On the convergence speed of AMSGRAD and beyond

Abstract—**In the best paper of ICLR(2018), which named "on the convergence of ADAM and beyond". The author proposes AMSGRAD algorithm, which guarantees that it is converging as the number of iterations increases. However, through comparison experiments, this paper finds that there are two problems in the convergence process of AMSGRAD algorithm. First, the AMSGRAD algorithm is easy to oscillate; secondly, the AMSGRAD algorithm converges slowly. The above two problems can be solved in the following two ways. When $g_{t-1} * g_t > 0$, the momentum term in Momentum algorithm will be added to AMSGRAD algorithm to speed up the convergence. When $g_{t-1} * g_t \leq 0$, the SGD algorithm will replace AMSGRAD algorithm to update the model weight, the first-order and second-order moment estimations of parameter gradient are recalculated, and the adverse effect of historical moving average parameter gradient on current parameter gradient is eliminated, and the oscillation amplitude of objective function is reduced. Therefore, this paper proposes ACADG algorithm, which not only can improve the convergence speed, suppress the oscillation amplitude of objective function, but also improve the accuracy of training and test data sets.**

Index Terms—**AMSGRAD algorithm, objective function, convergence speed, oscillation amplitude, ACADG algorithm.**

I. Background

At present, the most learning algorithms in deep learning are based on the idea of iterative. The purpose is to find a set of parameters, optimize the model weight, and make objective function reach a minimum. 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 and improve the accuracy of training and test data sets. The other is an adaptive method$^{[6-7]}$, mainly includes RMSPROP[8] algorithm, ADAGRAD[9] algorithm, ADADELTA[10] algorithm, ADAM[11] algorithm and AMSGRAD[12] algorithm. These algorithms can calculate adaptive learning rate for each iteration, and the adaptive adjustment of learning rate can reduce the loss 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 and 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 parameter gradient may reduce during the iterative process of ADAM algorithm, resulting in ADAM algorithm not converging. In addition, they proposed AMSGRAD algorithm, which can ensure AMSGRAD algorithm gets convergence by recording the historical maximum of second-order moment estimation of parameter gradient. However, this paper finds two problems in AMSGRAD algorithm. First, the AMSGRAD algorithm is easy to oscillation. Second, the AMSGRAD algorithm converges slowly. Therefore, this paper proposes ACADG algorithm, which is a new adaptive learning rate optimization algorithm. When $g_{t-1} * g_t > 0$, the momentum term in Momentum algorithm will be added into AMSGRAD algorithm, and the momentum term will improve the convergence speed. When $g_{t-1} * g_t \leq 0$, the SGD algorithm can be used to update the model weight, and the first-order and second-order moment estimations of parameter gradient will be recalculated to eliminate the adverse effect of historical moving average parameter gradient on current parameter gradient, so as to reduce the oscillation amplitude of objective function . And the main contributions of this paper are as follows:

1) This paper find two problems in the process of AMSGRAD algorithm convergence, one is the oscillation and the other is the rate of convergence.
2) This paper proposes ACADG algorithm, which successfully solved the above two problems.
3) Through synthesis experiment, logistic regression and DNN experiment, CNN experiment, it is verified that ACADG algorithm not only has faster convergence speed and 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 Momentum algorithm and its model parameter update process at different iteration stages. The second part introduces the model parameter update rule of AMSGRAD algorithm, and analyzes its two problems through the working process of AMSGRAD algorithm.

A. Momentum algorithm

In physics, people use momentum to simulate the inertia of an object as it moves. After deduction,it is found that the momentum thought can used to deep learning optimization algorithm, and which improved the convergence rate. Then Momentum algorithm is proposed, and the updating rules for the model weight are designed as follows:

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

Where ρ is the momentum factor parameter, η is the learning rate, g is the parameter gradient, w is the model weight.

After analysis, the iterative process of momentum algorithm can be divided into three stages. At the initial stage of iteration, when the angle between model weight difference vector in previous step and parameters gradient vector in current step is less than 90 degrees, that is $g_{t-1} * g_t \leq 0$, the momentum term can help accelerate the convergence speed; In the middle of iteration, the momentum factor can increase the update range of model weight and make the objective function jump out of the unsatisfactory local optimal solution as far as possible. At the later stage of iteration, when the direction between parameter gradient vector in previous step and parameter gradient vector in current step are different, the objective function will oscillate back and forth around the optimal value, and the momentum factor will reduce the update range of the model weight, suppress the oscillation amplitude of objective function, and make the objective function faster converge to the optimal solution.

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

B. AMSGRAD algorithm

AMSGRAD algorithm is a new exponential moving average gradient optimization algorithm, and the specific update steps of its model weight are as follows:

1) Calculate the biased first-order (m) and second-order (v) moment estimators of parameter gradient, respectively.
2) The biased first-order and second-order moment estimators of parameter gradient are used 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 weight with following formula:

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{v_{max} + \epsilon}} \widehat{m}_t$$

Where η is the learning rate, ϵ is a constant and used to prevent the denominator from being 0.

Through the derivation of mathematical formula, the author has proved that AMSGRAD algorithm can overcome the non-convergence problem in ADAM algorithm and ensure AMSGRAD algorithm reach convergence as the number of iterations increases. However, this paper found the following two problems from the convergence process of AMSGRAD algorithm:

1) In order to ensure the convergence of AMSGRAD algorithm, AMSGRAD algorithm must record the historical maximum value of unbiased second-order moment estimation of gradient parameters, and use it to calculate adaptive learning rate. However, the adaptive learning rate of AMSGRAD algorithm may decrease during the iterative process, which increases convergence time, reduces convergence speed, and reduces the accuracy of training and test data sets.
2) When $g_{t-1} * g_t \leq 0$, the angle between parameter gradient vector in previous step and parameter gradient vector in current step is greater than 90 degrees. Using AMSGRAD algorithm to update the model weight will cause the unbiased first-order moment estimate of parameter gradient in previous step to react to the current parameter gradient vector, and the reaction will last for a long time, even causing the objective function to experience severe oscillations during convergence.

III. ACADG Algorithm

Aiming at the two problems in AMSGRAD algorithm, this paper proposes the following ideas:

1) In order to alleviate the oscillation problem of AMSGRAD algorithm, the method of exponential attenuation learning rate can be adopted in model training.
2) When $g_{t-1} * g_t > 0$, a momentum term is added into AMSGRAD algorithm based on the idea of Momentum algorithm, which not only can help AMSGRAD algorithm to accelerate the convergence speed, but also can help the objection function to jump out of the unsatisfactory local optimal solution as far as possible and improve the accuracy of training and the test data sets.
3) When $g_{t-1} * g_t \leq 0$, whether AMSGRAD algorithm or ADAM algorithm, the first-order moment estimate of parameter gradient with previous step will have an inverse effect on the current parameter gradient vector. In order to eliminate this reaction, SGD algorithm can be used to update the model weight, and the first-order and second-order moment estimates of parameter gradient are recalculated. In this way, it can be ensured that the moving average parameter gradient with previous step is always positively correlated with the current parameter gradient vector, and the moving average parameter gradient with previous step always plays a positive role in the convergence speed. In addition, it also can reduce the oscillation amplitude of objective function.

Based on the above ideas, this paper designs ACADG algorithm, as shown below:

Algorithm 1 ACADG

```
Input: input parameters $\eta_0$, $\beta_1$, $\beta_1$, $m_0$, $v_0$, $T$, $\epsilon$, $\varepsilon$
Output: $w$
$\eta_0$, $\beta_1$, $\beta_1$, $m_0$, $v_0$, $T$, $\epsilon$, $\varepsilon$ represent the initial learning rate, the first-order moment estimation coefficient of parameter gradient, the second- order moment estimation coefficient of parameter gradient, the initial first-order moment estimation of parameter gradient, the initial second-order moment estimation of parameter gradient,a constant and used to prevent the denominator from being 0, the minimum target error accuracy, respectively.
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
        $w_t = w_{t-1} - \frac{\eta}{\sqrt{v_{max}+\epsilon}} \widehat{m}_t + \beta_1 \triangle w_{t-1}$
    if $f(w_t) \leq \varepsilon$ then
        return $w_t$
return $w_t$
```

IV. Experiments

In order to verify the effectiveness of ACADG algorithm, three experiments are conducted in this paper. The first is synthesis experiment, which uses different objective functions to verify the convergence of ACADG algorithm; Then is 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; and the last is CNN experiment, which uses different data sets to verify the performance of ACADG algorithm. In three experiments, both ADAM algorithm and AMSGRAD algorithm are used as comparison objects. And the convergence of algorithm, the convergence speed of model, the oscillation amplitude of objective function, the accuracy of training and test data sets are used as the basis for three algorithms. The specific code of all experiments 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 designed two objective functions, 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 iteration number, and p is the probability. As can be seen from the above settings, whether it is first objective function or second objective function, the minimum value is obtained at $x = 1$.

The experimental parameters in three algorithms are consistent, and ACADG(0.01, 0.9, 0.999, 0,0,1e7,1e-8,1e-9). The experimental results are shown as follows:

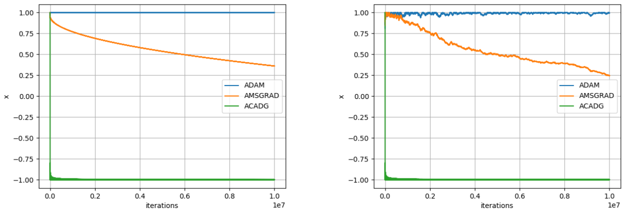

Fig. 1: The performance of three algorithms on synthesized data. The first and second images represent the results of first and second objective functions, respectively.

As can be seen from Fig.1, both first and second objective functions have the following three phenomena:

1) ADAM algorithm converges to 1, but AMSGRAM and ACADG algorithms will converge to -1 as the number of iterations increases.
2) The convergence speed of objective function in ACADG algorithm is obviously faster than that of AMSGRAD and ADAM algorithms.
3) In the process of convergence, under the condition of local optimal solution does not meet the requirements, the oscillation amplitude of AMSGRAD algorithm is much larger than that of ACADG algorithm.

B. Logistic regression and DNN experiment

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

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

Logistic regression only has the input layer with 784 and the output layer with 10. The DNN has a hidden layer, and the number of nodes is 100. In DNN, the dropout is used. Dropout means that the weights of some hidden layer nodes in network do not work randomly with a certain probability during model training, nodes that don't work can be temporarily considered not part of the network structure, but their weight must be preserved because it may become a working node on the next iteration.

The experimental parameters in three algorithms are consistent, and ACADG(0.001, 0.9, 0.999, 0,0,1e5,1e-8,1e-9), and

TABLE I: The performance of Mnist data set on logistic regression and DNN

algorithm	Logistic regression		DNN	
	Train	Test	Train	Test
ADAM	88.08%	88.83%	92.02%	92.52%
AMSGRAD	87.05%	88.12%	92.04%	92.34%
ACADG	92.16%	92.28%	97.32%	96.80%

the objective function is cross entropy loss function, and the probability of dropout in DNN is set to 0.9.The experimental results are shown as follows:

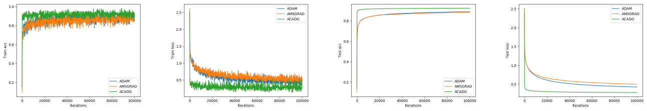

Fig. 2: The performance of Mnist data set on logistic regression. The left two graphs and the right two graphs show the accuracy and loss of three algorithms on training and test data sets, respectively.

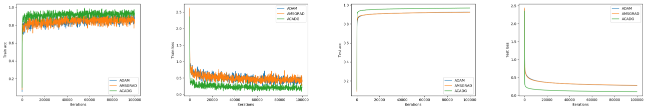

Fig. 3: The performance of Mnist data set on DNN. The left two graphs and the right two graphs show the accuracy and loss of three algorithms on training and test data sets, respectively.

It can be seen from training data set in Fig. 2 and Fig. 3 that ACADG algorithm performs better in the convergence speed, and the accuracy of training data set is higher than that of AMSGRAD and ADAM algorithms. From the results of test data set in Figure 2 and Figure 3, it can be seen that ACADG algorithm performs best in terms of the accuracy of test data set, and the performance of AMSGRAD algorithm is not as good as ADAM algorithm under the same number of iterations, this phenomenon also proves that AMSGEAD algorithm has a slower convergence speed, higher oscillation amplitude of objective function. As can be seen from Table1, the performance of Mnist data set on logistic regression is as follows: The accuracy of ACADG algorithm on training data set is 5.11% and 4.08% higher than that of AMSGRAD and ADAM algorithms, respectively. The accuracy of ACADG algorithm on 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 is as follows: The accuracy of ACADG algorithm on the training data set is 5.30% and 5.28% higher than that of AMSGRAD and ADAM algorithms, respectively. The accuracy of ACADG algorithm on training data set is 4.28% and 4.46% higher than that of AMSGRAD and ADAM algorithm, respectively.

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

C. *CNN experiment*

In order to study the performance of ACADG algorithm on complex neural network structures. This paper uses Mnist and cifar-10 data sets to perform experiment on CNN model, 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 and trucks, and 50,000 samples in training data set, 10,000 samples in test data.

The network structure to Mnist data set of CNN is 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] and output [10]. The convolution operation has a step size of [1,1,1,1], the convolution kernel is [5,5], and the padding is SAME; the pooling operation uses max_pool form, and the pooling step size is [1,2,2,1], the padding is SAME; the number of hidden layer nodes is 1024.

The network structure to cifar-10 data set of CNN is 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) and output [10]. The convolution operation has a step size of [1,1,1,1], the convolution kernel is [5,5], and the padding is SAME; the pooling operation uses max_pool form, and the pooling step size is [1,2,2,1], the padding is SAME; the number of hidden layer nodes is 1024.

The experimental parameters in three algorithms are consistent, and ACADG(0.0001, 0.9, 0.999, 0,0,5e4,1e-8,1e-9) to Mnist data set, ACADG(0.0001, 0.9, 0.999, 0,0,le5,1e-8,1e-9) to cifar-10 data set , and the objective function is cross entropy loss function. The experimental results are shown as follows:

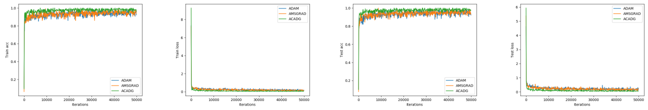

Fig. 4: The performance of Mnist data set on CNN. The left two graphs and the right two graphs show the accuracy and loss of three algorithms on training and test data sets, respectively.

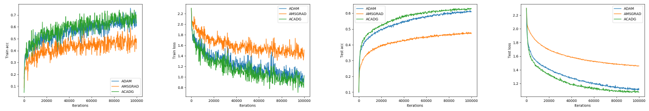

Fig. 5: The performance of cifar-10 data set on CNN. The left two graphs and the right two graphs show the accuracy and loss of three algorithms on training and test data sets, respectively.

As can be seen from Fig. 4 and Fig. 5 , whether it is Mnist data set or cifar-10 data set in CNN, the ACADG algorithm is superior to AMSGRAD and ADAM algorithms in the convergence speed, the oscillation amplitude, the accuracy of training and test data sets. As can be seen from Table 2, the performance of Mnist data set on CNN is as follows: The accuracy of ACADG algorithm on training data set is 2.44% and 3.13% higher than that of AMSGRAD and ADAM

TABLE II: The performance of mnist and Cifar-10 data sets on CNN

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%

algorithms, respectively. The accuracy of ACADG algorithm on 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 is as follows: The accuracy of ACADG algorithm on training data set is 16.41% and 3.91% higher than that of AMSGRAD and ADAM algorithms, respectively. The accuracy of ACADG algorithm on 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: under the conditions of convex optimization problem, non-convex optimization problem, complex model, different data sets. ACADG algorithm is superior to AMSGRAD and ADAM algorithm 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 parameters in the model than ADAM and AMSGRAD algorithms.

V. Summary

This paper proposes ACADG algorithm, which is an adaptive gradient optimization algorithm that can accelerate convergence. 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 comparison experiments, it is found that ACADG algorithm is superior to AMSGRAD and ADAM algorithm in above three aspects, whether it is convex optimization problem or non-convex optimization problem.

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. Do?an?ay. ”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).