IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 September 2022, accepted 22 October 2022, date of publication 3 November 2022, date of current version 9 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3219208

RESEARCH ARTICLE

Low Power Neural Network by Reducing SRAM Operating Voltage

KEISUKE KOZU[1], YUYA TANABE[1], MASATO KITAKAMI[2], (Member, IEEE), AND KAZUTERU NAMBA[2]

[1]Graduate School of Science and Engineering, Chiba University, Chiba 263-8522, Japan
[2]Graduate School of Engineering, Chiba University, Chiba 263-8522, Japan

Corresponding author: Kazuteru Namba (namba@ieee.org)

This work was supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant 20K11728 and Grant 21H01382.

ABSTRACT With advancements in machine learning technology, networks are becoming increasingly complex, and the extent of the computation involved is increasing. Consequently, the computation time and power consumption of the learning process are increased. The error tolerance of neural networks has attracted attention as an approach to solving this problem. Because neural networks can tolerate small errors, it is possible to reduce the calculation speed and power consumption at the expense of accuracy. In this study, we propose a method to reduce the power consumption of the circuit by lowering the operating voltage of the static random-access memory (SRAM) that is utilized to store the weights. In the proposed method, using two different operating voltages of SRAM, we used different bit error rates (BERs) for error-tolerant and non-error-tolerant. We demonstrated the relationship between the BER and recognition rate, and the appropriate combination of the BER and circuit configuration that maintains a high recognition rate.

INDEX TERMS Neural network, SRAM, approximate computing.

I. INTRODUCTION

Advances in machine learning technology have facilitated the recognition and classification of data by computers with at least the same or improved levels of accuracy compared to humans. Machine learning is widely used in various applications such as image classification [1] and speech recognition [2]. These algorithms require considerable computation to achieve high recognition accuracy, which increases computation time and power consumption.

The error tolerance of neural networks has attracted significant attention as an approach to solving this problem. A neural network is an algorithm that imitates the human brain with a hierarchical structure of multiple neurons connected in layers back and forth. Because features are distributed and stored across neurons, errors in a few neurons or slight fluctuations in the weight of stored synapses do not result in a complete loss of information. Therefore, we sacrifice the accuracy of the synaptic weights to reduce the calculation time and minimize power consumption. Previous research has included methods related to weight sharing [3], quantization techniques that reduce the number of bits required to store weights through approximate computing [4], and in-memory computing that applies the physical properties of circuit elements [5], [6], [7]. In this study, we improved energy efficiency by lowering the supply voltage of static random-access memory (SRAM) used for the weights. In similar research, there is a method that involves using 6T-SRAM and 8T-SRAM with different bit error rates (BERs) and another method uses 9T-SRAM and 12T-SRAM to perform inference with high accuracy even at low voltages [8], [9]. However, the increasing circuit area poses an issue [10], [11].

This paper presents a low-power neural network system. The proposed scheme uses only 6T-SRAM to avoid increasing circuit area. The energy efficiency was improved by lowering the supply voltage of the SRAM storing the weights. However, SRAMs have a higher BER when the supply voltage is reduced because of the effect of static noise margin [12], [13], [14]. Higher BER can lead to low recognition accuracy. So, this study, first, clarifies the conditions of the operating voltage that can achieve a recognition accuracy 99% as high as that of error-free circuits (in which BER is

The associate editor coordinating the review of this manuscript and approving it for publication was Khursheed Aurangzeb.

0 for any memory cells). A single-precision floating point (32 bits) is used to store the weights. The floating point consists of three parts: the sign, exponent, and mantissa. When an error occurs in the exponent part of the floating point, as compared to the mantissa part, it is expected to have a significant impact on recognition accuracy. Therefore, using two different operating voltages, we designed the system in such a manner that the BER can be set low for bits that have a large impact on recognition accuracy and high for bits that have a smaller impact. Appropriate bit selection reduces power consumption while simultaneously maintaining a high recognition accuracy.

This paper is organized as follows: Section II introduces preliminary knowledge, Section III describes the proposed memory structure, Section IV presents the simulation and evaluation, and Section V concludes the paper.

II. REVIEW

A. NEURAL NETWORKS

An NN is an algorithm designed to mimic neural circuits in the human brain and has a hierarchical structure consisting of a large number of neurons. The strength of the connections between the neurons in each layer is set by a parameter called ''weights'' The values of the neurons in the next layer are obtained by performing multiply-accumulate operations on the weights and values of the neurons, and further passing them through a nonlinear activation function. There are several types of NNs such as convolutional neural networks (CNNs) and recurrent neural networks. In this paper, we discussed the most basic type of NN, that is, the fully connected neural network.

An NN involves two processes: learning and inference. In the learning process, the weights were optimized using a training dataset. In the inference process, the weights optimized in the training process were further used to classify and evaluate the data. The learning process consists of a forward propagation process and a backpropagation process. In the forward propagation process, the sum-of-products operation of neurons and weights and nonlinear transformations using activation functions are performed in order, starting from the input layer. The values of the output layer obtained in this process were passed through the objective function to derive the error from the target value. In backpropagation, the weights of each layer are adjusted using the error backpropagation method to reduce the error. In the inference process, only the forward propagation method was used for the evaluation.

The proposed scheme targets errors occurring in the training mode. When an error occurs, weight information stored in SRAM makes unwished change. The changed wrong weight is used in the inference mode. So, this can lower recognition accuracy.

B. FLOATING POINT REPRESENTATION

A floating-point number is expressed without a fixed decimal point. It consists of a significant part, an exponent part, and a mantissa part. The IEEE754 floating-point format defines half-precision (16bits), single precision (32bits), double precision (64bits), and quadruple precision (128bits). In this study, we used a single-precision floating point (32bits) in IEEE754 format. The IEEE754 floating-point format can represent not only normalized numbers but also denormalized numbers, infinity, and not a number (NaN). The correspondence between each bit and the type of data that can be represented is listed in Table 1. When an error occurs in the exponential part, it is necessary to consider the possibility that the value is no longer a normalized number or numerical error.

TABLE 1. Data expressed in floating point.

Type	Exponent part	Mantissa part
Zero	$(00000000)_2$	0
Denormalized number	$(00000000)_2$	≠0
Normalized number	≠0, ≠255	Arbitrary number
Infinity	$(11111111)_2$	0
Not a Number	$(11111111)_2$	≠0

C. SRAM

Static random-access memory (SRAM) is a volatile memory. Compared with dynamic random access memory (DRAM), which is also a volatile memory, SRAM consumes less power and is capable of high-speed processing. In contrast, it is more difficult for SRAM to achieve a higher density than DRAM because of the complex structure of the recording element. In this study, we used SRAM to store weights. Figure 1 shows a model of the BER versus the operating voltage for the SRAM. We used the graph of the SRAM failure rate presented by Yang et al. [11]. It shows the BER for SRAM with a process rule of 28 nm when the operating voltage varies from 0.45 V to 0.8 V. Low operating voltage brought about high BER. This is caused by static noise margin [12], [13], [14].

III. PROPOSED METHOD

Figure 2 shows a model of the memory structure for storing NN weights in memory. Each weight was stored at a single-precision floating point. If the operating voltage of the SRAM is lowered in the model shown in Figure 2, the BER of all bits will be higher, and errors will strongly affect the recognition rate. Therefore, we propose the model shown in Figure 3. In the model, we used two different operating voltages for the SRAM, where we supplied a higher voltage for the bits that are strongly affected by the errors and a lower voltage for the bits that are more tolerant to the errors. By lowering the operating voltage of the SRAMs using error-tolerant bits, we reduced the overall power consumption. At the floating point, bit flipping in the high-order bits significantly affects the value. Therefore, we set a low BER for the top n bits of the exponential part and a high BER for the other bits.

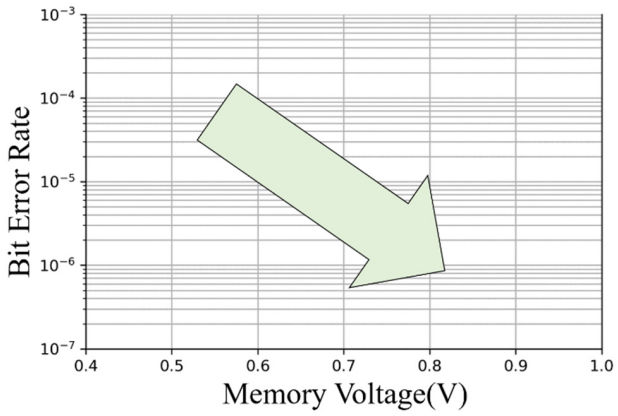

FIGURE 1. A model for SRAM operating voltage and bit error rate.

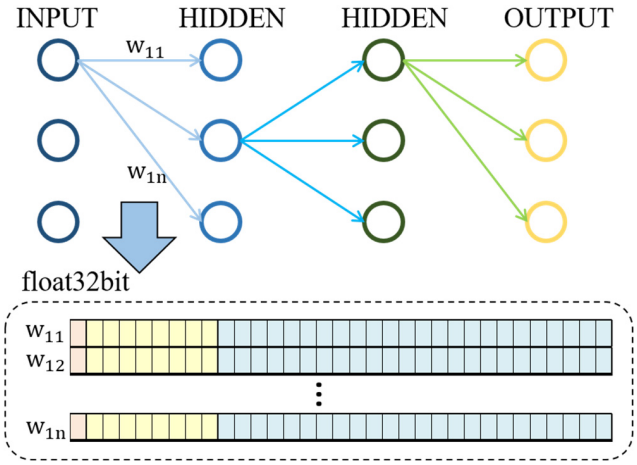

FIGURE 2. A model of the memory structure for NN weights.

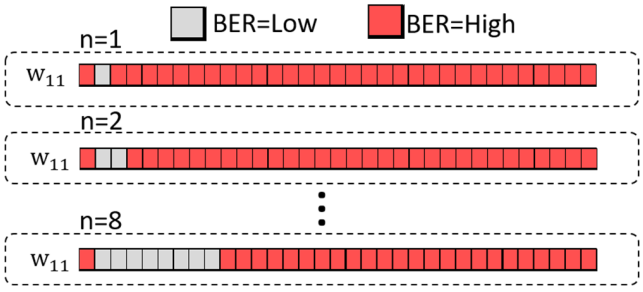

FIGURE 3. Proposed model of memory structure for NN weights.

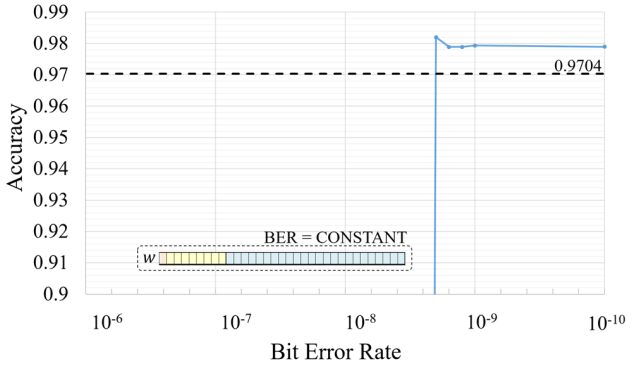

FIGURE 4. Recognition rate when BER is constant.

IV. SIMULATION

A. SIMULATION CONDITIONS

In Section 4, we reveal how an increase in BER affects the recognition rate. We used Python as the simulation tool and employed the MNIST dataset to train a fully connected DNN composed of four neuron layers. The number of nodes in each layer is set to 784-256-128-10 [15]. The mini-batch method was used for training, where the number of steps was 5,000 and the batch size was 600. The number of training data is 60,000 and the number of test data is 10000. The learning rate is set to 0.5, and the ReLU function is used for the middle layer activation function and the SoftMax function for the output layer activation function. The neural network weights are initialized with a Gaussian distribution with a standard deviation of 0.01. For the evaluation criterion, we used 0.9704, which is 99% of the recognition rate of 0.9803 when the BER of all the bits was 0.

B. SIMULATION OF EXISTING MODEL

Figure 4 shows the relationship between the BER and recognition rate when the BER is assumed to be constant for all bits. For the recognition rate, we used the average value of the successful learning among the three simulations. When BER was less than $10^{-8.7}$, the recognition rate was greater than 0.9704. However, when the BER was greater than $10^{-8.7}$, the learning was unsuccessful.

We consider the cause of the learning failure to be an error that occurs in the exponential part. Learning failed because certain weight values increased significantly and propagated to the entire system. It is also possible that the learning process fails because the floating point indicates a nonnormalized number. In the single-precision floating point, the exponent part is $(01111111)_2$ when the number is greater than or equal to 1.0, and less than 2.0. If there is an error in the most significant bit of the exponent, the weights will indicate NaN or infinity and the learning will fail. However, in this study, we did not observe any learning failures that could be attributed to this cause. This is because the weights diverged owing to errors earlier than the learning progressed, and the weights exceeded 1.0.

C. SIMULATION OF PROPOSED MODEL

Figure 5 shows the relationship between the BER and recognition rate when the BER of the high-order n bits of the exponential part is set to 0. For the recognition rate, we used the average value of successful learning among the three simulations. In Figure 4, the recognition rate is polarized, whereas, in Figure 5, it gradually decreases. This indicates that the error in the most significant bit of the exponent has

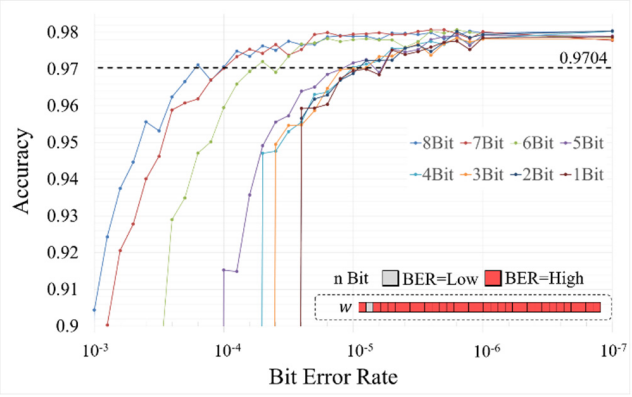

FIGURE 5. Recognition rate when the high-order n bits of the exponential part are protected.

TABLE 2. Efficiency of memory voltage at each bit.

Bit Number	Maximum BER	Memory Voltage (V)	Overall Memory Voltage Ratio (%)
1	$10^{-5.3}$	0.60	61.3
2	$10^{-5.1}$	0.59	61.6
3	$10^{-5.2}$	0.59	62.8
4	$10^{-5.0}$	0.59	64.1
5	$10^{-5.0}$	0.59	65.4
6	$10^{-4.3}$	0.52	61.0
7	$10^{-4.1}$	0.51	61.7
8	$10^{-3.8}$	0.47	60.3

a strong impact on the recognition rate. When the BER was less than $10^{-5.2}$, the recognition rate was 0.9704 or higher under all the conditions. Among those with a recognition rate of 0.9704 or higher, the highest BER was for n = 8 and BER = $10^{-3.8}$.

Table 2 lists the memory voltage characteristics. We obtained the maximum BER among the BERs with a recognition rate of 0.9704 or higher, and the corresponding operating voltage of the SRAM. In addition, we determined the ratio between the memory voltage of all bits at 1 V and the memory voltages under each condition. When n = 8, the efficiency was the highest and the memory voltage decreased to approximately 60%. When n $\neq$ 8, the memory voltage decreased to approximately 61-65%

D. SIMULATION OF LOW VOLTAGE SRAM

In Subsection D, we simulated the behavior of the SRAM at the memory voltage calculated in Subsection C. We used the most efficient voltage 0.47V for evaluation and 1V for comparison. We used HSPICE and a 45 nm and 22 nm PTM for the simulation and set the operating environment at 27°C and 1.0V supply voltage. In this study, the parasitic capacitance is simulated as 1pF HSPICE was a circuit simulator provided by Synopsys.

Table 3 (A) summarizes the delay times and power consumption of the 45 nm RTM models. In terms of reading delay time, the delay time was 0.219ns when the memory voltage

TABLE 3. Characteristics of low voltage SRAM (A) 45 nm mode.

Memory Voltage (V)	Read Delay Time (ns)	Write Delay Time (ns)	Power Consumption (nW)
1	0.219	1.022	107.29
0.47	2.502	0.548	73.02
	(B) 22 NM MODEL		
Memory Voltage (V)	Read Delay Time (ns)	Write Delay Time (ns)	Power Consumption (nW)
1	0.051	0.659	934.84
0.47	0.821	0.376	453.94

was 1V, while it was 2.502ns when the memory voltage was 0.47V, a significant increase. We assume that this is owing to the parasitic capacitance of the bit lines because the read delay time decreased significantly in a simulation where the parasitic capacitance was lowered from 1 pF. In terms of write delay time, the delay time was 1.022ns when the memory voltage was 1V, while it was 0.548ns at 0.47V, almost half the delay time. This is because the voltage of 0.47 V is almost half of 1 V. This shortens the time to charge memory cells. The power consumption was 107.29 nW when the memory voltage was 1 V, whereas it was 73.02 nW at 0.47 V, a reduction of approximately 32%. Table 3 (B) shows the results using the 22 nm PTM model. The simulation conditions are the same as for the 45 nm PTM model shown in Table 3 (A). The comparison results between 1 V and 0.47 V are similar to those for 45 nm. For example, the reading latency for 1 V is much shorter than that for 0.47 V. The writing latency for 1 V is almost twice as long as that for 0.47 V.

V. CONCLUSION

In this paper, we proposed a method to reduce power consumption by using two different operating voltages for the SRAM that stores the weights during neural network training. The existing model requires a BER of $10^{-8.7}$ or less to achieve as 99% high recognition accuracy as the error-free circuit, but by protecting the most significant bit of the exponential part, we achieved the 99% high recognition accuracy with a BER of $10^{-5.3}$ or less. By protecting all eight bits of the exponential part, we achieved a BER of $10^{-3.8}$, keeping the 99% high accuracy. The highest efficiency was achieved when the high-order 8 bits of the exponential part were protected, resulting in a memory voltage of 0.47V, reducing the overall memory voltage by about 40%. In addition, the power consumption was reduced by about 32%, but there were still issues such as an increase of 1142% in the read delay time.

The proposed scheme is evaluated with only the MNIST dataset and a small NN. We expect the proposed scheme is available for a large dataset and NN, such as ResNet. The future work includes the evaluation of such a large system.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2012, pp. 1097–1105.

[2] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," 2014, *arXiv:1412.5567*.
[3] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Efficient inference engine on compressed deep neural network," *ACM SIGARCH Comput. Archit. News*, vol. 44, no. 3, pp. 243–254, Jun. 2016, doi: 10.1145/3007787.3001163.
[4] S. Venkataramani, A. Ranjan, K. Roy, and A. Raghunathan, "AxNN: Energy-efficient neuromorphic systems using approximate computing," in *Proc. ISLPED*, Aug. 2014, pp. 27–32.
[5] G. Cristiano, M. Giordano, S. Ambrogio, L. P. Romero, C. Cheng, P. Narayanan, H. Tsai, R. M. Shelly, and G. W. Burr, "Perspective on training fully connected networks with resistive memories: Device requirements for multiple conductances of varying significance," *J. Appl. Phys.*, vol. 124, Oct. 2018, Art. no. 151901.
[6] P. Narayanan, "Toward on-chip acceleration of the backpropagation algorithm using nonvolatile memory," *IBM J. Res. Develop.*, vol. 61, nos. 4–5, pp. 11:1–11:11, Jul. 2017.
[7] S. Ambrogio, P. Narayanan, H. Tsai, C. Mackin, K. Spoon, A. Chen, A. Fasoli, A. Friz, and G. W. Burr, "Accelerating deep neural networks with analog memory devices," in *Proc. 2nd IEEE Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Sep. 2020, pp. 149–152.
[8] C. Roy and A. Islam, "Design of low power, variation tolerant single bitline 9T SRAM cell in 16-nm technology in subthreshold region," *Microelectron. Rel.*, vol. 120, May 2021, Art. no. 114126, doi: 10.1016/j.microrel.2021.114126.
[9] Y.-W. Chiu, Y. H. Hu, M. H. Tu, J. K. Zhao, Y. H. Chu, S. J. Jou, and C. T. Chuang, "40 nm bit-interleaving 12T subthreshold SRAM with data-aware write-assist," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 9, pp. 2578–2585, Sep. 2014, doi: 10.1109/TCSI.2014.2332267.
[10] L. Chang, "An 8T-SRAM for variability tolerance and low-voltage operation in high-performance caches," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 956–963, Apr. 2008, doi: 10.1109/JSSC.2007.917509.
[11] G. Srinivasan, P. Wijesinghe, S. S. Sarwar, A. Jaiswal, and K. Roy, "Significance driven hybrid 8T-6T SRAM for energy-efficient synaptic storage in artificial neural networks," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2016, pp. 151–156.
[12] L. Yang, D. Bankman, B. Moons, M. Verhelst, and B. Murmann, "Bit error tolerance of a CIFAR-10 binarized convolutional neural network processor," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5, doi: 10.1109/ISCAS.2018.8351255.
[13] L. Yang and B. Murmann, "SRAM voltage scaling for energy-efficient convolutional neural networks," in *Proc. 18th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2017, pp. 7–12.
[14] X. Sun, R. Liu, Y.-J. Chen, H.-Y. Chiu, W.-H. Chen, M.-F. Chang, and S. Yu, "Low-VDD operation of SRAM synaptic array for implementing ternary neural network," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 10, pp. 2962–2965, Oct. 2017, doi: 10.1109/TVLSI.2017.2727528.
[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

KEISUKE KOZU received the B.E. and M.E. degrees from Chiba University, in 2020 and 2022, respectively. He was a Student of the Graduate School of Science and Engineering, Chiba University.

YUYA TANABE received the B.E. degree from Chiba University, in 2021. He is currently a Student of the Graduate School of Science and Engineering, Chiba University.

MASATO KITAKAMI (Member, IEEE) received the B.S. degree in electrical and electronic engineering, the M.S. degree in computer science, and the Dr.Eng. degree from the Tokyo Institute of Technology, Tokyo, Japan, in 1991, 1993, and 1996, respectively. He joined the Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, in April 1996, and moved to the Department of Information and Image Sciences, Chiba University, in December 1999. From April 2001 to March 2003, he was with the VLSI Design and Education Center, The University of Tokyo. Since April 2017, he has been with the Graduate School of Engineering, Chiba University, where he is currently an Associate Professor. His research interests include error control coding, dependable parallel/distributed systems, and error control in data compression. He is a member of the IEICE. He received the Young Engineer Award from the IEICE Japan, in 1999.

KAZUTERU NAMBA received the B.E., M.E., and Ph.D. degrees from the Tokyo Institute of Technology, in 1997, 1999, and 2002, respectively. He joined Chiba University, in 2002. He is currently an Associate Professor with the Graduate School of Engineering, Chiba University. His current research interest includes dependable computing. He is a member of the IEICE and the IPSJ.