A Neural Network Algorithm of Learning Rate Adaptive Optimization and Its Application in Emitter Recognition

Jihong Jiang[1], Yan Gou[1,2], Wei Zhang[1,3], Jian Yang[4], Jie Gu[3], and Huaizong Shao[1,4(✉)]

[1] University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, People's Republic of China
hzshao@uestc.edu.cn

[2] Southwest China Institute of Electronic Technology, Chengdu 610036, Sichuan, People's Republic of China

[3] Science and Technology on Electronic Information Control Laboratory, Chengdu 610036, People's Republic of China

[4] Peng Cheng Laboratory, Shenzhen 519012, Guangdong, People's Republic of China

Abstract. The setting of the learning rate in neural network training is very important. A too low learning rate will reduce the network optimization speed and prolong the training time while a too high learning rate is easy to exceed the optimal value, leading to the difficulty of model convergence. To solve this problem, based on the analysis of two common learning rate strategies, the attenuating learning rate and the adaptive learning rate, combined with the Adam algorithm, this paper proposes an adaptive learning rate algorithm based on the value of the current loss function and the previous one, and verifies the effectiveness of the algorithm by using the actual radiation source signal. The experimental results show that compared with the Adam algorithm, the number of network training iterations is reduced by 45.5% and the recognition accuracy has increased by 3.6%, which effectively improves the learning speed and reduces the training time.

Keywords: Neural network · Learning rate · Algorithm optimization · Emitter recognition · Application

1 Introduction

At present, the combination of deep learning and emitter identification is more and more closely. The realization of the recognition system is usually based on the construction and training of deep learning network model, and its goal is to minimize the loss function. The existing methods to improve the performance of deep learning systems are as follows: one is to optimize the network model structure, such as increasing the number of network

Supported by National Natural Science Foundation of China: NSFC61871092

D. Jiang and H. Song (Eds.): SIMUtools 2021, LNICST 424, pp. 390–402, 2022.
https://doi.org/10.1007/978-3-030-97124-3_29

layers, replacing simple neuron units with complex LSTM neurons [1]; the second is to optimize the initialization mode of the model to ensure that the early gradient has some beneficial properties [2], or has a lot of sparsity [3]. The third is to choose a better learning algorithm.

Gradient Descent algorithm [4] is a relatively simple and commonly used learning rate algorithm in machine learning. On this basis, researchers gradually put forward some extended methods, such as the Batch Gradient Descent algorithm [5], Stochastic Gradient Descent algorithm (SGD) [6], Mini-batch Gradient Descent algorithm [7], and Stochastic Gradient Descent with Momentum [8–10]. The Stochastic Gradient Descent algorithm with Momentum includes classical Momentum method and acceleration gradient algorithm based on momentum variation. In some basic gradient descent algorithms, the parameters are usually updated by a given global learning rate. It is difficult to optimize all variables to the minimum due to the different dependence of the variables to be optimized on the objective function. To solve this problem, researchers gradually optimize the method of updating the learning rate, and finally, the adaptive learning rate algorithm is born. For example, Jhon Duchi (2011) proposed AdaGrad algorithm [11], Tieleman, T. and Hinton, G. (2012) proposed RMSProp algorithm [12], Zeiler (2012) proposed AdaDelta algorithm [13], Kingma (2014) proposed Adam [14] algorithm, Timothy, D. (2016) proposed Nadam [15] algorithm, GH Wei et al. (2018) proposed ANGD algorithm [16].

This paper mainly analyzes the learning rate attenuation methods such as piecewise constant attenuation, exponential attenuation and cosine attenuation, as well as adaptive learning rate algorithms such as AdaGrad, RMSProp, and Adam. Combined with Adam algorithm, an algorithm of adaptive adjustment of learning rate based on the value of the current loss function and the previous one is proposed. Finally, the effectiveness of the algorithm is verified by the actual emitter signal.

2 Common Learning Rate Strategies in Neural Networks

At present, the learning rate strategies in deep learning are mainly divided into attenuation learning rate and adaptive change learning rate. The attenuation learning rate mainly includes piecewise constant attenuation, exponential attenuation, cosine attenuation and so on. Adaptive learning rate mainly includes AdaGrad algorithm, RMSProp algorithm, and Adam algorithm.

2.1 Attenuating Learning Rate

In the process of neural network training, the learning rate strategy, which decreases with the number of iterations, mainly includes piecewise constant attenuation, exponential decay, cosine attenuation, and so on [10].

1) Piecewise constant decay: Different learning rates are set at the defined training interval. The attenuation curve is shown in Fig. 1 (a). In the figure, the total number of training iterations is 50, and the learning rate constant is updated every 10 iterations.
2) Exponential decay: The learning rate is updated by exponential decay, and the update rules are as follows,

$$DLr = Lr \times Dr^{Gstep/Dstep} \tag{1}$$

where DLr is the learning rate after attenuation, Lr is the initial learning rate, Dr is the attenuation coefficient, $Gstep$ is the current number of training steps, $Dstep$ is the decay period. The attenuation curve is shown in Fig. 1(b). In the figure, the red line represents the ladder-type exponential decay. The green one is the standard exponential decay. The Lr is 0.5, the Dr is 0.9, and the $Dstep$ is 10.
3) Polynomial decay: It set an initial learning rate and a minimum learning rate. According to the polynomial attenuation rule, the learning rate gradually decreases from the initial value to the minimum value. When the learning rate decreases to the minimum value, the minimum value can be maintained for continuous training, otherwise the learning rate can be increased again to particular value, and then attenuated to the minimum value. The polynomial attenuation rules are as follows,

$$Gstep = minGstep, Dstep \tag{2}$$

$$DLr = Lr - ELr \times 1 - \frac{Gstep^{p}}{Dstep} + ELr \tag{3}$$

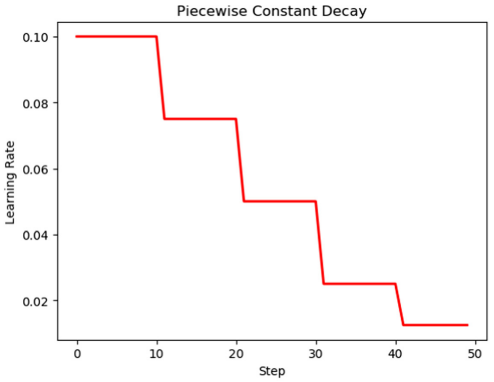
where ELr is the minimum learning rate. The default value is 0.0001, p is the polynomial power, the default value is 1.

If the learning rate decreases to the minimum value, continue to repeat the ascending and descending process, and the update formulas are as follows.

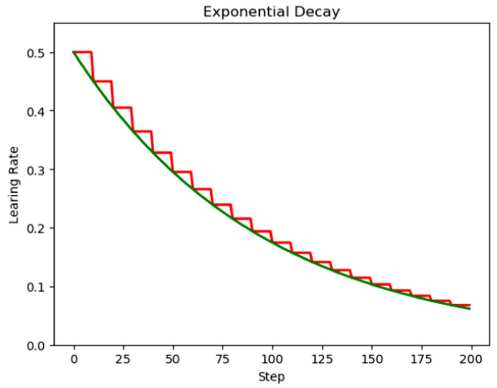
$$Dstep = Dstep \times ceil(G\text{step}/Dstep) \tag{4}$$

$$DLr = Lr - ELr \times 1 - \frac{Gstep^{p}}{Dstep} + ELr \tag{5}$$

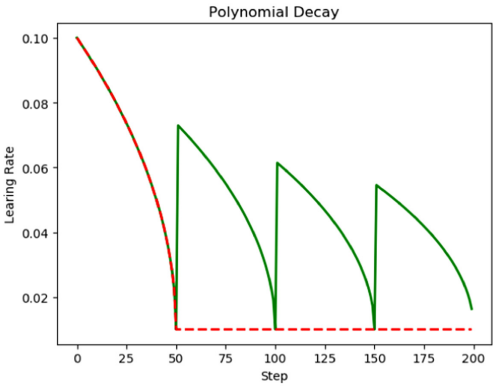
The polynomial attenuation curve of the learning rate is shown in Fig. 1 (c). In the figure, the Lr is 0.1, the ELr is 0.01, the $Dstep$ is 50, and the p is 0.5.



(a)



(b)



(c)

Fig. 1. Learning rate curve of different attenuating learning rate strategies: The abscissa represents the steps, and the ordinate represents the learning rate. The red line indicates that the learning rate does not rise after it decays to *ELr* and remains unchanged. The green line indicates that the learning rate will rise and fall after it decays to *ELr*. (a) piecewise constant decay; (b) exponential decay; (c) polynomial decay. (Color figure online)

2.2 Adaptive Learning Rate

The basic idea of adaptive learning rate is to make the learning rate automatically adapt to the parameters of the model through dynamic changes in the process of neural network model training. The commonly used adaptive learning rate algorithms mainly include the AdaGrad algorithm, RMSProp algorithm, Adam algorithm, and so on.

1) AdaGrad algorithm: The learning rate strategy in the algorithm is suitable for data with sparse features. For data with sparse features, it uses a higher learning rate to update parameters; for data with non-sparse features, it uses a lower learning rate to update parameters. It is assumed that in the classical stochastic gradient descent algorithm, any network parameters are updated with the same learning rate η. In the t^{th} training update, the gradient $g_{t,i}$ of the objective function to the parameters θ_i is as follows.

$$g_{t,i} = \nabla_\theta J(\theta_i) \tag{6}$$

The updating equation of network parameters is as follows.

$$\theta_{t+1,i} = \theta_{t,i} - \eta g_{t,i} \tag{7}$$

In the AdaGrad algorithm, each network parameter is updated with a different learning rate. The updating equation of network parameters is as follows,

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,i} + \varepsilon}} \cdot g_{t,i} \tag{8}$$

where ε is a smoothing parameter, which generally takes e^{-8}, η is the global learning rate and needs to be set manually. $G_{t,i}$ represents the cumulative square gradient of the previous t-step parameter θ_i.

$$G_{t,i} = G_{t-1,i} + g_{t,i}^2 \tag{9}$$

The vector representation of network parameter update rules is as follows.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \varepsilon}} \odot g_t \tag{10}$$

Although the AdaGrad algorithm can accelerate the gradient descent when dealing with sparse data, it is easy to lead to premature and excessive attenuation of the learning rate due to the continuous accumulation of the gradient square value of parameters from the beginning of training, so it cannot effectively update the network parameters.

2) RMSProp algorithm: RMSProp is an improved AdaGrad algorithm. Aiming at the problem that the learning rate of AdaGrad algorithm decays too fast, RMSProp introduces the attenuation coefficient. It improves the cumulative gradient sum of squares into an exponential decay moving average, which weakens the influence of

the historical gradient on the current learning rate. The exponential attenuation of the square of the gradient is calculated as follows,

$$E\left[g^2\right]_t = \gamma E\left[g^2\right]_{t-1} + (1-\gamma)g_t^2 \tag{11}$$

where t is the number of update iterations, g^2 is the gradient square value of network parameters, γ is the attenuation coefficient, which is generally taken as 0.9. The size of γ determines the influence of the historical gradient value on the current learning rate. The update rule of network parameter θ is as follows,

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E\left[g^2\right]_t + \varepsilon}} \odot g_t \tag{12}$$

where η is the global learning rate, the recommended value is 0.001, and g_t is the gradient of network parameters in the t^{th} iteration.

3) Adam algorithm: The basic idea of this algorithm is to calculate the first and second moment estimates of gradient, so that the learning rate can automatically adapt to the parameters of the network model. The gradient attenuation mode of Adam algorithm is as follows,

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t \tag{13}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2 \tag{14}$$

where m_t is the weighted average of the gradient, v_t is the weighted biased difference of the gradient, and the initialization value is 0. Both β_1 and β_2 are attenuation factors. When β_1 and β_2 approach to 1, m_t and v_t approach to 0. The bias correction of m_t and v_t is as follows.

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{15}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{16}$$

The renewal equation of network parameter θ is as follows, where t is iteration times, η is the global learning rate, and ε is the smoothing parameter.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}}\hat{m}_t \tag{17}$$

3 Learning Rate Algorithm Based on Loss Function

The piecewise constant attenuation strategy has higher requirements on artificial parameter adjustment. The learning rate strategies such as exponential attenuation and cosine attenuation have some problems such as slow convergence speed, serious time-consuming and easy to cross the optimal value. The AdaGrad algorithm is easy to lead to learning rate attenuation too fast and lack of certain robustness. The Adam algorithm usually needs to combine with the attenuation learning rate for model training, only using the default learning rate, the model cannot achieve the optimal convergence effect. Aiming at some existing problems, this paper proposes a learning rate algorithm.

3.1 Algorithm Design

At the beginning of training, the network parameters deviate from the optimal value greatly, and its loss function value is large. At this time, the higher learning rate can be used to accelerate the convergence speed of the model. In the later stage of training, the value of the loss function is close to the optimal value, and the value of the learning rate should be gradually reduced to avoid exceeding the optimal value. Therefore, this algorithm introduces a loss function on the basis of exponential attenuation, and adjusts the step factor of learning rate based on the current loss value and the previous loss value, so as to achieve the purpose of training optimization. The adaptive learning rate update formula is as follows,

$$lr(t) = p^t \beta_0 sigmoid[L(t)L(t-1) - d] \tag{18}$$

where t is the number of training iterations of deep convolution neural network, $lr(t)$ is the learning rate of the t^{th} iteration, p is the attenuation factor, β_0 is the initial amplitude, $L(\cdot)$ is the loss function, $sigmoid(\cdot)$ is the sigmoid function, d is the unit number of sigmoid function right translation, where the value is 8. The sigmoid function curve and the curve after right translation are shown in Fig. 2.

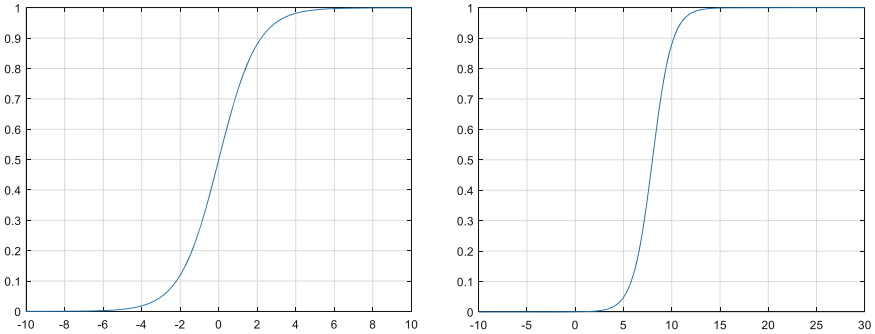


Fig. 2. Function image comparison: (a) sigmoid function curve; (b) sigmoid function curve after translation.

When the neural network is used to solve the classification problem, the loss function $L(t)$ selects the cross entropy loss function $H(p, q)$, and its expression is as follows,

$$H(p, q) = -\sum_x p(x) \log q(x) \tag{19}$$

$$L(t) = H(p, q) \tag{20}$$

where $p(x)$ is the probability distribution of the standard results, $q(x)$ is the probability distribution of the prediction results of the current network model, and x is the corresponding input of the current network model.

When the neural network is used to solve the regression problem, the loss function $L(t)$ usually adopts the mean square error $MSE(y, y')$, and its expression is as follows,

$$MSE(y, y') = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^{'}\right)^2 \tag{21}$$

$$L(t) = MSE(y, y') \tag{22}$$

where y_i represents the correct result of the i^{th} data in a batch data, and $y_i^{'}$ is the predicted value of the output of the neural network.

3.2 Parameter Updating of Neural Network

In the emitter recognition problem, the feature extraction function of Convolution Neural Network greatly simplifies the tedious engineering of constructing signal features under traditional conditions. In the training process of the convolutional neural network, the adaptive learning rate can effectively update the parameters of the convolutional neural network and accelerate the convergence of the network model. Note that the partial derivative of the loss function to the net input is $\partial L/\partial u = \delta$. The specific update process is as follows.

In the convolution layer, according to formulas (23) to (26), the variation of convolution kernel parameter Δk and addition bias Δb_a can be obtained respectively, and then the convolution kernel parameters and addition bias can be updated.

$$\frac{\partial L}{\partial K_{ij}^{l}} = \sum_{u,v}\left(\delta_j^l\right)_{uv}\left(p_i^{l-1}\right)_{uv} \tag{23}$$

$$\frac{\partial L}{\partial b_j^l} = \sum_{u,v}\left(\delta_j^l\right)_{uv} \tag{24}$$

$$\Delta k = -lr(t)\frac{\partial L}{\partial k_{ij}^l} \tag{25}$$

$$\Delta b = -lr(t)\frac{\partial L}{\partial b_j^l} \tag{26}$$

The i is the input neuron, j is the output neuron, l is the network layer number, u and v are the position coordinates of convolution or pooling operation in the input characteristic graph, δ is the partial derivatives of the loss function to net input, $lr(t)$ is the current learning rate, p represents the local region in the input characteristic graph that participates in convolution operation each time. The dot product of the partial derivative of the loss function to the net input of the j^{th} output neuron in layer l and the characteristic graph of the input of the i^{th} input neuron in layer $l-1$ is as follows.

$$\sum_{u,v}\left(\delta_j^l\right)_{uv}\left(p_i^{l-1}\right)_{uv} \tag{27}$$

In the pooling layer, according to formulas (28)–(31), the variation of multiplication bias $\Delta\beta$ and addition bias Δb can be obtained respectively, and then the multiplication bias and addition bias of the pooling layer can be updated. The formulas are as follows,

$$\frac{\partial L}{\partial \beta_j^l} = \sum\nolimits_{u,v} \left(\delta_j^l \circ down\left(x_j^{l-1}\right)\right)_{uv} \tag{28}$$

$$\frac{\partial L}{\partial b_j^l} = \sum\nolimits_{u,v} \left(\delta_j^l\right)_{uv} \tag{29}$$

$$\Delta\beta = -lr(t)\frac{\partial L}{\partial \beta_j^l} \tag{30}$$

$$\Delta b = -lr(t)\frac{\partial L}{\partial b_j^l} \tag{31}$$

Where the x is the characteristic graph, $down(\cdot)$ is the down sampling function, the symbol $\circ$ is the dot product.

In the fully connected layer, according to the formula (32)–(35), the change of the weight Δw and bias Δb can be obtained respectively, and then the w and b can be updated.

$$\frac{\partial L}{\partial w^l} = \delta^l x^{l-1} \tag{32}$$

$$\frac{\partial L}{\partial b^l} = \delta^l \tag{33}$$

$$\Delta w = -lr(t)\frac{\partial L}{\partial w^l} \tag{34}$$

$$\Delta b = -lr(t)\frac{\partial L}{\partial b^l} \tag{35}$$

If the parameters of each layer in the deep convolution neural network model reach the convergence goal, the training of the neural network model is completed. Otherwise, the learning rate is updated by the adaptive learning rate algorithm.

4 Experimental Analysis of Actual Radiation Source Data

In this paper, the experimental verification of adaptive learning rate adjustment algorithm based on loss function is completed on the actual signal of the emitter. At the same time, the convergence performance of different learning rate algorithms for the emitter identification model is compared and analyzed in terms of recognition accuracy, training iterations, and convergence value of loss function.

4.1 Experimental Description

Experimental Data Set. The original data consists of two batches of signal data generated by five communication stations in different environments. Among them, the first batch of data is used as the training set and the second batch of data is used as the test set. Each communication station contains 5 kinds of center frequencies and data of 5 transmission rates are collected at each center frequency. Therefore, a communication station contains 25 data files, and 5 sources provide 125 data files. The center frequency and transmission rate of each communication station are shown in Table 1.

Table 1. Description of center frequency and transmission rate of each radio station

Communication station ID	Center frequency (MHz)	Transmission rate (kbps)
1, 2, 3, 4, 5	225	64
	300	128
	380	192
	450	512
	512	1024

Data Processing. The original communication radio signal is one-dimensional signal in time domain. After preprocessing such as denoising, frame detection and effective data segment extraction, the two-dimensional time-frequency diagram is obtained by short-time Fourier transform. In this paper, the number of FFT points is 512, and the number of overlapping windows is 89. After STFT transformation, two channel time-frequency map is obtained, with the size of $257 \times 257 \times 2$. Then, the depth self-encoder is used for feature extraction and dimensionality reduction to obtain a $128 \times 128 \times 2$ time-frequency map. Finally, it is input into Convolutional Neural Network for feature extraction and classification recognition, and the classification results are output.

Network Model. The neural network model is the convolution neural network which includes 12 convolution layers and 6 pooling layers. The whole network model uses 3×3 convolution kernel. For the convolution layer in the middle, the same padding is used, the convolution layer close to the input and output adopts the valid padding, and the optimization algorithm adopts the Adam algorithm.

Parameter Setting of Learning Rate. In the adaptive adjustment algorithm of the learning rate based on loss function, it is recommended to set the parameters as $p = 0.99$, $\beta_0 = 1$, $b = 8$, which can be adjusted according to the actual training objectives.

4.2 Comparing with Other Learning Rate Algorithm

The algorithm proposed in this paper is compared with the RMSProp algorithm, Adam algorithm, and exponential decay learning rate combined with the Adam algorithm. Under different learning rate algorithms, the change of emitter individual recognition accuracy with iteration times is shown in Fig. 3.

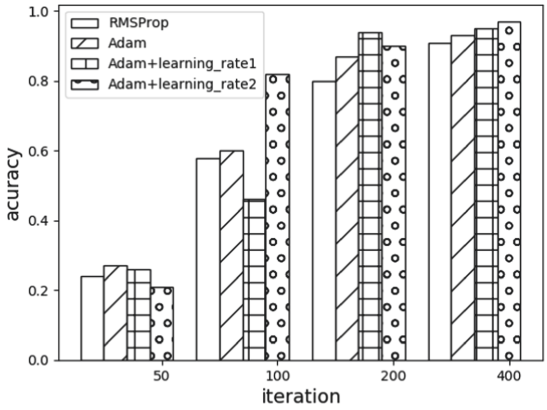


Fig. 3. Comparison of recognition accuracy of communication stations under four learning rate algorithms: the four learning rate algorithms are RMSProp, Adam, Adam with *learning_ rate1* and Adam with *learning_ rate2*.

Among them, *learning_ rate1* stands for exponential decay, *learning_ rate2* represents the learning rate based on the loss function. In the RMSProp algorithm, the attenuation coefficient is 0.9 and the global learning rate is 0.001. In the exponential decay learning rate algorithm, the initial learning rate is 0.001, the attenuation speed is 100, and the attenuation coefficient is 0.9. Among the four learning rate algorithms, the combination of learning rate based on loss function and Adam algorithm improves the recognition accuracy fastest, and finally achieves the highest recognition accuracy of emitter.The convergence performance of neural network model under different learning rate algorithms is shown in Table 2.

When the neural network model tends to converge, the training iterations needed by this algorithm are the least, which is 152 times, and the recognition accuracy is the highest, which is 97.48%. Experiments show that the convergence performance of the model is optimal under the learning rate algorithm. Compared with exponential decay learning rate algorithm, the number of iterations is reduced by 78 times and the recognition accuracy is improved by about 1%. Compared with the Adam algorithm, the number of training iterations is reduced by 127 times, and the recognition accuracy is improved by 3.6%. The results show that our algorithm can effectively improve the performance of the emitter individual identification model, which proves that the algorithm has a fast convergence speed and high recognition accuracy.

Table 2. Comparison of convergence performance of different learning rate algorithms for neural network model

Learning rate algorithm	Convergence performance		
	Epoch	Loss	Accuracy (%)
RMSProp	410	0.31	91.68
Adam	279	0.26	93.82
Adam + learning_ rate1	230	0.12	96.67
Adam + learning_ rate2	152	0.092	97.48

5 Conclusion

In this paper, the performance optimization of the learning rate algorithm for emitter identification is studied. Based on the analysis of attenuation learning rate and adaptive learning rate, a learning rate adaptive adjustment algorithm based on loss function is proposed. The algorithm aims to derive the size of step factor of current learning rate by calculating the value of the current loss function and the previous step one, so as to realize the fine control of the learning rate size by the loss function, and effectively adjust the change of the learning rate. Finally, the effectiveness of the algorithm is verified by the actual emitter signals. The experimental results show that our algorithm can improve the model convergence speed of the emitter individual identification tasks and reduce the training time.

References

1. Shi, X., Chen, Z., Wang, H., et al.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems, pp. 802–810 (2015)
2. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: International Conference on Artificial Intelligence and Statistics, vol. 15, pp. 315–323 (2012)
3. Le, Q.V., Jaitly, N., Hinton, G.E.: A simple way to initialize recurrent networks of rectified linear units. Computer Science (2015)
4. Nesterov, Y.: Introductory lectures on convex optimization. Appl. Optim. **87**, xviii, 236 (2004)
5. Meng, X., Bradley, J., Yavuz, B., et al.: MLlib: machine learning in apache spark. J. Mach. Learn. Res. **17**, 1235–1241 (2015)
6. Guenter, B., Dong, Y., Eversole, A., et al.: Stochastic gradient descent algorithm in the computational network toolkit (2013). Research.microsoft.com
7. Khirirat, S., Feyzmahdavian, H.R., Johansson, M.: Mini-batch gradient descent: faster convergence under data sparsity. In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC). IEEE (2017)
8. Srinivasan, V., Sankar, A.R., Balasubramanian, V.N.: ADINE: an adaptive momentum method for stochastic gradient descent (2017)
9. Shang, F., Liu, Y., Cheng, J., et al.: Fast stochastic variance reduced gradient method with momentum acceleration for machine learning (2017)

10. Li, S., Lu, X.: Static restart stochastic gradient descent algorithm based on image question answering. J. Comput. Res. Dev. **56**, 1092 (2019)
11. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 257–269 (2011)
12. Tieleman, T., Hinton, G.: Lecture 6.5—RMSProp, COURSERA: Neural Networks for Machine Learning. Technical report (2012)
13. Zeiler, M.D.: ADADELTA: an adaptive learning rate method (2012)
14. Kingma, D., Ba, J.: Adam: a method for stochastic optimization (2014)
15. Timothy, D.: Incorporating Nesterov momentum into Adam (2016)
16. Wei, W.G.H., Liu, T., Song, A., et al.: An adaptive natural gradient method with adaptive step size in multi-layer perceptrons. In: Chinese Automation Congress, pp. 1593–1597 (2018)