华南理工大學

South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Wu kun

Supervisor:
Mingkui Tan or Qingyao Wu

Student ID：201721045480

Grade:
Undergraduate or Graduate

December 14.2017

Linear Regression, Linear Classification and Gradient Descent

Abstract—Linear regression is a basic and commonly used type of predictive analysis. The overall idea of regression is to examine two things: does a set of predictor variables do a good job in predicting an outcome variable, This article is to introduce the algorithm of Linear Regression and Gradient Descent

I. Introduction

To establish notation for future use, we'll use x(i) to denote the "input" variables (living area in this example), also called input features, and y(i) to denote the "output" or target variable that we are trying to predict (price). A pair (x(i),y(i)) is called a training example, and the dataset that we'll be using to learn—a list of m training examples (x(i),y(i));i=1,...,m—is called a training set. Note that the superscript "(i)" in the notation is simply an index into the training set, and has nothing to do with exponentiation.

II. Methods and Theory

We can measure the accuracy of our hypothesis function by using a cost function. This takes an average difference (actually a fancier version of an average) of all the results of the hypothesis with inputs from x's and the actual output y's.

$$J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2$$

To break it apart, it is 12 $\bar{x}$ where $\bar{x}$ is the mean of the squares of hθ(xi)−yi , or the difference between the predicted value and the actual value.

repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0,\theta_1)$$

}

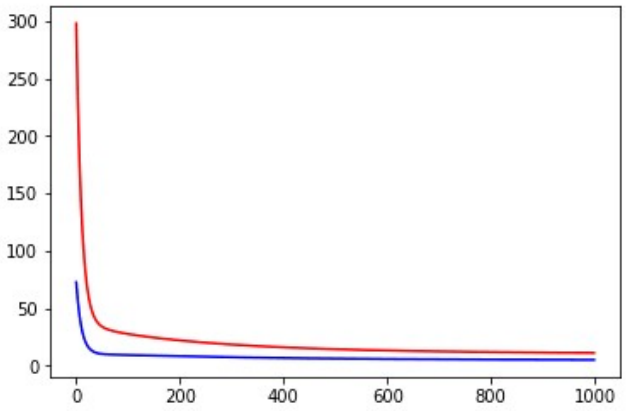
At each iteration j, one should simultaneously update the parameters θ1,θ2,...,θn. Updating a specific parameter prior to calculating another one on the j(th) iteration would yield to a wrong implementation.

The point of all this is that if we start with a guess for our hypothesis a The point of all this is that if we start with a guess for our hypothesis and then repeatedly apply these gradient descent equations, our hypothesis will become more and more accurate.

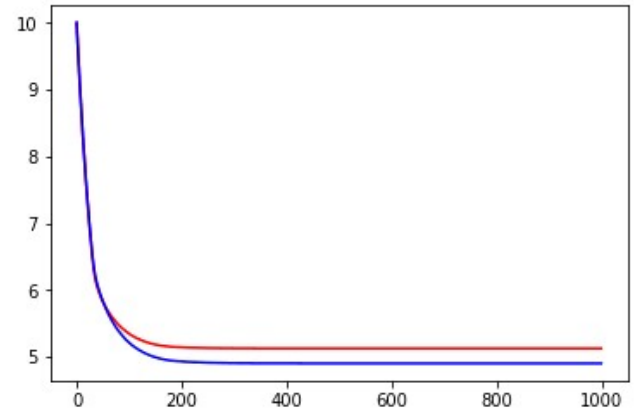
So, this is simply gradient descent on the original cost function J. And then repeatedly apply these gradient descent equations, our hypothesis will become more and more accurate.

III. Experiment

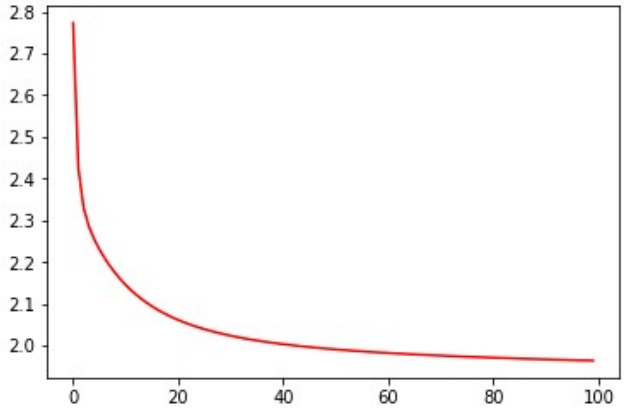
Split the dataset to train dataset and test dataset then use gradient descent algorithm to minimize the loss value of a zero initialization linear module

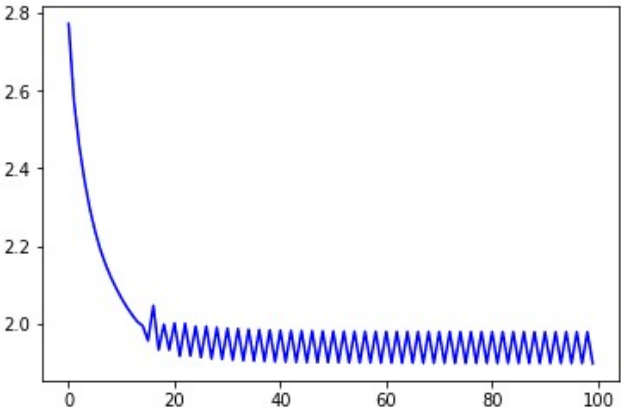


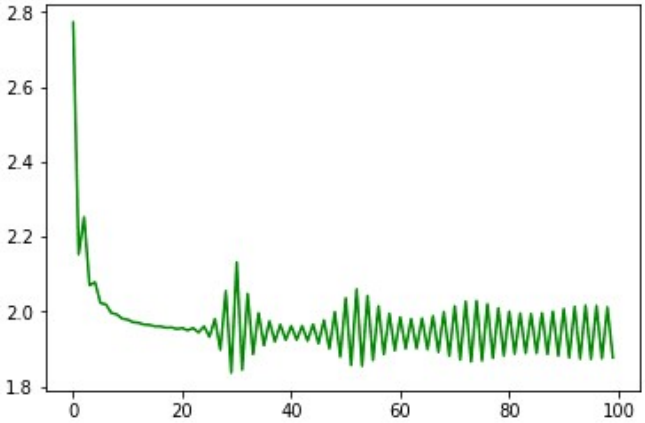
The L_train and L_validation was decreased to almost 0 after 1000 iterations of gradient descent.



Linear SVM module's L_train and L_validation was changed during iterations of gradient descent







These 3 image above is different optimized gradient descent approach's loss value's change during 100 iterations. The red one represente NAG and the blue one is Adadelta's cruve, the third is RMSprop's.

IV. CONCLUSION

In these experiment we can see that use cost function and gradient descent approach we can train module which is able to classify different kind of data, And use optimized gradient dexcent approach will speed up the training process.