# Binary Neural Network with 16 Mb RRAM Macro Chip for Classification and Online Training

Shimeng Yu[1*], Zhiwei Li[1], Pai-Yu Chen[1], Huaqiang Wu[2#], Bin Gao[2], Deli Wang[2], Wei Wu[2], and He Qian[2]
[1]Arizona State University, Tempe, AZ, USA, [*]Email: shimengy@asu.edu
[2]Tsinghua University, Beijing, China, [#]Email: wuhq@tsinghua.edu.cn

***Abstract*—**On-chip implementation of large-scale neural networks with emerging synaptic devices is attractive but challenging, primarily due to the pre-mature analog properties of today's resistive memory technologies. This work aims to realize a large-scale neural network using today's available binary RRAM devices for image recognition. We propose a methodology to binarize the neural network parameters with a goal of reducing the precision of weights and neurons to 1-bit for classification and <8-bit for online training. We experimentally demonstrate the binary neural network (BNN) on Tsinghua's 16 Mb RRAM macro chip fabricated in 130 nm CMOS process. Even under finite bit yield and endurance cycles, the system performance on MNIST handwritten digit dataset achieves ~96.5% accuracy for both classification and online training, close to ~97% accuracy by the ideal software implementation. This work reports the largest scale of the synaptic arrays and achieved the highest accuracy so far.

## I. Introduction

Recent advances in neuro-inspired learning algorithms have shown tremendous successes in the intelligent tasks such as image recognition when they are run on the powerful supercomputers (generally with GPU accelerators) [1]. However, the memory wall problem (the performance gap between on-chip processor and off-chip memory) in the von-Neumann architecture has become the bottleneck of executing learning algorithms with deep neural networks. For mobile applications, off-chip memory access incurs significant overhead of latency and energy consumption. Therefore, it is imperative to load the large-scale weight matrices of the neural networks on chip. In the CMOS ASIC design, SRAM is commonly used as the synaptic element [2]. However, SRAM cell occupies >200 $F^2$ (F is the technology feature size), thus it is not area-efficient to fully load the large-scale weight matrices on chip using SRAM cells.

Emerging non-volatile memories (NVMs) such as phase change memory (PCM) and resistive random access memory (RRAM) offer much higher integration density (4~6 $F^2$), therefore they are competitive candidates for synaptic elements [3]. In the recent years, PCM [4] and RRAM [5-7] based "analog" synapses that exploit the multilevel states have been demonstrated at single device level. So far, there are a few experimental implementations of simple networks on small-scale (e.g. 12×12 $TiO_x/Al_2O_3$ crossbar [8]) to medium-scale (e.g. 256×256 PCM array [9]). However, there are practical design challenges identified by the recent device-algorithm co-simulations for large-scale integration [10-11]: Although the neural networks are capable of tolerating the random effects such as device variations or noises to certain degree, the systematical effects, particularly the nonlinearity of weight update (the conductance vs. # of programming pulses), may greatly degrade the learning accuracy. Unfortunately, almost all the reported "analog" synapses [4-11] suffer from this nonlinear weight update. Further device engineering is needed to improve the linearity of weight update. Alternatively, we propose using the more mature digital RRAM as "binary" synapses.

This work aims to binarize the multilayer neural network with back-propagation, namely the binary neural network (BNN), inspired by the recent trend of network pruning and parameter compression in the deep learning community [12]. The goal is to reduce the precision of both weight and neuron to 1-bit for classification and <8-bit for online training. To validate our proposal, we experimentally implemented the BNN on Tsinghua's 16 Mb RRAM macro chip fabricated in 130 nm CMOS process. Even under finite bit yield and endurance cycles, the benchmarked system performance on MNIST handwritten digit dataset achieves ~96.5% accuracy that is closed to the floating-point software implementation (~97%).

## II. 16 Mb RRAM Macro Chip

A 16 Mb RRAM macro chip was designed by Tsinghua and fabricated in 130 nm CMOS process, and we used this chip for the demonstration of BNN. Fig. 1 shows the architectural organization of the 16 Mb chip, with 16 blocks and each block has two arrays (512×1024) sharing the sense amplifiers. The I/O width is 8-bit. The array core uses 1-transistor-1-resistor (1T1R) architecture. Fig. 2 shows the photo image of the fabricated die. The front-end fabrication (up to M4) was done in a commercial 200 mm CMOS foundry, and the back-end RRAM process was done in-house. The RRAM device structure is $TaO_x/HfO_2$ stack following our prior work [13] and it is monolithically integrated on top of the CMOS transistors between M4 and M5. Table 1 summarizes the design specifications of this prototype. The typical programming conditions are: SET: 2.7 V/50 ns; RESET: 2.6 V/50 ns. The measurement results show a bit-yield >99% of the 16 Mb macro. Fig. 3 shows the cycling endurance testing results up to ~$10^6$ cycles. No verify technique was used in the endurance test. Certainly the yield and performance could be further optimized for digital memory application, this prototype is sufficient to implement the BNN. This is because that the neural networks are inherently resilient to the random effects (i.e. the bit error due to finite yield), and the most of the RRAM cells do not update very frequently in the training process when we binarize the weight representation.

978-1-5090-3902-9/16/$31.00 ©2016 IEEE

## III. Binary Neural Network for Classfication

We used the 16 Mb RRAM macro to validate the proposed BNN. The multilayer perceptron algorithm with the sigmoid activation and back-propagation was used for the MNIST handwritten digit recognition (Fig. 4). We binarized the MNIST dataset to black and white and cropped the edges to be 20×20 images (Fig. 5). The network has three layers: input layer of 400 neurons corresponding to the 20×20 images, hidden layer of 200 neurons, and output layer of 10 neurons corresponding to the 10 classes of digits. Therefore, it has two weight matrices ($W_{1\text{-}2}$: 400×200, and $W_{2\text{-}3}$: 200×10). For ideal software baseline, we trained the network with floating point (64-bit) on CPU and the recognition accuracy saturates ~97% after tens of training epochs (Fig. 6). One epoch is 60,000 training images. Recognition is done by another set of 10,000 testing images.

For classification, we performed offline training in software after certain number of epochs, then we truncated the precision of weights/neurons to 1-bit for hardware implementation. Fig. 7 shows the accuracy vs. the training epoch. The BNN with such 1-bit classification shows an accuracy ~96.5% that is close to the ideal software baseline. Then we loaded the pre-trained weight matrices (after 50 epochs) into one 512×1024 array of the 16 Mb RRAM macro. Because of the +/- weights in the BNN (+1, 0, -1), we used two columns to represent the weight by taking the differential output. Fig. 8 shows that we assigned two regions in one array: 400×400 for $W_{1\text{-}2}$, and 200×20 for $W_{2\text{-}3}$. When we programmed the pre-trained matrices to the array, some of the bits could not be programmed to the desired states. Fig. 9 shows the experimentally measured pattern: the error bits that differ from the desired states are highlighted in the red color. Despite of the finite bit yield ~99%, the 1-bit classification can still achieve reasonably high accuracy ~96.3%. We performed simulations of even worse yield in Fig. 10, showing that the average accuracy can be >90% when the bit yield is only 90%. The redundant and massively parallel networks provide such resilience to random bit errors.

## IV. Binary Neural Network for Online Training

For online training when the weights are updated on the hardware during the run-time, the precision requirement is significantly higher than the classification. Fig. 11 shows the accuracy vs. precision of weights/neurons, indicating at least 6-bit is needed. The reason is that the back-propagation passes the small training errors from the output layer to the input layer, if the precision is insufficient, such small errors will be skipped. In the next, we used 8 binary RRAM cells to represent 1-bit of the sign (+/-) and 7-bit of the weights from the most significant bit (MSB) to the least significant bit (LSB). Then 4 arrays of 512×1024 are needed to implement the BNN for online training. The primary concern for online training is the RRAM's endurance, as the weights are updated frequently. We tracked the weight update history in the online training process of 50 epochs. Fig. 12 and Fig. 13 shows the number of switching cycles of RRAM cells (sign, and from MSB to LSB) for $W_{1\text{-}2}$, and $W_{2\text{-}3}$, respectively. We can see that LSB updates more frequently than MSB, and $W_{2\text{-}3}$ updates more frequently than $W_{1\text{-}2}$. Nevertheless, most of the RRAM cells switch $<10^4$ cycles. The simulation on BNN then fixes the RRAM states if one cell switches more than the endurance limit. Fig. 14 shows the accuracy vs. training epochs for different endurance limits. If endurance limit is low (e.g. $10^3$ cycles), the peak of accuracy only achieves ~94.8% (see the zoom-in of that regime in Fig. 15). More epochs beyond the endurance limits actually decreases the accuracy. Even with a moderate endurance limit ~$10^4$ cycles, the BNN can still achieve high accuracy ~96.9%. The reduction of precision brings improvement on energy efficiency (by 42.7% from 12-bit to 8-bit) as shown in Table 2.

## V. Conclusions

This work demonstrated the BNN on the 16 Mb RRAM macro chip, with a new record of the scale of synaptic array up to 512×1024. Even with low weight/neuron precision, the BNN achieves high learning accuracy (~96.5% for MNIST) under the non-perfect bit yield and endurance. The trade-offs of binary synapses vs. analog synapses are that binary synapses need a few more cells for online training (thus larger area and power) but they can avoid the nonlinear weight update problem in analog synapses, thus higher accuracy. The area of the neuro-synaptic core is typically limited by the relatively large pitch of peripheral neuron circuits [11], thus a few more synaptic cells can be acceptable. The proposed BNN implementation is also applicable to the neuromorphic designs with other binary memories such as SRAM, PCM and even STT-MRAM.


## Acknowledgment

This work is supported by NSF-CCF-1552687 and China Key Research and Development Program (2016YFA0201803).


## References


[1] Y. LeCun, et al., "Deep learning," *Nature*, vol. 521, pp. 436-444, 2015.

[2] P. A. Merolla, et al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, pp. 668-673, 2014.

[3] S. B. Eryilmaz, et al., "Device and system level design considerations for analog-non-volatile-memory based neuromorphic architectures," *IEEE IEDM*, 2015.

[4] M. Suri, et al., "Phase change memory as synapse for ultra-dense neuromorphic systems: application to complex visual pattern extraction," *IEEE IEDM*, 2011.

[5] S. Park, et al., "Neuromorphic speech systems using advanced ReRAM-based synapse," *IEEE IEDM*, 2013.

[6] I-T. Wang, et al., "3D synaptic architecture with ultralow sub-10 fJ energy per spike for neuromorphic computation," *IEEE IEDM*, 2014.

[7] D. Garbin, et al., "Variability-tolerant convolutional neural network for pattern recognition applications based on OxRAM synapses," *IEEE IEDM*, 2014.

[8] M. Prezioso, et al., "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, pp. 61-64, 2015.

[9] S. Kim, et al., "NVM neuromorphic core with 64k-cell (256-by-256) phase change memory synaptic array with on-chip neuron circuits for continuous in-situ learning," *IEEE IEDM*, 2015.

[10] G. W. Burr, et al., "Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element," *IEEE IEDM*, 2014.

[11] S. Yu, et al., "Scaling-up resistive synaptic arrays for neuro-inspired architecture: challenges and prospect," *IEEE IEDM*, 2015.

[12] M. Courbariaux, et al., "BinaryConnect: training deep neural networks with binary weights during propagations," *NIPS*, 2015.

[13] X. Huang, et al., "Optimization of TiN/$TaO_x$/$HfO_2$/TiN RRAM arrays for improved switching and data retention," *IEEE IMW*, 2015.

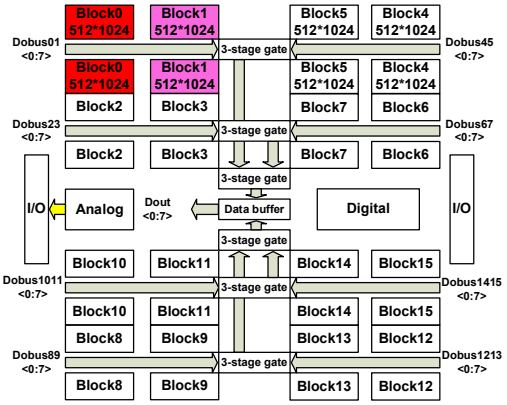


Fig. 1 The organization of Tsinghua's 16 Mb RRAM macro. There are 16 blocks and each block has two arrays (512×1024) sharing the sense amplifiers. The I/O width is 8-bit. The array core uses 1-transistor-1-resistor (1T1R) architecture.

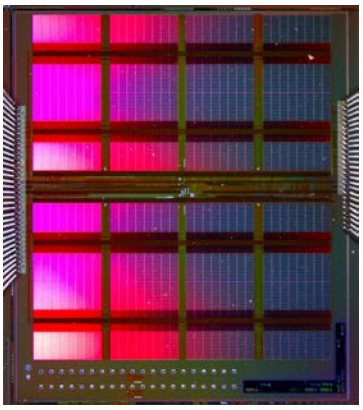

Fig. 2 The photo image of 16 Mb die. The front-end fabrication was done in a commercial CMOS foundry in 130 nm, and the back-end RRAM process was done in-house.

| | |
|---|---|
| Capacity | 16 Mb |
| Tech Node | 130 nm |
| $V_{DD\_Digital}$ | 1.8 V |
| $V_{DD\_Analog}$ | 5 V |
| $V_{WL\_SET}$ | 2-5 V/ 50 ns |
| $V_{BL\_SET}$ | 2-3 V/ 50 ns |
| $V_{WL\_RESET}$ | 3.5-5 V/ 50 ns |
| $V_{SL\_RESET}$ | 2-3 V/ 50 ns |
| I/O Width | 8 |

Table 1 Design parameters for the prototype chip.

The $TaO_x/HfO_2$ based RRAM device structure is monolithically integrated on top of the CMOS transistors between M4 and M5. The bit yield is measured to be >99%.

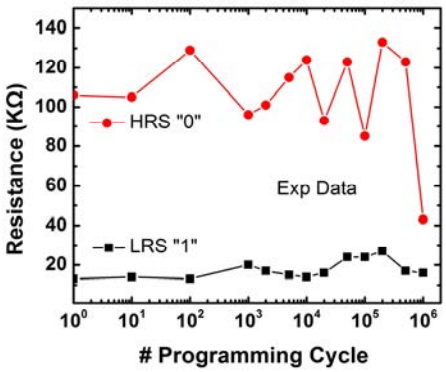


Fig. 3 Experiments on RRAM cycling endurance. The set/reset programming pulses are repeated up to $10^6$ cycles. No verify scheme is used.

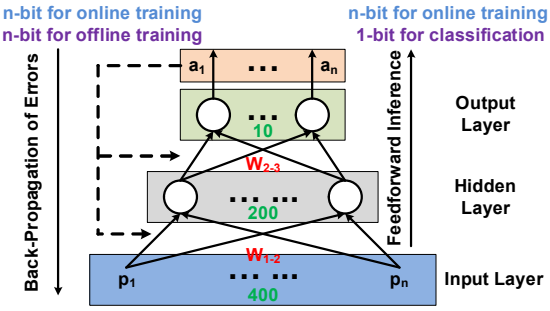


Fig. 4 The network topology of the multilayer perceptron algorithm with the sigmoid activation and back-propagation. This work binarizes the weights and neurons into binary representation, namely binary neural network (BNN). For classification with offline training, the feedforward inference is reduced to 1-bit. For online training, both feedforward inference and back-propagation is reduced to n-bit (n<8).

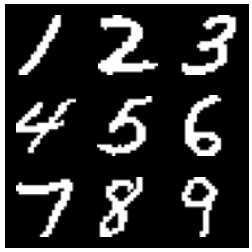

Fig. 5 Samples of MNIST handwritten digit dataset binarized to black and white and cropped the edges to be 20×20 pixels. The network has three layers: input layer of 400 neurons, hidden layer of 200 neurons, and output layer of 10 neurons Therefore, it has two weight matrices ($W_{1\text{-}2}$: 400×200, and $W_{2\text{-}3}$: 200×10).

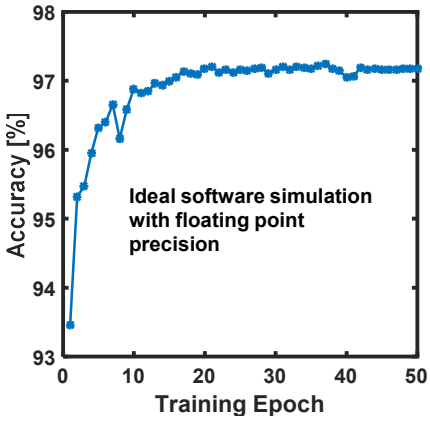


Fig. 6 Accuracy of the BNN implemented by software (64-bit floating point). The ideal baseline is ~97%. One epoch is 60,000 training images. Recognition is done by another set of 10,000 testing images.

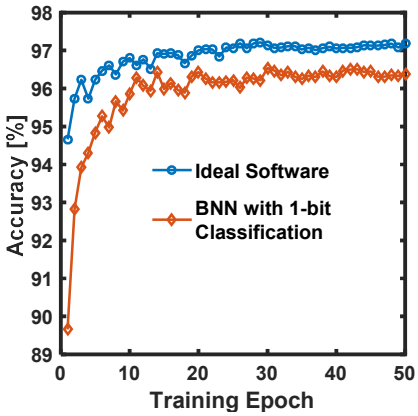


Fig. 7 Accuracy of the BNN with 1-bit weights and neurons for classification. Offline training is performed in software after certain number of epochs, then the precision is truncated to 1-bit. ~96.5% accuracy is achieved.

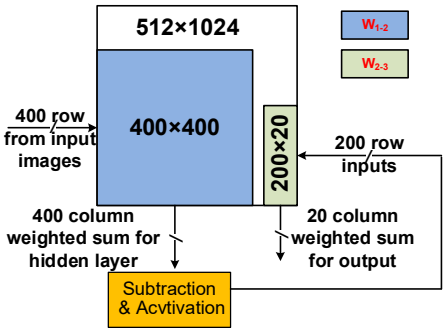


Fig. 8 Implementation of the BNN into one array of the 16 Mb macro. Because of the +/- weights in the BNN (+1, 0, -1), two columns are used to represent the weight by taking the differential output. Therefore, the size of the data pattern doubles: 400×400 for $W_{1\text{-}2}$, and 200×20 for $W_{2\text{-}3}$. After the offline training, the pre-trained weight matrices (after 50 epochs) are loaded into two regions of one 512×1024 array by one-time programming.

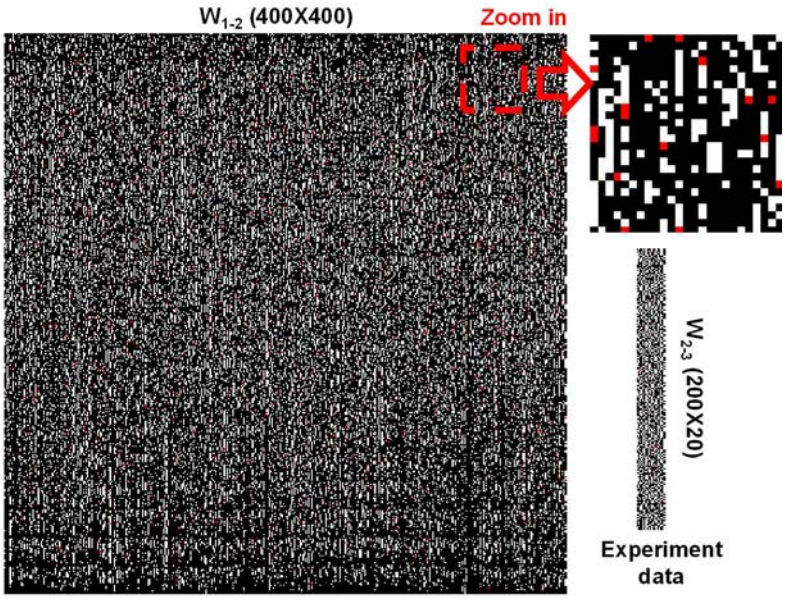


Fig. 9 Experimental measured data pattern of the weights after one-time programming into the 16 Mb macro. The error bits (that differ from the desired state in the algorithm) are marked in red color. The yield of programming is ~99%.

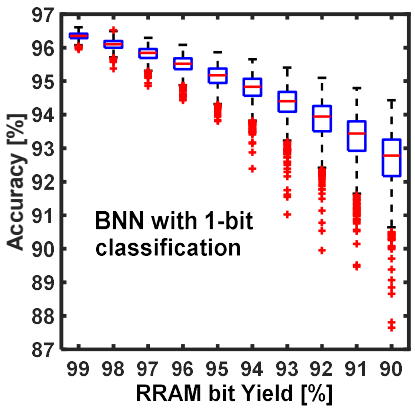


Fig. 10 Accuracy of the BNN with 1-bit classification under different RRAM bit yield. the average accuracy can be still >90% when the bit yield is only 90%. The network can tolerate the random bit error to certain degree.

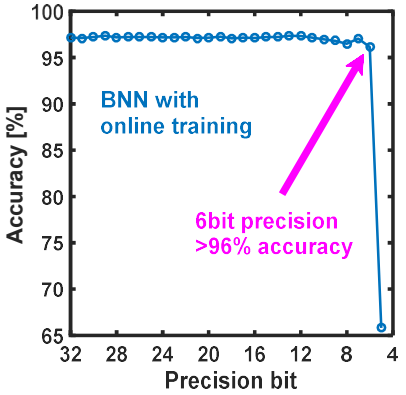


Fig. 11 Accuracy of the BNN with online training for different precision of weights and neurons. At least 6-bit is required.

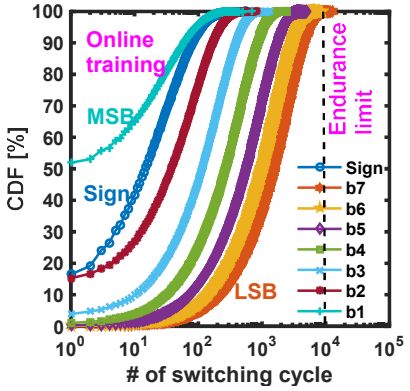


Fig. 12 Distribution of the switching cycles of the bits in weight matrix ($W_{1-2}$) during online training. LSB bits updates more frequently than MSB bits. Most cells update less than endurance limit ($10^4$ cycles).

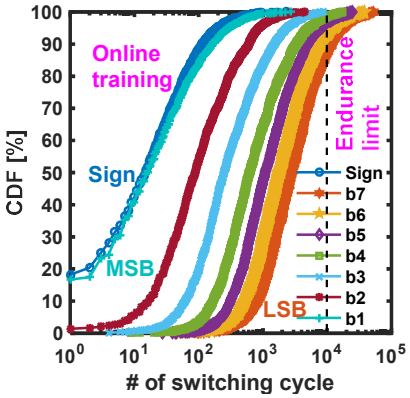


Fig. 13 Distribution of the switching cycles of the bits in weight matrix ($W_{2-3}$) during online training. $W_{2-3}$ bits updates more frequently than $W_{1-2}$ bits.

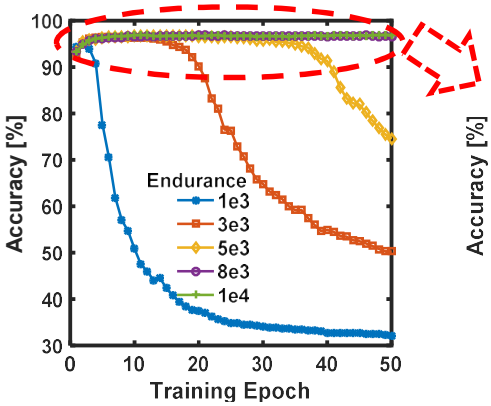


Fig. 14 Accuracy of the BNN with online training for different endurance limits. No degradation is observed if endurance >8,000 cycles.

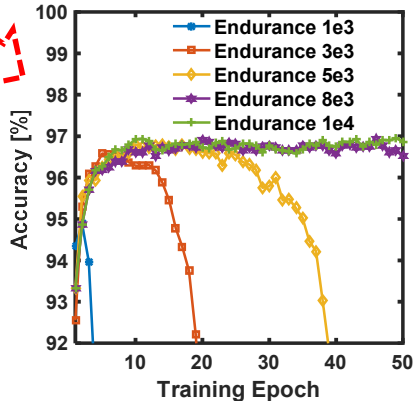


Fig. 15 Zoom-in of Fig. 14 to show the peak of the accuracy. Lower endurance results in less peak of accuracy. With $10^4$ cycles, ~96.9% accuracy is achievable for online training.

| Precision | 12-bit | 8-bit |
|---|---|---|
| Accuracy | 97.2% | 96.9% |
| # switching | 6.96E8 | 3.99E8 |
| Energy (mJ) | 9.11 | 5.22 |

Table 2 The reduction of energy consumption of lower precision in online training. The number of switching cycles and energy are counted during the 50 epochs. 8-bit weight/neuron has similar accuracy as 12-bit, but reducing energy by 42.7%. The energy here is only for the weight update.