A Zero-Gating Processing Element Design for Low-Power Deep Convolutional Neural Networks

Lin Ye, Jinghao Ye, Masao Yanagisawa, and Youhua Shi
Graduate School of Fundamental Science and Engineering,
Waseda University
Okubo 3-4-1, Shinjuku-ku, Tokyo, 169-8555 Japan
Email:{lin.ye, youhua.shi}@islab.cs.waseda.ac.jp

***Abstract*—Convolution neural networks (CNNs) have shown great success in many areas such as object detection and pattern recognition. However, the high computational complexity of state-of-the-art deep CNNs makes them extreme difficult to be run on resource-constrained mobile and wearable devices. To address this design challenge, in this paper we first analyzed the filters' weights of pre-trained models from four state-of-the-art CNNs. We found that in all the CNNs that we analyzed, from about 20% (AlexNet) to 43% (VGG-19) of the weights are zeros, which lead to redundant large amounts of computation. Then, based on this observation, a zero-gating processing element (PE) design was proposed for low-power deep CNNs, in which the vast number of zeros in both activation maps and filter weights are explored to eliminate redundant computation for power reduction. We implemented our proposal with VGG-16 using ImageNet dataset. Experiments were conducted for evaluations of area and total power consumption. Compared with the baseline PE design without zero-gating, overall the proposed zero-gating PE can achieve 37% power saving while the corresponding area overhead is less than 8%.**

Keywords—CNNs, zero-gating, activation map, filter, power consumption.

I. INTRODUCTION

Convolutional Neural Networks (CNNs) have made great contributions to many areas such as image segmentation and object detection [1]. Current CNN architectures [2]-[4] achieve better precision in image recognition than human beings. However, CNNs often require large amounts of parameters to compute even for one single inference, causing the amount of whole computation to be extremely large.

In literature, several works on reusing parameters through tiling technology [5]-[8] have been proposed to limit the memory access for power reduction. In addition, it has been observed that about 80% of the input data in activation map is zero in the last few layers in CNNs due to data quantification [9]-[10] and then based on this observation processing element (PE) designs were proposed in [9]-[10]. Existing works only focus on the zeros in activations, while the parameters in filters' weights have not ever been considered. However, those filters' weights also contain a large number of zeros, which might be utilized for further elimination of redundant computation for power reduction.

For this purpose, a zero-gating processing element design for low-power deep CNN implementations is proposed in this paper, in which the great number of zeros in both activation maps and filter weights were explored to eliminate redundant computation for power reduction. In our results, the overall power consumption of PE arrays with our proposal can be reduced by 37% and 14% when compared to the baseline PE design and the existing only-activation-gated design, respectively.

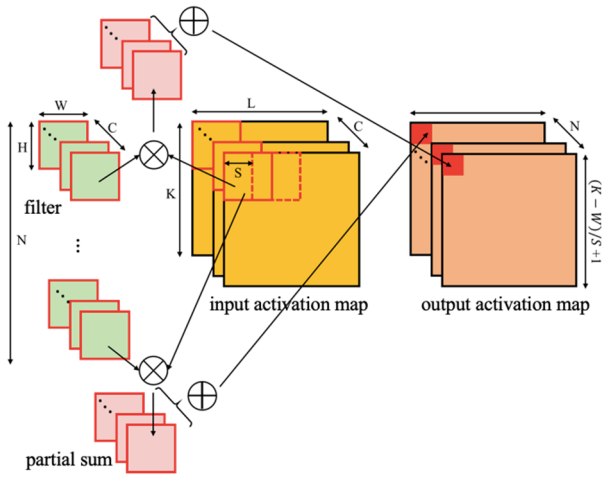


Fig. 1. Computation of 3D-Convolution in CNNs

The rest of the paper is organized as follows. Section 2 gives a brief introduction of CNN and the existing PE designs. Section 3 presents the analysis of zero occurrence in filters' weights of pre-trained models from four state-of-the-art CNNs and then Section 4 describes the proposed zero-gating PE design. Evaluation results of the VGG-16 implementation are presented in Section 5. Finally, Section 6 concludes this work.

II. BACKGROUND AND RELATED WORKS

A. Convolutional Neural Networks (CNNs)

A typical CNN is made of convolutional layers, non-linear layers, pooling layers, and full-connected layers, and each layer has 3D-dimentional activation maps ($L \times K \times C$) and filters ($W \times H \times C \times N$) as shown in Fig.1. With different stride (s), convolutions between activations and weights form the output to be$((L-W)/S+1) \times ((K-W)/S+1)$. The corresponding calculation can be expressed as:

$$O[n][x][y] = \sum_{c=0}^{C-1} \sum_{w=0}^{W-1} \sum_{h=0}^{H-1} F[n][c][w][h] \times I[c][x+sw][y+sh] \qquad (1)$$

where, $0 \le n \le N-1$, $0 \le x \le ((L-W))/S$, $0 \le y \le (K-W)/S$. The computation of 3D-convolutions are performed in the CNN layers and it has been shown that the convolutional layers occupy more than 90% of the total computation [6].

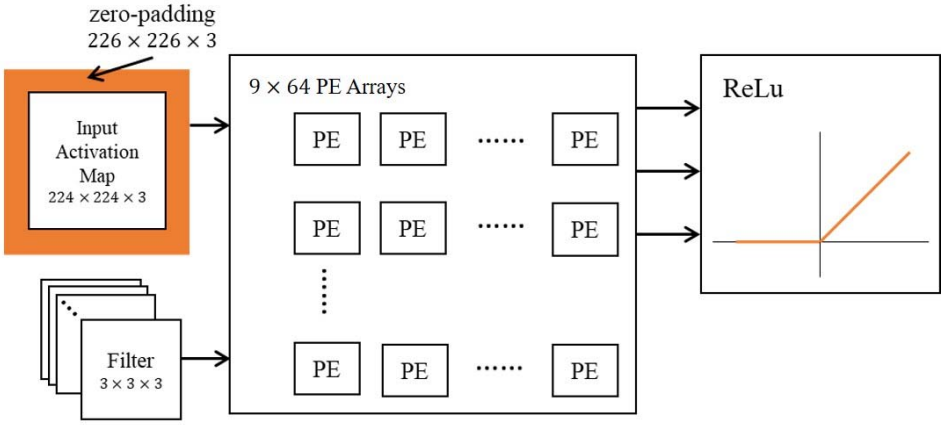


Fig. 2. Computation of the first convolutional layer

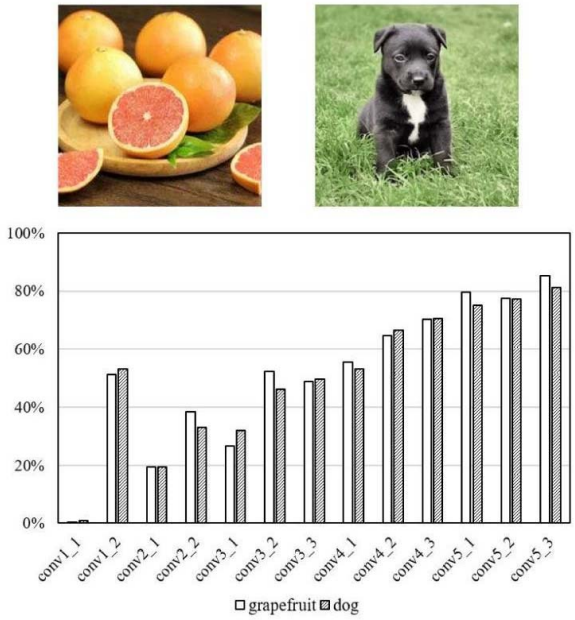


Fig. 3. Percentages of zeros in activation maps of VGG-16.

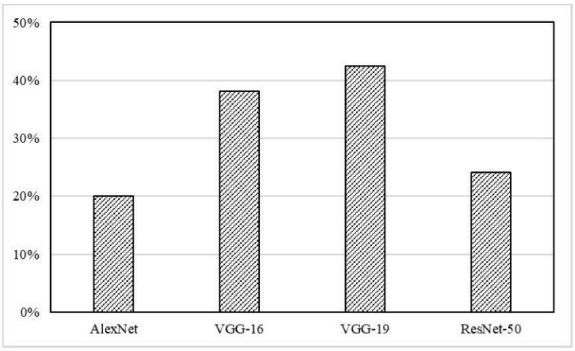


Fig. 4. Percentages of zeros in filters' weights.

B. Existing PE Designs for Computation Reduction

Recently, two PE designs have been proposed by utilizing the zeros in activations for computation reduction. In [9], a PE with zero buffer is proposed. Once a zero is detected, the zero buffer generates a signal to prevent registers from reading the activations and weights. In [10], a PE design with zero skipping is proposed. If a zero in an activation is detected, the input passes the current PE directly to the next PE, and in the next PE, the same zero checking operation should be performed again. Both above PE designs can eliminate redundant computation due to the zeros in activations. However, the zeros in filters' weights are ignored.

In addition, bit width reduction in PEs is another way for computation reduction [11]-[12], however, for a pre-trained model, it might lead to significant classification accuracy loss compared to the exact implementation.

III. ZERO ANALYSIS

It was shown in [9] and [10] that about 80% of the activations are zeros in the last few layers due to ReLu and zero-padding operations as shown in Fig. 2. Zero-padding is used to pad the input data with zeros around the activation map before computation. And ReLu is a non-linear function following the convolutional layers so that all activations can be thresholded at zero after the convolution [13]. In our work, we import activation maps using Keras [14]. Two example images (size: 224×224) are used to find the detailed percentages of zeros in each layer of VGG-16. The example images and results are shown in Fig. 3. As we can see from Fig. 3, the results of two images have corresponding trend. Obviously, the results are consistent with the observation shown in [9]. There are little zero in the first layer. However, nearly 50% of the activations are zero in the second layer (conv1_2). Then the percentages drop dramatically in conv2_1. The percentages increase gradually from conv2_2 to conv4_3, from about 20% to over 80%. The percentages of last three layers are still around 80%. By using these zero inputs in activation maps, the following MAC operations in PEs could be eliminated so as to reduce the power consumption.

As mentioned above, the existing works only considered the zeros in activation maps for redundant computation elimination, while the other input of each layer, the filters' weights, has not been taken into consideration. Therefore a zero analysis of filters' weights was conducted in our work.

We imported the filters' weights of pre-trained models of four state-of-the-art CNNs (i.e. AlexNet, VGG-16, VGG-19 and ResNet-50) at the data precision of 8-bit from Keras [14]. The percentages of zeros in the weights of each CNN's filters are shown in Fig. 4. From the figure, it can be observed that AlexNet has 20% of zero in weights on average. VGG-16 and

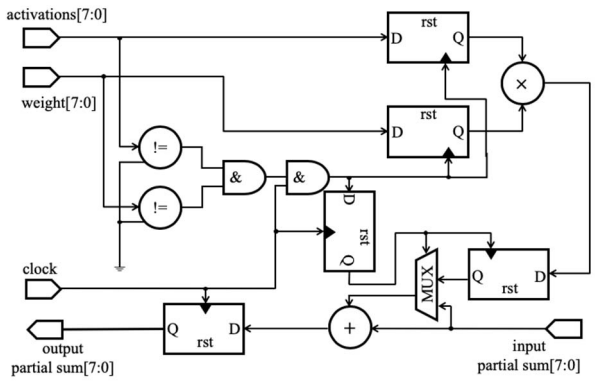
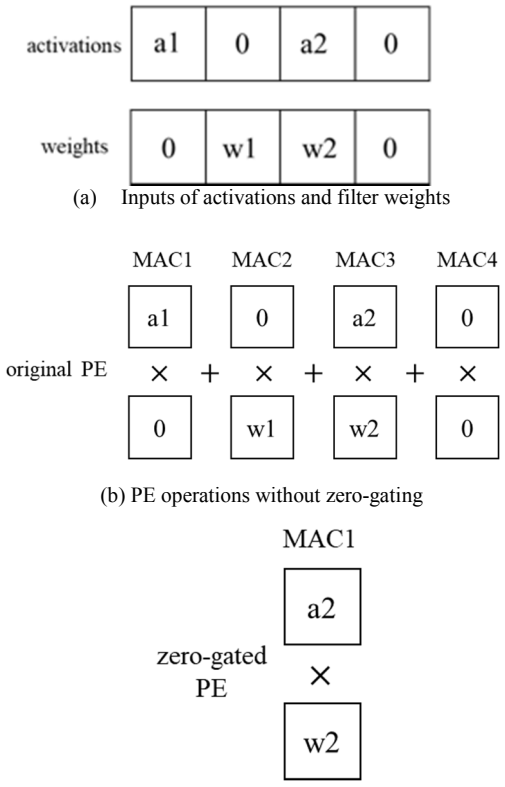


Fig. 5. The proposed zero-gating PE design



(a) Inputs of activations and filter weights

(b) PE operations without zero-gating

(c) PE operations with the proposed zero-gating

Fig. 6. An example of PE operations wi./wo. zero-gating

VGG-19 has 38% and 43% of zero on average, respectively. ResNet-50 has 24% of zero for its extremely irregular architectures. The analysis results clearly show that considerable amounts of zeros exist in filters' weights, which can be explored to be combined with the zeros in activation maps for further redundant computation reduction.

IV. Proposed Zero-gating PE design

In our work, the CNN architecture as shown in Fig.2 was used for a general CNN implementation, in which an array of (9×64) PEs is included. The proposed zero-gating PE design is shown in Fig. 5. To explore the possible elimination of redundant computation, in our work the zeros in both the activation maps and the filters' weights are taken into consideration for the purpose of low power consumption. The zero-gating logic is implemented through clock-gating by exploiting zeros in the inputs. The module (!=) means that each bit of both activations and weights are compared with zero. This module is made up with several connected OR gates. We set two control signals, one for weights and the other for activations in zero checking. If a zero input appears in either activations or weights, a clock gating signal is enabled to disable the update of the registers of activations and weights from reading new data and then prevent the followed multiplier from switching for power saving. One multiplexer is set at the output of the adder to choose the corresponding result from the adder or the input partial sum when the clock gating signal is enabled.

TABLE 1. Design Parameters

Technology	SMIC 40nm
Supply Voltage	1.1 V
PE Array Area	$1.96 \times 10^5\ \mu m^2$
Number of PEs	576
Data precision	8-bit fixed
Clock frequency	510MHz

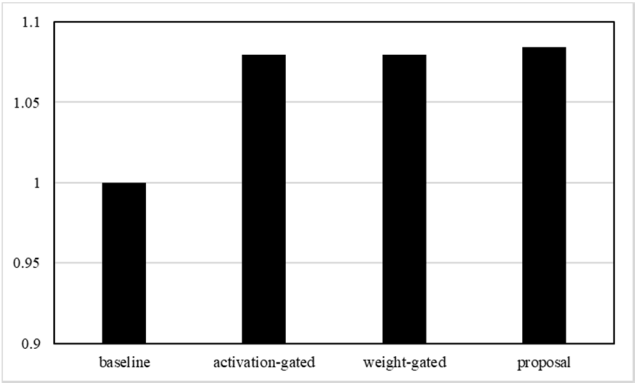
Fig. 6 shows an example of how the required computation can be eliminated through the proposed zero-gating method. Original PE computes all pairs of activation and weight. Fig. 6 (a) shows the input sequences of activations and weights. Fig. 6 (b) and (c) shows the required computation with and without the proposed zero-gating. From the figure, it can be clearly observed that there is only one MAC operation in the proposed zero-gating PE while four and two MAC operations are required in the baseline PEs without zero-gating and with zero-gating only in activations, respectively. The proposed zero-gating PE design works in full power only when neither activation nor weight is zero, which is promising for low power deep CNNs with no loss of precision.

Moreover, it should be mentioned that the proposed PE design can also be applied to data reuse for resource reduction. For example, if filters' weights are reused, then same activations are read to different PE groups. When a round of computation finishes, only new activations are needed till the whole computation of the present filter ends. Therefore, for one filter, only one zero-check operation is required, which can lead to hardware overhead reduction.

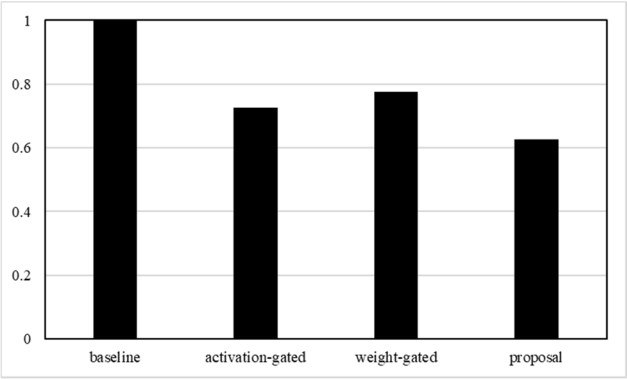
V. Evaluation Results

In our work, VGG-16 [3], one state-of-the-art CNN with 13 CONV layers, is used as a representative example for CNN implementation. The pre-trained VGG-16 models from Keras was used to evaluate the performance of the proposed zero-gating PE design. Moreover, the input pictures are randomly chosen from ImageNet [15] for power evaluation. The design parameters are listed in TABLE I.

The comparison results on area and power consumption of the baseline PE design, the PE design with only activation gating such as that in [9] and [10] and the proposed zero-gating PE are illustrated in Fig. 7, where the normal PE without zero gating is normalized as the baseline in our evaluation. For better illustration, the PE design with only filters' weight gating is also provided. It can be observed from Fig. 7(a) that the area overhead of our proposed zero-gating PE is increased only by 8% and 1% when compared to the baseline design and the PE with only activation gating [9]-[10], respectively. On the other hand, as for the power saving shown in Fig. 7(b), it can be observed that the proposed zero-gating PE achieves 37% and 14% power saving, respectively.



(a) Area



(b) Power consumption

Fig. 7. Comparison results of area and power consumption for different PE designs

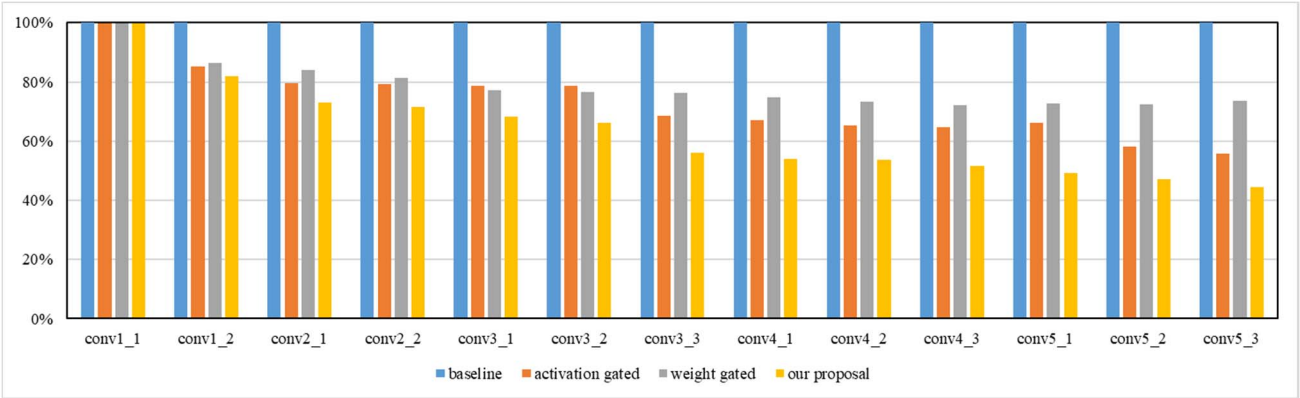


Fig. 8. Power consumption in each layer of VGG-16

Fig. 8 shows the detailed power savings in each layer of VGG-16 for the four implementations. From the figure, it can be observed that, except the first layer (i.e. conv1_1), great power consumption can be achieved through redundant computation elimination. The results show that the activation-gated only designs [9]-[10] can achieve more power savings than the weight-gated only design, and the proposed zero-gating design achieved the best result. Moreover, in the proposed zero-gating design, except the first and second layers (i.e. conv1_1 and conv1_2), more than 40% power saving can be achieved, and for the last three layers up to 53% of the power consumption can be reduced when compared to that in the baseline design.

VI. Conclusion

In this paper, a zero-gating processing element design for low-power deep CNN implementations is proposed, in which the great number of zeros in both activation maps and filter weights are explored to eliminate redundant computation for power reduction. Compared to the baseline PE design without zero-gating, the proposed zero-gating PE achieves 37% power saving with additional 8% area overhead. As mentioned before, this work could be combined with data compression which can reduce the memory access. Both of them are intended to reduce the total power consumption. And this will be our future work.

References

[1] Y. Lecun, Y. Bengio, and G.Hinton, "Deep Learning," Nature 512, 2015, pp.436-444.

[2] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," NIPS, 2012, pp. 1-9.

[3] K. Simonyan, and A. Zisserman, "Very Deep Convolutional Neural Networks for Large Scale Image Recognition," arXiv:1409.1556, pp. 1-14, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," CVPR, 2016, pp.1-9.

[5] Y. Ma, Y. Cao, S. Vridhula, and J. Seo, "Optimizing the Convolutional Operation to Accelerate Deep Neural Networks on FPGA," IEEE Trans. VLSI Syst., iss. 99, pp 1-14, April 2018.

[6] J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, T. Tang, N. Xu, S. Song, Y. Wang, and H. Yang, " Going Deeper with Embedded FPGA Platform for Convolutional Neural Networks," FPGA, 2016, pp. 26-35.

[7] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, J. Cong, "Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks," FPGA, 2015, pp. 161-170.

[8] Y. Ma, Y. Cao, S.Vrudhula, and J.Seo, "An Automatic RTL Compiler for High-Throughput FPGA Implementation of Deverse Deep Convolutional Neural Networks," FPGA, 2017, pp. 1-8.

[9] Y. Chen, T. Krishna, J. Emer, and V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," IEEE J. Solid-State Circuits, vol. 52, no. 1, pp. 127–138, January 2017.

[10] S. Yin, P. Ouyang, S. Tang, F. Tu, X. Li, S. Zheng, T. Lu, J. Gu, L. Liu, and S. Wei, "A High Energy Efficient Reconfigurable Hybrid Neural Networks Processor for Deep Learning Applications," IEEE J. Solid-State Circuits, vol. 53, no. 4, pp. 968-982, December 2017.

[11] K. Hedge, J. Yu, R. Agrawal, M. Yan, M. Pellauer, and C. W. Fletcher, "UCNN: Exploiting Computational Reuse in Deep Neural Networks via Weight Repetition," ISCA, 2018, pp. 1-14.

[12] N. Jouppi et. al, "In-Datacentre Performance Anlysis of a Tensor Processing Unit," ISCA, 2017, pp. 1-17.

[13] V. Nair, and G. Hinton, "Rectified Linear units Improve Restricted Boltzmann Machine," ICML, 2010, pp. 1-8.

[14] F. Chollet, "Keras: The Python Deep Learning library," 2015. [Online]. Available: https://keras.io/.

[15] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "ImageNet: A Large-Scale Hierarchical Image Database," CVPR, 2009, pp. 248-255.