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:
Jiaqi Zhong

Supervisor:
Qingyao Wu

Student ID：
201720144948

Grade:
Graduate

December 9, 2017

Logistic Regression, Linear Classification and Stochastic Gradient Descent

Abstract—This experiment of machine learning is about logistic regression, linear classification and stochastic gradient descent, and the experiment is carried out on a small scale data set with optimized method including NAG, RMSProp, AdaDelta and Adam. The purpose of the experiment is to give us a better understanding of the principles of logistic regression, linear classification and stochastic gradient descent, and to understand the process of optimization and tuning.

I. Introduction

There are two parts to this experiment of machine learning. The first part of the experiment was logistic regression and stochastic gradient descent. In this section, I used the logistic regression model to conduct experiments on the a9a of LIBSVM Data. The second part was linear classification and stochastic gradient descent. In this section, I also experimented on the a9a of LIBSVM Data with support vector machine model.

II. METHODS AND THEORY

A. *Logistic regression*

An explanation of logistic regression can begin with an explanation of the standard logistic function. The logistic function is useful because it can take any real input t, whereas the output always takes values between zero and one and hence is interpretable as a probability. The logistic function is defined as follows:

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

Logistic regression is an important machine learning algorithm. The goal is to model the probability of a random variable Y being 0 or 1 given experimental data.Consider a generalized linear model function parameterized by θ ,

$$h_\theta(x^{(i)}) = \frac{1}{1 + e^{-\theta^T x}}$$

If we attempt to model the probability that y is 0 or 1 with the function:

$$Pr(y|x;\theta) = h_\theta(x)^y(1 - h_\theta(x))^{(1-y)}$$

And the loss function is:

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))]$$

The gradient of the loss function is:

$$\frac{\partial J(\theta)'}{\partial \theta_j} = \frac{1}{\mathrm{m}}[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}]$$

B. *SVM*

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models 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 (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). 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 hypothesis function is:

$$f(x) = w^T x + b$$

And the loss function is:

$$f(\vec{w}, b) = \left[\frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i(w \cdot x_i + b)\right)\right] + \lambda\|w\|^2$$

C. *Stochastic gradient descent*

Stochastic gradient descent (SGD) in contrast performs a parameter update for each training example $x^{(i)}$ and the label $y^{(i)}$:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}; y^{(i)})$$

Batch gradient descent performs redundant computations for large datasets, as it recomputes gradients for similar examples before each parameter update. SGD does away with this redundancy by performing one update at a time. It is therefore usually much faster and can also be used to learn online. SGD performs frequent updates with a high variance that cause the objective function to fluctuate heavily.

While batch gradient descent converges to the minimum of the basin the parameters are placed in, SGD's fluctuation, on the one hand, enables it to jump to new and potentially better local minima. On the other hand, this ultimately complicates convergence to the exact minimum, as SGD will keep overshooting. However, it has been shown that when we slowly decrease the learning rate, SGD shows the same convergence behaviour as batch gradient descent, almost certainly converging to a local or the global minimum for non-convex and convex optimization respectively.

D. *NAG*

Nesterov accelerated gradient (NAG) is a way to give our momentum term this kind of prescience. We know that we will use our momentum. The approximate future position of our parameters:

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

E. *RMSProp*

RMSprop is an unpublished, adaptive learning rate method proposed by Geoff Hinton in Lecture 6e of his Coursera Class. RMSprop and Adadelta have both been developed independently around the same time stemming from the need to resolve Adagrad's radically diminishing learning rates. RMSprop in fact is identical to the first update vector of Adadelta that we derived above:

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

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.

F. AdaDelta

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*.Instead of inefficiently storing *w* previous squared gradients, the sum of gradients is recursively defined as a decaying average of all past squared gradients, and the Adadelta update rule:

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t$$
$$\theta_{t+1} = \theta_t + \Delta\theta_t$$

G. Adam

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 vtvt like Adadelta and RMSprop, Adam also keeps an exponentially decaying average of past gradients m_t, similar to momentum:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

m_t and v_t 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 m_t and v_t are initialized 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. β_1 and β_2 are close to 1).

They counteract these biases by computing bias-corrected first and second moment estimates:

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

They then use these to update the parameters just as we have seen in Adadelta and RMSprop, which yields the Adam update rule:

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

III. Experiment

A. 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.

B. Implementation

(1) Logistic Regression and Stochastic Gradient Descent:

a)Load the training set and validation set.

b)Initalize logistic regression model parameters. I chosen to set all parameter into zero.

c)Select the loss function and calculate its derivation, find more detail in PPT.

d)Calculate gradient G toward loss function from partial samples.

e)Update model parameters using different optimized methods(NAG, RMSProp, AdaDelta and Adam).

f)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}, $L_{RMSProp}$, $L_{AdaDelta}$, L_{Adam}.

g)Repeate step d to f for several times, and drawing graph of L_{NAG}, $L_{RMSProp}$, $L_{AdaDelta}$, L_{Adam} with the number of iterations.

The experimental results of different optimized methods(NAG, RMSProp, AdaDelta and Adam) are shown in Fig. 1, Fig. 2, Fig. 3 and Fig. 4.

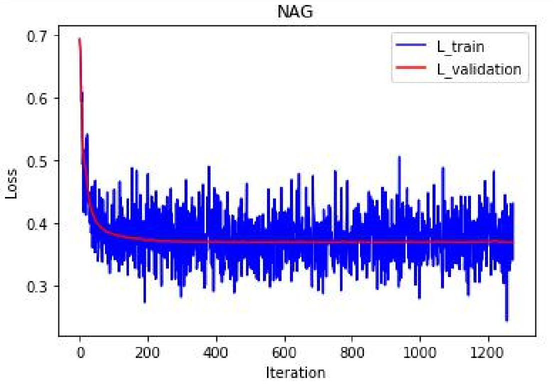

Fig. 1. the result of NAG

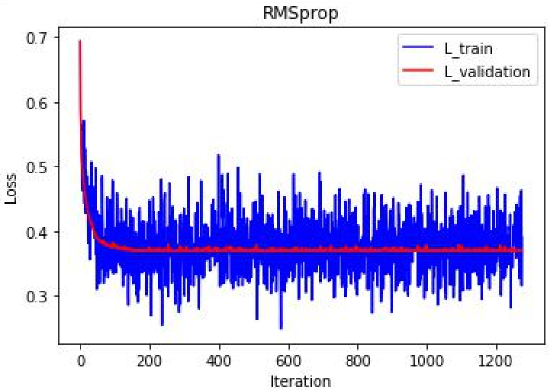

Fig. 2. the result of RMSProp

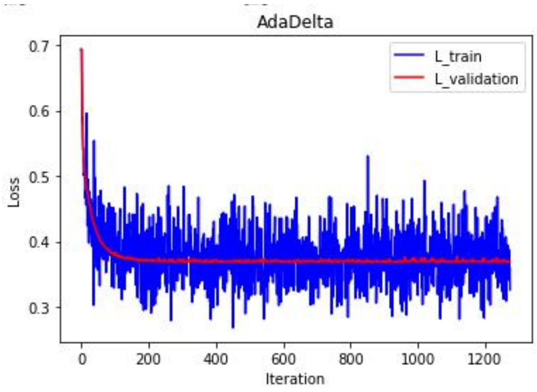

Fig. 3. the result of AdaDelta

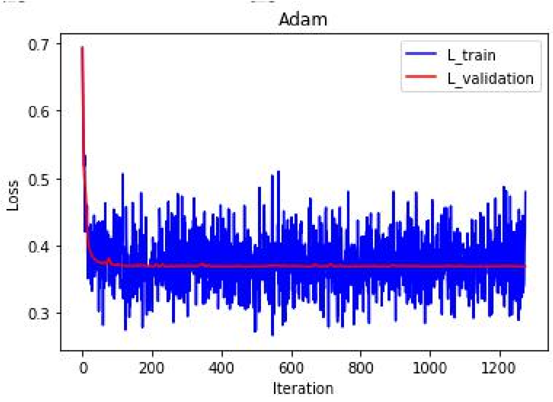

Fig. 4. the result of Adam

(2) Linear Classification and Stochastic Gradient Descent:

a)Load the training set and validation set.

b)Initalize SVM model parameters. I chosen to set all parameter into zero.

c)Select the loss function and calculate its derivation, find more detail in PPT.

d)Calculate gradient G toward loss function from partial samples.

e)Update model parameters using different optimized methods(NAG, RMSProp, AdaDelta and Adam).

f)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}, $L_{RMSProp}$, $L_{AdaDelta}$, L_{Adam}.

g)Repeate step d to f for several times, and drawing graph of L_{NAG}, $L_{RMSProp}$, $L_{AdaDelta}$, L_{Adam} with the number of iterations.

The experimental results of the loss and accuracy of different optimized methods(NAG, RMSProp, AdaDelta and Adam) are shown in Fig. 5, Fig. 6, Fig. 7 and Fig. 8.

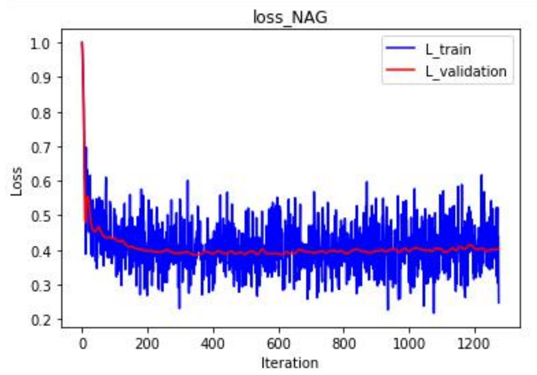

Fig. 5(a). the loss of NAG

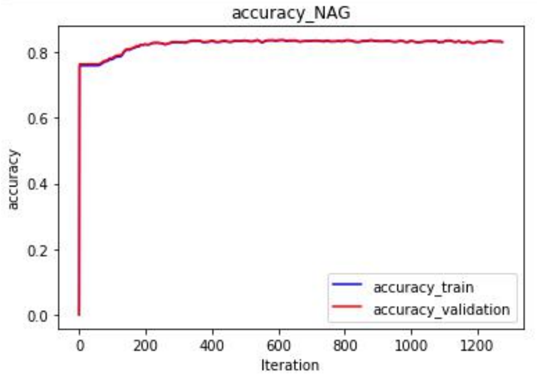

Fig. 5(b). the accuracy of NAG

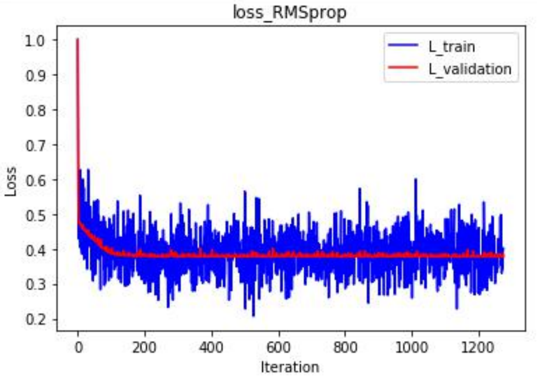

Fig. 6(a). the loss of RMSProp

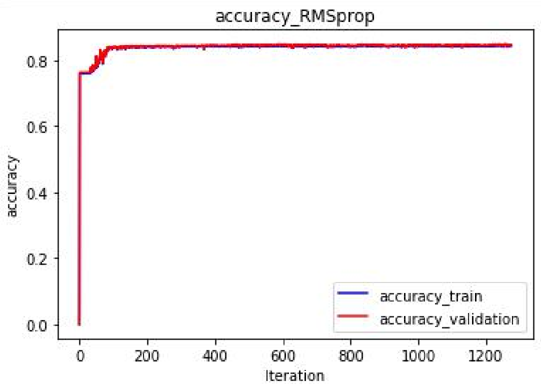

Fig. 6(b). the accuracy of RMSProp

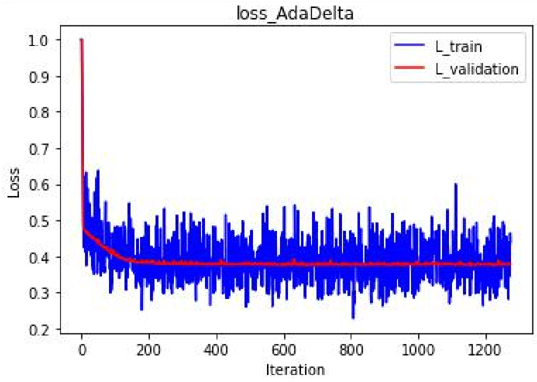

Fig. 7(a). the loss of AdaDelta

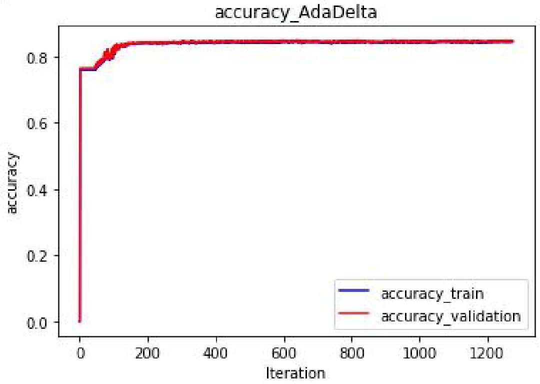

Fig. 7(b). the accuracy of AdaDelta

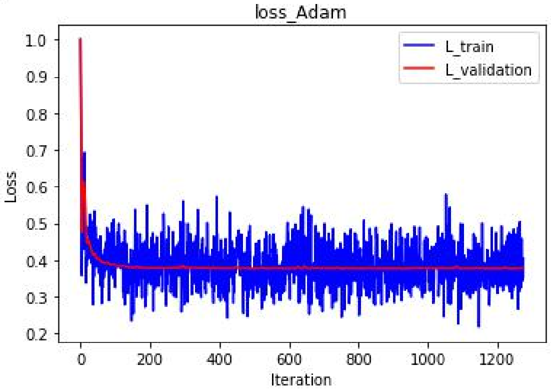

Fig. 8(a). the loss of Adam

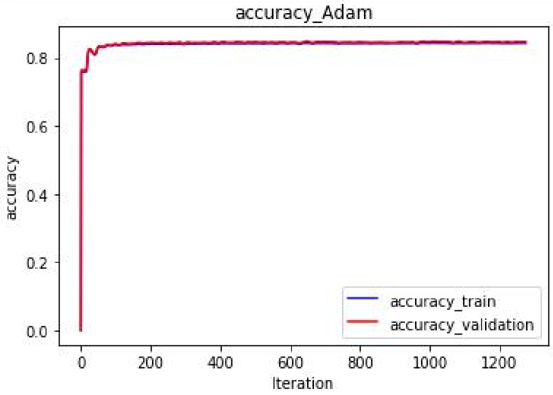

Fig. 8(b). the accuracy of Adam

IV. CONCLUSION

In this experiment, a total of two small experiments are included, one is logistic regression and stochastic gradient descent; the other is linear classification and stochastic gradient descent. The former uses a logistic regression model; the latter uses a support vector machine model. And both of them used four different optimized methods(NAG, RMSProp, AdaDelta and Adam). After this experiment, I have further understood of the difference between gradient descent and stochastic gradient descent and understood the differences and relationships between Logistic regression and linear classification.Further understood the principles of SVM and practice on larger data; and realize the process of optimization and adjusting parameters is very important and crucial.