IEICE TRANS. INF. & SYST., VOL.E107–D, NO.8 AUGUST 2024

PAPER Special Section on Multiple-Valued Logic and VLSI Computing

Error-Tolerance-Aware Write-Energy Reduction of MTJ-Based Quantized Neural Network Hardware

Ken ASANO[†,††a)], ***Nonmember***, **Masanori NATSUI**[†b)], ***Member***, ***and*** **Takahiro HANYU**[†c)], ***Senior Member***

SUMMARY The development of energy-efficient neural network hardware using magnetic tunnel junction (MTJ) devices has been widely investigated. One of the issues in the use of MTJ devices is large write energy. Since MTJ devices show stochastic behaviors, a large write current with enough time length is required to guarantee the certainty of the information held in MTJ devices. This paper demonstrates that quantized neural networks (QNNs) exhibit high tolerance to bit errors in weights and an output feature map. Since probabilistic switching errors in MTJ devices do not have always a serious effect on the performance of QNNs, large write energy is not required for reliable switching operations of MTJ devices. Based on the evaluation results, we achieve about 80% write-energy reduction on buffer memory compared to the conventional method. In addition, it is demonstrated that binary representation exhibits higher bit-error tolerance than the other data representations in the range of large error rates.

key words: *MTJ device, error tolerance, quantized neural network, deep learning*

1. Introduction

Deep Neural Networks (DNNs) have revolutionized various fields of artificial intelligence, from computer vision and natural language processing to autonomous driving and medical diagnostics. As the demand for real-world applications of DNNs increases, energy-efficient neural network hardware is required, especially for resource-constrained devices. In response to the above demand, the use of nonvolatile memory utilizing magnetic tunnel junction (MTJ) [1], [2] devices has attracted increased research attention. An MTJ device with unlimited endurance, a short switching time, and CMOS compatibility is a promising candidate for realizing low-power, high-performance logic circuits [3]–[5].

However, the stochastic behavior [6] of MTJ devices is a critical issue in terms of energy consumption. The switching probability of MTJ devices depends on a write current applied to the device. Conventionally, CMOS-based integrated circuits are designed based on worst-case criteria that guarantee expected operation with a sufficient margin. Therefore, a large write current with enough time length is required for reliable switching operations, resulting in large write energy. As one approach to address this issue, researchers have proposed a method to reduce write energy by utilizing the high error tolerance of neural networks [7]–[10]. Since write errors have a small impact on the performance of neural networks, large write energy is not required for reliable switching operations. Based on the above concept as shown in Fig. 1, our previous work [10] achieved write-energy reduction of MTJ-based weight storage memory.

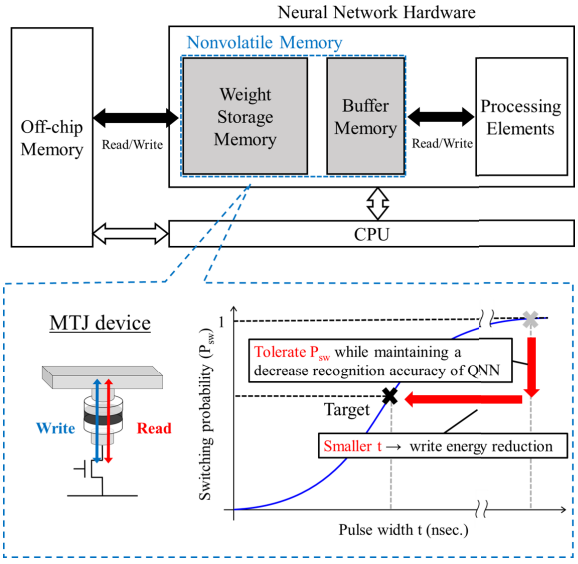

Fig. 1 Basic concept to achieve write-energy reduction. Since write errors in MTJ devices have a small impact on the performance of neural networks, write operations can be performed under low-energy conditions.

In this paper, we extend our previous work to hardware using MTJ devices for both storage and buffer memory, and investigate write energy reduction based on the proposed method. Since neural network hardware requires many memory accesses, it is desirable to have the largest possible on-chip buffer memory to minimize accesses to off-chip memory. MTJ-based nonvolatile memory is a promising candidate for large buffer memory in neural network hardware because of its high density compared to SRAM, which is conventionally used as on-chip memory. Solving the write energy problem of MTJ devices would enable the realization of energy-efficient neural network hardware in which both storage memory and buffer memory are implemented with high-density embedded nonvolatile memory.

The main contributions of this paper are as follows: (1) We evaluate the impact of write errors in storage and buffer memory on network performance in neural networks with

Manuscript received October 27, 2023.
Manuscript revised March 7, 2024.
Manuscript publicized April 22, 2024.

[†]Research Institute of Electrical Communication, Tohoku University, Sendai-shi, 980–8578 Japan.
[††]Graduate School of Engineering, Tohoku University, Sendai-shi, 980–8578 Japan.
a) E-mail: asano.ken.q2@dc.tohoku.ac.jp
b) E-mail: masanori.natsui.a8@tohoku.ac.jp
c) E-mail: takahiro.hanyu.c4@tohoku.ac.jp

DOI: 10.1587/transinf.2023LOP0007

various quantization methods. (2) We show that the binary representation has relatively high error tolerance and that the impact of errors in buffer memory on the inference results of the network is smaller than that of storage memory. (3) Based on the simulation results, we evaluate the effect of utilizing error tolerance on write energy reduction.

2. Basic Concepts

2.1 Quantization

Quantization is a technique to reduce computational costs and memory requirements by representing the weights and activations with lower precision values such as fixed-point number representation [11], [12], binary representation [13]–[16], or other low-precision formats. This technique significantly reduces the memory footprint of DNN models and accelerates multiply-and-accumulate (MAC) operations that are the dominant workloads of DNNs. Quantization not only streamlines the processing of neural networks but also minimizes the bandwidth needed for data transfer. When implementing DNN hardware, one of the most significant overheads comes from off-chip memory accesses. Off-chip memory accesses require up to several orders of magnitude higher energy than computation [17]. By using low-precision data types to represent parameters, more data can be stored in the same memory space, which leads to fewer off-chip memory accesses.

Binarized neural networks (BNNs) [13]–[16], the extreme case of quantization, are promising candidates for implementing compact and energy-efficient DNN hardware. In BNNs, both weights and activations are constrained to binary values (−1 or 1). The binarized convolution layer in BNNs is simplified as follows:

$$O = sign(popcount(XNOR(W_i, X_i) - T)), \quad (1)$$

where O, $sign$, $popcount$, W_i, X_i, and T represent output features, the sign function, the bitcounting operation, and a learned threshold, respectively. As shown in Eq. (1), multiplication is replaced by the XNOR operation, and addition is replaced by the bitcounting operation, leading to compact hardware implementations.

2.2 MTJ Device

An MTJ device, one of the spintronics devices, is a promising storage device for energy-efficient hardware implementation of DNNs as zero standby current can be achieved thanks to non-volatility. Figure 2 shows a device structure and R-I characteristics of a perpendicular MTJ device. The MTJ device consists of two ferromagnetic layers separated by a thin barrier layer. By controlling the direction of the magnetization of the free layer with respect to the fixed layer with a bi-directional write current, the MTJ device exhibits two distinct resistance states: (1) The state when the two magnetic layers have anti-parallel spin directions is low resistance (R_{AP}). (2) The state when they have parallel spin

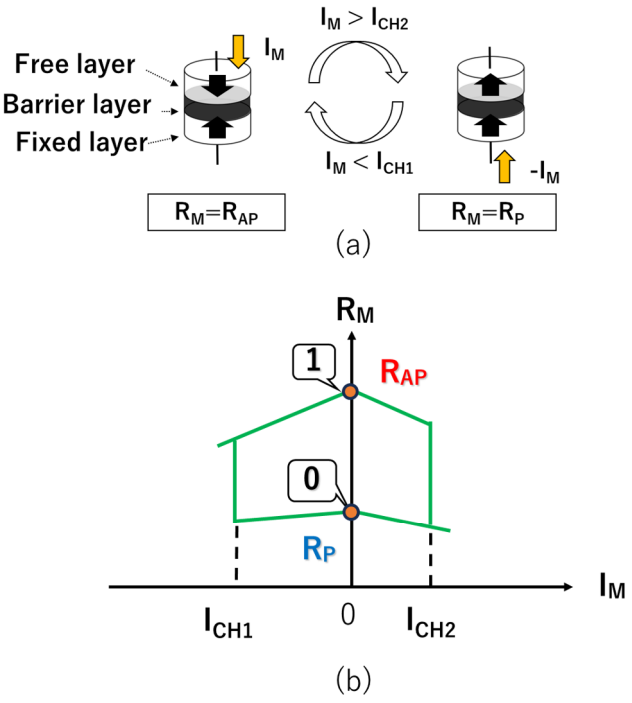

Fig. 2 MTJ device: (a) Device structure. Depending on the magnetization of the free layer, MTJ devices have two different resistance states. (b) R-I characteristic. R_M is changed by applying a bidirectional write current.

directions is low resistance (R_P). Since the resistance state remains when a power supply is detached from the MTJ device, the MTJ device can be considered as a 1-bit nonvolatile memory. The MTJ device provides several advantages, including high read/write speed, high endurance, and high density compared to other nonvolatile devices, such as resistive random access memory (ReRAM) and phase change memory (PCRAM). The effectiveness of the MTJ device in realizing energy-efficient neural network hardware has been demonstrated through several examples [18]–[21].

Although the MTJ device has the potential for realizing energy-efficient logic circuits, its stochastic characteristic induces write errors, causing bit errors in logic circuits. A state transition of the MTJ device depends on the magnitude, direction, and duration of the applied write current. Assuming that the magnitude of the write current is constant, the relationship between the switching probability P_{SW} and the duration of an applied write current to the MTJ device can be approximated as follows:

$$P_{SW}(t) = 1 - \exp(-\frac{t}{\tau_P}), \quad (2)$$

where t is the duration of the applied write current, and τ_P is the parameter determined by the composision of the MTJ device. Since P_{SW} depends on an applied write current, write errors occur when a sufficient amount of write current is not applied to the MTJ device for enough time length. In applications where write errors could compromise the overall system functionality, we need to resolve this by using a large write current or by adding a kind of error-correcting

mechanism such as error-correcting code (ECC) [22], [23], resulting in large energy consumption.

2.3 Write-Energy Reduction by Tolerating Write Errors

One approach for the challenge of large write energy is to perform write operations under low-energy conditions by leveraging high error tolerance of neural networks [7]–[10], [24], [25]. Since neural networks have the property that errors in the data inside the network have a small impact on its performance, the write energy can be reduced by allowing a certain amount of switching errors to the extent that the performance of the neural network is not degraded. This method is a new approach that achieves power savings by combining the stochastic characteristics of the device and the error tolerance of neural networks, leading to advantages such as simpler control circuits and a smaller area compared to conventional methods such as the self-write termination (SWT) method [26], [27].

In this paper, we consider using MTJ devices for buffer memory and weight storage memory. To confirm that the concept of reducing write energy is useful, we evaluate the impact of write errors in each memory on the performance of QNNs. In addition, we evaluate the relationship between data representation and bit-error tolerance of QNNs. In Sect. 3, we describe the evaluation method in more detail.

3. Evaluation Setup

Figure 3 shows the layer structure of QNNs evaluated in this paper. We use typical convolution neural networks based on VGG neural network [28] architecture for image classification tasks. QNNs consist of six convolution layers with filter counts of 128, 128, 256, 256, 512, and 512, and three fully-connected layers with 1024, 1024, and 10 neurons. QNNs use no bias and their all convolution layers have a kernel size of 3×3.

To evaluate the impact of data representations on bit-error tolerance, we consider four types of quantization methods for weights and activations: (1) binary representation (Binary), (2) 8-bit fixed-point number representation (Fxp8), (3) 16-bit fixed-point number representation (Fxp16) and (4) 32-bit fixed-point number representation (Fxp32) as shown in Fig. 4. Since numerical value changes caused by bit errors depend on the value range, we unify the value range for each data representation. In addition, to evaluate the impact of errors on recognition accuracy, it is desirable that the recognition accuracy of each data representation be as close as possible.

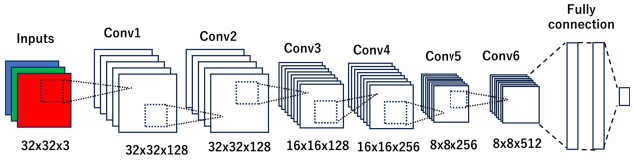

Fig. 3 Layer structure of QNNs.

Therefore, we also consider Binary_x3, a model that achieves the same level of recognition accuracy as fixed-point representations despite using a binary representation by tripling filter counts for each layer. Binary_x3 consists of six convolution layers with filter counts of 384, 384, 768, 768, 1536, and 1536, and three fully-connected layers with 1024, 1024 and 10 neurons. Binary_x3 uses no bias and their all convolution layers have a kernel size of 3×3.

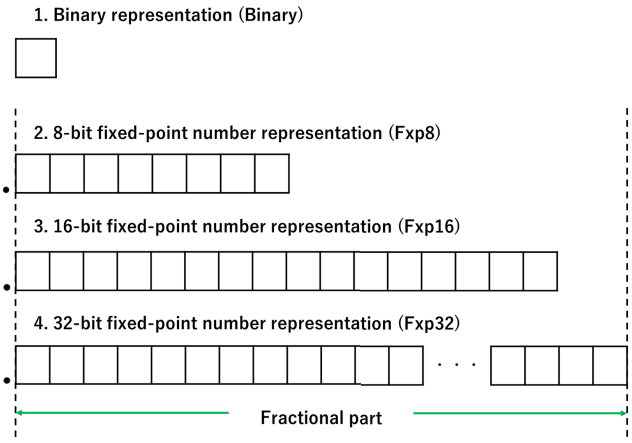

Fig. 4 List of quantization methods for weights and activations.

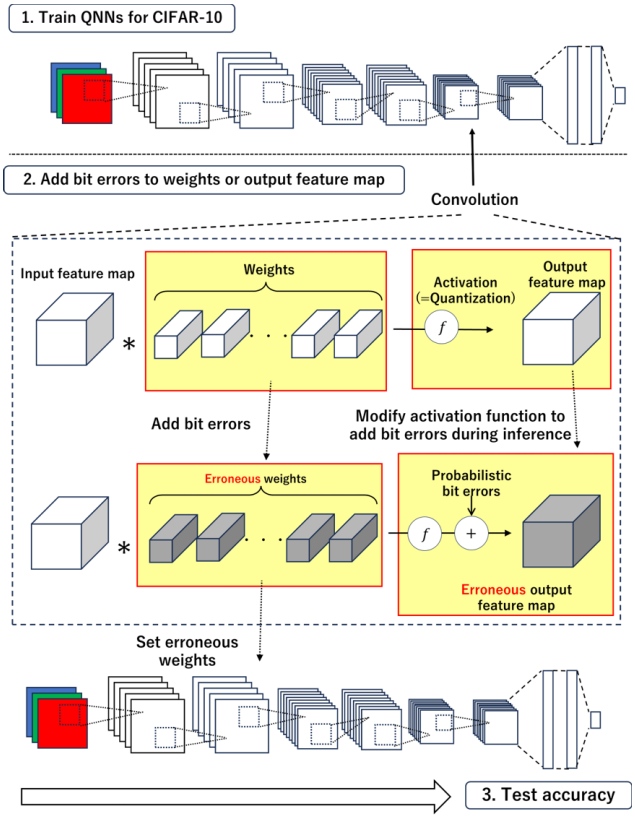

Fig. 5 The method to evaluate bit-error tolerance of neural networks. To evaluate the impact of write errors in buffer and weight storage memory on the processing of QNNs, we artificially add bit errors to an output feature map and weights, respectively.

Table 1 shows the recognition accuracy of QNNs trained on the CIFAR-10 dataset. The CIFAR-10 dataset is widely used as a benchmark dataset for the image classification task. The CIFAR-10 contains a total of 60,000 labeled images, which are divided into 10 classes. There are 50,000 training images and 10,000 test images. The resolution of all images is 32 by 32 pixels. Based on the learning algorithm of the BNN [16], our QNNs use quantized weights and activations during both the training phase and the test phase. Under these conditions, regardless of data representation, QNNs achieve recognition accuracy in the upper 80%.

Figure 5 shows the method to evaluate the bit-error tolerance of QNNs. To evaluate the impact of write errors in weight storage memory or buffer memory on the performance of QNNs, we add probabilistic bit errors to the weights or an output feature map after training. In neural network hardware, once training is completed, the write frequency of weights is basically once. On the other hand, buffer memory stores the output of each layer during the computation process. Therefore, we set the evaluation conditions as follows: (1) The process of adding bit errors to the weights is applied only once after training the model. (2) We modify the activation function to add bit errors each time an output feature map is generated. After adding bit errors to the weights or modifying the activation function, we evaluate the relationship between the bit-error rate and recognition accuracy of QNNs. Figure 6 shows the method to emulate the switching errors of MTJ devices. When weights or an output feature map are represented by binary values, the signs of -1 and $+1$ are flipped with a uniform probability for each element. When weights or an output feature map are represented by fixed-point number representation, we first convert them to a bit string. Then, we generate bit errors with a uniform probability for each bit and convert them back to fixed-point number representation.

Table 1 Test accuracy of QNNs trained on CIFAR-10 dataset.

Quantization method	Binary	Binary_3x	Fxp8	Fxp16	Fxp32
Test accuracy [%]	85.07	86.62	87.29	87.36	87.23

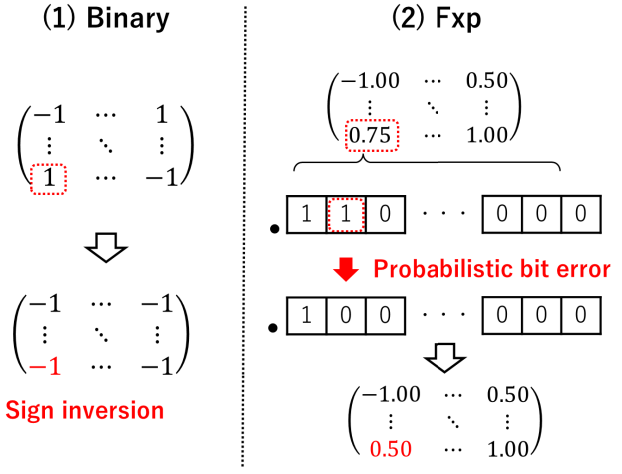

Fig. 6 The method to emulate switching errors of MTJ devices. Bit errors are generated in each bit with uniform probability.

Based on the above methods, we evaluate the impact of switching errors in MTJ devices on the recognition accuracy of QNNs. Based on the evaluation result, we also discuss the effect of write-energy reduction by tolerating switching errors.

4. Evaluation Results

4.1 Bit-Error Tolerance

Figure 7 shows the relationship between the bit-error rate and the test accuracy of QNNs. The simulation of test accuracy was repeated 10 times and the mean values are presented. QNNs show recognition accuracy close to the original accu-

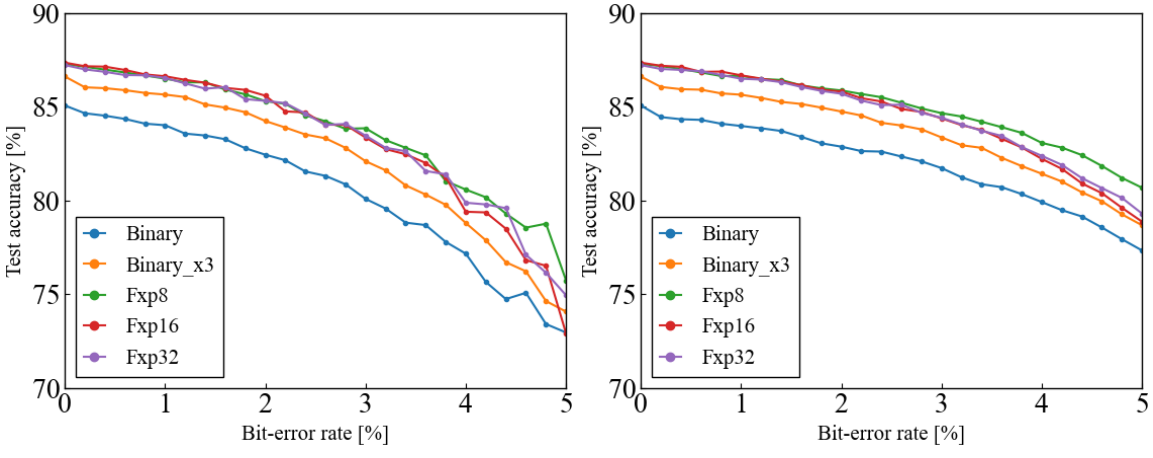

Fig. 7 (a) Bit-error rate in weights vs. test accuracy. (b) Bit-error rate in output feature map vs. test accuracy. The impact of errors in buffer memory on the inference results of the network is smaller than that of storage memory.

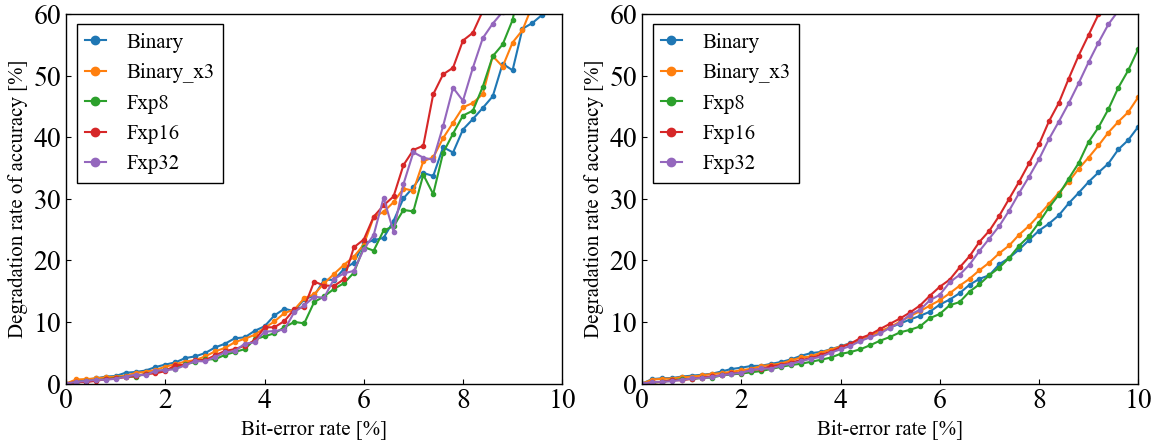

Fig. 8 (a) Bit-error rate in weights vs. degradation rate of accuracy. (b) Bit-error rate in output feature map vs. degradation rate of accuracy. Binary representation has relatively high error tolerance in the range of large error rates.

racy even when bit errors are added. In addition, comparing the case of adding errors to weights and the case of adding errors to an output feature map, QNNs show higher tolerance to bit errors in an output feature map. This result means that write errors in buffer memory are more tolerable than write errors in weight storage memory. Since the write frequency of buffer memory is generally more frequent than that of weight storage memory, we can realize large write-energy reduction by utilizing this property. To verify the impact of data representation on bit-error tolerance, we evaluated the degradation rate of accuracy from the result in Fig. 7, as shown in Fig. 8. In the range of relatively small error rates, each data representation exhibits almost the same bit-error tolerance. On the other hand, in the range of large error rates, Binary shows a higher bit-error tolerance than the other data representations in both cases where bit errors occur in weights and in an output feature map.

To discuss the reason why Binary shows high bit-error tolerance, we introduce the metric called root mean squared error ratio (RRMSE). According to H. Huang et al. (2023), soft error influence on neural network accuracy depends on RRMSE [29] defined as follows:

$$RRMSE = \sqrt{\sum_{l}^{L} \left(\frac{var(\boldsymbol{\Delta}_l)}{var(\boldsymbol{x}_l)} \right)}, \tag{3}$$

where L, $\boldsymbol{\Delta}_l$ and $\boldsymbol{x}_l$ represent the total number of neural network layers, the variation induced by soft errors in layer l and the input activations in layer l. Figure 9 shows the relationship between the bit-error rate and RRMSE of QNNs evaluated in this paper. We calculate RRMSE using 1,000 images of the CIFAR-10 dataset, and the mean values are presented. The QNN with binary representation shows a smaller increase in RRMSE with an increasing bit error rate compared to other data representations. This means that binary representation is insensitive to bit errors. Although further evaluation is needed to clarify the relationship be-

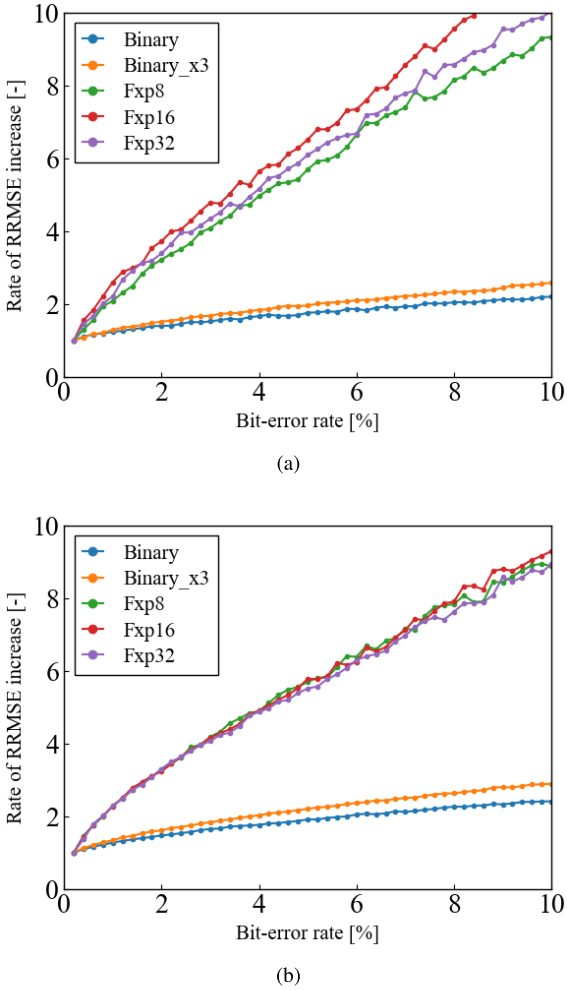

Fig. 9 Relationship between bit-error rate and rate of RRMSE increase. (a) RRMSE when adding bit errors to weights. (b) RRMSE when adding bit errors to an output feature map. Binary representation shows a smaller increase in RRMSE with an increasing bit error rate compared to other data representations.

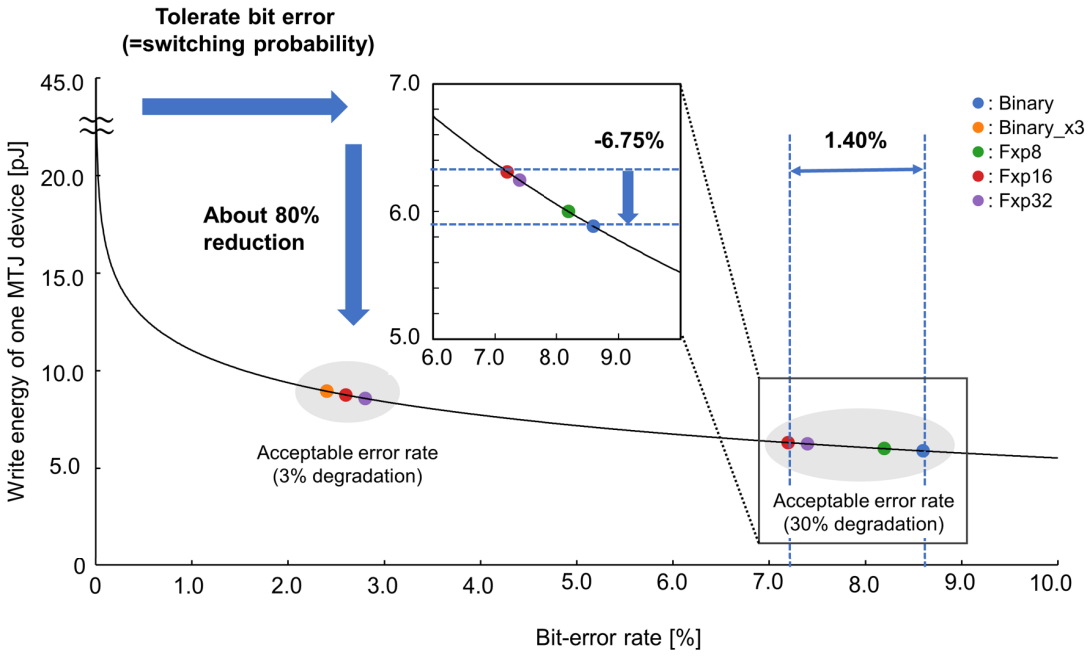

Fig. 10 The relationship between the write energy of one MTJ device and bit-error rate. When the acceptable degradation rate of accuracy is 3%, the acceptable error rate for each data representation remains almost the same. On the other hand, when the acceptable degradation rate of recognition accuracy is as large as 30%, there is a difference in the acceptable error rate for each data representation.

Table 2 Write-energy reduction compared to write energy at 10^{-6}%.

Quantization method	Acceptable error rate (3% degradation)	Acceptable error rate (30% degradation)	Write energy [pJ] (3% degradation)	Write energy [pJ] (30% degradation)
Binary	2.40	8.60	8.94 (-79.8%)	5.88 (-86.7%)
Binary_x3	2.40	8.20	8.94 (-79.8%)	6.00 (-86.4%)
Fxp8	2.80	8.20	8.57 (-80.6%)	6.00 (-86.4%)
Fxp16	2.60	7.20	8.75 (-80.2%)	6.31 (-85.7%)
Fxp32	2.80	7.40	8.57 (-80.6%)	6.24 (-85.9%)

tween data representation and error tolerance, RRMSE can be an effective metric.

4.2 Write-Energy Reduction

In this section, we discuss the effect of write-energy reduction in buffer memory based on the result of Fig. 8 for the following two reasons: (1) QNNs evaluated in this paper show higher tolerance to bit errors in an output feature map than bit errors in weights. (2) The write frequency of buffer memory is higher than that of weight storage memory, and the impact of reducing write energy is more significant.

Figure 10 shows the relationship between the write energy of one MTJ device and the bit-error rate based on Eq. (2). The magnitude of the write current applied to the MTJ device is fixed at 150 μA. The parameter τ_P, which has a negative correlation with the magnitude of the write current, is configured with a value of 1.48×10^{-8}. Since the switching probability P_{SW} and the duration of the write current t are positively correlated, we can reduce t by tolerating a certain amount of bit-error rate, resulting in write-energy reduction. In this paper, we define 10^{-6}% as the conventional error rate required for reliable switching operations, based on the range where no error occurs when writing all parameters. Assuming that QNNs can tolerate 3% or less degradation rate of accuracy, the proposed method can reduce the write energy of the MTJ device in buffer memory by about 80%. Note that because the energy required to switch MTJ devices dominates the write process, a similar degree of energy reduction is anticipated regardless of the memory configuration. We also evaluated the effect of write-energy reduction when QNNs can tolerate a degradation rate as large as 30%. In this case, there is a difference in the acceptable error rate for each data representation, with Binary exhibiting the largest acceptable error rate. In terms of write energy, this results in a 6.75% difference compared to the data representation with the lowest tolerable error rate. The results of the above evaluations are summarized in Table 2.

Our findings are useful for applications that do not require the high performance of individual models, such as

ensemble learning, including bagging and boosting. In such applications, collective behavior and decision-making are more crucial than the performance of single nodes. This means that the entire system can operate efficiently even if the performance of individual nodes is sometimes limited or imprecise. As a concrete example, Ref. [30] proposed an ensemble learning method that enables object detection with an overall average accuracy of over 80% by combining multiple models with an overall average accuracy of about 60%. BNNs are promising candidates for energy-efficient solutions in such applications.

5. Conclusion

In this paper, we evaluate the impact of bit errors in the weights and output feature map on the performance of QNNs and find that QNNs exhibit high bit-error tolerance. By utilizing this property, we achieved about 80% write-energy reduction on buffer memory compared to the conventional method. In addition, it is demonstrated that the BNN evaluated in this work exhibits higher bit-error tolerance than the other data representations. This finding is important for energy-efficient implementation in applications where the low performance of individual learners is not a problem.

The results of the comparison between Binary and Binary_x3 in Fig. 8 (b) suggest that the error tolerance of neural networks may depend not only on data representation but also on model size and layer structure. Therefore, as a future perspective, it is important to clarify the impact of model size and layer structure on bit error tolerance and distinguish it from the impact of data representation in order to clarify how to derive an appropriate neural network architecture that meets the required specifications.

Acknowledgements

This work was supported in part by JSPS KAKENHI (JP21H03405, JP21H04868), JST CREST (JPMJCR19K3), NEDO (JPNP16007), and CRP (R05/B17) of RIEC, Tohoku Univ., Japan.

References

[1] S. Ikeda, J. Hayakawa, Y.M. Lee, F. Matsukura, Y. Ohno, T. Hanyu, and H. Ohno, "Magnetic tunnel junctions for spintronic memories and beyond," IEEE Trans. Electron Devices, vol.54, no.5, pp.991–1002, 2007.

[2] S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H. Gan, M. Endo, S. Kanai, J. Hayakawa, F. Matsukura, and H. Ohno, "A perpendicular-anisotropy cofeb–mgo magnetic tunnel junction," Nature materials, vol.9, no.9, pp.721–724, 2010.

[3] N. Sakimura, Y. Tsuji, R. Nebashi, H. Honjo, A. Morioka, K. Ishihara, K. Kinoshita, S. Fukami, S. Miura, N. Kasai, T. Endoh, H. Ohno, T. Hanyu, and T. Sugibayashi, "10.5 a 90nm 20mhz fully nonvolatile microcontroller for standby-power-critical applications," 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), pp.184–185, 2014.

[4] M. Natsui, G. Yamagishi, and T. Hanyu, "Design of a highly reliable nonvolatile flip-flop incorporating a common-mode write error detection capability," Japanese Journal of Applied Physics, vol.60, no.SB, p.SBBB02, Feb. 2021.

[5] M. Natsui, D. Suzuki, A. Tamakoshi, T. Watanabe, H. Honjo, H. Koike, T. Nasuno, Y. Ma, T. Tanigawa, Y. Noguchi, M. Yasuhira, H. Sato, S. Ikeda, H. Ohno, T. Endoh, and T. Hanyu, "A 47.14- μW 200-mhz mos/mtj-hybrid nonvolatile microcontroller unit embedding stt-mram and fpga for iot applications," IEEE J. Solid-State Circuits, vol.54, no.11, pp.2991–3004, 2019.

[6] D. Bedau, H. Liu, J.Z. Sun, J.A. Katine, E.E. Fullerton, S. Mangin, and A.D. Kent, "Spin-transfer pulse switching: From the dynamic to the thermally activated regime," Applied Physics Letters, vol.97, no.26, p.262502, 2010.

[7] T. Hirtzlin, M. Bocquet, J.-O. Klein, E. Nowak, E. Vianello, J.-M. Portal, and D. Querlioz, "Outstanding bit error tolerance of resistive ram-based binarized neural networks," 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), pp.288–292, IEEE, 2019.

[8] T. Hirtzlin, B. Penkovsky, J.-O. Klein, N. Locatelli, A.F. Vincent, M. Bocquet, J.-M. Portal, and D. Querlioz, "Implementing binarized neural networks with magnetoresistive ram without error correction," 2019 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), pp.1–5, IEEE, 2019.

[9] L. Yang, D. Bankman, B. Moons, M. Verhelst, and B. Murmann, "Bit error tolerance of a cifar-10 binarized convolutional neural network processor," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), pp.1–5, IEEE, 2018.

[10] K. Asano, M. Natsui, and T. Hanyu, "Write-energy relaxation of mtj-based quantized neural-network hardware," 2023 IEEE 53rd International Symposium on Multiple-Valued Logic (ISMVL), pp.7–11, IEEE, 2023.

[11] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.2704–2713, 2018.

[12] D. Lin, S. Talathi, and S. Annapureddy, "Fixed point quantization of deep convolutional networks," International conference on machine learning, pp.2849–2858, PMLR, 2016.

[13] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," Advances in neural information processing systems, vol.28, 2015.

[14] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," European conference on computer vision, vol.9908, pp.525–542, Springer, 2016.

[15] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1," arXiv preprint arXiv:1602.02830, 2016.

[16] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," Advances in neural information processing systems, vol.29, 2016.

[17] Y.-H. Chen, T. Krishna, J.S. 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, 2017.

[18] Y. Pan, P. Ouyang, Y. Zhao, W. Kang, S. Yin, Y. Zhang, W. Zhao, and S. Wei, "A multilevel cell stt-mram-based computing in-memory accelerator for binary convolutional neural network," IEEE Trans. Magn., vol.54, no.11, pp.1–5, 2018.

[19] S. Jain, A. Ranjan, K. Roy, and A. Raghunathan, "Computing in memory with spin-transfer torque magnetic ram," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.26, no.3, pp.470–483, 2017.

[20] S. Jung, H. Lee, S. Myung, H. Kim, S.K. Yoon, S.W. Kwon, Y. Ju, M. Kim, W. Yi, S. Han, B. Kwon, B. Seo, K. Lee, G.-H. Koh, K. Lee, Y. Song, C. Choi, D. Ham, and S.J. Kim, "A crossbar array of magnetoresistive memory devices for in-memory computing," Nature, vol.601, no.7892, pp.211–216, 2022.

[21] M. Natsui, D. Suzuki, N. Sakimura, R. Nebashi, Y. Tsuji, A. Morioka, T. Sugibayashi, S. Miura, H. Honjo, K. Kinoshita, S. Ikeda, T. Endoh, H. Ohno, and T. Hanyu, "Nonvolatile logic-in-memory array processor in 90nm mtj/mos achieving 75% leakage reduction using cycle-based power gating," 2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers, pp.194–195, IEEE, 2013.
[22] B. Del Bel, J. Kim, C.H. Kim, and S.S. Sapatnekar, "Improving stt-mram density through multibit error correction," 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp.1–6, IEEE, 2014.
[23] W. Kang, W. Zhao, Z. Wang, Y. Zhang, J.O. Klein, Y. Zhang, C. Chappert, and D. Ravelosona, "A low-cost built-in error correction circuit design for stt-mram reliability improvement," Microelectronics Reliability, vol.53, no.9-11, pp.1224–1229, 2013.
[24] Z. Yan, Y. Shi, W. Liao, M. Hashimoto, X. Zhou, and C. Zhuo, "When single event upset meets deep neural networks: Observations, explorations, and remedies," 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), Beijing, China, pp.163–168, 2020.
[25] G. Gambardella, J. Kappauf, M. Blott, C. Doehring, M. Kumm, P. Zipf, and K. Vissers, "Efficient error-tolerant quantized neural network accelerators," 2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), IEEE, pp.1–6, Oct. 2019.
[26] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy reduction for stt-ram using early write termination," Proceedings of the 2009 International Conference on Computer-Aided Design, pp.264–268, 2009.
[27] N. Strikos, V. Kontorinis, X. Dong, H. Homayoun, and D. Tullsen, "Low-current probabilistic writes for power-efficient stt-ram caches," 2013 IEEE 31st International Conference on Computer Design (ICCD), pp.511–514, IEEE, 2013.
[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
[29] H. Huang, X. Xue, C. Liu, Y. Wang, T. Luo, L. Cheng, H. Li, and X. Li, "Statistical modeling of soft error influence on neural networks," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol.42, no.11, pp.4152–4163, Nov. 2023.
[30] J. Lee, S.-K. Lee, and S.-I. Yang, "An ensemble method of cnn models for object detection," 2018 International Conference on Information and Communication Technology Convergence (ICTC), pp.898–901, 2018.

Ken Asano received the B.E. degree in electronic engineering from Gunma University, Kiryu, Japan, in 2022, where he is currently pursuing the M.E. degree with the Research Institute of Electrical Communication.

Masanori Natsui received the B.E. degree in electronic engineering and the M.S. and Ph.D. degrees in information Sciences from Tohoku University, Sendai, Japan, in 2000, 2002, and 2005, respectively. He is currently an Associate Professor with the Research Institute of Electrical Communication, Tohoku University. His research interest includes automated circuit design technique, nonvolatile-based circuit architecture and its application, and design of high-speed low-power integrated circuit-based on multiple-valued current-mode circuit technology. Dr. Natsui was a recipient of the IEEE Sendai Section Student Award in 2003, the Excellent Paper Award of The Institute of Electronics, Information and Communication Engineers of Japan in 2010, and the Kenneth C. Smith Early Career Award for Microelectronics Research in 2012.

Takahiro Hanyu received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1984, 1986, and 1989, respectively. He is currently a Professor with the Research Institute of Electrical Communication, Tohoku University. His research interests include nonvolatile logic circuits and their applications to ultra-low-power and/or highly dependable VLSI processors, and post-binary computing and its application to brain-inspired VLSI systems. He received the Sakai Memorial Award from the Information Processing Society of Japan, in 2000, the Judge's Special Award from the 9th LSI Design of the Year from the Semiconductor Industry News of Japan, in 2002, the Special Feature Award from the University LSI Design Contest from ASP-DAC, in 2007, the APEX Paper Award of the Japan Society of Applied Physics, in 2009, the Excellent Paper Award of IEICE, Japan, in 2010, the Ichimura Academic Award, in 2010, the Best Paper Award of IEEE ISVLSI 2010, the Paper Award of SSDM 2012, the Best Paper Finalist of IEEE ASYNC 2014, and the Commendation for Science and Technology by MEXT, Japan, in 2015.