On the convergence speed of AMSGRAD and beyond

Tao Tan, Shiqun Yin, Kunling Liu, Man Wan
Faculty of Computer and Information Science
Southwest University
Chongqing, China 400715
1169311978@qq.com; Corresponding author: qqqq-qiong@163.com; 121398910@qq.com; 1812658139@qq.com

***Abstract*—In ICLR's (2018) best paper "On the Convergence of Adam and Beyond", the author points out the shortcomings in Adam's convergence proof, proposes an AMSGRAD algorithm that can guarantee convergence as the number of iterations increases. However, through some comparative experiments, this paper finds that there are two problems in the convergence process of AMSGRAD algorithm. Firstly, the AMSGRAD algorithm is easy to oscillate; Secondly, the AMSGRAD algorithm converges slowly. After analysis, the above two problems can be solved by the following ways. When $g_{t-1}g_t > 0$, this paper adds the momentum term in Momentum algorithm to the AMSGRAD algorithm to accelerate convergence. When $g_{t-1}g_t \leq 0$, this paper use SGD algorithm instead of AMSGRAD algorithm to update the model weights. In order to eliminate some negative effects of the previous parameter gradient on the current parameter gradient and reduce the oscillation amplitude of the objective function, the first-order and second-order moment estimations of the parameter gradient are recalculated when $g_{t-1}g_t \leq 0$. Therefore, this paper proposes the ACADG algorithm, which not only can improve the convergence speed, suppress the oscillation amplitude of the objective function, but also can improve the accuracy of training and test data sets.**

***Keywords*-AMSGRAD algorithm; Momentum algorithm; oscillation amplitude; ACADG algorithm; convergence speed;**

I. INTRODUCTION

At present, most learning algorithms in deep learning are based on iterative ideas, and the purpose is to find a set of network parameters, optimize the model weights, and minimize the objective function. The existing optimization algorithms are mainly based on the idea of SGD[1−3] algorithm, and can be divided into two categories.

One is the momentum method, such as Momentum[4] algorithm and NAG[5] algorithm. Using the idea of physical momentum, these algorithms not only can suppress the oscillation amplitude of objective function, speed up the convergence of algorithm, but also can make the objective function jump out of the unsatisfactory local optimal solution, improve the accuracy of training and test data sets.

The other is the adaptive method[6−7], which mainly includes RMSPROP[8] algorithm, ADAGRAD[9] algorithm, ADADELTA[10] algorithm, ADAM[11] algorithm and AMSGRAD[12] algorithm. These algorithms can calculate the adaptive learning rate for each iteration, find the appropriate step sizes, speed up the model convergence, and reduce the loss value of training and test data sets.

The above algorithms have been recognized in many practical applications. For example, in image recognition, the idea of replacing SGD algorithm with ADAM algorithm can greatly improve the convergence speed, improve the accuracy of training and test data sets.

In "On the Convergence of Adam and Beyond", the author points out that the unbiased second-order moment estimation of the parameter gradient and the number of iteration steps do not always maintain a non-decreasing function relationship in the iterative process of the ADAM algorithm. Thus the adaptive learning rate and the number of iteration steps do not always maintain a non-increasing function relationship, and the uncertainty of the adaptive learning rate may lead to the phenomenon that the ADAM algorithm does not converge during the iterative process. In addition, they proposed the AMSGRAD algorithm. By recording the historical maximum of the unbiased second-order moment estimation of the parameter gradient, the non-decreasing of the unbiased second-order moment estimation of the parameter gradient and the non-increment of the adaptive learning rate are ensured, which ensures that the AMSGRAD algorithm always converges during the iterative process.

However, this paper finds two problems in AMSGRAD algorithm. Firstly, the AMSGRAD algorithm is easy to oscillate. Secondly, the AMSGRAD algorithm converges slowly. Therefore, this paper proposes the ACADG algorithm, which is a new adaptive learning rate optimization algorithm. In short, the main contributions of this paper are as follows:

1) This paper finds that there are two problems in the convergence process of AMSGRAD algorithm, one is oscillation amplitude, the other is convergence speed.
2) This paper proposes the ACADG algorithm and successfully solved the above two problems.
3) Through synthetic experiment, logistic regression and DNN experiment, CNN experiment, it is verified that ACADG algorithm not only has faster convergence speed, lower oscillation amplitude, but also performs better in training and test data sets than AMSGRAD and ADAM algorithms.

II. Related work

This section is divided into two parts. The first part introduces the Momentum algorithm, which mainly includes the update process of model weights in different iteration stages. The second part introduces the AMSGRAD algorithm, and compares the parameter update process of the AMSGRAD algorithm with that of the ADAM algorithm.

A. Momentum algorithm

In physics, people use momentum to simulate the inertia of an object as it moves. Through derivation, it is found that the momentum idea can be used in the deep learning optimization algorithm to improve the convergence speed. Then the Momentum algorithm is proposed, and the update rules of model weights are designed as follows:

$$\triangle w_t = \rho \triangle w_{t-1} - \eta g_t$$
$$w_t = w_{t-1} + \triangle w_t$$

Where w is the model weights, t is the number of iteration steps, ρ is the momentum factor, η is the learning rate, g is the parameter gradient.

After analysis, the iterative process of the Momentum algorithm can be divided into three stages.

In the initial stage of the iteration, the angle between the parameter gradient vector in the previous step and the parameter gradient vector in the current step is less than 90 degrees, that is $g_{t-1}g_t > 0$, the momentum term can help the algorithm to accelerate convergence.

In the middle stage of the iteration, the momentum factor can increase the update range of the model weights, so that the objective function can jump out of the unsatisfactory local optimal solution as much as possible.

In the later stage of the iteration, the angle between the parameter gradient vector in the previous step and the parameter gradient vector in the current step is greater than 90 degrees, that is $g_{t-1}g_t \leq 0$. The objective function will oscillate around the relatively satisfactory local optimal value. The momentum factor will reduce the update range of the model weights, suppress the amplitude of the objective function, and make the objective function converge to the relatively satisfactory local optimal solution more quickly.

To sum up, the momentum term not only can improve the convergence speed of the algorithm, but also can appropriately suppress the amplitude of the objective function. It can also help the objective function to jump out of the unsatisfactory local optimal solution, improve the accuracy of training and test data sets.

B. AMSGRAD algorithm

AMSGRAD algorithm is a new exponential moving average gradient optimization algorithm, and its purpose is to solve the convergence problem of ADAM algorithm. The specific implementation steps of AMSGRAD algorithm are as follows:

1) Calculate the biased first-order (m) and second-order (v) moment estimators of parameter gradient, respectively.
2) Use m and v to calculate the unbiased first-order ($\widehat{m}$) and second-order ($\widehat{v}$) moment estimators of parameter gradient, respectively.
3) Record the historical maximum (v_{max}) of unbiased second-order moment estimator of parameter gradient.
4) Update the model weights with following formula:
$$w_t = w_{t-1} - \frac{\eta}{\sqrt{v_{max}+\epsilon}}\widehat{m}_t$$
5) Repeat the above four steps until meet the stopping criterion.

Where w is the model weights, t is the number of iteration steps, η is the learning rate, ϵ is a constant and used to prevent the denominator from being 0.

The biggest difference between AMSGRAD algorithm and ADAM algorithm in the update process of model weights is the calculation method of unbiased second-order moment estimation of parameter gradient. The ADAM algorithm directly uses the unbiased second-order moment estimation of the parameter gradient to calculate the adaptive learning rate, while the AMSGRAD algorithm uses the historical maximum of the unbiased second-order moment estimation of the parameter gradient to calculate the adaptive learning rate. Thus, it is found that the AMSGRAD algorithm ensures the convergence of the model by making the adaptive learning rate maintain a non-incremental functional relationship with the iterative steps.

III. ACADG Algorithm

This section is divided into three parts. The first part introduces two disadvantages of the AMSGRAD algorithm in the iterative process. The second part introduces the A-CADG algorithm. The third part introduces the performance of ACADG algorithm in convergence speed and oscillation amplitude.

A. Disadvantages of the AMSGRAD algorithm

Through the derivation of mathematical formulas[12], the AMSGRAD algorithm proves that it can overcome the non-convergence problem in ADAM algorithm, and guarantees that the optimization algorithm converges with the increase of the number of iterations. However, this paper finds the following two problems from the convergence process of the AMSGRAD algorithm:

1) In order to ensure the convergence of the AMSGRAD algorithm, the AMSGRAD algorithm must record the historical maximum of the unbiased second-order moment estimate of the parameter gradient, and use it to calculate the adaptive learning rate. However, the adaptive learning rate of the AMSGRAD algorithm is lower than that of the ADAM algorithm during the iterative process, which will increase convergence time, reduce convergence speed.

2) When $g_{t-1}g_t \leq 0$, the angle between the parameter gradient vector in the previous step and the parameter gradient vector in the current step is greater than 90 degrees. Using the AMSGRAD algorithm to update the model weights will cause the unbiased first-order moment estimate of the parameter gradient in the previous step to have a negative effect on the current parameter gradient vector. Sometimes, this negative effect will last for a long time, and even cause the objective function to experience severe oscillations during convergence.

B. The ACADG algorithm

In order to solve the above two problems of AMSGRAD algorithm, this paper designs ACADG algorithm, as shown below:

Algorithm 1 ACADG

Input: η_0, β_1, β_2, m_0, v_0, $\triangle w_0$, T, ϵ, ε
Output: w

for $t = 1, 2, 3, \cdots, T$ **do**
  $\eta_t = \frac{\eta_0}{\sqrt{t}}$
  $g_t = \bigtriangledown f(w_{t-1})$
  $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$
  $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
  $\widehat{m}_t = \frac{m_t}{1-\beta_1^t}$
  $\widehat{v}_t = \frac{v_t}{1-\beta_2^t}$
  $\widehat{v}_t = max(\widehat{v}_{t-1}, \widehat{v}_t)$
  if $g_{t-1}g_t \leq 0$ **then**
    $w_t = w_{t-1} - \eta_t g_t$
    $m_t = g_t$
    $v_t = g_t^2$
  else
    $w_t = w_{t-1} - \frac{\eta_t}{\sqrt{v_{max}+\epsilon}} \widehat{m}_t + \beta_1 \triangle w_{t-1}$
  $\triangle w_t = w_t - w_{t-1}$
  if $f(w_t) \leq \varepsilon$ **then**
    return w_t
return w_t

where η_0 is the initial learning rate, β_1 is the coefficient of the first-order moment estimation of parameter gradient, β_2 is the coefficient of the second-order moment estimation of parameter gradient, m_0 is the initial first-order moment estimation of parameter gradient, v_0 is the initial second-order moment estimation of parameter gradient, $\triangle w_0$ is the initial value of the model weights differences, T is the number of iterations, ϵ is a constant and used to prevent the denominator from being 0, ε is the minimum loss value that satisfies the stopping condition, w is the model weights.

C. The performance of ACADG algorithm

Compared with the AMSGRAD algorithm, it is found that the ACADG algorithm has the following three changes.

1) The ACADG algorithm uses an exponential decay learning rate method in model training.
2) When $g_{t-1}g_t > 0$, the ACADG algorithm adds the momentum term to the AMSGRAD algorithm based on the idea of the Momentum algorithm.
3) When $g_{t-1}g_t \leq 0$, the ACADG algorithm uses the SGD algorithm to update the model weights, and recalculates the first-order and second-order moment estimates of the parameter gradient.

The ACADG algorithm divides the training process into two cases by computing $g_{t-1}g_t$.

When $g_{t-1}g_t > 0$, the momentum term has a positive effect on the first-order moment estimation of the parameter gradient in the current steps. Through the momentum term, the ACADG algorithm can speed up the decline of the loss function, and make the loss function quickly reach a satisfactory local optimal solution. Therefore, the ACADG algorithm accelerates the convergence speed of the model.

When $g_{t-1}g_t \leq 0$, whether the AMSGRAD algorithm or the ADAM algorithm, the first-order moment estimation of the parameter gradient in the previous steps will have a negative effect on the parameter gradient vector in the current steps. Sometimes, this negative effect will last for a long time, and even cause the objective function to experience severe oscillations during convergence. In this case, the ACADG algorithm uses the SGD algorithm to update the model weights. Regardless of the negative effect, the ACADG algorithm will not be affected. Therefore, the ACADG algorithm reduces the oscillation amplitude of the loss function.

In addition, by recalculating the first-order and second-order moment estimates of the parameter gradient and using the exponential decay learning rate, the ACADG algorithm can quickly jump out of the unsatisfactory local optimal solution, improve the accuracy of training and test data sets, and also can help the model to quickly converge to satisfactory local optimal solution and reduce the oscillation amplitude of the loss function.

IV. EXPERIMENTS

In order to verify the effectiveness of the ACADG algorithm, this paper designs three experiments. The first one is a synthetic experiment, which uses two different objective functions to verify the convergence, convergence speed and oscillation amplitude of ACADG algorithm. The next one is a logistic regression and DNN experiment, which uses different models to verify the convergence speed and the oscillation amplitude of ACADG algorithm on convex optimization and non-convex optimization problems. The last one is a CNN experiment, which uses two different data sets to verify the versatility of the ACADG algorithm. In the three experiments, the ADAM and AMSGRAD algorithms are used as comparison objects. The convergence of the algorithm, the convergence speed of the model,

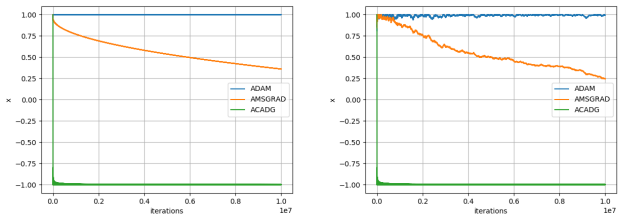

Figure 1: The performance of the three algorithms on the synthetic data. The first and second images represent the results of the three algorithms in the first and second objective functions, respectively.

the oscillation amplitude of the objective function, the accuracy on training and test data sets are used as the evaluation criteria for measuring the performance of the three algorithms. The specific code for all experiments is at $https://github.com/Tantao122200/acadg\_new$.

A. Synthetic experiment

In order to verify the convergence, convergence speed and oscillation amplitude of ACADG algorithm in synthetic experiment. This paper designs two different objective functions[12], and as follows:

$$f_t(x) = \begin{cases} 1010x & t\%101 = 1 \\ -10x & otherwise \end{cases} \tag{1}$$

$$f_t(x) = \begin{cases} 1010x & p = 0.01 \\ -10x & otherwise \end{cases} \tag{2}$$

Where x belongs to a constraint set and $x \in [-1, 1]$, t is the number of iteration steps, p is the probability and $p \in [0, 1]$.

As can be seen from the above settings, whether the first objective function or the second objective function, the minimum value is obtained at $x = -1$.

In the ACADG algorithm, the relevant experimental parameters η_0, β_1, β_2, m_0, v_0, $\triangle w_0$, T, ϵ, ε are set as 0.01, 0.9, 0.999, 0, 0, 0, 1e7, 1e-8, 1e-9, respectively. The experimental parameters related to the AMSGRAD and ADAM algorithms are consistent with those in ACADG algorithm, and the experimental results are shown as Flg.1.

As can be seen from Fig.1, the three algorithms have the following three phenomena in the first objective function and the second objective function:

1) As the number of iterations increases, the ADAM algorithm converges to 1, but both the AMSGRAM algorithm and ACADG algorithm converge to -1.
2) The convergence speed of the objective function in the ACADG algorithm is significantly faster than that of the AMSGRAD algorithm.
3) During the convergence process, the oscillation amplitude of the AMSGRAD algorithm is much larger than that of the ACADG algorithm.

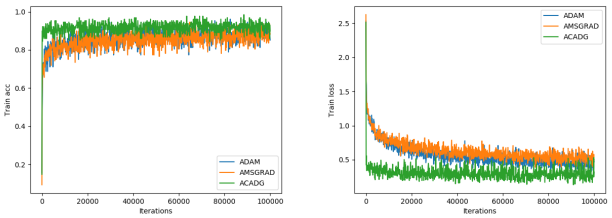

Figure 2: The training performance of the three algorithms on logistic regression model. The first and second images represent the accuracy and loss on Mnist training data set, respectively.

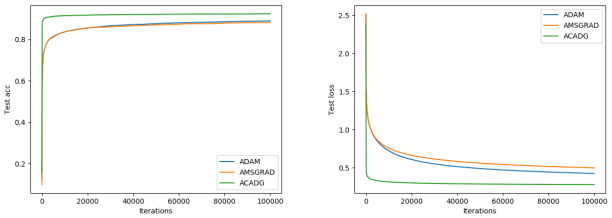

Figure 3: The test performance of the three algorithms on logistic regression model. The first and second images represent the accuracy and loss on Mnist test data set, respectively.

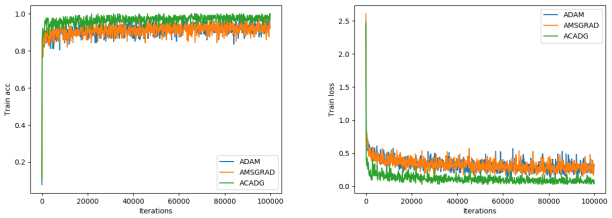

Figure 4: The training performance of the three algorithms on DNN model. The first and second images represent the accuracy and loss on Mnist training data set, respectively.

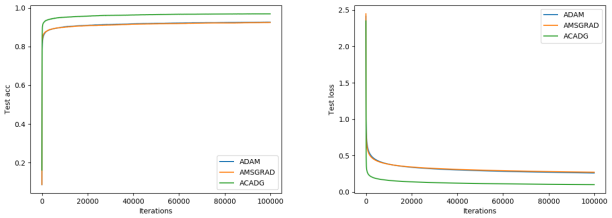

Figure 5: The test performance of the three algorithms on DNN model. The first and second images represent the accuracy and loss on Mnist test data set, respectively.

B. Logistic regression and DNN experiment

In order to verify the performance of ACADG algorithm on convex optimization problem and non-convex optimization problem. This paper uses the Mnist data set to perform experiments on logistic regression model and DNN model, respectively.

Mnist is a handwritten digital recognition image set, and which is commonly used in deep learning. It contains 60,000 black and white images with 28*28, and 50,000 samples in training data set, 10,000 samples in test data set, 10 classification labels.

Logistic regression model has only the input layer of

Table I: The performance of the three algorithms on logistic regression and DNN model.

Algorithm	Logistic regression		DNN	
	Train	Test	Train	Test
ADAM	88.08%	88.83%	92.62%	92.65%
AMSGRAD	87.05%	88.12%	92.34%	92.49%
ACADG	92.16%	92.28%	97.77%	96.91%

784 and the output layer of 10. DNN model has a hidden layer in addition to the input layer and the output layer in Logistic regression model, where the number of nodes is 100. In addition, the cross entropy loss function is used as the objective function in the experiment.

In the ACADG algorithm, the relevant experimental parameters η_0, β_1, β_2, m_0, v_0, $\triangle w_0$, T, ϵ, ε are set as 0.001, 0.9, 0.999, 0, 0, 0, 1e5, 1e-8, 1e-9, respectively. The experimental parameters related to the AMSGRAD and ADAM algorithms are consistent with those in ACADG algorithm, and the experimental results are shown as Flg.2-5 and Table1.

From the training loss of Fig.2, the ACADG algorithm reaches the convergence point of the model faster than the AMSGRAD and ADAM algorithms under the same number of iterations. For example, when the number of iteration steps is 20,000, the training loss of the ADAM and AMSGRAD algorithms is still in a decreasing state, but the ACADG algorithm has reached the convergence point and oscillated between the local optimal solution. From the test loss in Fig.3, the loss value of the AMSGRAD algorithm on the test data set is higher than that of the ADAM algorithm, which indicates that the lower learning rate in the AMSGRAD algorithm reduces the performance of the model. From Fig.4 and Fig.5, the ACADG algorithm has higher accuracy and lower loss than the AMSGRAD and ADAM algorithms in the training and test data sets.

As can be seen from Table1, the performance of Mnist data set on logistic regression model is as follows: the accuracy of ACADG algorithm on the training data set is 5.11% and 4.08% higher than that of AMSGRAD and ADAM algorithms, respectively; the accuracy of ACADG algorithm on the test data set is 4.14% and 3.45% higher than that of AMSGRAD and ADAM algorithms, respectively. The performance of Mnist data set on DNN model is as follows: the accuracy of ACADG algorithm on the training data set is 5.43% and 5.15% higher than that of AMSGRAD and ADAM algorithms, respectively; the accuracy of ACADG algorithm on the test data set is 4.42% and 4.26% higher than that of AMSGRAD and ADAM algorithms, respectively.

Therefore, whether it is a convex optimization problem or a non-convex optimization problem, the ACADG algorithm is superior to AMSGRAD and ADAM algorithms in terms of the convergence speed, the oscillation amplitude, the accuracy of training and test data sets.

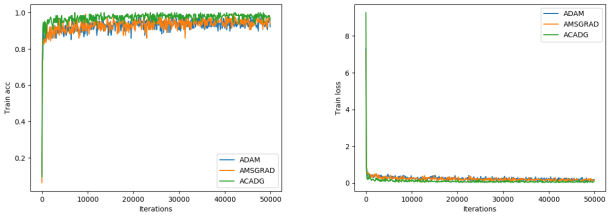

Figure 6: The training performance of the three algorithms on CNN model. The first and second images represent the accuracy and loss on Mnist training data set, respectively.

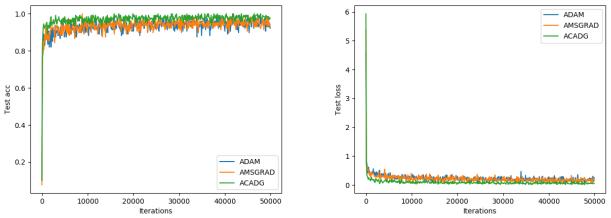

Figure 7: The test performance of the three algorithms on CNN model. The first and second images represent the accuracy and loss on Mnist test data set, respectively.

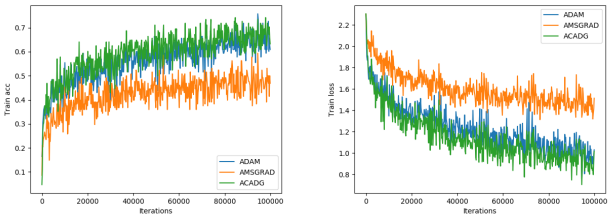

Figure 8: The training performance of the three algorithms on CNN model. The first and second images represent the accuracy and loss on Cifar-10 training data set, respectively.

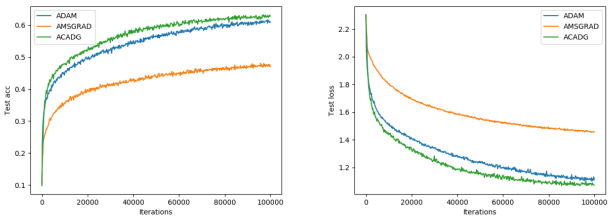

Figure 9: The test performance of the three algorithms on CNN model. The first and second images represent the accuracy and loss on Cifar-10 test data set, respectively.

C. CNN experiment

In order to verify the performance of ACADG algorithm on complex neural networks. This paper performs experiment on CNN model using the Mnist and Cifar-10 data sets, respectively.

Cifar-10 is a common data set for deep learning, which consists of 60,000 RGB color images of 32*32, and covering 10 categories: airplanes, cars, birds, furs, deer, dogs, frogs, horses, boats, trucks. The training data set includes 50,000 samples, and the test data set includes 10,000 samples.

Table II: The performance of the three algorithms on CNN model.

algorithm	Mnist data		Cifar-10 data	
	Train	Test	Train	Test
ADAM	96.09%	96.41%	61.72%	61.06%
AMSGRAD	96.88%	96.10%	49.22%	47.46%
ACADG	99.22%	99.22%	65.63%	63.05%

In the CNN model, this paper uses the dropout mechanism. Dropout means that some working nodes in the network will become inactive nodes with a certain probability during the model training. These inactive nodes can be temporarily not considered part of the network structure, but their weights must be preserved because they may become working nodes in the next iteration. In addition, the cross entropy loss function is used as the objective function in the experiment.

On the Mnist data set, the CNN model is composed of input[-1, 28, 28, 1], convolution[5, 5, 1, 32], pooling[2*2], convolution[5, 5, 32, 64], pooling[2*2], flat[7*7*64], hidden layer[1024], output[10]. The convolution operation has a step size[1,1,1,1], the convolution kernel[5,5], and the padding "SAME"; the pooling operation uses max_pool form, and the pooling step size[1,2,2,1], the padding "SAME".

On the Cifar-10 data set, the CNN model is composed of input[-1, 32, 32, 3], convolution[5, 5, 3, 16], pooling[2*2], convolution[5, 5, 16, 32], pooling[2*2], flat[8*8*32], hidden layer[100], dropout(0.5), output[10]. The convolution operation has a step size[1,1,1,1], the convolution kernel[5,5], and the padding "SAME"; the pooling operation uses max_pool form, and the pooling step size[1,2,2,1], the padding "SAME".

In the ACADG algorithm, the relevant experimental parameters η_0, β_1, β_2, m_0, v_0, $\triangle w_0$, $T_{Mnist}(T_{Cifar})$, ϵ, ε are set as 0.0001, 0.9, 0.999, 0, 0, 0, 1e5(5e4), 1e-8, 1e-9, respectively. The experimental parameters related to AMSGRAD and ADAM algorithms are consistent with those in ACADG algorithm, and the experimental results are shown as Flg.6-9 and Table2.

From Fig.6 and Fig.7, the ACADG algorithm has higher accuracy than the AMSGRAD and ADAM algorithms in the training and test data sets. From Fig.8 and Fig.9, the lower adaptive learning rate in the AMSGRAD algorithm seriously affects the accuracy of the model, but the ACADG algorithm is not affected by the historical maximum of the unbiased second-order moment estimation of the parameter gradient, and the performance of the ACADG algorithm is even better than that of the ADAM algorithm.

As can be seen from Table2, the performance of Mnist data set on CNN model is as follows: the accuracy of ACADG algorithm on the training data set is 2.44% and 3.13% higher than that of AMSGRAD and ADAM algorithms, respectively; the accuracy of ACADG algorithm on the test data set is 3.10% and 2.81% higher than that of AMSGRAD and ADAM algorithms, respectively. The performance of Cifar-10 data set on CNN model is as follows: the accuracy of ACADG algorithm on the training data set is 16.41% and 3.91% higher than that of AMSGRAD and ADAM algorithms, respectively; the accuracy of ACADG algorithm on the training data set is 15.59% and 1.99% higher than that of AMSGRAD and ADAM algorithms, respectively.

It can be seen from the above three experiments: in the experiments of convex optimization problem, non-convex optimization problem, complex model, different data sets, ACADG algorithm is superior to AMSGRAD and ADAM algorithms in the convergence speed, the oscillation amplitude, the accuracy of training and test data sets. In fact, after many experiments, this paper finds that ACADG algorithm is more robust to the model weights than ADAM and AMSGRAD algorithms.

V. Conclusion

This paper proposes ACADG algorithm, which is an adaptive gradient optimization algorithm. The ACADG algorithm not only can improve the convergence speed, suppress the oscillation amplitude of objective function, but also can improve the accuracy of training and test data sets. Through some comparative experiments, it is found that ACADG algorithm is superior to AMSGRAD and ADAM algorithms in above three aspects, whether it is convex optimization problems or non-convex optimization problems.

Acknowledgment

This work is supported by the Science & Technology project (41008114, 41011215, and 41014117). Corresponding author: Shiqun Yin, qqqq-qiong@163.com.

References

[1] Zhang, Sixin, A. Choromanska, and Y. Lecun. "Deep learning with Elastic Averaging SGD." (2014):685-693.

[2] Chakroun, Imen, T. Haber, and T. J. Ashby. "SW-SGD: The Sliding Window Stochastic Gradient Descent Algorithm." (2017):2318-2322.

[3] Zhang, Wei, et al. "Staleness-aware async-SGD for distributed deep learning." (2016):2350-2356.

[4] Qian, Ning. "On the momentum term in gradient descent learning algorithms." (1999):145-151.

[5] Mukherjee, Indraneel, et al. "Parallel Boosting with Momentum." (2013):17-32.

[6] Leimkuhler, Benedict, and X. Shang. "Adaptive Thermostats for Noisy Gradient Systems." (2015).

[7] Arablouei, Reza, S. Werner, and K. Dogancay. "Adaptive frequency estimation of three-phase power systems with noisy measurements." (2013):2848-2852.

[8] Kurbiel, Thomas, and S. Khaleghian. "Training of Deep Neural Networks based on Distance Measures using RMSProp." (2017).

[9] Mukkamala, Mahesh Chandra, and M. Hein. "Variants of RMSProp and Adagrad with Logarithmic Regret Bounds." (2017).

[10] Zeiler, Matthew D. "ADADELTA: An Adaptive Learning Rate Method." (2012).

[11] Kingma, Diederik, and J. Ba. "Adam: A Method for Stochastic Optimization." (2014).

[12] Sashank J. Reddi, Satyen Kale and Sanjiv Kumar. "On The Convergence of ADAM and Beyond." (2018).