

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**   **庄仁鑫**

**Student ID 201530613979**

**E-mail 529266221@qq.com**

**Tutor**   **谭明奎**

**Date submitted** **2017. 12 . 08**

**1. Topic:** 1. Linear Regression, Linear Classification and Gradient Descent；2. Logistic Regression, Linear Classification and Stochastic Gradient Descent

**2. Time:** 2017.12.02 - 2017.12.08

**3. Reporter:** 庄仁鑫

**4. Purposes:**

For topic 1: Further understand of linear regression and gradient descent;

Conduct some experiments under small scale dataset;

Realize the process of optimization and adjusting parameters;

For topic 2: Compare and understand the difference between gradient descent and stochastic gradient descent;

Compare and understand the differences and relationships between Logistic regression and linear classification;

Further understand the principles of SVM and practice on larger data.

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

For topic 1: Housing in LIBSVM Data, including 506 samples and each sample has 13 features.

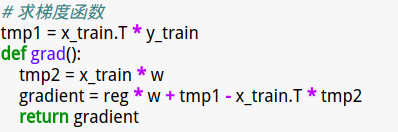
For topic 2: australian in LIBSVM Data, including 690 samples and each sample has 14 features.

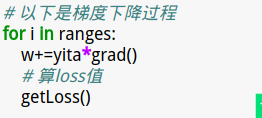
Every sample begins with the label followed by several features. For regression problem the label is a continuous real value and for classification problem the label is the class which is either +1 or -1.

1. **Experimental steps:**
2. .Load the data set and then split the data into training data set and validation data set.
3. .Initialize linear model parameters.
4. .Choose loss function and derivation.
5. .Calculate gradient G toward loss function from all samples.Denote the opposite direction of gradient G as D.
6. .Use gradient descent to update the parameter. For linear classification, select the appropriate threshold, mark the sample whose predict scores greater than the threshold as positive, on the contrary as negative.
7. .Get the loss under the training set and the loss by validating under validation set.
8. .Repeat step 5 to 8 for several times, and drawing graph of loss of training as well as loss of validation with the number of iterations.
9. **Code:**

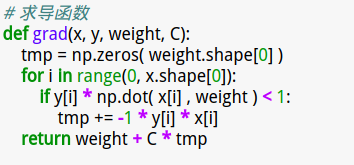
By screenshot:

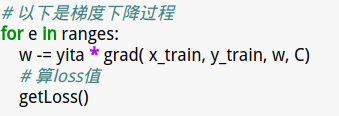
Regression:





Classification:





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

For both linear regression and linear classification, I use hold-out validation. See the code:





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

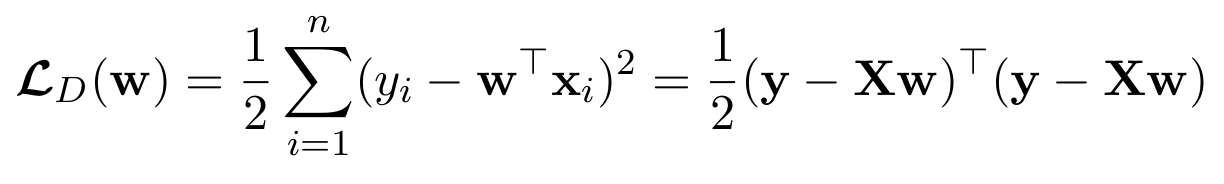
For both linear regression and linear classification, I randomly initialize the model parameter w. See the code:

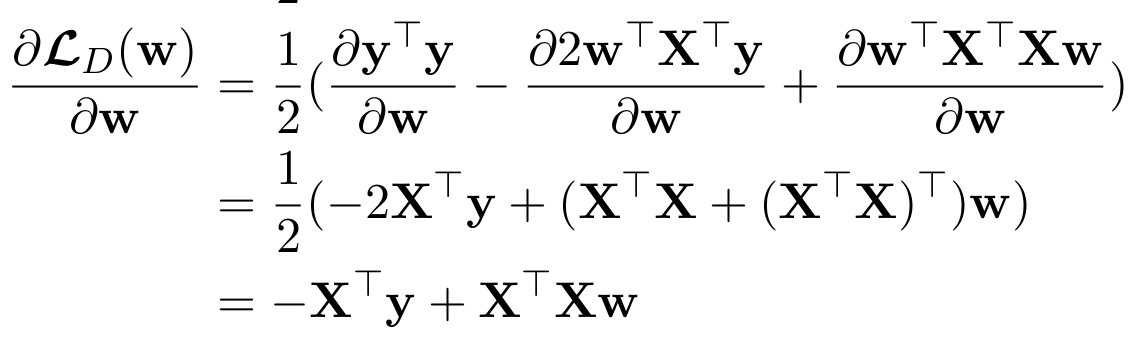




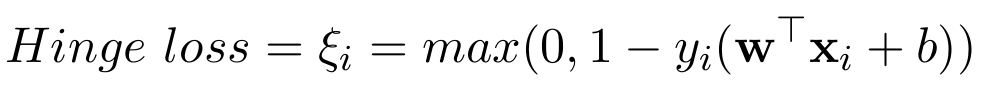
1. **The selected loss function and its derivatives:**

For linear regression, the loss function is square loss.

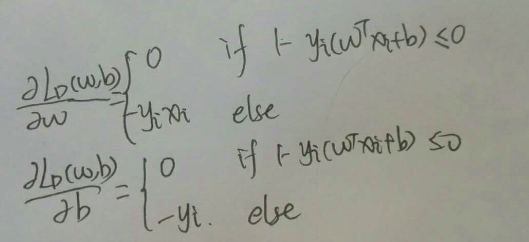
 It’s continuously differentiable. Here is its derivative:



For linear classification, the loss function is hinge loss.



It’s partially differentiable. Here is its derivative:



1. **Experimental results and curve:**

**For linear regression:**

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

1. .The learning rate is yita = 0.01 and regularizer reg=0.00001
2. .The learning rate is yita = 0.001 and regularizer reg=0.00001
3. .The learning rate is yita = 0.001 and regularizer reg=1
4. .The learning rate is yita = 0.001 and regularizer reg=100
5. .The learning rate is yita = 0.001 and regularizer varies from 0.1 to 0.0001

## Assessment Results (based on selected validation):

1. .The curve is smooth and the validation loss reached a small value.
2. .The curve is smooth too but the validation loss at the end of iterations is greater than the last situation. So the learning rate yita = 0.01 is reasonable.
3. .The curve is smooth and the validation loss at the end of iterations is small. The performance is the same as the first situation.
4. .The curve is abnormally rising. This is an inadequate regularizer.
5. .The performance is much the same as the first situation.

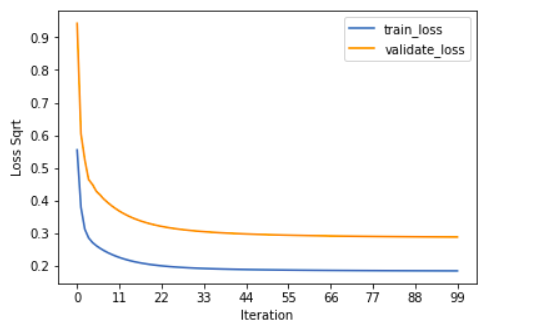
## Predicted Results (Best Results):

ranges=range(0,100)

yita=0.001 # 学习率

reg=0.1 # lambda

## Loss curve:



**For linear classification:**

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

1. .The learning rate is yita = 0.01 and C = 1.
2. .The learning rate is yita = 0.001 and C = 1.
3. .The learning rate is yita = 0.0001 and C = 1.
4. .When turning up the C, the curve is becoming steeper and steeper. So C should remain as 1.

## Assessment Results (based on selected validation):

1).It turns out that the curve is jumping up and down. So yita should be decreased.

2).This time the curve is much smoother, but it’s too steep. Might miss the minimum.

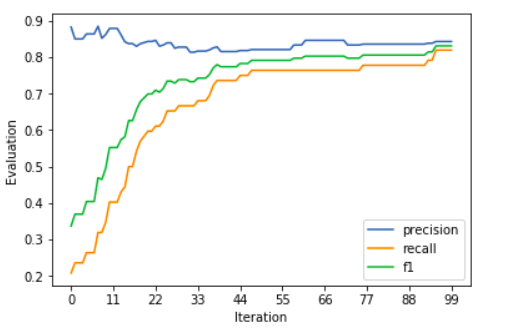
3).Now the curve is satisfactory. It decreases at an adequate speed. Besides, the precision, recall and F1 value are high.

## Predicted Results (Best Results):

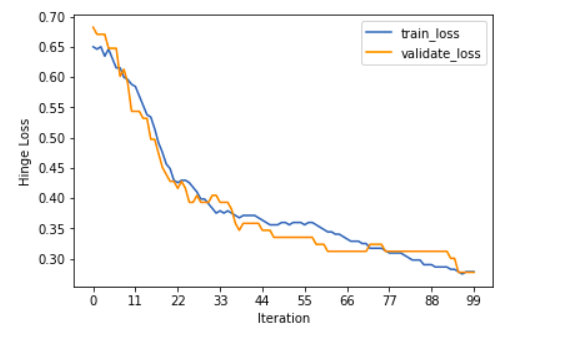
yita = 0.0001 学习率

C = 1

ranges=range(0, 100) 循环次数



## Loss curve:



1. **Results analysis:**

**For linear regression:**

For this data set, the loss is easy to converge. The Hyper-parameter is easily adjusted. And after relatively less iterations, the validation loss reaches satisfactorilly small value.

**For linear classification:**

With iterations going on, the classifier is performing better and better. Although there exists vibration, the overall performance is satisfactory.

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

For linear regression, the output is a continuous value. For linear classification, we should choose a threshhold to determine the class label according to the comparision of the output value and the threshhold. The methods are the same, gradient descent. And we both need to adjust the Hyper-parameter to get a good learner, which is evaluated by the loss function you chose.

1. **Summary:**

From this experiment, I learned the basic method to train a learner to address linear regression and linear classification problems. This should be a firm foundation for future studies of machine learning.