



# Spiking Neural Network (SNN) With Memristor Synapses Having Non-linear Weight Update

Taeyoon Kim, Suman Hu, Jaewook Kim, Joon Young Kwak, Jongkil Park, Suyoun Lee, Inho Kim, Jong-Keuk Park and YeonJoo Jeong\*

Center for Neuromorphic Engineering, Korea Institutes of Science and Technology, Seoul, South Korea

Among many artificial neural networks, the research on Spike Neural Network (SNN), which mimics the energy-efficient signal system in the brain, is drawing much attention. Memristor is a promising candidate as a synaptic component for hardware implementation of SNN, but several non-ideal device properties are making it challengeable. In this work, we conducted an SNN simulation by adding a device model with a non-linear weight update to test the impact on SNN performance. We found that SNN has a strong tolerance for the device non-linearity and the network can keep the accuracy high if a device meets one of the two conditions: 1. symmetric LTP and LTD curves and 2. positive non-linearity factors for both LTP and LTD. The reason was analyzed in terms of the balance between network parameters as well as the variability of weight. The results are considered to be a piece of useful prior information for the future implementation of emerging device-based neuromorphic hardware.

## OPEN ACCESS

### Edited by:

Kyeong-Sik Min,  
Kookmin University, South Korea

### Reviewed by:

Jong-Ho Bae,  
Kookmin University, South Korea  
Stephan Menzel,  
Helmholtz-Verband Deutscher  
Forschungszentren (HZ), Germany

### \*Correspondence:

YeonJoo Jeong  
jeongyeonjoo@kist.re.kr

**Received:** 25 December 2020

**Accepted:** 15 February 2021

**Published:** 11 March 2021

### Citation:

Kim T, Hu S, Kim J, Kwak JY, Park J, Lee S, Kim I, Park J-K and Jeong Y (2021) Spiking Neural Network (SNN) With Memristor Synapses Having Non-linear Weight Update. *Front. Comput. Neurosci.* 15:646125. doi: 10.3389/fncom.2021.646125

**Keywords:** spiking neural network, memristor, non-linearity, homeostasis, LTP/LTD ratio

## INTRODUCTION

The rapid growth of technological and industrial interests in artificial intelligence (AI) represented by machine learning (ML) was appearing in the various tasks from recognition of images (Liu et al., 2020) and sounds (Jung et al., 2020) to behavioral controls of autonomous cars and robots (Atzori et al., 2016; Gao et al., 2019). The basic structure of most ML algorithms follows deep neural networks (DNN). Although the existing deep learning models have proven their powerful learning abilities, they demand expensive computing resources with a huge power budget (Demirci, 2015), making them increasingly difficult to be used on edge devices such as smartphones and watches, etc. This has led researchers to explore alternative computing paradigms inspired by the human brain, e.g., neuromorphic computing, having remarkable power efficiency.

A spiking neural network (SNN) is an artificial neural network constructed using the knowledge observed in biology, in which neurons communicate with each other using spikes via synapses connecting the neurons with adjustable weight values (Ghosh-Dastidar and Adeli, 2009). Since the spike is commonly a binary voltage pulse, neurons utilize a population in the temporal or spatial domain to encode analog input data, and hence the learning rule should involve the spatiotemporal data to train a targeted decoding system. Indeed, SNN updates synaptic weights based on localized learning rules using spatiotemporal information such as spike time-dependent plasticity (STDP), and several tasks including unsupervised and supervised learning are successfully demonstrated (Wade et al., 2008; Lee et al., 2019b). It is expected that the energy efficiency of the computing in the brain may results from the sparsity of neuron spike with low frequency and the localized approach

(Yi et al., 2015). However, when it comes to the hardware implementation, due to the inherent asynchrony and parallelism in SNN operations, conventional von Neumann systems cannot truly realize the potential of power efficiency (Jeong et al., 2016). In the regards, neuromorphic hardware has been actively studied in two approaches: using conventional complementary metal-oxide-semiconductor (CMOS) technology (Indiveri et al., 2011; Merolla et al., 2014; Imam and Cleland, 2020) and emerging type of devices such as memristor, phase-change memory (PCM), and spin-based device (Hassan et al., 2018; Nandakumar et al., 2018; Wang et al., 2018; Yang et al., 2021). Memristor, also called resistive switching memory, is one of the emerging devices that can be used as an efficient synapse block when building a future neuromorphic system. Specifically, it has a tunable conductance directly representing a synaptic weight in biology and a spike signal received from the pre-neuron is transferred to the following post-neuron in the form of an electric current (or charge) proportional to the conductance of memristor (Jo et al., 2010). In a simple crossbar array structure, the current from all the connected synapses is summed up at the post-neuron in a parallel fashion with high efficiency. In addition, memristor has mimicked various biological phenomena related to the human learning process such as short-/long-term memory, STDP, hetero-synaptic plasticity, etc (Chang et al., 2013; Yao et al., 2020). Thus, it can serve as a promising artificial synaptic component.

Despite the above merits of using the emerging device, several non-ideal effects in memristor can make it challenge to be implemented in neuromorphic hardware. For example, variation in device conductance and operation voltage, limited reliability, and non-linear conductance update can severely degrade network performances and the hardware system often requires additional operation protocols or circuits to compensate the non-idealities (Jeong et al., 2017; Brivio et al., 2018; Frascaroli et al., 2018; Li et al., 2018; Cüppers et al., 2019). However, most of the previous papers have focused on the impact of the non-ideal device properties in DNN (Agarwal et al., 2016) and few articles only studied on SNN. In Querlioz et al. (2013), SNN simulation was conducted to examine how variations in the device properties affect network performance. The network, having lateral inhibition, homeostasis mechanism, and simplified STDP rule, showed good immunity to device variations observed in weight update ( $\Delta w$ ) as well as the range of the weight. Even assuming a severe 100% of standard deviation in the device parameters, the MNIST accuracy reduced only 10%. In Woo et al. (2019), the authors confirmed the robustness of SNN against device variation again, using a model of the double-gate MOSFET device. The accuracy degradation was only 3% by 50% of the standard deviation. There is still a lack of study on the impact of device non-linearity in SNN performance with detailed analysis. A synapse in the neuromorphic hardware is ideally defined that the conductance should be updated depending on the timing difference between neuron spikes to achieve long-term potentiation (LTP) or long-term depression (LTD). However, in most of the emerging devices, the conductance change varies from the target value since it is also a function of the current conductance state showing the non-linear change. Despite active

research so far, it is still struggling to fabricate highly linear devices (Chandrasekaran et al., 2019). Hence, systematic research on how and why the network degrades by the device non-ideality is strongly demanded future robust implementation of emerging device-based neuromorphic hardware.

In this work, a high-level SNN simulation including a device model was conducted to examine the impact of the non-linear conductance update on the network performance. SNN keeps the classification accuracy high even with severe device non-linearity, if a device meets one of the two conditions: 1. symmetric LTP and LTD curves and 2. positive non-linearity factors for both LTP and LTD. In addition, we analyzed that balances in network parameters such as LTP/LTD ratio and homeostasis are broken by the non-ideal device characteristics, consequently causing degradation of the accuracy. And some of the imbalance like homeostasis can be compensated partially by selecting optimal network parameters considering imperfect device properties. The results can provide useful information for the future implementation of emerging device-based neuromorphic hardware.

## RESULTS AND DISCUSSIONS

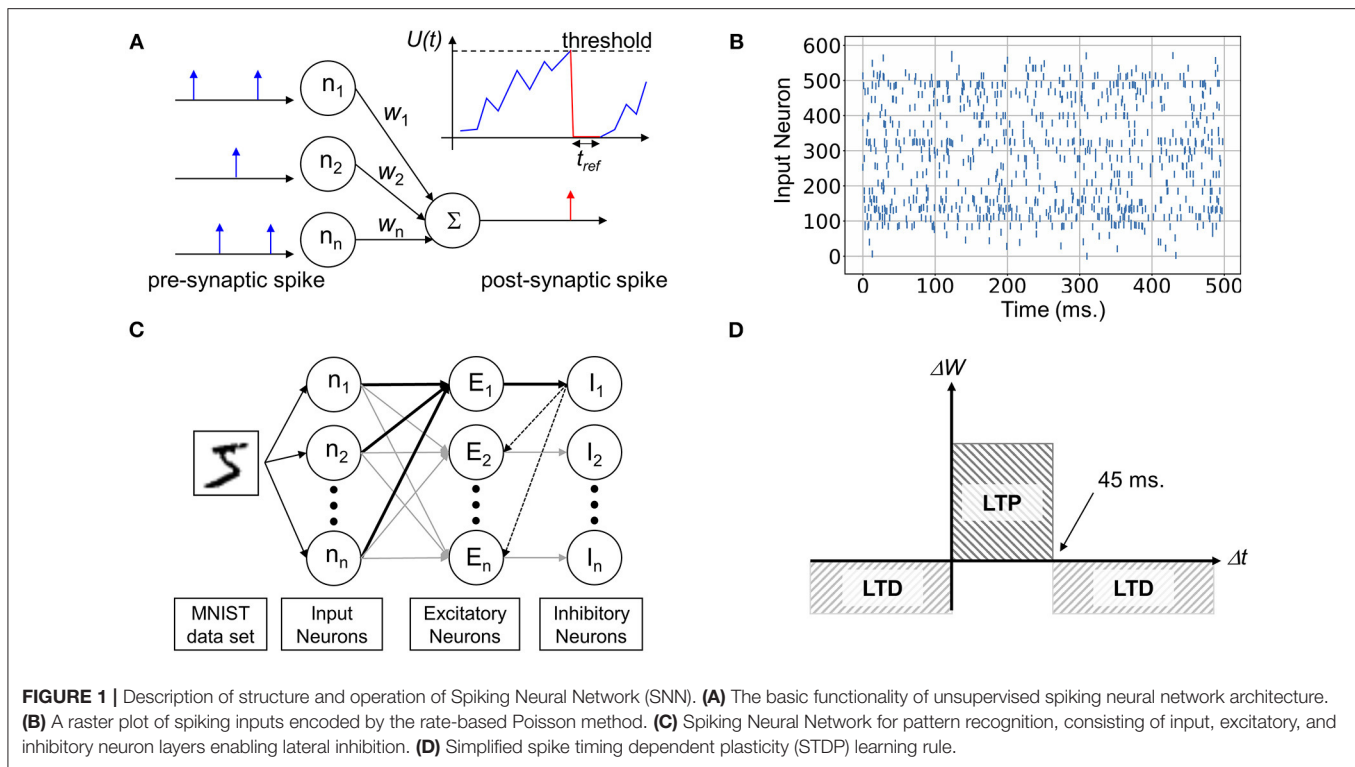
### Spiking Neural Networks Framework

In this study, a high-level simulation was conducted using a personal Python code based on the previous papers (Querlioz et al., 2013; Du et al., 2015). Input neurons are fully connected to the output neuron via synapses having different connection strength,  $w$ , as shown in **Figure 1A**. Depending on the conductivity, weights, pre-synaptic spikes generate post-synaptic current (PSC), which is gathered at the output neuron nodes and increases the membrane potential  $U(t)$ . In the leaky integrate-and-fire (LIF) neuron model, the potential spontaneously decays with a time constant  $\tau$  as following (Brunel and Sergi, 1998):

$$\tau \frac{dU_j(t)}{dt} = -U_j(t) + \sum_{i=1}^n w_{ji} \times n_i(t) \quad (1)$$

where,  $\tau$  is the leakage time constant, and  $n_i(t)$  is the input value of  $i_{th}$  neuron, and  $w_{ji}$  is a synaptic weight (conductance) between neuron  $i$  and  $j$ . Membrane potential  $U(t)$  increases whenever PSC is generated by input spikes and it will decay spontaneously with time constant,  $\tau$ . When the potential crosses over the pre-defined threshold level, it fires a post-synaptic spike and  $U(t)$  instantaneously relaxes to the resting state and maintains the level for a refractory time,  $t_{ref}$ , without responding to any received signals. Among several input encoding methods (Ponulak and Kasinski, 2011), we used the rate coding, in which the input, e.g., a pixel of an image, is converted to a spike frequency according to Equation (2) (A and B are constant value,  $A = 41/20$ ,  $B = 2.004545$ ), and then a neuron creates a spike-train by Poisson translation (Du et al., 2015) as shown in **Figure 1B**. Each input continues 500 ms to make the Poisson events with a 1 ms unit clock time.

$$frequency \text{ (Hz)} = \frac{1}{A - B \times \frac{pixel \ data}{255}} \quad (2)$$



Using the framework, we designed a two-layer SNN in **Figure 1C** to examine the impact of non-linear device properties. The MNIST handwritten dataset converted to a Poisson spike train was fed into the network, where the meaningless 2 edge pixels from the  $24 \times 24$  image were removed for simulation speed. Therefore, the SNN contains fully wired  $576 \times 300$  excitatory synaptic connections for 300 output neurons. To enable the winner-take-all (WTA) mechanism, the excitatory output neurons were connected to the subsequent inhibitory neurons in a one-to-one manner. All the inhibit neurons are fully reconnected to the excitatory neurons except for the self-inhibition path. Once one output neuron fires, it suppresses the membrane potential of the other neurons through the lateral inhibition path. This enables competition between neurons and prevents multiple columns from learning the same features (Diehl and Cook, 2015). In addition, for homeostasis, the threshold voltage deciding neuron's firing was adjusted after training every 600 samples and this will be discussed in section Excessive Firing Phenomenon and Homeostasis in detail. The distribution of firing frequency and initial and final threshold was described in **Supplementary Figure 2**.

### Updates of Synaptic Weights

It is widely believed that STDP underlies the learning process in the brain by adjusting the strength of synaptic connections (Feldman, 2012). The learning principle detects the causal relationship between a pre-synaptic and post-synaptic spike from their temporal correlation. If the pre-synaptic neuron sends a spike a few milliseconds before firing of the post-neuron, the synaptic connectivity is strengthened through a potentiation

process, whereas the weight is depressed in the reverse spike timing order. In biology, the weight change,  $\Delta w$ , is a function of timing difference,  $\Delta t$ , where  $\Delta w$  decays exponentially with increasing  $\Delta t$  undefined. However, for the simplicity of hardware implementation, we used the simplified version of STDP having fixed  $\Delta w$  for LTP and LTD as shown in **Figure 1D**.  $\Delta t$  up to 45 ms leads to an identical LTP and otherwise, LTD occurs.

### Learning and Classification

To test the performance of the designed SNN, the MNIST dataset was used: 60,000 samples for the training and 10,000 samples for the following testing process. We assumed that synapses require 256 voltage pulses to reach their maximum weight value: in other words, it can memory 256 different states ( $2^8$ ). And during the simplified STDP process, we assumed that weight change in LTP is 4-times stronger than that of LTD since it allows the best performance. The simulation result for different LTP/LTD ratio were described in **Supplementary Figure 1**. In actual hardware, various LTP/LTD ratios can be readily achieved by modulating the amplitude or duration of the LTP and LTD pulses. The network learns representative features in the input samples through updating the synaptic weights and 72 trained features out of 300 are shown in **Figure 2A**. The initial values of weight conductance were created by a uniform random distribution. The initial and final weight distribution were shown in **Figure 7A**. Since the STDP learning rule is suitable for unsupervised learning, it is hard to evaluate the network performance quantitatively. We manually assigned 0–9 labels to each feature to run a classification task. In detail, feeding 60,000 training samples, we find the most resembling one among the

300 features and count the input label whenever the feature wins. Then, the most winning label is assigned and used in the test. In the ideal device case with perfect linearity, the accuracy reached over 89% from the average of the last 10 accuracies in the evolution as shown in **Figure 2B**.

## Non-linearity in Synaptic Weight Updates

Here, we introduced non-linear update properties in a synaptic weight change, which is observed in most of the resistance switching memory devices particularly in filament-based types (Jang et al., 2015; Jeong et al., 2015). Whenever applying programming pulses, a conductance representing synaptic weight is changed and the update curve for LTP and LTD can be numerically described by Equation (3) (Agarwal et al., 2016),

$$w \text{ (LTP)} = \text{normalized } g = \alpha \left( 1 - e^{-\frac{\nu \beta p}{256}} \right) + g_{\min}$$

$$w \text{ (LTD)} = \text{normalized } g = g_{\max} - \alpha \left( 1 - e^{-\nu \left( 1 - \frac{p}{256} \right)} \right) \quad (3)$$

$$\alpha = \frac{g_{\max} - g_{\min}}{1 - e^{-\nu}}$$

, where  $g_{\max}$  and  $g_{\min}$  are the max and min values in the boundaries of the weight conductance. The  $\nu$  and  $p$  are non-linearity factor and the number of applied pulses respectively, and the denominator 256 was used for normalization.  $\beta$  is used to enable different LTP/LTD ratio and we used  $\beta = 4$ . When  $\nu$  is zero, it is a fully linear curve and keeps the weight change at a fixed value regardless of accumulated pulse numbers as shown in **Figure 3A**. However, with higher  $\nu$ , it deviates from the linear case, where the change is very rapid at the small pulse number, while it becomes slower as accumulating pulses. Another important concept is symmetry in LTP and LTD curves. From the above equation, LTP and LTD curves are symmetric when they have the same  $\nu$  value, but opposite sign, e.g.,  $(10, -10)$  or  $(-10, 10)$  for  $(\nu_{LTP} \text{ and } \nu_{LTD})$ . It is notable that depending on device mechanisms LTP and LTD could have various  $\nu$  values as reported (Liu et al., 2018). Thus, we systematically tested the SNN performance for 121 combinations of  $\nu_{LTP}$  and  $\nu_{LTD}$  ranging from  $-10$  to  $10$  and **Figure 3B** shows a contour map of classification accuracies. First, it is interesting that the spiking network maintains the high accuracy (red color) in a quite wide range of  $\nu$ , even for considerably worse nonlinearity cases like  $\nu_{LTP}, \nu_{LTD} = (-10, 10)$ . This is quite different from the DNN simulation results, where the accuracy degrades monotonically as it gets farther away from the linear value,  $\nu = 0$  (Agarwal et al., 2016). Therefore, SNN seems to be more tolerable to the non-linear weight updates of neuromorphic hardware. Next, the accuracy map can be divided into three parts that should be analyzed separately: P1, P2, and P3. In the P1 area, LTP and LTD have symmetric curves and overall accuracy is very high except for the region with large  $\nu_{LTP}$ . The second part, P2, is an area having high  $\nu$  for both LTP and LTD, and it also provides high accuracy in most of the conditions. In contrast, P3 shows low accuracy throughout the area. In the next section, a

detailed analysis of network behavior is provided. Comparing the mapping results with the nonlinearity of the real device, we found that many devices lie in the high accuracy region, since more devices locate in the 1st quadrant (P2) showing high accuracy than the 3rd quadrant (**Supplementary Figure 3**).

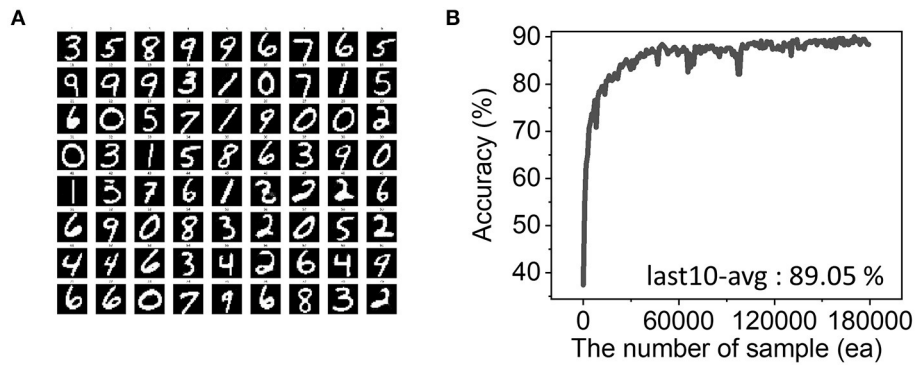
## P1: Symmetry in LTP and LTD Curves

In **Figure 3B**, the area of P1 extends from the top-left to the bottom-right of the map along the diagonal line and this mainly covers symmetrical LTP and LTD regions. To figure out the reason of the high accuracy in P1, we first selected five distinctive  $(\nu_{LTP}, \nu_{LTD})$  points that can represent the simulation conditions well: the points are  $(0,0)$ ,  $(10,-10)$ ,  $(-10,10)$ ,  $(10,10)$ , and  $(-10,-10)$ . It is expected that a different non-linearity value causes different  $\Delta w$  during weight update operation and this ultimately results in a different SNN accuracy. Hence, we plotted how  $\Delta w$  changes depending on the non-linearity as well as current weight value when applying a single LTP or LTD pulse as shown in **Figures 4A–E**. The insets show the corresponding LTP and LTD curves. For the linear case  $(0,0)$ ,  $\Delta w$  keeps the pre-defined value regardless of the current weight, and consequently, the ratio is also fixed to four from the  $\beta$  value in Equation (3) as shown in **Figures 4A,F**. In contrast, when the case changes to the non-linear conditions, the situation completely differs and there are two types of behavior. First, for the condition of  $(10,-10)$ ,  $w_{LTP}$  is getting smaller at a constant rate as the current weight level increases, while an absolute value of  $w_{LTD}$  is also decreasing as shown in **Figure 4B**. Hence, it is expected that the LTP/LTD ratio calculated from the absolute values may maintain a fixed value, even though the actual amount of  $\Delta w$  varies as a function of the current weight. Indeed, in **Figure 4F**,  $(10,-10)$  exhibits a constant LTP/LTD ratio independent of  $w$  and moreover, the value is close to our parameter  $\beta = 4$  making the network has the best performance. The same thing happens in  $(-10,10)$  despite of the opposite  $w$  dependency (**Figure 4C**). It should be noted that both  $(10,-10)$  and  $(-10,10)$  have symmetric LTP and LTD curves and are included in the region P1. Thus, the high accuracy in P1 can be accounted for the symmetric curves, leading to a constant LTP/LTD ratio close to the pre-defined  $\beta$  and as a result, keeping the network stable by balancing with other given parameters. Therefore, symmetry in weight updates is considered extremely important for SNN hardware to achieve high performance. On the other hand,  $(10,10)$  and  $(-10,-10)$  out of the P1 have asymmetric curves. As shown in **Figures 4D,E**,  $\Delta w$  change for LTP and LTD has the same direction in the plot and hence the ratio keeps changing whenever  $w$  varies (**Figure 4F**). Due to the continual ratio change, the balance between network parameters designed for the best performance could not be maintained and the accuracy becomes lower. This could not be solved by just changing network parameters because it is impossible to make a balance between fixed other parameters with the continuously changing LTP/LTD ratio.

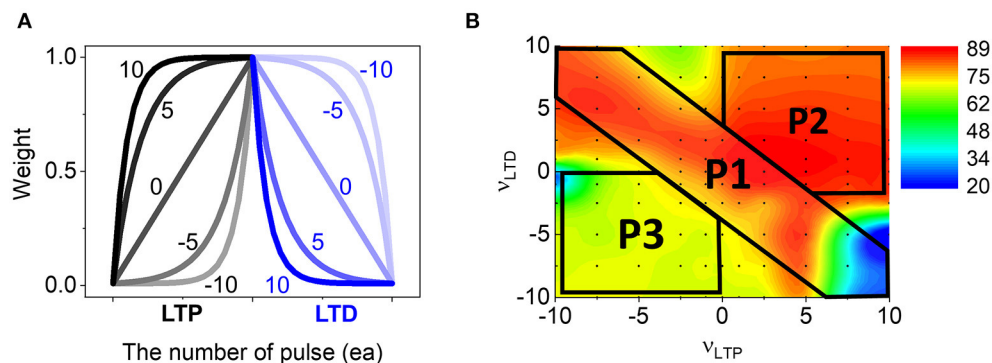
## Excessive Firing Phenomenon and Homeostasis

The next question lies in the low accuracy conditions even in the same P1, appearing at the bottom-right region with high





**FIGURE 2 | (A)** Trained 72 weights patterns out of 300 output neurons. The network successfully learns input features during training. The white pixel is the maximum weight, while the black one is the minimum. **(B)** Evolution of accuracy with 180,000 training samples. Our simulation proves 89.05% classification accuracy on average from the last 10 samples.



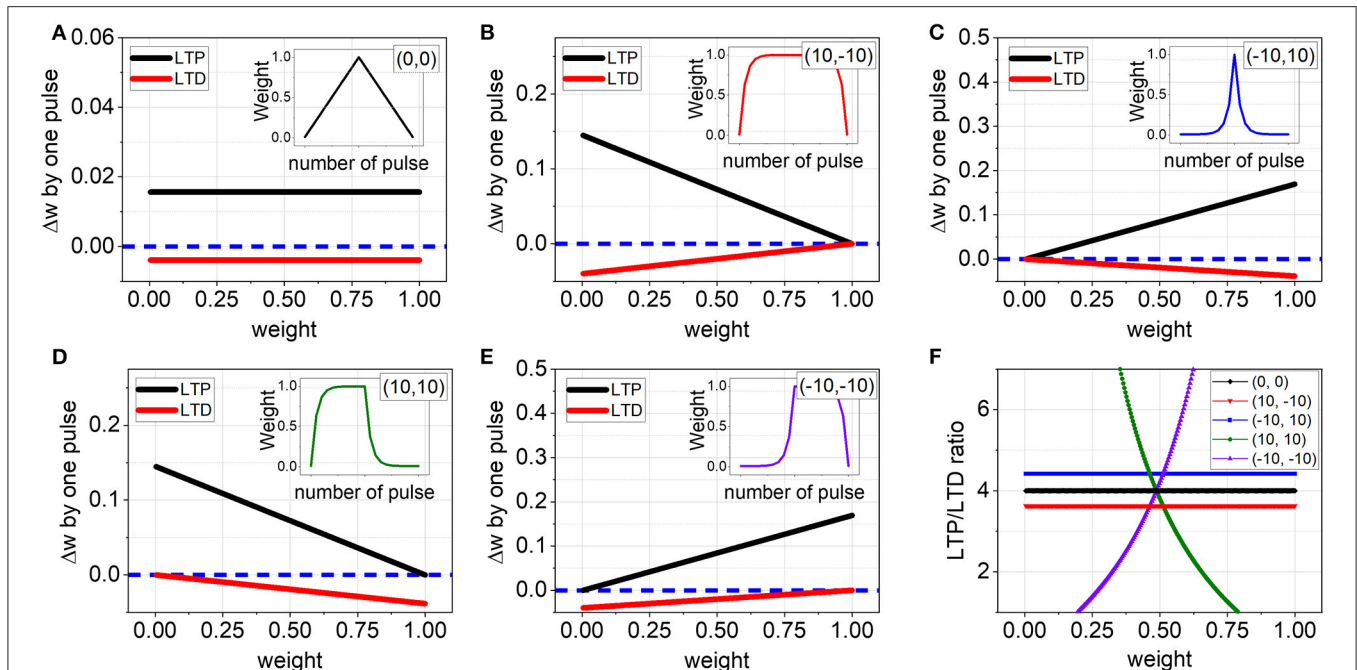
**FIGURE 3 | (A)** The change of weights by repeating LTP and LTD pulses. Depending on the non-linearity factors, the evolution shows different weight update curves: “0” indicates the perfect linear case. **(B)** The final accuracy for 121 different cases of  $v_{LTP}$  and  $v_{LTD}$ . Three areas, P1, P2, and P3 indicated in the map can depict the overall accuracy behavior.

$v_{LTP}$ . For example, in the (10,−10) case, despite the symmetric property keeping the parameter balance well, it gives very low and unstable accuracy during training compared to (0,0) as shown in **Figure 5A**. To investigate the reason, we plotted how many times of post-firing occurs in **Figures 5B,C**. In (0,0), the accumulated firing number grows almost linearly (black line in **Figure 5B**, zoomed in **Figure 5C**) since in the network, the threshold of the neuron is adjusted according to Equation (4) to keep the firing frequency similar as homeostasis in biology.

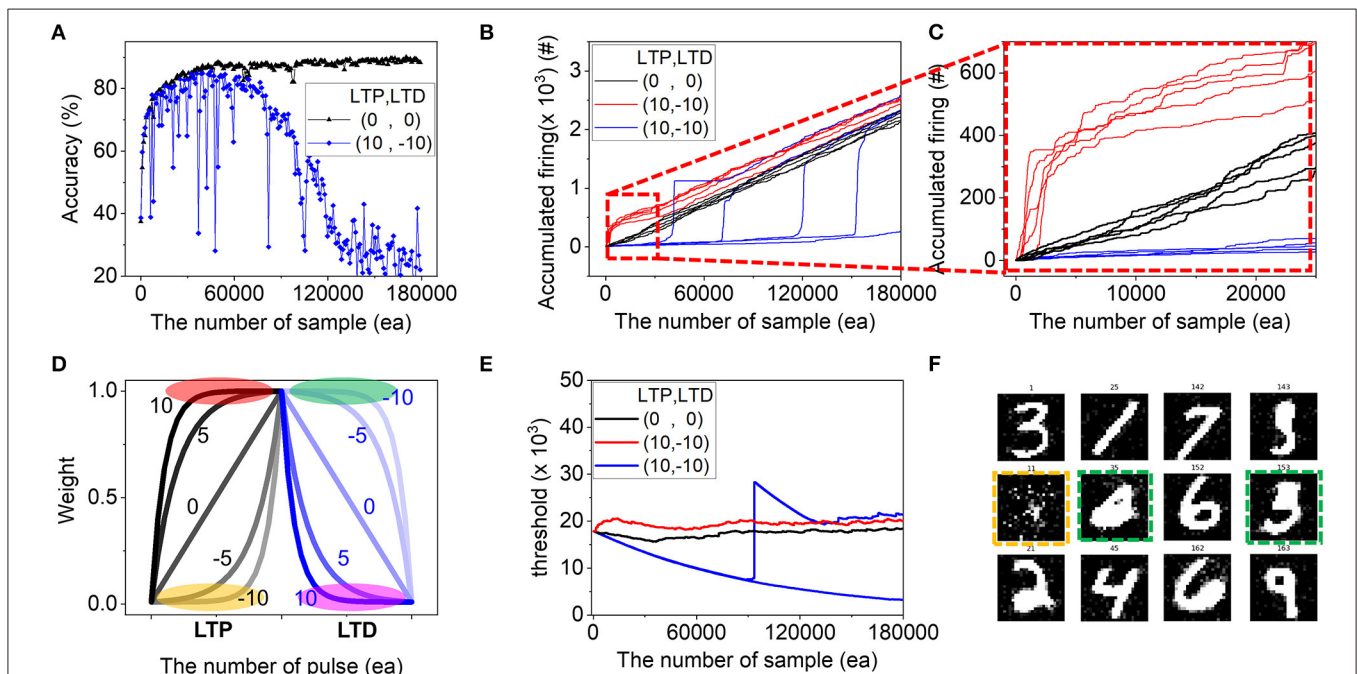
$$\Delta threshold = (f_{actual} - f_{target}) \times threshold \times \gamma \quad (4)$$

, where  $f_{actual}$  is an actual firing count and  $f_{target}$  is a predefined target firing count. The  $\gamma$  is a homeostasis factor deciding the threshold changing rate. As a result, the network can keep the firing rate almost constantly and all parameters balance well although the weights, one of the network components change at every training cycle according to the learning algorithm. However, in (10,−10), the stability breaks, and some of the neurons (red line in **Figure 5B**) fire with higher frequency than

(0,0), whereas others (blue line in **Figure 5B**) keep silent for a while and start to fire excessively from some point. This is due to a combination of strong potentiation and weak depression in (10,−10). The purpose of the potentiation process is to increase weights. Thus, the flat area in the low  $v_{LTP}$  (yellow area in **Figure 5D**) makes it hard for LTP to work properly since the weight update is negligible for many applied pulses. However, the flat area in high  $v_{LTP}$  (red area in **Figure 5D**) is already at a high level and the slow LTP process has negligible effects on the potentiation itself. Hence, it can be said that for LTP, positive  $v_{LTP}$  leads to stronger potentiation relatively than negative  $v_{LTP}$ . In contrast, positive (negative)  $v_{LTD}$  can result in a strong (weak) depression (pink and green area in **Figure 5D**). Therefore, (10,−10) is considered to make strong LTP and weak LTD, and consequently some neurons fire at a higher rate than (0,0). In the meantime, the threshold adjust function in Equation (4) is optimized for the (0,0) case and the parameters used do not work perfectly in the abnormal high firing rate of (10,−10). Thus, the network ends up failing to make a uniform distribution of neuron's firing. As a result, the firing concentrates on some neurons, and others are delayed in triggering their firing (blue



**FIGURE 4 |**  $\Delta w$  by a single LTP or LTD pulse for (A) (0,0), (B) (10,-10), (C) (-10,-10), (D) (10,10), (E) (-10,10). For the linear case (0,0),  $\Delta w$  is independent of the current weight, while  $\Delta w$  is a function of the current weight in all other non-linear cases. The insets show the corresponding LTP and LTD curves. And blue dotted line indicates  $\Delta w = 0$ . (F) The ratio of  $\Delta w$  between LTP and LTD. The ratio keeps a constant value only for (0,0), (10,-10), and (-10,10).



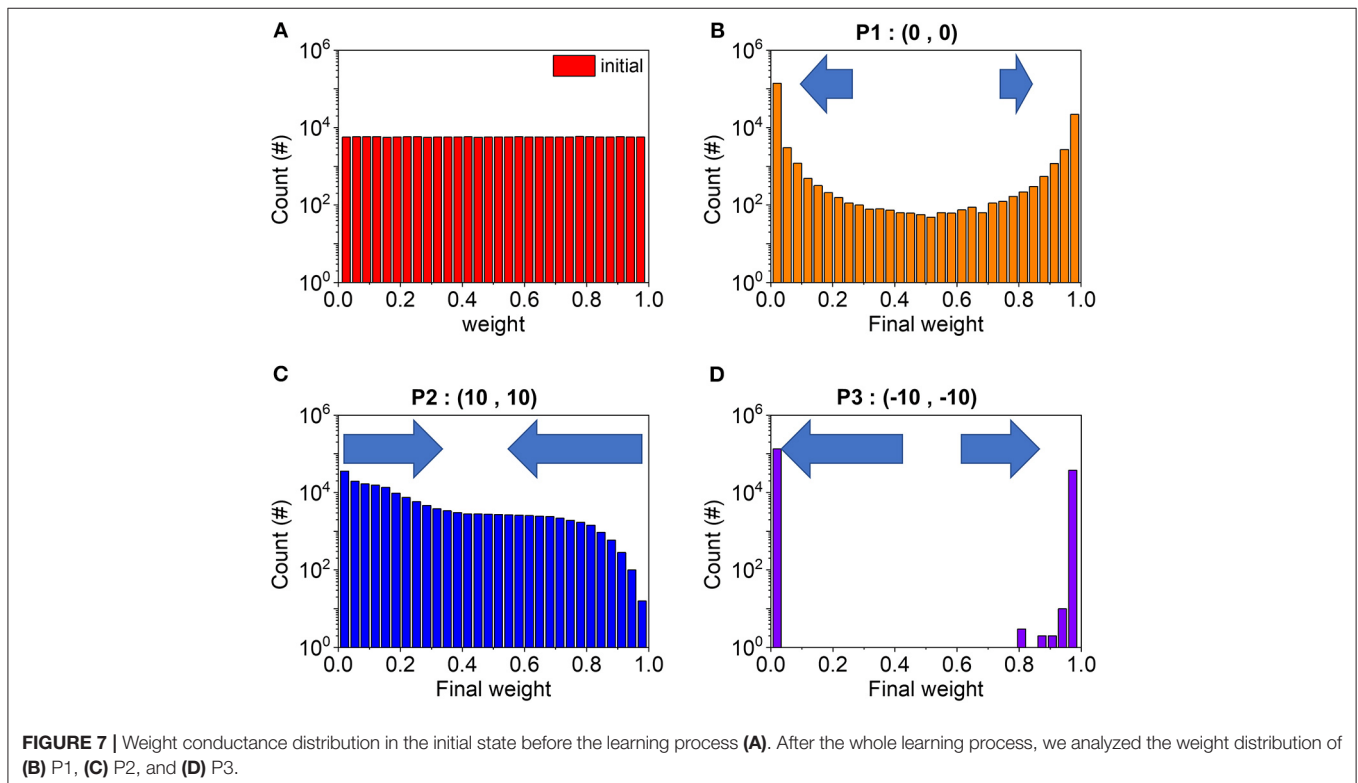
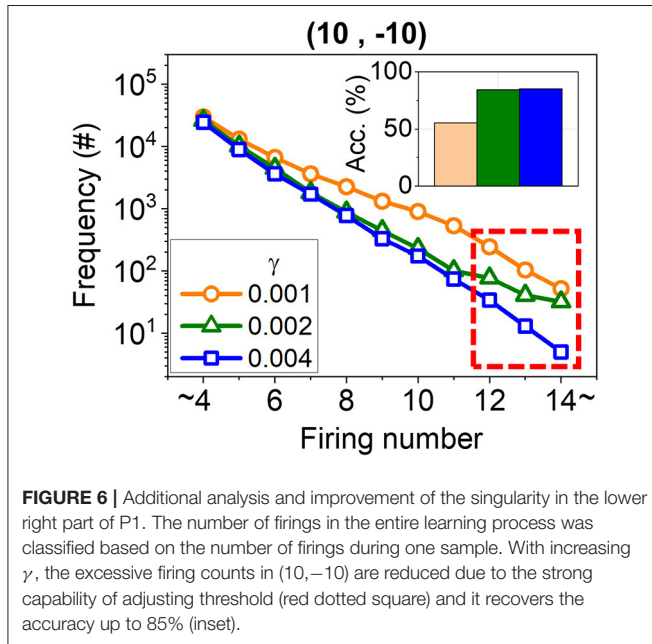
**FIGURE 5 |** Analysis of the low accuracy at the bottom-right region of P1. (A) Evolution of accuracy during training at  $(v_{LTP}, v_{LTD}) = (0,0)$  and  $(10,-10)$  cases. (B,C) Accumulated firing times was plotted during training. (D) The change of weight according to the number of pulses with different LTP and LPD non-linearity. The flat area in high and low at  $v_{LTP}$  and  $v_{LTD}$  was highlighted. The evolution of (E) threshold and some of the (F) final weight of the entire output neuron are shown, where some are over-trained with thick white digits (green box) and others are incomplete due to the delayed firing (orange box).

line in **Figures 5B,C**). The evolution of threshold and some of the final weight of the entire output neuron are shown in **Figures 5E,F**, where some are over-trained with thick white digits (green box) due to the over-firing and others are incomplete due to the delayed firing (orange box). This imbalance in homeostasis

causes low accuracy and instability during training as shown in the bottom-right area of P1 despite the high symmetry. Finally, simulation results with various homeostasis factor,  $\gamma$ , are shown in **Figure 6**. With increasing  $\gamma$ , the excessive firing counts in (10, -10) are reduced due to the strong capability of adjusting threshold (red dotted box) and it recovers the accuracy up to 85% (inset). Hence, the selection of optimized parameters considering device properties can partially alleviate the homeostasis problem in neuromorphic hardware.

## P2 and P3: Difference in Weight Distribution

Lastly, we looked into the P2 and P3 area in the accuracy map. To explain the accuracy results, weight distributions of  $576 \times 300 = 172,300$  synapses are extracted as shown in **Figure 7**. Before training, weights are randomly generated and show uniform distribution (**Figure 7A**). With running the SNN algorithm, it learns input features and makes synaptic patterns primarily composed of black and white pixels as shown in **Figure 2A**. Actually, for (0,0) representing P1, the weights after training concentrate on the edge values, black and white, and the distribution draws a U-shape (**Figure 7B**). However, for (-10,-10) representing P3, due to the weak LTP and LTD as mentioned in the previous section, weights barely get out from the edge value once they reach the boundary. Therefore, they accumulate at the edge during training, preventing the proper learning process, and in the end, the final weights show the extreme distribution in **Figure 7D**. This is the reason why P3 marks the low accuracy: stuck of the weights at the edge



values with negligible migration, disabling fine tuning of them. Contrastingly, for (10,10) representing P2, both operations, LTP and LTD, are strong together and the weights update its value more actively according to the algorithm without the stuck issue. As a result, more weights place at the middle of the weight range in the P2 case as shown in **Figure 7C**. Although P2 is expected to face the parameter imbalance mentioned above due to the asymmetry in LTP and LTD curves, the strong plasticity results in a more active learning process and recovers the balance problems by helping the network learn the best patterns with high accuracy.

## CONCLUSION

We have conducted an SNN simulation with memristor synapse models having non-linear conductance change. It implemented the three main neuron functions [LIF (Lee et al., 2019a), adjustable threshold (Woo et al., 2019), WTA(Hikawa, 2016)] that can be implemented by hardware. The network consisting of excitatory and inhibitory layers achieved over 89% of classification accuracy for MNIST dataset by using 300 output neurons. Using the same framework, 121 cases with different non-linearity factors were simulated and the performance was evaluated. We found that SNN has a strong tolerance for the device non-linearity and keeps the accuracy high for a wide range of non-linearity factor. In addition, we showed that balance in network parameters such as LTP/LTD ratio and homeostasis is very critical to maintaining high accuracy. Symmetric LTP and LTD curves help the network keep the balance due to the constant LTP and LTD ratio. It was also found that when both  $v_{LTP}$  and  $v_{LTD}$  are positive, the variability of weights is very active without stuck at the edge values because of the strong LTP and LTD process. This results in enhanced learning capability and allows high accuracy. Thus, for hardware implementation

of SNN, especially using emerging devices, the device property should be optimized to keep the network in balance with high learning ability.

## DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author.

## AUTHOR CONTRIBUTIONS

YJ directed and supported this project. TK designed the simulation and collected and analyzed the data. SH participated in the design of the simulation. All authors were involved in the discussion of the results and commented on the manuscript.

## FUNDING

This research was supported in part by the Korea Institute of Science and Technology (KIST) Open Research Program (ORP) through Grant 2E31031 and by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education NRF-2020R1A6A3A01096318 and by the National Research Foundation program through NRF-2019M3F3A1A02072175.

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fncom.2021.646125/full#supplementary-material>

## REFERENCES

- Agarwal, S., Plimpton, S. J., Hughart, D. R., Hsia, A. H., Richter, I., Cox, J. A., et al. (2016). "Resistive memory device requirements for a neural algorithm accelerator," in *2016 International Joint Conference on Neural Networks (IJCNN)* (Vancouver, BC), 929–938.
- Atzori, M., Cognolato, M., and Müller, H. (2016). Deep learning with convolutional neural networks applied to electromyography data: a resource for the classification of movements for prosthetic hands. *Front. Neurobot.* 10:9. doi: 10.3389/fnbot.2016.00009
- Brivio, S., Conti, D., Nair, M. V., Frascaroli, J., Covi, E., Ricciardi, C., et al. (2018). Extended memory lifetime in spiking neural networks employing memristive synapses with nonlinear conductance dynamics. *Nanotechnology* 30:015102. doi: 10.1088/1361-6528/aae81c
- Brunel, N., and Sergi, S. (1998). Firing frequency of leaky integrate-and-fire neurons with synaptic current dynamics. *J. Theor. Biol.* 195, 87–95. doi: 10.1006/jtbi.1998.0782
- Chandrasekaran, S., Simanjuntak, F. M., Saminathan, R., Panda, D., and Tseng, T.-Y. (2019). Improving linearity by introducing Al in HfO<sub>2</sub> as a memristor synapse device. *Nanotechnology* 30:445205. doi: 10.1088/1361-6528/ab3480
- Chang, T., Yang, Y., and Lu, W. (2013). Building neuromorphic circuits with memristive devices. *IEEE Circ. Syst. Mag.* 13, 56–73. doi: 10.1109/MCAS.2013.2256260
- Cüppers, F., Menzel, S., Bengel, C., Hardtdegen, A., Witzleben, M., von Böttger, U., et al. (2019). Exploiting the switching dynamics of HfO<sub>2</sub>-based ReRAM devices for reliable analog memristive behavior. *APL Mater* 7:091105. doi: 10.1063/1.5108654
- Demirci, M. (2015). "A survey of machine learning applications for energy-efficient resource management in cloud computing environments," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (Miami, FL), 1185–1190.
- Diehl, P. U., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9:99. doi: 10.3389/fncom.2015.00099
- Du, Z., Rubin, D. B.-D., Chen, Y., He, L., Chen, T., Zhang, L., et al. (2015). "Neuromorphic accelerators: a comparison between neuroscience and machine-learning approaches," <https://ieeexplore.ieee.org/xpl/conhome/7837567/proceeding2015> 48th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO) (Waikiki, HI), 494–507.
- Feldman, D. E. (2012). The spike-timing dependence of plasticity. *Neuron* 75, 556–571. doi: 10.1016/j.neuron.2012.08.001
- Frascaroli, J., Brivio, S., Covi, E., and Spiga, S. (2018). Evidence of soft bound behaviour in analogue memristive devices for neuromorphic computing. *Sci. Rep.* 8:7178. doi: 10.1038/s41598-018-25376-x
- Gao, P., Zhang, Y., Zhang, L., Noguchi, R., and Ahamed, T. (2019). Development of a recognition system for spraying areas from unmanned aerial vehicles using a machine learning approach. *Sensors* 19:313. doi: 10.3390/s19020313
- Ghosh-Dastidar, S., and Adeli, H. (2009). Spiking neural networks. *Int. J. Neural Syst.* 19, 295–308. doi: 10.1142/S0129065709002002



- Hassan, N., Hu, X., Jiang-Wei, L., Brigner, W. H., Akinola, O. G., Garcia-Sanchez, F., et al. (2018). Magnetic domain wall neuron with lateral inhibition. *J. Appl. Phys.* 124:152127. doi: 10.1063/1.5042452
- Hikawa, H. (2016). "Improved winner-take-all circuit for neural network based on frequency-modulated signals," in *2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS)* (Monte Carlo), 85–88.
- Imam, N., and Cleland, T. A. (2020). Rapid online learning and robust recall in a neuromorphic olfactory circuit. *Nat. Mach. Intell.* 2, 181–191. doi: 10.1038/s42256-020-0159-4
- Indiveri, G., Linares-Barranco, B., Hamilton, T. J., Schaik, A., van Etienne-Cummings, R., Delbruck, T., et al. (2011). Neuromorphic silicon neuron circuits. *Front. Neurosci.* 5:73. doi: 10.3389/fnins.2011.00073
- Jang, J.-W., Park, S., Burr, G. W., Hwang, H., and Jeong, Y.-H. (2015). Optimization of conductance change in  $\text{Pr}_{1-x}\text{Ca}_x\text{MnO}_3$ -based synaptic devices for neuromorphic systems. *IEEE Electron Device Lett.* 36, 457–459. doi: 10.1109/LED.2015.2418342
- Jeong, D. S., Kim, K. M., Kim, S., Choi, B. J., and Hwang, C. S. (2016). Memristors for energy-efficient new computing paradigms. *Adv. Electron. Mater.* 2:1600090. doi: 10.1002/aelm.201600090
- Jeong, Y., Kim, S., and Lu, W. D. (2015). Utilizing multiple state variables to improve the dynamic range of analog switching in a memristor. *Appl. Phys. Lett.* 107:173105. doi: 10.1063/1.4934818
- Jeong, Y., Zidan, M. A., and Lu, W. D. (2017). Parasitic effect analysis in memristor-array-based neuromorphic systems. *IEEE Trans. Nanotechnol.* 17, 184–193. doi: 10.1109/TNANO.2017.2784364
- Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10:12971301. doi: 10.1021/nl904092h
- Jung, Y. H., Hong, S. K., Wang, H. S., Han, J. H., Pham, T. X., Park, H., et al. (2020). Flexible piezoelectric acoustic sensors and machine learning for speech processing. *Adv. Mater.* 32:1904020. doi: 10.1002/adma.202070259
- Lee, D., Kwak, M., Moon, K., Choi, W., Park, J., Yoo, J., et al. (2019a). Various Threshold Switching Devices for Integrate and Fire Neuron Applications. *Adv. Electron Mater.* 1800866. doi: 10.1002/aelm.201800866
- Lee, S., Kim, C.-H., Oh, S., Park, B.-G., and Lee, J.-H. (2019b). Unsupervised online learning with multiple postsynaptic neurons based on spike-timing-dependent plasticity using a thin-film transistor-type nor flash memory array. *J. Nanosci. Nanotechnol.* 19, 6050–6054. doi: 10.1166/jnn.2019.17025
- Li, C., Belkin, D., Li, Y., Yan, P., Hu, M., Ge, N., et al. (2018). Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. *Nat. Commun.* 9:2385. doi: 10.1038/s41467-018-04484-2
- Liu, H., Wei, M., and Chen, Y. (2018). Optimization of non-linear conductance modulation based on metal oxide memristors. *Nanotechnol. Rev.* 7, 443–468. doi: 10.1515/ntrev-2018-0045
- Liu, L., Wang, Y., and Chi, W. (2020). Image recognition technology based on machine learning. *IEEE Access* 1–1. doi: 10.1109/ACCESS.2020.3021590
- Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642
- Nandakumar, S. R., Gallo, M. L., Boybat, I., Rajendran, B., Sebastian, A., and Eleftheriou, E. (2018). A phase-change memory model for neuromorphic computing. *J. Appl. Phys.* 124:152135. doi: 10.1063/1.5042408
- Ponulak, F., and Kasinski, A. (2011). Introduction to spiking neural networks: information processing, learning, and applications. *Acta Neurobiol. Exp.* 71, 409–433.
- Querlioz, D., Bichler, O., Dollfus, P., and Gamrat, C. (2013). Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Trans. Nanotechnol.* 12, 288–295. doi: 10.1109/TNANO.2013.2250995
- Wade, J. J., McDaid, L. J., Santos, J. A., and Sayers, H. M. (2008). "SWAT: an unsupervised SNN training algorithm for classification problems," in *https://ieeexplore.ieee.org/xpl/conhome/4625775/* proceed in 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (Hong Kong), 2648–2655.
- Wang, Z., Joshi, S., Savel'ev, S., Song, W., Midya, R., Li, Y., et al. (2018). Fully memristive neural networks for pattern classification with unsupervised learning. *Nat. Electron.* 1, 137–145. doi: 10.1038/s41928-018-0023-2
- Woo, S. Y., Choi, K.-B., Kim, J., Kang, W.-M., Kim, C.-H., Seo, Y.-T., et al. (2019). Implementation of homeostasis functionality in neuron circuit using double-gate device for spiking neural network. *Solid State Electron.* 165:107741. doi: 10.1016/j.sse.2019.107741
- Yang, S., Shin, J., Kim, T., Moon, K.-W., Kim, J., Jang, G., et al. (2021). Integrated neuromorphic computing networks by artificial spin synapses and spin neurons. *NPG Asia Mater.* 13:11. doi: 10.1038/s41427-021-00282-3
- Yao, P., Wu, H., Gao, B., Tang, J., Zhang, Q., Zhang, W., et al. (2020). Fully hardware-implemented memristor convolutional neural network. *Nature* 577, 641–646. doi: 10.1038/s41586-020-1942-4
- Yi, G.-S., Wang, J., Tsang, K.-M., Wei, X.-L., and Deng, B. (2015). Input-output relation and energy efficiency in the neuron with different spike threshold dynamics. *Front. Comput. Neurosci.* 9:62. doi: 10.3389/fncom.2015.00062

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Kim, Hu, Kim, Kwak, Park, Lee, Kim, Park and Jeong. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.