

**The Experiment Report of**

***Deep Learning***

**College Software College**

**Subject Software Engineering**

**Members**  **Yuxuan Gao**

**Student ID 201720145075**

**E-mail cdsnow2017@163.com**

**Tutor**   **Mingkui Tan**

**Date submitted** **2017.12.8**

1. **Topic**

Linear Regression, Linear Classification and Gradient Descent

1. **Time**

December 8, 2017

1. **Reporter**

Yuxuan Gao

1. **Purposes**
2. Further understanding the principle of linear regression and gradient descent
3. Practice on a small set of data
4. The process of optimizing and adjusting the parameters
5. **Data sets and data analysis**

The experiment use two datasets. Linear regression use Housing dataset and linear classification use australian dataset.

The Housing dataset concerns housing values in suburbs of Boston, it has 13 continuous attributes and 1 binary-valued attribute, without missing attribute values.

The australian dataset has 10 continuous attributes and 4 binary-valued attribute without missing values, there are 2 classes to be classified .

1. **Experimental steps**

**6.1 Linear Regression and Gradient Descent**

* 1. Load the experiment data
  2. Use train\_test\_split function to devide dataset，33% for validation and others for training
  3. Initialize linear model parameters into zero
  4. Choose loss function and compute derivation **L**
  5. Calculate gradient and update **W** several times and record **Lost** of training set and testing set
  6. Adjustη, epoch etc, draw corresponding graph

**6.2 Linear Classification and Gradient Descent**

1. ~2) same as Regression
2. Initialize linear model parameters into normal distribution
3. ~6) same as Regression, adding a parameterλ
4. **Code**

See the ipynb file

1. **Selection of validation**

Linear regression and linear classification both use Hold-Out Method, 33% for validation and others for training

1. **The initialization method of model parameters**
2. For linear regression, initialize linear model parameters into zero
3. For linear classification, initialize linear model parameters into normal distribution and multiplied by 0.01

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

**10.1** **Linear Regression**

1. **loss function**

X is the matrix of dataset; Y is the matrix of values to be predi-cted; W is the matrix of weights; N is the number of dataset.

1. **gradient/** **derivative**

**10.2** **Linear Classification**

1. **loss function**
2. **gradient/** **derivative**
3. **Experimental results and curve**

**11.1 Linear Regression**

1. Hyper-parameter selection (η, epoch, etc.)
2. η selection

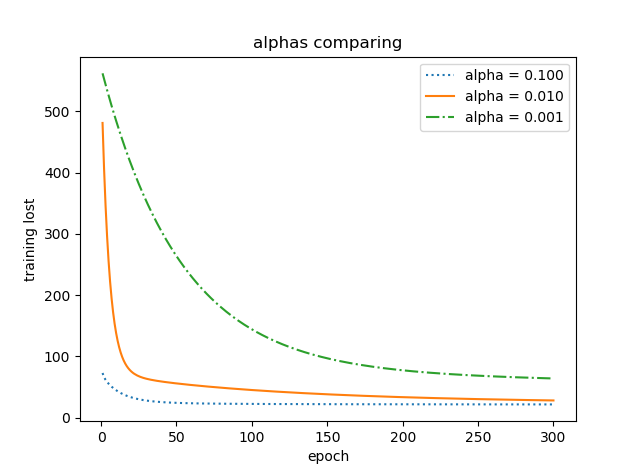


Fig 11.1 alphas comparing

From the figure we can see η = 0.1 is better.

1. epoch selection

From the Fig 11.1 we can find that alpha = 0.1 is better, so we choose alpha = 0.1 to compare training lost and testing lost as below:

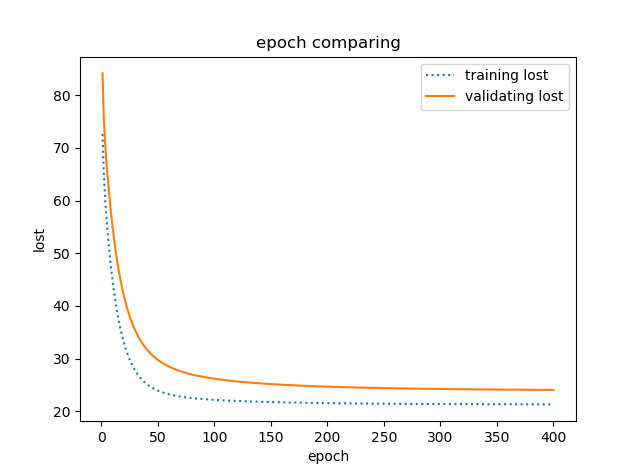


Fig 11.2 alphas comparing

From the figure we can find out when epoch is 150, the training lost and validating lost are closed to convergence. So we choose epoch = 150.

1. Assessment Results (based on selected validation)

When alpha = 0.1, epoch = 150:

|  |  |
| --- | --- |
| dataset | variance between predict result and Training set |
| Training set | 21.073 |
| Validating set | 27.154 |

1. Lost curve

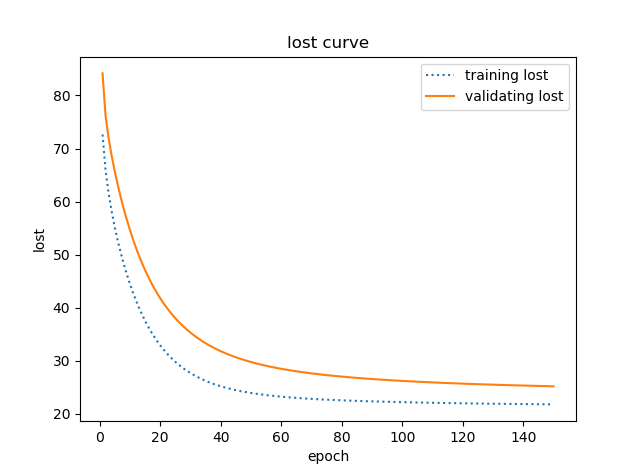


Fig 11.3 lost curve

**11.2 Linear Classification**

1. Hyper-parameter selection (η, epoch, etc.)
   1. η selection



Fig 11.4 alphas comparing of training lost



Fig 11.4 alphas comparing of training accuracy

From the figure we can see η = 0.01 is better.

* 1. epoch selection

From the Fig 11.1 we can find that alpha = 0.01 is better, so we choose alpha = 0.01 to compare training lost and testing lost as below:



Fig 11.5 alphas comparing

From the figure we can find out when epoch is 200, the training lost and testing lost are closed to convergence. So we choose epoch = 200.

* 1. Regular parameterλselection



Fig 11.6 **λ**comparing of training lost



Fig 11.7 **λ**comparing of training accuracy

From the figure we can see**λ** = 0.01 is better.

1. Assessment Results (based on selected validation)

When alpha = 0.01, reg = 0.01 epoch = 200:

|  |  |
| --- | --- |
| dataset | Accuracy(%) |
| Training set | 86.3 |
| Validation set | 84.7 |

1. Lost curve

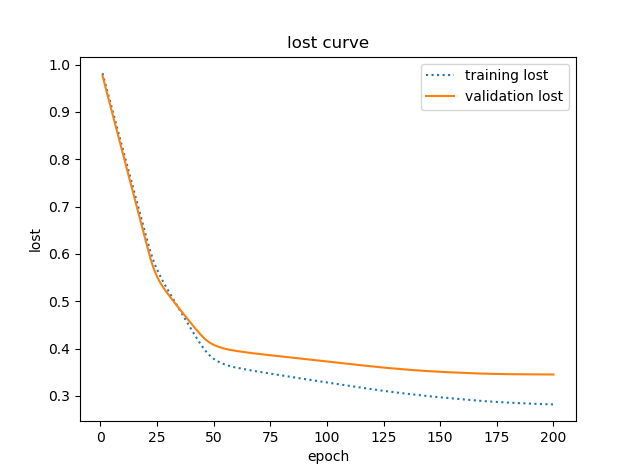


Fig 11.8 lost curve

**12. Results analysis**

Through the above experiments, we can find that as long as the lost function can be found and lost function is convex, we can use gradient descent method for seeking optimal solution, We can not only use linear regression forecasting a certain expectations, but also linear classification can be used for classification.

Hyper-parameter selection is an important factor of model’s quality. Excessive learning rate leads to the failure of the optimal solution of the loss curve, if learning rate is too small, the rate of convergence will be slow.

In general, the more iterations, the better the model. But a proper number of iterations is necessary, otherwise a waste of computing resources will be generated.

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

Supervised learning problems are categorized into "regression" and "classification" problems. In a regression problem, we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function. In a classification problem, we are instead trying to predict results in a discrete output. In other words, we are trying to map input variables into discrete categories. Linear regression and linear classification both use gradient descent to approximate optimal solution

**14. Summary**

From this experiment, I understand the theoretical basis and specific experimental method of linear regression and linear classification. The characteristics of gradient descent method are studied through experiment. By adjusting the values of different parameters, I understood the effects of hyper-parameter on the model.

Thanks to the teacher and assistant for giving us such an excellent experiment.