Effect of the Period of the Fourier Series Approximation for Binarized Neural Network

SeonYong Lee
Department of Electrical and Computer Engineering
INMC, Seoul National University
Seoul, Korea
sonyongyi@snu.ac.kr

Hee-Youl Kwak
Department of Electrical and Computer Engineering
INMC, Seoul National University
Seoul, Korea
ghy1228@gmail.com

Jong-Seon No
Department of Electrical and Computer Engineering
INMC, Seoul National University
Seoul, Korea
jsno@snu.ac.kr

***Abstract*—The construction of low complexity models for the neural networks is an important issue in practical, real-world scenarios. One of the most famous construction methods for a simple neural network model is to represent weights and activations by the 1-bit quantization, called binarized neural networks (BNNs). However, it is still under research on how to represent the gradient in the backpropagation of BNNs because the activation function is the sign function whose gradients are zero almost everywhere. One way to address this problem is to approximate the gradient of the sign function by the Fourier series representation. In this paper, we analyze the effect of the period and the number of terms of the Fourier series representation on the network accuracy. Since the period has a direct relationship with the degree of the approximation for the sign function and the oscillation behavior of the gradient function, the choice of the period significantly affects the accuracy of the BNN model. The experiments on the CIFAR-10 dataset demonstrate that a proper choice of the period can outperform the conventional BNNs with straight through estimator.**

***Index Terms*—Binary neural network (BNN), Fourier series representation (FSR), gradient approximation, period, Straight through estimator (STE)**

I. INTRODUCTION

Recently, machine learning techniques have been applied to a wide range of engineering fields and make a remarkable achievement in such fields. One of the most famous examples is superhuman performance in vision recognition with convolutional neural networks (CNNs) [1]. However, CNNs are not suitable for low-complexity applications such as mobile devices because CNNs need lots of memory and computation (energy) requirements. Since mobile devices (e.g., smartphone, laptop) have several limitations of a small battery, insufficient memory, and low GPU performance, many researchers have proposed several methods such as AlexNet [1], VGGNet [2], and MobileNetV2 [3] to implement cost-efficient architectures, and 1×1 convolution [4] to reduce the computational complexity of the arithmetic operation.

Another promising way to reduce the hardware cost dramatically is binarized neural networks (BNNs) [5], where weights and activations are represented by the 1-bit quantization. BNNs have a competitive advantage over other methods in that they can be constructed from a given well-designed neural networks without changing the key idea of underlying architectures. However, the simply converted BNN using the binary sign function is not feasible to train because the gradient of the sign function is zero almost everywhere, which hinders backpropagation in training. Therefore, BNNs need to employ special techniques to facilitate backpropagation such as straight through estimator (STE) [6] and the approximation by the Fourier series representation (FSR) [7].

Using the FSR, we can transform a periodic function with period T into the summation of n triangular functions. For BNNs, the sign function is approximated by the summation of n differentiable sinusoidal functions, and then their gradients are used for the backpropagation [7]. However, in [7], they did not consider the problem of selecting a proper period T and number of terms n.

In this paper, we investigate the effect of the period and number of terms on the approximation of the sign function by the FSR method and the resulting accuracy of the BNN model. As the period T decreases, the approximation becomes more accurate around the zero-point, but a small period T induces a problem of the fast oscillating in the gradient domain. In addition, as the number of terms n increases, the FSR represents the original function more accurately, but it grows the computational complexity linearly. In other words, there is a compromise on the period and number of terms to maximize the accuracy of the model with a reasonable complexity. We evaluate the model accuracy by simply replacing the STE method with the FSR method under the given BNN architecture [5]. Evaluation using the CIFAR-10 dataset shows that the BNN with the FSR can outperform the BNN with the STE if we choose proper values of the period and the number of terms in the FSR.

The remainder of the paper is organized as follows. Section II introduces preliminaries for the BNNs and the STE method. Section III describes the FSR method and its training algorithm. Section IV shows the performance evaluation results and discussions on the period and number of terms in the FSR. Finally, conclusion is given in Section V.

II. Preliminaries

A. Binarized Neural Networks

The CNNs are included in a class of full precision neural networks, where the weight matrix W and activation matrix A are represented with 32-bit or 64-bit. Instead, the quantized neural networks (QNNs) are those that represent the weight and activation with lower precision. As the extreme case of QNNs, BNNs use 1-bit to represent the binarized weight matrix W^b and binarized activation matrix A^b using the sign function, $\text{Sign}(x)$, as

$$\text{Sign}(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

B. Straight Through Estimator (STE)

Since the gradient of the sign function is zero almost everywhere, training BNNs with the traditional backpropagation method is nearly impossible. Thus, the STE method [6] is proposed to train the conventional BNNs. The key idea of the STE method is to alter the actual gradient to the coarse gradient, which enables to train BNNs. The STE method represents the coarse gradient of the sign function as

$$g_a \approx \text{Clip}\left(\frac{\partial C}{\partial a}, -1, 1\right),$$

where g_a is real gradient, C is the cost function, $a \in A$ is the layer activation, and the clip function, $\text{Clip}(x, -1, 1)$, is defined as

$$\text{Clip}(x, -1, 1) = \begin{cases} -1, & \text{if } x < -1, \\ x, & \text{if } -1 \leq x < 1, \\ +1, & \text{otherwise.} \end{cases} \tag{1}$$

Then, the gradient of (1), $\text{Clip}'(x, -1, 1)$ is given as

$$\text{Clip}'(x, -1, 1) = \begin{cases} 1, & \text{if } -1 \leq x < 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Although the idea is simple, this method works pretty well. But still, the model accuracy of such BNNs is not accurate enough to satisfy practical demands. For those reasons, researchers have tried to find a way to increase the model accuracy by replacing the STE into BNN+ [8], DSQ [9], and FDA-BNN [7].

III. Fourier Series BNN

The FSR is a very useful mathematical tool to approximate an original function. A function with period T can be represented by the summation of n triangular functions. Let $\text{FSR}(x; n, T)$ denote the FSR of the sign function, $\text{Sign}(x)$, with a period T and a finite number of terms n, which is given as

$$\text{FSR}(x; n, T) = \frac{c_0}{2} + \sum_{i=1}^{n} \left[c_i \cos \frac{2\pi i x}{T} + s_i \sin \frac{2\pi i x}{T} \right],$$

Algorithm 1 Training process of Fourier series BNN

Require: A layer weight matrix W, binarized weight matrix W^b, layer input a_k, binarized layer input a_k^b batchnormalization parameter θ_k

Forward propagation:
for k =1 to L **do**
  $W_k^b \leftarrow \text{Binarize}(W_k)$
  $s_k \leftarrow a_{k-1}^b W_k^b$
  $a_k \leftarrow \text{BatchNorm}(s_k, \theta_k)$
  if $k < L$ **then**
    $a_k^b \leftarrow \text{Binarize}(a_k)$
  end if
end for
Backward propagation:
Compute $g_{a_L} = \frac{\partial C}{\partial a_L}$ knowing a_L
for $k = L$ to 1 **do**
  if $k < L$ **then**
    $g_{a_k} \leftarrow g_{a_k^b} \text{FSR}'(x; n, T)(a_k)$ (3)
  end if
  $(g_{s_k}, g_{\theta_k}) \leftarrow \text{BackBatchNorm}(g_{a_k}, s_k, \theta_k)$
  $g_{a_{k-1}^b} \leftarrow g_{s_k} W_k^b$
  $g_{W_k^b} \leftarrow g_{s_k}^\top a_{k-1}^b$
end for
{Accumulating the parameters gradients:}
for k = 1 to L **do**
  $\theta_k^{t+1} \leftarrow \text{Update}(\theta_k, \eta, g_{\theta_k})$
  $W_k^{t+1} \leftarrow \text{Clip}(\text{Update}(W_k, \gamma_k \eta, g_{W_k^b}), -1, 1)$
  $\eta^{t+1} \leftarrow \lambda_\eta$
end for

where c_i and s_i are the ith Fourier series coefficients. For the sign function, the Fourier series coefficients are given as

$$c_i = 0 \text{ for all } i \text{ and } s_i = \begin{cases} \frac{4}{i\pi}, & \text{if } i \text{ is odd,} \\ 0, & \text{otherwise.} \end{cases}$$

Now we can express the FSR of the sign function as the following reduced form

$$\text{FSR}(x; n, T) = \frac{4}{\pi} \sum_{i=0}^{n} \frac{\sin{(2i+1)\frac{2\pi}{T}x}}{2i+1}, \quad |x| < T.$$

Thus, the gradient of the $\text{FSR}(x; n, T)$ is given as

$$\text{FSR}'(x; n, T) = \frac{8}{T} \sum_{i=0}^{n} \cos{(2i+1)} \frac{2\pi}{T} x, \quad |x| < T. \tag{4}$$

The above approximation of the gradient can replace the STE of BNNs. Algorithm 1 represents the Fourier series BNN using the derived gradient of $\text{FSR}'(x; n, T)$ in (4). Note that the main part of the algorithm follows the conventional BNN in [5], and we just replace the STE method with $\text{FSR}'(x; n, t)$ as shown in (3). During the backpropagation, the backpropagated gradient from the $(i+1)$th layer is multiplied with the gradient of the ith layer. The conventional BNN using the STE method calculates the backpropagated gradient by (2)

TABLE I
MODEL ACCURACIES WITH $n = 20$ AT 100 EPOCHS FOR THE CIFAR-10 DATASET

Model	Method	Period T	Accuracy(%)
VGG-Small	STE		90.9
	FSR	10	80.1
	FSR	50	90.4
	FSR	100	90.8
	FSR	150	**91.1**
	FSR	200	91.0
	FSR	1000	87.4
Resnet-18	STE		86.2
	FSR	10	53.0
	FSR	50	86.0
	FSR	100	**87.0**
	FSR	150	85.2
	FSR	200	84.8
	FSR	1000	74.3

TABLE II
MODEL ACCURACIES WITH $T = 150$ FOR VGG-SMALL AND $T = 100$ FOR RESNET-18 AT 100 EPOCHS FOR THE CIFAR-10 DATASET

Model	Method	Number of terms n	Accuracy(%)
VGG-Small	STE		90.9
	FSR	5	88.3
	FSR	10	90.0
	FSR	20	**91.0**
	FSR	50	90.8
Resnet-18	STE		86.2
	FSR	5	80.6
	FSR	10	84.6
	FSR	20	**87.0**
	FSR	50	85.3

while our method calculates the backpropagated gradient from the $(i+1)$th layer by (3).

IV. Results and Discussion

A. Evaluation Result

We evaluate the proposed FSR method in the BNN on the CIFAR-10 dataset. The CIFAR-10 dataset is one of the famous image classification benchmark datasets. It consists of 50,000 training data and 10,000 test data. Each image consists of 32×32 color pixels. Tested BNNs architectures are based on the VGG-small network and the ResNet-18 network. We train the BNN model with the FSR by an optimized learning rate setting in [5]. The stochastic gradient descent optimizer is used with a momentum of 0.9 and a weight decay of 0.999. Test environment builds on cuda toolkit==11.5 and pytorch. And the GPU specification is RTX 2080 Ti.

The results are summarized in Table I and Table II, which show that the period T and the number of terms n affect the model accuracy. From the result in Table I, we can see the model accuracy is improved by increasing T until a certain point but is degraded after that. In addition, Table II shows that the model accuracy is also improved as n grows but it does not after $n > 20$.

B. Effect of the Period and Number of Terms in the FSR

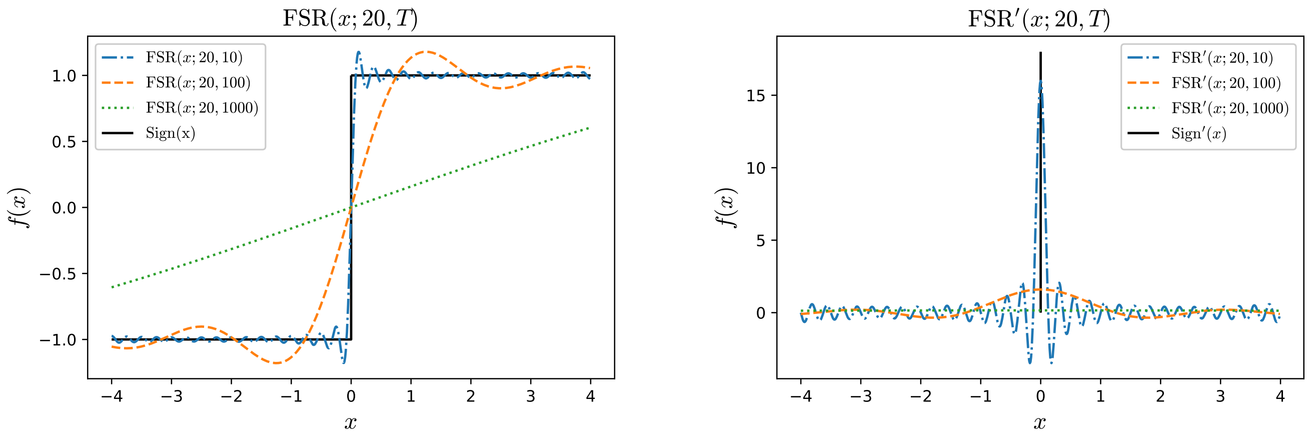
Fig. 1 and Fig. 2 show the variation of $\mathrm{FSR}(x; n, T)$ and $\mathrm{FSR}'(x; n, T)$ as a function of the period T and the number of terms n, respectively. The results apparently show that the better approximation for the sign function, $\mathrm{Sign}(x)$, can be achieved by the smaller period T and the larger number of terms n. Therefore, one can assume that a smaller period and a larger number of terms lead to better performance. However, the evaluation results in Section IV-A show that the smaller period does not guarantee better performance, and neither larger number of terms do too. We investigate the results and draw the following discussions.

First, the selection of a proper period T is very important to improve the accuracy of the model using the FSR method. Fig. 1(b) shows that small periods such as $T = 10$ induce a large variation of the gradient near the zero-point $x = 0$, which in turn causes the noisy gradient problem [11]. The noisy gradient problem is the The noisy gradient problem is known to interfere with the training of the network and occur more often as the variation of the gradient increases [11]. Thus, small periods are not preferable in terms of the gradient. On the contrary, large periods such as $T = 1{,}000$ in Fig. 1(a) cannot achieve the accurate approximation for the sign function. In other words, there is a trade-off between the accuracy of the approximation and the noisy gradient problem, which can be controlled by period T. Thus, there is a compromise on the period and Table II shows that the proper period T is 150 that maximizes the model accuracy.

Second, it is unnecessary to increase the number of terms n more than a threshold value because there is no additional accuracy gain as n grows over the threshold while the computation complexity grows linearly by n. Table 2 shows that the model accuracy has not improved over $n > 20$, which means a finite value of n is sufficient.

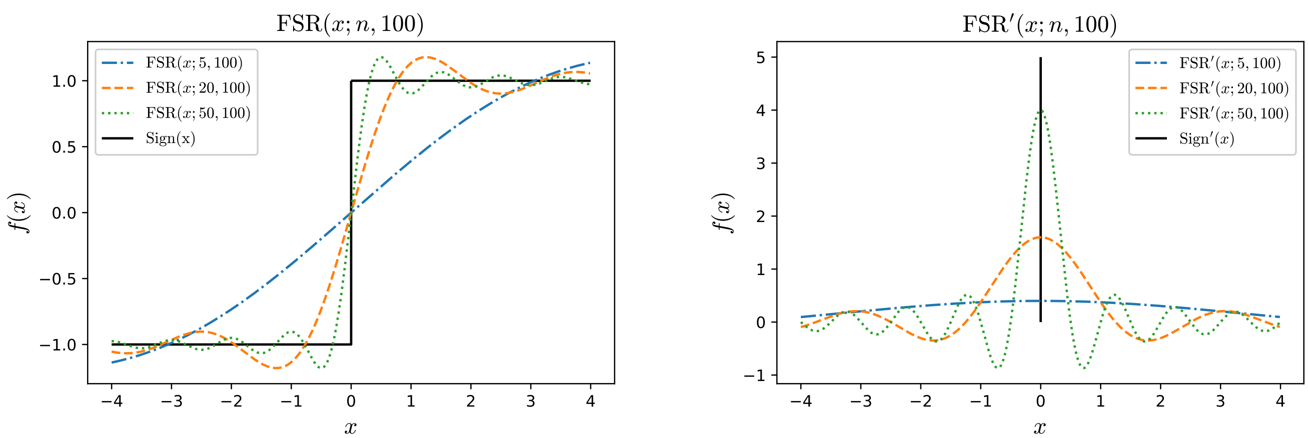
V. Conclusion

In this paper, we investigated the effect of the period and the number of terms in the FSR method for BNN. We replaced the STE method with the FSR method and conducted the evaluations with the various period and the various number of terms in the FSR method. We show that the proper period of the FSR method improves the model accuracy and the certain number of the terms improves the model accuracy. Since the proper period deals with a tradeoff between oscillation of the approximated gradient and precision of the approximated gradient of the sign function around the zero point, the proper period guarantees the better performance. The number of terms has a direct relationship with computational complexity. However, the number of terms does not always improve the model accuracy. Therefore, the proper number of terms is enough to generate the best performance of the BNN model. The experimental results proved that the proper period and the number of terms in the FSR of BNNs outperform the STE method in the model VGG-small network and the ResNet-18 network for the CIFAR-10 dataset.



(a) (b)

Fig. 1. Comparison of Sign(x) and FSR$(x; 20, T)$ with various values of T in terms of (a) original functions and (b) their gradients.



(a) (b)

Fig. 2. Comparison of Sign(x) and FSR$(x; n, 100)$ with various values of n in terms of (a) original functions and (b) their gradients.

ACKNOWLEDGMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-00400, Development of Highly Efficient PQC Security and Performance Verification for Constrained Devices)

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097-1105.

[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learn. Representations*, 2015.

[3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: inverted residuals and linear bottlenecks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510-4520.

[4] M. Lin, Q. Chen, and S. Yan, “Network in network,” 2014. [Online]. Available: http://arxiv.org/abs/1312.4400

[5] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, “Binarized neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4107-4115.

[6] Y. Bengio, N. Léonard, and A. Courville, “Estimating or propagating gradients through stochastic neurons for conditional computation,” 2013. [Online]. Available: http://arxiv.org/abs/1308.3432

[7] Y. Xu, K. Han, C. Xu, Y. Tang, C. Xu, and Y. Wang, “Learning frequency domain approximation for binary neural networks,” 2021. [Online]. Available: http://arxiv.org/abs/2103.00841

[8] S. Darabi, M. Belbahri, M. Courbariaux, and V. P. Nia, “Regularized binary network training,” 2020. [Online]. Available: https://arxiv.org/abs/1812.11800

[9] R. Gong, X. Liu, S. Jiang, T. Li, P. Hu, J. Lin, F. Yu, and J. Yan, “Differentiable soft quantization: Bridging full-precision and low-bit neural networks,” in *Proc. Int. Conf. Comput. Vision*, 2019, pp. 4852–4861.

[10] M. Rastegar, V. Ordonez, J. Redmon, and A. Farhadi, “Xnor-net: Imagenet classification using binary convolutional neural networks,” in *Proc. Eur. Conf. Comput. vision*, 2016, pp. 525-542.

[11] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, “On the variance of the adaptive learning rate and beyond,” 2019. [Online]. Available: http://arxiv.org/abs/1908.03265