Layer Skip Learning using LARS variables for 39% Faster Conversion Time and Lower Bandwidth

Yuki Miyauchi[1], Haruki Mori[1],Tetsuya Youkawa[1], Kazuki Yamada[1], Shintato Izumi[1],
Masahiko Yoshimoto[1], Hiroshi Kawaguchi[1], and Atsuki Inoue[2]
[1]Graduate School of System Informatics, Kobe University, Kobe, Japan
[2]Fujitsu Laboratories Ltd. Computer Systems Laboratory, Kawasaki, Japan
E-mail:miyauchi.yuki@cs28.cs.kobe-u.ac.jp

Abstract— **In this paper, a method for the improvement of the relationship between calculation time and recognition accuracy in deep learning is proposed. A major problem with respect to deep learning is that a large calculation time is required for higher recognition accuracy. Because of this problem, the implementation of deep learning in hardware and its application to real problems are limited. In this study, layer-wise adaptive rate scaling (LARS) variables are adopted to evaluate the necessity of the learning of each layer. When the variable of a certain convolution layer exceeds the threshold value, the learning for that layer is considered unnecessary; thus, the layer is skipped. When a layer recognized as the layer that does not require learning, only the lower layers below than that layer are learned in the next epoch. By adaptively skipping the layer, the calculation time is reduced. Furthermore, the recognition accuracy is improved. Consequently, the proposed methods accelerate the calculation time in VGG-F to achieve the highest accuracy for the top1 and top5 test accuracy by a speed up factor of 2.14, and 2.25, respectively. Moreover, the respective top1 and top5 test accuracy was improved by 3.0 %, and 2.8% which obtained as the final accuracy. In addition, the operation process was reduced by approximately 39.0 %, and required bandwidth was reduced by 38.9 %, when compared with the case of conventional full layer learning.**

Keywords— Deep neural network; Approximate computing; Memory bandwidth reduction;

I. INTRODUCTION

With the rapid advancement in computer performance, significant progress has been made in deep learning, especially in the field of image recognition. Deep learning is therefore expected to be used for the solution of advanced problems, which could not previously be solved using computers. At the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012, the image recognition accuracy using the deep learning method developed by the University of Toronto improved by approximately 10%, when compared with the conventional method [1]. In 2016, AlphaGo won the award for the top player in the world for the Go board game [2]. Deep learning is expected to be utilized in fields that have presented problems that were previously considered impossible. Moreover, deep learning algorithms are expected to be used in practical applications, and whether they should be installed in safety security systems such as automobile-mounted cameras and surveillance cameras is a topic of debate. Furthermore, the fields of application with respect to deep learning are diverse, and include the financial, medical, bioengineering, and Internet of Things (IoT) industries [3]. The development of deep learning technologies is expected to contribute significantly to the progress of a wide range of industrial fields. Moreover, to learn a network that consists of a large amount of data, a large learning time is required, even if current high-speed computing resources are used. A network called FaceNet, which recognizes images of human faces and was reported by Google, achieved a high accuracy of 99.63%. However, the learning of the network required 1000–2000 hours [4]. If the problems become more advanced, and further accuracy is required, it is predicted that a deeper and larger network will be required. The number of layers of the residual network (ResNet), which won the ILSVRC 2015 competition, was 152 [5]. In the future, as the complexity of problems increase, the scale of the required network will increase accordingly; thus, a reduction of the calculation load is essential.

This paper examines a method to improve the relationship between calculation time and recognition accuracy. We prepared unique variable which evaluates the necessity of the learning for each layer. When this variable corresponding to each convolution layer exceeds the specified threshold value, that layer is recognized as an unnecessary layer for the learning. In the next epoch, only the lower layers below than that layer are learned at backpropagation. By adaptively skipping the layer learning process in backpropagation, the improved calculation time and the higher recognition accuracy are realized.

II. RELATED WORK

In the field of neural networks, studies have been conducted on the improvement of the learning efficiency by shortening the calculation time while improving the generalization performance. Dropout is a useful technique that is used to achieve this [6]. Moreover, it is a method that involves the stochastic deactivation of nodes of the hidden layer, to achieve learning when updating. This method makes it possible to forcibly reduce the flexibility of the network, to increase the generalization performance and avoid overfitting. It refers to a state that is learned with respect to the training data; however, it is not adaptable to unknown data. Given that neural networks are complex models, it is prone to overfitting. Although Dropout was originally applied to the fully connected layer, it was confirmed that the performance was improved when applied to a convolution layer. In addition, *deep networks with stochastic depth* can be cited as another method to probabilistically skip learning [7]. It involves stochastic changes to the layer depth

during learning. In a short network, data transmission is performed efficiently, and learning can be conducted within a practical time period. However, the ability of expression is insufficient for complex problems. Conversely, deep networks can add complexity to the structure; however, learning is very difficult, and requires a large amount of time. The aim of *deep networks with stochastic depth* was to shorten the network during learning. This method is very simple because only one hyper parameter decides whether to skip layers. Experiments with VGG-F convolutional neural network (CNN) [8] revealed that the recognition accuracy was improved, in addition to the shortening of the calculation time. VGG-F network architecture has five CNN layers and three fully connected (FC) layers, as presented in Fig. 1 (a). Even in this case, the recognition accuracy was improved. Dropout reduces the network in the horizontal direction by the deactivation of nodes in the hidden layer, but *deep networks with stochastic depth* reduces the network in the vertical direction by changing the number of learning layers.

III. Proposed Method

In this study, the focus is on the weight of each convolution layer. nw_{epoch} for each layer is obtained by the ratio of norm value of the latest weight and the norm of delta weight. In our implementation, the delta weight at each layer are obtained by taking the difference between the first and last weight in an epoch, as shown in the equation (1), (2), and (3) [9]. The variable m means the number of elements of the weight. Fig. 1 (a)-(b) portrays the architecture model of the proposed layer skip learning. As shown in Fig. 1 (a), deep learning is carried out in all the layers with the conventional method. In addition, the method of stochastically skipping learning do not consider the state of each layer. nw_{epoch} is considered to be able to determine the necessity of learning with respect to each convolution layer, and to determine whether or not it occurs. As illustrated in Fig. 1 (b), when nw_{epoch} of a certain convolution layer l exceeds the set threshold value α at N epoch, it is determined that learning is not necessary for that layer. In (N+1) epoch, only layers below that layer are learned in a backward process. The efficiency is improved by adaptively learning only the layers that require learning, and by shortening the calculation time to reach a certain accuracy. This method has two advantages for the improvement of the generalization ability.

- It is easy to tune parameters, given that the judgment criteria for the existence of learning involves only one variable.
- A framework is not chosen, as it is a variable contained in the formula used for learning.

As α is increased, the skip layer decreases. Moreover, if a significantly large α is set, full layer learning is conducted. Algorithm 1 shows a pseudo code of the proposed layer skip learning. Given that VGG-F has large calculation in the first and second convolution layers, it is necessary to set α so as to skip the learning of these two layers as much as possible. Referring to Figs. 2 and 3, the first layer and the second layer occupy 58.4% of multiply-accumulate (MAC) operations and bandwidth. Because the MAC operations in the fully connected layers are much smaller than the convolution layers, these are excluded from the target of the skipping layer.

$$nw_{epoch,l} = \frac{\|w_{epoch,l}\|_2}{\|\triangle w_{epoch,l}\|_2} \tag{1}$$

$$\|w_{epoch,l}\|_2 = \sqrt{\sum_{i=1}^{m} |w_{epoch,l,i}|^2} \tag{2}$$

$$\|\triangle w_{epoch,l}\|_2 = \|w_{epoch,l} - w_{epoch-1,l}\|_2 \tag{3}$$

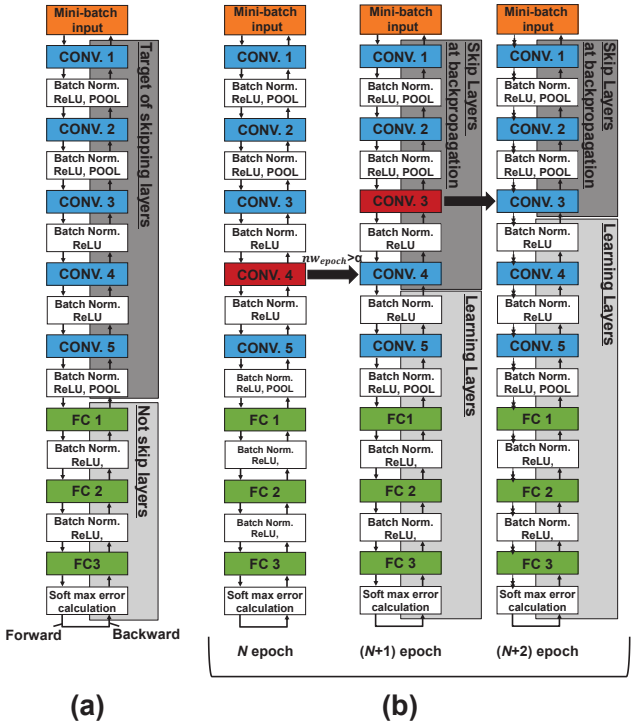

Fig. 1. Architecture of the proposed layer skip learning; (a) VGG-F network architecture, (b) An example of adaptive layer skipping.

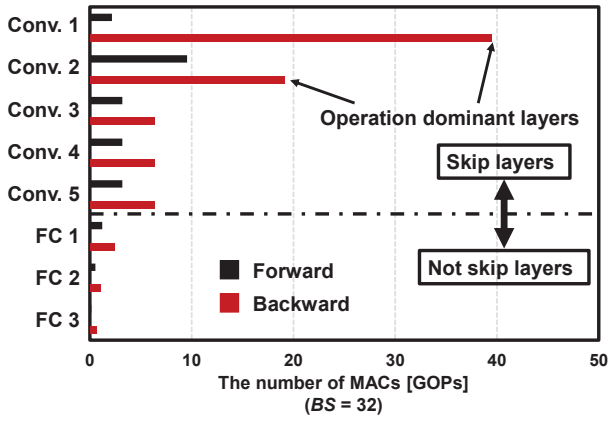

Fig. 2. The number of MAC operations of each layer.

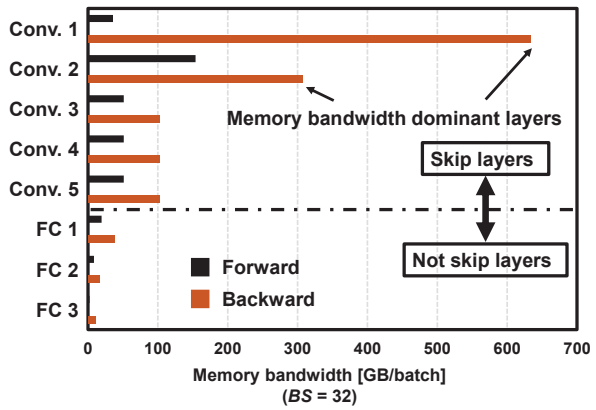

Fig. 3. Memory bandwidth of each layer.

Algorithm 1. Training l layers in the proposed method.

Input: A minibatch of inputs, a vector of activation Y_i, current weight W_i, value that determine the necessity of learning nw_{epoch}, and threshold value α
Output: updated weight W_{i+1}

for epoch=1 to N **do** //N: the number of epochs
  for i=1 to L **do** // L : the number of layers
    if layer type is "conv" **do**
      if $nw_{epoch-1}$ of l layer>α **do**
        if i<=l **do**
          return
        else
          $[dY_i, dW_i]$ =Backward($dY_{i+1}, Y_i\ W_i$)]
          W_{i+1}=UpdateParameters(W_i,dW_i)
        end if
      end if
    end if
  end for
endfor

IV. Experimetal Results

The dataset used in this paper is ImageNet [1], and it has 1.28 million images for training, and 50,000 images for validation. A mini-batch size of 32, learning rate of 0.001, weight decay of 0.0005, momentum of 0.9, and threshold value α of 2 were used. Figs. 4 and 5 present the results of the verification of the transition of the recognition accuracy on a graphics processing unit (GPU). Table I presents the recognition accuracy and calculation time at the end of 20 epochs. Table II presents the calculation time until the proposed method reached the maximum accuracy of the conventional method. The calculation time reached by the proposed method to the highest accuracy of the conventional method was higher for the top1 test accuracy by approximately 2.14, and higher for the top5 test accuracy by approximately 2.25. For the final accuracy as well, the top1 test accuracy was improved by approximately 3.0%, and the top5 test accuracy was improved by approximately 2.8%. By adaptively skip the unnecessary layer for the learning, the proposed method eliminates unnecessary machine time, and avoid falling into local minima and the overfitting. Moreover, while the recognition accuracy monotonically increased with the conventional method, the recognition accuracy of the proposed method sometimes deteriorated gradually. In the epochs wherein the accuracy was degraded, full layer learning was conducted without layer skipping. Skipping layers that do not require learning improves the accuracy over a short time period; however, if the skipping is continued, it adversely affects the final accuracy. For example, if several layers are skipped, the computation time will be shorter; however, the final accuracy will be lower than that for full layer learning. To prevent this, full layer learning is conducted every few epochs. The variables automatically determine whether or not to switch to full-layer learning. By this operation, the reduction of the calculation time and the improvement of the final accuracy are considered as compatible. By skipping layers, the operation can also be reduced. Fig. 6 presents comparison of the number of MAC operations. The operation was successfully reduced by approximately 39.0% using the proposed method, when compared with the case of full layer learning. Moreover, in Fig. 7, the bandwidth is similarly considered. With the proposed method, the bandwidth was reduced by approximately 38.9%, when compared with that of full layer learning.

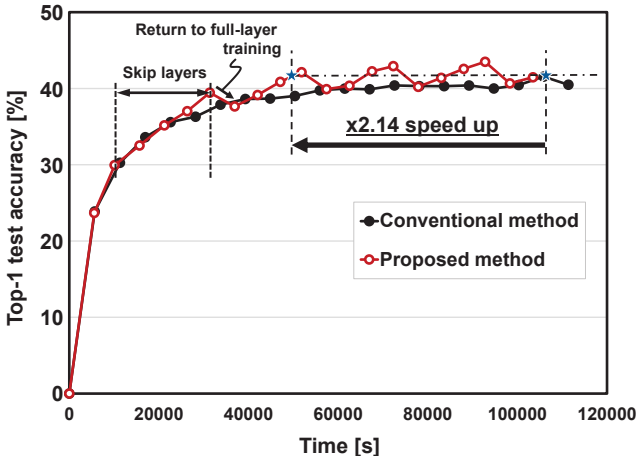

Fig. 4. Learning convergence curve of top1 test accuracy.

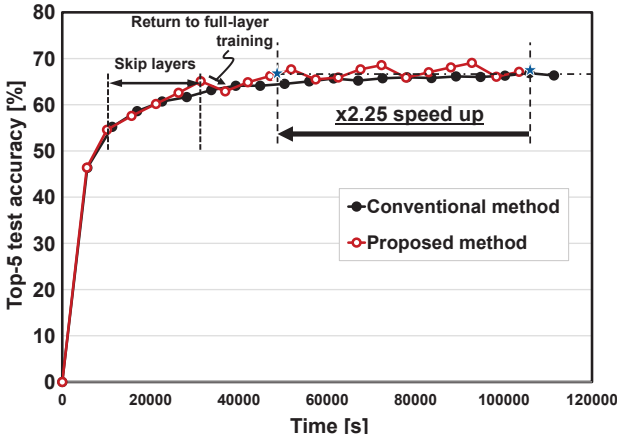

Fig. 5. Top5 learning convergence curve of top5 test accuracy.

TABLE I. RECOGNITION ACCURACY AND CALCULATION TIME AT THE END OF 20 EPOCH.

	Top1 test accuracy	Top5 test accuracy	Time[s]
Conventional method	0.405	0.663	111386[s]
Proposed method	0.435	0.691	103460[s]

TABLE II. CALCULATION TIME TO REACH THE HIGHEST ACCURACY.

	Top1 test accuracy	Top5 test accuracy
Conventional method	105836.2[s]	105836.2[s]
Proposed method	49463.3[s]	47081.8[s]

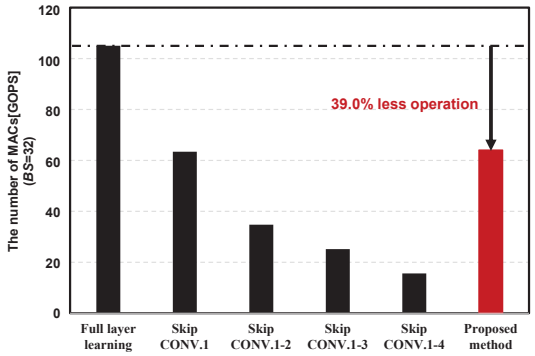

Fig. 6. Comparison of the number of MAC operations.

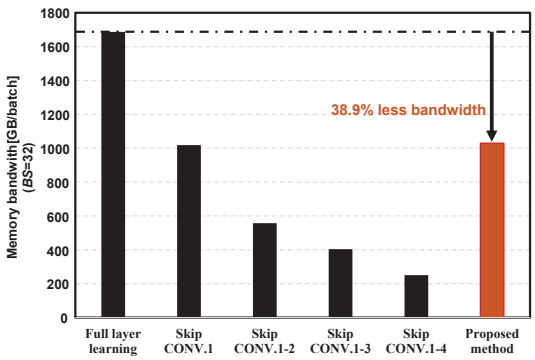

Fig. 7. Comparison of bandwidth.

V. CONCLUSIONS AND FUTURE RESEARCH

To solve real problems, a recognition with a high accuracy beyond that of human recognition is required. However, a large amount of time is required for the construction of a network that can achieve the required accuracy. In addition, the network is expected to become deeper in the future, to achieve the required accuracy. Moreover, the calculation time and hardware load will increase accordingly. Therefore, although it is theoretically possible, its implementation in hardware is difficult. Thus, such systems may not be able to be used for real problems. This study is not from a perspective of software to improve the recognition accuracy, but from a hardware perspective to improve the cost-performance of the recognition accuracy and calculation time. The proposed method improved the relationship between the calculation time and recognition accuracy by adaptively skipping layers that did not require learning. This was achieved by the introduction of variables to determine the necessity of learning. The study simplifies the process of implementation of deep learning in hardware, and can contribute to the solution of real problems. However, a limitation of the proposed method is that it is necessary to set thresholds in a trial and error manner to achieve compatibility between the recognition accuracy and calculation time. Therefore, further research is required to verify the method of setting the threshold, which is robust against the network change. In the VGG-F, the operation of the convolution layer in the upper layer is required more than that in the other layers. Thus, thresholds were set to increase the frequency with which the upper layers were skipped. However, other networks such as ResNet do not necessarily have many operations in the upper layer; thus, it is necessary to consider the number of layers, and the layers that should be reduced according to the network. In some cases, to achieve this, the operation could be adjusted by clumping multiple layers together.

ACKNOWLEDGMENT

This study was carried out by the research grant from Fujitsu Laboratories Ltd.

REFERENCES

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton, “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, 2012.

[2] Nature “Mastering the game of Go with deep neural networks and tree search,” Blog at http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html

[3] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*,vol. 521, pp. 436-444, 2015.

[4] Schroff, Florian, Dmitry Kalenichenko, and James Philbin, “Facenet: A unified embedding for face recognition and clustering,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[6] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” The Journal of Machine Learning Research 15(1) , 2014.

[7] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, Kilian Q. Weinberger, “Deep Networks with Stochastic Depth,” CoRR, arXiv:1603.09382, 2016.

[8] K Chatfield, K Simonyan, A Vedaldi, A Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” In Proc. BMVC, 2014.

[9] Yang You, Igor Gitman, Borls Ginsburg, “Large Batch Training of Convolutional Networks,” arXiv:1708.03888, 2017.

[10] Yang You, Zhao Zhang, Cho-Jui Hsieh, James Demmel, Kurt Keutzer, ”ImageNet Training in Minutes,” arXiv:1709.05011, 2017.

[11] Qian Zhang, Ting Wang, Ye Tian, Feng Yuan, Qiang Xu, “ApproxANN: an approximate computing framework for artificial neural network,” DATE ’15, pp. 701-706, 2015.

[12] Haruki Mori, Tetsuya Youkawa, Shintaro Izumi, Masahiro Yoshimoto, Hiroshi Kawaguchi, Atsuki Inoue, “A layer-block-wise pipeline for memory and bandwidth reduction in distributed deep learning,” Machine Learning for Signal Processing (MLSP), 2017.