华南理工大学

South China University of Technology

The Experiment Report of

Machine Learning

College Software College

Subject Software Engineering

Members Lin Guoshi

Student ID 201530612217

E-mail 1509282534@qq.com

Tutor Tan Mingkui

Date submitted 2017.12 . 15

1. Topic:

Logistic Regression, Linear Classification and Stochastic Gradient Descent

2. Time: 2017.12 .15

3. Reporter: Lin Guoshi

4. Purposes:

1.Compare and understand the difference between gradient descent and stochastic gradient descent.

2.Compare and understand the differences and relationships between Logistic regression and linear classification.

3.Further understand the principles of SVM and practice on larger data.

5. Data sets and data analysis:

Experiment uses a9a of LIBSVM Data, including 32561/16281(testing) samples and each sample has 123/123 (testing) features. Please download the training set and validation set.

Compare to the last experiment, the dataset of this time is much larger, so the gradient descent might be very slow, so i choose to set the batch size at a small number for reducing the running time, but actually, the result is still reasonable.

6. Experimental steps:

Logistic Regression and Stochastic Gradient Descent

1.Load the training set and validation set.

2.Initalize logistic regression model parameters, you can consider initalizing zeros, random numbers or normal distribution.

3.Select the loss function and calculate its derivation, find more detail in PPT.

4.Calculate gradient toward loss function from partial samples.

5.Update model parameters using different optimized methods(NAG，RMSProp，AdaDelta and Adam).

6.Select the appropriate threshold, mark the sample whose predict scores greater than the threshold as positive, on the contrary as negative. Predict under validation set and get the different optimized method loss LossNAG，LossRMSProp，LossAdaDelta and LossAdam.

7.Repeate step 4 to 6 for several times, and drawing graph of LossNAG，LossRMSProp，LossAdaDelta and LossAdam and with the number of iterations.

Linear Classification and Stochastic Gradient Descent

1.Load the training set and validation set.

2.Initalize SVM model parameters, you can consider initalizing zeros, random

numbers or normal distribution.

3.Select the loss function and calculate its derivation, find more detail in PPT.

4.Calculate gradient toward loss function from partial samples.

5.Update model parameters using different optimized methods(NAG，RMSProp，AdaDelta and Adam).

6.Select the appropriate threshold, mark the sample whose predict scores greater than the threshold as positive, on the contrary as negative. Predict under validation set and get the different optimized method loss LossNAG，LossRMSProp，LossAdaDelta and LossAdam.

7.Repeate step 4 to 6 for several times, and drawing graph of LossNAG，LossRMSProp，LossAdaDelta and LossAdam.

and with the number of iterations.

7. Code:

(Fill in the contents of 8-11 respectively for logistic regression and linear classification)

8. The initialization method of model parameters:

Just set the parameters all zeros for initialization.

9. The selected loss function and its derivatives:

Logistic regression

```
def logLoss(x,y,w):
    loss=0
    numOfsamples=x.shape[0]
    for i in range(numOfsamples):
        loss=loss-y[i]*math.log(h(x,w,i))-(1-y[i])*math.log(1-h(x,w,i))
    return loss/numOfsamples
```

For the logistic regression, i choose the log-likelihood loss function, by doing some experiments, i prefer this formulation because i found that the loss function can be decreased faster.

SVM

```
def HingeLoss(x,y,w):
    item1 = dot(w.T,w)*0.5
    item2 = 0
    for i in range(x.shape[0]):
        item2 = item2+max(0,1-y[i]*dot(x[i],w))
    return item1+item2
```

In SVM optimization problem, i choose the hinge loss as its objective function for its simple formulation.

10. Experimental results and curve:(Fill in this content for various methods of gradient descent respectively)

Hyper-parameter selection:

```
lr = 0.1
iteration = 100
batchsize = 400
p = 0.9
epsilon = 1e-6
beta = 0.9
```

At the first stage of the experiments, i choose the iteration and the batch size at a very small number for just knowing whether the hyper-parameter is good or not

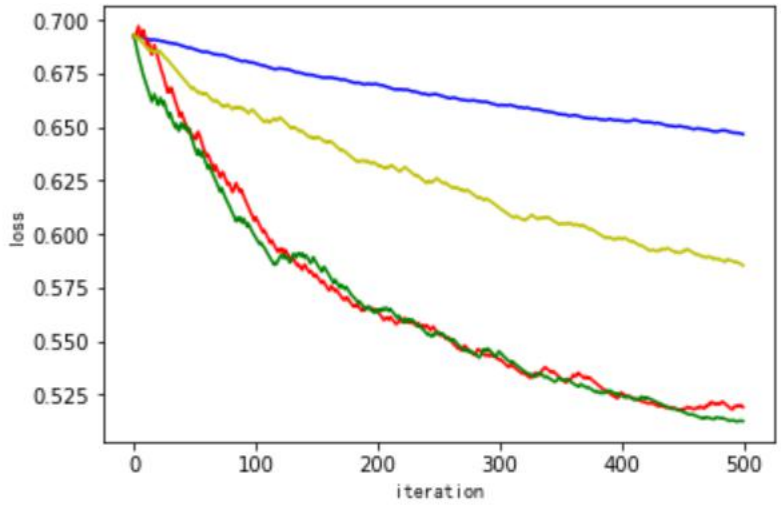
Finally, i would set the iteration larger to see the results of the different optimization methods.

Predicted Results (Best Results):

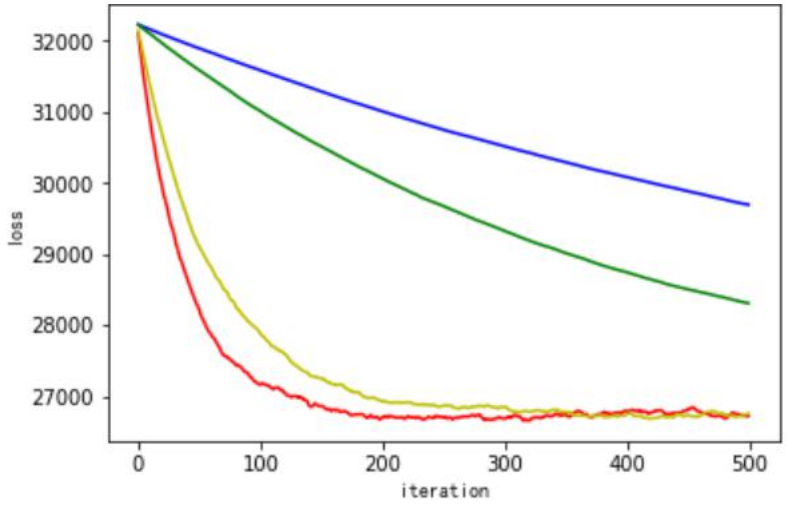
Loss curve:

11. Results analysis:

Lost curve for logistic regression



Lost curve for SVM



Blue for NAG, red for RMSProp, green for AdaDelta, yellow for Adam

Result analysis:

In this experiment, for make a comparison, i do try to keep the main hyper-parameter same for all the optimization method to compare the differences between them.In both of the two experiments, the result of the NAG is not that good in small number of iterations, the another reason may be that my batch size is small too. Obviously, for different model, different methods have different performances. But all that methods can make the loss function decreased after some iterations. And all the lost curves are not very smooth, maybe for the reason that the batch size is not large enough.

12. Similarities and differences between logistic regression and linear classification：

Similarities:

Linear regression and linear classification can work well on linear datasets and the two models are easy to implement but effective.

Differences:

Linear regression is choose to deal with the regression problem that predicts a continuous value while linear classification is choose to classify the samples into two or more classes.

13. Summary:

1. Different optimizer methods have different performances
2. But not a single method can always be the best one in all models
3. Gradient descent in mini-batch may lead to some concussion in the lost curve but it does not matter that the loss function decreases.