ACCELERATING SGD WITH MOMENTUM FOR OVERPARAMETERIZED LEARNING: REPRODUCABILITY CHALLENGE

Dian Yu, Yixuan Liu, Zhuo Wang
School of Electronics and Computer Science
University of Southampton
Southampton,UK
{dy1y19, yl5g19, zw5n19}@soton.ac.uk

ABSTRACT

This paper aims to reproduce an optimize algorithm called MaSS which accelerates SGD with momentum for over parameterized learning. This algorithm is designed for address the non-accelerating problem of the Nesterov SGD by introducing a compensation term to it. The reproducibility of the algorithm is confirmed and both edge and shortage of this optimize algorithm is concluded.

1 INTRODUCTION OF THE PAPER

The selected paper is about purposing a new optimizer algorithm for overparameterized learning. The authors consider that the new optimizer is better than SGD and any other related SGD algorithm such as SGD with Nesterov and Adam. Also, the authors prove that SGD+Nesterov with any constant hyper-parameter setting doesn't improve the convergence over optimal SGD[1]. Here is the pseudocode for MaSS.

Algorithm 1 MaSS–Momentum-added Stochastic Solver

Input: Step-size η_1, secondary step-size η_2, acceleration parameter $\gamma \in (0, 1)$
1: Initialize: $\mathbf{u}_0 = \mathbf{w}_0$
2: **while** n **do**ot converged do
3: $\quad \mathbf{w}_{t+1} \leftarrow \mathbf{u}_t - \eta_1 \tilde{\nabla} f(\mathbf{u}_t)$
4: $\quad \mathbf{u}_{t+1} \leftarrow (1+\gamma)\mathbf{w}_{t+1} - \gamma \mathbf{w}_t - \eta_2 \tilde{\nabla} f(\mathbf{u}_t)$
5: **end while**
Output: :weight $\mathbf{w}_t$

2 TARGET QUESTIONS

The SGD is a widely used optimizer that used for training modern neural networks and other machine learning models. Nesterov SGD may diverge for step sizes that ensure convergence of ordinary SGD. The difference from the classical optimizer with the same step size ensures accelerated convergence of the Nesterov's method over optimal gradient descent. This may result the non-accelerating issue, and the MaSS optimizer is designed to solve this problem. The MaSS convergence for same step sizes as SGD. The author points out that MaSS obtains an accelerated convergence rates over SGD for any mini-batch size in the linear setting and for full batch, the MaSS matches the accelerated rate of Nesterov's method. Although it is a new optimizer algorithm, it is necessary to evaluate the convergence speed and accuracy on the validation dataset The target of this reproducibility is to prove that the new optimizer is indeed better than SGD and SGD + nesterov.

3 EXPERIMENTAL METHODOLOGY

We evaluate whether the paper can be successfully reproduced by redeploying the methods used in the paper, which includes using general convolutional neural network(CNN) and resnet-32 on CIFAR-10 dataset and MNIST dataset, besides, we also use a fully connected network in MNIST dataset. According to the paper, we use SGD, SGD + Nesterov and Adam optimizer to compare whether the mass will always be better than them under the same conditions. When the result of one implementation is not the same as expected, we will execute the code again or make appropriate modifications according to the paper.

4 IMPLEMENTATION

We used the GPU computer from the university and Colaboratory from Google to reproduce the paper. In the case of RTX2070, it takes more than an hour to classify CIFAR-10 using ResNet-32. It takes half an hour to use a convolutional neural network with three convolution layers under the same task. Due to the consideration of time and resources, we did not use the learn rate reduction mentioned in the paper. In the paper, the source code of the paper reduces the learning rate three times when training ResNet-32, and a total of 300 epochs are trained. We deployed ResNet-32 for 150 epochs and CNN for 100 epochs. In our opinion, we consider that it is enough to evaluate the performance of MaSS via fewer epochs. The source code of this paper includes the new optimizer (mass) based on Keras, ResNet-32, CIFAR-10 data pre-process script and the training code of using ResNet-32 to classify CIFAR-10[2]. Besides, we also classified MNIST and CIFAR-10 using CNN and fully connected neural network. According to the paper, we use different learning rates and hyper-parameters to evaluate the optimizer as well.

5 ANALYSIS AND DISCUSSION

5.1 CNN AND RESNET ON CIFAR-10

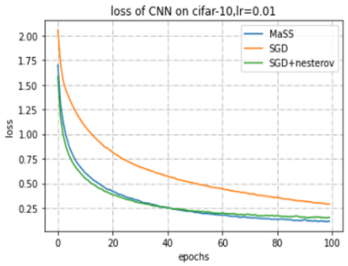

Figure 1: Traning Loss

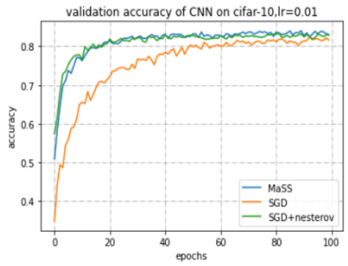

Figure 2: Validation Accuracy

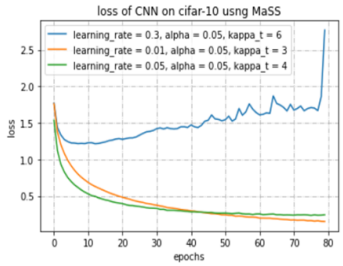

Figure 3: Loss with different lr

When using CNN to classify CIFAR-10, at the same number of iterations, the performance of MaSS is the best, its loss is the lowest, and SGD is the worst in this test as shown in figure 1, 2. It can also be seen that the accuracy of the test obtained by mass is within the acceptable range, which is consistent with the results in the paper. We also tried to test the effect of learning rate on the optimizer by using different learning rate (figure 3). We used CNN to run it on CIFAR-10 and Fine-tuned hyper-parameters. The result shows that when the learning rate is 0.3, the MaSS optimizer can not converge normally, which seems to be different from the result of the paper. The paper shows that at the learning rate of 0.3, the accuracy of mass is still acceptable.

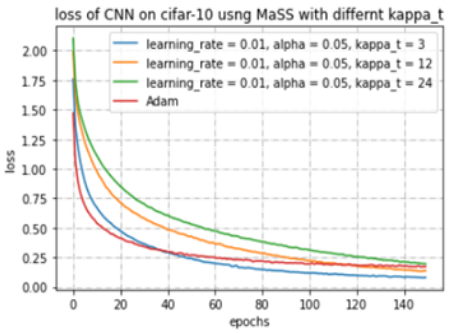

Figure 4: Loss with different κ

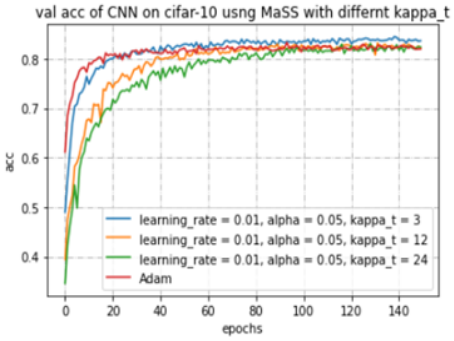

Figure 5: Validation accuracy with different κ

In the case of the same learning rate, adjust the hyper parameter κ. We found κ, the lower the value of it is, the better loss we got (figure 4, 5). Of course, the choice of this parameter is still between 2 and 24 as mentioned in the paper. Adam is used for comparison, which shows that Adam's convergence period is faster than that of mass at the beginning, but under the same number of epoch, the final loss of mass is lower. This shows that when the learning rate is 0.01, the advantage of the mass is shown.

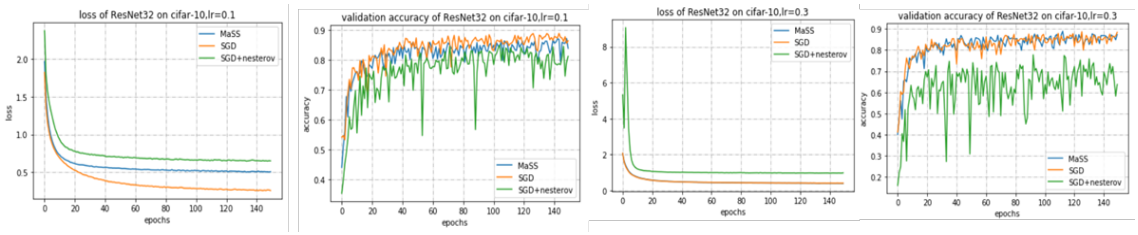

Figure 6: The results of using ResNet-32 on CIFAR-10

The figures above show when ResNet-32 is used to classify CIFAR-10, MaSS performs worse than the common SGD when using 0.1 as the learning rate(figure 6). When the learning rate is 0.3, the performance of MaSS is almost the same as that of SGD, which is basically the same as that of the paper. Although the accuracy of the final test set is not as high as it in the paper, it should be due to the learning rate reduction and the number of epoch.

5.2 RESNET, FCN AND CNN ON MNIST

When using the Resnet on the MNISTin lower learning rate like the 0.01 the MaSS optimizer has the edge on the iteration speed and the accuracy is stable at a very high value. The loss decreases faster than the other three algorithms, but the speed of convergence do not lead other optimizer at the learning rate of 0.1 (figure 7).

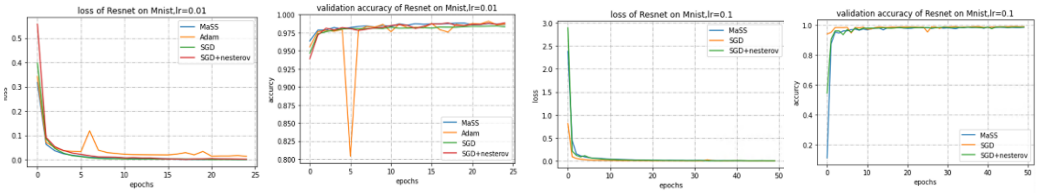

Figure 7: The results of using ResNet-32 on mnist

As the paper said that the MaSS accelerate the convergence comparing to the SGD and SGD+Nesterov, using lower epochs for observing the loss and validation accuracy's change at the first several epochs. In the implementation FCN(fully connected network) model on the MNIST, with the learning rate of 0.1, the MaSS has the fastest convergence of the loss function, but after the 6th or 7th epoch, the speed decrease, but it still faster than the SGD+Nesterov, and a little slower than the SGD. For the validation accuracy of the FCN on the MNIST, the MaSS also follow the trend on the loss and the SGD+Nesterov still has the worst performance. When the learning rate increase to the 0.01, the SGD+Nesterov optimize faster than the MaSS and the SGD has the worst performance.

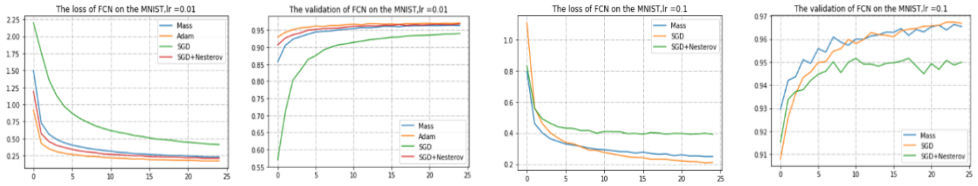

Figure 8: The results of using FCN on mnist

When using the CNN on the MNIST, when the learning rate is 0.3, the optimization effects of SGD, SGD+NESTEROV and MASS on CNN are not different. When the learning efficiency is 0.01, the convergence speed of Loss and accuracy of MASS is obviously faster than SGD and SGD+NESTEROV.

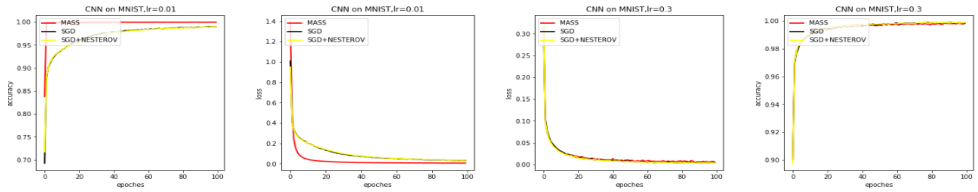

Figure 9: The results of using CNN on mnist

6 Conclusion

Considering the MaSS is an accelerating SGD, in the FCN(fully-connected network), the figure shows this clearly, it is widely known that a larger is preferred for faster convergence, and when the learning rate is 0.1, the MaSS has the fastest convergence speed, but after few epochs, it will not have edges on the SGD but still faster than the MaSS, in the paper, the author also said that a higher alpha can accelerate the convergence, but this situation did not happen in all the models, for example, it seems to be not clear on the ResNet, and another advantages cited on the paper is that the batch size make difference on the optimize algorithms, and it is also proved on the different models, when the batch is small, the MaSS tends to be have a higher speed of the convergence. In conclusion, the results of reproduction are basically the same as those in the paper, but in the case of high learning rate, there is no convergence in the case of mass sometimes.

The source code can be found here: `https://github.com/CodeYudian/COMP6248-Reproducability-Challenge`

References

[1]Liu, C. and Belkin, M., 2020. Accelerating SGD with momentum for over-parameterized learning.

[2]Team, K., 2020. Keras Documentation: Optimizers. [online] Keras.io. Available at: https://keras.io/api/optimizers/ [Accessed 25 May 2020].