华南理工大学

South China University of Technology

The Experiment Report of *Machine Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Shen Fu

Supervisor:
Qingyao Wu

Student ID:
201720144955

Grade:
Graduate

December 14, 2017

Linear Regression, Linear Classification and Gradient Descent

***Abstract*—In this experiment report, two experiments about linear regression, linear classification and gradient descent are presented. Both experiments are under small scale dataset. First experiment is about linear regression, which uses Housing in LIBSVM Data, including 506 samples and each sample has 13 features. Second experiment is linear classification, which uses Australian in LIBSVM Data, including 690 samples and each sample has 14 features. The detailed experiments theroy, implementation, results and iteration graphs are shown.**

I. Introduction

LINEAR regression is a linear approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. And in the field of machine learning, the goal of statistical classification is to use an object's characteristics to identify which class (or group) it belongs to. A linear classifier achieves this by making a classification decision based on the value of a linear combination of the characteristics.

To make a further understand of linear regression and gradient descent, two experiments in this report are about linear regression and linear classification, respectively. The experiments are both under small scale dataset and contain the process of optimization and adjusting parameters. Linear Regression uses Housing in LIBSVM Data, including 506 samples and each sample has 13 features. Linear classification uses australian in LIBSVM Data, including 690 samples and each sample has 14 features.

The experiments are run under the environment of python3. Python package including sklearn, numpy, jupyter, matplotlib are required. Therefore, the anaconda3 are used directly that has built-in python package above. The experimental code and drawing are completed on jupyter.

The rest parts of report are structured as follows: method and theory used in two experiments are presented in Section II. The overall implementation and results of experiments are shown in Section III and Section IV concludes the reports.

II. Methods and Theory

A. Linear Regression

In statistics, linear regression is a linear approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. The case of one explanatory variable is called simple linear regression. Given set of input, output pairs $D = \{(x_i, y_i)\}_{i=1}^{n}$, in which x_i is input and y_i is output, n is the size of vector.We want to use a function predicting y: $\widehat{y} = f(x)$. However,the prediction may be inconsistent with thegroundtruth. We calculate the differences by loss function:

$$\pounds D = \sum_{i=1}^{n} l(\widehat{y_i}, y_i) \tag{1}$$

where $l(\widehat{y_i}, y_i)$ can be chose as absolute value loss or least squares loss.

The model function of linear regression is

$$f(x; \omega_0, w) = \omega_0 + \omega_1 x_1 + ... + \omega_m x_m = w^T x + \omega_0 \tag{2}$$

in which w is a parameter. The process of training is finding minimizer of least squared loss:

$$w^* = arg \quad min \pounds D(w) \tag{3}$$

Since

$$\pounds D(w) = \frac{1}{2}\sum_{i=1}^{n}(y_i - w^T x_i)^2 = \frac{1}{2}(y - Xw)^T(y - Xw) \tag{4}$$

$$= \frac{1}{2}(y^T y - 2w^T X^T y + w^T X^T X w) \tag{5}$$

$$\frac{\partial \pounds D(w)}{\partial w} = \frac{1}{2}(\frac{\partial y^T y}{\partial w} - \frac{\partial 2 w^T X^T y}{\partial w} + \frac{\partial w^T X^T w}{\partial w}) \tag{6}$$

$$= -X^T y + X^T X w \tag{7}$$

$$w = (X^T X)^{-1} X^T y \tag{8}$$

Solve for optimal parameters w^*

$$w^* = (X^T X)^{-1} X^T y = argmin \pounds D(w) \tag{9}$$

B. Gradient descent

We use $d = -\frac{\partial \pounds D(w)}{\partial w}$ as the direction of optimization and have $\pounds D(w^{'}) = \pounds D(w + \eta d) \leq \pounds D(w)$, note that $\eta > 0$and

$$\eta[\frac{\partial \pounds_D(w)}{\partial w}]^T d = -\eta d^T d \leq 0 \tag{10}$$

Minimize loss by repeated gradient steps(when no closed form):
1.Compute gradient of loss with respect to parameters $\frac{\partial \pounds D(w)}{\partial w}$;
2.Update parameters with rate η

$$w^{'} \rightarrow w - \eta \frac{\partial \pounds_D(w)}{\partial w} \tag{11}$$

Learning rate η has a large impact on convergence. Too large η will cause oscillatory and may even diverge. Too small η will make too slow to converge.

C. Linear Classification

In the field of machine learning, the goal of statistical classification is to use an object's characteristics to identify which class (or group) it belongs to. A linear classifier achieves this by making a classification decision based on the value of a linear combination of the characteristics. A linear classier has the form:

$$f(x) = w^T x + b \tag{12}$$

where w is the normal to the line, and b is the bias, w is known as the weight vector and in 2D the discriminant is a line.Learning the SVM can be formulated as an optimization:

$$max_{w,b} \frac{2}{w} \tag{13}$$

$$s.t. w^T x_i + b \begin{cases} \geq 1 & y_i = +1 \\ \leq -1 & y_i = -1 \end{cases} \tag{14}$$

Introduce variable $\xi \geq 0$, for each i, which represents how much example i is on wrong side of margin boundary. If $\xi = 0$ then it is ok, if $0 < \xi_i < 1$ it is correctly classified, but with a smaller margin than $\frac{1}{\|w\|}$. If $\xi > 1$ then it is incorrectly classified. The optimization problem becomes:

$$min_{w,b} \frac{\|w\|^2}{2} + C \sum_{n}^{i=1} \xi_i \tag{15}$$

$$s.t. y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1, 2, ..., n \tag{16}$$

Hinge loss:

$$Hingeloss = xi_i = max(0, 1 - y_i(w^T x_i + b)) \tag{17}$$

The optimization problem becomes:

$$min_{w,b} \frac{\|w\|^2}{2} + \frac{C}{n} \sum_{n}^{i=1} max(0, 1 - y_i(w^T x_i + b)) \tag{18}$$

III. Experiments

A. Dataset

Linear Regression uses Housing in LIBSVM Data, including 506 samples and each sample has 13 features, which is divided into training set, validation set. Linear classification uses Australian in LIBSVM Data, including 690 samples and each sample has 14 features, which is divided into training set, validation set.

B. Implementation

1) Initialization: Both two experiments initialize with normal distribution.

2) selected loss function and derivatives: For linear regression and gradient descent

$$l = \frac{1}{2} \frac{1}{N} \sum_{i=1}^{n} (w^T x_i + b - y_i)^2 + \frac{1}{2} \lambda (\|w\|^2 + \|b\|^2) \tag{19}$$

$$l^{'} = \frac{1}{N} \sum_{i=1}^{n} x_i (w^T x_i + b - y_i) + \lambda (\|w\| + \|b\|) \tag{20}$$

are loss function and its derivatives. For linear classification and gradient descent is

$$hing\_loss(x, y; w, b, C) = max(0, 1 - Cy(wx + b) \tag{21}$$

$$l = \frac{1}{N} \sum_{i=1}^{n} hing\_loss(x_i, y_i) + \frac{1}{2} \lambda (\|w\|^2 + \|b\|^2) \tag{22}$$

$$l^{'} = -\frac{C}{N} \sum_{i=1}^{n} x_i, y_i I[hing\_loss(x_i, y_i) > 0] + \lambda (\|w\| + \|b\|) \tag{23}$$

are loss function and its derivatives.

3) process: The process of linear regression and gradient descent are as follows:

1. Load the experiment data using load_svmlight_file function in sklearn library.
2. Devide dataset. 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. Choose to set all parameter into zero, initialize it with normal distribution.
4. Choose loss function and derivation: as shown above.
5. Calculate gradient G toward loss function from all samples.
6. Denote the opposite direction of gradient G as D .
7. Update model: $W_t = W_{t-1} + \eta D$, where η is learning rate, a hyper-parameter that we can adjust.
8. Get the loss L_{train} under the training set and $L_{validation}$ by validating under validation set.
9. Repeate step 5 to 8 for several times, and drawing graph of L_{train} as well as $L_{validation}$ with the number of iterations.

The process of linear classification and gradient descent are as follows:

1. Load the experiment data.
2. Divide dataset into training set and validation set.
3. Initialize SVM model parameters. Set all parameter into zero, initialize it with normal distribution.
4. Choose loss function and derivation: as shown above.
5. Calculate gradient G toward loss function from all samples.
6. Denote the opposite direction of gradient G as D. 7. Update model: $W_t = W_{t-1} + \eta D$, where 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 L_{train} under the training set and $L_{validation}$ by validating under validation set.
9. Repeate step 5 to 8 for several times, and drawing graph of L_{train} as well as $L_{validation}$ with the number of iterations.

4) parameters: In first experiment, initial λ and η is set to be 0 and max number of epoch is 50. In second experiment, initial initial λ and η is set to be 0, C is 1, the threshold is 0.5 and max number of epoch is 50.

5) results: Linear Regression and Linear Classiffication The results of experiments are shown in Tables 1-2. The score is the opposite number of RMSE in Table I and the accuracy of classification model in Table II.

The best 5 models loss curves of linear regression are shown in Figs. 1-5.

The best 5 models loss curves of linear classification are shown in Figs. 6-10.

TABLE I
LINEAR REGRESSION

rank_test_score	mean_test_score	mean_train_score	params
1	-4.681854	-4.423003	'eta': 0.3, 'lamda': 0
2	-4.685007	-4.457880	'eta': 0.3, 'lamda': 0.01
3	-4.693212	-4.478356	'eta': 0.2, 'lamda': 0
4	-4.712620	-4.515686	'eta': 0.2, 'lamda': 0.01
5	-4.725967	-4.540245	'eta': 0.15, 'lamda': 0
6	-4.758658	-4.583277	'eta': 0.15, 'lamda': 0.01
7	-4.856789	-4.699688	'eta': 0.1, 'lamda': 0
8	-4.903707	-4.750715	'eta': 0.1, 'lamda': 0.01
9	-5.304375	-5.171324	'eta': 0.3, 'lamda': 0.1
10	-5.316694	-5.184696	'eta': 0.2, 'lamda': 0.1
11	-5.344523	-5.213879	'eta': 0.15, 'lamda': 0.1
12	-5.433636	-5.295621	'eta': 0.05, 'lamda': 0
13	-5.442221	-5.314059	'eta': 0.1, 'lamda': 0.1
14	-5.474032	-5.337602	'eta': 0.05, 'lamda': 0.01
15	-5.840682	-5.718619	'eta': 0.05, 'lamda': 0.1
16	-7.124982	-7.054805	'eta': 0.3, 'lamda': 0.5
17	-7.124983	-7.054806	'eta': 0.2, 'lamda': 0.5
18	-7.125009	-7.054834	'eta': 0.15, 'lamda': 0.5
19	-7.125871	-7.055720	'eta': 0.1, 'lamda': 0.5
20	-7.154038	-7.084576	'eta': 0.05, 'lamda': 0.5
21	-8.312507	-8.277844	'eta': 0.2, 'lamda': 1
22	-8.312507	-8.277844	'eta': 0.3, 'lamda': 1
23	-8.312507	-8.277844	'eta': 0.15, 'lamda': 1
24	-8.312508	-8.277845	'eta': 0.1, 'lamda': 1
25	-8.313410	-8.278792	'eta': 0.05, 'lamda': 1

TABLE II
LINEAR CLASSIFICATION

rank_test_score	mean_test_score	mean_train_score	params
1	0.863636	0.875541	'C': 1, 'eta': 0.05, 'lamda': 0.1, 'threshold': 0.6
2	0.861472	0.869048	'C': 1, 'eta': 0.1, 'lamda': 0.5, 'threshold': 0.4
3	0.859307	0.859307	'C': 1, 'eta': 0.1, 'lamda': 0.1, 'threshold': 0.6
3	0.859307	0.871212	'C': 1, 'eta': 0.05, 'lamda': 0.5, 'threshold': 0.4
3	0.859307	0.859307	'C': 1, 'eta': 0.02, 'lamda': 0.5, 'threshold': 0.5
3	0.859307	0.869048	'C': 1, 'eta': 0.1, 'lamda': 0.1, 'threshold': 0.5
3	0.859307	0.859307	'C': 2, 'eta': 0.02, 'lamda': 0.5, 'threshold': 0.4
3	0.859307	0.859307	'C': 1, 'eta': 0.1, 'lamda': 0.1, 'threshold': 0.4
3	0.859307	0.859307	'C': 2, 'eta': 0.1, 'lamda': 0.1, 'threshold': 0.5
10	0.857143	0.875541	'C': 1, 'eta': 0.1, 'lamda': 0.5, 'threshold': 0.5
10	0.857143	0.860390	'C': 1, 'eta': 0.05, 'lamda': 0.1, 'threshold': 0.5
12	0.854978	0.864719	'C': 1, 'eta': 0.05, 'lamda': 0.1, 'threshold': 0.4
12	0.854978	0.870130	'C': 2, 'eta': 0.1, 'lamda': 0.1, 'threshold': 0.4
14	0.850649	0.857143	'C': 2, 'eta': 0.05, 'lamda': 0.5, 'threshold': 0.4
15	0.848485	0.866883	'C': 1, 'eta': 0.1, 'lamda': 0, 'threshold': 0.4
16	0.846320	0.863636	'C': 1, 'eta': 0.02, 'lamda': 0.5, 'threshold': 0.4
16	0.846320	0.832251	'C': 1, 'eta': 0.05, 'lamda': 0, 'threshold': 0.5
16	0.846320	0.863636	'C': 1, 'eta': 0.05, 'lamda': 0, 'threshold': 0.4
16	0.846320	0.836580	'C': 1, 'eta': 0.1, 'lamda': 0, 'threshold': 0.5
16	0.846320	0.860390	'C': 1, 'eta': 0.1, 'lamda': 0, 'threshold': 0.6
21	0.844156	0.849567	'C': 2, 'eta': 0.05, 'lamda': 0, 'threshold': 0.4
21	0.844156	0.852814	'C': 1, 'eta': 0.05, 'lamda': 0.5, 'threshold': 0.5
21	0.844156	0.860390	'C': 2, 'eta': 0.05, 'lamda': 0, 'threshold': 0.5
24	0.841991	0.857143	'C': 2, 'eta': 0.05, 'lamda': 0.1, 'threshold': 0.4
24	0.841991	0.859307	'C': 2, 'eta': 0.05, 'lamda': 0, 'threshold': 0.6
26	0.839827	0.848485	'C': 2, 'eta': 0.1, 'lamda': 0, 'threshold': 0.4
27	0.837662	0.840909	'C': 1, 'eta': 0.05, 'lamda': 0.5, 'threshold': 0.6
28	0.835498	0.853896	'C': 1, 'eta': 0.1, 'lamda': 0.5, 'threshold': 0.6
28	0.835498	0.837662	'C': 1, 'eta': 0.02, 'lamda': 0.5, 'threshold': 0.6

6) Analysis: According to results obtained by two experiments, it can be shown that gradient decent is a valid method to optimize both regression problem and classification problem. At the beginning of optimization, the loss decrease quickly and the accuracy of model increase sharply. With epoch goes by, the model verges to be optimized. The greater learning rate is, the quicker loss decreases. However, large learning rate may lead to vibrating. The metrics that used to evaluate models are important. In general, a smaller loss leads to better metrics, but there are exceptions, as the relationship between accuracy and loss in classification experiments.

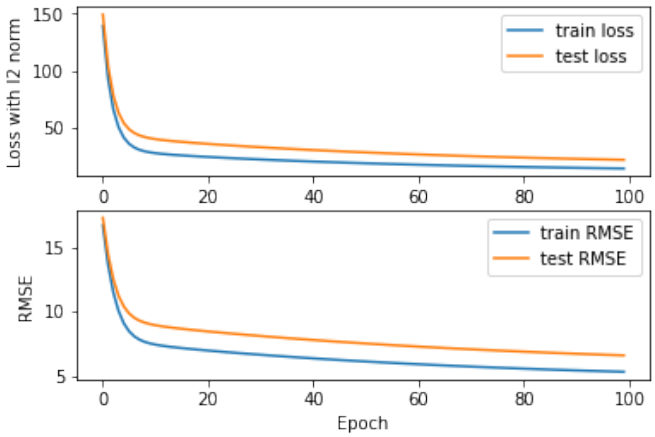

Fig. 1. Figure of 'eta': 0.05, 'lamda': 0

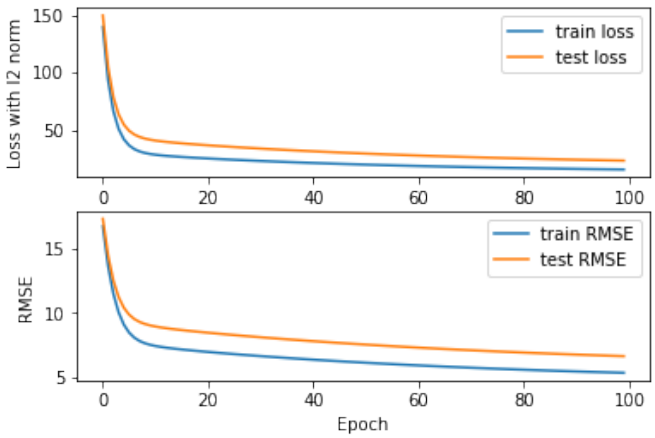

Fig. 2. Figure of 'eta': 0.05, 'lamda': 0.01

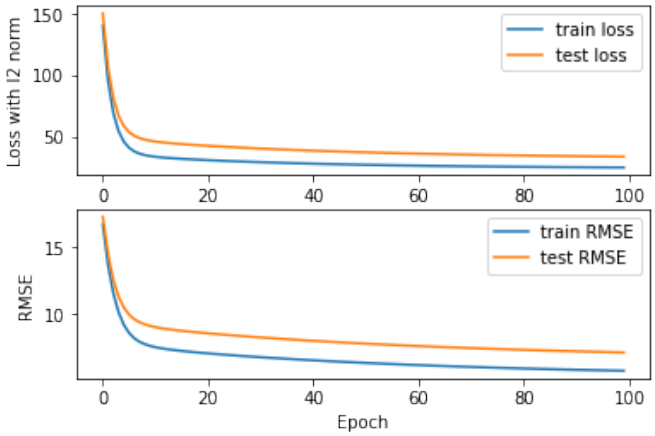

Fig. 3. Figure of 'eta': 0.05, 'lamda': 0.1

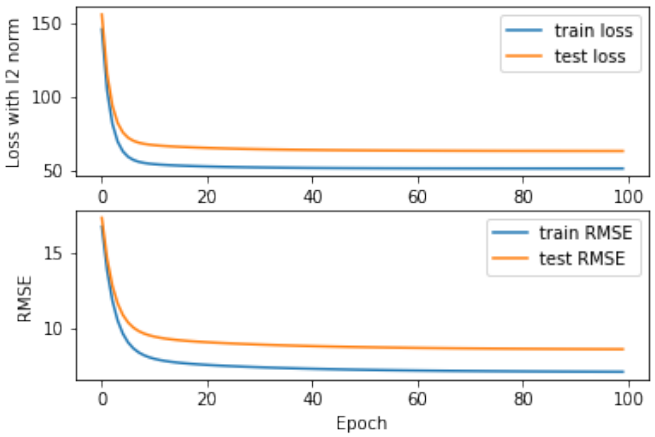

Fig. 4. Figure of 'eta': 0.05, 'lamda': 0.5

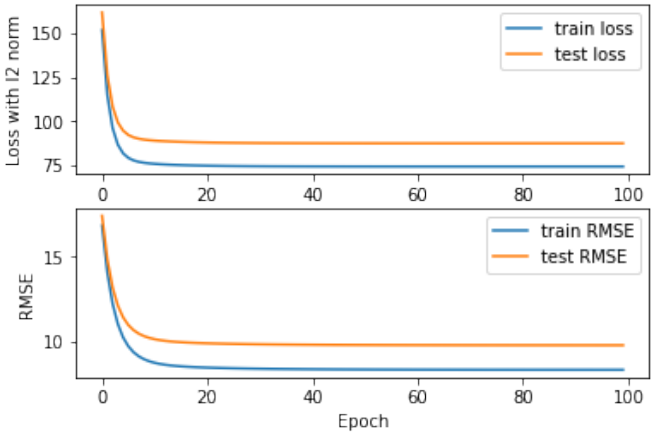

Fig. 5. Figure of 'eta': 0.05, 'lamda': 1

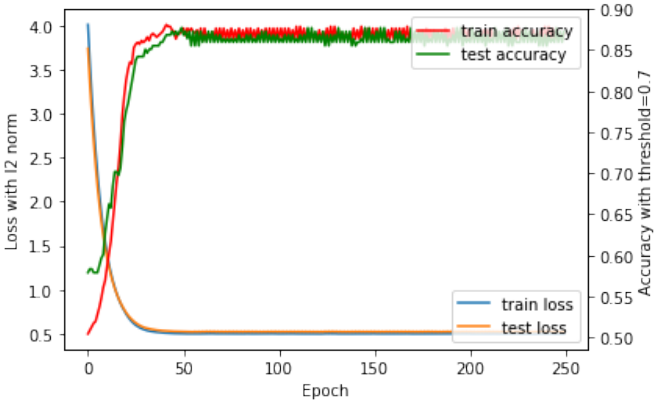

Fig. 6. Figure of 'C': 1, 'eta': 0.1, 'lamda': 0.5, 'threshold': 0.4

IV. Conclusion

This report shows two experiments about linear regression and linear classification, respectivly. Both experiments are conduct under small scale dataset. The experiments theroy, implementation are introduced and the detailed results are shown as tables and figures. Furthermore, parameters influence are tested in the experiment and the conclusion that gradient decent is a valid method to optimize both regression problem and classification problem can be drawn.

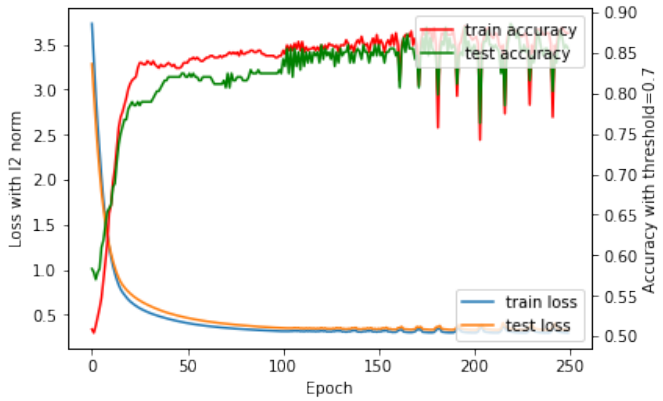

Fig. 7. Figure of 'C': 2, 'eta': 0.1, 'lamda': 0.1, 'threshold': 0.4

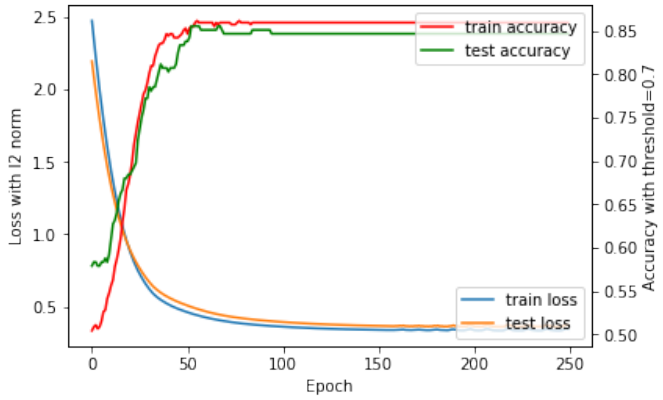

Fig. 8. Figure of 'C': 1, 'eta': 0.1, 'lamda': 0.1, 'threshold': 0.4

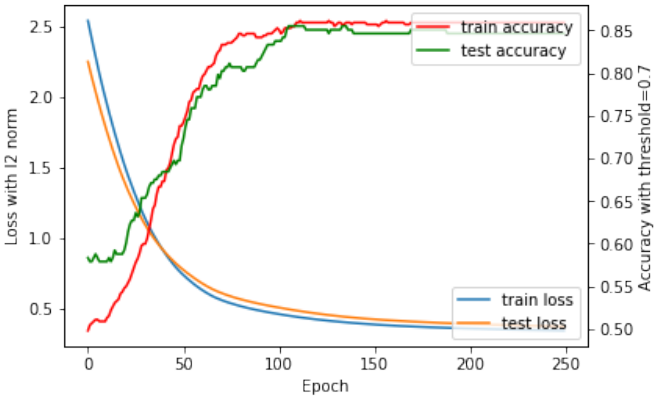

Fig. 10. Figure of 'C': 1, 'eta': 0.05, 'lamda': 0.1, 'threshold': 0.4

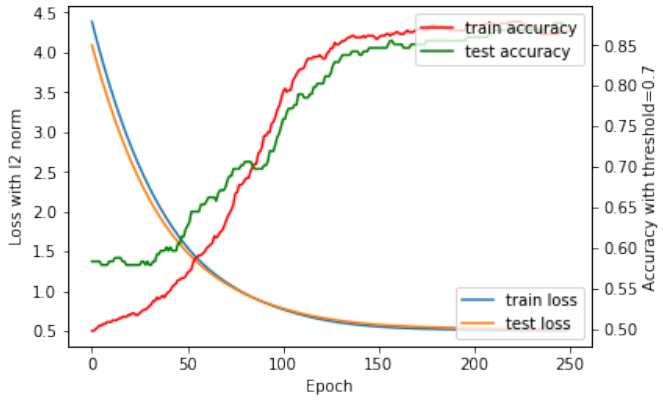

Fig. 9. Figure of 'C': 1, 'eta': 0.02, 'lamda': 0.5, 'threshold': 0.4