A Speed-up Channel Attention Technique for Accelerating the Learning Curve of a Binarized Squeeze-and-Excitation (SE) Based ResNet Model

Wu Shaoqing [1,*] and Hiroyuki Yamauchi [2,*]

[1] Graduate School, Fukuoka Institute of Technology, Fukuoka, Japan
[2] Department of Computer Science and Engineering, Fukuoka Institute of Technology, Fukuoka, Japan
Email: mfm22202@bene.fit.ac.jp (W.S.); yamauchi@fit.ac.jp (H.Y.)
*Corresponding author

***Abstract*—The use of 1-bit representation for network weights, as opposed to the conventional 32-bit, has been investigated to save on the required power and memory footprint. Squeeze-and-Excitation (SE) based channel attention techniques aim to further reduce the number of parameters by eliminating redundant channels. However, this approach leads to a significant drawback of an unstable and slow learning curve, especially when compared to fitting parameters in SE networks. To address this issue, this paper presents the first attempt to accelerate the learning curve, even with a 1-bit representation for weights across the entire Squeeze-and-Excitation Residual Network (SEResNet14). The proposed technique within the SE module significantly speeds up channel attention, yielding a steeper learning curve for the network. We also extensively investigate the impact of activation functions within the SE module, aiming to understand their performance-enhancing attributes when applied with the proposed technique. Experimental results demonstrate that even under stringent compression, an appropriate choice of activation function can still ensure the efficacy of our technique in the SE module. We found that the proposed technique results in: (1) a 60% reduction in the required number of epochs to achieve an error rate of 0.3; and (2) a decrease in the error rate by approximately 44% at the 10th epoch, compared to a baseline method that does not use the proposed scheme.**

***Keywords*—Residual Network 14 (ResNet14), CIFAR-10, Squeeze-and-Excitation (SE) attention mechanism, 1-bit quantization, model compression, activation functions, channel feature maps binarization, ultra-compact AI deployment**

I. Introduction

Deep learning has revolutionized various applications within the realm of computer vision, with Convolutional Neural Networks (CNNs) emerging as a predominant architecture. Modern high-performance CNNs often consist of recurring blocks with identical structures [1–7], leveraging principles from residual learning [8–10], and utilizing depthwise separable convolutions [11]. While these networks have demonstrated an impressive performance with 32-bit representation for the weight and the activation, it poses significant challenges to deploy them in real-world scenarios, especially on stringent power and memory-footprint constrained devices.

One approach to address this issue is to use the binarized (using 1 bit) representations for the model parameters, aiming to reduce the required power and memory footprint without any significantly sacrificing performances. Since the power consumption in the network is almost governed by the accesses to external memory (i.e., DRAM), which are placed far away from the AI chip, eliminating the need for the Dynamic Random-Access Memory (DRAM) accesses is the most essential attempt. It could only be done by reducing the required number of parameters to 1/100 so that the almost parameters can be stored in the AI chip and the accesses to the DRAM can be eliminated, as shown in Fig. 1. Since an external DRAM access cause a 100× larger power consumption than the internal one, it can reduce the power consumption to 1/100.

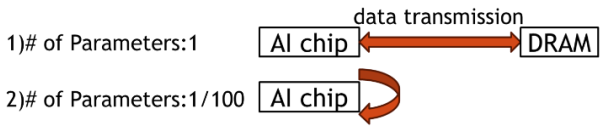


Fig. 1. Concept of how to reduce the energy consumption to 1/100. This can only be done by reducing the number of parameters to 1/00.

Residual Networks (ResNets) were introduced by He *et al.* [12] to solve the issues of loss in accuracy caused by the gradient vanishing problem. We noticed that this invention of the ResNets cause to return to rise of exponentially increased number of parameters in deeper networks and the increased pressure to introduce the stringent reduction of the parameters. The Squeeze-and-Excitation (SE) attention technique [13–16], introduced by Hu *et al.*, offers a channel-wise recalibration to enhance

Manuscript received December 7, 2023; revised December 26, 2023; accepted January 23, 2024; published May 10, 2024.

model accuracy without significant computational overhead.

Li *et al*. [17] integrated Squeeze-and-Excitation (SE) blocks into the High-Resolution Network (HRNet), leveraging the interdependencies among channels. They employed the Squeeze-and-Excitation (SE) attention mechanism to enhance and suppress features based on these dependencies. The proposed SE-HRNet improves the distinction of scene categories by utilizing rich features.

Zhang *et al*. [18] used the Squeeze-and-Excitation (SE) attention mechanism to improve the network's ability to identify key features for segmentation tasks. Their approach focuses on analyzing feature relationships without adding complexity or new spatial dimensions to the model. This approach improved the predictive accuracy of MRSE-Net in global remote sensing image water extraction tasks.

We also noticed that this SE network can be used for reducing the number of parameters by eliminating redundant channels.

Thus, key techniques for reducing the number of parameters are: (1) compact binarized SE-based ResNets (e.g., ResNet14 in this work); and (2) some technologies to prevent intolerable side effects caused by using the binarization (1-bit quantization) techniques.

Among the popular compression strategies, quantization, especially 1-bit quantization, has shown promise in drastically reducing model size. As shown in Fig. 2, we can reduce the number of parameters to 1/32 of the original size by using the binarization technique. However, the size reduction by only relying on 1-bit quantization is not enough to store the almost parameters in the AI chip when considering in the deployment of miniaturized AI. Thus, we noticed that the channel and spatial attention techniques will be needed for further reduction to 1/4. In this paper, only channel attention by SE module will be discussed due to the space limitation. If we could reduce the number of model parameters to 1/100 of the original size, which is current our research goal, then almost data accesses between the AI chip and the external memory (i.e., DRAM), will not be needed anymore, as shown in Fig. 1.

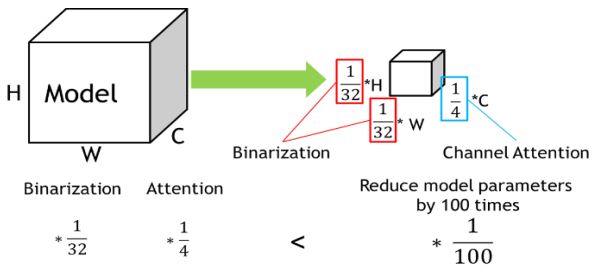


Fig. 2. Conceptual diagram of how to squeeze the model parameter size to less than 1/100 (i.e., 1/128=1/32×1/4).

Thus, we expect to apply the SE attention mechanism even under the 1-bit quantization condition to eliminate the redundant parameters in the channel direction.

However, 1-bit SE mechanism and side effects has not been discussed in the previous papers. Thus, this paper is the first paper to propose and discuss on this topic.

Activation functions play a crucial role within the SE module, as they significantly influence the recalibration process. Through the SE module decision, lower weights to certain channels are assigned, which means that these channels can contribute less to the output. In practical applications, further pruning of these channels can contribute to reduce the number of parameters.

Based on the experiment results of SE attention, we have noticed that 1/2 of the channel can be pruned and the 1/2 of the parameters can be removed.

We have also noticed that the spatial attention can be applied to further eliminate the parameters in the spatial direction, making it possible to reduce the model parameter quantity by over 100 times, as shown in Fig. 2. Convolutional Block Attention Module (CBAM) [19] integrates both channel and spatial attention mechanisms, and it might be the direction we choose for our next experiment.

In this study, we explore the accuracy and speed impacts of the novel integration of the SE attention with 1-bit quantization in the context of ResNet14 trained on the CIFAR-10 dataset. The CIFAR-10 dataset [20], consisting of 60,000 32×32 color images in 10 classes, has become a benchmark for evaluating the performance of various deep learning models.

Our primary goal is to investigate how extreme compression (1-bit quantization) affects the efficacy of the SE attention modules, providing the insights for further optimizations in ultra-compact AI deployments.

We have proposed a method to channel Feature Maps Binarization (FMB), in which some intermediate values in the channel attention during the early stages of training are forcibly binarized to investigate its impact on the model accuracy. This study compared the learning curves to investigate how much the proposed technique can contribute to reduce the error rate and required number of epochs to reach a certain error rate among the cases for using the different precisions: (1) float32bit as a baseline; (2) 1bit without using the proposed FMB technique; and (3) 1bit with using the FMB. We also examined the impact of the activation function of the SE module.

The main contributions of this article can be summarized as follows.

(1) We have investigated the impact of the activation functions in the SE module under 1-bit quantization on the model accuracy and speed of the learning curve. We compared those between the two cases for using the Tanh and Sigmoid activation functions.

(2) We have proposed the FMB technique and demonstrated that even under 1-bit quantization conditions, binarizing the output channel feature maps in the SE module is effective for enhancing model accuracy.

(3) Based on the binarization of channel feature maps, we also conducted additional discussions on the choice of activation functions.

The rest of this article is organized as follows. Section II elucidates the issues encountered with SEResNet14 under 1bit binarization. In Section III, we provide a detailed introduction to our proposed technique. We discussed the results in Section IV. In Section V, we conclude this article.

II. Problem Statement

A. Accuracy Loss Caused by Low Bits

When the parameters in deep learning models are quantized with 1-bit, one primary concern is unstable bang-bang behaviors in the learning curves, resulting in the loss in accuracy and error reduction speed. 1-bit quantization (i.e., binarization) is a method to represent the numerical values for the weights and activation in the neural network instead of the full bit (32 bits) representation. This process is particularly critical for deploying the models on the resource-constrained devices where memory footprint and computational power are limited.

Abdolrashidi *et al.* [21] achieve state-of-the-art results on ImageNet for 4-bit ResNet-50 with quantization-aware training, obtaining a top-1 eval accuracy of 77.09%. They concluded that 4-bit quantization is the optimal choice for balancing accuracy and parameter quantity, but in this article, extreme binarization is our goal of effort.

However, the shift from a 32-bit floating-point precision to a binarized one inherently causes approximating error (i.e., quantization error). For instance, in floating-point representation, there's a wide range of values that can be captured, ensuring that the minute differences between weights can be distinguished. But when we move to the 1-bit representations, many different weights may get rounded off to the same value (e.g., +1/−1) due to the lack of granularity, leading to a significant loss of precision.

In particular, big quantization errors in 1-bit representation can lead to an unstable bang-bang behavior in the learning curves, resulting in a slow-down of the error reduction speed. This effect becomes particularly noticeable in the networks with delicate architectures or those handling complex tasks. The balance between the number of bits used for quantization and the accuracy of the network becomes a critical design consideration.

When directly quantizing the 32-bit SEResNet14 to 1-bit, there is the significant losses in an accuracy and the speed of error reduction, as shown in Fig. 3.

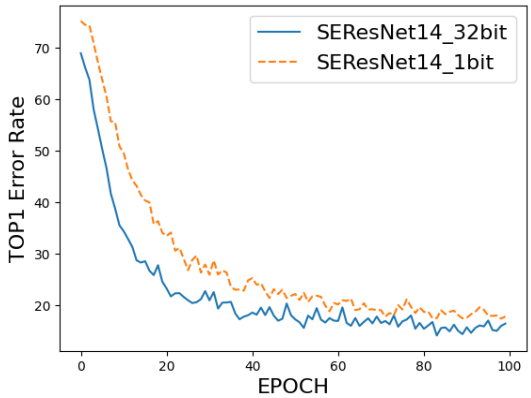


Fig. 3. Learning curve comparisons between the two cases for using Float32 and 1 bit for SEResNet14 model.

In some cases, post-quantization fine-tuning might be employed to recover some of the lost accuracy. This involves retraining the quantized model for a few epochs to adjust to its new, approximated weight values. However, even with fine-tuning, there might still be a noticeable drop in performance, especially when extremely low bit-widths (i.e., 1-bit) are chosen.

That is to say that, while low-bit quantization offers advantages in memory savings and computational efficiency, it comes at the cost of accuracy due to the inherent approximation involved. The challenge is to find the sweet spot where the benefits of quantization outweigh the potential decrease in model performance.

B. Completion Degree of Channel Feature Map

Based on our research, we found that the channel attention for the shallow compact networks (ResNet14) cannot be well achieved in the early stages of training, and it usually takes a few more EPOCHs to achieve a complete channel feature map. The following figure shows the channel attention map outputs in the first round of EPOCH from the SE module 1, located in the first block of SEResNet14. It shows that most of the attention outputs are perfectly classified as 0 and 1, but there are still a few intermediate values, as shown in Fig. 4. The existence of these intermediate values may affect the convergence speed of the whole model.

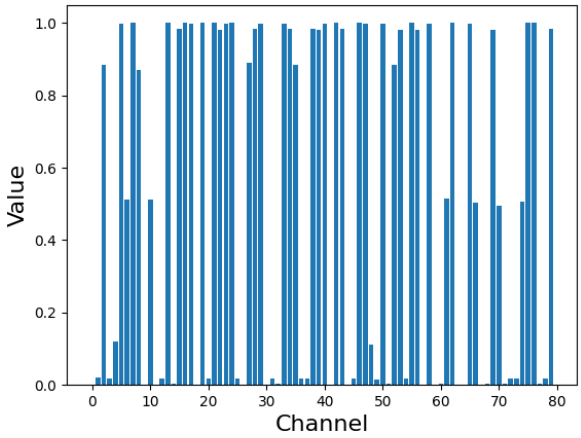


Fig 4. The channel feature map outputs from SE1 at the first round of training.

We found that those intermediate values are almost completely classified as 0 and 1 at the round of 10 (EPOCHs = 10), as shown in Fig. 5.

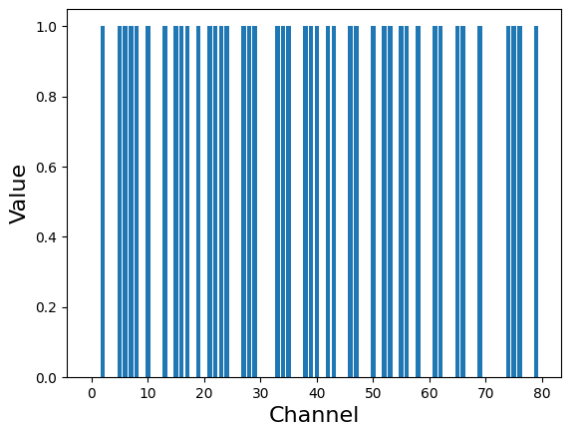


Fig. 5. The channel feature map outputs from SE1 at the 10th round of training.

III. Proposed Techniques

This section introduces our two proposed methods: (A) replacing the activation function in the SE module; and (B) the method to channel Feature Maps Binarization (FMB).

A. Activation Function of SE Module

In order to clarify how much of each feature map is useless and useful in the channel direction in practical use, we saved the number of the required feature maps in the channels at different depths of the network and attempted to find which connections can be pruned from them.

We found through many trials that the selection between the tanh and sigmoid functions in the SE module can affect the overall model accuracy. In recent years, various studies have highlighted that, under low-bit quantization scenarios, the Tanh activation function outperforms many of its counterparts [22]. In this experiment, we tested the model accuracy using the tanh and sigmoid functions separately, while combining with another proposed channel Feature Maps Binarization (FMB) method.

B. Channel Feature Map Binarization (FMB)

We found that the channel Feature Maps Binarization (FMB) enforces binary classification on certain intermediate values during the initial phase of model training. For the Sigmoid function, whose outputs are widely distributed in the range from 0 to 1, it takes more EPOCHs to finalize the decision. To solve this issue, we newly introduce the threshold to accelerate the decision. For example, the threshold for Sigmoid function is 0.5, output values greater than 0.5 are classified as 1, while values less than 0.5 are classified as 0.

In Eq. (1), F denotes the input feature map, which is a three-dimensional array with dimensions C×H×W, where C, H, and W represent the number of channels, the height, and the width, respectively.

$$F \in R^{C\times H\times W} \tag{1}$$

In Eq. (2), M_c represents the channel weights output by the SE module, which is also a three-dimensional array, but each channel (of the C channels) has only one unit (1×1), meaning that each channel has a specific weight.

$$M_c \in R^{C\times 1\times 1} \tag{2}$$

In Eq. (3), these two formulas define two weight matrices W_0 and W_1, which are used in the fully connected layers of the SE module. Here, r is a reduction ratio, used to adjust the complexity and the number of parameters of the model. W_0 reduces the number of channels from C to C/r, while W_1 increases it back from C/r to C. In this study, the reduction ratio is set to 1.

$$W_0 \in R^{C/r\times C} \; and \; W_1 \in R^{C\times C/r} \tag{3}$$

In Eq. (4), first, global average pooling is applied to the input feature map F to obtain the global feature of each channel. Then, two fully connected layers, W_0 and W_1, are used to learn the relationships between channels, with an activation function $ReLU$ between these two fully connected layers. Finally, either $Sigmoid$ or $Tanh$ function is applied to output the weights M_c for each channel.

$$\begin{aligned} M_c(F) &= Sigmoid\big(AvgPool(F)\big) \\ &= Sigmoid\left(W_1\left(ReLU\left(W_0\left(F_{avg}^c\right)\right)\right)\right) \end{aligned} \tag{4}$$

In contrast, for the $Sigmoid$ function, whose outputs values are distributed between 0 and 1, we use 0.5 as a threshold. Values greater than 0.5 are set to 1, and those less than 0.5 are set to 0, as shown in Eq. (5).

$$M_c'\big(M_c(F)\big) = \begin{cases} 1, & M_c(F) \geq 0.5 \\ 0, & M_c(F) < 0.5 \end{cases} \tag{5}$$

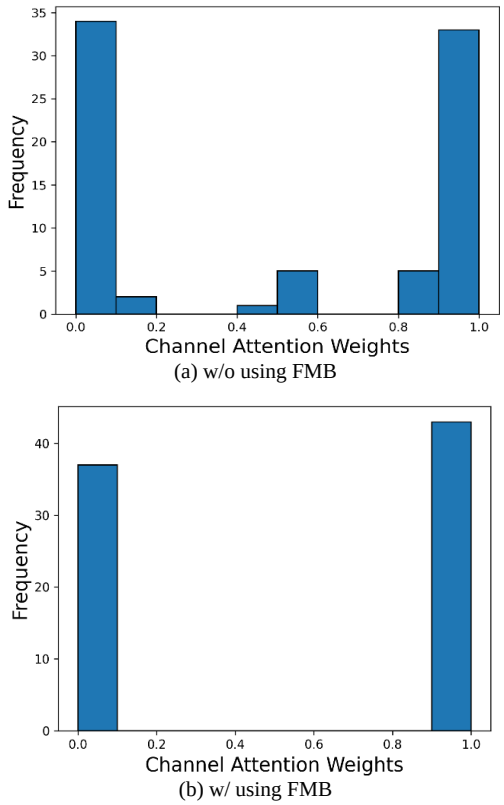
For the Tanh function, whose outputs values are distributed between −1 and 1, we use 0 as the threshold. Values greater than 0 are set to 1, and those less than 0 are set to −1, as shown in Eq. (6).

$$M_c'\big(M_c(F)\big) = \begin{cases} 1, & M_c(F) \geq 0 \\ -1, & M_c(F) < 0 \end{cases} \tag{6}$$

In Eq. (7), the final output F' of the SE Module is given by the product of the original feature map F with the adjusted channel weights M_c' obtained above.

$$F' = M_c'\big(M_c(F)\big) \odot F \tag{7}$$

In Fig. 6, to more intuitively demonstrate the effect of the proposed FMB, we have plotted the frequency histograms of the channel feature maps output from the SE1 module before and after binarization.



(a) w/o using FMB

(b) w/ using FMB

Fig. 6. Comparison of output feature map value distributions between the cases of (a) w/o and (b) w/ using FMB technique.

In PyTorch, the "torch.where" function is an extremely useful tool that allows for selecting elements from two tensors based on a condition. The functionality of Eq. (1) and Eq. (2) can be effectively implemented using "torch.where".

IV. Results and Discussion

A. Impacts of Activation Function of SE Module

In the Squeeze and Excitation (SE) module, the choice of activation function depends on the required nonlinear conversion effect, and the Rectified Linear Unit (ReLU) function is usually used to introduce nonlinearity in the Squeezed operation. Sigmoid can convert values between 0 and 1 to generate attention weights, typically during the SE module's citation process.

However, we have found that using the sigmoid activation function for the SEResNet14 under the stringent condition of using 1-bit quantization does not necessarily bring a good result.

Fig. 7 shows the comparisons of the learning curves for 1bit quantized SEResNet14 between the two cases for using Sigmoid and Tanh activation functions in SE modules (The error rate mentioned in this study refers to the TOP1 error rate). As can be seen in Fig. 7, the case for using the Tanh for the activation function is slightly better than the case for using the Sigmoid function under the stringent conditions of 1-bit quantization environment.

This can be due to the advantages from the Tanh, which the 0 inputs will be mapped near zero and differentiable and negative and positive inputs will be mapped more strongly toward −1 and 1, i.e., strong splitting manner compared with the cases for the Sigmoid. It can be said that using Tanh provides a better matching with 1-bit quantization technique without using the FMB as the activation function, resulting in larger contribution for better learning curves. This will be discussed in the following sub-section more in detail.

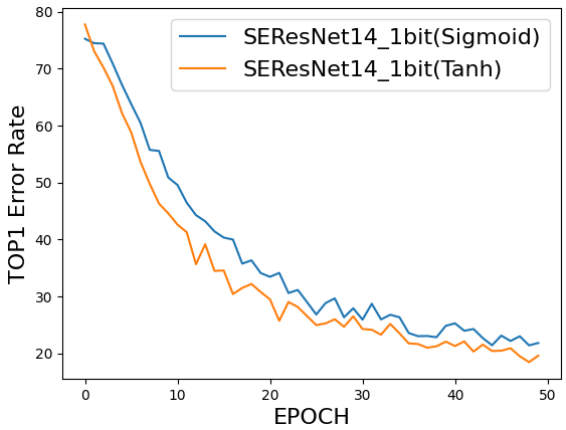


Fig. 7. Comparisons of the learning curves between the two cases between using Sigmoid and Tanh for the activation functions in 1bit quantized SE module for SEResNet14.

B. Impacts of FMB Technique

In Table I, we picked up the value from Fig. 8 to compare how much each technique combination contribute to reduce the error rate for the four cases in the SE processes using: (1) Sigmoid with FMB; (2) Tanh with FMB; (3) Tanh only w/o using FMB; and (4) Sigmoid only w/o using FMB as baseline. Based on Table I, it is found that the combination of using the Sigmoid and the FMB can provide the best reduction of the error rate and its speed.

TABLE I. Error Rate Reduction Percentages brought by Using the Combinations of FMB and Activation Functions for 1-Bit after Training 9 Epochs

Activation Functions	w/ using FMB	w/o using FMB
Sigmoid	44.64%	0% (baseline)
Tanh	40.72%	19.95%

In Table II, we compare the error rate for the 1-bit quantized model at the 6th epoch. Here, we introduce the results for the conventionally usual used Float32 case for comparison. We found that even under 1-bit quantization conditions, thanks to using the Sigmoid/Tanh active functions combined with the FMB method in the SE module can outperform the error rate for the case of using the Float32 in the early stages of training. This highlights the efficacy of the FMB method in enhancing error rate and its reduction speed.

TABLE II. Error Rate Comparisons at 6th Epoch between the Three Cases of (1) Sigmoid with FMB in 1-bit, (2) Tanh with FMB in 1-bit, and (3) Sigmoid w/o Using FMB in 32-bit

Evaluation Metrics	Sigmoid w/ using FMB	Tanh w/ using FMB	Float32 w/o using FMB
Precision	1-bit	1-bit	32bit FP
Activation	Sigmoid	Tanh	Sigmoid
Error Rate	36.27%	46.54%	50.29%

We also presented detailed learning curves in Fig. 8 to illustrate the extent of improvement when using FMB compared to not using it. It is most worthy of notice that the relationship of the degree of enhancement between Sigmoid and Tanh is completely reversed depending on the cases for w/ and w/o using FMB. It is clearly shown that Sigmoid and Tanh provide a better error reduction, respectively, as shown in Fig. 8.

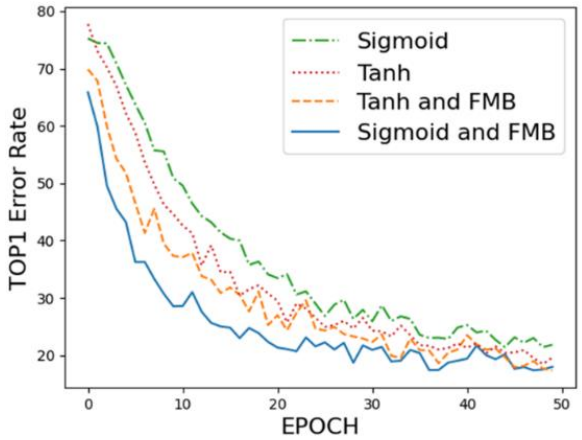


Fig. 8. When employing various activation functions with FMB, the model's error rate decline curve is presented as follows in 1bit SEResNet14.

It is evident from for the early stages of training, such as the first 50 epochs, that the model with the Sigmoid activation function combined with the FMB technique in the SE module provides the fastest decline in the error rate. This is contrary to the phenomenon observed when the cases without using FMB, where Tanh outperforms the Sigmoid function, as shown in Fig. 7.

If we set a fixed target error rate of 30%, using the Sigmoid function combined with the FMB technique provides to meet the target within 9 epochs. In contrast, when only using the Sigmoid function, it takes 23 epochs. This means a 60% speed-up of the learning curve, which the required EPOCHs are reduced from 23 to 9, is provided by using the proposed FMB with Sigmoid active function in the SE module, as shown in Table III.

TABLE III. COMPARISON OF THE REQUIRED EPCHS TO REACH AT 30% ERROR RATE

Evaluation Metrics	w/ using FMB	w/o using FMB
Epochs	9	23
Error rate	30%	30%
Precision	1-bit	1-bit

Fig. 9 shows the acceleration of the learning curve with using the FMB. It is due to the speed up of the channel attention provided by the FBM mechanism.

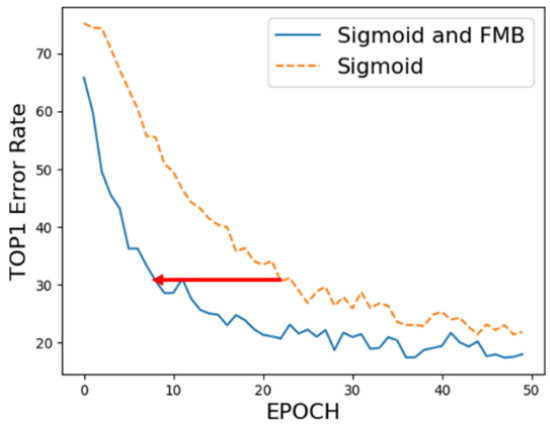


Fig. 9. Comparisons of the learning curves between w/ and w/o using FMB for using Sigmoid activation for 1-bit precision model.

C. Discussions

The technique proposed in this study, including the binarization of SE channel feature maps and the replacement of activation functions, can significantly mitigate the error rate degradation caused by 1-bit quantization. This ensures that the faster learning curve for the 1-bit quantized model can be realized even if compared with the FP precision in the initial training phases. As shown in Fig. 10, after 100 training epochs, the differences in the error rate seem to be smaller. This is because more other factors can be involved and highlighted around the error rate of 12% for the TOP1 accuracy. To make the effectiveness of the proposed technique more highlighted, we focused on the steep error rate reduction phases in this paper.

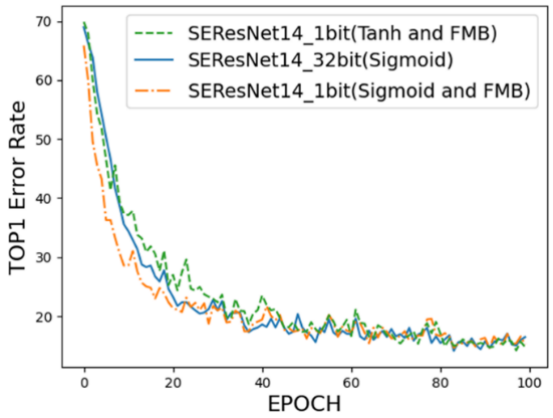


Fig. 10. Comparisons of the learning curves across the whole EPCHs until 100.

V. CONCLUSION

This paper proposed the channel Feature Maps Binarization (FMB) technique and investigates the optimal pairing with activation functions to lower the error rate and increase the speed of error reduction during phases of steep learning curves.

Based on the results, the proposed FMB technique for speed up the channel attention can significantly contribute to realize the reductions in the error rate and the required number of the training iterations to achieve the target error rate.

According to the summaries in Tables I–III, the following reductions compared with the baseline are provided by the proposed techniques:

1) About 45% error rate reduction with the combination of the FMB and Sigmoid activation functions (Table I).
2) Even compared with the 32bit FP precision (50.3%), smaller error rate of 36.3% can be realized (Table II).
3) The required number of epochs can be reduced from 23 to 9 to reach the error rate of 30% (Table III).

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Wu and Yamachi conducted the research and analyzed the data and wrote the paper. Wu conducted an experiment. All authors had approved the final version.

REFERENCES

[1] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint, arXiv:1704.04861, 2017.

[2] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 2018, pp. 4510–4520.

[3] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. International Conference on Machine Learning*, 2019, vol. 97, pp. 6105–6114.
[4] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 2018, pp. 6848–6856.
[5] S. Chakraborty, Amrita, T. Choudhury, R. Sille, C. Dutta, and B. K. Dewangan, "Multi-view deep CNN for automated target recognition and classification of synthetic aperture radar image," *Journal of Advances in Information Technology*, vol. 13, no. 5, pp. 413–422, October 2022.
[6] M. Ashrafuzzaman, S. Saha, and K. Nur, "Prediction of stroke disease using deep CNN based approach," *Journal of Advances in Information Technology*, vol. 13, no. 6, pp. 604–613, December 2022.
[7] S. N. Kumar and C. S. Kumar, "Fusion of CNN-QCSO for content based image retrieval," *Journal of Advances in Information Technology*, vol. 14, no. 4, pp. 668–673, 2023.
[8] K. He *et al*., "Deep residual learning for image recognition," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[9] M.S. Puchaicela-Lozano, L. Zhinin-Vera, A. J. Andrade-Reyes, D. M. Baque-Arteaga, C. Cadena-Morejón, A. Tirado-Espín, L. Ramírez-Cando, D. Almeida-Galárraga, J. Cruz-Varela, and F. V. Meneses, "Deep learning for glaucoma detection: R-CNN ResNet-50 and image segmentation," *Journal of Advances in Information Technology*, vol. 14, no. 6, pp. 1186–1197, 2023.
[10] S. Bunrit, N. Kerdprasop, and K. Kerdprasop, "Improving the representation of CNN based features by autoencoder for a task of construction material image classification," *Journal of Advances in Information Technology*, vol. 11, no. 4, pp. 192–199, November 2020. doi: 10.12720/jait.11.4.192-199
[11] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 2017, pp. 1800–1807.
[12] A. Krizhevsky. (2009). Learning multiple layers of features from tiny images. [Online]. Available: https://api.semanticscholar.org/CorpusID:18268744
[13] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.
[14] H. Zhu *et al*., "MS-HNN: Multi-scale hierarchical neural network with squeeze and excitation block for neonatal sleep staging using a single-channel EEG," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 2195–2204, 2023. doi: 10.1109/TNSRE.2023.3266876
[15] X. Jin, Y. Li, J. Wan, X. Lyu, P. Ren, and J. Shang, "MODIS green-tide detection with a squeeze and excitation oriented generative adversarial network," *IEEE Access*, vol. 10, pp. 60294–60305, 2022. doi: 10.1109/ACCESS.2022.3180331
[16] J. Ai, S. Hou, M. Wu, B. Chen, and H. Yan, "MPGSE-D-LinkNet: multiple-parameters-guided squeeze-and-excitation integrated D-LinkNet for road extraction in remote sensing imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 5508205, 2023. doi: 10.1109/LGRS.2023.3306725
[17] L. Li, T. Tian, H. Li, and L. Wang, "SE-HRNet: A deep high-resolution network with attention for remote sensing scene classification," in *Proc. the 2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2020)*, Waikoloa, HI, USA, 2020, pp. 533–536. doi: 10.1109/IGARSS39084.2020.9324633
[18] X. Zhang, J. Li, and Z. Hua, "MRSE-Net: Multiscale residuals and Se-attention network for water body segmentation from satellite images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 5049–5064, 2022. doi: 10.1109/JSTARS.2022.3185245
[19] S. Woo *et al.*, "CBAM: Convolutional block attention module," in *Proc. the European Conference on Computer Vision (ECCV)*, 2018.
[20] H. Bai *et al.*, "BinaryBERT: Pushing the limit of BERT quantization," in *Proc. Annual Meeting of the Association for Computational Linguistics*, 2020.
[21] A. Abdolrashidi *et al*., "Pareto-optimal quantized ResNet is mostly 4-bit," in *Proc. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, USA, 2021, pp. 3085–3093. doi: 10.1109/CVPRW53098.2021.00345
[22] K. Abdelouahab, M. Pelcat, and F. Berry, "Why TanH is a hardware friendly activation function for CNNs," in *Proc. the 11th International Conference on Distributed Smart Cameras (ICDSC 2017)*, Association for Computing Machinery, New York, USA, 2017, pp. 199–201. https://doi.org/10.1145/3131885.3131937