South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
ZiWen Zhao

Supervisor:
Mingkui Tan

Student ID:
201530613764

Grade:
Undergraduate or Graduate

December 9, 2017

Linear Regression, Linear Classification and Gradient Descent

Abstract—

I. Introduction

Adam
Adam is essentially a RMSprop with momentum term, which dynamically adjusts the learning rate of each parameter by using the first order moment estimation of the gradient and the two order moment estimation. The main advantage of Adam is that after the offset correction, each iterative learning rate has a definite range, which makes the parameters more stable.
Adagrad
Adagrad is a kind of optimization algorithm based on gradient, it can be different for each parameter of the adaptive learning rate, the sparse feature, get a big update on learning, non sparse feature, get smaller learning updates, so the optimization algorithm for sparse feature data.
RMSprop
In fact, RMSprop is intermediate form of Adadelta, is to reduce the learning rate of Adagrad in the fast decay
NAG
NAG and in the calculation of the gradient parameters, the loss function is subtracted from the momentum term is calculated J (theta theta - gamma V T - 1), the estimated parameters of the next position.
Adadelta
There are three problems in the Adagrad algorithm
The learning rate is monotonous and decreasing, and the learning rate is very small in the later period of training.
It needs to manually set up a global initial learning rate
When the XT is updated, the units of the left and right sides are different
Adadelta puts forward a more beautiful solution to the above three problems.

II. Methods and Theory

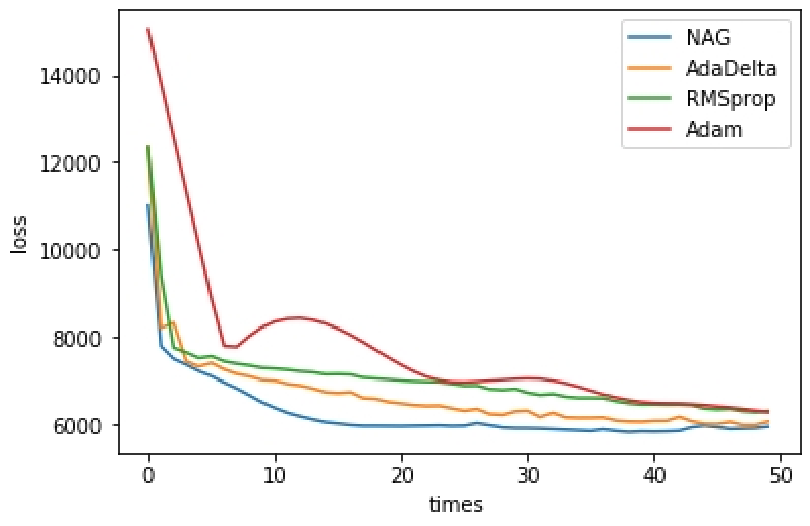
Adam
$mt=\beta 1mt-1+(1-\beta 1)gt$
$vt=\beta 2vt-1+(1-beta2)g2t$
MT and VT are gradient weighted mean and weighted weighted partial difference respectively. They are initially 0 vectors. The authors of Adam find that they tend to have 0 vectors (close to 0 vectors), especially when the attenuation factor (decay rate) is 1, and the beta 2 is close to 1. In order to improve this problem, MT and VT deviation correction (bias-corrected):
$mt\hat{}=mt1-betat1$
$vt\hat{}=vt1-betat2$ $\theta t+1=\theta t-\eta vt\hat{}-- \surd +\epsilon mt\hat{}$

RMSprop
$E[g2]t=\gamma E[g2]t-1+(1-\gamma)g2t$
θ $t+1=\theta t-\eta E[g2]t+\epsilon-------- \surd \odot gt$

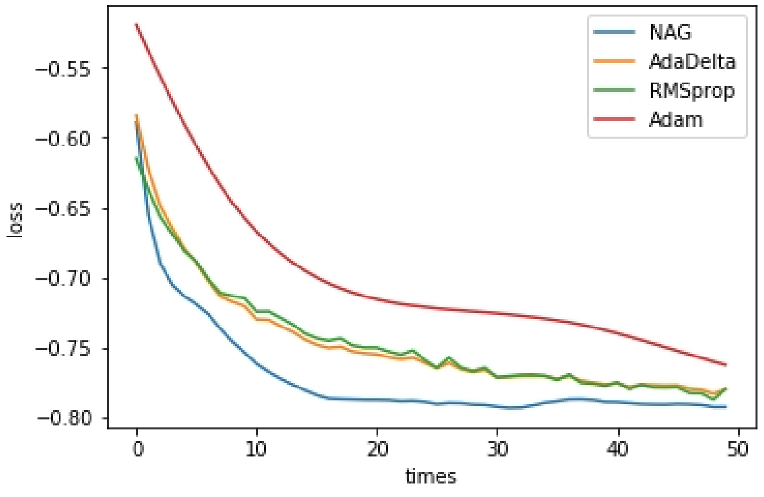
NAG
$vt=\gamma vt-1+\eta\cdot\nabla\theta J(\theta-\gamma vt-1)$
$\theta=\theta-vt$

III. Experiment

#NAG 实现



```
ganma = 0.9
aita = 0.001
wnag,vnag = NAG(X_trainset, y_trainset, wnag, vnag, ganma, aita)

nag_train.append(hingeloss(X_trainset, y_trainset, wnag))
nag_test.append(hingeloss(X_test, y_test, wnag))

#RMSProb 实现
ganma = 0.9
aita = 0.01
wrms,Grms = RMSprob(X_trainset, y_trainset, wrms, Grms, ganma, aita, e)
rms_train.append(hingeloss(X_trainset, y_trainset, wrms))
rms_test.append(hingeloss(X_test, y_test, wrms))


ganma=0.9
aita=0.001

wada,Gada,Tada = AdaDelta(X_trainset, y_trainset, wada, Gada, ganma, Tada, aita, e)
ada_train.append(hingeloss(X_trainset, y_trainset, wada))
ada_test.append(hingeloss(X_test, y_test, wada))
```

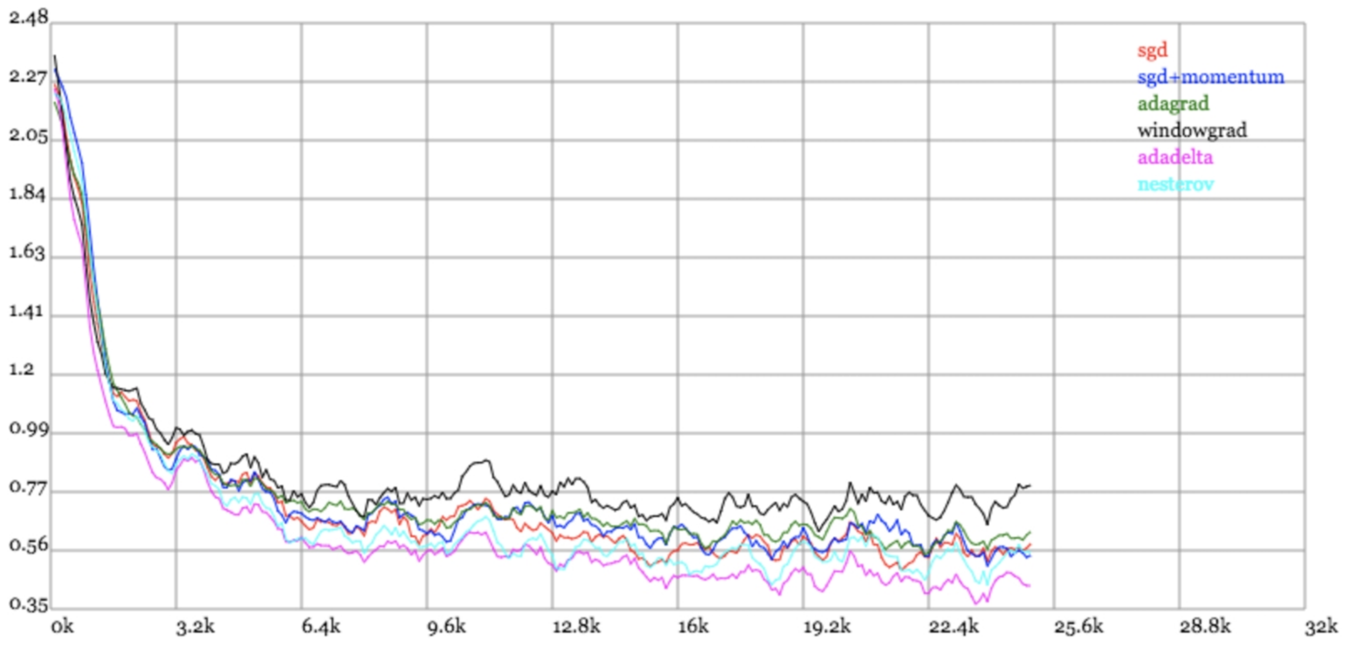


```
#Adam 实现
ganma = 0.9
aita = 0.01
```

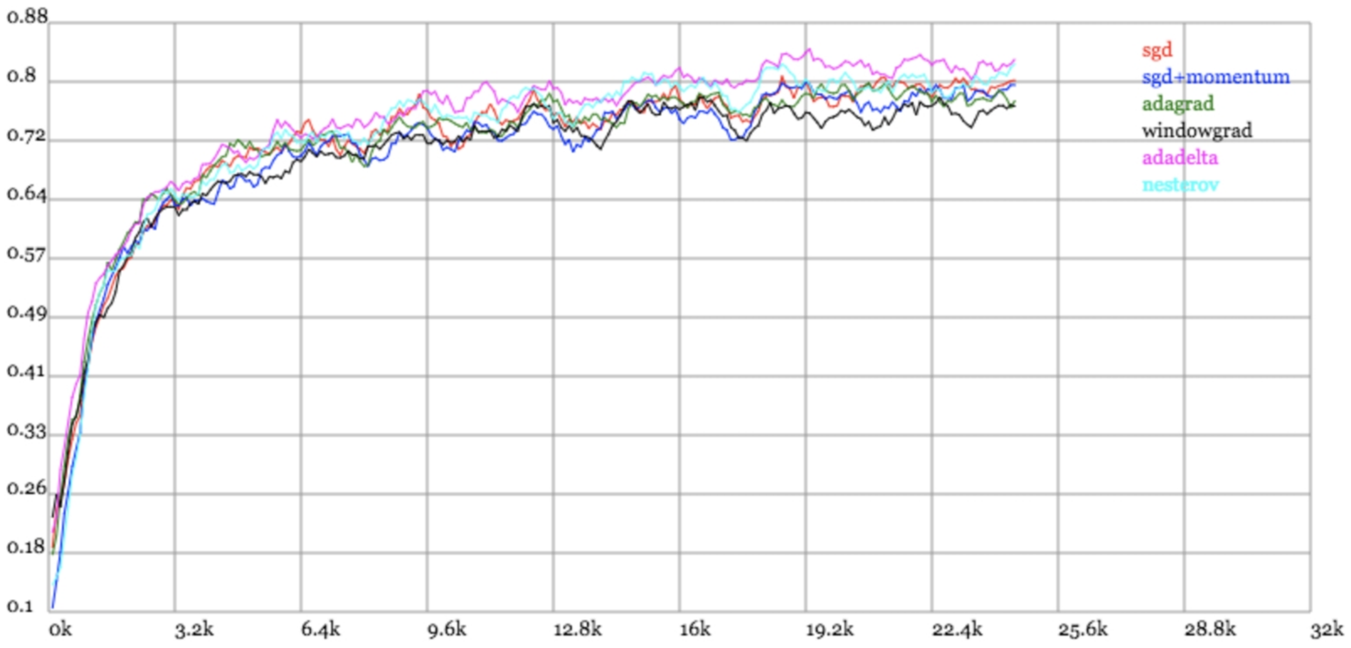
```
    wadam,Gadam,madam=Adam(X_trainset, y_trainset, wadam, Gadam, madam, ganma, i+1, aita, belta)
    adam_train.append(hingeloss(X_trainset, y_trainset, wadam))
    adam_test.append(hingeloss(X_test, y_test, wadam))


  ganma=0.9
    aita=0.001
    wnag,vnag = NAG(X_trainset, y_trainset, wnag, vnag, ganma, aita)
    nag_train.append(cost(X_trainset, y_trainset, wnag))
    nag_test.append(cost(X_test, y_test, wnag))


   #RMSProb 实现
    ganma = 0.9
    aita = 0.02
    wrms, Grms = RMSprob(X_trainset, y_trainset, wrms, Grms, ganma, aita, e)
    rms_train.append(cost(X_trainset, y_trainset, wrms))
    rms_test.append(cost(X_test, y_test, wrms))

    #Adadelta
    ganma = 0.95
    etha = 0.01
    wada, Gada, Tada = AdaDelta(X_trainset, y_trainset, wada, Gada, ganma, Tada, e)
    ada_train.append(cost(X_trainset, y_trainset, wada))
    ada_test.append(cost(X_test, y_test, wada))

    #Adam 实现

    ganma = 0.9
    aita = 0.01
    wadam,Gadam,madam=Adam(X_trainset, y_trainset, wadam, Gadam, madam, ganma, i+1, aita, belta)
    adam_train.append(cost(X_trainset, y_trainset, wadam))
    adam_test.append(cost(X_test, y_test, wadam))
```

IV. CONCLUSION

Karpathy makes a comparison of the performance of these methods on the MNIST, and the conclusion is:
Adagrad is more stable than SGD and momentum, that is, there is no need to adjust the parameters. The refined SGD and momentum series methods are better than adagrad in both convergence speed and precision. Under the fine tuning parameters, the general Nesterov is superior to the momentum than the SGD. Adagrad, on the one hand, does not need to adjust the parameters, on the other hand, its performance is better than other methods.
The experimental results are as follows:



Testing Accuracy vs. Number of examples seen


0k
3.2k
6.4k
9.6k
12.8k
16k
19.2k
22.4k
25.6k
28.8k
32k
sgd
sgd+momentum
adagrad
windowgrad
adadelta
nesterov