

**The Experiment Report of**

***Deep Learning***

**College Software College**

**Subject Software Engineering**

**Members**   **Qi Chen**

**Student ID 201720144924**

**E-mail 821788277@qq.com**

**Tutor**   **Jian Chen**

**Date submitted** **2017.12.7**

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

**2. Time: 2017-12-02**

**3. Reporter: Qi Chen**

**4. Purposes:**

1. Further understand the process of linear regression and gradient descent.
2. Conduct some experiments under small scale dataset to explore the performance of different algorithm.
3. Realize and analyze the process of optimization and adjusting parameters.

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

1. We use ‘Housing’ dataset in LIBSVM Data, which contains 506 samples with 13 features, for the experiments of Linear Regression.
2. We use ‘australian’ dataset in LIBSVM Data, which contains 690 samples with 14 features, for the experiments of Linear Classification.

**6. Experimental steps:**

1. Load the dataset and divide dataset into training set and validation set.
2. Initialize linear regression or SVM model parameters with normal distribution.
3. Define the loss function.
4. Compute the gradient with respect to the weight.
5. Using gradient descent to update the weight.
6. Repeat step (4) and (5) for several times until convergence.

**7. Code: (Double click the object to see the code)**

* Linear Regression and Gradient Descent



* Linear Classification and Gradient Descent



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

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

Initialize the parameters by zero.

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

1. Loss function:
   1. Linear Regression:
   2. Linear Classification (SVM):
2. Gradient with respect to the weight:
   1. Linear Regression:
   2. Linear Classification (SVM):

**11. Experimental results and curve:**

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

1. Linear regression

= 0.1, epoch = 100

1. Linear classification

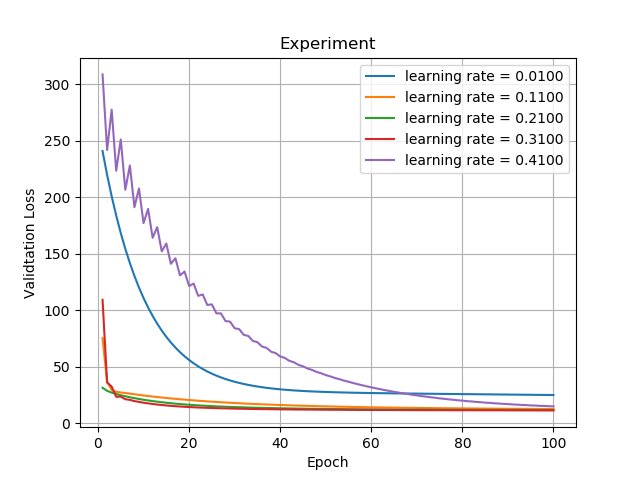
= 0.1, , epoch = 100

## Assessment Results (based on selected validation):

* Linear regression

1. Learning rate

In this section, we chose 5 different learning rates [0.01, 0.11, 0.21, 0.32, 0.41] to optimize the linear regression algorithm, the results are as the following:

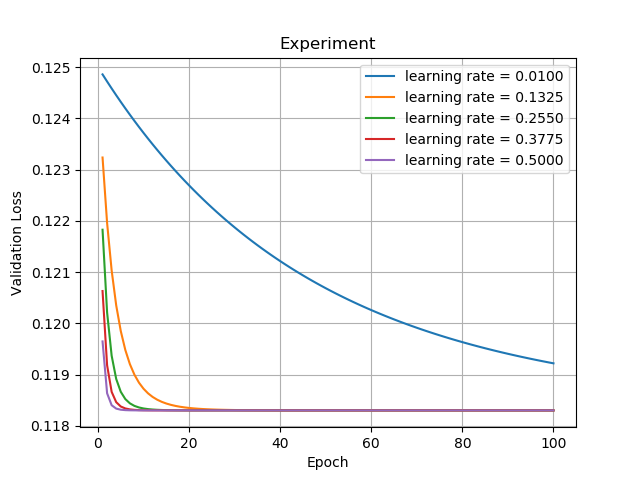
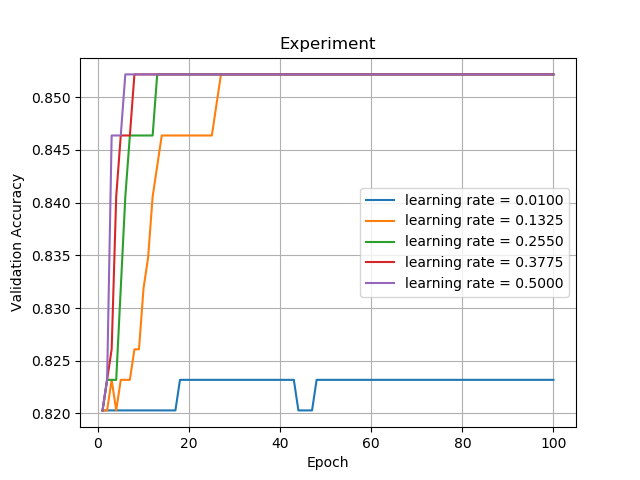


In this figure, we can obverse that during the learning rate increasing, the validation loss decrease progressively. However, when the learning rate is out of a boundary, the loss will be vibrating and can not achieve the best performance.

* Linear classification

1. Learning rate

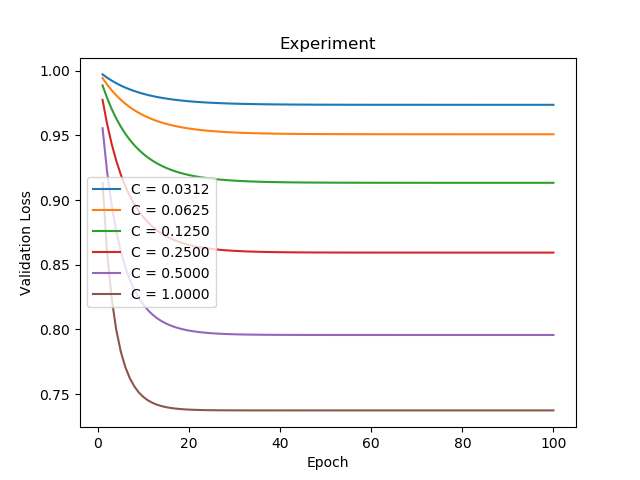
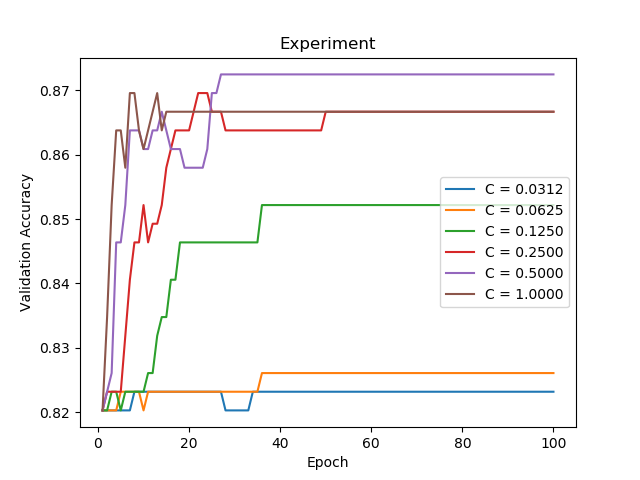
In this section, we chose 5 different learning rates [0.01, 0.1325, 0.2550, 0.3775, 0.5000] to optimize the linear classification algorithm, the results are as the following:



In this figure, we can obverse that during the learning rate increasing, the validation loss decrease progressively. Moreover, when the learning rate is increasing, the speed of convergence also improve.

1. Hyper-parameter C

In this section, we chose 6 different hyper-parameters C [] to optimize the linear classification algorithm, the results are as the following:



As we can see, in this figure, during the hyper-parameters C increasing, the validation loss decrease progressively, but the accuracy first increases and then decline when the parameter C become larger.

## Predicted Results (Best Results):

According to the experiments before, we chose the best hyper-parameters to optimize our model:

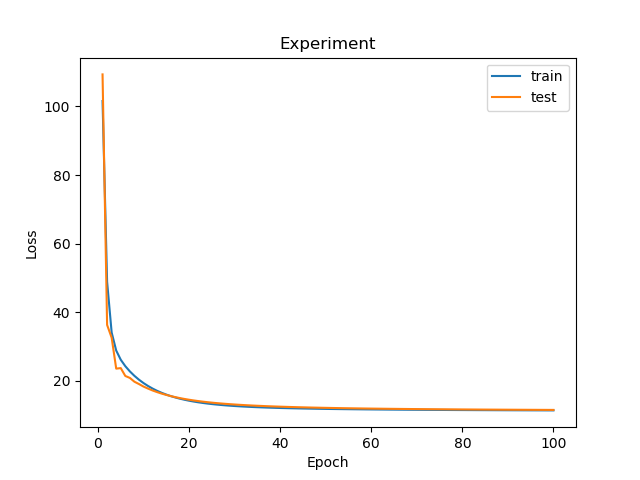
1. Linear regression

The chosen parameters are:

= 0.31, epoch = 100

The result of this parameters is:

Validation Loss = 11.504951



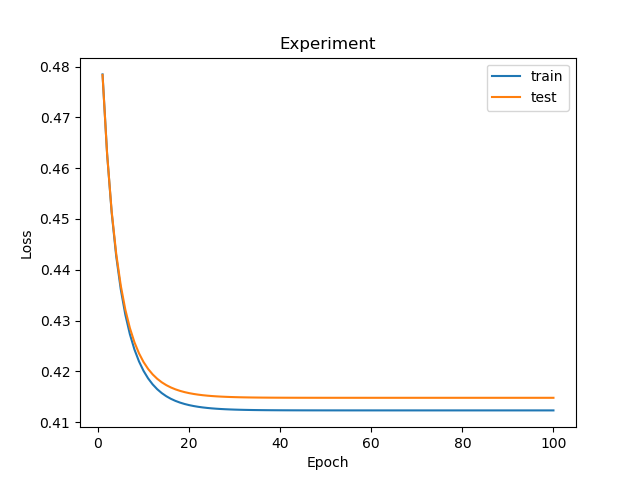
1. Linear classification

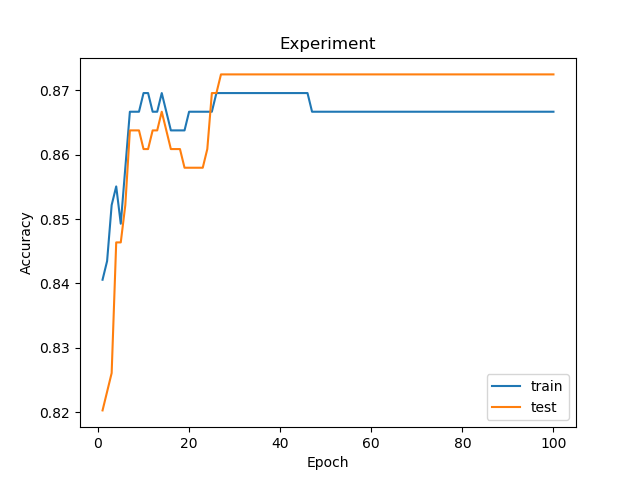
The chosen parameters are:

= 0.1, , epoch = 100

The result of this parameters is:

Accuracy = 0.87246





**12. Results analysis:**

For the linear regression and linear classification, the proper value of learning rate is significant. When value is too large or too small, the loss will not achieve the minimum. Specially, for the linear classification (i.e. support vector machine in this experiment), we have to adjust another hyper-parameter C to balance the loss and the penalty in object function.

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

**Similarities：**Both linear regression and linear classification are linear model. The purpose of these algorithm is to learn a mapping between input domain and output domain.

**Differences:** Linear regression and linear classification use different loss function. On the other hand, for the linear regression, we learning a model to simulate the mapping between input X and output y. But for the linear classification task, we want to find a hyper plane to separate the different target.

**14. Summary:**

Both linear regression and linear classification are linear model, they are easy to train but have a limited capacity to some difficult tasks, which can not be simulate or separate by the naïve modes.