Learning rate adjustment and optimization of RepVGG network based on warmup strategy

Ying Lin

Beijing-Dublin International College, Beijing University of Technology, Beijing, 100124, China

linying@emails.bjut.edu.cn

Abstract. Artificial neural networks have developed rapidly in recent years and play an important role in the academic field. In this paper, the RepVGG artificial neural network model is adjusted by the learning rate algorithm, so as to realize the optimization of the model including but not limited to accuracy. The main optimization strategy is to add the warmup strategy based on the learning rate algorithm of the original model so that the model can obtain good prior information on the data early in the training process, so as to converge quickly in the later training. Through a series of tests and simulations, the RepVGG-A0 model improves the Top1 accuracy by about 2.6% to 68.56% and the Top5 accuracy by about 0.38% to 94.32% on imagesetter dataset within 25 training epochs. The precision and recall are improved to 68.43% and 68.63%, respectively.

Keywords: RepVGG, Learning Rate, Warmup Strategy, Optimization.

1. Introduction

Deep learning and artificial neural networks are research fields that have developed rapidly in recent years. As the first artificial neural network model to achieve the Top1 accuracy of more than 80% on the imageNet dataset, the RepVGG network is far faster and more accurate than other networks [1]. Artificial neural networks have a huge number of hyperparameters such as batch size, number of network layers, and learning rate. For hyperparameter optimization of artificial neural networks, the adjustment of learning rate can greatly improve the quality of the model. This paper aims to optimize the learning rate algorithm of the RepVGG network, so that the performance of the model, including but not limited to accuracy, can be improved. The second part of the paper is an introduction to RepVGG network. It mainly describes the principle and structure of RepVGG network, its characteristics, and the learning rate algorithm used in it. The third part elaborates on the warmup strategy, including the constant warmup strategy and the gradual warmup strategy, and their training error analysis. The main content of the fourth part is the configuration and simulation of the experiment, mainly including data set selection, experimental parameters, and performance improvement analysis. Through a series of experiments and adjustments, the RepVGG model has a certain degree of improvement in precision, average recall, and average precision.

2. Architecture of RepVGG and Learning rate algorithm

2.1. Architecture of RepVGG

RepVGG-A0 is used as the backbone network in this paper. The network utilizes a multiple branch model similar to ResNet for training and becomes a unique path model of the VGG type for reasoning, see Figure 1 [1].

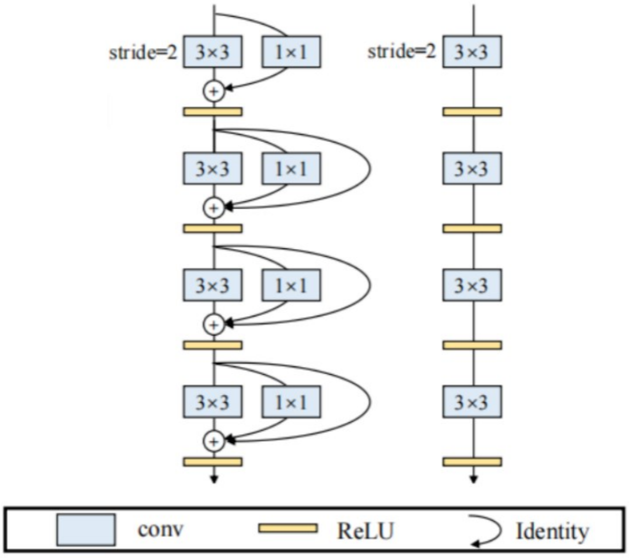

Figure 1. Architecture of RepVGG: (a) RepVGG formation phase (b) RepVGG inferenced stage [1]

Figure 1(a) shows the network structure used by RepVGG for formation and Figure 1(b) for logic. The multi-branch fusion technology of this network model combines the convolutional layer with the Batch Normalization layer (BN), and the multiple convolutional cores with different sizes are equally transformed into the multiple 3×3 convolutional cores and eventually fused into the single 3×3 convolutional cores. The network model of deploy mode is designed. This model consists only of 3×3 convolution and Rectified Linear Units (ReLU) activation function, which has a better receptive field than other large convolutions, thus avoiding the problem of gradient disappearance in training multi-branch models [2].

The reasons for using the single path model in reasoning are as follows:

Fast: For the multi-branch model, the hardware needs to calculate each branch separately. After the fast branch is calculated, it can only be further integrated after the other branches are calculated. As a result, the hardware computing power cannot be fully utilized.

Memory saving: The memory efficiency of a multibranched topology is weak because the results of each branch must be saved for addition or attachment, which significantly increases peak memory usage.

Flexibility: A typical architecture gives us the flexibility to customize each convolution layer to meet our demands and trim for improved performance-efficiency trade-off.

After the multi-branch is transformed into a single-way model, many operators are fused (such as Conv2d and BN fusion), which makes the computation smaller, and the number of kernel startups is also reduced after the operator is reduced.

As Figure 2 shows, structure reparameterization is mainly divided into two steps [1].

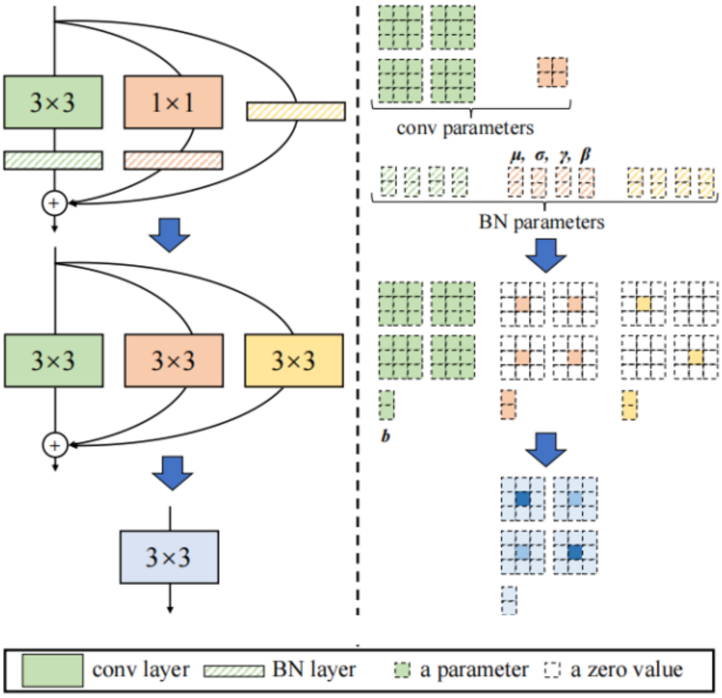

Figure 2. Structural reparameterization of RepVGG (a) Specific composition of layers (b) Parameter components of the layers

1) The first step is mainly to fuse the Conv2d operator with the BN operator and convert the branch with only BN into a Conv2d operator. The second step is to fuse 3x3 convolution layers on each branch into a convolution layer. According to Figure 3, RepVGG is VGG-style with a vanilla topology and heavy use of 3×3 convs. However, as opposed to VGG, it does not employ max pooling accounting for that it is desired that the body only adopts a single type of function operator [2].

The instance named RepVGG-A0 has five stages with 1, 2, 4, 14, and 1 layers and to further reduce parameters and computations, it is chosen to interleave 3×3 transform layers with dense transform layers in exchange for accuracy and efficiency, which is done by setting the number of groups g at layer 3, layer 5, layer 7,... at layer 21 to 1,2, or 4.

2.2. *Learning rate algorithm*

The learning rate algorithm of RepVGG-A0 is cosine annealing, which is one of the simplest warm restart approaches. Simulates a restart of Stochastic Gradient Descent (SGD), where *i* is the index running. Restart is simulated through enhancing the learning percentage η_t, and the old value of *xt* is employed as the initial solution. *xt* here is to resolve the slope loss feature solution, that is, the weight within the neural net. Since restarting is about ignoring local optimization through adding the LR, *xt* needs to be the old value. This increase in volume controls the extent to which previously available information can be used [3]. On *i*-th run, cosine annealing is used to attenuate the learning rate as follows.

$$\eta_{\mathrm{t}} = \eta^{\mathrm{i}}{}_{\min} + \frac{1}{2}(\eta^{\mathrm{i}}{}_{\max} - \eta^{\mathrm{i}}{}_{\mathrm{m}im})(1+\cos(\frac{T_{cur}}{T_i})\pi) \tag{1}$$

η_t is the learning rate. $\eta^i{}_{min}$ and $\eta^i{}_{max}$ are the ranges of the learning rate. T_{cur} means the number of epochs have been performed since the last restart and T_i means the total running epochs.

3. Warmup strategy

In automatic learning and statistics, the learning rate (LR) is an adjustment parameter within an optimization algorithm that determines the size of the step in each iteration and converts the loss function to a minimal value.

LR is one of the most cost-effective parameter adjustments that can affect the performance of the hyperparameter of the deep learning model that needs to be optimized. Most cases of loss value non-convergence are due to improper selection of learning rate.

Selecting an appropriate learning rate is very helpful to find the minimum value of global loss and improve the speed of model training [4]. Obviously, a low learning rate slows down convergence, a high learning rate at the beginning of the convergence speed is rapid, however, it will not be able to converge to the minimum value, the very high level of learning may even directly exceed the minimum value without any convergence effect [5].

The changing trend of the learning rate used by RepVGG-A0 is gradually decreasing. As learning goes on, the model gets closer to the extreme point. If the level of learning is too high, the model will cross the extreme point or diverge. However, there are mostly saddle points or poor minimum points in the parameter space of the model [6]. The former will seriously affect the learning efficiency of the model, and the latter will make the final performance of the model very poor. In addition, the model should converge to the extreme points in a broad basin of attraction in the parameter space, which has a strong generalization ability [7].

Based on the above problems, researchers found that the occasional increase in the learning rate during training led to a poor performance of the model in the short term. [3]. However, the final training outcomes performed better on the test set than the traditional gradual decay strategy. Because this will make the model escape from the saddle point faster, which will accelerate the model convergence [4]. Moreover, if the model converges to the extreme point of the narrow basin of attraction, then suddenly increasing the learning rate can also make the model escape from the extreme point of the narrow basin of attraction and converge to the extreme point of the wider basin of attraction.

3.1. *Constant warmup strategy* and *Gradual warmup strategy*

Zeng et al. use a 110-layer ResNet formed on cifar10 with a learning rate of 0.01 until the formation error is less than 80% (about 400 steps), then followed by a learning rate of 0.1 [8].

Shulman proposes another warm-up that gradually changes the learning rate from low to high. This banister prevents sudden increases from low to high learning rates, allowing for healthy convergence at the beginning of training [9].

3.2. *Training error*

Figure 3 is the simulation of training error at ImageNet using ResNet-50.

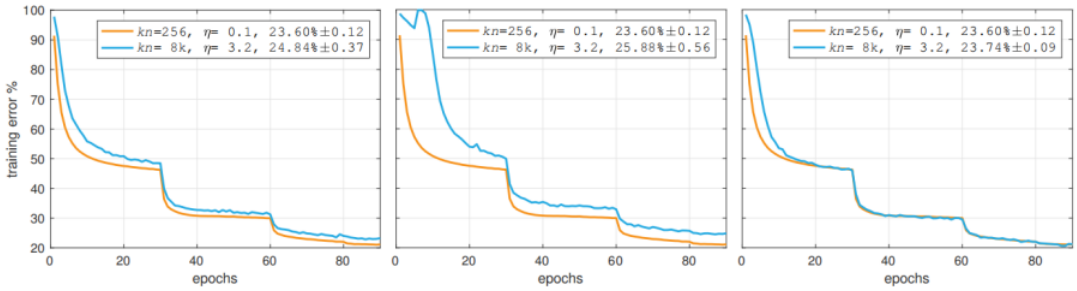

Figure 3. Training error for minibatch size 8192 compared to minibatch size 256 with various warmup strategies (a) no warmup (b) constant warmup (c) gradual warmup [9]

Without warming up (Figure 3(a)), the formation curve with huge batch size kn = 8k is below the level of training at small kn = 256 that spans the whole epochs. A constant warm-up strategy (Figure

3(b)) messes up the outcomes. In the case of gradual warming up, the formation error of the big mini-batch corresponds to the basic formation curve obtained by mini-batch training, see Figure 3(c) [9]. Although the large mini-batch curve starts higher in the warm-up phase due to the lower LR, it catches up soon after [9].

Therefore, using the warmup learning percentage, that is, training with the initial low LR, and increasing the learning rate a little at each step until reaching the initial relatively large learning rate, that is, after the warm-up phase is completed, the cosine annealing learning algorithm is employed, which will allow for faster and better convergence of the model.

4. Experiments and Analysis

4.1. Dataset

Imagenette was proposed by the University of San Francisco in 2020. It is a subset of 10 easily classified classes from Imagenet (including tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, and parachute) [10]. In the experiment, 1500 images were selected from the test set, with 150 images of each category, and 3200 images were selected from the training set, with 320 images of each type.

4.2. Configuration

All parameters of RepVGG-A0 remain unchanged. The training epoch is set to 25. The experiments compare the top5 and top1 accuracy, average precision, and an average recall of no warm-up and constant warm-up. The number of warm-up epochs was 5 and the cosine annealing of the original model was still used after the warm-up phase. That is, learning first increases linearly within a small value to a preset learning percentage through warmup strategy, then decays by the cosine annealing algorithm [11].

4.3. Analysis of experimental results

Figure 4 represents the trend towards changing learning rates in the initial model, and Figure 5 shows the learning rate change with the warmup strategy added.

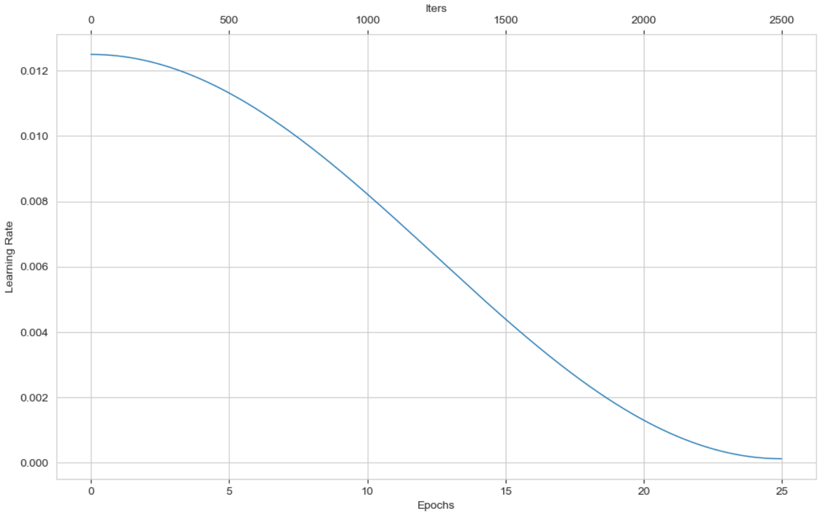

Figure 4. RepVGG-A0 learning rate curve

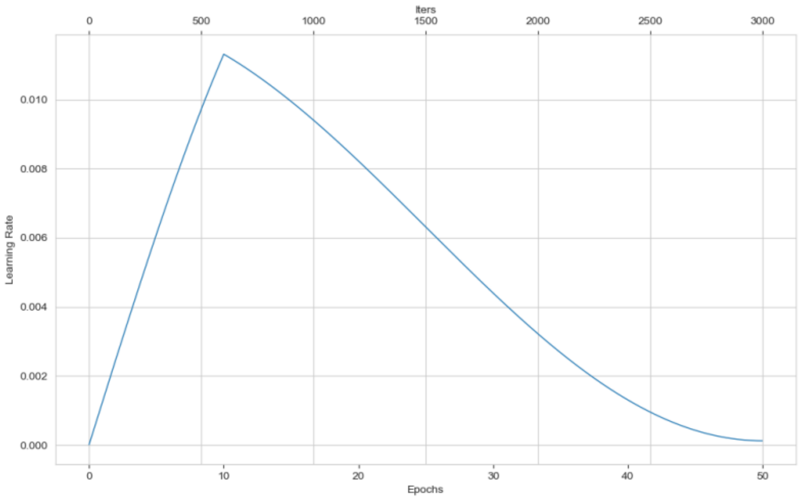

Figure 5. Learning rate after adding the warmup strategy to RepVGG-A0

Figure 6 illustrates the top 1 accuracy of the initial model and the optimized one after 25 epochs of training.

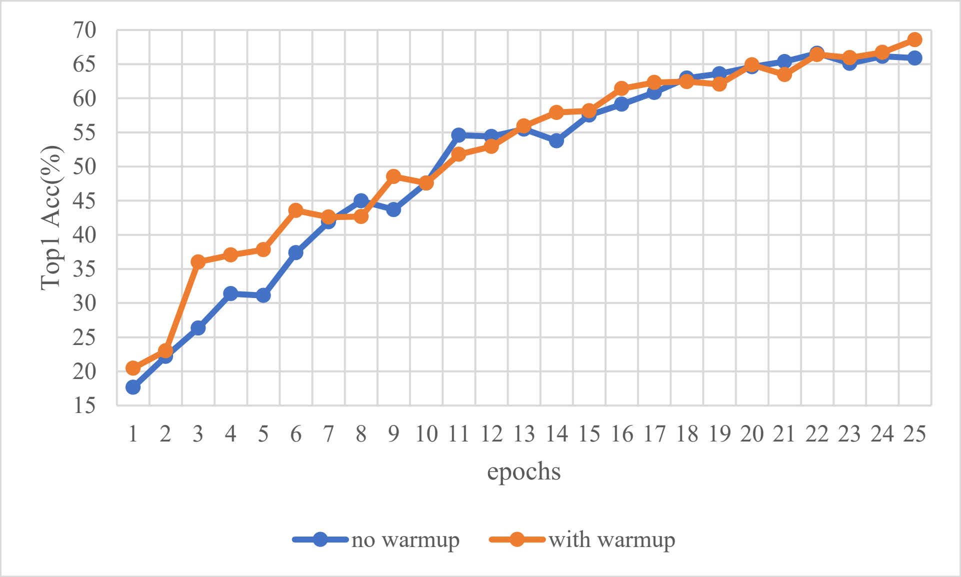

Figure 6. Top1 accuracy of RepVGG-A0 with and without the warmup strategy

Although the accuracy after optimization is low at the beginning, with the increase in the number of cycles, the accuracy gradually increases and even exceeds the original model. At the 25th epoch, it is 68.56%, and the original model is 65.88%, which increases by about 2.6% compared with the original one.

Figure 7 shows the top 5 accuracy of the model before and after optimization. At the 25th cycle, the accuracy is 93.94% before optimization, and 94.32% after optimization, with an increase of about 0.38%.

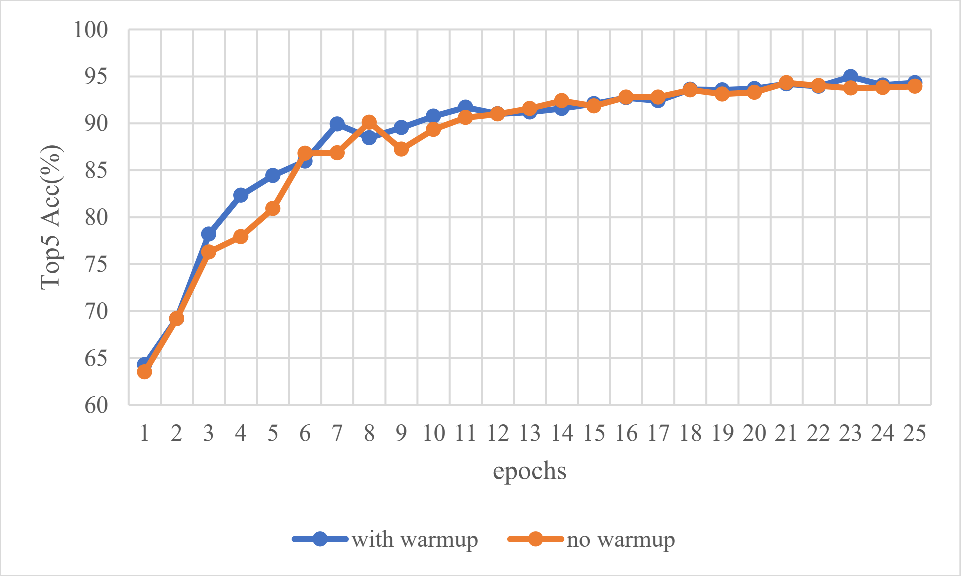

Figure 7. Top5 accuracy of RepVGG-A0 with and without the warmup strategy

Figure 8 depicts the mean precision before and after optimization and Figure 9 depicts the mean recall. Both ratios are relatively large with the learning rate of the warmup strategy as the epoch increases, although the ratio of the early warmup strategy is relatively jittered. In the 25th epoch, the average precision before and after optimization is 65.84% and 68.43%, and the average recall is 65.94% and 68.63%, respectively.

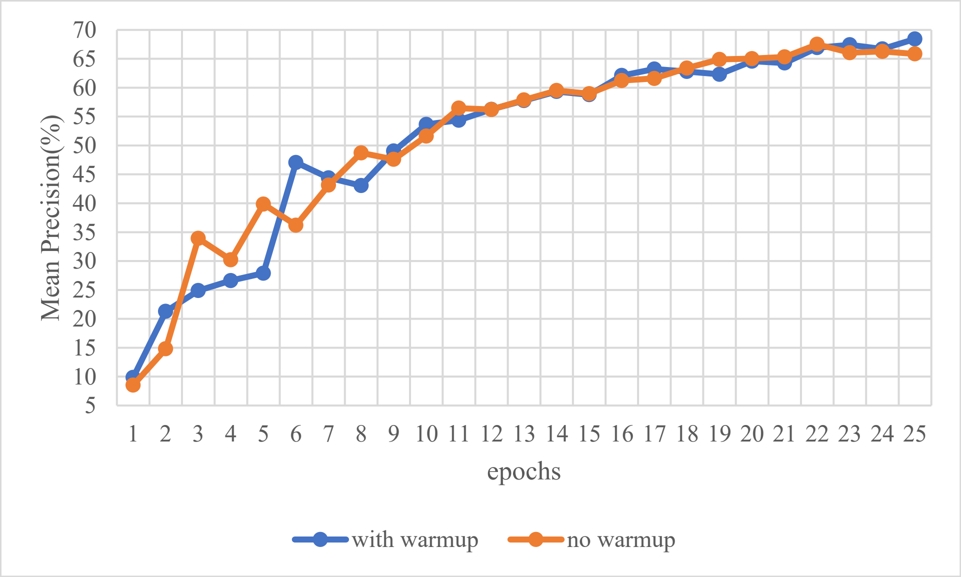

Figure 8. Mean precision of RepVGG-A0 with and without the warmup strategy

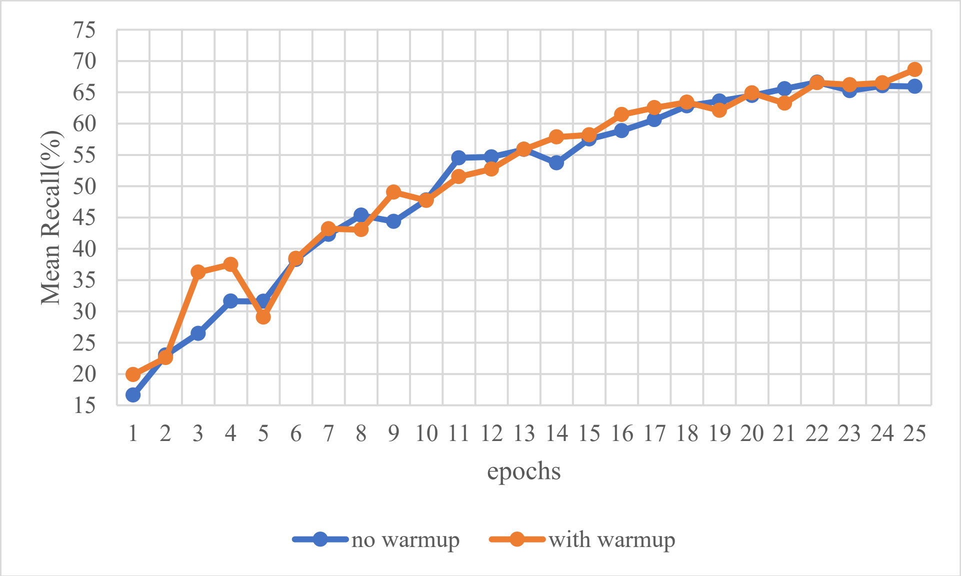

Figure 9. Mean recall of RepVGG-A0 with and without the warmup strategy

Since the weights of the model are randomly initialised at the start of the training, choosing a large learning rate at this point may cause the model to oscillate. That is because at the beginning of the epoch, the data of each batch is unfamiliar to the model, and the model will quickly adjust its parameters based on the input data, thus, if we use a large learning rate, there is a high risk of overfitting the model. After adding the warmup strategy, the model can slowly become steady under the small learning rate of warming up. At this moment, the model has some prior knowledge about the data. After the pattern is steady, the pre-set learning percentage is chosen for training, which sets the pattern convergence speed become more rapid and the model's impact has improved.

5. Conclusion

This paper introduces the structure of the RepVGG model, the warmup learning percentage strategy, and the learning percentage algorithm optimization of the RepvGG-A0 model based on the warmup strategy. This paper mainly uses the constant warmup strategy to adjust the learning rate of the RepVGG-A0 model to make its performance of the model be improved. Among performance, within 25 epochs, the Top1 accuracy of RepVGG-A0 on imagesetter dataset is improved to 68.56%, an increase of about 2.6%, and the Top5 accuracy is improved to 94.32%, an increase of about 0.38%. Moreover, the average precision and recall increased to 68.43% and 68.63%, respectively. By adding the warmup strategy, the model can better learn some prior information of the data in the early stage of training, to achieve a good convergence effect in the further learning of the data in the later stage. For future work, the RepVGG network should also adjust in the batch size, which may contribute to the stability of model convergence and the improvement of generalization ability.

References

[1] Great Again,” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2021.

[2] W. Li, X. Luo, Z. Meng, and J. Chen, “Facial expression recognition combined with improved RepVGG-A0 network and relabeling,” Oct. 2022.

[3] J. Sirignano and K. Spiliopoulos, “Stochastic Gradient Descent in Continuous Time,” SSRN Electronic Journal, 2017.

[4] G. Ioannou, T. Tagaris, and A. Stafylopatis, “AdaLip: An Adaptive Learning Rate Method per Layer for Stochastic Optimization,” Neural Processing Letters, Jan. 2023.

[5] K. Zeng, J. Liu, Z. Jiang, and D. Xu, “A Scaling Transition Method from SGDM to SGD with 2ExpLR Strategy,” Applied Sciences, vol. 12, no. 23, p. 12023, Nov. 2022.

[6] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. Wilson, “Averaging Weights Leads to Wider Optima and Better Generalization.”

[7] P. Goyal et al., “Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour.” 2020.

[8] K. Zeng, J. Liu, Z. Jiang, and D. Xu, “A Scaling Transition Method from SGDM to SGD with 2ExpLR Strategy,” Applied Sciences, vol. 12, no. 23, p. 12023, Nov. 2022.K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 10, 2015.

[9] D. Shulman, “Optimization Methods in Deep Learning: A Comprehensive Overview,” Feb. 2023.

[10] J. Howard and S. Gugger, “Fastai: A Layered API for Deep Learning,” Information, vol. 11, no. 2, p. 108, Feb. 2020, doi: https://doi.org/10.3390/info11020108.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 10, 2015. C. Zhang, M. Yao, W. Chen, S. Zhang, D. Chen, and Y. Wu, “Gradient Descent Optimization in Deep Learning Model Training Based on Multistage and Method Combination Strategy,” Security and Communication Networks, vol. 2021, pp. 1–15, Jul. 2021.

[12] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, “RepVGG: Making VGG-style ConvNets