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:
Xuanyi Zhang

Supervisor:
Qingyao Wu

Student ID：
201530613658

Grade:
Undergraduate

December 14, 2017

Logistic Regression, Linear Classification and Stochastic Gradient Descent

Abstract—In this experiment we use logistic regression and and linear regression to compare differences of several ways of gradient descent and stochastic gradient descent. Also we have better understanding the principles of SVM which is used for larger set of data.

I. Introduction

Logistic regression is a regression model where the dependent variable (DV) is categorical. This article covers the case of a binary dependent variable—that is, where the output can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick.

linear classifier achieves this by making a classification decision based on the value of a linear combination of the characteristics. An object's characteristics are also known as feature values and are typically presented to the machine in a vector called a feature vector. Such classifiers work well for practical problems such as document classification, and more generally for problems with many variables (features), reaching accuracy levels comparable to non-linear classifiers while taking less time to train and use.

The core idea of logistic regression is that if the output of the regression is a continuous value and the range of values is not finite, then we try to map this continuous result value to the result value that can help us judge classification. Therefore, in essence, the logic of regression is based on the return, made a special improvement, and was used for classification issues.

Linear Classifiers: The simplest linear mapping is introduced first: In the formula, each input sample is drawn as a column vector of length D, where W and b are both parameters and parameters W are called weights. The size is K x D and the parameter b is the bias vector K x 1, which affects the output but does not relate to the original sample. Convolution neural network samples to the classification of scores in the same way, but the mapping function f (x) will be more complex and more parameters.

II. Methods and Theory

A. Logistic Regression:

Considered about the accuracy and difficulty of the experiment,we use the cost function with the cross entropy error measure and the regularized term:

$$J(w) = \frac{1}{n}\sum_{i=1}^{n} log(1 + e^{-y_i \cdot w^T x_i}) + \frac{\lambda}{2}\|w\|_2^2,$$

Gradient Descent:

$$\frac{\partial J(w)}{\partial w} = \lambda w - \frac{1}{n}\sum_{i=1}^{n} \frac{y_i x_i}{1+e^{-y_i \cdot w^T x_i}}.$$

B. Linear Classification:

Because the data is large, so we choose the cost function with hinge loss:

$$J(w,b) = \frac{\|w\|^2}{2} + \frac{C}{n}\sum_{i=1}^{n} max(0, 1 - y_i(w^T x_i + b)).$$

Gradient descent:

$$\frac{\partial J(w,b)}{\partial w} = w + \frac{C}{n}\sum_{i=1}^{n} g_w(x_i)$$
$$\frac{\partial J(w,b)}{\partial b} = \frac{C}{n}\sum_{i=1}^{n} g_b(x_i)$$

C.SGD:

$$\mathbf{g}_t \leftarrow \nabla J_i(\boldsymbol{\theta}_{t-1})$$
$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta \mathbf{g}_t$$

D.NAG:

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

E. RMSProp:

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

F. AdaDelta:

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

G. Adam:

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

III. EXPERIMENT

A. Dataset:

Experiment uses a9a of LIBSVM Data, including 32561/16281(testing) samples and each sample has 123/123 (testing) features.

B. Parameters:

Logistic Regression:

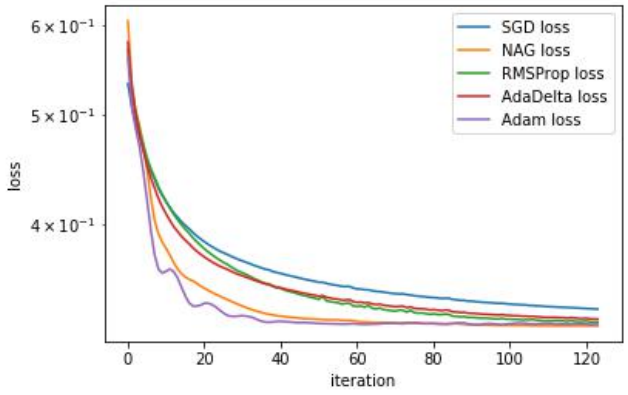
	γ	ε	λ	learning rate
SGD			0.0001	0.5
NAG	0.9		0.0001	0.2
RMSProp	0.9	1e-4	0.0001	0.015
AdaDelta	0.95	1e-4	0.0001	
Adam	0.999	1e-8	0.0001	0.05

Linear Classification:

	γ	ε	C	learning rate
SGD			21	0.01
NAG	0.9		21	0.01
RMSProp	0.9	1e-6	21	0.01
AdaDelta	0.95	1e-5	24	
Adam	0.999	1e-6	30	0.01

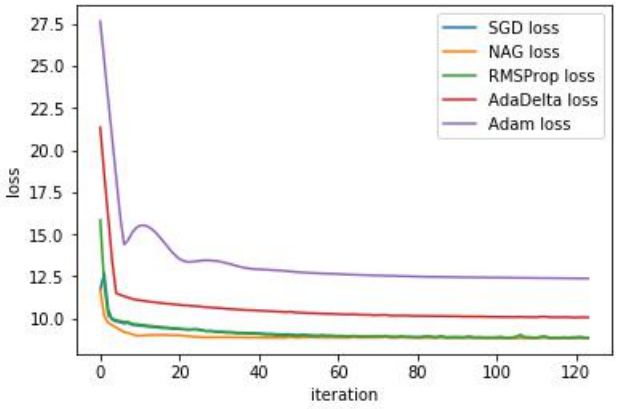
IV. CONCLUSION

Logistic Regression:




```
Accuracy of SGD: 0.844665561083
Accuracy of NAG: 0.852036115718
Accuracy of RMSProp: 0.849026472575
Accuracy of AdaDelta: 0.846999570051
Accuracy of Adam: 0.850193477059
```

Linear Classification:




```
Accuracy of SGD: 0.833548307843
Accuracy of NAG: 0.836496529697
Accuracy of RMSProp: 0.8359437381
Accuracy of AdaDelta: 0.835513789079
Accuracy of Adam: 0.834960997482
```

For sparse data, try to use the learning rate can be adaptive optimization method, without manual adjustment, and the best default value.

SGD typically trains longer and can easily get stuck in saddle points, but results are more reliable with good initialization and learning rate scheduling schemes.

If you are concerned about faster convergence and need to train deeper and more complex networks, we recommend that you use a learning rate adaptive optimization method.

Adadelta, RMSprop, Adam are relatively similar algorithms that perform similarly under similar conditions.