华南理工大学

South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:Xianzhe Wu
Shoukai Xu and Yaofu Chen

Supervisor:
Mingkui Tan

Student ID：201530613078
201530611111 and 201530600000

Grade:Undergraduate
Undergraduate or Graduate

December 9, 2017

Experimental Study on Stochastic Gradient Descent for Solving Classification Problems

Abstract—
We conducted two experiments on stochastic gradient descent , using logistic regression and linear classification . We used four methods to optimize the process of gradient descent in each experiment . We wanted to compare the efficiency and the results in four cases of each experiment .

I. INTRODUCTION

In experiments , our main idea was to use two models to solve classification problems . They were logistic regression model and support vector machine model . As for stochastic gradient descent in each model , we used four methods to optimize . The methods were respectively NAG , RMSProp , AdaDelta and Adam .

We wanted to find out the influence of adjusting parameters to different optimizing process and compared the efficiency between optimizing methods . And we would figure out the difference between models .

II. METHODS AND THEORY

In logistic regression :
Our loss function

$$J(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\log(1+e^{-y_i\cdot\mathbf{w}^\top\mathbf{x}_i}) + \frac{\lambda}{2}||\mathbf{w}||_2^2$$

The update of w

$$\mathbf{w}' \to \mathbf{w} - \eta\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = (1-\eta\lambda)\mathbf{w} + \eta\frac{1}{n}\sum_{i=1}^{n}\frac{y_i\mathbf{x}_i}{1+e^{y_i\cdot\mathbf{w}^\top\mathbf{x}_i}}$$

In linear classification :
Our loss function

$$L : \frac{\|\mathbf{w}\|^2}{2} + \frac{C}{n}\sum_{i=1}^{n}\max(0, 1-y_i(\mathbf{w}^\top\mathbf{x}_i + b))$$

The update of w
W' -> W - η (W+gw(Xi))

if $1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b) >= 0$:

$$\begin{aligned} g_{\mathbf{w}}(\mathbf{x}_i) &= \frac{\partial(-y_i(\mathbf{w}^\top\mathbf{x}_i + b))}{\partial\mathbf{w}} \\ &= -\frac{\partial(y_i\mathbf{w}^\top\mathbf{x}_i)}{\partial\mathbf{w}} \\ &= -y_i\mathbf{x}_i \end{aligned}$$

if $1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b) < 0$:

$$g_{\mathbf{w}}(\mathbf{x}_i) = 0$$

Four optimizing methods :
NAG:

$$\begin{aligned} &\mathbf{g}_t \leftarrow \nabla J(\boldsymbol{\theta}_{t-1} - \gamma\mathbf{v}_{t-1}) \\ &\mathbf{v}_t \leftarrow \gamma\mathbf{v}_{t-1} + \eta\mathbf{g}_t \\ &\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \mathbf{v}_t \end{aligned}$$

Here we used a new variable v to predict the next position which gt would reach . And v was used to get the weighted average direction from the direction now and the directions before .(We set γ as 0.9)

RMSProp:

$$\begin{aligned} &\mathbf{g}_t \leftarrow \nabla J(\boldsymbol{\theta}_{t-1}) \\ &G_t \leftarrow \gamma G_t + (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t \\ &\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot \mathbf{g}_t \end{aligned}$$

Gt used the past gradient information to judge which features were often updated .
(We set γ as 0.9 and ε as 1e - 8)

AdaDelta:

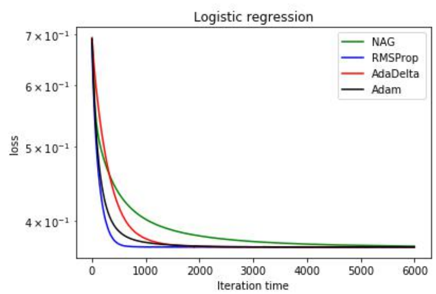
$$\begin{aligned} &\mathbf{g}_t \leftarrow \nabla J(\boldsymbol{\theta}_{t-1}) \\ &G_t \leftarrow \gamma G_t + (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t \\ &\Delta\boldsymbol{\theta}_t \leftarrow -\frac{\sqrt{\Delta_{t-1} + \epsilon}}{\sqrt{G_t + \epsilon}} \odot \mathbf{g}_t \\ &\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} + \Delta\boldsymbol{\theta}_t \\ &\Delta_t \leftarrow \gamma\Delta_{t-1} + (1-\gamma)\Delta\boldsymbol{\theta}_t \odot \Delta\boldsymbol{\theta}_t \end{aligned}$$

In this method , $\sqrt{\Delta_{t-1} + \epsilon}$ was used to estimate the learning rate . In other words , this method estimated next step size by the past step size information .
(We set γ as 0.95)

Adam:

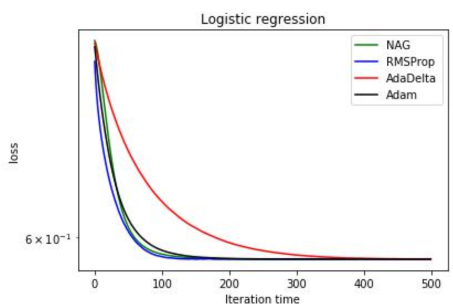
$$
\begin{aligned}
\mathbf{g}_t &\leftarrow \nabla J(\boldsymbol{\theta}_{t-1}) \\
\mathbf{m}_t &\leftarrow \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t \\
G_t &\leftarrow \gamma G_t + (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t \\
\alpha &\leftarrow \eta \frac{\sqrt{1-\gamma^t}}{1-\beta^t} \\
\boldsymbol{\theta}_t &\leftarrow \boldsymbol{\theta}_{t-1} - \alpha \frac{\mathbf{m}_t}{\sqrt{G_t+\epsilon}}
\end{aligned}
$$

III. EXPERIMENT



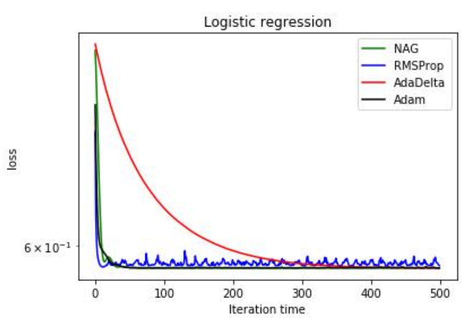
耗时 0:03:42.719528
最终准确率为 0.8409188624777347

```
#确定学习率和
lamda=0.01
n=0.001
count=0
max_count=600
```

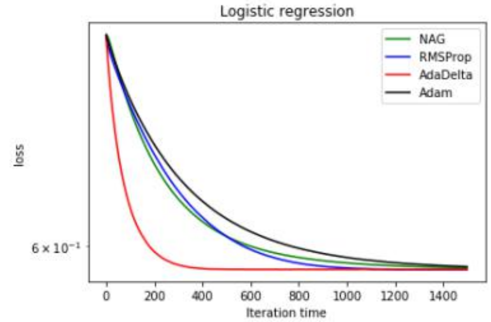


耗时 0:00:20.097700
最终准确率为 0.7637737239727289

```
#确定学习率和
lamda=1
n=0.001
count=0
max_count=500
```

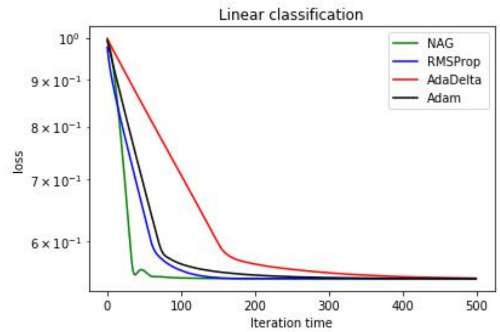



```
#确定学习率和训练
lamda=1
n=0.01
count=0
max_count=500
```



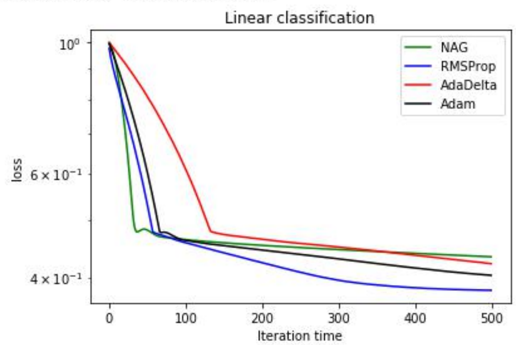
耗时 0:00:55.357440
NAG的准确率为 0.7637737239727289
RMSProp的准确率为 0.7637737239727289
AdaDelta的准确率为 0.7637737239727289
Adam的准确率为 0.7637737239727289

```
#确定学习率和训练次
lamda=1
n=0.0001
count=0
max_count=1500
```



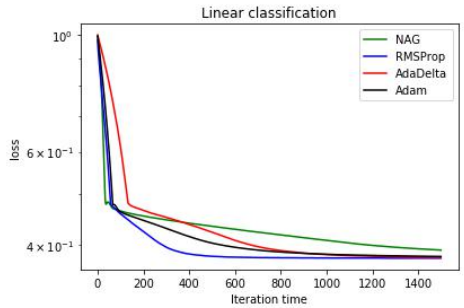
耗时 0:05:45.147613
NAG的准确率为 0.7637737239727289
RMSProp的准确率为 0.7637737239727289
AdaDelta的准确率为 0.7637737239727289
Adam的准确率为 0.7637737239727289

```
#确定学习率和训练
lamda=1
n=0.001
count=0
max_count=500
```



耗时 0:05:09.287004
NAG的准确率为 0.7724341256679566
RMSProp的准确率为 0.8443584546403784
AdaDelta的准确率为 0.7989067010625883
Adam的准确率为 0.8259320680547878

```
#确定学习率和训练
lamda=0.01
n=0.001
count=0
max_count=500
```




```
耗时 0:15:01.943600
NAG的准确率为 0.8352681039248203
RMSProp的准确率为 0.8463853571647934
AdaDelta的准确率为 0.8452797739696579
Adam的准确率为 0.8452797739696579
```

```
#确定学习率和训
lamda=0.01
n=0.001
count=0
max_count=1500
```

From the picture 1 and 2 , we can see that the increase of lamda speed up the learning process and the convergence . But we can also notice the accuracy rate decreases . And in the four optimizing methods , the efficiency of NAG is influenced to a great extent .

From the picture 2 , 3 and 4. We can see AdaDelta is the most stable curve and when learning rate was set smaller , it became the most efficient . When n increase from 0.001 to 0.01 ,RMSProp can't converge .

From the picture 2 and 5 . We can see the results in both model were similar . But from the picture 1, 6 and 7, logistic regression model costed less study rounds than linear classification .

IV. CONCLUSION

The influence of adjusting parameters to different optimizing process and the comparison of efficiency between optimizing methods :

Increasing lamda will speed up whole learning and may decrease the accuracy rate .It influence NAG most in logistic regression .

Increasing learning rate , RMSProp firstly become hard to converge . NAG and Adam converge quickly . Decreasing learning rate , AdaDelta is the most efficient .

In general , logistic regression's learning process is shorter than linear classification .