S-DFP: shifted dynamic fixed point for quantized deep neural network training

Yasufumi Sakai¹ · Yutaka Tamiya¹

Received: 2 April 2021 / Accepted: 4 December 2021
© The Author(s) 2021

Abstract

Recent advances in deep neural networks have achieved higher accuracy with more complex models. Nevertheless, they require much longer training time. To reduce the training time, training methods using quantized weight, activation, and gradient have been proposed. Neural network calculation by integer format improves the energy efficiency of hardware for deep learning models. Therefore, training methods for deep neural networks with fixed point format have been proposed. However, the narrow data representation range of the fixed point format degrades neural network accuracy. In this work, we propose a new fixed point format named shifted dynamic fixed point (S-DFP) to prevent accuracy degradation in quantized neural networks training. S-DFP can change the data representation range of dynamic fixed point format by adding bias to the exponent. We evaluated the effectiveness of S-DFP for quantized neural network training on the ImageNet task using ResNet-34, ResNet-50, ResNet-101 and ResNet-152. For example, the accuracy of quantized ResNet-152 is improved from 76.6% with conventional 8-bit DFP to 77.6% with 8-bit S-DFP.

Keywords Neural networks · Quantization · Training

1 Introduction

Deep neural networks (DNNs) have shown remarkable performance on various tasks such as classification tasks [1, 2], semantic segmentation [3], and object detection [4, 5]. However, along with improved network performance, the DNN training time has increased considerably. Moreover, network models have deepened and become more complex.

To reduce the DNN training time, training methods using quantized neural networks have been proposed. Two earlier reports [6, 7] have proposed DNN training methods using quantized weight and activation. One reported method [8] uses 16-bit floating point (FP16) to quantize the gradient in addition to weight and activation for DNN training. The DNN training time is reduced according to

the precision. Therefore, it is effective for using reduced precision for data representation and calculation [9]. Some earlier methods [10, 11] transform the data format representing single precision floating point (FP32) into narrower bit-width floating point format such as FP16 and 8-bit floating point (FP8). Calculations using integer precision improve the energy efficiency of hardware for deep learning models in comparison with the use of floating point format [12]. However, because the data representation range of integers is narrower than that of floating point, the accuracy of the quantized model with integers is generally lower than when using floating point for quantization. Several methods [13, 14] have been proposed to prevent degradation of accuracy when using integers. One method [13] uses 16-bit dynamic fixed point precision (DFP16) for DNN training on the ImageNet task with ResNet-50 [15] to prevent accuracy degradation. Although another method [14] proposes DNN training using 8-bit fixed point, [14] does not evaluate the accuracy using complex DNN models such as ResNet.

In this paper, we propose a new fixed point format named shifted dynamic fixed point (S-DFP) to prevent accuracy degradation due to quantized neural networks

✉ Yasufumi Sakai
sakaiyasufumi@fujitsu.com

Yutaka Tamiya
tamiya.yutaka@fujitsu.com

¹ Fujitsu Limited, 4-1-1, Kamiodanaka, Nakahara, Kawasaki, Japan

training. S-DFP can change the data representation range of dynamic fixed point format by adding bias to the exponent. We evaluated the effectiveness of S-DFP for quantized neural network training on the ImageNet task using ResNet-34, ResNet-50, ResNet-101, and ResNet-152. Using the proposed method, the evaluated models can be trained using 8-bit S-DFP (S-DFP8) without any marked accuracy degradation. For example, when the accuracy of evaluated ResNet-152 with FP32 is 77.6%, the accuracy of quantized ResNet-152 was improved from 76.6% with conventional DFP8 to 77.6% with S-DFP8.

Our contributions to quantized DNNs training are the following.

- We propose a training method of quantized DNN models using S-DFP8 with no marked accuracy degradation.
- The proposed data format S-DFP can change the data representation range of quantized variables by adding bias to the shared exponent of DFP. Accuracy degradation on quantized DNN training is prevented by changing the data representation range of the weight gradient with S-DFP during weight update.
- We demonstrate that our proposed method can be trained on the ImageNet task with ResNet-34, ResNet-50, ResNet-101, and ResNet-152 with no marked accuracy degradation.

The rest of this paper, the related earlier works are reviewed in Sect. 2, and Section 3 describes the proposed S-DFP format for quantized neural networks training. Then, the experimental results are presented in Sect. 4. Finally, the conclusions and future works are presented in Sect. 5.

2 Related works

A brief review of earlier works assessing DNN quantization methods is presented below.

2.1 Quantization for deep neural network training

Floating point format and integer format are used mainly for quantized neural network training. The floating point format has a wide representable range. Therefore, the expected loss of information is small when applying floating point format to quantization for DNN training. One reported method [8] proposed quantization method using FP16 with no marked accuracy degradation. Two earlier reports [10, 11] described the use of FP8 for DNN training. Neural network calculation by integer format improves the energy efficiency of hardware for deep learning

models [12]. Therefore, training methods using fixed point precision have also been proposed. Earlier studies [13, 14] have assessed training methods with fixed point format, with high expected energy efficiency of hardware. One of those methods [13] uses DFP16 to prevent accuracy degradation. Although another method [14] uses a DNN training method with DFP8, its accuracy has not been evaluated the accuracy using complex DNN models such as ResNet.

2.2 Dynamic fixed point format

An earlier report [13] described the DNN training method with DFP format. Figure 1 shows the IEEE-754 standard FP32 format and DFP8 format. Although floating point has one exponent for each variable, DFP shares an exponent with multiple variables. The single shared exponent is applied generally to the respective tensors, such as the weight, activation, and gradient. When quantizing the FP32 vector of $X = [x_1, \dots, x_n]^T$ using DFP, the value of shared exponent of DFP e_s is derived from the exponent of the absolute maximum value of X as presented below.

$$e_m = \text{exponent}(\max_{x \in X} |x|) \quad (1)$$

$$e_s = e_m - (BW - 2) \quad (2)$$

where $\text{exponent}()$ represents the exponent of input argument, e_m denotes the exponent of absolute maximum value of X , x is an element of X , and BW is the bit-width of the integer part of DFP, as shown in Fig. 1.

Figure 2 shows a distribution of weight gradient and data representation range before and after quantization with DFP. The data representation range of DFP is represented the difference between e_s and e_m as shown in Fig. 2. Therefore, the bit-width of the integer part represents the data representation range, as given in Eq. (2). The value of quantized element with DFP is represented as shown below.

$$x_{i,q} = \text{round}(x_i / 2^{e_s}) \times 2^{e_s} \quad (3)$$

Single precision floating point (FP32)

8-bit Dynamic fixed point (DFP8)

Fig. 1 IEEE-754 standard single precision floating point format (FP32) and 8-bit dynamic fixed point format (DFP8)

Fig. 2 Distribution of the weight gradient before and after quantization with DFP format in the first convolution layer on ImageNet task with ResNet-50 trained by FP32

where x_i is the i -th element of X and $x_{i,q}$ is the i -th element of X_q . Because the shared exponent of DFP e_s is derived from the absolute maximum value of quantization target vector X , the small value included in the vector before quantization is out of the data representation range of DFP, as shown in Fig. 2. Furthermore, the small value is rounded to zero or the minimum value that can be represented by DFP. Therefore, when the distribution is broad, most small values including the vector before quantization are rounded to zero when quantizing with DFP.

3 Quantized neural networks training with shifted dynamic fixed point

We propose a training method using S-DFP format for quantized DNN training without any marked accuracy degradation. S-DFP can change the data representation range of DFP by adding bias to the exponent. This section first presents the cause of accuracy degradation when using DFP8 for quantized DNN training. Subsequently, we show the proposed quantized DNN training method with 8-bit S-DFP (S-DFP8).

3.1 Analysis of accuracy degradation with dynamic fixed point

Figure 3 shows the validation accuracy on the ImageNet task with ResNet-34 when using FP32 and DFP8. From Fig. 3(b), when using only DFP8 for training, the accuracy is degraded markedly. One reason for accuracy degradation when using DFP is that the step size in the weight update becomes large by quantizing the weight gradient with DFP in comparison with when using FP32. Figure 4 shows the relation between the step size in the weight and the loss function when using DFP8 and FP32. In DNN training

using stochastic gradient descent (SGD), the loss function is reduced by updating the weight with the weight gradient. Additionally, the value of the loss function is optimized by the step size depending on the value of weight gradient. When using DFP for quantization, because the small values included in the vector before quantization are out of the data representation range of DFP, the small values are rounded to zero or the minimum value that can be represented by DFP, as shown in Fig. 2. Therefore, information of the small values which are containing the vector before quantization is lost. In contrast, since loss of information of large values contained in the vector before quantization is small, the weight gradient quantized with DFP includes more information of large values than information of small values. Therefore, when using the weight gradient quantized with DFP, it is difficult for the weight to reach a position that minimizes the loss function because the weights are not updated by the small step size of the weight gradient as shown in Fig. 4a. On the other hands, as shown in Fig. 3b, when switching the data format of DNN model from DFP8 to FP32 in the middle of training, the step size in weight update is also switched from a large value to a small value as shown in Fig. 4b. Consequently, because the weight can move the position at which the loss function is reduced, the accuracy of quantized model becomes equivalent to the accuracy achieved with FP32.

3.2 Shifted dynamic fixed precision format

Figure 5 shows an overview of the proposed S-DFP format to reduce the step size during weight updating. As shown in Fig. 5a, the conventional DFP format is intended mainly to represent large values without overflow by application of large value shared exponents while the range of underflow becomes large. Therefore, because only the information of large values remains in the weight gradient quantized by the conventional DFP, the step size of the weight update becomes large. On the other hand, the proposed S-DFP format can reduce underflowed entries by changing the exponent value of DFP, as shown in Fig. 5b. To shift the data representation range of DFP so that the data representation range of DFP includes small values in the vector before quantization, we modify the definition of shared exponents e_m and e_s to e_{m_mod} and e_{s_mod} respectively, as shown in Eqs. (4) and (5).

$$e_{m_mod} = e_m - bias \quad (4)$$

$$e_{s_mod} = e_{m_mod} - (BW - 2) \quad (5)$$

In these equations, $bias$ is the parameter to shift the data representation range of DFP. The value of quantized element with DFP using modified exponent is represented as presented below.

Fig. 3 Top-1 validation accuracy on the ImageNet task with ResNet-34: **a** whole validation accuracy and **b** accuracy after 60 epochs

$$x_{i,q} = \text{round}(x_i / 2^{e_{s_mod}}) \times 2^{e_{s_mod}} \quad (6)$$

The proposed S-DFP format uses these modified exponents instead of conventional exponents.

e_{s_mod} becomes small by setting a positive value to *bias*, the data representation range of DFP is shifted to the area of small values depending on the value of *bias*. Therefore, information of small values including the vector before quantization remains in the vector after quantization. In contrast, large values including the vector before quantization are overflowed. Consequently, when As shown in Fig. 5, because the modified shared exponent quantizing the weight gradient by S-DFP, the step size during weight update becomes smaller than when quantizing the weight gradient using conventional DFP.

Algorithm 1 DNN training procedure applying the proposed quantization method

- 1: Set the training duration to *num*.
- 2: **for** $i = 0$ to *num* **do**
- 3: Calculate the loss function by the quantized weight and activation with conventional DFP.
- 4: Backpropagate by quantized activation gradient with conventional DFP.
- 5: **if** Validation accuracy is saturated **then**
- 6: Quantize the weight gradient with proposed S-DFP.
- 7: **else**
- 8: Quantize the weight gradient with conventional DFP.
- 9: **end if**
- 10: Update weight with the quantized weight gradient.
- 11: **end for**

Algorithm 1 summarizes the training method for quantized DNN models using the proposed S-DFP.

Fig. 4 Relation between the step size of the weight update and loss function: **a** because of the large step size when using DFP8, it is difficult to reach the minimum loss function; **b** because the step size is small when using FP32, the loss function can reach the minimum

4 Experiment

We evaluated the proposed quantization method on the ImageNet classification task using ResNet-34, ResNet-50, ResNet-101, and ResNet-152. The DNN models were trained by momentum SGD with L2 weight decay. The mini-batch size was 256. The momentum coefficient was 0.9. The coefficient of weight decay was 10^{-4} . The initial learning rate was 0.1. The learning rate was multiplied by 0.1 every 30 epochs. We quantized the weight, activation, weight gradient and activation gradient using conventional DFP8. Furthermore, we quantized the weight gradient with proposed S-DFP8 from 70 epochs at when the validation accuracy is saturated. We used stochastic rounding for the quantized model training proposed in [16]. During the weight update, the FP32 weights were subtracted by the DFP8 weight gradient to prevent accuracy degradation, as demonstrated in [8].

4.1 Parameters setting for S-DFP

Because the step size for the weight update is changed depending on the S-DFP value, the value of *bias* affects the

Fig. 5 Overview of the proposed quantization method: **a** When using conventional DFP, the large value information remains, whereas the small value information is lost. **b** When using proposed S-DFP, the data representable range of DFP is shifted by adding *bias* to the exponent. Consequently, the large value information is lost and the small value information remains

validation accuracy and loss function during training. Figure 6 represents the relation among *bias*, validation accuracy and training loss function on the ImageNet task with ResNet-34 when switching the data format for quantization from conventional DFP to S-DFP in the middle of training for the quantized model. When the *bias* is small, because the step size is large, the training loss function does not become small. As a result, the maximum

Table 1 Comparison of Top-1 maximum validation accuracies for the respective quantization methods

Model	FP32 Acc. (%)	Conventional DFP8 Acc. (%)	S-DFP8 Acc. (%)	<i>bias</i>
ResNet-34	72.2	72.0	72.2	8
ResNet-50	75.5	75.1	75.4	8
ResNet-101	76.8	75.9	76.8	8
ResNet-152	77.6	76.6	77.6	8

Fig. 7 Effects of S-DFP applying timing on the ImageNet classification task with ResNet-34. Suitable timing for applying S-DFP to quantized model training is after saturating validation accuracy

validation accuracy is also small. When the *bias* is large, because the step size becomes small, the number of weight updating times becomes large to reduce the loss function sufficiently. Consequently, unless the number of training times are increased, the maximum validation accuracy can

Fig. 6 Effects of *bias* when applying S-DFP8 from 70 epochs on the ImageNet classification task with ResNet-34. Because the step size in the weight update is changed depending on the *bias*, *bias* affects the training convergence speed and the maximum validation accuracy

Fig. 8 Experimental results on the ImageNet classification task with ResNet-34, ResNet-50, ResNet-101, and ResNet-152: (left side) Top-1 validation accuracy during whole training duration. (right side) Top-1 validation accuracy after 60 epochs. When using ResNet-34, ResNet-101, and ResNet-152, because S-DFP8 was applied to

quantized DNN models from 70 epochs, accuracy with S-DFP8 was improved after 70 epochs. When using ResNet-50, because S-DFP8 was applied to quantized DNN models from 80 epochs, accuracy with S-DFP8 was improved after 80 epochs

Table 2 Comparison of Top-1 validation accuracies with related works on ImageNet task with ResNet-50

Method	Format	Bit-width w, a, gw, ga*	FP32 Acc. (%)	Reduced Precision Acc. (%)
[13]	DFP	16, 16, 16, 16	75.7	75.7
[17]	DFP	8, 8, 8, 8	74.7	68.0
Ours	S-DFP	8, 8, 8, 8	75.5	75.4

*w, a, gw, and ga, respectively, stand for weight, activation, weight gradient, and activation gradient

be expected to be small. In this paper, *bias* for each evaluated model, we chose the *bias* which achieved the best maximum validation accuracy. The *bias* of each evaluated DNN model is shown in Table 1.

Figure 7 shows the validation accuracy when applying S-DFP for weight gradient quantization at the timing of 60 epochs and 70 epochs. When applying S-DFP with a positive *bias* to weight gradient quantization, the step size for weight update becomes small. Therefore, when S-DFP is applied from 60 epochs with low accuracy, the maximum validation accuracy becomes small unless the number of training times are increased. In contrast, when S-DFP is applied from 70 epochs with high accuracy, because the maximum validation accuracy is improved to that of FP32 without increasing the number of training times, the timing

of S-DFP applied for weight gradient quantization is suitable for after the validation accuracy saturation. In this paper, we applied S-DFP for weight gradient quantization from 70 epochs when using ResNet-34, ResNet-101, and ResNet-152. Moreover, we applied S-DFP for weight gradient quantization from 80 epochs when using ResNet-50.

4.2 Experimental results

The experimental results of validation accuracy are shown in Fig. 8 and Table 1. The validation accuracy of the quantized model with conventional DFP8 are degraded as the number of layers of ResNet gets deeper. On the other hand, the validation accuracy of the quantized model with proposed S-DFP was achieved equivalent accuracy with FP32 regardless of the depth in ResNet. Therefore, the proposed S-DFP is expected to significantly improve the accuracy of the quantized model regardless of the DNN model type, such as transformer other than ResNet.

Table 2 presents a comparison of the accuracies with those obtained from earlier studies [13] that used 16-bit DFP and [17] that used 8-bit DFP for quantized DNN training. The proposed method achieved a smaller accuracy degradation than the that of conventional DFP8. In addition, the proposed method achieved the equivalent

accuracy with conventional 16-bit DFP in spite of using 8-bit S-DFP. Therefore, the proposed S-DFP format is more suitable for accelerating DNN training while maintaining accuracy in comparison with conventional DFP format.

5 Conclusion

In this paper, we proposed a quantization method using S-DFP format for deep neural networks training. To prevent accuracy degradation due to quantized DNN training when using conventional DFP format, S-DFP shifts the data representable range of DFP from a large value area to a small value area by adding *bias* to the exponent of DFP. Using the proposed S-DFP format for quantized DNNs training, the quantized model can be trained using 8-bit fixed point precision without significant accuracy degradation on the ImageNet task with ResNet-32, ResNet-50, ResNet-101, and ResNet-152. Since the validation accuracy of the quantized model with proposed S-DFP was achieved equivalent accuracy with FP32 regardless of the depth in ResNet, the proposed S-DFP is expected to significantly improve the accuracy of the quantized model regardless of the DNN model type, such as transformer other than ResNet. In addition, the proposed method achieved the equivalent accuracy with conventional 16-bit DFP in spite of using 8-bit S-DFP. Therefore, the proposed method is more suitable for accelerating DNN training while maintaining accuracy in comparison with conventional DFP format. Our future work is expected to incorporate consideration of the optimization method used for *bias* and applying timing of S-DFP. Furthermore, we expect to apply the proposed method to various models for natural language processing such as transformer [18] and BERT [19].

Acknowledgements The authors would like to thank Mr. Tetsutaro Hashimoto, Mr. Yukihiro Kawabe, Dr. Tsuguchika Tabaru, and Mr. Hisakatsu Yamaguchi for their valuable feedback.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted

use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 25:1097–1105
2. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*
3. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440
4. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587
5. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C Y, Berg A C (2016) SSD: Single Shot Multibox Detector. In *European conference on computer vision*, pp. 21–37
6. Zhou S, Wu Y, Ni Z, Zhou X, Wen H, Zou Y (2016) DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*
7. Rastegari M, Ordonez V, Redmon J, Farhadi A (2016) XNOR-Net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pp. 525–542
8. Micikevicius P, Narang S, Alben J, Diamos G, Elsen E, Garcia D, Ginsburg B, Houston M, Kuchaiev O, Venkatesh G, Wu, H (2017) Mixed precision training. *arXiv preprint arXiv:1710.03740*
9. Chen C Y, Choi J, Gopalakrishnan K, Srinivasan V, Venkataramani S (2018) Exploiting approximate computing for deep learning acceleration. In: *2018 Design, automation and test in Europe conference and exhibition*, pp. 821–826
10. Wang N, Choi J, Brand D, Chen C Y, Gopalakrishnan K (2018) Training deep neural networks with 8-bit floating point numbers. *arXiv preprint arXiv:1812.08011*
11. Sun X, Choi J, Chen CY, Wang N, Venkataramani S, Srinivasan V, Cui X, Zhang W, Gopalakrishnan K (2019) Hybrid 8-bit floating point (HFP8) training and inference for deep neural networks. *Adv Neural Inf Process Syst* 32:4900–4909
12. Horowitz M (2014) Computing's energy problem (and what we can do about it). In: *IEEE International solid-state circuits conference digest of technical papers*, pp. 10–14
13. Das D, Mellempudi N, Mudigere D, Kalamkar D, Avancha S, Banerjee K, Sridharan S, Vaidyanathan K, Kaul B, Georganas E, Heinecke A, Dubey P, Corbal J, Shustrov N, Dubtsov R, Fomenko E, Pirogov V. (2018) Mixed precision training of convolutional neural networks using integer operations. *arXiv preprint arXiv:1802.00930*
14. Wu S, Li G, Chen F, Shi L (2018) Training and inference with integers in deep neural networks. *arXiv preprint arXiv:1802.04680*
15. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778
16. Gupta S, Agrawal A, Gopalakrishnan K, Narayanan P (2015) Deep learning with limited numerical precision. In: *International conference on machine learning*, pp. 1737–1746

17. Yang Y, Deng L, Wu S, Yan T, Xie Y, Li G (2020) Training high-performance and large-scale deep neural networks with full 8-bit integers. *Neural Netw* 125:70–82
18. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L, Polosukhin I (2017) Attention is all you need. *arXiv preprint* [arXiv:1706.03762](https://arxiv.org/abs/1706.03762)
19. Devlin J, Chang M W, Lee K, Toutanova K (2018) BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.