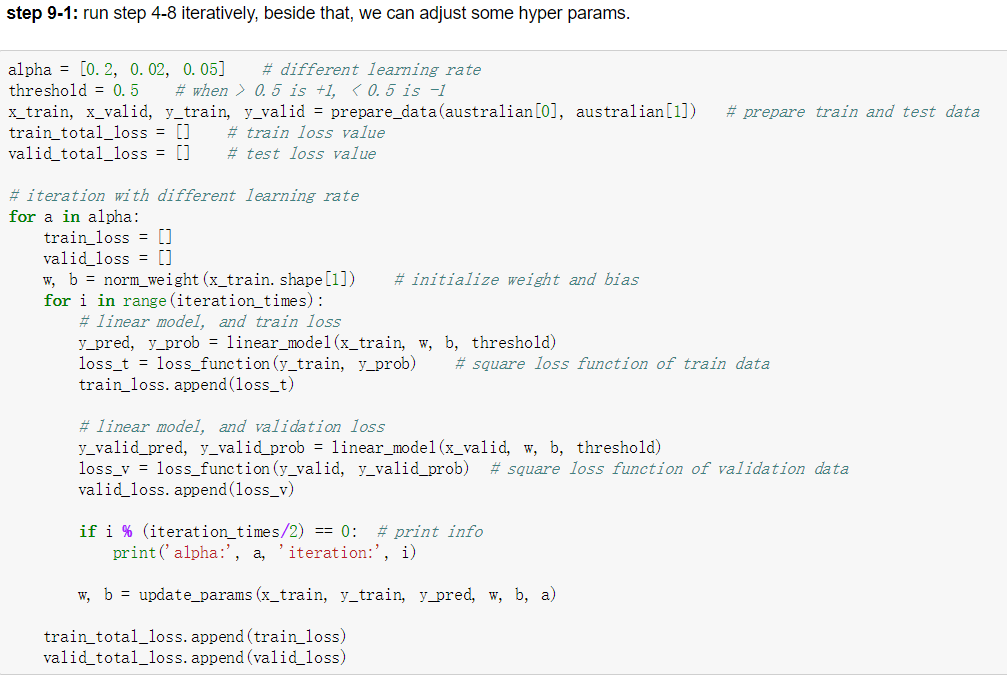
Repeat step 5 to 8 for several times, and drawing graph of as well as with the number of iterations.

**7. Code:**

(1) Linear Regression:

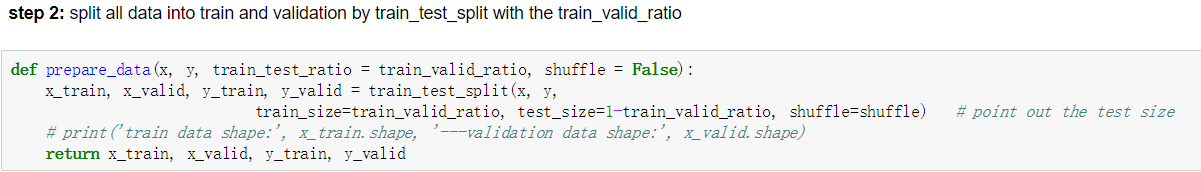


(2) Linear Classification:



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

Randomly select 0.2 of the total dataset as validation set.



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

Initialize parameters with normal distribution

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

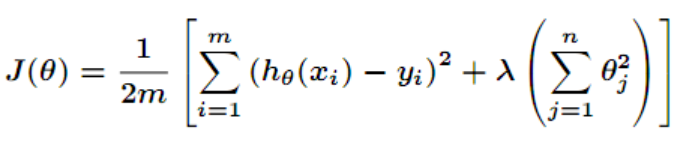
(1) Linear Regression

Input: x

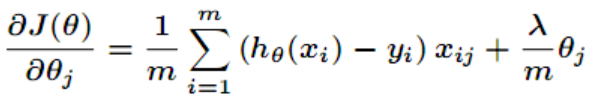
Output: y\_pred

True: y

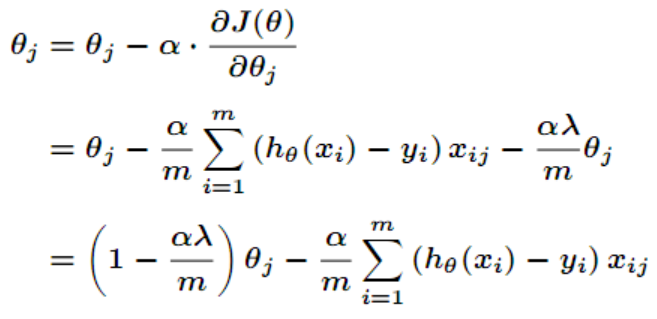
L2 regulation:



Derivation:



Update parameters:



​

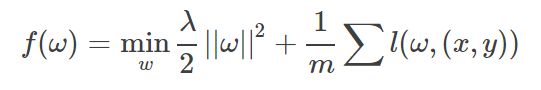
(2) Linear Classification

Input: x

Output: y\_pred

True: y

Target function:



And,

C:\Users\cike\AppData\Local\Temp\1512699372(1).png

Update parameters:

C:\Users\cike\AppData\Local\Temp\1512699615(1).png

**11. Experimental results and curve:**

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

1. Linear regression

η = [0.36, 0.1, 0.01] and epoch=500.

1. linear classification

η = [0.36, 0.1, 0.01] and epoch=500.

## Assessment Results (based on selected validation):

1. Linear regression

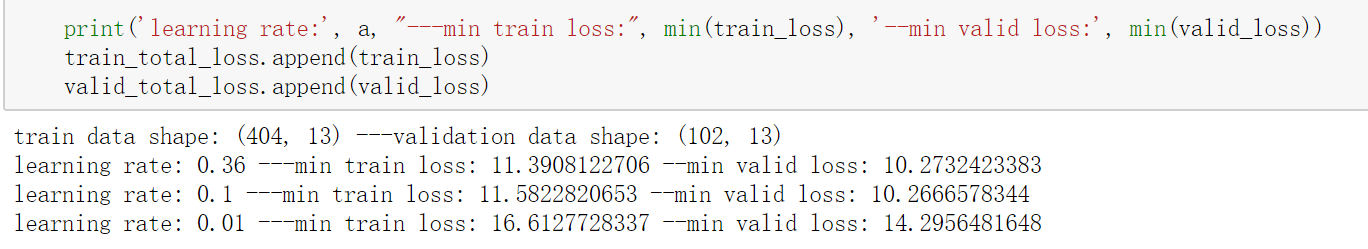
validation

(2) Linear classification

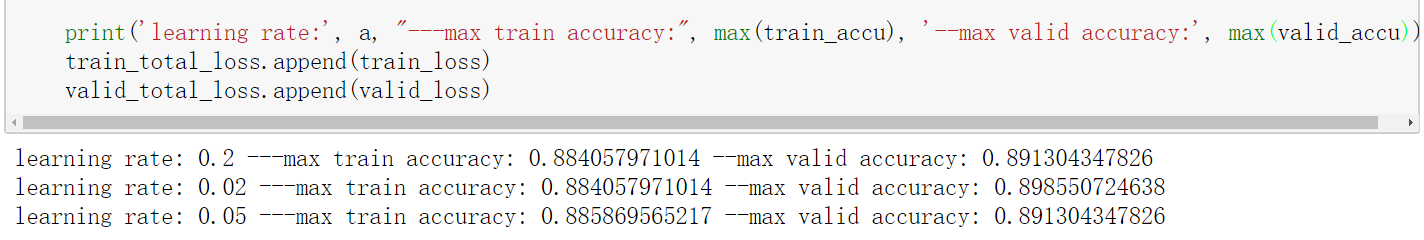
accuracy

## Predicted Results (Best Results):

1. Linear regression:

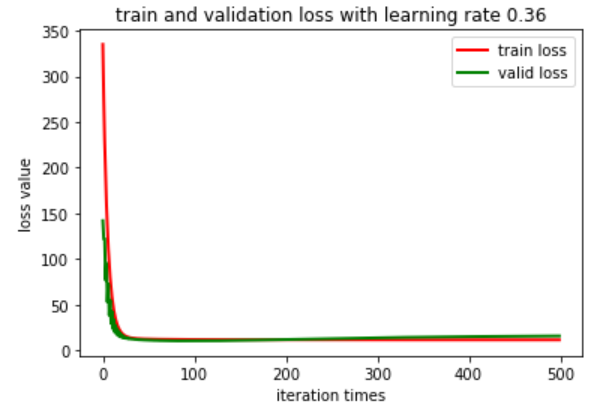


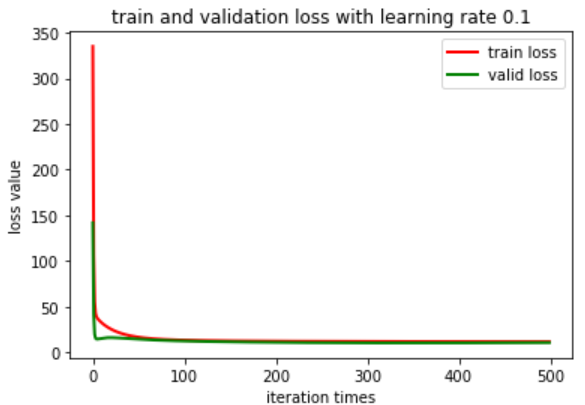
1. Linear classification:

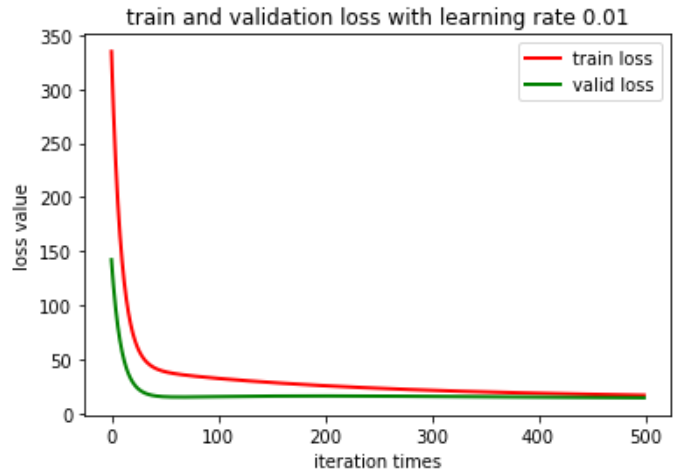


## Loss curve:

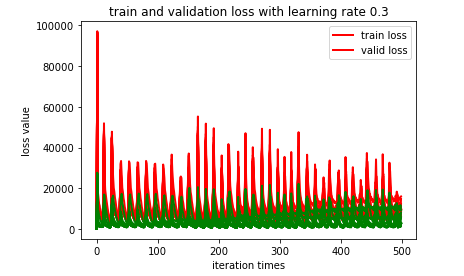
Linear regression:

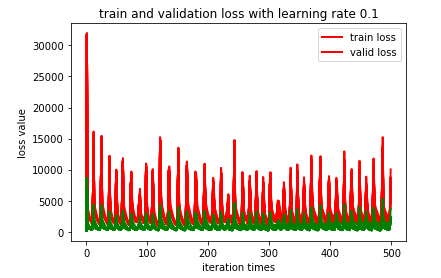


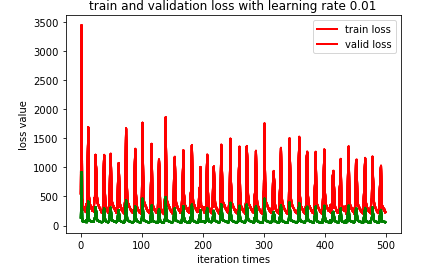




Linear Classification:







**12. Results analysis:**

The loss of both train and validation drop by increasing iteration times, which means the parameters become more and more useful. The curve tends to be smooth with different learning rate. When the learning rate set large one, the validation loss will larger than train loss, which means overfitting. And when the learning rate set small one, the validation loss will smaller than train loss.

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

Although with different loss function and model, linear regression and linear classification can be viewed as optimization problem, which is solved by minimizing the error of the model to the data. The most difference between them is the linear regression has a continuous output, while linear classification has a categorical output. In our setting, we use linear SVM model with hinge loss function in the linear classification, when we compute output of input data, we should transfer the output to label by predefined threshold.

**14. Summary:**

In my experiments, I use linear model and mean square loss function in the linear regression, and linear SVM model with hinge loss function in the linear classification. I obtain many knowledge about linear regression and classification, also gradient descent, especially the gradient descent, which is a powerful tool to gain proper parameters. Besides that, I realize that there are many parameters should be adjusted in Machine Learning, like learning rate, epochs and so on. Particularly, Machine Learning is data hungry, which means we should get more and more data to train our model, then we can get a better performance.