

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**   **袁强**

**Student ID 201530613511**

**E-mail yuanqiangscut@163.com**

**Tutor**  **谭明奎老师，陈耀佛助教**

**Date submitted** **2017. 12. 8**

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

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

**3. Reporter:** 袁强

**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. The dataset for linear regression including 506 samples and each sample has 13 features. I use 2/3 for training and the others for validation.

2. The dataset for linear classification including 690 samples and each sample has 14 features. I use 2/3 for training and the others for validation.

**6. Experimental steps:**

1. Load the experiment data and divide it into training set and validation set.

2. Initialize model parameters, I set all parameters into zero.

3. Choose loss function. In linear regression, choose least squard loss and in linear classification choose hinge loss.

4. Calculate gradient G toward loss function for all samples.

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

6. Update model: Wt = Wt-1 + ηD.

7. In linear regression: Get the loss Ltrain under the training set and Lvalidation by validating under validation set.

In linear classification: Select the appropriate threshold, mark the sample whose predict scores greater than the threshold as positive, on the contrary as negative. And then, get the loss Ltrain under the training set and Lvalidation by validating under validation set.

8. Repeate step 4 to 7 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)

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

Both linear regression and linear classification are hold-out.

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

Both are set all parameters into zero.

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

Linear Regression:

Loss function:

Derivatives:

Linear Classification:

Loss function:

Derivatives:

**11. Experimental results and curve:**

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

Linear Regression: learning rate η is 0.0001

Iteration num is 20

Linear Classification: learning rate η is 0.01

Iteration num is 200

Threshold is 0.5

C is 1

## Assessment Results (based on selected validation):

Linear Regression: blue line means the true validation label values

and red line means the prediction in validation set.

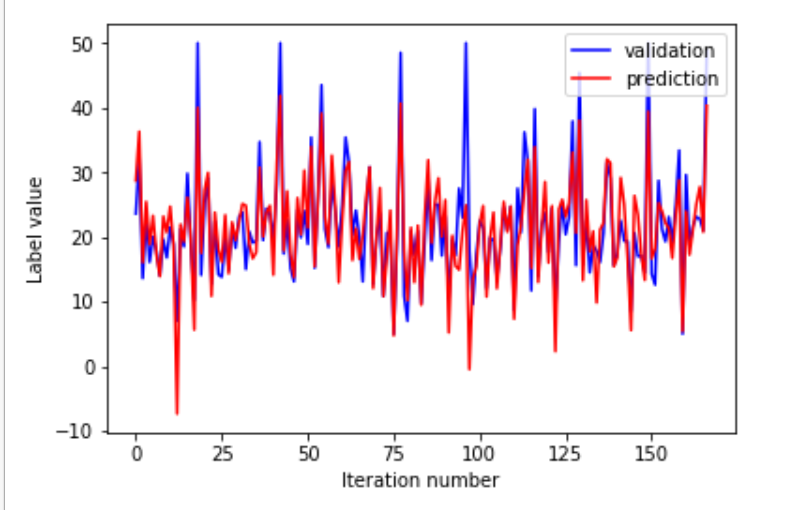


Figure 1

Linear Classification:

The accuracy rate is about 82% after 125 iterations.

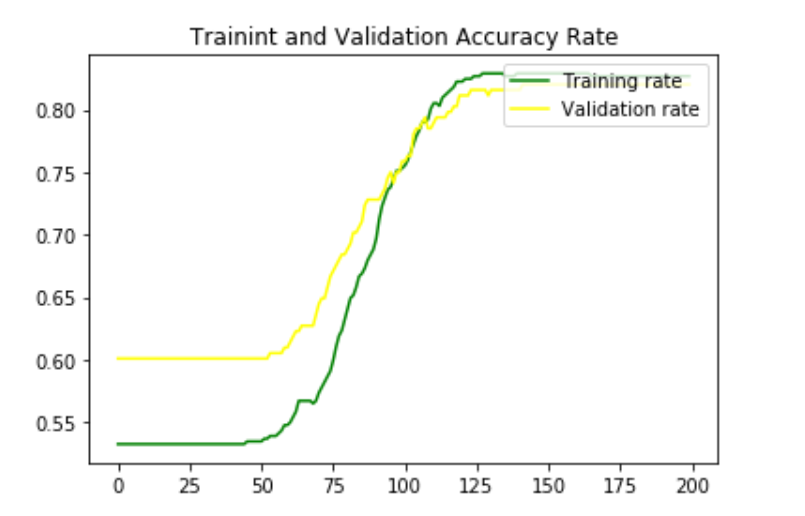


Figure 2

## Predicted Results (Best Results):

Linear Regression:

## Loss curve:

Linear Regression:

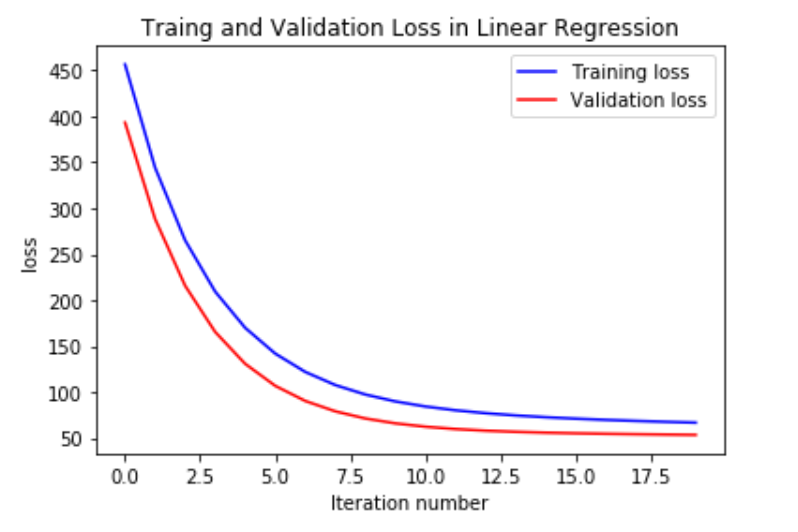


Figure 3

Linear Classification:

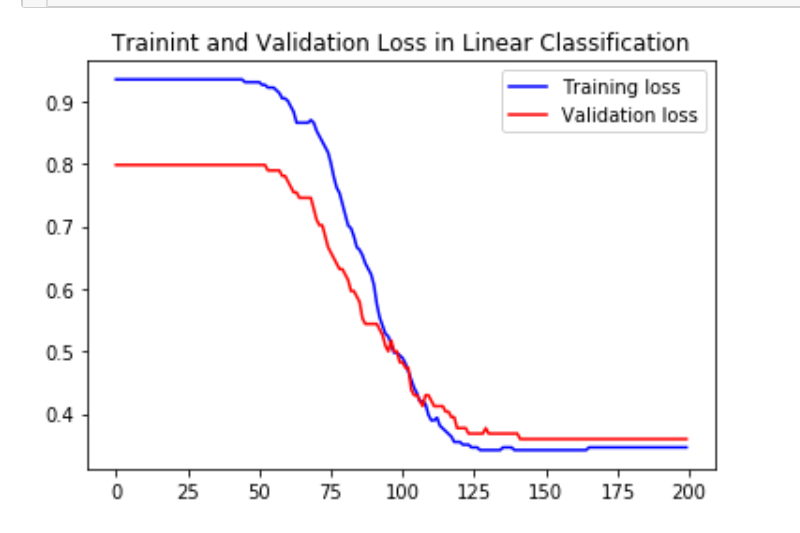
\

Figure 4

**12. Results analysis**

Linear Regression:

In Figure 1 we can see that the prediction curve fits well.

Linear Classification:

In Figure 2 we can see the accuracy rate nearly 82%.

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

Similarities:

Linear regression and linear classification have the same operation process.

Differences:

1. Linear regression is a prediction task, and the other one is a classification task.
2. They have different loss function.

**14. Summary:**

In this lab, I have a further understanding of linear regression and gradient descent. When I do these tasks in practice, I do meet some problems. And after solving them, I learned a lot. Besides, this lab actually make me aware of that machine learning is not a boring subject. When the program runs successfully and gives a reasonable result, I feel very exciting. This is the first lab, and I learned a lot from it.