華南理工大學

South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Songlin Bao

Supervisor:
Mingkui Tan

Student ID：
201530611029

Grade:
Undergraduate

December 9, 2017

Comparison of Various Stochastic Gradient Descent Methods for Solving Classification Problems

Abstract—In order to compare the distinction and connection between gradient descent and stochastic gradient descent, also between the logistic regression and linear classifier, we implement this experiment and update parameters using different optimization methods to compare the effectiveness between NAG, RMSProp, AdaDelta and Adam. The results show that Adam is smooth and fastest . NAG 's convergence speed is slowest.

I. Introduction

Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient of the function at the current point. Stochastic gradient descent, 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. Many improvements on the basic stochastic gradient descent algorithm have been proposed and used. This experiment aims to simply compared the differences between NAG, RMSProp, AdaDelta and Adam. We implement two binary classification models: logistic regression and linear SVM. And use NAG, RMSProp, AdaDelat and Adam to update the model parameters respectively.

II. Methods and Theory

A. Logistic regression

In statistics, logistic regression is a regression model where the dependent variable (DV) is categorical. The output can take two values, "0" and "1", or "-1" and "1", which represent outcomes.

Logistic regression is used for predicting dependent variables that take membership in one of a limited number of categories.and find the best fitting model to describe the relationship. The loss function is:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i log(p_i) + (1-y_i) log(1-p_i)$$

$$p_i = sigmoid(w^T x_i)$$

and the gradient of loss function is:

$$\frac{\partial Loss}{\partial w} =- \frac{1}{N}\sum_{i=1}^{N} x_i(p_i - y_i)$$

B. SVM

SVM is supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier . An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

The loss function is:

$$\mathrm{L}_D(w,b) = \frac{\| w \|^2}{2} + C\sum_{i=1}^{N} \max(0, 1 - y_i(w^T x_i + b))$$

The gradient of the loss function is:

$$\frac{\partial L}{\partial w} = w^T - Cx^T y$$

$$\frac{\partial L}{\partial b} = w^T$$

C. optimization methods

(1) NAG

$$g_t \leftarrow \nabla L_D(\theta_{t-1} - \gamma v_{t-1})$$

$$v_t \leftarrow \gamma v_{t-1} + \eta g_t$$

$$\theta_t \leftarrow \theta_{t-1} - v_t$$

(2) RMSProp

$$g_t \leftarrow \nabla L_D(\theta_{t-1})$$

$$G_t \leftarrow \gamma G_t + (1-\gamma) g_t \odot g_t$$

$$\theta_t \leftarrow \theta_{t-1} - \frac{\eta}{\sqrt{G_t + \varepsilon}} \odot g_t$$

(3) AdaDelta

$$\mathrm{g}_t \leftarrow \nabla L_D(\theta_{t-1})$$

$$G_t \leftarrow \gamma G_t + (1-\gamma)\mathrm{g}_t \odot \mathrm{g}_t$$

$$\Delta\theta_t \leftarrow -\frac{\sqrt{\Delta_{t-1} + \varepsilon}}{\sqrt{G_t + \varepsilon}} \odot \mathrm{g}_t$$

$$\theta_t \leftarrow \theta_{t-1} + \Delta\theta_t$$

$$\Delta_t \leftarrow \gamma\Delta_{t-1} + (1-\gamma)\Delta\theta_t \odot \Delta\theta_t$$

(4) Adam

$$g_t \leftarrow \nabla L_D(\theta_{t-1})$$

$$m_t \leftarrow \beta m_{t-1} + (1-\beta) g_t$$

$$G_t \leftarrow \gamma G_t + (1-\gamma) g_t \odot g_t$$

$$\alpha \leftarrow \eta \frac{\sqrt{1-\gamma^t}}{1-\beta^t}$$

$$\theta_t \leftarrow \theta_{t-1} - \alpha \frac{m_t}{\sqrt{G_t + \varepsilon}}$$

III. Experiment

A. Dataset

The experiment uses the a9a data in LIBSVM Data, which contains 32561 / 16281 (testing) samples with 123/123 (testing) attributes per sample.

B. Experimental procedure

- Logistic Regression and Stochastic Gradient Descent

- Load the training set and validation set.
- Initalize logistic regression model parameters, you can consider initalizing zeros, random numbers or normal distribution.
- Select the loss function and compute its derivation
- Calculate gradient G toward loss function from partial samples.
- Update model parameters using different optimized methods(NAG,RMSProp,AdaDelta and Adam).
- Select the appropriate threshold, mark the sample in vaildation set whose predict result greater than the threshold as positive, on the contrary as negative. Predict the validation set and get the different optimized method loss L_{NAG} $L_{RMS\,\mathrm{Prop}}$ $L_{AdaDelta}$ L_{Adam}.
- Repeate step 4 to 6 for several times, and drawing graph of L_{NAG} $L_{RMS\,\mathrm{Prop}}$ $L_{AdaDelta}$ L_{Adam} with the number of iterations.

- Linear Classification and Stochastic Gradient Descent

- Load the training set and validation set.
- Initalize SVM model parameters, you can consider initalizing zeros, random numbers or normal distribution.
- Select the loss function and compute its derivation
- Calculate gradient G toward loss function from partial samples.
- Update model parameters using different optimized methods(NAG,RMSProp,AdaDelta and Adam).
- Select the appropriate threshold, mark the sample in vaildation set whose predict result greater than the threshold as positive, on the contrary as negative. Predict the validation set and get the different optimized method loss L_{NAG} $L_{RMS\,\mathrm{Prop}}$ $L_{AdaDelta}$ L_{Adam}.
- Repeate step 4 to 6 for several times, and drawing graph of L_{NAG} $L_{RMS\,\mathrm{Prop}}$ $L_{AdaDelta}$ L_{Adam} with the number of iterations.

Thefollowing are the detail parameters responding to different optimization methods in two models and the figure.

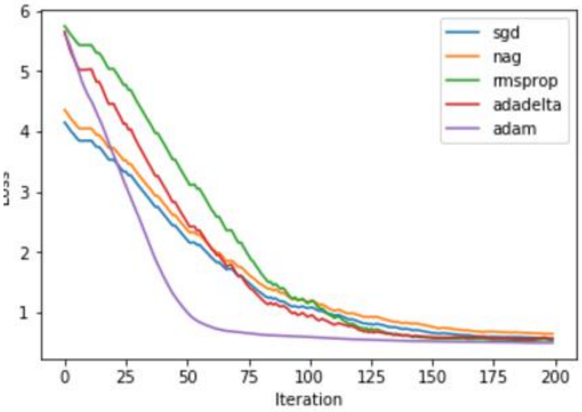
- Logistic Regression and Stochastic Gradient Descent

(1) NAG
lr = 0.01
gamma = 0.9

(2) RMSProp
lr = 0.01
gamma = 0.95
delta = 10e-7

(3) AdaDelta
gamma = 0.9
delta = 10e-7
lr = 10

(4) Adam
delta = 10e-8
beta = 0.9
gamma = 0.999
lr = 0.01



- Linear Classification and Stochastic Gradient Descent

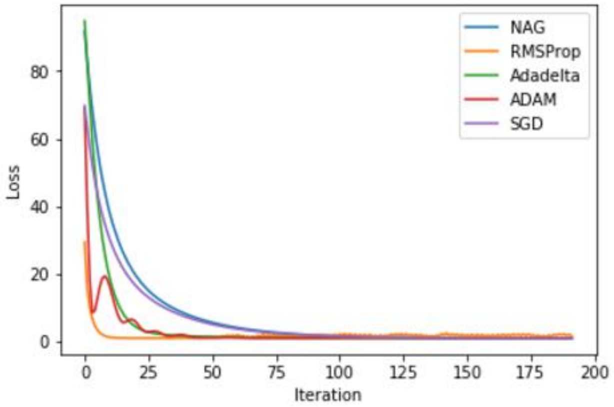
(5) NAG
lr = 0.01
gamma = 0.9

(6) RMSProp
lr = 0.1
gamma = 0.95
delta = 10e-7

(7) AdaDelta
gamma = 0.9
delta = 10e-7
lr = 0.1

(8) Adam
delta = 10e-8
beta = 0.9
gamma = 0.999

lr = 0.01



In this two figure, with the iteration, loss gradually decreases, and finally tents to smooth.In the logistic regression, Adam declines fastest and has the best convergency. While NAG has the slowest convergence rate. In SVM, Adam declines fastest in the begining. And RMSprop has the best convergency. NAG still has the slowest convergence ratedecline faster. The optimization results of the two models show that Adam and RMSProp have a faster and relatively stable performance, and can find better fitting model.

IV. CONCLUSION

In this experiment, we Compare and understand the logistic regression and linear SVM and optimize them with NAG, RMSProp,AdaDelta and Adam.

Logistic regression and linear SVM are all the linear classifier. Linear SVM is not directly dependent on data distribution.The classification plane is not affected by a kind of point. But logistic regression is affected by all the points.

Many improvements on the basic stochastic gradient descent algorithm have been proposed and used. In particular, in machine learning, the need to set a learning rate has been recognized as problematic. Setting this parameter too high can cause the algorithm to diverge; setting it too low makes it slow to converge. A conceptually simple extension of stochastic gradient descent makes the learning rate a decreasing function η of the iteration number t, giving a learning rate schedule, so that the first iterations cause large changes in the parameters, while the later ones do only fine-tuning. By comparison, we know that Adam and RMSProp have a faster and relatively stable performance.