The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:

Fankai Xu

Student ID：201721046012

Supervisor:

Qingyao Wu

Grade:

graduate

December 15, 2017

Logistic Regression, Linear Classification and Stochastic Gradient Descent

Abstract—Stochastic gradient descent (often shortened to SGD), also known as incremental gradient descent, is a stochastic approximation of the gradient descent optimization and iterative method for minimizing an objective function that is written as a sum of differentiable functions. Compare the differences and relationships between Logistic regression and linear classification.

I. Introduction

The purpose of this report is to introduce two kinds of Classifier. In this experiment, we use Logistic Regression Classification and SVM Linear Classification to fit a relatively large data-set. Compare diffierent kind of optimizers like NAG, RMSProp, AdaDelta and Adam, to see characters of each optimizer.Another purpose is to compare Stochastic Gradient Descent and Gradient Desent.In this experiment, the most import thing is to set appropriate parameters (i.e. η the learning rate, gamma, epoch, **W** the initial matrix, batch_size). For the sake of get a great fit function, we try different learning rate and hyper-parameters for optimizers in the experiment.

II. Methods and Theory

Logistic regression and SVMs are closely related algorithms, even though this isn't obvious from the usual presentation. In particular, the loss functions are very similar, and linear SVMs and logistic regression can often be substituted for one another without a big difference in performance.

Stochastic Gradient Descent (SGD) is often used in solving optimization problems related to model fitting on huge training data set, searching for the best possible parameters that have minimal cost. Usually, the cost function is continuous and smooth.

When dealing with large data set and performing online (real time) learning, high order gradient descent methods (e.g.Newton's method) is not applicable. Even conjugate gradient descent (CG) is not good because it requires accurate computation of the conjugate descent direction, hence the full data set is necessarily involved in each gradient computation.

On the other hand, using SGD, each time the gradient direction is computed based on only part of the data set (or even a single data point), so that the computational cost of each gradient step is constant, doesn't scale up with the size of the data set. In addition, it is reported that SGD converges much faster than regular gradient descent (GD).

Optimizer is also a important part of Gradient Descent method.

NAG (Nesterov accelerated gradient) is a way to accelerate the speed of descending. The formula below is how it update the parameters.

$$\begin{aligned} v_t &= \gamma v_{t-1} + \eta \nabla_\theta J(\theta - \gamma v_{t-1}) \\ \theta &= \theta - v_t \end{aligned} \tag{1}$$

Adadelta is an extension of Adagrad that seeks to reduce its aggressive, monotonically decreasing learning rate. Instead of accumulating all past squared gradients, Adadelta restricts the window of accumulated past gradients to some fixed size w.

RMSprop and Adadelta have both be-en developed independently around the same time stemming from the need to resolve Adagrad's radically diminishing learn-ing rates. RMSprop in fact is identical to the first update vector of Adadelta that we derived above:

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

RMSprop as well divides the learning rate by an exponentially decaying average of squared gradients. Hinton suggests γ to be set to 0.9, while a good default value for the learning rate η is 0.001.

Adaptive Moment Estimation (Adam) is another method that computes adaptive learning rates for each parameter. In addition to storing an exponentially decaying average of past squared gradients vt like Adadelta and RMSprop, Adam also keeps an exponentially decaying average of past gradients mtx, similar to momentum:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned} \tag{3}$$

mt and vt are estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradients respectively, hence the name of the method. As mt and vtvt are init

ialized as vectors of 0's, the authors of Adam observe that they are biased towards zero,especially during the initial time steps, and especially when the decay rates are small (i.e. β1 and β2 ar e close to 1). They counteract these biases by computing bias-c orrected first and second moment estimates:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (4)$$

Formula below is Adam's update rule:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \quad (5)$$

III. Experiment

A. Dataset

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

B. Implementation

In Logistic Regression experiment, first we loaded a9a da-taset for training and a9at dataset for validation, and t hen changed format of the datas and reshaped it. We set epoch and learning rate with proper value, then initialized logistic regression model parameters. 10000 or 5000 dat as were choosen as inputs of stochastic gradient descend. Then computed the loss values. We choosed four kind of optimizers for optimizing the compute of loss value and mo -del parameters(NAG, RMSProp, AdaDelta, Adam).Final-ly, we plot a graph(Fig.1 and Fig.2) to show how loss value change with iteration of each epoch . It' all about how the experiment proceed.

It's strange that when we choosed AdaDelta as optimi-zer, the loss value would become NaN after epoches of tra n- ning.We still do not figure it out.

In SVM Lineaner Classification experiment,we did almo -st the same works, the only different is that we choosed di -fferent loss function. In Logistic Regression experiment, the loss function is cross-entropy loss function, but in SVM experiment, we choosed hing-loss function.

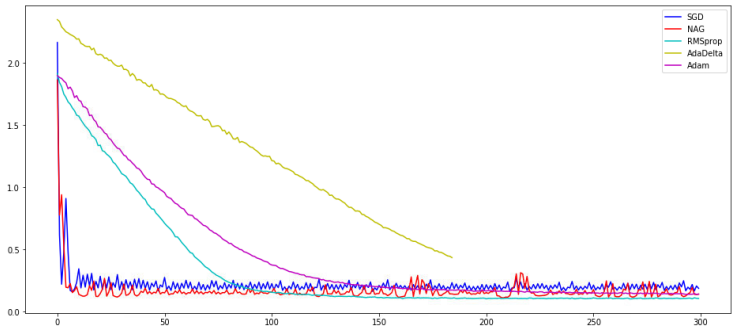

Fig.1. Loss values tendency graph for SGD, NAG, RAMSpr op, AdaDelta, Adam in SVM experiment.

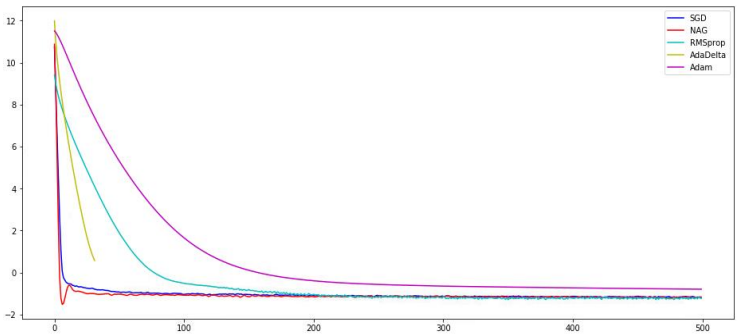

Fig.2. Loss values tendency graph for SGD, NAG, RAMSpr op, AdaDelta, Adam in Logistic Classification experiment.

IV. Conclusion

From these two experiment, we can see that SVM Lineaning Regression and Logistic Classification can do the same classifi -cation job for not that large dataset. Stochastic Gradient De-scend is faster than Gradient Descend in trainning, because it use less datas to trainning in each epoch. For each optimizer,we can see from the figures, they have different loss change rate, but all did a good job.