The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Tianxin Cai

Supervisor:
Qingyao Wu

Student ID：
201530611067

Grade:
Undergraduate or Graduate

December 14, 2017

Logistic Regression, Linear Classification and Stochastic Gradient Descent

Abstract—This is an Experimental Study on Stochastic Gradient Descent for Solving Classification Problems, we'll use 4 different optimization methods (NAG, RMSProp, AdaDelta and Adam) to update model parameters and then compare their performance on logistic regression and SVM.

I. INTRODUCTION

SVM and Logistic Regression can also solve binary classification problem. In this experiment, I use both of them to classify the data set. Sometimes, the convergence speed of gradient descent is slow. So I use Stochastic Gradient Descent to update model parameters. But there is also some problems. Thus, some optimization methods can be used such as NAG, RMSProp, AdaDelta and Adam. All of them can perform well.

II. METHODS AND THEORY

1. *Logistic Regression*

The loss function of Logistic Regression is:

$$J(\omega) = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log h_w(x_i) + (1-y_i)\log\left(1-h_w(x_i)\right)\right]$$

Where

$$h_w(x) = g(\omega^T x) = \frac{1}{1+e^{-\omega^T x}}$$

The derivation of loss function is:

$$\frac{\partial \mathrm{J}(\omega)}{\partial \omega} = -y\frac{1}{h_\omega(x)}\frac{\partial h_\omega(x)}{\partial \omega} + (1-y)\frac{1}{1-h_\omega(x)}\frac{\partial h_\omega(x)}{\partial \omega}$$

2. *Linear Classification*

This experiment use SVM for linear classification.

Hinge loss:

$$\text{Hinge loss} = \xi_i = \max(0, 1-y_i(w^T x_i + b))$$

So, the loss function is：

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

The derivation of loss function is:

$$\frac{\partial \mathrm{J}(\omega)}{\partial \omega} = \omega + \frac{C}{n}\sum_{i=1}^{n} g_\omega(x_i)$$

3. *Stochastic Gradient Descent*

SGD works similar as GD, but more quickly by estimating gradient from a few examples at a time, which has many advantages:

Gradient is easy to calculate ("instantaneous")

Less prone to local minima

Small memory footprint

Get to a reasonable solution quickly

Can be used for more complex models and error surfaces

4. *Optimization Methods*

1) Nesterov accelerated gradient(NAG):

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$
$$\theta = \theta - v_t$$

2) AdaDelta:

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2$$
$$\Delta\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$
$$\theta_{t+1} = \theta_t + \Delta\theta_t$$

3) RMSProp:

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1 g_t^2$$
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

4) Adaptive Moment Estimation(Adam):

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t$$

III. EXPERIMENT

1. Dataset:
   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.
2. Environment for Experiment:
   python3, at least including following python package: sklearn，numpy，jupyter，matplotlib
3. Experiment Step:
   1) Load the training set and validation set.
   2) Initialize logistic regression/SVM model parameters, you can consider initializing zeros, random numbers or normal distribution.

3) Select the loss function and calculate its derivation, find more detail in PPT.
4) Calculate gradient G 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 L_{NAG}, $L_{RMSProp}$, $L_{AdaDelta}$ and L_{Adam}.
7) Repeat step 4 to 6 for several times, and drawing graph of L_{NAG}, $L_{RMSProp}$, $L_{AdaDelta}$ and L_{Adam} with number of iterations

4. Results:
The loss graph of Logistic Regression is shown in Figure 1 and the loss graph of SVM is shown in Figure 2.

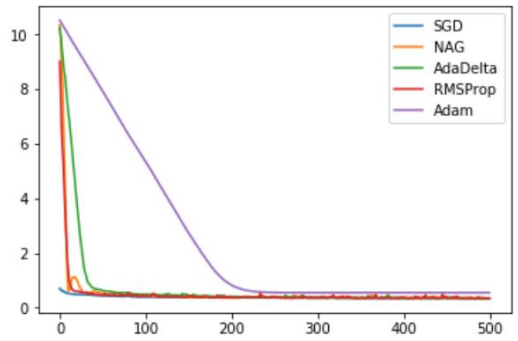

Figure 1. The loss graph of Logistic Regression

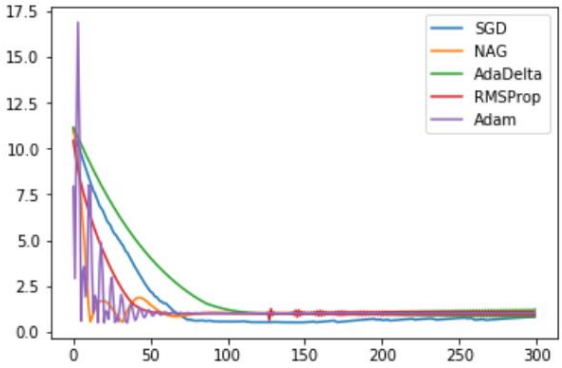

Figure 2. The loss graph of SVM

From these figures, we can see:
Comparison of GD and SGD:
Sometimes, the SGD shocks very violently, we can increase the batch size to make the curve more smooth. But I don't know why the Adam in SVM is so violent. The parameters have been adjusted many times but I can't find an ideal ones. Maybe my code is wrong somewhere.
What the optimization methods can do ?
The optimization methods can accelerate the convergence speed of gradient descent. But sometimes, they maybe don't work well because I don't find the best parameters.
Comparison of SVM and logistic regression:
1. Both of them can increase the data's weight which can influence the classification much and decrease the data's weight which influences the classification little.
2. Logistic regression use log loss and SVM use hinge loss.
3. SVM considers local problem and Logistic regression considers global problem.

IV. CONCLUSION

This experiment is much more difficult than the first one. Firstly, it's a little bit difficult to calculate the derivation of loss function of logistic regression and SVM.
Then, how to code the formula is more difficult.
Thirdly, the optimization methods of SGD is complex but interesting. How amazed I was when I first saw such so many parameters. I try to attain the best one but maybe I failed. However, it is these difficulties that inspire my determination to learn Machine Learning. Thank all my teachers and TAs for helping us a lot.