frontiers in Neuroscience

ORIGINAL RESEARCH
published: 19 February 2021
doi: 10.3389/fnins.2021.601109

Efficient Spike-Driven Learning With Dendritic Event-Based Processing

Shuangming Yang[1], Tian Gao[1], Jiang Wang[1], Bin Deng[1], Benjamin Lansdell[2] and Bernabe Linares-Barranco[3]*

[1] School of Electrical and Information Engineering, Tianjin University, Tianjin, China, [2] Department of Bioengineering, University of Pennsylvania, Philadelphia, PA, United States, [3] Microelectronics Institute of Seville, Seville, Spain

A critical challenge in neuromorphic computing is to present computationally efficient algorithms of learning. When implementing gradient-based learning, error information must be routed through the network, such that each neuron knows its contribution to output, and thus how to adjust its weight. This is known as the credit assignment problem. Exactly implementing a solution like backpropagation involves weight sharing, which requires additional bandwidth and computations in a neuromorphic system. Instead, models of learning from neuroscience can provide inspiration for how to communicate error information efficiently, without weight sharing. Here we present a novel dendritic event-based processing (DEP) algorithm, using a two-compartment leaky integrate-and-fire neuron with partially segregated dendrites that effectively solves the credit assignment problem. In order to optimize the proposed algorithm, a dynamic fixed-point representation method and piecewise linear approximation approach are presented, while the synaptic events are binarized during learning. The presented optimization makes the proposed DEP algorithm very suitable for implementation in digital or mixed-signal neuromorphic hardware. The experimental results show that spiking representations can rapidly learn, achieving high performance by using the proposed DEP algorithm. We find the learning capability is affected by the degree of dendritic segregation, and the form of synaptic feedback connections. This study provides a bridge between the biological learning and neuromorphic learning, and is meaningful for the real-time applications in the field of artificial intelligence.

Keywords: spiking neural network, credit assignment, dendritic learning, neuromorphic, spike-driven learning

OPEN ACCESS

Edited by:
Angelo Arleo,
Centre National de la Recherche Scientifique (CNRS), France

Reviewed by:
Charles Augustine,
Intel, United States
Sio Hoi Ieng,
Université Pierre et Marie Curie, France

****Correspondence:***
Bin Deng
dengbin@tju.edu.cn

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *31 August 2020*
Accepted: *21 January 2021*
Published: *19 February 2021*

Citation:
Yang S, Gao T, Wang J, Deng B, Lansdell B and Linares-Barranco B (2021) Efficient Spike-Driven Learning With Dendritic Event-Based Processing. Front. Neurosci. 15:601109. doi: 10.3389/fnins.2021.601109

INTRODUCTION

Learning requires assigning credit to each neuron for its contribution to the final output (Bengio et al., 2015; Lillicrap et al., 2016). How a neuron determines its contribution is known as the credit assignment problem. In particular, the training of deep neural networks is based on error back-propagation, which uses a feedback pathway to transmit information to calculate error signals in the hidden layers. However, neurophysiological studies demonstrate that the conventional error

back-propagation algorithm is not biologically plausible. One problem is known as weight transport: backpropagation utilizes a feedback structure with the exact same weights as the feed-forward pathway to communicate gradients (Liao et al., 2016). This symmetric feedback structure has not been proven to exist in biological neural circuit. Several studies have presented solutions to modify or approximate the backpropagation algorithm in a more biologically plausible manner (Lee et al., 2015; Scellier and Bengio, 2017; Lansdell and Kording, 2019; Lansdell et al., 2019). In fact, active channels in dendrites can drive different forms of spiking activities (Schmolesky et al., 2002; Larkum, 2013). A potential solution is thus to segregate signals into dendritic compartments, so that the credit signals can be kept separate from other ongoing computation (Richards and Lillicrap, 2019). Recent work shows how spiking neural networks can implement feedback structures that allow efficient solving of the credit assignment problem by dendritic computation (Urbanczik and Senn, 2014; Wilmes et al., 2016; Bono and Clopath, 2017; Guerguiev et al., 2017). Further, other work has shown that even feedback systems that crudely approximate the true feedback weights can solve some learning tasks (Zenke and Ganguli, 2018; Lee et al., 2020). Together these works show that the credit assignment problem can be largely solved by biologically plausible neural systems.

An ongoing challenge in neuromorphic computing is to present general and computationally efficient algorithms of deep learning. Previous works have shown how neuromorphic approaches for deep learning can be more efficient compared to Von Neumann architecture (Esser et al., 2015; Indiveri et al., 2015; Neftci et al., 2017). However, these systems have yet to be fully realized. By design, learning in neuromorphic hardware operates under similar constraints to learning in biological neural networks. The credit assignment problem, and the problem of weight transport also manifest in this setting: neuromorphic learning systems that do not require weight transport enjoy less data transfer between components. In this way, biologically plausible approaches to deep learning can also be used to make neuromorphic computing more efficient. Previous neuromorphic systems have been presented for high-performance brain-inspired computation, providing tests of biological learning models and real-time applications (Qiao et al., 2015; Yang et al., 2015, 2018, 2020, 2021).

Recent proposals for solutions to the credit assignment problem have not been considered in neuromorphic computing. Here we present a novel dendritic event-based processing (DEP) algorithm to facilitate the efficient implementation of the credit-assignment task on neuromorphic hardware. The presented DEP algorithm is inspired by the primary sensory areas of the neocortex, providing the segregation of feed-forward and feedback information required to compute local error signals and to solve the credit assignment problem. In the DEP algorithm, a binarization method and a dynamic fixed-point solution are presented for the efficient implementation of deep learning. The paper is organized as follows: section "Introduction" describes the preliminaries of this study, including neuromorphic computing and the spiking neural network (SNN) model. Learning with stochastic gradient descent (SGD) in spiking neural networks is introduced and explained in section "Materials and Methods." Section "Results" presents the experimental results. And finally, the discussions and conclusions are proposed in sections "Discussion" and "Conclusion," respectively.

MATERIALS AND METHODS

Learning With Dendrites in Event-Driven Manner

Learning needs neurons to receive signals to assign the credit for behavior. Since the behavioral impact in early network layers is based on downstream synaptic connections, credit assignment in multi-layer networks is challenging. Previous solutions in artificial intelligence use the backpropagation of error algorithm, but this is unrealistic in the neural systems. Rather than requiring weight transport, current biologically plausible solutions to the credit assignment problem use segregated feed-forward and feedback signals (Lee et al., 2015; Lillicrap et al., 2016). In fact, the cortico-cortical feedback signals to pyramidal neurons can transmit the necessary error information. These works show how the circuitry needed to integrate error information may exist within each neuron. The idea is that both feed-forward sensory information in the neocortex and the higher-order cortico-cortical feedback are received by different dendritic compartments, including basal and apical dendrites (Spratling, 2002). In a pyramidal cell, distal apical dendrites are distant from the soma, and communicate with the soma based on active propagation using the apical dendritic shaft, driven predominantly by voltage-gated calcium channels (Katharina et al., 2016). Further, there exist dynamics of plateau potentials that generate prolonged upswings in the membrane potential. These are based on the nonlinear dynamics of voltage-gated calcium current, and drive bursting at the soma (Larkum et al., 1999). The plateau potentials of the apical dendritic activities can induce learning in pyramidal neurons *in vivo* (Bittner, 2015).

Inspired by these phenomena, a previous study has proposed a learning algorithm with segregated dendrites (Guerguiev et al., 2017). Based on this work, an efficient learning algorithm for neuromorphic learning is presented in this study. The idea is that the basal dendritic compartment is coupled to the soma for processing bottom-up sensory information, and the apical dendritic compartment is used to process top-down feedback information to calculate credit assignment and induce learning using plateau potentials. The basic computing unit we use on the large-scale conductance-based spiking neural network (LaCSNN) system is based on the integrate-and-fire (IF) principle. As shown in **Figure 1**, the simple spiking behaviors of the IF neurons can be triggered by excitatory input spikes. The new state of the neural membrane potential with an input arriving is determined by the last updating time and the previous state. Thus, the event-driven neuron only updates when an input spike is received. Then the membrane potential decay after the last update is retroactively calculated and applied. The synaptic weight is then used to contribute to the resulting membrane potential. A spike event is emitted when the membrane potential exceeds a spike threshold, and then the neural activity is reset and

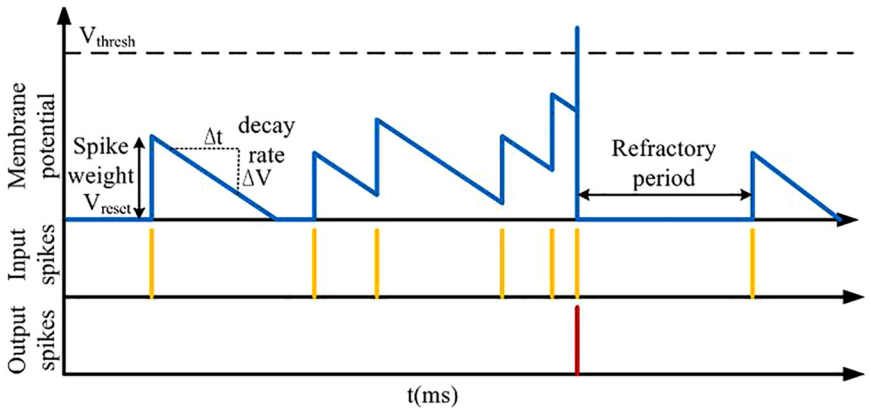

FIGURE 1 | Event-driven neural computing, showing the process of synaptic weights and time affecting the neural membrane potential and the refractory period.

mutual inhibition with coupled neurons is realized. Finally the membrane potential and spiking event are written to memory to store the network state of the next update of neural activity.

Network Architecture With SGD Algorithm

The network diagram utilizes the SNN model in the previous study by Guerguiev et al. (2017) as shown in **Figure 2**, which consists of an input layer with m = 784 neurons, a hidden layer with n = 500 neurons, and an output layer with p = 10 neurons. Since our primary interests are in the realization of neuromorphic networks, the proposed model is restricted to discrete systems based on the Euler method, where N is the time step for discretization. This way of representation is popular in the hardware implementation of spiking neural networks because of its feasibility of implementation and routing. Poisson spiking neurons are used in the input layer, whose firing rate is determined by the intensity of image pixels ranging from 0 to Φ_{max}. In the hidden layer, neurons are modeled using three functional compartments, which are basal dendrites, apical dendrites and soma. The membrane potential of the ith neuron in the hidden layer is updated as follows:

$$\tau\frac{V_i^0(N+1)-V_i^0(N)}{\Delta T} = -V_i^0(N) + \frac{g_b}{g_l}\left(V_i^{0b}(N) - V_i^0(N)\right) + \frac{g_a}{g_l}\left(V_i^{0a}(N) - V_i^0(N)\right) \quad (1)$$

where g_l, g_b, and g_a stand for the leak conductance, the basal dendrites conductance, and the apical dendrite conductance, and ΔT is the integration step. The superscript "0," "a," and "b" represent hidden layer, basal dendrite and apical dendrite. The parameter $\tau = C_m/g_l$, is a time constant, where C_m represents the membrane capacitance. The variables V^0, V^{0a}, and V^{0b} represent the membrane potentials of soma, apical dendrite and basal dendrite, respectively. The dendritic compartments are defined as weighted sums for the ith hidden layer neuron as follows:

$$\begin{cases} V_i^{0b}(N) = \sum_{j=1}^{m} W_{ij}^0 s_j^{input}(N) + b_i^0 \\ V_i^{0a}(N) = \sum_{j=1}^{p} Y_{ij} s_j^1(N) \end{cases} \quad (2)$$

where $W_{ij}{}^0$ and Y_{ij} are synaptic weights in the input layer and feedback synapses, respectively. The constant $b_i{}^0$ is defined as a bias term, and s^{input} and s^1 are the filtered spiking activities in the input layer and output layer, respectively. The variable s^{input} is calculated based on the following equations as

$$s_j^{input}(t) = \sum_k \kappa\left(t - t_{jk}^{input}\right) \quad (3)$$

where $t_{jk}{}^{input}$ represents the kth spiking time of the input neuron j, and the response kernel is calculated as

$$\kappa(t) = \left(e^{-t/\tau_L} - e^{-t/\tau_s}\right)\Theta(t) \Big/ (\tau_L - \tau_s) \quad (4)$$

where τ_L and τ_s are long and short time constants, and Θ is the Heaviside step function. The filtered spike trains at apical synapses s^1 is modeled based on the same method. The spiking activities of somatic compartments are based on Poisson processes, whose firing rates are based on a non-linear sigmoid function $\sigma(.)$ for the ith hidden layer neuron as follows:

$$\Phi_i^0(N) = \phi_{\max}\sigma\left(V_i^0(N)\right) = \phi_{\max}\frac{1}{1+e^{-V_i^0(N)}} \quad (5)$$

where Φ_{max} represents the maximum firing rates of neurons.

Plateau Potentials and Weight Updates

Based on the learning algorithm of Guerguiev et al., two phases are alternated to train the network: the forward and target phases as shown in **Figure 3**. In the forward phase $I_i(t) = 0$, while it induces any given neuron i to spike at maximum firing rate or be silent according to the category of the current input image when

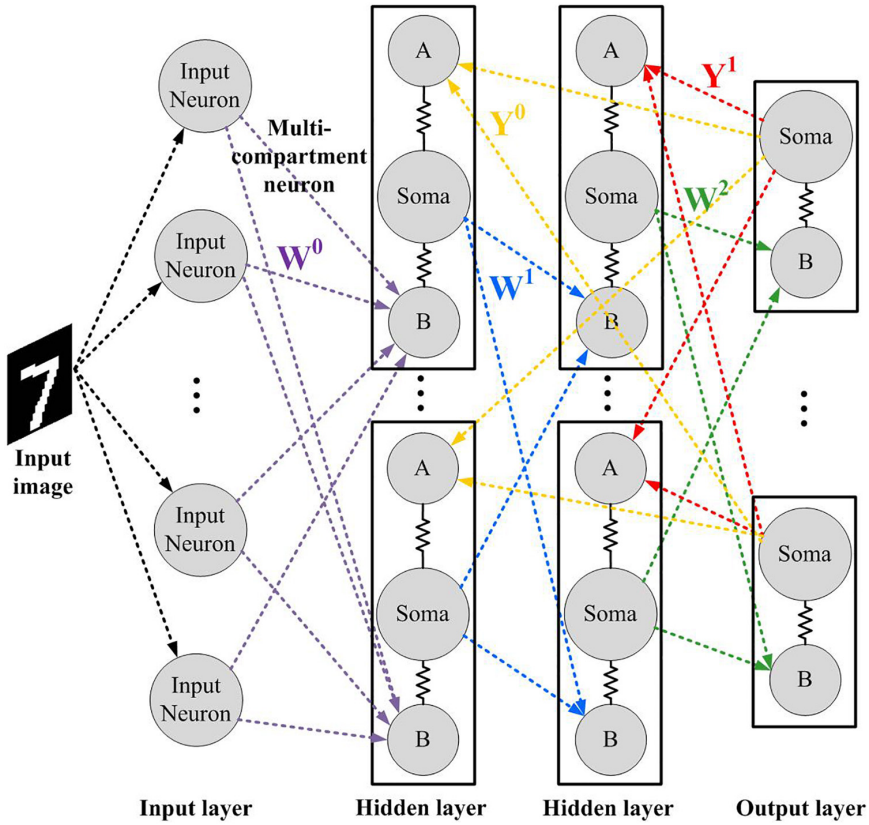

FIGURE 2 | The network architecture in this study. Matrices *Y* and *W* represent the feedback and feed-forward synaptic weight matrix, respectively.

the network undergoes target phase. At the end of the forward phase and the target phase, the set of plateau potentials α_t and α_f are calculated, respectively.

At the end of each phase, plateau potentials are calculated for apical dendrites of hidden layer neurons, which are defined as follows

$$\begin{cases} \tau \frac{V_i^1(N+1)-V_i^1(N)}{\Delta T} = -V_i^1(N) + I_i(N) \\ \qquad\qquad + \frac{g_d}{g_l}\left(V_i^{1b}(N) - V_i^1(N)\right) \\ V_i^{1b}(N) = \sum_{i=1}^{m} W_{ij}^1 s_j^0(N) + b_i^1 \end{cases} \tag{6}$$

where t_1 and t_2 represent the end times of the forward and target phases, respectively. Δt_s = 30 ms represents the settling time for the membrane potentials, and Δt_1 and Δt_2 are formulated as follows

$$\begin{cases} \Delta t_1 = t_1 - (t_0 + \Delta t_s) \\ \Delta t_2 = t_2 - (t_1 + \Delta t_s) \end{cases}. \tag{7}$$

The temporal intervals between plateaus are sampled based on an inverse Gaussian distribution randomly. Although the system computes in phases, the specific length of the phases is not vital, provided there has been a long enough time to integrate the input currents.

Learning With Feedback Driven Plateau Potentials

During the forward phase, an image is presented to the input layer without teaching current at the output layer between time t_0 to t_1. At t_1 a plateau potential is computed in the hidden layer neurons and the target phase begins. During the target phase the image is also presented into the input layer that also receives teaching current, forcing the correct neuron in the output layer to its maximum firing rate while others are silent. At time t_2 another set of plateau potentials in the hidden layers are computed. Plateau potentials for the end of both the forward and the target phases are calculated as follows

$$\begin{cases} \alpha_i^f = \sigma\left(\frac{1}{\Delta t_1}\int_{t_1-\Delta t_1}^{t_1} V_i^{0a}(N)\,dt\right) \\ \alpha_i^t = \sigma\left(\frac{1}{\Delta t_2}\int_{t_2-\Delta t_2}^{t_2} V_i^{0a}(N)\,dt\right) \end{cases} \tag{8}$$

where Δt_s represents a time delay of the network dynamics before integrating the plateau, and $\Delta t_i = t_i - (t_{i-1} + \Delta t_s)$. The superscript "*t*" and "*f*" represent target and forward phases, respectively.

The basal dendrites in the hidden layer update the synaptic weights W_0 based on the minimization of the loss function as follows

$$L^0 = \left|\left|\phi^{0^*} - \phi_{\max}\sigma\left(\bar{v}^0 f\right)\right|\right|_2^2. \tag{9}$$

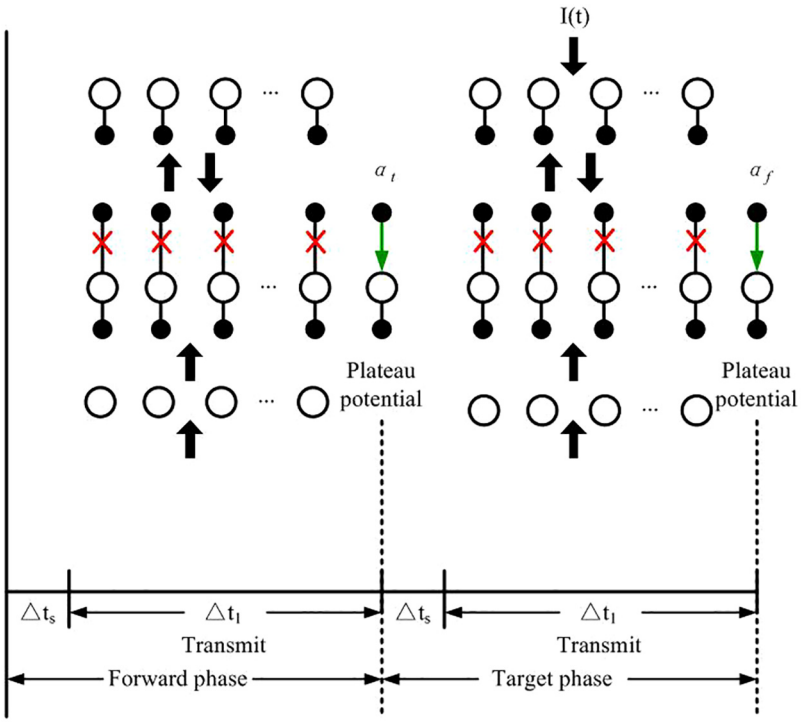

FIGURE 3 | Network computing phases for learning proposed by Guerguiev et al. The green arrows represent the signal transmission from apical dendrite to soma, and red crosses stand for the disconnection between apical dendrite and somatic compartment. The black arrows represent the transmission of spike signals between layers.

And the target firing rate is defined as

$$\phi_i^{0*} = \bar{\Phi}_i^{0^f} + \alpha_i^t - \alpha_i^f \tag{10}$$

where the variable and are plateau potentials in the forward and target phases. It should be noted that as long as neural units calculate averages after the network has reached a steady state, and the firing rates of the neurons are in the linear region of the sigmoid function, then we have the following equation for the hidden layer as:

$$\phi_{\max}\sigma\left(\bar{V}^{0f}\right) \approx \phi_{\max}\overline{\sigma\left(V^0\right)}^f = \overline{\phi^0}^f \tag{11}$$

Then the formulation can be obtained as

$$L^0 \approx \left|\left|\alpha^t - \alpha^f\right|\right|_2^2. \tag{12}$$

And the formulation can be described as follows

$$\begin{cases} \frac{\partial L^0}{\partial W^0} \approx -k_b\left(\alpha^t - \alpha^f\right)\phi_{\max}\sigma'\left(\bar{V}^{0f}\right)^\circ \bar{s}^{inputf} \\ \frac{\partial L^0}{\partial b^0} \approx -k_b\left(\alpha^t - \alpha^f\right)\phi_{\max}\sigma'\left(\bar{V}^{0f}\right) \end{cases} \tag{13}$$

where the constant k_b is given as

$$k_b = g_b/(g_l + g_b + g_a). \tag{14}$$

In this study, Φ^{0*} is treated as a fixed state for the hidden layer neurons to learn. The synaptic weights of basal dendrites are updated to descend the approximation of the gradient as follows

$$\begin{cases} W^0 \to W^0 - \eta^0 P^0 \frac{\partial L^0}{\partial W^0} \\ b^0 \to b^0 - \eta^0 P^0 \frac{\partial L^0}{\partial b^0} \end{cases}. \tag{15}$$

In the target phase the activity is also fixed and no derivatives are used for the membrane potentials and firing rates. The feedback weights are held fixed.

Piecewise Linear Approximation (PWL) for Digital Neuromorphic Computing

Here we simplify the above model for efficient use in neuromorphic architectures. In order to avoid the complicated computation induced by nonlinear functions, the PWL approach is used in this study. Both the functionsσ(x) and σ'(x) are modified based on the PWL method, which can be formulated as follows

$$f_{PWL} = \begin{cases} a_1 x + b_1, & \text{when } x \le s_1 \\ a_2 x + b_2, & \text{when } s_1 < x \le s_2 \\ \ldots \\ a_i x + b_i, & \text{when } x > s_{i-1} \end{cases} \tag{16}$$

where a_i and b_i are the slope and intercept of the modified PWL function f_{owl}, respectively (i = 1, 2,..., n). Since the range of the segment points are constrained, an exhaustive search algorithm is used in the determination of the PWL functions.

TABLE 1 | Parameter values of the PWL methods.

σ(x)	A	b	Condition
$i = 1$	0.0078125	0.05	$x \leqq -3.4$
$i = 2$	0.0625	0.24	$-3.4 < x \leqq -1.3$
$i = 3$	0.25	0.5	$-1.3 < x \leqq 1.3$
$i = 4$	0.0625	0.76	$1.3 < x \leqq 3.4$
$i = 5$	0.0078125	0.95	$3.4 < x$
$i = 6$	0	0.9999	$\sigma(x) \leqq 0$
$i = 7$	0	0.0001	$\sigma(x) \geqq 1$
σ'(x)	**A**	**b**	**Condition**
$i = 1$	0.0078125	0.05	$x \leqq -3.2$
$i = 2$	0.03125	0.15	$-3.2 < x \leqq -2$
$i = 3$	0.0625	0.25	$-2 < x \leqq 0$
$i = 4$	−0.0625	0.25	$0 < x \leqq 2$
$i = 5$	−0.03125	0.15	$2 < x \leqq 3.2$
$i = 6$	−0.0078125	0.05	$x > 3.2$
$i = 7$	0	0.0001	$\sigma'(x) \leqq 0$

The determination of the coefficient values are based on an error evaluation criterion as follows

$$CF_{RE} = \frac{1}{n}\sqrt{\sum_{i=1}^{n} \frac{\left(f_{ori}\left(i\right) - f_{PWL}\left(i\right)\right)^2}{f_{ori}\left(i\right)^2}} \tag{17}$$

where n represents the total sampling points, and f_{ori} represents the original function. If the modified function cannot meet the accuracy requirement represented by CF_{RE}, its segment number will be added by 1 until it can be guaranteed. Since the multiplication operation is replaced by "adder" and "shifter" in the proposed study, the coefficient value a_i in the PWL functions should be a power of 2 (for example: 1, 2, 4 or 0.5, 0.25, etc.). The parameter values of the PWL methods are listed in **Table 1**. The PWL functions are depicted in **Figure 4**.

Binarization for Filtered Spike Trains

The digital neuromorphic algorithm requires less multiplication operations. Therefore, in this study we use the Otsu's thresholding method to binarize the filtered spike trains, which can iterate all possible threshold values and compute the expansion measure of each pixel level of the threshold (Otsu, 1978). Therefore, each pixel will fall in either foreground or background. Firstly, separate all the pixels into two clusters based on the threshold as follows

$$\begin{cases} q_1\left(t\right) = \sum_{i=1}^{t} p\left(i\right) \\ q_2\left(t\right) = \sum_{i=t+1}^{L} p\left(i\right) \end{cases} \tag{18}$$

where p represents the image histogram. Secondly, the mean of each cluster is calculated by the formulation as follows

$$\begin{cases} \mu_1\left(t\right) = \sum_{j=1}^{t} \frac{i \cdot p\left(i\right)}{q_1\left(t\right)} \\ \mu_2\left(t\right) = \sum_{j=t+1}^{L} \frac{i \cdot p\left(i\right)}{q_2\left(t\right)} \end{cases} \tag{19}$$

Thirdly, calculate the individual class variance as follows

$$\begin{cases} \lambda_1^2\left(t\right) = \sum_{i=1}^{t} \left[i - \mu_1\left(t\right)\right]^2 \frac{p\left(i\right)}{q_1\left(t\right)} \\ \lambda_2^2\left(t\right) = \sum_{i=t+1}^{L} \left[i - \mu_2\left(t\right)\right]^2 \frac{p\left(i\right)}{q_2\left(t\right)} \end{cases} \tag{20}$$

Fourthly, square the difference between the means formulated as follows

$$\begin{aligned} \lambda_b^2\left(t\right) &= \lambda^2 - \lambda_w^2\left(t\right) \\ &= q_1\left(t\right)\left[1 - q_1\left(t\right)\right]\left[\mu_1\left(t\right) - \mu_2\left(t\right)\right]^2 \end{aligned} \tag{21}$$

where λ_b, λ, and λ_w represent between-class variance, total class variance and within-class variance, respectively. Finally, the formulation can be maximized and the solution is t that is maximizing $\lambda_b{}^2(t)$.

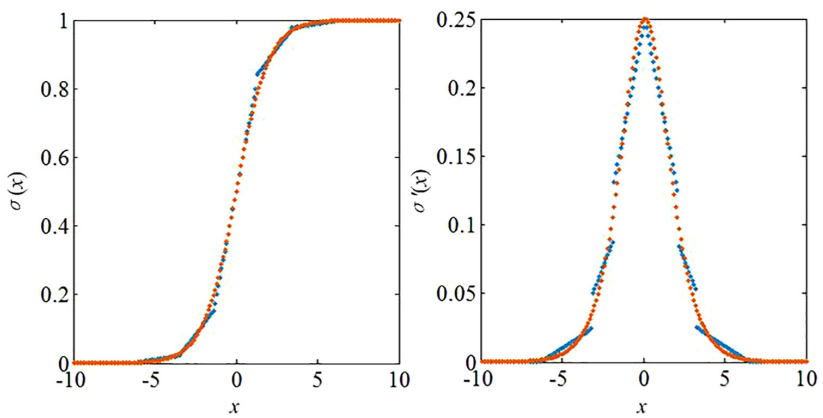

FIGURE 4 | PWL functions in the proposed algorithms.

Considerations for Training With Low Bitwidth Weights

Neuromorphic hardware is largely made out of arithmetic elements and memories. Multipliers are the most space and power hungry arithmetic elements of the digital neuromorphic implementation. The realization of a deep neural network is mainly dependent on matrix multiplications. The key arithmetic operation is the multiply-accumulate operation. The reduction of the precision of the multipliers, especially for the weight matrix, is vital for the efficient realization of deep neural networks. Recent researches have focused on the reduction of model size and computational complexity by using low bitwidth weights of neural networks (Courbariaux et al., 2014). Other neuromorphic hardware systems implement bistable synapses based on a 1-bit weight resolution, which is shown to be sufficient for memory formation (Bill, 2010). However, the models do not only use spike timings, but also use additional hardware resources to read the postsynaptic membrane potential (Sjöström et al., 2001). Therefore, this study trains the proposed DEP algorithm using dynamic fixed point representation. In dynamic fixed point, each number is represented as follows

$$(-1)^{s} \cdot 2^{-FL} \sum_{i=0}^{B-2} 2^{i} \cdot x_i \tag{22}$$

where B represents the bit-width, s the sign bit, FL is the fractional length, and x the mantissa bits.

The proposed algorithm is presented in **Figure 5**. In the pseudo code, the synaptic weight matrix W is the input of the algorithm. Total_bit represents the total bit width of the fixed-point number, and IF_bit is the integer bitwidth. The fractional bitwidth is represented by LF_bit. The integer and fractional parts are represented by W_IF and W_LF. The binary integer and fractional parts are represented by W_IF_bit and W_LF_bit, respectively. The symbol bit is represented by W_s, and R_max defines the fault-tolerant ratio. The error rate refers to the difference between the binary number and the original decimal number divided by the original decimal number. If the error rate exceeds the defined fault-tolerant rate, a specialized process will be used for the binary number. Since the large error occurs in the situation when the considered number is close to 0, this number will be set to 0 if the error rate exceeds R_max. The term W is an a^*b synaptic weight matrix to be trained. The first loop is in the line 2. This loop is in the line 2, which is the row loop of the matrix. The second loop is in the line 3, which is the column loop of the matrix. There are two judgments in the proposed algorithm. The first judgment is to determine the symbol bit. If it is negative, then the symbol is 1. If it is positive, then the symbol is 0. The second judgment is to determine the positive and negative when the binary number is converted to decimal number. If the sign bit is 1, it is negative. And it is positive when the sign bit is 0. The third judgment is to consider the error rate between the newly converted number and the original number. If the error rate exceeds the fault-tolerant ratio, the newly converted number will be replaced by 0 for efficient calculation on neuromorphic systems. Finally the updated synaptic weight matrix W_new is output by the processing of the proposed algorithm. By using the proposed algorithm, the memory usage on hardware can be optimized and the energy efficiency of neuromorphic systems can be further improved.

Algorithm: Training with dynamic fixed point representation

```
Input: W, Output: W_new;
W is a*b matrix.
For i=1:a
    For j=1:b
        If W(i,j)<0
            W_s=1;
        Else
            W_s=0;
        End
        W_IF_bit=W_IF→bit;
        IF_bit=length(W_IF_bit);
        LF_bit= Total_bit- IF_bit-1;
        W_LF_bit= (W_LF ,LF_bit)→bit;
        W_new(i,j)= W_IF_bit→dec+ W_LF_bit→dec;
        If W_s=1
            W_new(i,j)=-W_new(i,j);
        End
        If (W_new(i,j)- W(i,j))/ W(i,j)>R_max
            W_new(i,j)=0;
        end
    End
End
```

FIGURE 5 | Pseudocode of the algorithm for training with dynamic fixed point representation.

RESULTS

To demonstrate the effectiveness of the proposed learning algorithm, the standard Modified National Institute of Standards and Technology database (MNIST) benchmark is employed. The MNIST dataset contains 70,000 28 × 28 images of handwritten digits. The image number in the training and testing sets are 60,000 and 10,000, respectively. The dataset is divided into 10 categories for 10 integers 0–9, and each image has an associated label. We trained the networks with no hidden layer, with one hidden layer and two hidden layers on the 60,000 MNIST training images for 10 epochs, and tested the classification accuracy using the 10,000 image test set. As shown in **Figure 6A**, the network with no hidden layer has poor classification performance of 62.1%. In contrast, the three-layer network with hidden layer has an accuracy of 95.1% by the 10th epoch. The proposed network can take advantage of the multi-layer architecture to enhance the learning performance, which is the critical characteristics of deep learning (Bengio and LeCun, 2007). Another critical characteristic of deep learning is the capability to generate representations, which obtains task-related information and ignores irrelevant

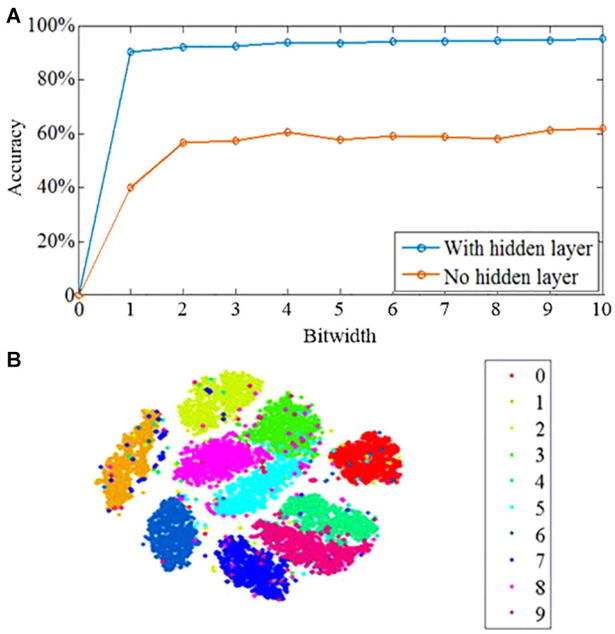

FIGURE 6 | Learning performance of the DEP algorithm. **(A)** The learning accuracy across 10 epochs of training. **(B)** Results of t-SNE dimensionality reduction applied to the activity patterns of the hidden layer after 10 epochs of training.

sensory details (LeCun et al., 2015; Mnih et al., 2015). The t-distributed stochastic neighbor embedding algorithm (t-SNE) is used to investigate the information abstraction of the proposed algorithm. The t-SNE algorithm can reduce the dimensionality of data with the preservation of local structure and nonlinear manifolds in high-dimensional space. It is a powerful approach to visualize the structure of high-dimensional data (Maaten and Hinton, 2008). The t-SNE algorithm is applied to the hidden layer, which shows that the categories are better segregated with only a small amount of splitting or merging of category clusters as shown in **Figure 6B**. Therefore, the proposed algorithm has the capability of learning the developing representations in the hidden layer, in which the categories are quite distinct. It reveals that the proposed algorithm can be applied in a deep learning framework. In addition, the proposed algorithm relies on the phenomenon of feedback alignment, in which the feed-forward system comes to align with the feedback weights so that a useful error signal is provided.

The proposed DEP learning algorithm in a network with one hidden layer trained on permutation invariant MNIST is explored, although it can be generalized to other datasets in theory. Rather than seeking for the optimized classification performance, the equivalent non-spiking neural networks trained by standard BP and random BP are compared with the proposed algorithm, with the parameters tuned to obtain the highest classification accuracy in the current classification task. Weight updates are conducted during each digit input into the spiking network, which is different from the batch gradient descent that performs weight updates once per the entire dataset. As shown in **Figure 7**, the DEP algorithm requires fewer iterations of the dataset to obtain the peak classification performance in comparison with the two alternative methods. The reason is that the spiking neural network with DEP algorithm can be updated multiple times during each input, which results in faster convergence of learning. In addition, for the equivalent computational resources, online learning with gradient descent strategy has the capability to deal with more data samples and requires less on-chip memory for implementation (Bottou and Cun, 2004). Therefore, for the same number of calculation operations per unit time, online gradient-descent-based learning converges faster than batch learning. Since potential applications of neuromorphic hardware is with real-time streaming data, it is essential for the online learning with DEP algorithm.

In order to further demonstrate the learning performance of the proposed DEP algorithm, a comparison between the proposed DEP algorithm with two layers and the SNN with point LIF neuron model is presented. As shown in **Figure 8A**, the learning accuracy of the SNN model with dendrites, i.e., the proposed DEP algorithm, is higher than the conventional SNN with point neuron model. Besides, we further apply the DEP algorithm in the feature detection tasks to see whether the proposed algorithm could also learn feature detection maps from continuous sensory streams. Previous study has shown that SNN models can defect features from background activities (Masquelier et al., 2008). In order to provide a good benchmark for the proposed DEP algorithm on the feature detection task, the ability of the DEP algorithm is examined for feature detection tasks. In this task, there are eight categories, and each category represents on direction, including 0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, and 157.5°. Each image consists of 729 (27 × 27) pixels. Besides, 10% pixels are randomly

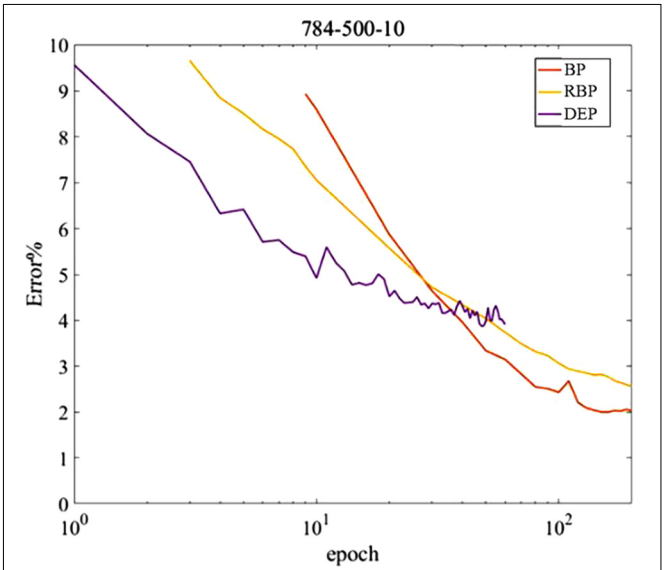

FIGURE 7 | MNIST classification error based on spiking neural network with DEP learning rule and fully connected artificial neural networks with backpropagation (BP) and random backpropagation(RBP) learning rules.

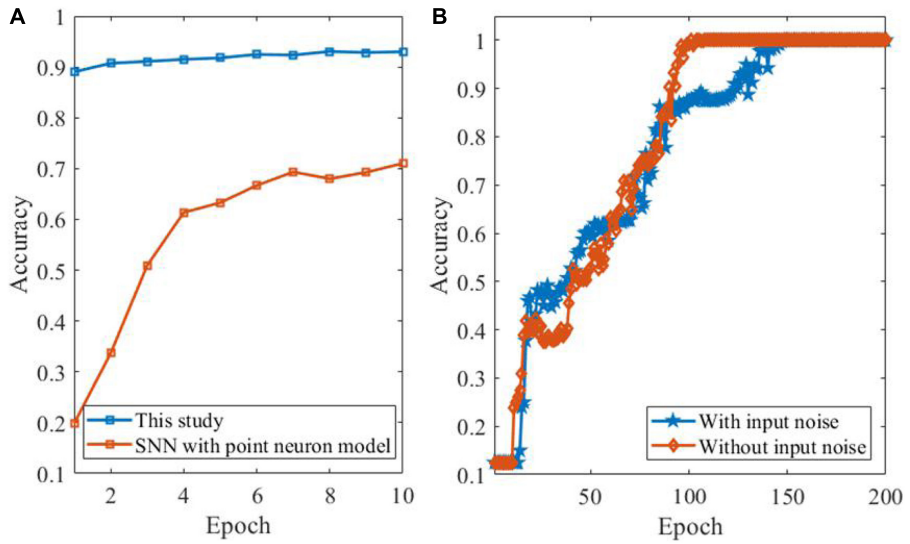

FIGURE 8 | Learning performance by comparison and based on feature detection task. **(A)** Comparison between the proposed DEP algorithm with two layers and SNN with point LIF neuron model. **(B)** Performance evaluation of feature detection by DEP algorithm with and without input noise, respectively.

selected to add Gaussian noise to make the data set with input noise. **Figure 8B** shows the learning performance of the DEP algorithm. It reveals that the DEP algorithm can successfully detect feature patterns contaminated by background noise using spike-based framework.

As shown in **Figure 9**, learning on neuromorphic system can be energy efficient by using the proposed DEP algorithm, because only active connections in the network induce synaptic operations (SynOps) operation. In order to show the learning efficiency, the number of multiply-accumulate (MAC) operations using the BP algorithm is compared with SynOps number with the proposed algorithm. This advantage is critical and promising for neuromorphic computing because SynOps in a dedicated neuromorphic system use much less power than MAC operations on a GPU platform. The learning accuracy of the proposed algorithm increases quickly but the final accuracy is lower than an ANN.

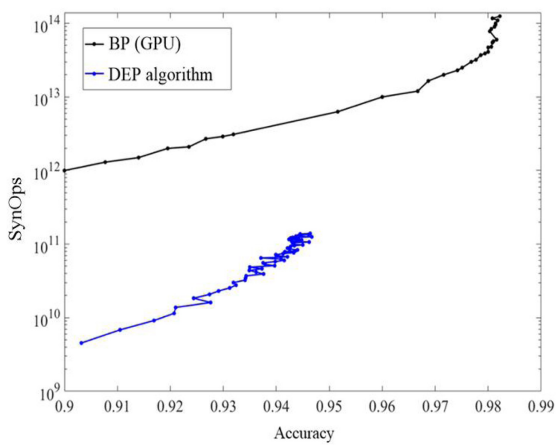

FIGURE 9 | The comparisons of accuracy and SynOps between the proposed DEP algorithm and BP algorithm on graphical processing unit (GPU).

As shown in **Figure 10**, the response of the proposed DEP algorithm after stimulus onset is one synaptic time constant. It leads to 11% error and improves as the spikes number of the neurons in the output layer increases. Classification using the first spike induced less than 20 k SynOps events, most of which exist between the input and hidden layer. In the state-of-the-art neuromorphic system, the energy consumption of a synaptic operation is around 20 pJ (Merolla et al., 2014; Qiao et al., 2015). On such neuromorphic system, single spike classification based on the proposed network can potentially induce 400 pJ, which is superior to the state-of-the-art work in digital neuromorphic hardware (≈2 μJ) at this accuracy (Esser et al., 2016) and potentially 50,000 more efficient than current GPU technology. In addition, an estimation of the power consumption during training by using BP and the proposed DEP algorithm is also presented according to **Figure 9**. It reveals that about 10^{11} SynOps is cost by the end of 60 epochs, therefore about 33 mJ is cost in an epoch during training. Previous study has demonstrated that 1,000–5,000 mJ will be cost by BP algorithm on conventional GPU platform (Rodrigues et al., 2018). Therefore, there is a 96.7–99.3% reduction for the power consumption by the proposed DEP algorithm during training. The reasons for the low energy cost can be divided into three aspects. Firstly, the segregated dendrite can generate a plateau potential within 50–60 ms, which determines the training time of the proposed network. The training time can be thereby reduced in this way, which can cut down the number of spikes with the decreasing of the training time for each image. Secondly, the conventional BP algorithm induces a trend to make neurons spike with maximum firing rate, and induces synchronization

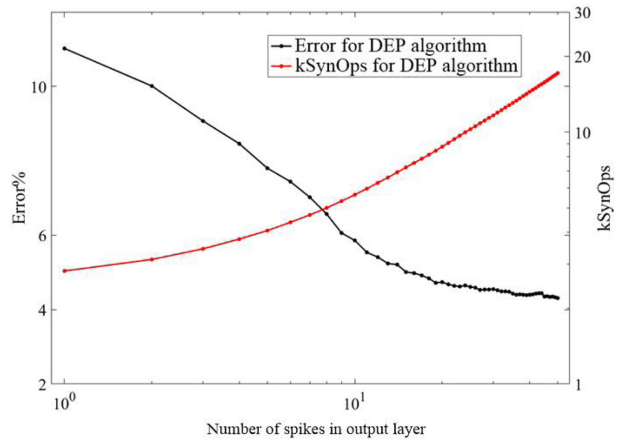

FIGURE 10 | Classification error in the proposed DEP network as a function of the number of spikes in the output layer, and total number of synaptic operations incurred up to each output spike.

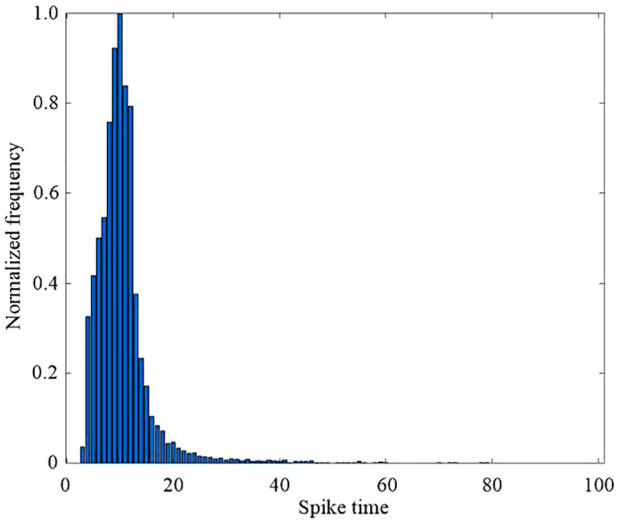

FIGURE 11 | Histograms of spike times in the output layer spike across the 10,000 test set images.

within layers. This means a larger number of spikes. Thirdly, the communication between layers in the proposed algorithm uses a Poisson filter, and Φ_{max} is set to be 0.2. These results suggest that the proposed DEP learning algorithm can take full use of the spiking dynamics, with the learning accuracy comparable to the spiking network that is trained specifically for single spike recognition in previous study (Mostafa, 2017).

Figure 11 shows the distribution of spike times in the output layer, which is the times at which the proposed SNN makes a decision for all the 10,000 test set images. The proposed SNN with DEP algorithm makes a decision after most of the hidden layer neurons have spiked. The network is thus able to make more accurate and robust decisions about the input images, based on the plateau potentials generated by the dendrites in the proposed DEP algorithm.

As shown in **Figure 12**, **30** neurons in the hidden layer are selected randomly to explore the selectivity for 10 categories of MNIST data set. The negative log probability for each of the 30 neurons to spike for each of the 10 categories is explored, which means the negative log probability for a neuron to participate in the classification of a specified category. Probability is calculated from the response of the SNN to the 10,000 test digits. It reveals that some neurons are highly selective, while most of the neurons are more broadly tuned. Some of the neurons are mostly silent, but all the neurons in the SNN model contribute to at least one category of classification with the 10,000 test digits. In other word, neurons are typically broadly tuned and contribute to the classification of more than one categories.

We further investigate the necessary bit widths of the fixed-point and dynamic fixed-point, respectively. The bit width of the integer part using the fixed-point calculation is set to 8 to avoid the overflow problem during computation. In contrast, the dynamic fixed-point is not required to determine the bit width of either integer or fractional part. As shown in **Figure 13**, the fixed-point representation of the fractional part requires 14 bits to obtain a satisfied learning performance that exceeds 90%. Therefore, the satisfied total bit width for fixed-point representation is 22 bit (8 bit for integer part and 14 bit for fractional part). The dynamic fixed-point representation just needs 16 bits to realize high-performance learning. Therefore, the dynamic fixed-point representation in the proposed algorithm provides an efficient approach to reduce the computational hardware resource cost and power consumption for neuromorphic computing.

Figure 14 shows the digital neuromorphic architecture at the top level, which contains an input layer, a hidden layer with five physical neural processors, and an output layer with 10 physical neural processors. The input layer and hidden layer are all implemented to use time-multiplexing. The global counter processors the time-multiplexed input neurons and hidden layer neurons sequentially. The FSM module represents the finite-state machine which controls the timing procedure of the whole neuromorphic system. Three parts are contained in the neuron processor in the hidden layer, which are apical dendrite unit, soma unit and basal dendrite unit. The neuron processor in the output layer consists of two parts that are apical dendrite unit and soma unit. The input of the teaching current $I(t)$ is also mastered by the FSM. The green arrows represent the synaptic connections with learning mechanisms, and black arrows describe the invariant synaptic coupling.

The detailed description of the FSM is shown in **Figure 15A**. There are eight states in the FSM diagram, including idle, first time delay, forward phase, first plateau potential (PP) computation, second time delay, target phase, second PP computation and weight updating. By using the FSM controller, the digital neuromorphic system can operate in high performance with definite timing sequence. **Figure 15B** depicts the internal architecture of the time-multiplexed system. It consists of a physical input neuron, two physical hidden neurons, a global

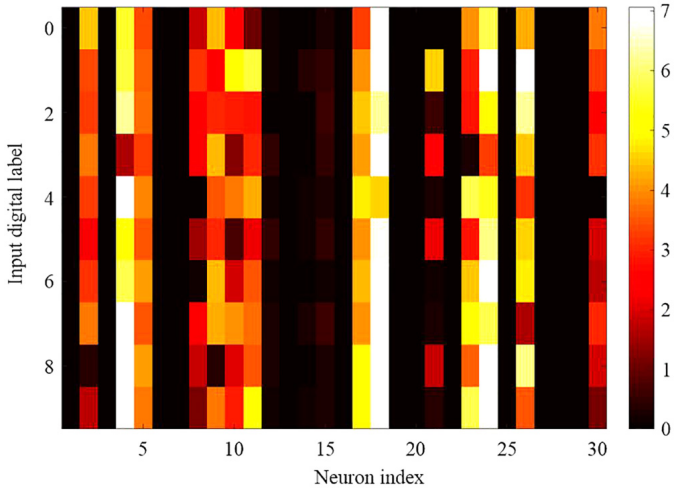

FIGURE 12 | Selectivity and tuning properties of 30 randomly selected hidden neurons in the proposed SNN network with DEP algorithm. It is plotted by heat map with color called YlOrRd, whose color gradually changes across yellow, orange, and red.

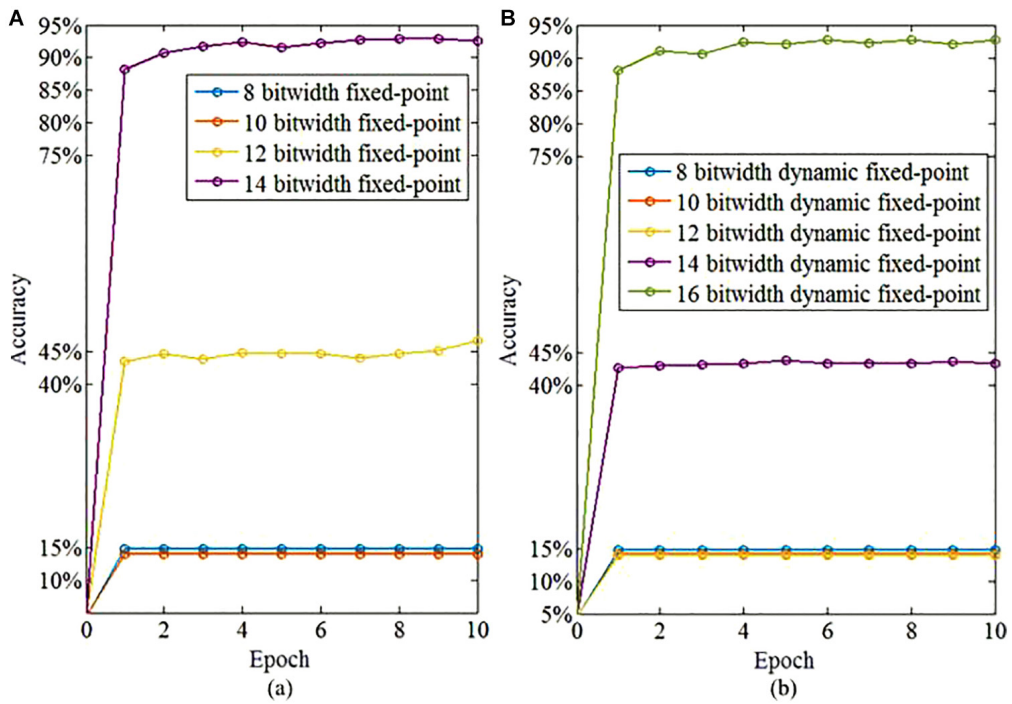

FIGURE 13 | The learning accuracy based on fixed-point and dynamic fixed-point representations. **(A)** The learning accuracy based on fixed-point representation of fractional part with different bitwidth. **(B)** The learning accuracy based on dynamic fixed-point representation with different bitwidth.

counter, and two weight buffers for each physical hidden neuron. The global counter processes the time-multiplexed physical input and hidden neurons sequentially. The weight buffers store the synaptic weights of the physical neurons. The input digit signals remains available until all the time-multiplexed physical neurons finish their computation. We can also employ the pipeline architecture, by which the maximum operating frequency of the neuromorphic system can be further enhanced.

DISCUSSION

This study presents a multi-layer feed-forward network architecture using segregated dendrites and the corresponding two-phase learning scheme. Specifically, a piecewise linear approximation and a dynamic fixed-point representation are first introduced in the dendritic learning framework for cost and energy efficient neuromorphic computing. It relies on the

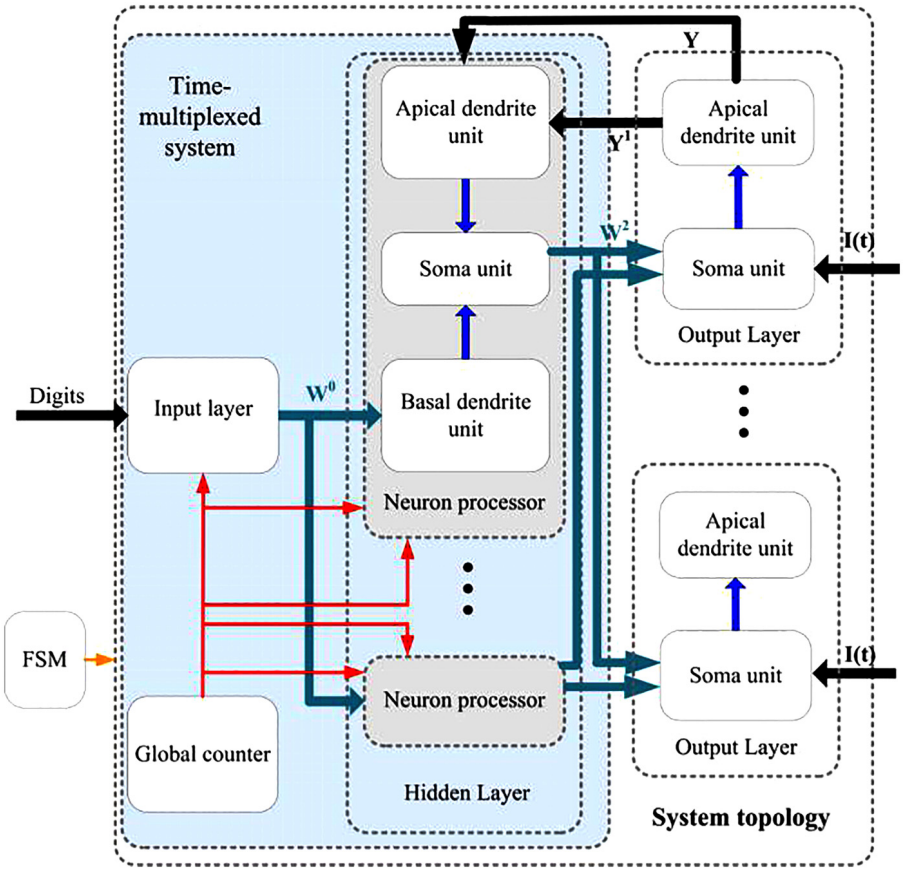

FIGURE 14 | Top-level entity for the neuromorphic architecture of the proposed learning algorithm.

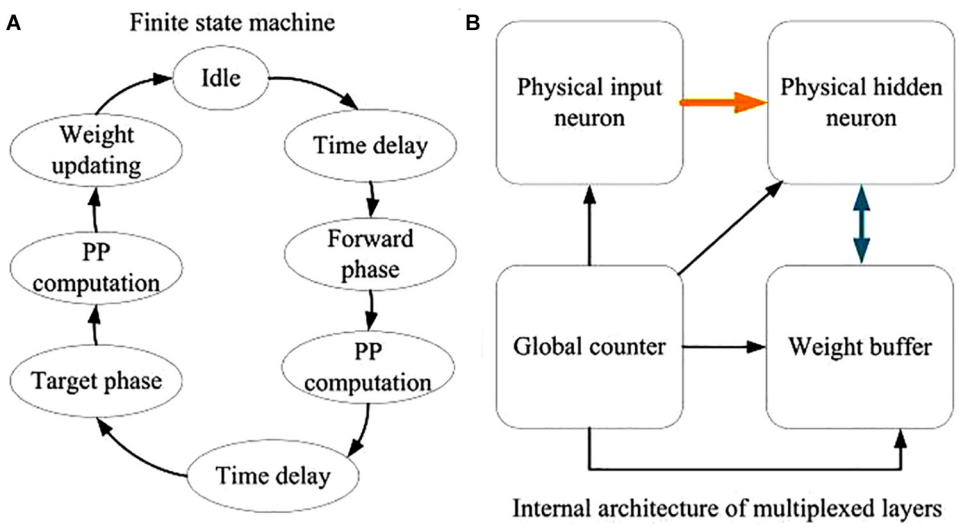

FIGURE 15 | Descriptions of the digital neuromorphic architecture. **(A)** The FSM diagram. **(B)** The internal architecture of the time-multiplexed system.

feedback alignment phenomenon, in which the feed-forward weights are aligned with the feedback weights to provide useful error signals for learning. The model is optimized for the efficient neuromorphic realization by using the PWL approximation, as well as the binarization for synaptic events. A dynamic fixed-point representation technique is further presented to optimize the proposed DEP algorithm. It reveals that the proposed algorithm with hidden layers can induce higher

learning performance, which means that it contains the deep learning capability. In addition, the energy efficient property is proven by comparison with the conventional artificial neural network with BP algorithm. The reasons for this result are likely due to two reasons. First, the gradient descent algorithm suffers from the local optimization problem. When the local optimization is realized, the global optimization cannot be obtained. Spikes emanating from error-coding neurons will be so sparse toward the end of the training that it will prevent the successful adjustments of the weight. The low learning rate will aggravate this problem. A scheduled adjustment of the error neuron sensitivity may solve this problem. Second, the proposed algorithm has not fully utilized the nonlinear dynamics of the neural dendrite. The dynamics of the dendritic compartment have the capability of predicting the dynamics, which may help to improve the performance when considering the dendritic prediction feature. These two methods of modifications, as well as more complicated learning rules, such as momentum or learning rate decay, are left for future work. The output layer neurons spike after the spiking activities of the most neurons in the hidden layer, thus induce a more accurate and robust classification results. The broadly tuned property of the neurons in the hidden layer of the proposed SNN shows that the proposed DEP algorithm can engage each neuron to participate in the classification task. In addition, it shows the superior performance by using the proposed dynamic fixed-point representation by comparing it with the traditional fixed-point computation, which shows that the proposed method can reduce the hardware resource cost considerably. Therefore, our study demonstrates a biologically plausible learning algorithm in a neuromorphic architecture, and realizes the efficient learning by using the DEP approach. In summarize, the proposed DEP algorithm has four aspects of advantages. Firstly, the proposed DEP algorithm cost less SynOps number in comparison with the conventional BP algorithm as shown in **Figure 8**. It means less power consumption can be realized on neuromorphic hardware. Secondly, faster learning speed can be achieved by the DEP algorithm shown in **Figure 7**, which is meaningful for on-chip online learning. Thirdly, the solution of credit assignment by dendrites is a vital mechanism for learning in human brain. Therefore, the proposed DEP algorithm is more biologically plausible, which is also a significant ambition of neuromorphic computing. Fourthly, the proposed DEP algorithm is more useful for the online learning with network architecture using more than one layer. As shown in Figure, single point neuron model is not suitable for learning with gradient descent when the network layer number increasing to two.

In the field of neuromorphic computing, neuromorphic systems with on-line learning ability provide a platform to develop brain-inspired learning algorithms, which strive to emulate in digital or analog technologies human brain properties. Online learning requires to be realized based on the input of asynchronous and event-based sequential data flow. Since neuromorphic computing supports continual and lifelong learning naturally, this study presents a SNN model that can deal with the asynchronous event-based spatio-temporal information, which is applicable for neuromorphic systems directly. It provides a novel view for neuromorphic online learning and continual learning, which is meaningful to bridge the gap between neuroscience and machine intelligence. Previous studies have presented a number of neuromorphic systems equipped with synaptic plasticity for general-purpose sensorimotor processors and reinforcement learning (Neftci, 2013; Qiao, 2015; Davies et al., 2018). However, current neuromorphic computing ignores the learning capability to further improve the deep learning performance. Inspired by other neuromorphic studies, more low-power and high-speed techniques can be considered in the future work to obtain a better learning effect.

Previous studies have proposed new algorithms, including attention-gated reinforcement learning (AGREL) and attention-gated memory tagging (AuGMEnT) learning rules, explaining the mechanism of the reinforcement learning optimization in a biologically realistic manner using synapses in deep networks (Roelfsema and Ooyen, 2005; Rombouts et al., 2015). The feedback coupling strength is proportional to the feed-forward strength in these models, which means the learning principles are computationally equivalent to the error back-propagation. It indicates the human brain can solve the credit-assignment problem in a manner that is equivalent to deep learning. However, AGREL algorithm uses the top-down probabilistic model to compute rather than the description and representation of learning from the neural dynamics point of view. There is also no bottom-to-top modeling using spiking neurons in AuGMEnT algorithm. Thus, these two algorithms cannot be employed in neuromorphic computing. Interestingly, we can combine these two algorithms with the presented DEP algorithm to improve the learning performance further.

Efficient learning to solve the credit assignment problem is helpful for the performance improvement of deep learning. This study presents the DEP algorithm for neuromorphic learning, which is meaningful for the communities of both neuromorphic engineering and deep learning. Recently, neuromorphic computing has wide applications. Neuromorphic vision sensors capture the features of biological retina, which has changed the landscape of computer vision in both industry and academia (Chen et al., 2019; Zhou et al., 2019). Although neuromorphic systems with deep learning capability are still in research phases, the development of neuromorphic computing is calling for more biologically realistic processing strategies. Looking forward, with such systems with learning ability, the bridges between machine and biological learning can translate into adaptive and powerful embedded computing systems for a wide category of applications, such as object recognition, neuro-robotic control, and machine learning.

CONCLUSION

This paper presented a biologically meaningful DEP algorithm with dynamic fixed-point representation, as well as its digital neuromorphic architecture on LaCSNN. The PWL approximation method and the binarization approach for synaptic events are used in the proposed algorithm for the optimization of efficient implementation. Experimental results show that the learning performance of the proposed DEP algorithm can be improved by adding a hidden layer, which

shows the deep learning capability of DEP. Different levels of dendrite segregation will influence the learning accuracy of the network, and the manners of the synaptic feedback connections also play vital roles in the learning performance. By using the fixed-point representation in this work, the hardware resource cost can be cut down by reducing the bit width of the computational elements. This study provides a bridge between the biological learning and neuromorphic learning, which can be used in the applications including object recognition, neuro-robotic control, and machine learning.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This datacan be found here: MNIST http://yann.lecun.com/exdb/mnist/.

AUTHOR CONTRIBUTIONS

SY, TG, and BL-B developed the theoretical approach for DEP algorithm with spiking neurons. TG implemented the source code. JW and BL revised the manuscript and made critical suggestions on this work. SY wrote the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This study was funded partly by the National Natural Science Foundation of China with grant numbers (Grant Nos. 62071324 and 62006170) and partly by China Postdoctoral Science Foundation (Grant No. 2020M680885).

REFERENCES

Bengio, Y., and LeCun, Y. (2007). Scaling learning algorithms towards AI. *Large Scale Kernel Mach.* 34, 1–41.

Bengio, Y., Lee, D. H., Bornschein, J., Mesnard, T., and Lin, Z. (2015). Towards biologically plausible deep learning. *arXiv* [Preprint]. arXiv:1502.04156

Bill, J. (2010). Compensating inhomogeneities of neuromorphic VLSI devices via short-term synaptic plasticity. *Front. Comput. Neurosci.* 4:129. doi: 10.3389/fncom.2010.00129

Bittner, K. C. (2015). Conjunctive input processing drives feature selectivity in hippocampal CA1 neurons. *Nat. Neurosci.* 18:1133. doi: 10.1038/nn.4062

Bono, J., and Clopath, C. (2017). Modeling somatic and dendritic spike mediated plasticity at the single neuron and network level. *Nat. Commun.* 8:706.

Bottou, L., and Cun, Y. L. (2004). "Large scale online learning," in *Proceedings of the Advances in Neural Information Processing Systems*, 217–224.

Chen, G., Cao, H., Ye, C., Zhang, Z., and Knoll, A. (2019). Multi-cue event information fusion for pedestrian detection with neuromorphic vision sensors. *Front. Neurorobot.* 13:10. doi: 10.3389/fnbot.2019.00010

Courbariaux, M., Bengio, Y., and David, J. P. (2014). Training deep neural networks with low precision multiplications. *arXiv* [Preprint]. arXiv:1412.7024

Davies, M., Srinvasa, N., and Lin, T. H. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/mm.2018.112130359

Esser, S. K., Appuswamy, R., and Merolla, P. (2015). Backpropagation for energy-efficient neuromorphic computing. *Adv. Neural Inf. Process. Systems* 28, 1117–1125.

Esser, S. K., Merolla, P. A., Arthur, J. V., and Cassidy, A. S. (2016). Convolutional networks for fast, energy efficient neuromorphic computing. *Proc. Natl. Acad. Sci. U.S.A.* 113, 11441–11446.

Guerguiev, J., Lillicrap, T. P., and Richards, B. A. (2017). Towards deep learning with segregated dendrites. *ELife* 6:e22901.

Indiveri, G., Corradi, F., and Qiao, N. (2015). "Neuromorphic architectures for spiking deep neural networks," in *Proceedings of the 2015 IEEE International Electron Devices Meeting (IEDM),* Washington, DC, 4–2.

Katharina, A., Henning, S., and Susanne, S. (2016). Inhibition as a binary switch for excitatory plasticity in pyramidal neurons. *PLoS Comput. Biol.* 12:e1004768. doi: 10.1371/journal.pcbi.1004768

Lansdell, B. J., and Kording, K. P. (2019). Spiking allows neurons to estimate their causal effect. *bioRxiv* [Preprint]. doi: 10.1101/253351

Lansdell, B. J., Prakash, P. R., and Kording, K. P. (2019). Learning to solve the credit assignment problem. *arXiv* [Preprint]. arXiv:1906.00889

Larkum, M. (2013). A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends Neurosci.* 36, 141–151. doi: 10.1016/j.tins.2012.11.006

Larkum, M. E., Zhu, J. J., and Sakmann, B. (1999). A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature* 398, 338–341. doi: 10.1038/18686

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444.

Lee, D. H., Zhang, S., Fischer, A., and Bengio, Y. (2015). "Difference target propagation," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (New York, NY: Springer International Publishing), 498–515.

Lee, J., Zhang, R., Zhang, W., Liu, Y., and Li, P. (2020). Spike-train level direct feedback alignment: sidestepping backpropagation for on-chip training of spiking neural nets. *Front. Neurosci.* 14:143. doi: 10.3389/fnins.2020.00143

Liao, Q., Leibo, J. Z., and Poggio, T. (2016). How important is weight symmetry in backpropagation. *arXiv* [Preprint]. arXiv: 1510.05067

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nat. Commun.* 7, 1–10. doi: 10.1016/j.artint.2018.03.003

Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. *J. Mach. Learn Res.* 9, 2579–2605.

Masquelier, T., Guyonneau, R., and Thorpe, S. J. (2008). Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. *PLoS One* 3:e1377. doi: 10.1371/journal.pone.0001377

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., and Cassidy, A. S. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642

Mnih, V., Kavukcuoglu, K., and Silver, D. (2015). Human-level control through deep reinforcement learning. *Nature* 518:529.

Mostafa, H. (2017). Supervised learning based on temporal coding in spiking neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 3227–3235.

Neftci, E. (2013). Synthesizing cognition in neuromorphic electronic systems. *Proc. Natl. Acad. Sci. U.S.A.* 110, 3468–3476.

Neftci, E. O., Augustine, C., and Paul, S. (2017). Event-driven random back-propagation: enabling neuromorphic deep learning machines. *Front. Neurosci.* 11:324. doi: 10.3389/fnins.2017.00324

Otsu, N. (1978). A threshold selection method from gray-scale histogram. *IEEE Trans. Syst. Man Cybern.* 8, 62–66. doi: 10.1109/tsmc.1979.4310076

Qiao, N. (2015). A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses. *Front. Neurosci.* 9:141. doi: 10.3389/fnins.2015.00141

Qiao, N., Ning, H., and Corradi, F. (2015). A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses. *Front. Neurosci.* 9:141. doi: 10.3389/fnins.2015.00141

Richards, B. A., and Lillicrap, T. P. (2019). Dendritic solutions to the credit assignment problem. *Curr. Opin. Neurobiol.* 54, 28–36. doi: 10.1016/j.conb.2018.08.003

Rodrigues, C. F., Riley, G., and Luján, M. (2018). "SyNERGY: an energy measurement and prediction framework for convolutional neural networks

on Jetson TX1[C]," in *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2018*, Turin, 375–382.

Roelfsema, P. R., and Ooyen, A. (2005). Attention-gated reinforcement learning of internal representations for classification. *Neural Comput.* 17, 2176–2214. doi: 10.1162/0899766054615699

Rombouts, J. O., Bohte, S. M., and Roelfsema, P. R. (2015). How attention can create synaptic tags for the learning of working memories in sequential tasks. *PLoS Comput. Biol.* 11:e1004060. doi: 10.1371/journal.pcbi.1004060

Scellier, B., and Bengio, Y. (2017). Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Front. Comput. Neurosci.* 11:24. doi: 10.3389/fncom.2017.00024

Schmolesky, M. T., Weber, J. T., Zeeuw, C. I. D., and Hansel, C. (2002). The making of a complex spike: ionic composition and plasticity. *Ann. N. Y. Acad. Sci.* 978, 359–390. doi: 10.1111/j.1749-6632.2002.tb07581.x

Sjöström, P. J., Turrigiano, G. G., and Nelson, S. B. (2001). Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron* 32, 1149–1164. doi: 10.1016/s0896-6273(01)00542-6

Spratling, M. W. (2002). Cortical region interactions and the functional role of apical dendrites. *Behav. Cogn. Neurosci. Rev.* 1, 219–228. doi: 10.1177/1534582302001003003

Urbanczik, R., and Senn, W. (2014). Learning by the dendritic prediction of somatic spiking. *Neuron* 81, 521–528. doi: 10.1016/j.neuron.2013.11.030

Wilmes, K. A., Sprekeler, H., and Schreiber, S. (2016). Inhibition as a binary switch for excitatory plasticity in pyramidal neurons. *PLoS Comput. Biol.* 12:e1004768. doi: 10.1371/journal.pcbi.1004768

Yang, S., Wang, J., and Deng, B. (2020). Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons. *IEEE Trans. Neural Netw. Learn. Syst.* 31, 148–162. doi: 10.1109/tnnls.2019.2899936

Yang, S., Wang, J., and Li, S. (2015). Cost-efficient FPGA implementation of basal ganglia and their Parkinsonian analysis. *Neural Netw.* 71, 62–75. doi: 10.1016/j.neunet.2015.07.017

Yang, S., Wang, J., Hao, X., Li, H., Wei, X., Deng, B., et al. (2021). BiCoSS: Toward large-scale cognition brain with multigranular neuromorphic architecture. *IEEE Trans. Neural Netw. Learn. Syst.* [Epub ahead of print]. doi: 10.1109/TNNLS.2020.3045492

Yang, S., Wang, J., Deng, B., Liu, C., Li, H., and Fietkiewicz, C. (2018). Real-time neuromorphic system for large-scale conductance-based spiking neural networks. *IEEE Trans. Cybern.* 49, 2490–2503. doi: 10.1109/tcyb.2018.2823730

Zenke, F., and Ganguli, S. (2018). Superspike: Supervised learning in multilayer spiking neural networks. *Neural Comput.* 30, 1514–1541. doi: 10.1162/neco_a_01086

Zhou, F., Zhou, Z., Chen, J., Choy, T. H., and Chai, Y. (2019). Optoelectronic resistive random access memory for neuromorphic vision sensors. *Nat. Nanotechnol.* 14, 776–782. doi: 10.1038/s41565-019-0501-3

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.