IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 27, 2020, accepted June 10, 2020, date of publication June 16, 2020, date of current version June 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3002884

PACL: Piecewise Arc Cotangent Decay Learning Rate for Deep Neural Network Training

HAIXU YANG[1,*], JIHONG LIU[1,*], HONGWEI SUN[1], AND HENGGUI ZHANG[2,3,4,5,6]

[1]College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
[2]International Laboratory for Smart Systems, Northeastern University, Shenyang 110004, China
[3]School of Physics and Astronomy, The University of Manchester, Manchester M13 9PL, U.K.
[4]Peng Cheng Laboratory, Shenzhen 518055, China
[5]Pilot National Laboratory for Marine Science and Technology, Qingdao 266237, China
[6]Key Laboratory of Intelligent of Computing Medical Image, Ministry of Education, Northeastern University, Shenyang 110004, China

Corresponding author: Jihong Liu (liujihong@mail.neu.edu.cn)

*Haixu Yang and Jihong Liu are joint first authors.

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61572152 (to HZ), and in part by the Science Technology and Innovation Commission of Shenzhen Municipality under Grant JSGG20160229125049615 and Grant JCYJ20151029173639477 (to HZ).

ABSTRACT Deep neural networks (DNNs) are currently the best-performing method for many classification problems. For training DNNs, the learning rate is the most important hyper-parameter, choice of which affects the performance of the model greatly. In recent years, some learning rate schedulers, such as HTD, CLR, and SGDR, have been proposed. These methods, some of which make use of the cycling mechanism to improve the convergence speed and accuracy of DNN, but performance degradation occurs in the convergence process. Others have good accuracy, but their convergence speed is too slow. This paper proposed a new learning rate schedule called piecewise arc cotangent decay learning rate (PACL), which can not only improve the convergence speed and accuracy of DNN but also significantly reduce performance degradation zone caused by the cycling mechanism. It is easy to implement, but almost at no extra computing expense. Finally, we demonstrate the effectiveness of PACL, on training CIFAR-10, CIFAR-100, and Tiny ImageNet with ResNet, DenseNet, WRN, SEResNet, and MobileNet.

INDEX TERMS Deep neural networks, learning rate schedulers, arc cotangent, optimization.

I. INTRODUCTION

Deep learning is an active field of machine learning. Its purpose is to establish a special deep neural network (DNN) [1]. DNN has demonstrated good performance in classification tasks [2]. However, its performance is greatly affected by the right choice of learning rates [3]. At present, deep learning uses gradient descent methods [4] to optimize learning rate parameters. Though many adaptive optimization algorithms are proposed in recent years [5]–[8], the essence of those methods is to improve the gradient descent method [9].

Learning rate [3], the step size of the gradient descent method in a search process [10], is an important hyper-parameter in training processes of deep learning model [11]. The convergence rate will be very slow, if the learning rate is set too small, and the model may fall into the local minimum. If the learning rate is set too large, it may lead the model to oscillate between output results [12]. As such, the final result of the model is greatly influenced by the learning rate [13].

Piecewise decay method may help to get an ideal result in theory, but the process of tuning the learning rate is tedious and time-consuming [14]. Although adaptive methods can help to adjust the learning rate of each iteration by itself, the final result is usually worse than piecewise decay [15].

There are many different learning rate schedulers proposed in the past [16]–[18]. In particular, with the cyclical learning rate (CLR) [16] and stochastic gradient descent with warm restarts (SGDR) [17] method, it has been demonstrated that compared with monotonically decreasing the learning rate, let the learning rate cyclically changes between reasonable boundaries can get better effect.

In this paper, we designed a new learning rate scheduler, called piecewise arc cotangent decay learning rate (PACL), which resets the learning rate and piecewise decay in each cycle. As compared with traditional learning rate schedules,

The associate editor coordinating the review of this manuscript and approving it for publication was Victor S. Sheng.

such as exponential and piecewise decay, PACL can greatly improve the convergence speed of networks. Compared with SGDR and CLR, PACL has a larger proportion of small learning rates, as such better accuracy and a more stable system can be achieved. In addition, it almost doesn't need extra computing expenses.

The contributions of this paper are:

1. A new learning rate scheduler is proposed. It can be an alternative to the existing schemes. The scheduler has the features of a warm restart, initializing the learning rate for every some epochs or iterations. It decays the learning rate with piecewise arc cotangent function, and has a smaller proportion of large learning rates and decays the learning rate rapidly in each cycle.

2. Some learning rate schedulers with cycling mechanisms have a large performance degradation zone in the convergence process. The PACL significantly reduces the performance degradation area caused by the cycling mechanism. The performance degradation area of PACL in each cycle is only one-third of the cycle.

3. PACL improves the convergence speed of the network and the convergence capability in the training process. DNN training with PACL has a faster convergence rate and higher classification accuracy. In addition, compared with the adaptive algorithm, it is easy to implement, and almost no extra computing expense.

The structure of the paper is as follows. Section II reviews some optimizers and learning rate schedulers proposed in the past. Section III describes the proposed PACL scheduler. Section IV shows the experiment results of PACL against other learning rate schedulers on different networks and datasets. Section V concludes the contributions of this paper and discusses some possible future works.

II. RELATED WORKS AND MOTIVATIONS

In this section, we review some optimizers like stochastic gradient descent (SGD) [19], and SGD with momentum [20]. Then we review some common learning rate schedulers proposed in recent years, such as the stochastic gradient descent with warm restarts (SGDR) [17], and cyclical learning rate (CLR) [16].

A. OPTIMIZERS

Training DNN is usually considered as the non-convex optimization problem [14], with which a loss function is first defined and then minimized by the optimization algorithm.

Gradient descent, originally proposed by Cauchy in 1847 [21], [22], is an iterative optimization algorithm for finding the minimum of a function. To find such a minimal of a function using gradient descent, one takes steps proportional to the negative of the gradient of the function at the current point. The excellent performance of deep learning is attributable to the gradient descent optimization algorithm.

Stochastic gradient descent (SGD) [19], becomes an extension of the gradient descent, originated from the stochastic approximation proposed by Robbins and Monro [4] in 1951 and was initially applied to pattern recognition [23] and neural network [24]. In recent years, with the rapid rise of deep learning, SGD has become a mainstream and very effective method to solve machine learning optimization problems. The parameters θ of a deep neural network update by stochastic gradient descent (SGD) is as follows

$$\theta_{t+1} = \theta_t - \frac{\alpha_t}{n} \sum_{i=1}^{n} \nabla f_i(\theta_t) \tag{1}$$

where α_t denotes the learning rate, which is used to adjust the amplitude of the parameter update. $f_i(\cdot)$ is the loss function of the i-th sample with respect to θ_t. The updating process of SGD is simple and efficient, and the iteration cost is independent on the total sample. But there is inevitably noise in the actual data so that it is difficult for SGD to approach the minimum in the best direction.

The random classical momentum algorithm (CM) [25] adds momentum term based on SGD. The historical parameter changes are integrated to speed up the optimization process. Momentum is designed to accelerate DNN training. But CM has a problem: it keeps accumulating speed and may miss the optimal solution.

B. LEARNING RATE SCHEDULERS

The learning rate is an important hyper-parameter in deep learning. Therefore, how to choose the learning rate has become the most important issue. Common learning rate schedules include time-based decay [26], piecewise decay [15], and exponential decay [15].

The piecewise decay drops the learning rate by a factor of every few epochs. Generally, the learning rate is reduced to half or one-tenth for every 10 epochs. Another schedule commonly used is exponential decay which updates the learning rate as the following [15]:

$$lr = lr_0 * e^{-kt} \tag{2}$$

where lr_0 denotes the initial learning rate. k is decay rate, t is iteration number.

Adjusting the learning rate manually is an expensive process, and it is difficult to find the best learning rate under the current model quickly. Therefore, many adaptive methods are proposed in recent years, such as Adagrad [5], Adadelta [6], RMSProp [7], and Adam [8]. Adagrad is a sub-gradient method that can incorporate the gradient information in earlier iterations. The update rule for Adagrad is as follows:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{G_t + \varepsilon}} g_t \tag{3}$$

where α is a global learning rate shared by all dimensions, g_t is the gradient in the t iteration. G_t is the sum of the squares of the past gradients to all parameters θ. ε is a smoothing term that avoids division by zero (usually on the order of 10^{-8}). Adagrad overcomes the trouble of manually adjusting the learning rate. But the optimization efficiency in the later stage of training is very low because Adagrad accumulates a lot of historical gradients, as a result, makes the learning rate too small. To solve the problem, Adadelta,

an improved version of Adagrad, makes the gradient decay exponentially following the time in training to avoid the continuous reduction of the learning rate. In Adadelta, we don't need to set the default learning rates, because we use the ratio of the running average of the previous time step to the current gradient. Adam, an efficient algorithm for gradient-based optimization of the stochastic objective function, combines the advantages of Adagrad and RMSProp, which is suitable for large data sets and high-dimensional spaces.

These adaptive algorithms have been successfully applied to various practical problems, especially Adam has become one of the most popular algorithms for neural network training. But some studies have pointed out that the generalization capability of these adaptive algorithms is worse than that of SGD in many applications [27], [28]. After Adaptive Learning Rates, SGDR [17] and CLR [16] were proposed which have better generalization capability than adaptive algorithms. Besides experimentations with Adaptive Learning Rates are computationally expensive which CLR is not.

Stochastic gradient descent with warm restarts (SGDR) [17] improves the performance of SGD. SGDR used warm restart mechanisms to initialize the learning rate every some epochs or iterations, and it decays the learning rate with a cosine annealing for each batch. SGD with warm restarts requires 2 to 4 fewer epochs than the common learning rate schedule schemes to achieve comparable or even better results.

Cyclical learning rates (CLR) [16] is similar to the SGDR method. Instead of monotonically decreasing the learning rate, CLR lets the learning rate cyclically change between reasonable boundaries. Allowing the learning rate to rise and fall in training will have a temporary negative impact to the network, but it is beneficial overall.

C. MOTIVATIONS

Exponential and piecewise decay are widely used in the training of state-of-the-art DNN architectures. The idea of both exponential and piecewise decay is to set an initial value for the learning rate and allows it to decay with some algorithms. The discrete change of learning rate makes the change of learning performance discrete and sudden, which shows that it is possible to improve learning performance steadily by changing the learning rate constantly [18].

Intuitively, with the increase of training iterations, we should keep the learning rate decreasing to reach convergence. However, it may be more useful to use a learning rate that changes periodically in a given range. Because the periodic high learning rate can make the model jump out of the local minimum and saddle point in the training process.

Dauphin *et al.* [29] pointed out that the saddle point is more difficult to converge than the local minimum. If the saddle point happens at an ingenious equilibrium point, a small learning rate usually does not produce a large enough gradient change to make it skip the point. This is the advantage of the periodic high learning rate, which can make the model skip the saddle point faster.

The effect of SGDR and CLR demonstrated that instead of monotonically decreasing the learning rate, letting the learning rate cyclically rises and fall in training will improve classification accuracy and rate of convergence.

Motivated by the formerly mentioned methods with the use of piecewise decay and cyclical learning rate, we designed a new learning rate scheduler which implements cycle mechanism and piecewise decay according to arc cotangent.

III. PIECEWISE ARC COTANGENT DECAY LEARNING RATE (PACL)

This section introduces a new scheduling method, named piecewise arc cotangent decay learning rate (PACL).

A. THE PROPOSED PACL

Fig.1 shows the decay model of piecewise arc cotangent decay learning rate (PACL), which controls the learning rate according to

$$Lr = Lr_{min} + (Lr_{max} - Lr_{min}) * \frac{arccot\,(T_i) - arccot(T_{fin})}{\frac{\pi}{2} - arccot(T_{fin})} \quad (4)$$

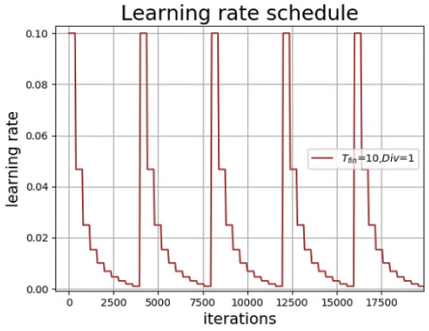


FIGURE 1. The decaying of PACL.

where Lr_{min} and Lr_{max} are ranges for the learning rate. T_{fin} represents total epochs or iterations in a cycle. T_i denotes how many epochs or iterations have been performed in a cycle. $Lr = Lr_{max}$ when $T_i = 0$, and $Lr = Lr_{min}$ when $T_i = T_{fin}$. Arc cotangent function is introduced in equation (4) which makes the learning rate scheduler more effective. Due to the characteristics of arc cotangent function, PACL has a larger proportion of small learning rates, as such a more stable system can be achieved.

Compared with conventional learning rate scheduler, such as exponential and piecewise decay, PACL has a periodic mechanism, which enables the model to skip the saddle point faster during the later stage to achieve better performances. In addition, PACL makes the learning rate decrease rapidly in the period, which enables us to set a large initial learning rate to improve the convergence speed of the model in the early stage.

Compared with SGDR and CLR, PACL reduces the proportion of large learning rate and decays the learning rate rapidly in each cycle. It will be more beneficial to optimize

the neural network. What's more, the setting of the minimum learning rate can make the learning rate far away from zero, which is more helpful to the early training of the network, because when the learning rate is close to zero, the noise will dominate the update of DNN weights [18].

B. ESTIMATE MAXIMUM AND MINIMUM BOUNDARY

We use "LR range test", which was first introduced by Smith [16] to estimate reasonable maximum and minimum learning rate boundaries. Fig.2 shows an example of running "LR range test" with the CIFAR-10 dataset. We set the initial learning rate to a very low value such as 10^{-5}, and set the final learning rate to a high value such as 1. Then, we run the model for one epoch while letting the learning rate increase from the lowest to the highest value we set. With the increase of the learning rate, it will eventually become too large, which will lead to the increase of test loss. We can see a typical curve from an LR range test from Fig.2, where the test loss has a distinct trough and peak. Generally, Lr_{max} is set when loss rises, and Lr_{min} is set when the gradient of loss is minimum.

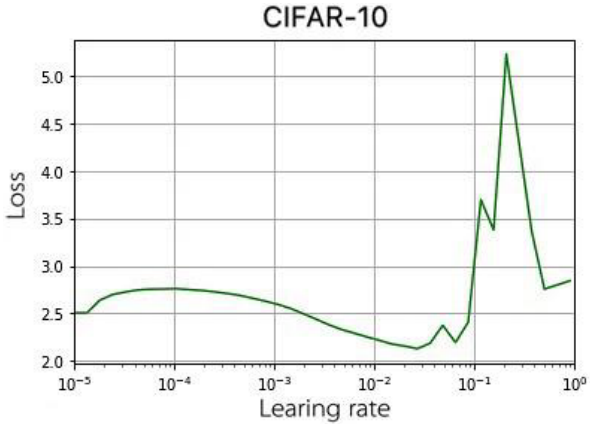


FIGURE 2. The change of loss with the increase of learning rate in one epoch.

C. ESTIMATE FREQUENCY OF LEARNING RATE TRANSFORMATION

In the practical application of PACL, div is introduced to represent the update frequency of the learning rate in each epoch. For example, the learning rate updated twice in each epoch when div = 2. The introduction of Div can make the learning rate update piecewise or linearly, which makes the change of learning rate more flexible in the cycle.

We compare the PACL algorithms with different T_{fin} and Div on the CIFAR-10 dataset. Fig.3 shows the learning rate is initialized to Lr_{max}, and decay to Lr_{min} by PACL with different parameters in each cycle. Fig.4. show the accuracy of PACL with different parameters on the CIFAR-10 dataset. The results for $T_{fin} = 5$ and $Div = 1$ show better performance, and therefore we use $T_{fin} = 5$ and $Div = 1$ in our later experiments.

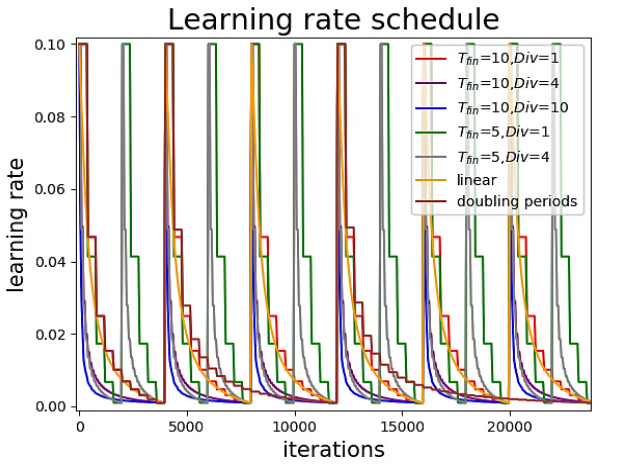


FIGURE 3. Seven different instantiations of this new learning rate schedule: PACL with $Div = 1$ (red line), $Div = 4$ (purple line), $Div = 10$ (blue line) for $T_{fin} = 10$; PACL with $Div = 1$ (green line), $Div = 4$ (grey line) for $T_{fin} = 5$; PACL with $T_{fin} = 10$ and update the learning rate every step (orange line); $Div = 1$ and Initial $T_{fin} = 10$ with doubling periods at every new cycle start (brown line).

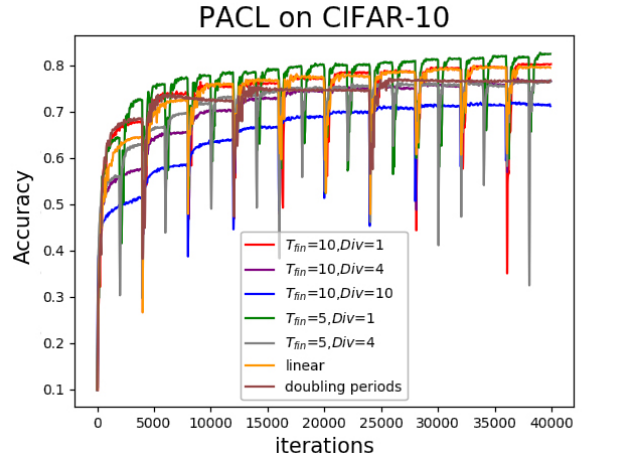


FIGURE 4. Test accuracy on CIFAR-10. PACL with $Div = 1$ (red line), $Div = 4$ (purple line), $Div = 10$ (blue line) for $T_{fin} = 10$; PACL with $Div = 1$ (green line), $Div = 4$ (grey line) for $T_{fin} = 5$; PACL with $T_{fin} = 10$ and update the learning rate every step (orange line); $Div = 1$ and Initial $T_{fin} = 10$ with doubling periods at every new cycle start (brown line).

IV. EXPERIMENTAL AND ANALYSIS

In this section, we demonstrate the effectiveness of PACL training with different networks. In the subsections below, our algorithm (PACL) is used for training on CIFAR-10, CIFAR-100 and Tiny ImageNet dataset, and compared PACL with six types of schedulers: exponential decay, piecewise decay, fixed learning rate, CLR, SGDR, and HTD.

A. EXPERIMENTAL PLATFORM

The experiments in part C of chapter IV were executed on a computer with Windows10 operating system, Intel (R) Core (TM) i5-3470MCPU, GeForce RTX^{TM} 2080,32GB RAM and by programming in Python. We used HUAWEI's ModelArts servers for all the rest of the experiments.

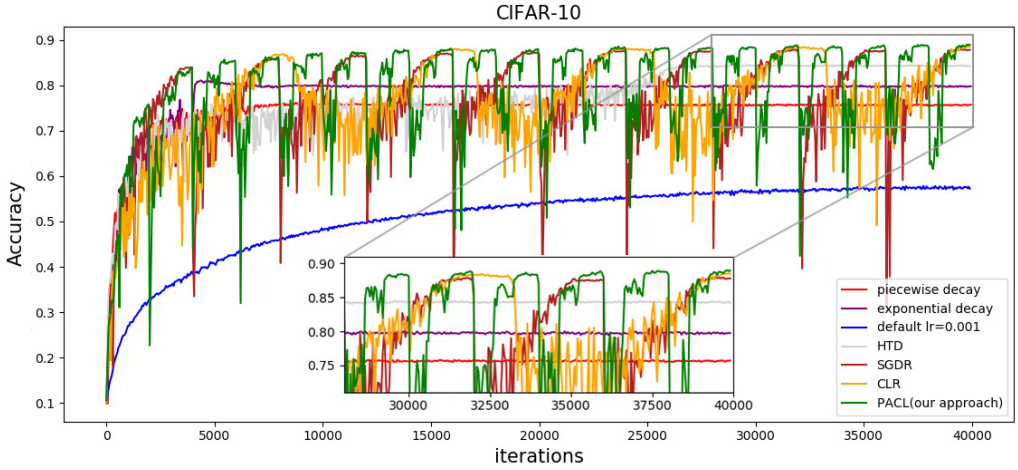


FIGURE 5. Test accuracy on CIFAR-10 with different learning rate schemes: piecewise decay (red line), exponential decay (purple line), default learning rate (blue line), SGDR (brown line), CLR (orange line) and our approach(green line).

Each server contains one NVIDIA Tesla P100 GPU, Intel E5-2690V4 CPU, and 64GB RAM. All the proposed models are run over highly efficient GPU using the PyTorch deep learning framework.

B. DATASET

The CIFAR-10 dataset [30] consists of 60000 color images. These images are 32×32, divided into 10 categories, and each category has 6000 images. There are 50000 training images and 10000 test images. The CIFAR-100 dataset [30] is just like the CIFAR-10 but has 100 classes instead of 10, and each class has 600 images. There are 500 training images and 100 testing images per class.

The Tiny ImageNet [31] is similar to the ImageNet [32], but it has only 200 categories. Each category has 500 images for training, 50 for testing, and 50 for verification. The images are 64×64 pixels.

C. EXPERIMENT ON CIFAR-10

We train the ResNet-32 [33] with seven types of schedulers mentioned earlier on the CIFAR-10 dataset. The networks are trained by SGD with momentum of 0.9 and a mini-batch size of 128. Using L2 regularization, regularization coefficient $= 0.001$, to avoid overfitting, small values of L2 can help prevent overfitting the training data.

Experiments in this part, we do not use any image preprocessing. For exponential and piecewise decay, we use an initial learning rate of 0.1, and the former is decayed by a factor of 10 times after every ten epochs while the latter is decayed by 0.9 times after every epoch. For CLR and SGDR, we set $Lr_{max} = 0.5$, $Lr_{min} = 0.001$, and for HTD, we set $Lr_{max} = 0.5$, $Lr_{min} = 0$. For our algorithm (PACL), we set initial parameters as $Lr_{max} = 0.5$, $Lr_{min} = 0.001$, $T_{fin} = 5$, $Div = 1$. All tests are trained for 100 epochs, 390 iterations in each epoch.

Fig.5. and Fig.6.provide a comparison among exponential decay, piecewise decay, fixed learning rate, HTD, CLR, SGDR, and PACL on the CIFAR-10 dataset. As can be seen from Fig.5. although PACL (green curve) has a temporary fall in performance in the training process compared with other algorithms, it can make DNN convergence faster and final accuracy higher. The PACL (green curve) not only reaches an accuracy of 85.87% after only 5,450 iterations but also the final accuracy of 88.96 is significantly higher than that of other algorithms without cycle mechanism. In addition, compared with CLR (orange curve) and SGDR (brown curve), PACL has less performance loss during training. The final accuracy is also slightly higher than that of CLR and SGDR by 0.23% and 0.35%. We define performance degradation as 90% below the highest accuracy in a stable period. The performance degradation area of PACL is only one-third on average in the stable period. This phenomenon can be found clearly in Fig.5. The same results are also reflected in the test of the Tiny-Image dataset.

D. EXPERIMENT ON DIFFERENT NETWORK

Shortcut (or short path) is a very effective structure in the development of the CNN model. Neural network models with shortcut structures, such as ResNet [33], WRN [34], and DenseNet [35], have excellent performance in computer vision tasks. In addition, SENet [36] and MobileNet [37] also have good performance in image recognition due to their unique structure. In this part, we provide comparisons between CLR, SGDR, HTD, and PACL based on the network mentioned earlier.

For PACL, the learning rate bounds set followed TABLE 1 We set Hyper-parameter $T_{fin} = 10$, $div = 1$ for ResNet, DenseNet, WRN, SENet and MobileNet. Specially, we set $T_{fin} = 5$, $div = 1$ for DenseNet on CIFAR-10.

For image preprocessing, we normalize the input data using the channel means and standard deviations. For

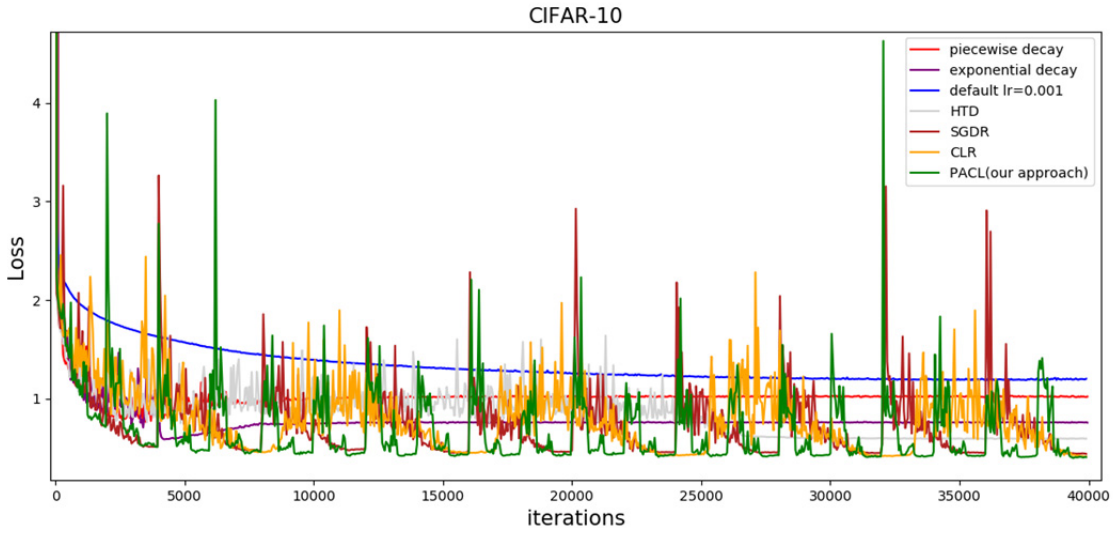
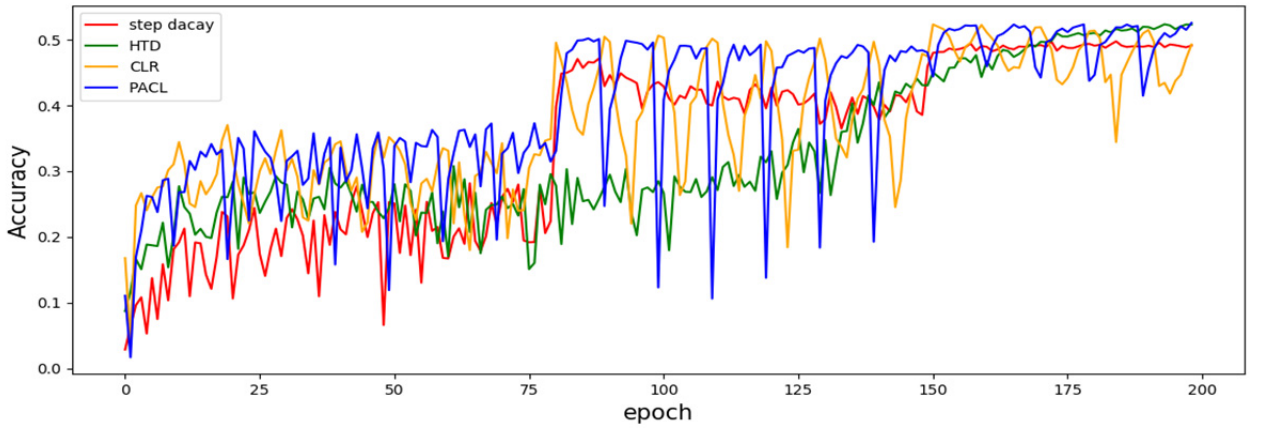
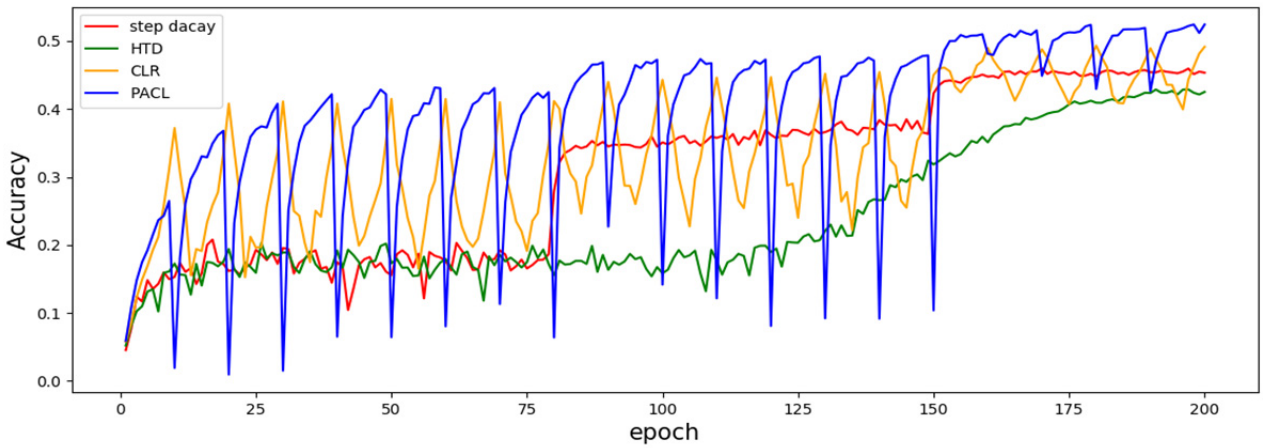


FIGURE 6. Test loss on CIFAR-10 with different learning rate schemes: piecewise decay (red line), exponential decay (purple line), default learning rate (blue line), SGDR (brown line), CLR (orange line) and our approach(green line).



(a). Training with ResNet-50



(b). Training with MobileNet V2

FIGURE 7. Test accuracy on CIFAR-10 with different learning rate schemes: piecewise decay (red line), exponential decay (purple line), default learning rate (blue line), SGDR (brown line),CLR (orange line) and our approach(green line).

data augmentation, we padded the picture by 4 pixels on each side, then, perform random cropping and horizontal flipping.

TABLE 2 compares the result of accuracy performance when training the network by PACL to other methods. In the table, the left two columns give the architecture used in the

TABLE 1. Learning rate settings for PACL.

Lr_{max}	Lr_{min}	Epoch
0.1	0.05	0-150
0.05	0.01	151-200
0.01	0.001	201-250
0.005	0.001	251-300
0.001	0.0005	301-400

TABLE 2. Comparison of PACL on ResNet, WRN, and DenseNet. The table shows the average accuracy of 3 runs on the CIFAR-10 and CIFAR-100. Some literature use error rate as an evaluation standard, we convert the error rate into accuracy to compare. The character * indicates results are directly obtained from the original paper.

Network	Depth-k	Method	CIFAR-10	CIFAR-100
ResNet[33]	110	piecewise decay*	93.57	-
		CLR*	93.6	72.5
		HTD*	94.32	73.22
		PACL	**94.73**	**75.84**
DenseNet-BC[35]	100-12	piecewise decay*	**95.90**	77.73
		CLR*	94.9	75.9
		HTD*	95.53	**77.83**
		PACL	95.49	77.80
WRN[34]	20-10	piecewise decay*	95.83	79.50
		SGDR*	95.76	79.67
		HTD*	95.78	80.27
		PACL	**96.24**	**81.11**
SEResNet[36]	29-16x64d	piecewise decay	95.29	81.28
		CLR	96.27	81.07
		SGDR	96.21	81.11
		HTD	96.23	81.27
		PACL	**96.29**	**81.30**
MobileNet V2[37]	x2_0	piecewise decay	93.17	72.94
		CLR	93.39	72.62
		SGDR	93.41	72.76
		HTD	93.45	72.83
		PACL	**93.73**	**73.20**

experiments. The third column gives the learning rate update method. The other two columns show the average accuracy from three runs.

For ResNet, the original test accuracy of 93.57% on CIFAR-10 can be improved to 94.73%, and accuracy of 75.84% on CIFAR-100. The accuracy of WRN trained with PACL can achieve 96.24% and 81.11% on CIFAR-10 and CIFAR-100 respectively. Performance improvement can also be reflected in SEResNet and MobileNet. PACL is outperforming the most current leading methods except for DenseNet-BC-100-12.The accuracy performance of PACL on DenseNet-BC-100-12 is similar to HTD, 95.49, and 77.80 respectively.

E. EXPERIMENT ON TINY IMAGENET

In this part, we provide comparisons between piecewise decay, CLR, HTD, and PACL on the Tiny ImageNet dataset.

We trained the ResNet-50 and MobileNet V2 on the Tiny ImageNet dataset using settings similar to the experiment on the CIFAR datasets: SGD with the momentum of 0.9; the L2 regularization coefficient = 0.001; the mini-batch size of 128. For piecewise decay, we used the initial learning rate 0.1, which decays by 0.1 at 80 and 150 epochs. For PACL and CLR, the learning rate followed TABLE 3 and set Hyper-parameter $T_{fin} = 10, div = 1$. We set $Lr_{max} = 0.1, Lr_{min} = 0$ for HTD.

TABLE 3. Learning rate settings for PACL.

Networks	Lr_{max}	Lr_{min}	Epoch
ResNet-110	0.1	0.05	0-80
	0.01	0.005	81-150
	0.001	0.005	151-200
MobileNetV2	0.1	0.01	0-80
	0.05	0.005	81-150
	0.01	0.001	151-200

TABLE 4. Classification accuracy of different learning rate schemes test on Tiny ImageNet dataset.

Network	Depth-k	Method	Top-1	Top-5
ResNet[33]	110	piecewise decay	51.44	74.36
		CLR	52.33	75.49
		HTD	52.36	73.08
		PACL	**52.50**	**75.54**
MobileNet V2[37]	x2_0	piecewise decay	44.20	71.62
		CLR	49.32	75.89
		HTD	42.85	70.21
		PACL	**52.44**	**77.00**

TABLE 4 compares the result of accuracy performance when training the network by PACL to other methods. For ResNet, the top-1 accuracy of PACL is slightly higher than that of other methods, but for MobileNet, the top-1 accuracy of PACL is higher than that of other methods.

Fig.7. compares the results of running with the piecewise decay, CLR, HTD, and PACL for the ResNet and MobileNetV2. As can be seen from Fig.7.(a) and Fig.7. (b) that convergence rate and final accuracy of PACL (blue curve) is better in comparison to any other algorithms.

Especially, compared with CLR (orange curve), PACL greatly reduces the performance degradation zone in each cycle. The performance degradation area of PACL in each cycle is only one-third of the cycle.

V. CONCLUSION

In this paper, we propose a new scheduling method, named piecewise arc cotangent decay learning rate (PACL), to improve the performance of DNNs. PACL combines the advantages of piecewise decay and CLR, adopts the mechanism of the cyclic learning rate, and the learning rate piecewise decay in each cycle. Compared with other learning rate schedulers with circular mechanisms, PACL significantly reduces the performance degradation zone caused by the

cycling mechanism. Besides, the setting of a minimum learning rate can make the learning rate far away from zero, which improves the effect of learning rate scheduler with circular mechanisms in the early stage of network training. Training DNNs with PACL can improve not only the accuracy of the network but also its convergence speed. Finally, we demonstrate the effectiveness of PACL, on CIFAR-10, CIFAR-100, and Tiny ImageNet, training with ResNet, DenseNet, WRN, SEResNet, and MobileNet. Future work should consider the application of PACL in some popular adaptive optimization algorithms such as Adam.

REFERENCES

[1] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu, "Sequential click prediction for sponsored search with recurrent neural networks," presented at the 28th AAAI Conf. Artif. Intell., 2017. [Online]. Available: https://arxiv.org/abs/1404.5772

[2] H. Lu and Q. Zhang, "Review on the application of deep convolution neural network in computer vision," *J. Data Acquisition Process.*, vol. 31, no. 1, pp. 1–17, 2016.

[3] X.-H. Yu, G.-A. Chen, and S.-X. Cheng, "Dynamic learning rate optimization of the backpropagation algorithm," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 669–677, May 1995.

[4] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[5] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.

[6] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*. [Online]. Available: https://arxiv.org/abs/1212.5701

[7] T. Tieleman and G. Hinton, "Lecture 6.5-RmsProp: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[8] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," presented at the 3rd Int. Conf., 2017. [Online]. Available: https://arxiv.org/abs/1412.6980v5

[9] J. Shi and D. Wang, "Research progress of random gradient descent," *Acta Automatica Sinica*, 2019, doi: 10.16383/j.aas.c190260.

[10] D. Babichev and F. Bach, "Constant step size stochastic gradient descent for probabilistic modeling," presented at the 34th Conf. Uncertainty Artif. Intell., 2018. [Online]. Available: https://arxiv.org/pdf/1804.05567.pdf

[11] H. Jeff, I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *Genetic Program. Evolvable Mach.*, pp. 1–3, 2017.

[12] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, 2nd ed. Berlin, Germany: Springer, 2012, pp. 437–478.

[13] Q. Fu and Y. Luo, "Improving learning algorithm performance for spiking neural networks," in *Proc. 17th Int. Conf. Commun. Technol.*, 2017, pp. 1916–1919.

[14] B. Y. Hsueh and W. Li, "Stochastic gradient descent with hyperbolic-tangent decay on classification," in *Proc. 19th IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2019, pp. 435–442.

[15] Y. Feng and Y. Li, "An overview of deep learning optimization methods and learning rate attenuation methods," *Hans J. Data Mining*, vol. 8, no. 4, pp. 186–200, 2018.

[16] L. N. Smith, "Cyclical learning rates for training neural networks," presented at the 17th IEEE Winter Conf. Appl. Comput. Vis., 2017. [Online]. Available: https://arxiv.org/abs/1506.01186

[17] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," presented at the 5th Int. Conf. Learn. Represent., 2017. [Online]. Available: https://arxiv.org/abs/1608.03983

[18] W. An, H. Wang, Y. Zhang, and Q. Dai, "Exponential decay sine wave learning rate for fast deep neural network training," in *Proc. IEEE Vis. Commun. Image Process.*, Dec. 2017, pp. 1–4.

[19] L. Bottou, "Large-scale machine learning with stochastic gradient descent," presented at the COMPSTAT, 2010. [Online]. Available: https://leon.bottou.org/publications/pdf/compstat-2010.pdf

[20] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999.

[21] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999, p. 187.

[22] A. Cauchy, "Méthode générale pour la résolution des systemes d'équations simultanées," *Comptes Rendus Sci. Paris*, vol. 25, pp. 536–538, Oct. 1847.

[23] S. Amari, "A theory of adaptive pattern classifiers," *IEEE Trans. Electron. Comput.*, vol. EC-16, no. 3, pp. 299–307, Jun. 1967.

[24] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*, D. Sad, Ed. Cambridge, U.K.: Cambridge Univ., 1998, pp. 15–21.

[25] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. ICML*, 2013, vol. 28, no. 3, pp. 1139–1147.

[26] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, "Classification of radar returns from the ionosphere using neural networks," in *Johns Hopkins APL Tech. Dig.*, vol. 10, 1989, pp. 262–266.

[27] L. Luo, "Adaptive gradient methods with dynamic bound of learning rate," presented at the 7th Int. Conf. Learn. Represent. (ICLR), May 2019. [Online]. Available: https://arxiv.org/pdf/1902.09843.pdf

[28] J. Jiao, X. Zhang, F. Li, and Y. Wang, "A novel learning rate function and its application on the SVD++ recommendation algorithm," *IEEE Access*, vol. 8, pp. 14112–14122, 2020.

[29] Y. N. Dauphin, H. de Vries, J. Chung, and Y. Bengio, "Equilibrated adaptive learning rates for non-convex optimization," in *Proc. 29th Annu. Conf. Neural Inf. Process. Syst.*, 2015, pp. 1504–1512.

[30] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Handbook Syst. Autoimmune Diseases*, vol. 1, no. 4, pp. 14–23, 2009.

[31] Y. Le and X. Yang. (2015). *Tiny ImageNet Visual Recognition Challenge*. Tiny Imagenet Data Sets. [Online]. Available: http://tiny-imagenet.herokuapp.com

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778. [Online]. Available: https://arxiv.org/abs/1512.03385

[34] S. Zagoruyko and N. Komodakis, "Wide residual networks," presented at the Conf. Comput. Vis. Pattern Recognit., 2016. [Online]. Available: https://arxiv.org/abs/1605.07146

[35] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," presented at the IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017. [Online]. Available: https://arxiv.org/abs/1608.06993

[36] H. Jie, L. Shen, and G. Sun, "Squeeze-and-excitation networks," presented at the IEEE Conf. Comput. Vis. Pattern Recognit., 2016. [Online]. Available: https://arxiv.org/abs/1709.01507

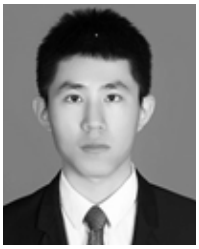
[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4510–4520.

[38] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng "On optimization methods for deep learning," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.*, 2011, pp. 265–272.

[39] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent converges to minimizers," Mar. 2016, *arXiv:1602.04915*. [Online]. Available: https://arxiv.org/abs/1602.04915

HAIXU YANG received the B.S. degree from the Taiyuan University of Science and Technology, Shanxi, China, 2017. He is currently pursuing the M.S. degree with the College of Information Science and Engineering, Northeastern University, China. His research interests include image processing, pattern recognition, and deep learning.

JIHONG LIU received the Ph.D. degree in pattern recognition and intelligent system from Northeastern University, China, in 2003. She is currently an Associate Professor with the School of Information Science and Engineering, Northeastern University. Her research interests include computational cardiology, intelligent information processing, and biomedical signal acquisition.



HONGWEI SUN received the B.S. degree in electrical information engineering from Shijiazhuang Tiedao University, China, in 2014. He is currently pursuing the M.S. degree with the College of Information Science and Engineering, Northeastern University, China. His research interests include deep learning, natural language processing, and signal processing of sEMG.

HENGGUI ZHANG received the Ph.D. degree in mathematical cardiology from the University of Leeds, in 1994. He worked as a Postdoctoral Research Fellow at the School of Medicine, Johns Hopkins University, from 1994 to 1995, and the University of Leeds, from 1996 to 2000, where he was also a Senior Research Fellow, from 2000 to 2001. In October 2001, he moved to UMIST to take up the Lectureship. He also worked as a Lecturer at UMIST, from 2001 to 2004, a Senior Lecturer (from 2004 to 2006) and a Reader (from 2006 to 2009) at The University of Manchester. He currently holds the Chair of Biological Physics Group, School of Physics and Astronomy, The University of Manchester. He is a Professor of biological physics. He has published more than 400 scientific articles, among them over 200 articles were published in prestigious peer-reviewed journals in his field. Related works have attracted wide public interests, and been covered by many prestigious media, such as BBC. He has been elected as a Fellow of world renowned societies as recognition of distinctions.