The Experiment Report of Machine Learning



December 9, 2017

Grade:

Undergraduate

Student ID：201530611289

Supervisor:

Mingkui Tan

Author:

Xingyu Chen

**SUBJECT:**SOFTWARE ENGINEERING

**SCHOOL:** SCHOOL OF SOFTWARE ENGINEERING

[[1]](#footnote-0)

**Logistic Regression, Linear Classification and Stochastic Gradient Descent**

Abstract—In this experiment we compare the difference between different ways of stochastic gradient descent.We also compare the differences and relationships between logistic regression and linear classification.From this experiment we can get a better understanding of the principles of SVM and the practice on larger data.

# INTRODUCTION

Gradient decent(batch gradient decent) and stochastic gradient decent is two major ways to apply on algorithms that use gradient decent.The main difference of this two method is that they use different size of training examples to compute gradient .Batch gradient decent usually use all the training examples while stochastic gradient decent only use one or several examples(called minibatch stochastic gradient decent) to compute gradient and update parameters. The motivation of stochastic gradient decent is that sometimes the data set is too large to compute gradient or it will cost a lot of resources to compute ,in such case we can’t simply compute the gradient over the whole training set ,so we randomly chose some examples ,which are also obey the origin distribution .By using this technique we can deal with a great amount of data and just use a small amount of resource to make the gradient decent algorithm work .

Despite the way of selecting examples to compute gradient ,there are also many ways to update parameters.In this experiment ,we compare four different ways of stochastic gradient decent ,namely NAG，RMSProp，AdaDelta and Adam.These four ways are all variation of stochastic gradient decent .

Linear Classification based on SVM and logic regression are the two models we use to apply gradient decent .Logic regression is actually a classification model and the main idea of it is to find a best line the separate the different class of data such that the loss can be minimized. The SVM model is similar to logic regression but SVM can have a better performance by changing the linear non-separable problems in low dimension to the linear separable problem in high dimension .

In this experiment we implements the two models mentioned above and use different gradient decent technique to compare the performance of them .

# METHODS AND THEORY

*A .Linear Regression*

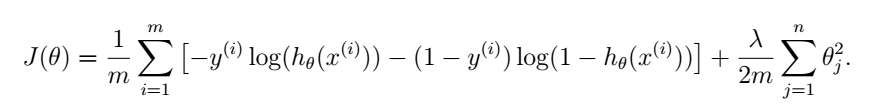
In the linear regression , we implement a binary classifier ,which map the positive example as 1 and negative example as 0. Based on this ,we need a function to map real number into [-1,1] ,so we chose sigmoid function



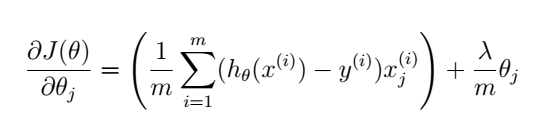
And then we define our hypothesis as



The  denotes the parameter to learn in the model and the denotes training examples .b is the bias term .With the hypothesis function ,we then define the loss function as



The term at the tail of formula is the regularization term and λ is the regularization parameter .m denotes the number of training examples and j denotes the number of features .For intuition ,the loss function will calculate the loss of examples that are classified to the wrong class .The next step is to compute the gradient of the loss function ,the result is



Note that in the actual implement we merge b to  by adding a dimension to  and add a column to training examples ,so the derivative result above will not include b .With all these formula we can use gradient decent to implement linear regression .

*B . Linear Classification*

In this section ,we illustrate the mathematics in linear classification .We use a different way to represent positive examples and negative examples ,that is ,use 1 to denote positive examples and -1 to denote negative examples .

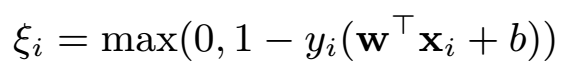
First we define our hypothesis as

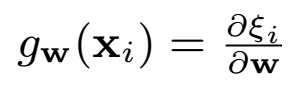


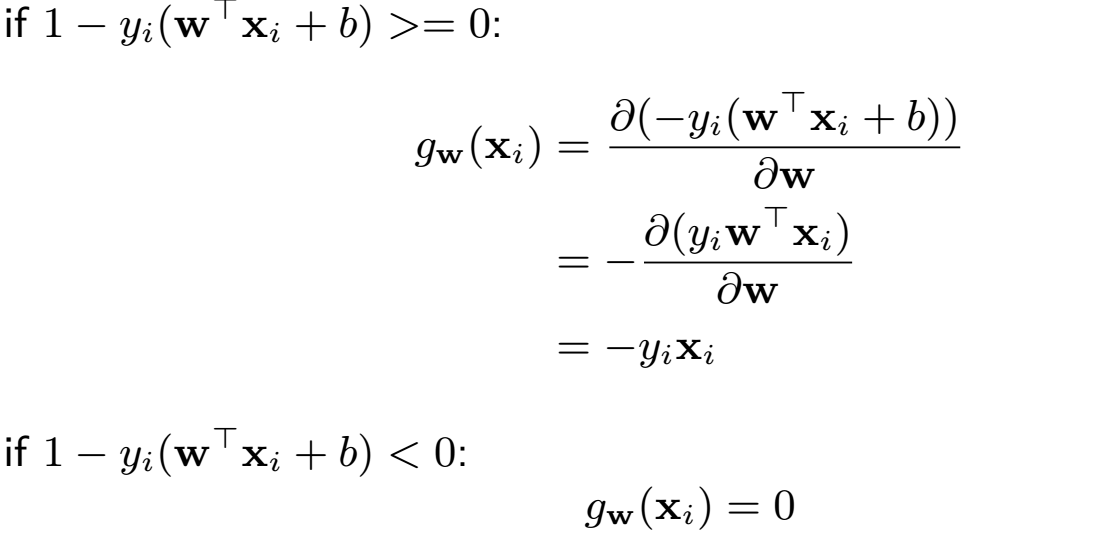
Given a training example ,we predict the label as 1 when  and as 0 when .Then we define the objective function (loss function ) as



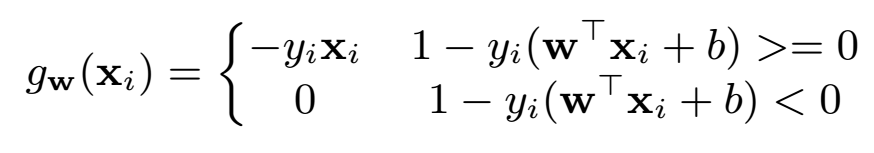
In order to apply gradient decent ,we compute the gradient of the objective function .We denote



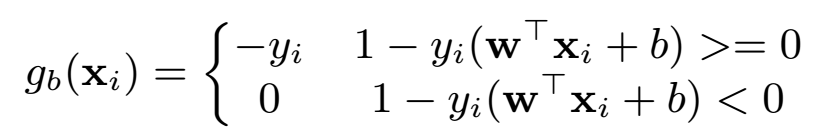




So we have



Then we denote  ,we have the derivative

 To summarize ,we have



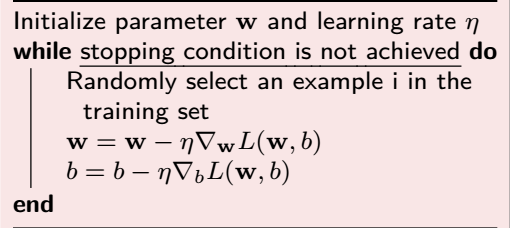


Now we can apply gradient decent on linear classification .

*C . Batch Gradient decent and Stochastic Gradient Decent*

In the section we describe how to implement Batch Gradient Decent and Stochastic Gradient Decent .

To illustrate the process of stochastic gradient decent ,we use linear classification model as example .The whole progress are list as figure 1.



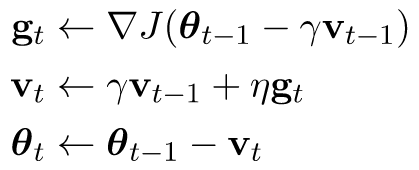
*Figure 1*

The most important thing is that we chose one example to calculate gradient in each iteration until the loss function is converged (namely the stopping condition ) .To contrast ,batch gradient decent uses all the examples to compute gradient and minibatch stochastic gradient decent uses some examples whose number is defined as  .

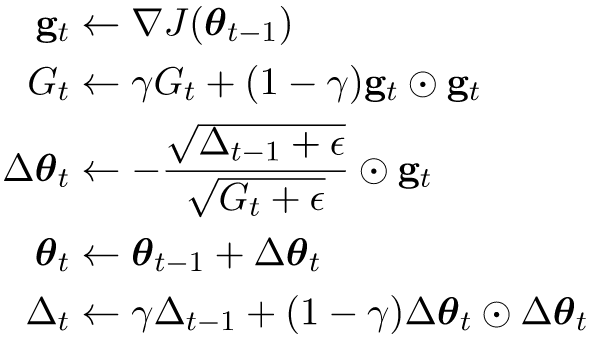
*D .NAG，RMSProp，AdaDelta and Adam*

The four variation of stochastic all have their own motivation and mathematics details ,and in this section we are not going to describe how to derive thesis optimized version of stochastic gradient decent but just list the update rules of each technique .

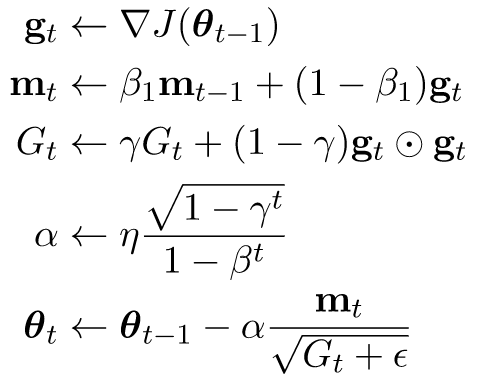
Figure 2 to figure 5 illustrate the update rules of each technique .



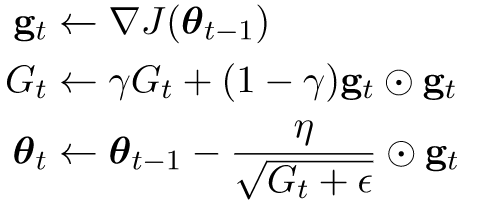
*Figure 2 :NAG update rules*

**

*Figure 3 :AdaDelta update rules*

**

*Figure 4 :Adam update rules*

**

*Figure 5 :RMSProp update rules*

Note that the  in the formula denotes learning rate and ,,are all parameters set by user .

# Experiment

In this part we will describe how we do our experiment in details .We have two model and we apply four ways of stochastic gradient on each model .In order to compare the different methods ,we plot the loss with the number of iterations ,to see how the loss function converged .

To be more specific ,we first load the data and initialize the parameters , then in each iteration we randomly select a batch of examples ,compute gradient on it and use the result to update the parameter in four ways .We record the value of loss function in every iteration and plot it in the end of algorithm .

We initialize the parameter in random values in [0,1] ,the parameters we use are list in table 1 and table 2，the result of our experiment is illustrate in figure 6 to figure 9.

# conclusion

1. [↑](#footnote-ref-0)