Proceedings of the 4th International Conference on Computing and Data Science (CONF-CDS 2022)
DOI: 10.54254/2755-2721/2/20220515

An error based adaptive learning rate stochastic gradient descent algorithm in convolutional neural network

Li Qianyi

Dalian University of Technology, Dalian, Liaoning, China

li15524867927@163.com

Abstract. If the learning rate of convolutional neural network (CNN) is set improperly, the efficiency and accuracy of the algorithm will be greatly affected. To solve this problem, a learning rate adaptive algorithm is proposed to improve the traditional SGD: based on parameter prediction, the historical training error is used to update the learning rate. Under the condition of given initial learning rate, experiments on classical data sets prove the effectiveness of the above algorithms, and the adaptive learning rate stochastic gradient descent algorithm can keep the convergence of the network. Training accuracy is relatively stable; Shorter training time; And improve the learning accuracy.

Keywords: convolutional neural network, stochastic gradient descent, adaptive learning rate updating algorithm, parameter prediction.

1. Introduction

Convolutional neural network (CNN) is a kind of deep learning network, which is based on artificial neural network, and is the most widely used network structure in image recognition and speech recognition [1]. Gradient descent is a common method in optimization, including batch gradient descent, Stochastic gradient descent and small batch Stochastic gradient descent [2]. The Stochastic gradient descent algorithm is the fastest way to gain insight into the performance of the neural network. It can update the parameters of the model by randomly selecting a sample online.

Stochastic gradient descent (SGD) is a commonly used optimization algorithm in neural networks.Learning rate is one of the hyperparameters that is difficult to set in the model. SGD has excellent optimization effect due to the proper selection of learning rate. Improper setting of learning rate will directly affect the optimization effect [3]. There are disadvantages in using the global unified learning rate. For some parameters, the stochastic gradient descent algorithm has been optimized to near the minimum value, but some parameters still have large gradients.If the learning rate is large, optimization oscillation will occur.On the contrary, if the gradient is too small, the convergence speed is too slow and the computational efficiency is very low.

2. Convolutional neural network

2.1. Basic structure of convolutional neural network

The convolutional neural network is a deep neural network that has been upgraded from the Feedforward neural network, where each neuron only interacts with a neighboring local neuron. The convolutional

DOI: 10.54254/2755-2721/2/20220515

neural network consists of a convoluted layer, a pooled layer, and a fully connected layer, which are stacked up to form a convolutional neural network. There are three characteristics of convolutional neural network, namely, local receptive field, weight sharing and downsampling [4,5]. Local receptive field improves the ability of network to extract local features, weight sharing reduces the number of parameters in the network and reduces the complexity of computation, and weight sharing and downsampling are combined to reduce the complexity of network computation.

Convolution operation is a matrix transformation of image elements. Its input layer is connected with a small part of the input layer through a size convolution kernel, and then the local features of the region are extracted. Convolution reduces data processing redundancy and computational complexity by representing adjacent pixels in a single pixel. The convolution formula is as follows:

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l) \quad (1)$$

where,$M_j$represents the input feature graph, x_i^{l-1} represents the $(l-1)$ feature graph of layer's activation function value. k_{ij}^lrepresents the bias of the jth graph of the lth layer. The $f(*)$ on the outside is the activation function.

After extracting image features by convolution operation, it is necessary to classify the acquired features. Reduced sampling is a method to reduce the dimension of the feature extracted by the convolution kernel. After the pool layer is set in the convolution layer, the characteristic dimension of the output of the convolution layer is reduced through the pool process, and the over-fitting phenomenon is prevented while the parameters are reduced. The common pool operations in CNN include maximum pool, mean pool and random pool. After the pooling operation, the dimension of the network will be greatly reduced. The sampling formula is as follows:

$$x_j^l = f(\beta_j^l bool(x_j^{l-1}) + b_j^l) \quad (2)$$

where, the down-sampling function is represented by$bool(*)$. When summing all the pixels of $n * n$blocks in different regions of the input image, the desampling function can reduce the whole image by $n * n$times. The re-planning value is obtained by downsampling, multiplied by the bias coefficient β, and finally the value is output by an activation function.

The fully connected layer transforms the two-bit feature graph of the convolution output into a one-dimensional vector, and each node is connected with the node of the previous layer to synthesize the output feature of the previous layer. This layer contains the most weight parameters. A common convolutional neural network is to have one or two fully connected layers at the end of each layer, in order to highly refine the output features and make it easier for the final classifier to classify or regress.

2.2. Backpropagation

Neural networks have two basic operation modes: forward propagation and learning. Forward propagation is the process by which an input signal passes through one or more network layers in the previous section and is then output at the output layer. The goal is to estimate network parameters by training samples and expected outputs. For the convolutional neural network, the main parameters to be optimized are the convolution kernel parameters k, the downsampling layer parameters β, the full connection layer network weights ω, and so on. The core of the Backpropagation is to allow each network layer to generate computational errors, and to reverse the learning rules to make the network output closer to expectations [6].

The Backpropagation is based primarily on the gradient descent, where the network parameters are first initialized to a random value and then gradient descent in a direction that reduces the training error [7].

3. Convolutional neural network

3.1. Principle of stochastic gradient descent algorithm

For most supervised learning models in machine learning, in order to optimize the weight value, the cost loss function of the model should be created, and then the appropriate optimization algorithm should be

DOI: 10.54254/2755-2721/2/20220515

chosen to achieve the minimum value of the loss function. To get the minimum loss value of the function, we need to calculate the gradient of the loss function and reduce the loss value according to the descending direction of the gradient. The optimal solution is obtained by updating and adjusting the weight value to reach the minimum loss value. Stochastic gradient descent is an improved algorithm based on gradient descent. Instead of gradient descent for all samples, the computation is reduced by one sample update iteration at a time. Stochastic gradient descent has the advantages of fast training speed and good convergence, and is the most popular algorithm studied at home and abroad. The formula is as follows:

$$\begin{aligned} g(\varphi) &= \sum_{j=0}^{n} \varphi_j x_j \\ h(\varphi) &= \frac{1}{2m}\sum_{i=1}^{m}(y_i - g_\varphi(x_i))^2 \\ \varphi: &= \varphi - \eta\nabla_\varphi h(\varphi) \end{aligned} \tag{3}$$

φ represents the network parameter weight, ∇_φ represents gradient, $h(\varphi)$ as a loss function, $g(\varphi)$ is target function, y_i represents the ith sample's value， m represents the total number of times the entire iteration has occurred, η is learning rate. j shows the total number of parameters in CNN.

As the formula shows, the learning rate is very important for the gradient descent algorithm. If the learning rate is set too small, a large number of iterations will be needed to find the optimal solution, and the convergence rate is slow, but it will increase the probability of missing the optimal solution, and there may be no optimal solution [8]. It turns out that learning rate is the single most important factor that affects the effectiveness of Stochastic gradient descent. In order to adapt the selected learning rate to the Stochastic gradient descent, a Stochastic gradient descent algorithm with adaptive learning rate is proposed in this paper.

3.2. Adaptive learning rate updating algorithm based on stochastic gradient descent algorithm

So far, many methods have been proposed to set the learning rate, among which the adaptive learning rate method can set the learning rate most suitable to the problem. There are many kinds of adaptive learning rate algorithms. In 2011, Duchi [10] proposed AdaGrad (Adaptive Gradient Algorithm) , which can make better use of sparse Gradient information when data distribution is sparse, and can converge more effectively than traditional SGD. The iterative process is as follows:

$$\begin{aligned} g &= \frac{1}{m}\nabla_\theta \sum_i L(f(x_i;\theta), y_i) \\ r &:= r + g * g \\ \Delta\theta &= -\frac{\epsilon}{\delta+\sqrt{r}} * g \\ \theta: &= \theta + \Delta\theta \end{aligned} \tag{4}$$

In the fomula above, g is gradient, θ represents initial parameter, r is gradient cumulative variable, initial value is zero; δ was set as a small constant， the order of magnitude is about 10^{-7}.

e improved RMSProp algorithm is an adaptive learning rate method proposed by Geoff Hinton, but has not been published. In Adagrad algorithm, the learning rate in the denominator of learning rate will increase gradually, which makes the learning rate become very small at the later stage of iteration and is not helpful to find the final optimal solution. In 2014, Diederik Kingma of OpenAI and Jimmy Ba [11] of the University of Toronto combined Momentum with RMSProp to correct for the deviation and came up with the Adam method. The Adam method updates the estimates of biased first-order moments and second-order moments, and then eliminates their deviations. In addition to these classical methods, the researchers set up an adaptive learning rate method for a specific problem by using a specific parameter or error in the problem. In practice, the decreasing learning rate is often used to ensure the best convergence effect for both convex function and non-convex function. However, the artificial learning rate needs to set the list and threshold in advance according to the characteristics of the training data set

DOI: 10.54254/2755-2721/2/20220515

[12], otherwise it cannot reflect the characteristics of the data set well, so it cannot adapt to adaptive optimization.

4. Adaptive learning rate stochastic gradient descent

At present, the impulse is added to the SGD implementation in the common CNN network structure to prevent the continuous iteration at the minimum point of the loss function and the training can not be stopped [13]. This article also adds an impulse to the SGD on CNN, set to 0.5. In order to solve the problem of difficult selection of learning rate in SGD, many adaptive learning rate methods have emerged [14]. This paper proposes a method to update the learning rate by using the errors of the previous training data set to make the learning rate more suitable for network training. Compared with the non-adaptive learning rate SGD method, it solves the problem of optimizing oscillation or skipping the optimal solution caused by improper learning rate setting.

In the algorithm, the learning rate is defined as:

$$H = \frac{\eta_0}{\sqrt{\sum r+\delta}} \tag{5}$$

Wherein, η is the learning rate after each step update, η_0 is the learning rate of the initial input, r is the error generated after each iteration, and δ is the constant value.

5. Experimental results and analysis

5.1. Experimental data set and parameter setting

The data set selected for the experiment is the general object image data set of CIFAR-10 collected by Alex Krizhevsky et al. There are 60,000 true color images with size of 32 * 32 in 10 categories. Of these, 50,000 were used for training and 10,000 for testing.

In order to reduce the effect of oscillation on the algorithm, momentum = 0.5 is added. The initial learning rate η_0 is set to 0.5. Because the changes of η_0 and δ are in harmony, the experiment is carried out on a fixed basis η_0. If the parameter δ need to be modified in practical application, the value should not be too small, otherwise the learning rate will increase after the square root operation and can not converge.

5.2. Analysis of experimental results

There are many criteria to evaluate the algorithm, including the accuracy of the cross-validation set, the convergence of the optimization algorithm, the accuracy of the training set and the training time [12]. In this paper, the latter two algorithms are selected as the criteria for evaluation. The results of training with CNN on the CIFAR-10 dataset are shown in Figures 1 and 2. Analysis Figure 1, it is found that when the parameter $\delta = 10$, the accuracy of the error adaptive learning rate algorithm is slightly lower than that of the Stochastic gradient descent algorithm in the first 30 iterations due to the influence of initial learning rate η_0 and parameter δ. The accuracy of the adaptive learning rate algorithm is slightly lower than that of the non adaptive Stochastic gradient descent algorithm, but after several training iterations, the accuracy is much higher than that of the non adaptive learning rate algorithm, the adaptive learning rate Stochastic gradient descent algorithm is approximately 5% more accurate than the Stochastic gradient descent algorithm without adaptation. Figure 2 shows the gradual decay of the learning rate during training. When $\delta = 20$, the number of iterations is about 20 times more accurate than an algorithm that does not use adaptive learning rates, the difference in accuracy is not much different from $\delta = 10$. When $\delta = 30$, it needs more iteration times to reach the same accuracy curve intersection point, and the final result is not ideal. However, the disadvantage of modifying parameter δ is that the training time will be longer. The training time pairs are shown in Table 1.Because δ is set too high, the initial learning rate will be very small. In practical problems, too small learning rate will consume a lot of training time. Therefore, the parameter selection needs to be set according to different practical problems, and the appropriateδ will be selected to achieve the optimal effect.

DOI: 10.54254/2755-2721/2/20220515

Table 1. Experimental time comparison.

Algorithm name	Time
losssum_decay(δ = 10)	98.1476102
losssum_decay(δ = 20)	100.235130
losssum_decay(δ = 30)	105.660900
Adagrad	100.126781
RMSProp	100.447890
Adam	99.858080

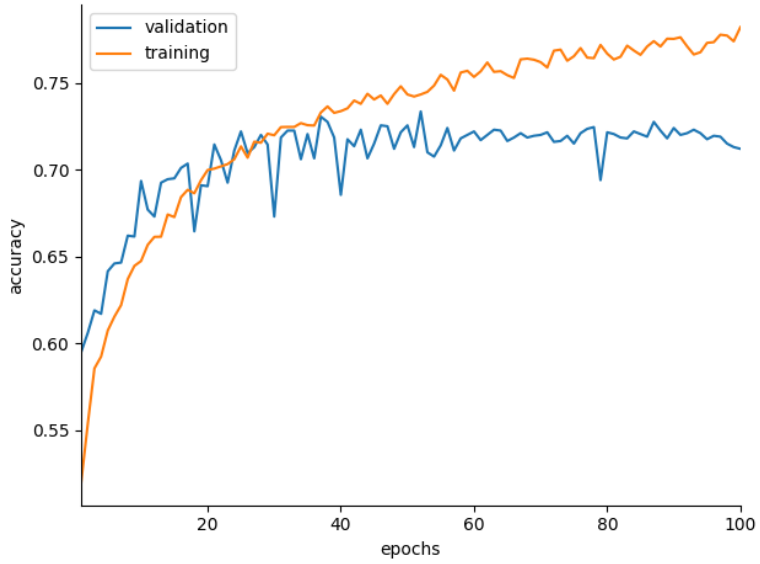

Figure 1. The error-based adaptive learning rate Stochastic gradient descent ($\delta = 10$) on the CIFAR-10 dataset.

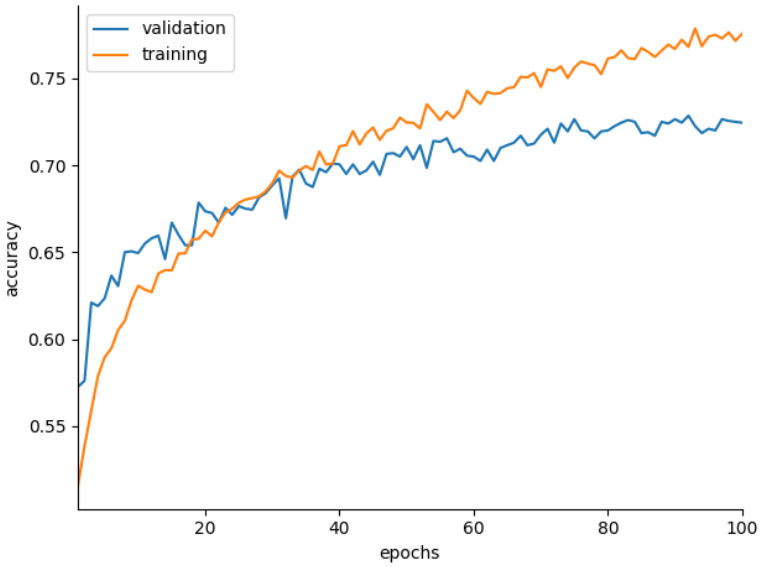

Figure 2. The error-based adaptive learning rate Stochastic gradient descent ($\delta = 20$) on the CIFAR-10 dataset.

DOI: 10.54254/2755-2721/2/20220515

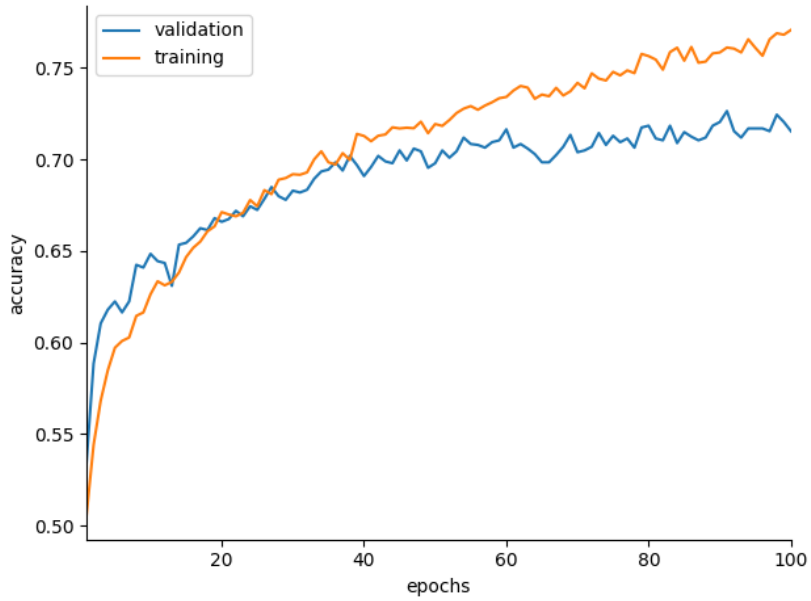

Figure 3. The error-based adaptive learning rate Stochastic gradient descent ($\delta = 30$) on the CIFAR-10 dataset.

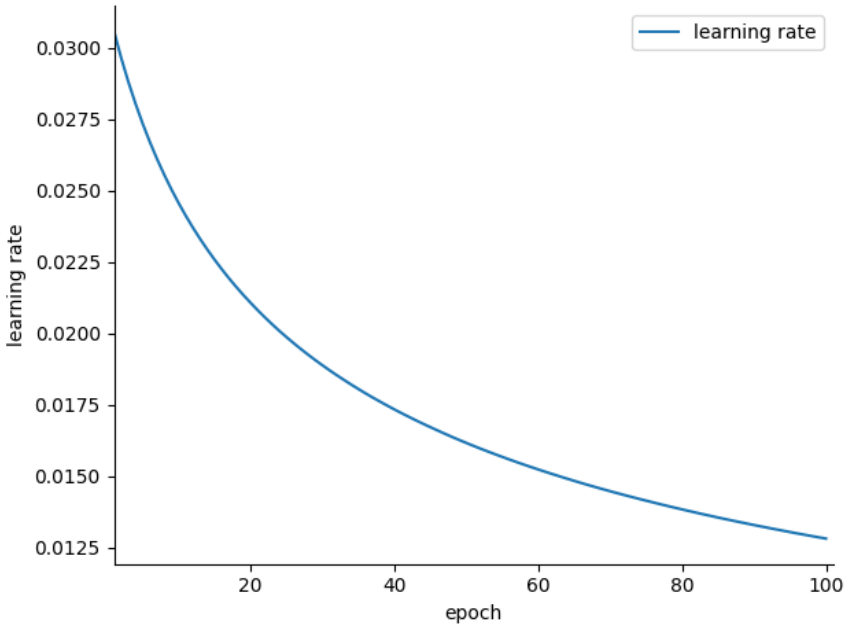

Figure 4. Learning rate decay curve of error adaptive learning rate Stochastic gradient descent algorithm ($\delta = 10$).

DOI: 10.54254/2755-2721/2/20220515

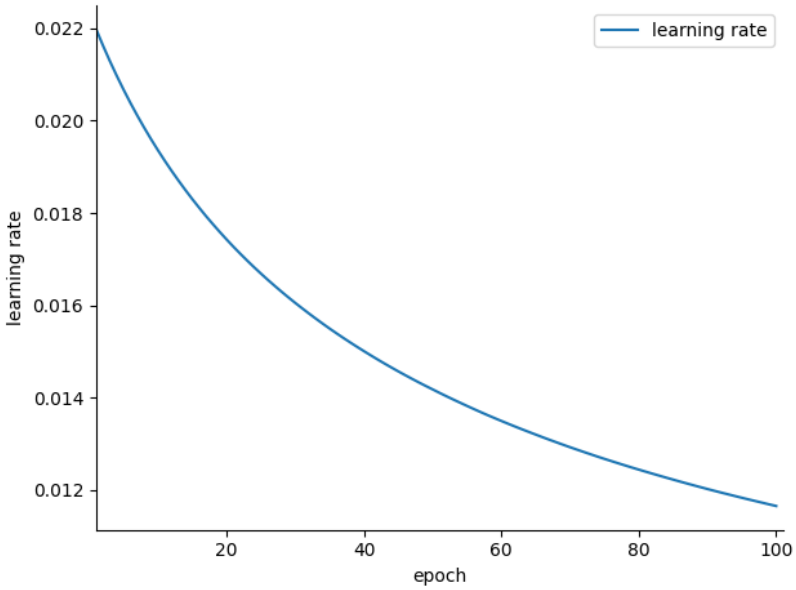

Figure 5. Learning rate decay curve of error adaptive learning rate Stochastic gradient descent algorithm ($\delta = 20$).

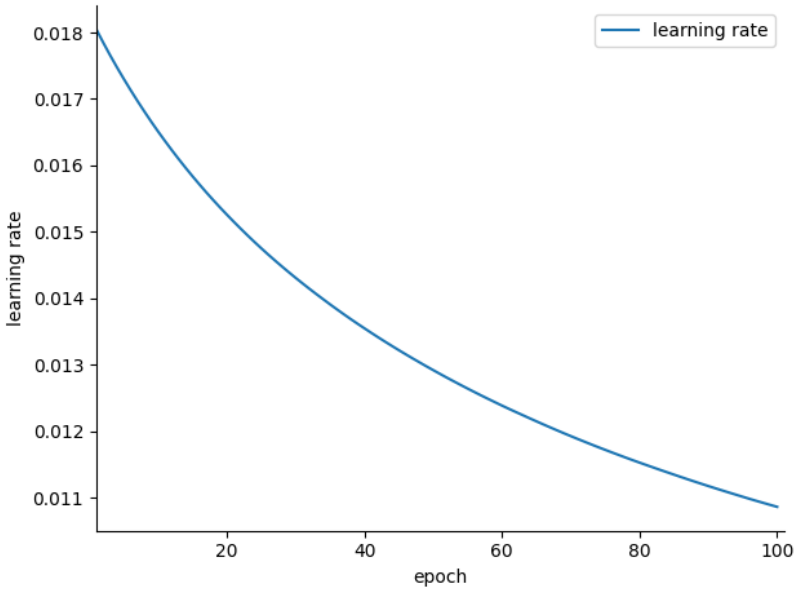

Figure 6. Learning rate decay curve of error adaptive learning rate Stochastic gradient descent algorithm ($\delta = 30$).

DOI: 10.54254/2755-2721/2/20220515

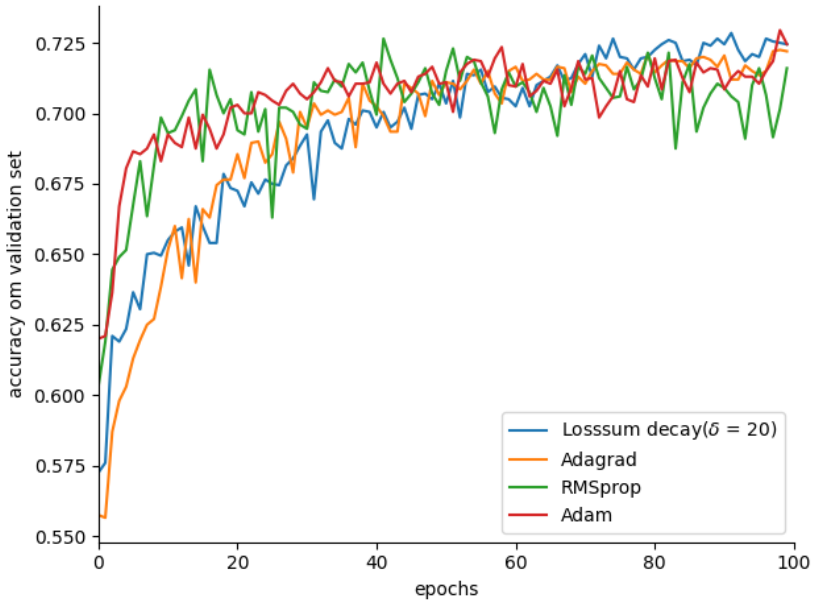

Figure 7. The accuracy curves ($\delta = 20$) of different adaptive learning rate algorithms on the CIFAR-10 dataset.

Figure 7 shows a comparison curve of the accuracy of different adaptive learning rate algorithms. The graph shows that the accuracy of the Stochastic gradient descent algorithm is in the middle of the centralized algorithm, but when the training times increase, the algorithm presented in this paper is the most stable.

6. Conclusion

In this paper, an error-based adaptive learning rate Stochastic gradient descent algorithm is used to test the CIFAR-10 dataset in the convolutional neural network architecture. And the adaptive learning rate $\eta = \frac{\eta_0}{\sqrt{\sum r+\delta}}$; η_0 is initial learning rate, r is the error generated during the previous iteration, δ is a constant. Control $\eta_0 = 0.5$ unchanged, Change the value of constant δ for numerical experiment, the algorithm is compared with the common adaptive learning rate algorithm for numerical stability, accuracy and training time. The experimental results show that the value of δ is has a great influence on the accuracy of training results, and its influence is non-monotonic. In comparison with other adaptive learning rate algorithms, the accuracy of this algorithm is in the middle level, but it has less oscillation compared with other methods.

References

[1] Lu Hongtao, Zhang Qinchuan. Applications of deep convolutional neural network in computer vision[J]. Journal of Data Acquisition and Processing, 31(1):1-17(2016).

[2] Sun Yanan, Lin Wenbin. Applications of gradient descent method in machine learning[J]. Journal of Suzhou University of Science and Technology (Natural Science Edition), 35(02):26-31 (2018).

[3] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization[J]. Journal of Machine Learning Research, 12(7):257-269 (2011).

[4] Alex Krizhevsky, Ilya Sutskever, Hinton GE. ImagNet classification with deep convolutional neural networks[C]// Advances in Neural Information Processing System. Cambridge: MIT Press, pp.1097-1105 (2012).

[5] Liang M, Hu X. Recurrent convolutional neural nerwork for object recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 367-3375 (2015).

DOI: 10.54254/2755-2721/2/20220515

[6] Dyda R O, Hart P E, Stork D G [Author], Li Hongdong, Yao Tianxiang[Translator]. Pattern Classification. Beijing: China Machine Press, (2003).
[7] Bouvrie J. Notes On Convolutional Neural Networks, MIT CBCL Tech Report, Cambridge, MA, (2006).
[8] Deng Xing, Deng Zhenrong, Xu Liang, et al. Optimized collaborative filtering recommendation algorithm[J]. Computer Engineering and Design, pp.37(5): 1259-1264 (206).
[9] Tomoumi Takase and Satoshi Oyama and Masahito Kurihara. Effective neural network training with adaptive learning rate based on training loss[J]. Neural Networks, 2018, 101: 68-78.
[10] Duchi J, Hazan E, Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization[J]. Journal of Machine Learning Research, 2011, 12: 2121-2159.
[11] King M, Diederik B, Adam J. A Method for Stochastic Optinmization[J] (2014).
[12] Wang Changsong, Zhao Xiang. General method for evaluating optimization algorithm and its application[J]. Journal of Computer Applications, pp. 30(A01): 76-79(in Chinese) (2010).
[13] Sutkever I, Martens J, Dahl G, Hinton G. On the importance of initialization and momentum in deep learning[C]// International Conference on Machine Learning, 2013: 1139-1147.
[14] Jin Haidong, Liu Quan, Chen Donghuo. An integrated stochastic gradient descent Q-learning method with adaptive learning rate[J]. Chinese Journal of Computers, pp. 42(10): 2203-2215. (2019).