International Journal of Latest Research in Engineering and Management" (IJLREM)
ISSN: 2456-0766
www.ijlrem.org ‖ Volume 06 Issue 02 ‖ February 2022 ‖ PP 01-10

Barzilai-Borwein step based AMSgrad algorithm for convolutional neural network models

[1]LingYun Wang

[1] *（School of mathematics and statistics, Nanjing University of Science and Technology, China.）*

Abstract: In this paper, we construct a new algorithm combining the Barzilai-Borwein step with the AMSgrad algorithm for solving convolutional neural network models. The convergence analysis of the new method is presented. Numerical experiments on convolutional neural network show the efficiency of the new algorithm.

Keywords: Barzilai-Borwein step size, convolutional neural network model, AMSgrad algorithm.

1. Introduction

With the development of machine learning and deep learning, deep neural network (DNN) [1] plays an important role in solving natural language processing (NLP), computer vision (CV) [2][3], target detection, and other problems. And its variants also can deal with many practical problems, for example, convolutional neural networks (CNNs) have been widely used in image classification, and VGG, GooleNet, and the residual neural network ResNet [2] models have improved the accuracy rate of image recognition, Neural Networks (RNNs) and its improved model Long Short Term Memory Neural Networks (LSTMs) have solved many engineering technical problems in natural language processing and achieved ultra-high accuracy levels.

On the other hand, with the development of neural networks, the corresponding parameters are growing dramatically. Then it becomes difficult to train complex neural network models and find the parameter values that approximate the optimal solutions. Therefore, it is important to find a suitable iterative algorithm to minimize the loss function and find the optimal parameter values.

Among the algorithms for convolutional neural network models, the gradient-type algorithmsare the popular way because they only need the first-order information and can parallelized easily. Stochastic gradient descent (SGD) [4-7] uses a stochastic gradient as the search direction, andit saves computational cost and improves iteration efficiency. But the traditional stochastic gradient descent (SGD) optimization is very sensitive under many conditions, it can lead to the termination with low precision. Many improved methods have been proposed for this kind problem, such as the SGD algorithm for momentum acceleration [8] and Newton's method [9], the AdaGrad method [10] and the RMSprop method [11] and so on. Afterwards the Adam method [13] combines the advantages of both AdaGrad and RMSprop to derive a more effective step size and becomes the most popular method. However，the Adam method can not converge to the minimal value of the objective optimization problem in some situations [14]. In 2018, AMSGrad method, an improved algorithm of Adam algorithm, was proposed [14]. The corresponding method modified the Adam algorithm by using the maximum of the second-order moment estimation of the gradient.

In addition, Barzilai-Borwein algorithm [15] is better than gradient descent method for many optimization problems[16]. It uses the information of both current iteration point and the previous iteration point to determine the step size alone the negative gradient direction. For the strictly convex quadratic minimization problem, Barzilai and Borwein [15] proved that the algorithm converges superlinearly when $n = 2$. As to the case $n > 2$, Raydan [17] obtained the global convergence results, and Dai and Liao [18, 19] further proved that the algorithm converges linearly. With the help of nonmonotone line search [20], Raydan applied the BB algorithm to solve the minimization problem for general nonquadratic functions.

Inspired the works above, we combine the BB step with the AMSgrad algorithm to obtaina modified algorithm for convolutional neural network. And we analyze the convergence of the new method under some suitable conditions.

This paper is organized as follows. In Section 2, we present the mathematical preliminaries. We propose the BB-AMS grad algorithm and analyzes its convergence in Section 3. Finally, we give the experiments results in Section 4.

2. Preliminaries

Let us consider the following optimization problem for solving the Convolutional Neural Network (CNN):

$$\min_{t\in T} f_t(\theta) \quad (2.1)$$

where $f_t(\theta):(t\in T)$ is a smooth convex function.

The AMS grad algorithm is an improvement of Adam algorithm [13], which effectively solves the problem that Adam algorithm does not converge in some cases, and also has a good effect on solving the problem (2.1). The update rule of the AMSgrad algorithm is as follows:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t,$$

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

$$\tilde{v}_t = \max(v_t, \tilde{v}_{t-1}),$$

$$\theta_t = \theta_{t-1} - \alpha_t \cdot \frac{m_t}{\sqrt{\tilde{v}}_t},$$

where $g_t = \nabla_\theta f_t(\theta_t)$, is the gradient value of the objective function at time *t*, $g_t^{\ 2}$ is the square of the gradient of the objective function, β_1 and $\beta_2 \in (0,1)$ are exponential decayrate, m_t is the biased first-order moment estimate of the gradient of the moment *t*, and v_t is the second-order biased origin moment estimate of the gradient at moment *t*.

Next, we introduce the Barzilai-Borwein algorithm [15]. The step size of this algorithm is calculated in the following format:

$$\alpha_{BB\_t} = \frac{<\Delta\theta, \Delta\theta>}{<\Delta g, \Delta\theta>} \quad (2.2)$$

where $\Delta\theta = \theta_t - \theta_{t-1}$ $\Delta g = g_t - g_{t-1}$. Alsothe step size α_{BB_t} can be obtained by another equation $\alpha_{BB\_t} = \min_\alpha \| \alpha\Delta\theta - \Delta g \|^2$ and get the BB long step, $\alpha_{BB\_t} = \frac{<\Delta g, \Delta g>}{<\Delta\theta, \Delta g>}$. In this paper, we only use (2.2).

3. The Barzilai-Borwein step based AMSgrad algorithm

In this section, we give the following modified AMSgrad algorithm based on Barzilai-Borwein step.

Algorithm 1: Barzilai-Borwein step based AMSgrad algorithm (BB-AMSgrad)

Parameter initialization: $f_t(\theta):(t\in T)$ Is the objective function, and the initial step size $\alpha=1$， $m_0=v_0=0$，

$t=1$(Initial time), select the initial point θ_1,

When the time t is the initial time 1:

$$\theta_2=\theta_1-\frac{1-\beta_1}{\sqrt{1-\beta_2}}\cdot\frac{g_1}{\|g_1\|_2}\ (3.1)$$

While (when the iteration termination condition is not satisfied and the time t is greater than 1) do:

$g_t=\nabla_\theta f_t(\theta_t)$,

$m_t=\beta_1 m_{t-1}+(1-\beta_1)g_t$, (3.2)

$v_t=\beta_2 v_{t-1}+(1-\beta_2)g_t^2$, $\tilde{v}_t=\max(\tilde{v}_{t-1},v_t)$,

$$\alpha_{BB\_t}=\frac{<\theta_t-\theta_{t-1},\theta_t-\theta_{t-1}>}{<\theta_t-\theta_{t-1},g_t-g_{t-1}>},$$

$$\theta_{t+1}=\theta_t-\alpha_t\cdot\alpha_{BB\_t}\cdot\frac{m_t}{\sqrt{\tilde{v}_t}},\ (3.3)$$

$t:+=1.$

We analyze the convergence of the new algorithm using the similar way in [14]. The regret function [12] is defined as follows:

$$R(T)=\sum_{t=1}^{T}[f_t(\theta_t)-f_t(\theta^*)],\ （3.4）$$

where $\theta^*=\arg\min_\theta\sum_{t=1}^{T}f_t(\theta)$ .Let $g_{t,i}$ Represents its *i*-th component of $g_t=\nabla_\theta f_t(\theta)$, andwe define the matrix $g_{1:t,i}=[g_{1,i},g_{2,i}...g_{t,i}]$, and denote $\gamma=\frac{\beta_1^2}{\sqrt{\beta_2}}$. Before present the convergence result, we show the following lemma needed.

Lemma 1 [14, Lemma 2]. If all the parameters satisfy the conditions of Theorem 1, then the following inequalities hold:

$$\sum_{t=1}^{T}\alpha_t\|\tilde{v}_t^{-\frac{1}{4}}m_t\|^2\leq\frac{\alpha\sqrt{1+\log T}}{(1-\beta_1)(1-\gamma)\sqrt{(1-\beta_2)}}\sum_{i=1}^{d}\|g_{1:T,i}\|_2\ (3.5)$$

Now we present the convergence properties of the BB-AMSgrad method.

Theorem 1. Assume that the function f_t has bounded gradients, i.e., $\|\nabla_\theta f_t(\theta)\|\leq G$, for all $\theta\in R^d$ and distance between any θ_t generated by BB_AMSgrad is bounded, $\| \theta_n - \theta_m \|_2 \leq D$ for any $m, n \in \{1, 2, ..., T\}$. Also assume that there exist $Q_1, Q_2 > 0$, such that $Q_1 \leq \alpha_{BB\_t} \leq Q_2$. And $\gamma = \frac{\beta_1^2}{\sqrt{\beta_2}} < 1$ for $\beta_1, \beta_2 \in [0,1)$ .Let $\alpha_t = \frac{1}{\sqrt{t}}$ and $\beta_{1,t} = \beta_1 \cdot \lambda^{t-1}, \lambda \in (0,1)$. Then for all $T \geq 1$, the regret function satisfies the following inequality:

$$R(T) \leq \frac{D^2\sqrt{T}}{dQ_1(1-\beta_1)} \sum_{i=1}^{d} \sqrt{\tilde{v}_{T,i-1}} + \frac{\sqrt{1+\log T} \cdot Q_2}{(1-\beta_1)^2 \sqrt{1-\beta_2}(1-\gamma)} \sum_{i=1}^{d} \| g_{1:T,i} \|_2 + \frac{\beta_1 D^2 G}{2d^{\frac{3}{2}}(1-\beta_1)\cdot(1-\lambda)^2} , \tag{3.6}$$

Proof: From the assumptions and the equivalence of the norms, we can havethat : $\| \theta_n - \theta_m \|_\infty \leq \frac{D}{\sqrt{d}}$ and $\| \nabla_\theta f_t(\theta) \|_\infty \leq \frac{G}{\sqrt{d}}$ for any $t \in T$.

By the formula (3.3) in Algorithm 1, We can have the following expansion：

$$\begin{aligned} \| \tilde{v}_t^{\frac{1}{4}} (\theta_{t+1} - \theta^*) \|^2 &= \| \tilde{v}_t^{\frac{1}{4}} (\theta_t - \alpha_t \cdot m_t \cdot \alpha_{BB\_t} \cdot \tilde{v}_t^{-\frac{1}{2}} - \theta^*) \|^2 \\ &= \| \tilde{v}_t^{\frac{1}{4}} (\theta_t - \theta^*) \|^2 + \alpha_t^2 \cdot \alpha_{BB\_t}^2 \cdot \| \tilde{v}_t^{\frac{1}{4}} \cdot m_t \|^2 \quad (3.7) \\ &\quad - 2\alpha_t \cdot \alpha_{BB\_t} \cdot m_t (\theta_t - \theta^*). \end{aligned}$$

From(3.7) and (3.2) in Algorithm 1, we can get

$$\begin{aligned} g_t(\theta_t - \theta^*) = & \frac{1}{2\alpha_t(1-\beta_{1t})\alpha_{BB\_t}} [\| \tilde{v}_t^{\frac{1}{4}} (\theta_t - \theta^*) \|^2 - \| \tilde{v}_t^{\frac{1}{4}} (\theta_{t+1} - \theta^*) \|^2 \\ & + \frac{\alpha_t \alpha_{BB\_t}}{2(1-\beta_{1t})} \| \tilde{v}_t^{\frac{1}{4}} m_t \|^2 + \frac{\beta_{1t}}{(1-\beta_{1t})} m_{t-1} (\theta_t - \theta^*) \\ \leq & \frac{1}{2\alpha_t(1-\beta_{1t})\alpha_{BB\_t}} [\| \tilde{v}_t^{\frac{1}{4}} (\theta_t - \theta^*) \|^2 - \| \tilde{v}_t^{\frac{1}{4}} (\theta_{t+1} - \theta^*) \|^2 \\ & + \frac{\alpha_t \alpha_{BB\_t}}{2(1-\beta_{1t})} \| \tilde{v}_t^{\frac{1}{4}} m_t \|^2 + \frac{\beta_{1t}}{2(1-\beta_{1t})} \cdot \alpha_t \| \tilde{v}_t^{-\frac{1}{4}} m_{t-1} \|^2 \qquad (3.8) \\ & + \frac{\beta_{1t}}{2\alpha_t(1-\beta_{1t})} \| \tilde{v}_t^{\frac{1}{4}} (\theta_t - \theta^*) \|^2, \end{aligned}$$

where the inequality in the above equation is obtained from the Cauchy-Schwarz inequality. Then

$$\begin{aligned} \sum_{t=1}^{T} [f_t(\theta_t) - f_t(\theta^*)] \leq & \sum_{t=1}^{T} g_t(\theta_t - \theta^*) \\ \leq & \sum_{t=1}^{T} [\frac{1}{2\alpha_t \alpha_{BB\_t} \cdot (1-\beta_{1t})} [\| \tilde{v}_t^{\frac{1}{4}} (\theta_t - \theta^*) \|^2 - \| \tilde{v}_t^{\frac{1}{4}} (\theta_{t+1} - \theta^*) \|^2] \quad (3.9) \\ & + \frac{\alpha_t \alpha_{BB\_t}}{2(1-\beta_{1t})} \| \tilde{v}_t^{-\frac{1}{4}} m_t \|^2 + \frac{\beta_{1t} \cdot \alpha_t}{2(1-\beta_{1t})} \| \tilde{v}_t^{-\frac{1}{4}} m_{t-1} \|^2 \\ & + \frac{\beta_{1t}}{2\alpha_t(1-\beta_{1t})} \cdot \| \tilde{v}_t^{\frac{1}{4}} (\theta_t - \theta^*) \|^2], \end{aligned}$$

where the second inequality is obtained by summing both sides of (3.8).

Thenfrom Lemma 1, we have that:

$$\sum_{t=1}^{T}\alpha_t \|\tilde{v}_t^{\frac{1}{4}} m_t\|^2 \le \frac{\sqrt{1+\log T}}{(1-\beta_1)(1-\gamma)\sqrt{(1-\beta_2)}}\sum_{i=1}^{d}\|g_{1:T,i}\|_2 .$$

Since the BB step is bounded by the assumption, we obtain

$$\sum_{t=1}^{T}\alpha_t \cdot \alpha_{BB\_t} \|\tilde{v}_t^{\frac{1}{4}} m_t\|^2 \le \frac{\sqrt{1+\log T}\cdot Q_2}{(1-\beta_1)(1-\gamma)\sqrt{(1-\beta_2)}}\sum_{i=1}^{d}\|g_{1:T,i}\|_2 . \tag{3.10}$$

Substituting (3.10) into the (3.9), we can obtain:

$$\begin{aligned}
\sum_{t=1}^{T}[f_t(\theta_t)-f_t(\theta^*)] &\le \sum_{t=1}^{T}[\frac{1}{2\alpha_t\alpha_{BB\_t}(1-\beta_{1t})}[\|\tilde{v}_t^{\frac{1}{4}}(\theta_t-\theta^*)\|^2 - \|\tilde{v}_t^{\frac{1}{4}}(\theta_{t+1}-\theta^*)\|^2] \\
&\quad + \frac{\beta_{1t}}{2\alpha_t(1-\beta_{1t})}\|\tilde{v}_t^{\frac{1}{4}}(\theta_t-\theta^*)\|^2] + \frac{\sqrt{1+\log T}\cdot Q_2}{(1-\beta_1)^2(1-\gamma)\sqrt{(1-\beta_2)}}\sum_{i=1}^{d}\|g_{1:T,i}\|_2 \\
&\le \frac{1}{2\alpha_1(1-\beta_1)}\|\tilde{v}_t^{\frac{1}{4}}(\theta_1-\theta^*)\|^2 \\
&\quad + \frac{1}{2(1-\beta_1)}\sum_{t=2}^{T}[\frac{\|\tilde{v}_t^{\frac{1}{4}}(\theta_t-\theta^*)\|^2}{\alpha_t\alpha_{BB\_t}} - \frac{\|\tilde{v}_{t-1}^{\frac{1}{4}}(\theta_t-\theta^*)\|^2}{\alpha_{t-1}\alpha_{BB\_t-1}}] \\
&\quad + \sum_{t=1}^{T}[\frac{\beta_{1t}}{2\alpha_t(1-\beta_1)}\|\tilde{v}_t^{\frac{1}{4}}(\theta_t-\theta^*)\|^2] \\
&\quad + \frac{\sqrt{1+\log T}\cdot Q_2}{(1-\beta_1)^2(1-\gamma)\sqrt{(1-\beta_2)}}\sum_{i=1}^{d}\|g_{1:T,i}\|_2 \\
&= \frac{1}{2\alpha_1(1-\beta_1)}\sum_{i=1}^{d}\tilde{v}_{1,i}^{\frac{1}{2}}(\theta_{1,i}-\theta_i^*)^2 \\
&\quad + \frac{1}{2(1-\beta_1)}\sum_{t=2}^{T}\sum_{i=1}^{d}(\theta_{t,i}-\theta_i^*)^2[\frac{\tilde{v}_{t,i}^{\frac{1}{2}}}{\alpha_t\alpha_{BB\_t}} - \frac{\tilde{v}_{t-1,i}^{\frac{1}{2}}}{\alpha_{t-1}\alpha_{BB\_t-1}}] \\
&\quad + \frac{1}{2(1-\beta_1)}\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\beta_{1t}(\theta_{t,i}-\theta_i^*)^2\cdot\tilde{v}_{t,i}^{\frac{1}{2}}}{\alpha_t} \\
&\quad + \frac{\sqrt{1+\log T}\cdot Q_2}{(1-\beta_1)^2(1-\gamma)\sqrt{(1-\beta_2)}}\sum_{i=1}^{d}\|g_{1:T,i}\|_2 \\
&\le \frac{D^2}{2\alpha_1(1-\beta_1)}\sum_{i=1}^{d}\tilde{v}_{1,i}^{\frac{1}{2}} + \frac{D^2}{2(1-\beta_1)}\sum_{i=1}^{d}[\frac{\tilde{v}_{T,i}^{\frac{1}{2}}}{\alpha_T\alpha_{BB\_T}} - \frac{\tilde{v}_{1,i}^{\frac{1}{2}}}{\alpha_1}] \\
&\quad + \frac{D^2}{2(1-\beta_1)}\sum_{t=1}^{T}\sum_{i=1}^{d}\frac{\beta_{1,t}\tilde{v}_{t,i}^{\frac{1}{2}}}{\alpha_t} + \frac{\sqrt{1+\log T}Q_2}{(1-\beta_2)(1-\gamma)\sqrt{1-\beta_2}}\sum_{i=1}^{d}\|g_{1:T,i}\|_2,
\end{aligned} \tag{3.11}$$

Where the first inequality of (3.11) isby the condition $\beta_{1,t}=\beta_1\lambda^{t-1}\le\beta_1$ and formula (3.1) in

Algorithm 1. The next equality follows from the expansion formula for the 2-norm. The second inequality follows from $\| \theta_n - \theta_m \|_2 \leq D, \| \theta_n - \theta_m \|_\infty \leq \frac{D}{\sqrt{d}}$.

From Equation (3.11), we have

$$R(T) \leq \frac{D}{2\alpha_T \alpha_{BB\_T} d(1-\beta_1)} \sum_{i=1}^{d} \tilde{v}_{T,i}^{\frac{1}{2}} + \frac{D^2}{2d(1-\beta_1)} \sum_{t=1}^{T} \sum_{i=1}^{d} \frac{\beta_{1t} \tilde{v}_{t,i}^{\frac{1}{4}}}{\alpha_t} + \frac{\sqrt{1+\log T} \cdot Q_2}{(1-\beta_1)^2 (1-\gamma) \sqrt{(1-\beta_2)}} \sum_{i=1}^{d} \| g_{1:T,i} \|_2$$

$$= \frac{D^2 \sqrt{T}}{dQ_1(1-\beta_1)} \sum_{i=1}^{d} \sqrt{\tilde{v}_{T,i}} + \frac{\sqrt{1+\log T} \cdot Q_2}{(1-\beta_1)^2 \sqrt{1-\beta_2} (1-\gamma)} \sum_{i=1}^{d} \| g_{1:T,i} \|_2 + \frac{\beta_1 D^2 G}{2d^{\frac{3}{2}}(1-\beta_1) \cdot (1-\lambda)^2},$$

Where the second equation is from $\beta_{1t} = \beta_1 \cdot \lambda^{t-1}$. The proof is complete.

4. Numerical experiments

We use numerical experiments to test our algorithm BB_AMSgrad and compare it with several other related algorithms, AMSgrad, RMSprop and Adagrad. The experiments are completed in python 3. 6.

We use convolutional neural network to implement the classification ofMnist dataset [21], color images CIFAR10 [22], CIFAR100 [22] and Caltech-101datasets[23], and we choose Alexnet model for the convolutional neural network model, which has 5 convolutional layers, 3 pooling layers (the pooling operations taken are maximum pooling methods) and 3 fully connected layers. We only perform gradient descent optimization on the last three fully connected layers, and the parameter we need to iterate in the fully connected layer is the weight matrix ω and bias. After the last layer of softmax processing we can get the cross-entropy loss function of the whole model, and the optimization problem can be expressed as:

$$\min_{\omega,b} - \sum\nolimits_{i=0}^{n} y_i \ln y_i^{'} \quad (4.1)$$

Where y^i representing the true class of the sample, $y_i^{'}$ represents the prediction class of the model.

The MNIST dataset is a commonly used dataset in the field of machine learning, which consists of 60000 training samples and 10000 test samples, each of which is a grayscale handwritten digital picture with 28 X 28 pixels. The data set has ten categories from 0 to 9.

The CIFAR-10 dataset consists of 60000 32x32x3 color images of 10 classes with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images.

The CIFAR-100 dataset consists of 60000 32x32x3 color images of 100 classes with 600 images per class. There are 50000 training images and 10000 test images.

Caltech-101 is a dataset composed of 101 different categories of color images, each category has 40 to 800 color images, and the size of each image is 300x200x3.

The parameters usedinalgorithm are set as：

$$\alpha_t = 1/\sqrt{t}, \beta_1 = 0.9, \beta_2 = 0.99.$$

The results show that on these datasets, The BB_AMSgrad algorithm can iterate to lower loss values compared to the other three algorithms.

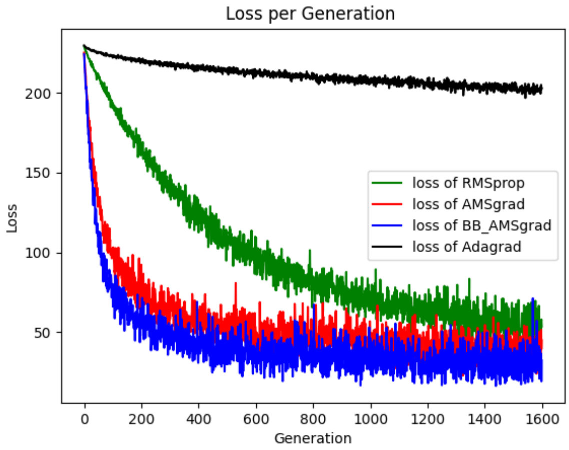

Fig 1: Iteration of the classification loss function on the Mnist dataset

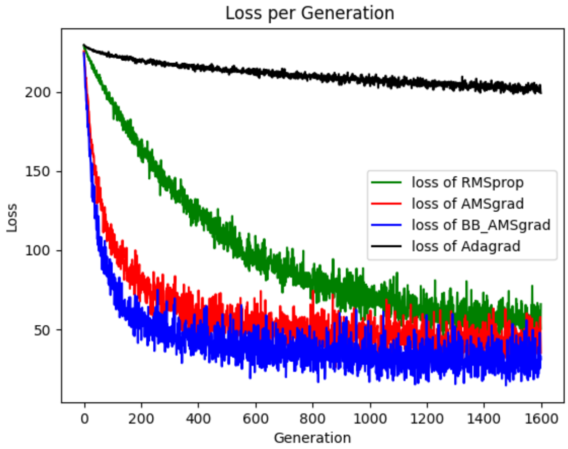

Fig 2:Iteration of loss function on CIFAR10 dataset

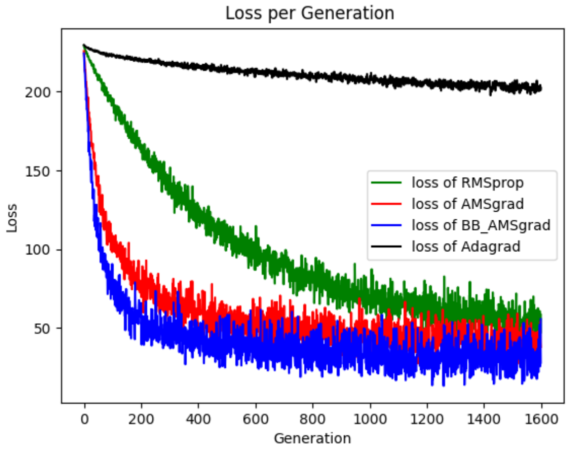

Fig 3: Iteration of loss function on CIFAR100 dataset

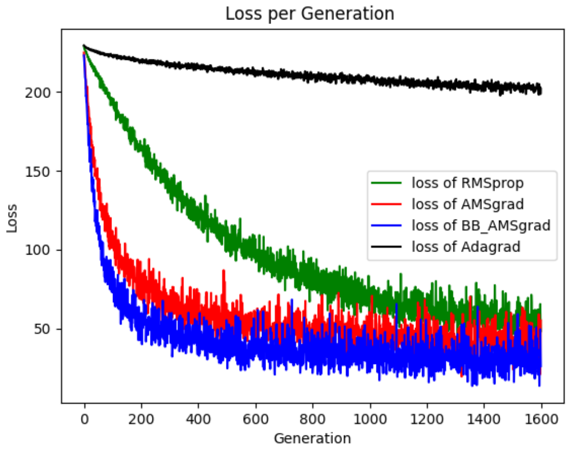

Fig 4: Iteration of loss function on Caltech-101 dataset

These figures show the behaviors of the algorithms for the loss function values defined by(4.1).Compared with RMSprop and Adagrad algorithms and AMSgrad algorithm，it is clear from the Figure 1-Figure 4 that our BB-AMSgrad algorithm has better results in experiments.

Finally, we give the Validation classification accuracy of these four algorithms on these fourdatasets. In this experiment, we run each algorithm four times, and the average of the values of accuracy is listedin the following table.

Table1: Validation classification accuracy

Optimization algorithm Data set	Mnist	CIFAR10	CIFAR100	Caltech-101
Adagrad	72.374%	64.248%	62.76%	64.25%
RMSprop	91.53%	86.352%	86.658%	87.32%
AMSgrad	92.232%	88.254%	86.854%	87.884%
BB_AMSgrad	92.7%	89.45%	88.54%	89.23%

It is obvious from this table that the BB_AMSgrad algorithm can achieve a higher Validation classification accuracy compared to the other three algorithms.

5. Conclusion

We presented the Barzilai-Borwein step with the AMSgrad (BB_AMSgrad) algorithm for solvingstochastic optimization problems in convolutional neural networks and showed its convergence. We compared BB_AMSgrad with AMSgrad, RMSprop and Adagrad. The numerical results show that our algorithm is advantageous in image classification tasks.In future work, we will analyze the boundedness of BB steps to obtain more stable convergence conditions.

6. Acknowledgements

The work was supported by the Nation Science Foundation of China (Grant no.11671205)

References

[1]. C Nicolò ，C Alex， Claudio G. *On the Generalization Ability of Online Learning Algorithms [J]*. IEEE Trans. Information Theory， 2004， 50: 2050-2057.

[2]. K He， X Zhang， S Ren， and Sun J. *Deep residual learning for image recognition[C].* Proceedings of the IEEE conference on computer vision and pattern recognition， 2016， 770-778.

[3]. K He， G Gkioxari， P Dollάr， and R Girshick . *Mask R-CNN[C]*. 2017 IEEE International Conference on Computer Vision， 2017， 2980-2988.

[4]. V Borkar. Stochastic approximation: a dynamical systems viewpoint [J]. *SIAM Review， 2009， 77: 306-306.*

[5]. L Bottou . Online algorithms and stochastic approximations[C]. *Online Learning and Neural Networks，* 1998， 9-42. D. Saad, Ed. Cambridge, UK:Cambridge University Press, 1998, revised, oct 2012. [Online].

[6]. A Nemirovski，A Juditsky，G Lan，Shapiro A. Robust stochastic approximation approach to stochastic programming[J]. *SIAM Journal on optimization， 2009， 19:1574-1609.*

[7]. H Robbins， S Monro. A Stochastic Approximation Method [J]. *Annals of Mathematical Statistics， 1951， 22(3):400-407.*

[8]. Q Ning. On the momentum term in gradient descent learning algorithms [J]. *Neural Netw， 1999， 12(1):145-151.*

[9]. Y Nesterov. A method for solving the convex programming problem with convergence rate O(1/k2)[J]. *Dokl Akad Nauk SSSR， 1983， 269:543 －547.*

[10]. John Duchi， Elad Hazan， Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization [J]. *Journal of Machine Learning Research， 2011 12:2121-2159.*

[11]. T Tieleman， G Hinton . *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude [J]*. COURSERA: Neural networks for machine learning， 2012，26-31.

[12]. ZinkevichMartin. *Online convex programming and generalized infinitesimal gradient ascent[C]*//Proceedings of the 20th international conference on machine learning (icml-03). 2003: 928-936.

[13]. P Diederik， B Jimmy. Adam: A Method for Stochastic Optimization [J]. *Computing Research Repository， 2014.*

[14]. J Sashank， Reddi， K Satyen， K Sanjiv. *On the Convergence of Adam and Beyond [J]*. Computing Research Repository， 2019.

[15]. J Barzilai， J Borwein . Two-point step size gradient methods [J]. *IMA journal of numerical analysis， 1988， 8(1): 141-148.*

[16]. H Iiduka. *Stochastic Fixed Point Optimization Algorithm for Classifier Ensemble [J]*. IEEE transactions

on cybernetics， 2019， 50(10):4370-4380.

[17]. M Raydan. On the Barzilai and Borwein choice of step length for the gradient method [J] *IMA journal of numerical analysis，1993，13(3):321-326.*

[18]. Y Dai， L Liao. R-linear convergence of the Barzilai and Borwein gradient method [J]. *IMA journal of numerical analysis， 2002， 22(1):1-10.*

[19]. Y Dai . A new analysis on the Barzilai-Borwein gradient method [J]. *Journal of the operational research society， 2013， 1(2):187-198.*

[20]. Y Dai，R Fletcher. On the asymptotic behaviour of some new gradient Methods[J]. *Mathematical programming， 2005， 103(3):541-559.*

[21]. data: http://yann.lecun.com/exdb/mnist/。

[22]. data: https://www.cs.toronto.edu/~kriz/cifar.html

[23]. data: http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html