Xie Yuxi
A0228328B

Report
Colossalai and LR Range Test

xieyuxi@u.nus.edu

Github Link: https://github.com/YuxiXie/LR_Range_Test

This is a report about *Colossalai and LR Range Test* [3] for CS5260 Assignment 6. In this assignment, learning rate range test is applied to help propose suitable learning rates for corresponding optimizers and learning-rate schedulers in training the model LeNet5 on MNIST dataset.

Specifically, the SGD optimizer is chosen for both LR range test and practical model training. And the scenarios of Multistep, OneCycle, and no scheduling are considered respectively to probe into how LR range test work under different training settings.

For the rest of the report, I present the result of LR range test in Section 1 and discuss how it can help train LeNet5 [2] on MNIST [1] dataset under different scheduling settings in Section 2.

1 LR Range Test

To begin with, LR Range Test is applied on the SGD optimizer during training, where the LR curve is exponentially increasing as displayed in Figure 2(a).

To catch the point lowest loss via the change of learning rates, the low and high bounds for learning rates are set as 10^{-6} and 1.18, respectively. And the number of epochs for training is set as 10.

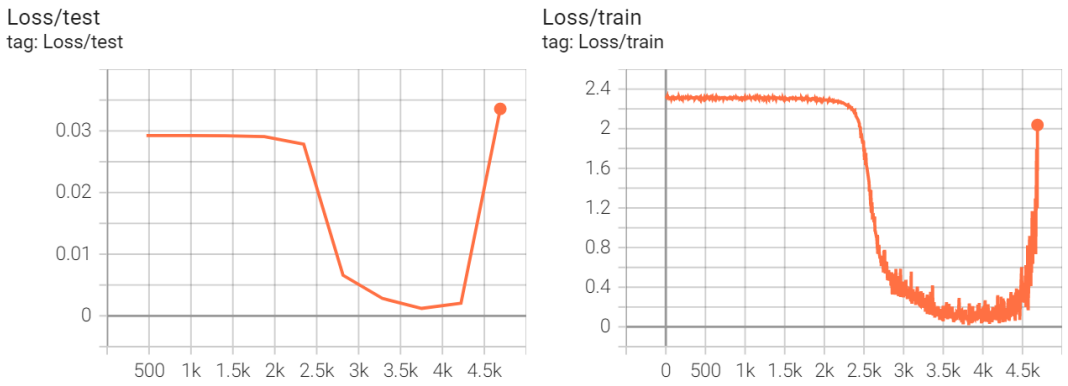

Figure 1: Loss curves on training (right) and testing (left) datasets.

Figure 1 illustrates the changes of loss on training (right) and testing (left) datasets respectively. The curve shape is consistent with the general cases in LR range test, *i.e.*, the loss first keep (nearly) unchanged, and then go down, and finally explode.

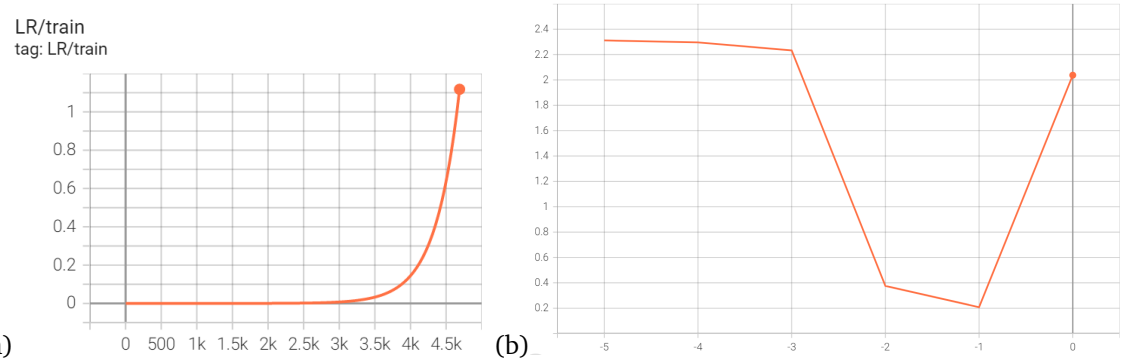

Figure 2: (a) LR curve during training. (b) Loss curve with the change of learning rates (in $\log_{10}$).

To locate the turning points of the loss curve with the corresponding learning rates, I also draw the loss curve on the $\log_{10}$ of the learning rate during training. As shown in Figure 2(b), the loss decrease to the lowest point when the learning rate is 10^{-1}. And at the point where learning rate is 10^{-3}, the loss starts to decrease significantly.

2 Experiment Results

Based on the learning rates at the turning points of the loss curves obtained in LR range test, two kinds of LR schedulers are chosen to probe into the effects of learning rates on training, with the MultiStep in Section 2.2 and OneCycle in Section 2.3.

2.1 No Scheduling

Before using LR range test results to tune schedulers' settings, I first compare the effects of different scales of learning rates on the loss curve. Specifically, I set the learning rate as 10^{-1} (red), 10^{-2} (blue), 10^{-3} (pink), respectively.

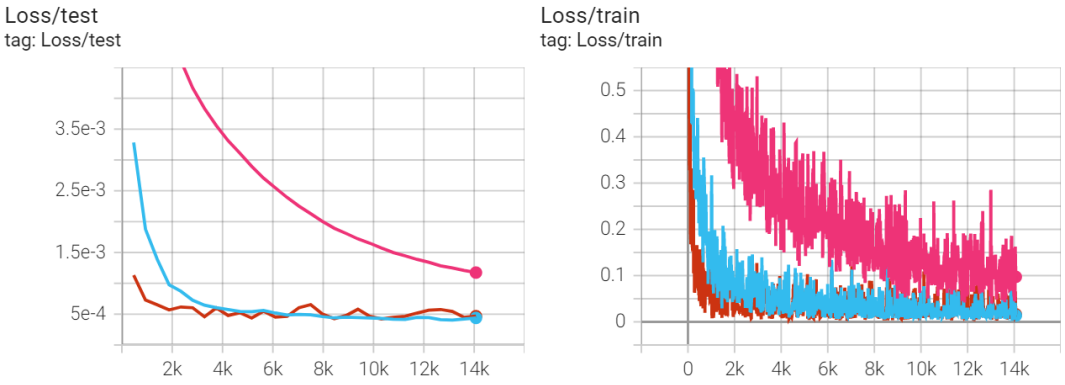

Figure 3: Loss curves with learning rate as 10^{-1} (red), 10^{-2} (blue), 10^{-3} (pink), respectively.

As shown in Figure 3, even learning rates of 10^{-1}, 10^{-2}, and 10^{-3} can all lead to decrease in loss, a bigger learning rate can obviously produce a faster convergence of the decreasing. Within 30 epochs of training, the learning rates of 10^{-1} and 10^{-2} can converge to similar performance on the testing data.

2.2 MultiStep LR Scheduler

Considering the performance of different learning rates illustrated in Section 2.1, I set the starting learning rate as 10^{-1} to speed up the convergence at the beginning. To determine the turning point where the learning rate will be decreased to 10^{-2} from the starting value, turning points of epoch 5 (green), 15 (grey), and 25 (orange) are compared with each other.

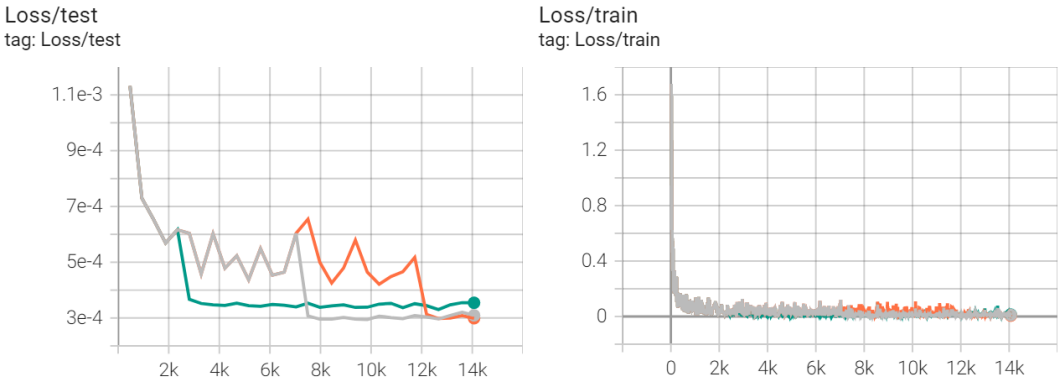

Figure 4: Loss curves with the turning point as epoch 5 (green), 15 (grey), and 25 (orange), respectively, in the MultiStep LR scheduler.

It is illustrated in Figure 4 that epoch 15 (grey) is a better turning point of the learning rate update, as it 1) achieves almost the lowest loss on the testing data among all three turning points; and 2) doesn't show a significant sign of over-fitting as there is just a slight increase in the loss change.

Similar comparison for the turning point of decreasing learning rate to 10^{-3} can also be conducted in the MultiStep LR scheduler here. However, as this value is of quite a small scale, the decrease in loss is

also too slight in the curve to obtain (which is quite similar with the curve in Figure 4). To summarize, for the MultiStep LR scheduler, I set the milestones as $[15, 25]$ (with gamma as 0.1) based on the above comparison results.

2.3 OneCycle LR Scheduler

For the OneCycle LR scheduler, it will be more straightforward to tune the setting as we only need to focus on tuning the maximum learning rate in the cycle.

In this case, I set the maximum learning rate as 10^{-1} according to the results obtained in Section 2.1. I also tune the `final_div_factor` to be 4 so that the minimum learning rate can be above 10^{-3} during training. For other parameters, I follow the default setting where `pct_start` is 0.3 and `div_factor` is 25.

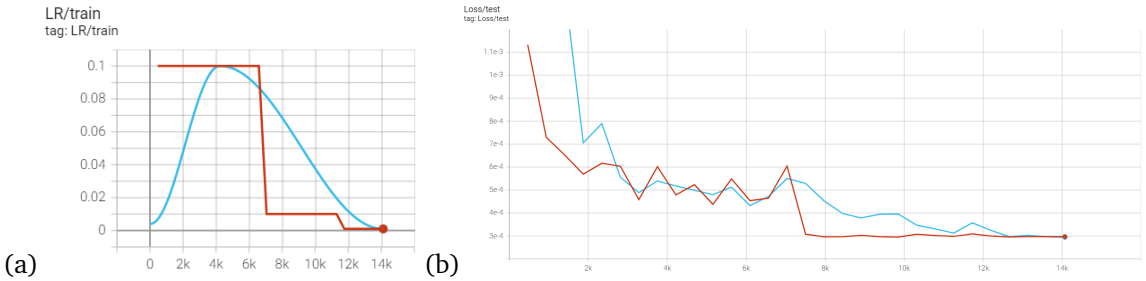

Figure 5: (a) LR curves during training with the OneCycle LR scheduler (blue) and the MultiStep LR scheduler (red). (b) Loss curves on testing data, where OneCycle's is in blue and MultiStep's is in red.

Figure 5(a) displays the change of learning rate during training. To compare between the two LR schedulers, I put the loss curves together in Figure 5(b). As shown in the figures, the OneCycle produces a more smooth loss curve due to smoother change in the learning rate.

2.4 Conclusion

To summarize, it is illustrated in this report that the LR range test helps to find suitable scales of learning rate to tune the settings in LR schedulers including MutiStep and OneCycle. And the warmup and smooth change of learning rate in OneCycle helps it to give a smoother loss curve than MutiStep.

In evaluation, the accuracy on testing dataset from the two training schedulers are as follows:

- MultiStep: 99.270%
- OneCycle: 99.180%

, which is consistent with the comparison result on the loss curves that the final performance is similar with each other.

References

[1] LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/* (1998).

[2] LeCun, Y., et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet 20*, 5 (2015), 14.

[3] Smith, L. N., and Topin, N. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications* (2019), vol. 11006, International Society for Optics and Photonics, p. 1100612.