NADOL: Neuromorphic Architecture for Spike-driven Online Learning By Dendrites

Shuangming Yang, *Member, IEEE*, Haowen Wang, Yanwei Pang, *Senior Member, IEEE*, Mostafa Rahimi Azghadi, *Senior Member, IEEE*, Bernabe Linares-Barranco, *Fellow, IEEE*

***Abstract*—Biologically plausible learning with neuronal dendrites is a promising perspective to improve the spike-driven learning capability by introducing dendritic processing as an additional hyperparameter. Neuromorphic computing is an effective and essential solution towards spike-based machine intelligence and neural learning systems. However, on-line learning capability for neuromorphic models is still an open challenge. In this study a novel neuromorphic architecture with dendritic on-line learning (NADOL) is presented, which is a novel efficient methodology for brain-inspired intelligence on embedded hardware. With the feature of distributed processing using spiking neural network, NADOL can cut down the power consumption and enhance the learning efficiency and convergence speed. A detailed analysis for NADOL is presented, which demonstrates the effects of different conditions on learning capabilities, including neuron number in hidden layer, dendritic segregation parameters, feedback connection, and connection sparseness with various levels of amplification. Piecewise linear approximation approach is used to cut down the computational resource cost. The experimental results demonstrate a remarkable learning capability that surpasses other solutions, with NADOL exhibiting superior performance over the GPU platform in dendritic learning. This study's applicability extends across diverse domains, including the Internet of Things, robotic control, and brain-machine interfaces. Moreover, it signifies a pivotal step in bridging the gap between artificial intelligence and neuroscience through the introduction of an innovative neuromorphic paradigm.**

***Index Terms*—Spike-driven learning, neuromorphic, spiking neural network (SNN), online learning, dendritic learning**

I. Introduction

Neuromorphic engineering is a promising avenue towards building the next generation of intelligent supercomputing systems [1]-[3]. Inspired by the advanced information processing scheme of biological neural systems, neuromorphic systems have achieved significant breakthrough when dealing with brain-inspired computation tasks. In comparison with general-purpose computers, neuromorphic systems are considerably more efficient and suitable for real-time an d large-scale neural computation. They exhibit substantial potential for implementing streamlined natural signal processing systems, pattern recognition systems, and real-time autonomous agents [4]-[6]. Distinguished by their massively parallel computing substrate and co-localized memory and computation, neuromorphic hardware possesses formidable capabilities in addressing the von Neumann bottleneck issue, enhancing computational efficiency, and reducing power consumption [7]. A prominent question arises regarding the efficient realization of learning properties in spiking neural networks (SNNs) within neuromorphic systems. Effectively achieving biologically realistic spike-driven online learning on neuromorphic computing systems continues to pose a noteworthy challenge.

The human brain boasts an extraordinary capacity for learning, enabling individuals to absorb information from sensory stimuli and continually optimize their learning processes as they acquire new skills. In recent years, researchers have delved into the concept of "learning to optimize" within neuroscience, with a specific focus on efficient learning rules that rapidly converge to favorable solutions. This notion was introduced by Lansdell and Konrad [8], emphasizing the pivotal role of dendrites in individual neurons as they encode learning signals. This form of learning, emerging from the realm of neuroscience, has paved the path for optimizing synaptic strengths in the sensory and association cortex, consequently enhancing overall learning performance throughout the neural network.

Within the domain of computational neuroscience, the challenge of establishing connections for cognitive behaviors is widely recognized as the "credit assignment problem" [9], [10]. While artificial neural networks have yielded potent techniques to address this problem, establishing a coherent link between the learning algorithms employed in these networks and the biological learning mechanisms of the human brain remains a crucial, yet unanswered, query.

To bridge this knowledge gap, a pivotal facet of learning in the biological brain involves performance enhancement through task exposure. By harnessing neural cortical physiology, a neural system can leverage the learning-to-optimize capability through the apical and basal components within the dendritic trees of pyramidal neurons [11]. Through the utilization of distinct neuronal compartments responsible for integrating various signals, individual neurons

This work was supported partly by the National Key Research and Development Program of China (Grant No. 2022ZD0160405), and supported in part by the National Natural Science Foundation of China (Grant No. 62006170, 62376185, and 62176179), and partly by Young Elite Scientists Sponsorship Program by CAST (2022QNRC001) (corresponding e-mail: yangshuangming@tju.edu.cn).

Shuangming Yang and Yanwei Pang are with School of Electrical and Information Engineering, Tianjin University, Tianjin, 300072 China, and are also with Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China. Haowen Wang is with School of Electrical and Information Engineering, Tianjin University, Tianjin, 300072 China.

Mostafa Rahimi Azghadi is with the College of Science and Engineering, James Cook University, Townsville, QLD 4814, Australia.

Bernabe Linares-Barranco is with the Microelectronics Institute of Seville, Seville 41092, Spain.

can achieve independent integration—a biologically plausible resolution to the credit assignment problem when it pertains to learning to optimize functions.

While neuromorphic engineering has exhibited substantial potential in crafting intelligent supercomputing systems, devising an efficient approach for realizing spike-driven online learning on neuromorphic computing systems has remained elusive. The integration of the learning-to-optimize capability in a biologically realistic manner has often been underestimated or inadequately implemented [12]. Current systems, like the TrueNorth system, lack support for synaptic plasticity rules and associated learning capabilities [5]. Consequently, there exists a pressing demand for a proficient neuromorphic learning architecture to surmount these limitations.

This paper introduces NADOL (Neuromorphic Architecture for Dendritic Online Learning), an innovative neuromorphic architecture to facilitate learning processes. NADOL harnesses biologically plausible dendritic learning mechanisms, enabling spike-driven online learning. Additionally, we digitize neuron and synaptic activities while optimizing them to mitigate hardware expenses. We implemented NADOL on BiCoSS—an advanced neuromorphic system tailored for intricate SNNs and based on field-programmable gate arrays (FPGAs). FPGAs have proven indispensable in high-performance SNN computation [13-15]. NADOL furnishes a valuable foundation for designing neuromorphic online learning systems and evaluating their learning efficacy. Ultimately, NADOL's online learning capabilities pave the way for acquiring novel features from dynamic environments—a critical stride toward achieving lifelong learning. Moreover, the presented work bears practical implications, with applications ranging from autonomous embedded robots and the Internet of Things (IoT) to brain-machine interfaces and experimental neuroscience platforms [16-19].

The structure of this paper unfolds as follows: Section II elucidates the dendritic learning mechanism and theory. A comprehensive digital neuromorphic architecture of NADOL is introduced in Section III. Section IV presents the experimental findings, followed by discussions in Section V. The paper culminates in Section VI with concluding remarks.

II. Dendritic Learning Mechanism and Theory

A. Neuromorphic architecture

Previous studies have shown that online learning can be realized by using feedback signals that transmit neural information about credit to compute local error signals in hidden layers [22], [23]. They have presented a pivotal study aimed at unraveling the intricacies of the human brain's learning mechanisms, offering the potential to harness multi-layer neural architectures that rival the efficacy of backpropagation. However, the models encounter challenges when aligning them with real brain functions, as they necessitate a distinct feedback pathway to transmit neural information for determining local error signals, as illustrated in Fig. 1(a). This pathway is accountable for computing the disparity in error signals within the hidden layers, a computation that involves contrasting feedback triggered by feedforward propagation of sensory information and feedback guided by teaching signals. To compute the necessary disparity, sensory information must be segregated from the feedback signals driving the learning process. However, this strategy lacks biological realism in the context of the human brain. Such an approach requires pairing within the feedback pathway, entailing that each neuron in the hidden layer should be paired with a corresponding feedback neuron. Unfortunately, there exists no substantiated evidence confirming this architectural arrangement. The presence of numerous error neurons for each hidden layer neuron to communicate an error signal inefficiently seems implausible. Consequently, a dedicated separate feedback pathway for learning, utilizing cell-by-cell interactions and signed signals, appears incompatible with the operations of the real brain.

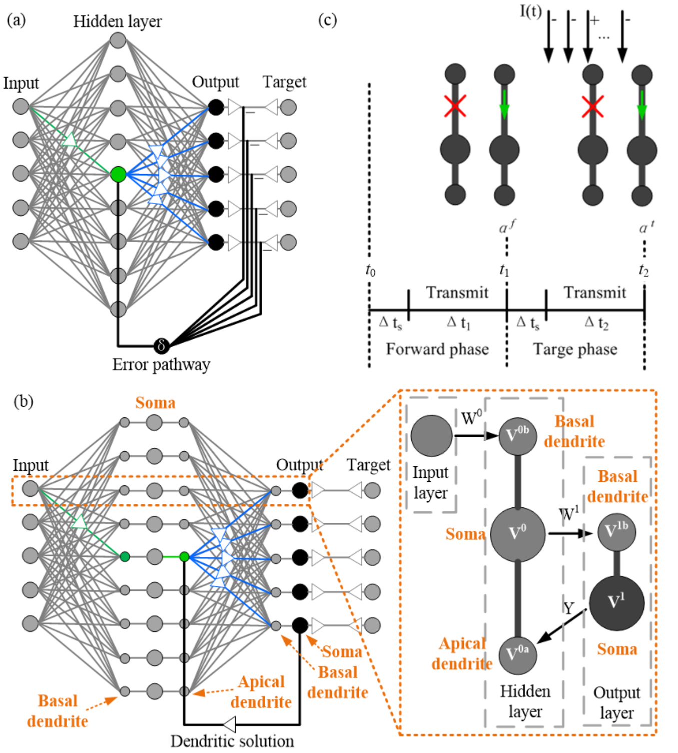

Fig. 1. Architectures for credit assignment for dendritic learning. (a) Deep learning architecture with the implicit feedback pathway in previous studies. (b) Network architecture with segregated dendrites proposal with details of the proposed network architecture with dendritic learning strategy. (c) Illustration of the two-phase training scheme on NADOL.

Inspired by neural morphology, different signals can be integrated at distinct dendritic locations. Previous studies have shown that feedback signals from higher-order regions are transmitted into the distal apical dendrites of pyramidal neuron in the primary sensory regions of neocortex, which are electronically considerably far from the basal dendrites that receives feedforward sensory information [24]. Therefore, this study employs the anatomy of pyramidal neurons to provide segregation of feedforward and feedback information to compute local error signals and perform learning in biologically plausible neural network. As shown in Fig. 1(b), since neurons in hidden layers involve segregated basal and apical dendritic compartments, the feedforward and feedback signals can be integrated separately to realize credit assignment for learning. The input signals are first encoded by spiking neurons in the input layer. Spikes from the input layer are then transmitted by synapses with synaptic weight W^0 to the basal dendrites in the hidden layer. After being processed by soma, the neural information are transmitted by synapse with synaptic weight W^1 to basal dendrites in the output layer. The soma in the output layer outputs the feedback information with synaptic weight Y to the apical dendrites in the hidden layer to realize the backpropagation task. This architecture builds upon prior research utilizing compartmental models [5], [22], [25]. The utilization of basal and apical dendritic compartments enables the integration of feedback signals alongside feedforward pathways. This ensures the generation of error information for hidden layers, facilitating accurate credit assignment-a biologically plausible process as observed in the mammalian neocortex. In essence, the significance of dendritic learning and its application for credit assignment encompasses four key aspects. Firstly, through the adoption of the dendritic learning methodology, a clear segregation between feedforward and feedback pathways is achieved. This segregation resolves the credit assignment challenge by employing diverse neuron sites. This circumvents the issue of gradient disappearance arising from the amalgamation of different information streams within a single compartment. Secondly, since backpropagation is absent in the brain, any neural-inspired hardware seeking to emulate the brain's functionality should adhere to biologically viable algorithms and architectures, such as the proposed dendritic learning framework. Thirdly, the dendritic learning approach directly furnishes feedback to hidden neurons, offering a hardware advantage by minimizing the processing time required for sequential error propagation. Fourthly, the utilization of dendritic learning permits a reduction in the number of connections for feedback pathways, given that the number of output neurons is typically significantly lower than that of input neurons. Consequently, the dendritic learning solution offers a pivotal avenue for comprehending the mechanics behind spike-driven learning in the human brain. This endeavor serves to bridge the gap between neuroscience and artificial intelligence, thereby advancing our understanding of these interconnected disciplines.

B. Algorithm and Theory

The proposed network model contains an input layer with 784 neurons, a hidden layer with 50 physical neurons and an output layer with 10 neurons. The two-phase training scheme is shown in Fig. 1(c). 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. It means that the connection between dendrite and soma is blocked during the transmit time, and the connection is realized at the end of the forward phase and target phase respectively. Two phases are alternated to train the network: the forward and target phases as shown in Fig. 1(c). In the forward phase $I(t)$=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 the network undergoes target phase. The values of $I(t)$ in the target phase will be positive and negative when correct and incorrect objectives are recognized respectively. At the end of the forward phase and the target phase, the set of plateau potentials α_f and α_t are calculated respectively. The term Δt_s represents a time delay of the network dynamics before integrating the plateau. Δt_1 and Δt_2 represent the time periods for the transmission of the spike information during forward and target phases respectively. During the transmission, the network dynamics are updated at each time-step. An image from the MNIST dataset is employed to transmitted into the input layer with one neuron per image pixel. Neurons in the input layer are simple Poisson spiking neurons where their firing rates are determined by their corresponding input image pixel intensity. Neurons in hidden layer contains three compartments, including the apical compartments with the membrane voltage V^{0a}, the basal dendrite compartment with voltage V^{0b}, and the soma compartment with voltage V^0. The output layer contains two-compartment neurons, one for each image category. Poisson spiking neurons are used in the input layer with the firing rate determined by the intensity of image pixels ranging from 0 to $\Phi_{\max}$. Neurons in the input layer are modeled with three compartments containing basal dendrites, apical dendrites and soma compartment. Feedforward signals from the input layer and feedback signals from the output layer are transmitted into basal and apical synapses. Presynaptic spikes from input layer neurons are filtered into spike trains $s^{\text{input}}(t)$ as follows

$$s_j^{input}(t)=\sum_k \kappa\left(t-t_{jk}^{input}\right) \tag{1}$$

where t_{jk}^{input} is the kth spiking time of input neuron j. The response kernel is described as follows

$$\kappa(t)=\left(e^{-t/\tau_L}-e^{-t/\tau_s}\right)\Theta(t)/\left(\tau_L-\tau_s\right) \tag{2}$$

where τ_s and τ_L are long and short time constants, and Θ is the Heaviside step function. The spike trains at apical synapses are filtered in the same manner. The basal and apical dendritic membrane voltages for neuron i are described as follows

$$\begin{cases} V_i^{0b}(t)=\sum_{j=1}^{784} W_{ij}^0 s_j^{input}(t)+b_i^0 \\ V_i^{0a}(t)=\sum_{j=1}^{10} Y_{ij}\, s_j^1(t) \end{cases} \tag{3}$$

where b^0 is bias term, W^0 is the synaptic matrix of feedforward signals in hidden layer, and Y represents the feedback weight matrix. Soma membrane potential is defined as

$$\begin{aligned} \tau\frac{dV_i^0(t)}{dt} &= \left(V^R-V_i^0(t)\right)+\frac{g_b}{g_l}\left(V_i^{0b}(t)-V_i^0(t)\right) \\ &\quad +\frac{g_a}{g_l}\left(V_i^{0a}(t)-V_i^0(t)\right) \\ &= \left(V^R-V_i^0(t)\right)+\frac{g_b}{g_l}\left(\sum_{j=1}^{784} W_{ij}^0 s_j^{input}(t)+b_i^0-V_i^0(t)\right) \\ &\quad +\frac{g_a}{g_l}\left(\sum_{j=1}^{10} Y_{ij}^0 s_j^1(t)-V_i^0(t)\right) \end{aligned} \tag{4}$$

where V^R is the resting potential and g_l is the leak conductance.

Conductance g_b is from the basal dendrite to the soma, and g_a is conductance from the apical dendrite to the soma. The synaptic weights Y_{ij} are randomly initialized using a normal distribution with mean value μ=0.0293 and standard deviation δ=0.6321. These weights are fix during learning. This configuration is based on the previous study by Lillicrap et al., which used random synaptic feedback weights to support error backpropagation [47]. In this study, researchers pointed out that a precise and symmetric backward connectivity pattern is impossible in human brain. This strong architectural constraint is not necessary for effective error propagation. In contrast, a simple mechanism was presented in study [47], which can assign the credit by multiplying errors by even random synaptic weights. This approach is effective to transmit teaching signals across multiple layers. It also provides a potential mechanism to explain how the brain could use error signals without architectural constraints on learning. Constant τ is defined as

$$\tau = \frac{C_m}{g_l}. \tag{5}$$

where C_m represents the membrane capacitance of the spiking neuron. The instantaneous firing rates of the hidden layer neurons are described by $\Phi(t)$, which is defined as follows

$$\phi_i^0(t) = \phi_{\max}\delta\left(V_i^0(t)\right) = \phi_{\max}\frac{1}{1+e^{-V_i^0(t)}} \tag{6}$$

where $\Phi_{\max}$ is the maximum firing rate of the neurons. The neurons in the output layer are modeled with a dendrite and a soma compartment, which is defined as follows

$$\begin{cases} V_i^{1b}(t) = \sum_{j=1}^{500} W_{ij}^1 s_j^0(t) + b_i^1 \\ \tau\dfrac{dV_i^1(t)}{dt} = \left(V^R - V_i^1(t)\right) + \dfrac{g_d}{g_l}\left(V_i^{1b}(t) - V_i^1(t)\right) + I_i(t) \end{cases} \tag{7}$$

where g_l is the leak conductance and g_d is the conductance from the dendrite to the soma. The current $I(t)$ are somatic currents to drive neurons in the output layer towards a desired somatic potential, which is defined as follows

$$I_i(t) = g_{Ei}(t)\left(E_E - V_i^1(t)\right) + g_{Ii}(t)\left(E_I - V_i^1(t)\right) \tag{8}$$

where $g_E(t)$ and $g_I(t)$ are time-varying excitatory and inhibitory conductances, and E_E and E_I represent the excitatory and inhibitory reversal potentials.

There are two processing phases in the network computing, which are forward and target phases. As shown in Fig. 1(c), during a forward phase an image to the input layer neuron without any teaching current into the output layer between time t_0 to t_1. During the target phase from t_1 to t_2, the image is also given into the input layer, and the teaching signals are received in the output layer. At the end of t_2, another plateau potential α^t is calculated across the hidden layer. At t_1 a plateau potential α^f is computed in all the hidden layer neurons. Plateau potentials α^f and α^t for forward and target phases are defined as follows

$$\begin{cases} \alpha_i^f = \sigma\left(\dfrac{1}{\Delta t_1}\displaystyle\int_{t_1-\Delta t_1}^{t_1} V_i^{0a}(t)\,dt\right) \\ \alpha_i^t = \sigma\left(\dfrac{1}{\Delta t_2}\displaystyle\int_{t_2-\Delta t_2}^{t_2} V_i^{0a}(t)\,dt\right) \end{cases} \tag{9}$$

where σ represents the nonlinear sigmoid function as

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{10}$$

The terms t_1 and t_2 represent the end times of the forward and target phases respectively, which are given by

$$\begin{cases} \Delta t_1 = t_1 - \left(t_0 + \Delta t_s\right) \\ \Delta t_2 = t_2 - \left(t_1 + \Delta t_s\right) \end{cases} \tag{11}$$

where Δt_s=30 ms. The plateau potentials are used in the hidden layer to update the corresponding weights of basal dendrites.

Feedforward synaptic weights are updated at the end of each target phase. A loss function is defined in output layer to update the synaptic weights W^1 to cut down the loss function as follows

$$\begin{aligned} L^1 &= \left\| \phi^{1*} - \phi_{\max}\sigma\left(\overline{V^1}^f\right)\right\|_2^2 \\ &\approx \left\| \overline{\phi^1}^t - \overline{\phi^1}^f \right\|_2^2 \end{aligned}. \tag{12}$$

The average membrane potential of soma in output layer neuron i in the forward phase is defined as follows

$$\overline{V_i^1}^f \approx k_d\overline{V_i^{1b}}^f = k_d\left(\sum_{j=1}^{500} W_{ij}^1\overline{s_j^0}^f + b_i^1\right) \tag{13}$$

where k_d=g_d/(g_l+g_d). Therefore, we can get the relationship as

$$\begin{cases} \dfrac{\partial L^1}{\partial W^1} \approx -k_d\phi_{\max}\left(\phi^{1*} - \phi_{\max}\sigma\left(\overline{V^1}^f\right)\right)\sigma'\left(\overline{V^1}^f\right)\circ\overline{s^0}^f \\ \dfrac{\partial L^1}{\partial b^1} \approx -k_d\phi_{\max}\left(\phi^{1*} - \phi_{\max}\sigma\left(\overline{V^1}^f\right)\right)\sigma'\left(\overline{V^1}^f\right) \end{cases}. \tag{14}$$

This equation of gradient is used in the output layer to update the weights based on gradient descent

$$\begin{cases} W^1 \to W^1 - \eta^1 P^1\dfrac{\partial L^1}{\partial W^1} \\ b^1 \to b^1 - \eta^1 P^1\dfrac{\partial L^1}{\partial b^1} \end{cases} \tag{15}$$

where η^1 represents a learning rate constant and P^1 stands for a scaling factor to normalize the firing rate scale.

The loss function for basal dendrites in the hidden layer to update their synaptic weights W^0 is defined as follows

$$L^1 = \left\| \phi^{0*} - \phi_{\max}\sigma\left(\overline{V^0}^f\right)\right\|_2^2. \tag{16}$$

The target firing rate Φ^{0*} is defined as follows

$$\phi_i^{0*} = \overline{\phi_i^0}^f + \alpha_i^t - \alpha_i^f \tag{17}$$

where α^f and α^t are plateau potentials in forward and target phases respectively. Therefore, the loss function in the hidden layer can be expressed as

$$L^0 \approx \left\|\alpha^t - \alpha^f\right\|_2^2 \tag{18}$$

The gradient can be described as

$$\begin{cases} \dfrac{\partial L^0}{\partial W^0} \approx -k_b\left(\alpha^t - \alpha^f\right)\phi_{\max}\sigma'\left(\overline{V^0}^f\right)\circ\overline{s^{input}}^f \\ \dfrac{\partial L^0}{\partial b^0} \approx -k_b\left(\alpha^t - \alpha^f\right)\phi_{\max}\sigma'\left(\overline{V^0}^f\right) \end{cases}. \tag{19}$$

with the parameter $k_b=g_b/(g_l+g_b+g_a)$. Basal weights are computed to descend the gradient as follows

$$\begin{cases} W^0 \to W^0 - \eta^0 P^0 \dfrac{\partial L^0}{\partial W^0} \\ b^0 \to b^0 - \eta^0 P^0 \dfrac{\partial L^0}{\partial b^0} \end{cases} . \tag{20}$$

where η^0 represents a learning rate constant and P^0 stands for a scaling factor to normalize the firing rate scale in the hidden layer.

III. Digital Neuromorphic Architecture

A. Top-level architecture

In order to increase the network scale significantly, a time-multiplexing method is used. The top-level architecture of the proposed neuromorphic system is shown in Fig. 2, including an input layer, the hidden layer with physical neurons and an output layer with 10 physical neurons. The input layer and the hidden layer are both implemented by using the time-multiplexing technique. A global counter processes the time-multiplexed layers sequentially. Finite-state machine (FSM) is employed to control the system timing, and weight updating units are used to calculate credit assignment signals and update the synaptic weights for deep learning. The updating weights for the physical neurons are stored in the weight buffer. The neurons in input layer are responsible for the generation of weight for each pixel of the input digit and the stimulus for each neuron in the first hidden layer by summing the weighted pixels. In order to save the hardware resource and avoid the use of 784 multipliers to calculate the multiplication between all the values of filtered spike trains and the corresponding weights, the input digit are pre-processed by converting the pre-calculated filtered spike trains (FSTs) to a binary value, so that there is a significant reduction of the computational elements in the computation of the hidden layer.

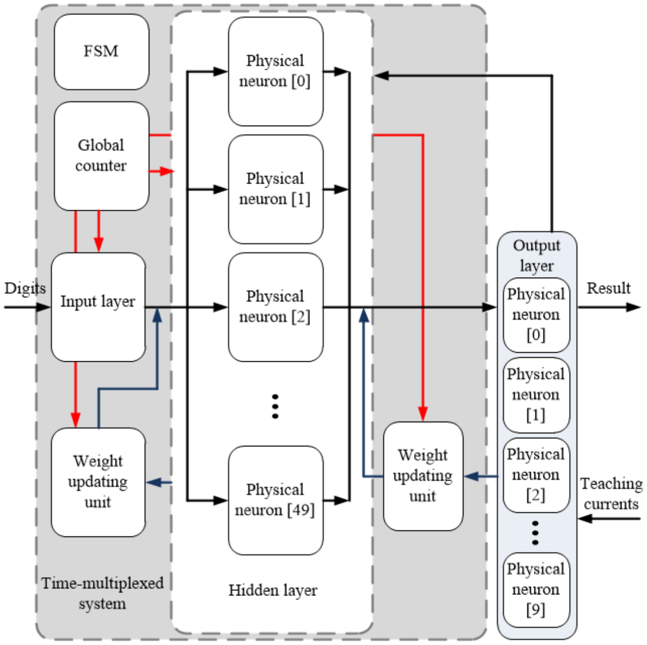

Fig. 2. The top-level architecture of the on-line learning network.

B. Equations discretization

Hardware implementation requires the representation of information transmission and processing in the form of discrete manner, rather than continuous differential form. Therefore, the Euler method was adopted for discretization of dynamics that mentioned above. The discrete form of soma membrane potential mentioned in equation (4) can be expressed as

$$V_i^0(n+1) = \begin{bmatrix} \left(V^R - V_i^0(n)\right) + \dfrac{g_b}{g_l}\left(V_i^{0b}(n) - V_i^0(n)\right) \\ + \dfrac{g_a}{g_l}\left(V_i^{0a}(n) - V_i^0(n)\right) \end{bmatrix} \frac{\Delta n}{\tau} \tag{21}$$

where n represents the number of iterative steps and Δn stands for the time step in the Euler method. τ represents the time constant.

The discrete form of neurons in the output layer that modeled with a dendrite compartment and a soma compartment that mentioned in equation (7) can be expressed as

$$\begin{cases} V_i^{1b}(n) = \sum\limits_{j=1}^{500} W_{ij}^1 s_j^0(n) + b_i^1 \\ V_i^1(n+1) = \begin{bmatrix} \left(V^R - V_i^1(n)\right) + I_i(n) \\ + \dfrac{g_d}{g_l}\left(V_i^{1b}(n) - V_i^1(n)\right) \end{bmatrix} \dfrac{\Delta n}{\tau} + V_i^1(n) \end{cases} \tag{22}$$

The discrete form of current I(t) that drive neurons in the output layer towards a desired somatic potential mentioned in equation (8) can be expressed as

$$I_i(n) = g_{Ei}(n)\left(E_E - V_i^1(n)\right) + g_{Ii}(n)\left(E_I - V_i^1(n)\right) \tag{23}$$

The discrete form of plateau potentials α^f and α^t for forward and target phases mentioned in equation (9) can be expressed as

$$\begin{cases} \alpha_i^f(n) = \sigma\left(\dfrac{1}{\Delta n_1} \sum\limits_{n_1-\Delta n_1}^{n_1} V_i^{0a}(n)\right) \\ \alpha_i^t(n) = \sigma\left(\dfrac{1}{\Delta n_2} \sum\limits_{n_2-\Delta n_2}^{n_2} V_i^{0a}(n)\right) \end{cases} \tag{24}$$

Therefore, a step-by-step breakdown of the continuous algorithm can be converted into the discrete counter parts based on Euler method.

C. Input and hidden layer architecture

The time-multiplexing architecture of the input layer and the hidden layer is shown in Fig. 3(a), which contains a global counter, two neuron processors for each hidden layer, one input layer and two buffers for the updating weights of the two hidden layers. The neuron processor in the hidden layer consists of three parts according to the neuronal morphological properties, including apical dendrite unit, soma unit and basal dendrite unit. Fig. 3(b) depicts the detailed architecture of the soma unit. Pipeline technique is employed in the calculation, aiming at the maximum working speed on the digital chip. ADD and SUB modules represent the pipelined adder and subtractor, and SHF module stands for the barrel shifter to replace the multiplication between a variable and a constant. By replacing the multipliers with the shifters, the hardware resource cost can be cut down significantly.

The working flow of the FSM is illustrated in Fig. 3(c), which contains eight states, 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 complex calculation stage can be controlled in good hardware performance.

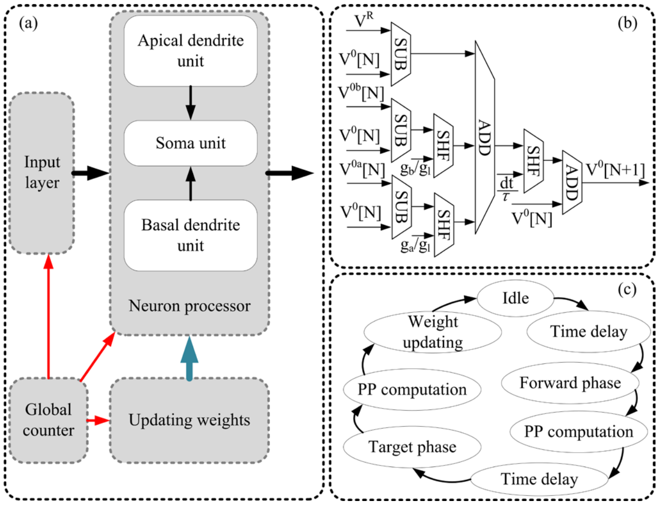

Fig. 3. The digital neuromorphic architecture of the hidden layer. (a) The architecture of the time-multiplexed system. (b) The architecture of the soma unit. (c) The diagram flow of the FSM.

Fig. 4 shows the digital neuromorphic architecture of the basal dendrite in the hidden layer. Each binary filtered spike train is employed to control a multiplexer with two inputs, one for its corresponding synaptic weight and the other for zero. In the proposed algorithm, 784 weighted pixels should be summed, which will cost a significant amount of logic elements on chip. In order to cut down the hardware resource overhead, the proposed architecture perform this sum in four clock cycles. The digital neuromorphic architecture of the basal dendrite unit is shown in Fig. 4, and it contains two progressive summation unit. Assuming that there are N_a weighted pixels input to the basal dendritic unit, then the basal dendritic unit can complete the summation of $N_a^{\ 2}$ weighted pixels in one operation. The blue area represents a multiplexer module, which contains 14 multiplexers that select data from 0 and W_0, generating 14 outputs. These 14 outputs are fed into a 1st-layer 14-input parallel adder module, represented in pink area. This module calculates the sum of the 14 inputs and produces one output. The combination of a blue module and a pink module can calculate the sum of 14 weighted pixels. We have construct 14 such module combinations, forming the multiplexer array and the parallel adder array. Therefore, at the output of the 1st-layer 14-input parallel adder array, we have 14 outputs. Subsequently, these 14 outputs are fed into a 2nd-layer 14-input parallel adder module for summation. For the bias terms, we have designed 14 buffers corresponding to blue area, and their outputs b_0 are fed into the 1st-layer 14-input parallel adder for summation, resulting in 14 outputs. Then, these 14 outputs are fed into the 2nd-layer 14-input parallel adder for summation, producing the bias summation result corresponding to the blue module. Finally, in the gray module, the bias summation result is added to the sum of the weighted pixels using a pipelined adder, and finally, the result will feed into the accumulator for the summation of four clock cycles. Therefore, if we want to sum up 784 weighted pixels in four clock cycles, we should sum up 196 weighted pixels in one clock. Since we have set up 14 multiplexer arrays and the corresponding 14 parallel adder modules, we only need to calculate the sum of 14 weighted pixels within one module. The proposed architecture of the input layer contains an input buffer, a global counter, a W_0 weight buffer and a b_0 bias buffer, a 14 multiplexers with 2 inputs, four 14-input parallel adders and an accumulator. The input buffer stores the input digits. The global counter will send the stored digit to the multiplexers for the generation of the weighted pixels. The lowest 14 bits are input into the multiplexer in the first clock cycle, and other pixel bits will be transmitted sequentially in another three clock cycles. The accumulator sums all the 784 pixels up in four clock cycles and send the stimulation into the first hidden layer. Pipeline technique is used in the design for the sake of the enhancement of the maximum operating frequency. The architecture of the apical dendrite unit is the same with the basal dendrite unit, without the bias part in its architecture. The basal and apical dendrite units in the hidden layer use the signals of FST. The detailed architecture for the FST computation is described in the following section.

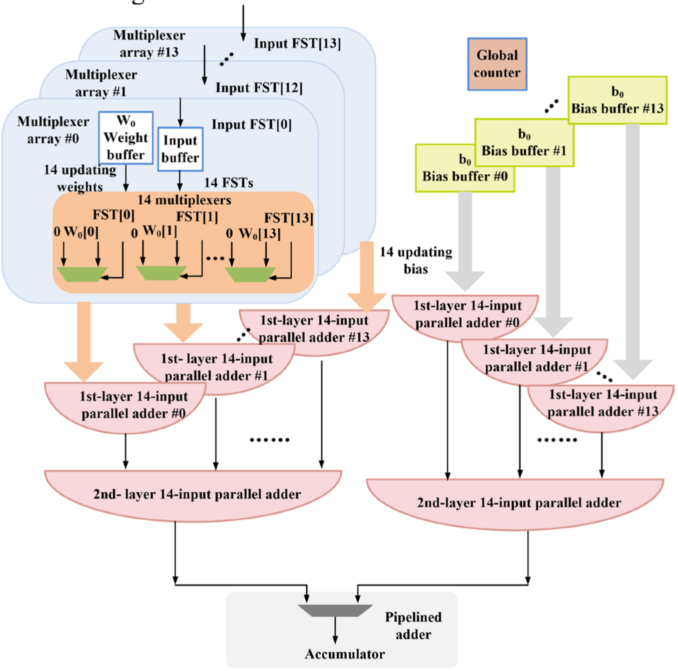

Fig. 4. The digital neuromorphic architecture of the basal dendrite unit.

D. Output layer architecture

The output layer architecture is similar with the hidden layer, without the apical unit realization. The computation of the basal dendrite unit uses the FST signals, whose computational architecture is depicted in Fig. 5(a). The PLA1 module is the hardware realization of the PLA equation of $\sigma(V^0)$. In terms of equations (1) and (2), a sliding time window of 10 ms is used for the computation of $s^{\text{input}}(t)$. Therefore, ten time deviation values for (t-t_{jk}^{input}) are computed, which are from $\kappa(0)$, $\kappa(1)$, $\kappa(2)$ to $\kappa(9)$. In order to simplify the on-chip computation, the values of $\kappa(t)$ for each time value are pre-calculated, which are shown in Table I. The values of the response kernel are obtained according to equation (2). The multiplication operations are replaced by shifting operations using SHF modules. The shift operation is a low-cost computational method that can effectively reduce hardware resource overhead and computation time. We convert the values in Table I into the shift number to represent it in digital neuromorphic architecture. For $\kappa(9)$, since 0.0269 can be expressed as $2^{-6}+2^{-7}+2^{-9}+2^{-10}+2^{-11}$, it can be implemented using five SHF modules, shifting by 6, 7, 9, 10, and 11 bits respectively. Similarly, for other values of κ, we can calculate them based on equation (2) and represent them as corresponding shift numbers. For $\kappa(8)$, since $0.0436=2^{-5}+2^{-7}+2^{-8}+2^{-11}+2^{-12}$, it can be implemented using five SHF modules, shifting left by 5, 7, 8, 11, and 12 bits respectively. For $\kappa(0)$, since $0.0475=2^{-5}+2^{-6}$, it can be implemented by two SHFs, shifting left by 5 bits and 6 bits respectively. As shown in Fig. 5(a), "PARA. ADD" represents the parallel adder that has multiple inputs, while "ADD" represents the adder with two inputs. However, since the shift operation can only handle integer values or powers of two in binary decimal representation, it introduces approximation errors when using non-power-of-two constants. We tested the displacement error of multiplication operation. As shown in Fig. 5(b), the architecture of the basal dendrite unit in the output layer uses accumulator to sum up the updating variable values of the time-multiplexed 100 neurons. The MUL module represents the multiplication operation, which uses logic elements to realize.

The detailed architecture of the MUL module is shown in Fig. 5(c). The proposed MUL module is used to realize the multiplication operation between two variables with powers of 2 using a logic shift block, aiming at multiplier-less implementation with lower resource cost and power consumption. Two variables "$a[n]$" and "$b[n]$" are input into the MUL module, and the value of $a[n]$ is expected to be in the range from 0 to1. The bus splitter is employed to split a bus into single-bit outputs, which are numbered from the least significant bit to the most. The MUX module contains several multiplexers to select the input data flow based on the information from the bus splitter. If the information from the bus splitter equals to 1, $b[n]$ is shifted leftwards. The variable $b[n]$ is routed into each data port and then all the outputs of multiplexers are summed up. The multiplication of two neural variables can be realized based on this method with no multiplier usage, which can reduce the on-chip resource significantly.

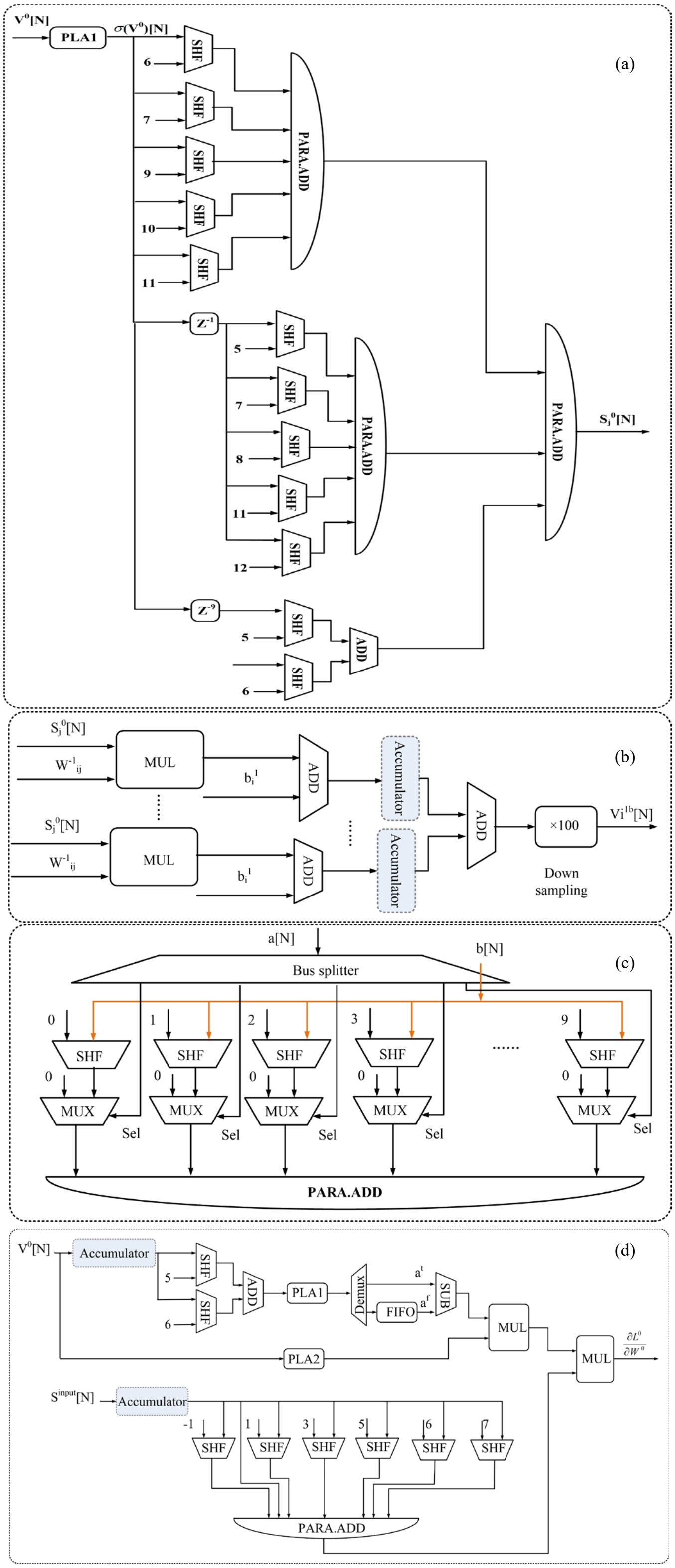

Fig. 5. Detailed digital neuromorphic architecture of the proposed neuron in output layer. (a) Architecture of the FST module. (b) Architecutre of the basal dendrite unit in the neuron processor. (c) The digital neuromorphic architecture of the MUL module. (d) Architecture of the weight updating unit.

TABLE I
PARAMETER VALUES OF THE RESPONSE KERNEL

Parameter	Value	Parameter	Value
$\kappa(0)$	0.0475	$\kappa(1)$	0.0510
$\kappa(2)$	0.0543	$\kappa(3)$	0.0571
$\kappa(4)$	0.0591	$\kappa(5)$	0.0600
$\kappa(6)$	0.0581	$\kappa(7)$	0.0533
$\kappa(8)$	0.0436	$\kappa(9)$	0.0269

E. Weight updating architecture

In order to realize the spike-driven on-line learning, the weight updating units are implemented in the proposed dendritic learning architecture as shown in Fig. 5(d). Demux module represents the demultiplexer, which select to output either the target PP or the forward PP for the computation of the gradient. The value of $\eta^0 P^0$ is pre-calculated as 3.68. The accumulator module is used to accumulate the input values of variables "$V^0[N]$" and "$s^{input}[N]$" within a certain time period.

In the proposed architecture, the PLA1 and PLA2 modules implement the piecewise linear approximation (PLA) functions, which are used to cut down the hardware resource cost of the nonlinear functions in the SNN model as shown in Fig. 6. The functions $\sigma(x)$ and $\sigma'(x)$ are modified using the PLA method, which is described as follows

$$f_{PLA}(x) = \begin{cases} k_1 x + b_1, \text{when } x \le p_1 \\ k_2 x + b_2, \text{when } p_1 < x \le p_2 \\ \ldots \\ k_i x + b_i, \text{when } x > p_{i-1} \end{cases} \qquad (25)$$

where k_i and b_i are the slope and intercept of the modified PLA functions, and i=1, 2,..., n. An exhaustive search algorithm is employed to determine the segment points on the nonlinear curves. The determination of the coefficient values are based on the error evaluation criterion as follows

$$CF_{error} = \sqrt{\sum_{i=1}^{n} \left(\left(f_{ori}(i) - f_{PLA}(i) \right)^2 \Big/ f_{ori}(i)^2 \right)} \Big/ n \qquad (26)$$

where n represents the total sampling points. The functions f_{ori} and f_{PLA} represent the original and approximated functions. If the modified function cannot meet the precision requirement of CF_{error}, the segment number will be added by 1 until it can be met. The multiplication operation is replaced with addition and shifting operations in the proposed digital neuromorphic architecture. The value k_i in the proposed PLA functions should be a power of 2, such as 1, 2, 4 or 0.5, 0.25, etc. The parameter values of the PLA methods are listed in Table II.

TABLE II
PARAMETER VALUES OF THE PROPOSED PLA FUNCTIONS

$\sigma(x)$	k	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	$\sigma(x) \geqq 1$
$\sigma'(x)$	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	$\sigma'(x) \leqq 0$

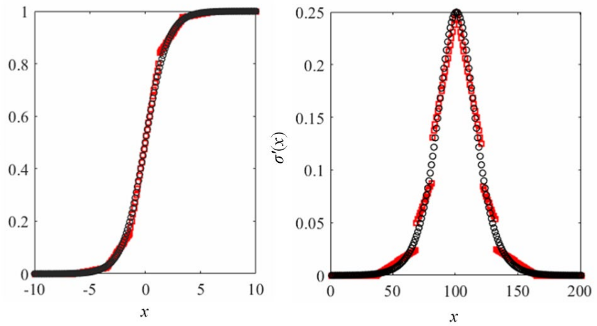

Fig. 6. PLA functions in the weight updating architecture.

IV. EXPERIMENTAL RESULTS

To demonstrate the capabilities of the proposed architecture, it is implemented on a single chip of BiCoSS, a digital neuromorphic system with powerful computational capability [27]. The computation of the fixed-point representation is realized in binary form. Although BiCoSS presents an essential

TABLE III Comparison of NADOL with Previous Digital Neuromorphic Architecture Implemented on FPGAs.

Performance	[49]	[50]	[51]	[21]	[35]	[36]	[37]	[52]	[53]	[54]	[55]	[56]	Ours
Platform	Virtex-6 FPGA	Virtex-7 FPGA	Spartan-6 FPGA	Spartan-6 FPGA	Kintex-7 FPGA	Virtex-6 FPGA	Cyclone-4 FPGA	Zynq-7045	Virtex-6	Virtex-7	Zynq-7030	Zynq-7405	BiCoSS
Max Frequency	189MHz	63.389 MHz	75 MHz	25 MHz	148.4 MHz	83.209 MHz	65.03 MHz	250 MHz	120 MHz	100 MHz	301.8 MHz	200 MHz	70.25 MHz
Learning speed	N/A	N/A	6.58 fps^3	6.25 fps^3	N/A	N/A	N/A	1349 fps	0.06 fps	61 fps	163.9 fps	22.5 fps	6.77-129.87 fps
Neuron model	LIF	H-H	LIF	LIF	LIF	Izhi.	LIF	LIF	LIF	LIF	LIF	LIF	MC-LIF
Online Learning	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Biological plausibility	Low	High	Low	Low	Low	Moderate	Low	High	High	High	High	High	High
Application	Vision	Simulation	Vision	Vision	Learning	Learning	Anditory	Vision	Vision	Vision	Vision	Vision	Learning

N/A: Data not available. Abbreviation: Slice registers (SRs) platform to implement the proposed architecture, the presented NADOL can be realized on different kinds of neuromorphic systems, including mixed-analog-digital systems.

We implemented our multi-compartment spiking neural network with dendritic compartments on the BiCoSS system. VHDL language was used in the synthesis process on the FPGA, which possesses stronger behavior description ability compared to other hardware description languages. The resource utilization is shown in Table III and compared with several representative FPGA implementations. Since the size of the input image is 28×28, the total number of pixels of the network input M is 784, and we denote the neuron nodes of hidden layer is represented by N. Therefore, the average encoded input spike counts E and output spike counts per training image F are 1232 and 1.05 respectively [52]. Assumming that $T_{training}$ represents the average learning time and T_{fps} represents the frame rate, they can be calculated as

$$T_{training} = 1.05 \times T \times \left[E \times \left(2N + 16 \right) + F \times \left(M + N + 16 \right) \right] \quad (27)$$

$$T_{fps} = 1 / T_{training} \quad (28)$$

where T represents the time clock and the value of T in our work is 70.25 MHz. In our experiment, we employed multiple hidden layer neuron sizes for performance investigation. When N=200, the $T_{training}$ and T_{fps} in our work are 7.7ms and 129.87 frames per second (fps) respectively. When N=4000, the $T_{training}$ and T_{fps} in our work are 147.7ms and 6.77 fps respectively.

Farsa et al. implemented a hardware unit for neural computing and an SNN neural morphology system for pattern recognition tasks on Virtex-6 FPGA [49]. Neil et al. introduced an event-driven low-power neural network accelerator [51], although specific resource utilization data were not provided, the system supports up to 65536 neurons. Bonabi et al. proposed a Hodgkin-Huxley (H-H) neuron model using the coordinate rotation digital computer algorithm, which exhibits strong biological plausibility [50]. Similarly, Gholami et al. presented the Izhikevich (Izhi.) neuron model with moderate biological plausibility [36]. However, the cost significantly increases with the number of multipliers, and these designs inevitably use DSP blocks for multiplication between two variables, making it difficult to scale the network. Ma et al. introduced the Darwin Neural Processing Unit, a highly configurable neuromorphic hardware coprocessor based on SNN [21], which achieves good performance and efficiency, but falls short in terms of maximum operating frequency compared to other works. Asgari et al. proposed a context-dependent learning system with low energy consumption and fast speed, utilizing the low power consumption advantage of the Kintex-7 FPGA, but only implemented 16 neurons [35]. Similarly, [37] implemented a neural morphology auditory system with 66 neurons. Both of these works lack good scalability. In contrast, the NADOL architecture proposed in this study innovatively implements a scalable, biologically plausible large-scale digital neuromorphic system. Its mechanism of utilizing dendrites for spatiotemporal credit allocation helps bridge the gap between neuroscience and artificial intelligence, serving as a low-power system for AI applications and a real-time online simulation platform for understanding neural mechanisms.

We chose MNIST classification to test the performance of NADOL, particularly its training efficiency. It is worth noting that, due to the adoption of dendritic learning mechanisms with strong biological plausibility in our NADOL architecture, the goal of the MNIST classification test is not to achieve higher accuracy compared to ANN or less biologically plausible SNN. Instead, we aim to demonstrate the high efficiency characteristics of the NADOL architecture. Therefore, we measured the training energy overhead (%) between training and inference. By examining this performance metric, we can directly compare it with learning systems based on ANN implemented on chips, including [46], [47], [20]. As shown in Fig. 7, the training energy overhead (%) of our proposed NADOL architecture is 6.6%, significantly lower than the 56.5% reported in [20], resulting in an 88.3% reduction. The improvement is realized by algorithmic modifications, parallel processing for on-chip learning, and the spike-driven sparse encoding in the SNN framework.

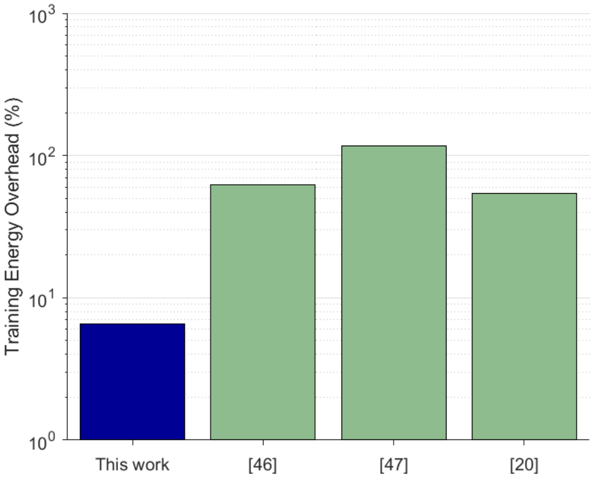

Fig. 7. Comparison of the learning energy over the inference energy with other neuromorphic on-line learning systems.

We then investigate the effects of the neuron number on the learning capability of NADOL. In the data presented in this study, the images in the MNIST training set are presented one at a time, and each exposure to the full set of images is considered an "epoch" of training. At the end of each epoch, the classification accuracy on a separate set of test images is assessed with a single forward phase for each image. The classification accuracy is judged by which output neuron has the highest average firing rate during the test image forward phases. In Fig. 8-10, we vary various parameters in our model to study their effect when performing more learning epochs. As shown in Fig. 8, NADOL shows a high classification capability in 10 epochs. However, the accuracy is only slightly enhanced with the neuron number increased from 500 to 3000. Adding more neurons beyond 4000 does not improve the learning accuracy. Therefore, NADOL with 4000 neurons is the most beneficial configuration with the strongest learning performance. NADOL presents a scalable architecture for learning by dendrites, which supports the scale up of the neuromorphic network model.

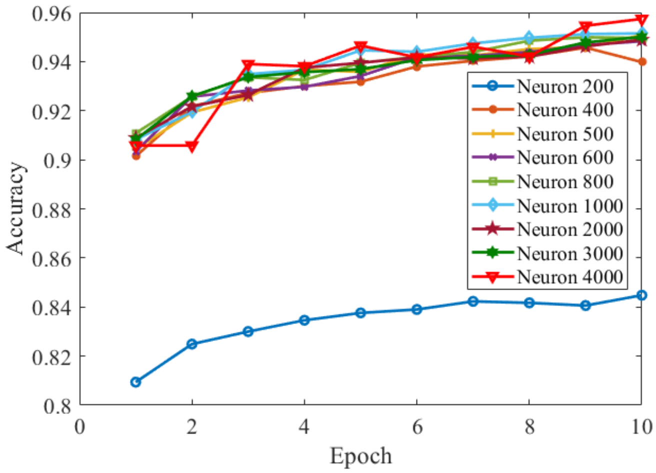

Fig. 8. Classification accuracy with different numbers of neurons in the hidden layer.

In order to further investigate the tradeoff between accuracy and power consumption in training the MNIST dataset on NADOL, we examined different configurations of the NADOL architecture as shown in Fig. 8. In Table IV, we provide the training energy overhead (%) between the training and inference phases. We can observe that as the number of neurons in the NADOL implementation increases, there is an increase in training energy overhead (%). However, the training energy overhead (%) remains lower than existing AI chips and neuromorphic chips [46][47][20][48], while maintaining a high learning accuracy based on strong biological plausibility learning mechanisms.

TABLE IV
COMPARISON WITH OTHER DIGITAL NEUROMORPHIC APPROACHES

Configuration	Training energy overhead (%)
#200	5.3%
#400	5.5%
#500	5.7%
#600	5.8%
#800	6.0%
#1000	6.2%
#2000	6.2%
#3000	6.3%
#4000	6.6%

All results are synthesized and verified with the same test vectors.

The learning accuracy of the proposed algorithm is further explored with different levels of apical dendritic segregation. Previous study has shown that the biological pyramidal neurons only show an attenuation of distal apical inputs to the soma [43]. As shown in Fig. 9(a), different levels of dendritic segregation will induce various levels of learning performance. It reveals that the strong apical attenuation with g_a=0.1 or 0.2 will induce a better learning performance in the proposed NADOL algorithm than the total attenuation or weak apical attenuation such as g_a=0.6 or 0.8. It reveals that the electronically segregated dendrites is a meaningful approach to realize the separation between feed-forward and feedback data low for learning to learn. Besides, we try to explore the effects of different types of synaptic feedback weights on the learning performance. Five types of feedback synapses are considered, which are random synaptic feedback, symmetric synaptic feedback, synaptic feedback with sinusoidal noise, Gaussian noise and square-wave noise. In order to include noise in the feedback synapses and investigate their effects on our model performance, sinusoidal, Gaussian and square-wave noises, which are three critical categories of conventional noise, are considered. Sinusoidal noise is the most single frequency component. Any complicated signal can be regarded as the combination of sinusoidal signals with different frequencies and amplitudes. Gaussian noise represents the noise whose probability density function obeys Gaussian distribution, i.e., normal distribution. The common Gaussian noise induces fluctuation noise, cosmic noise, thermal noise and shot noise. Square-wave noise contains odd harmonic components. Gibbs phenomenon appears by Fourier conversion to represent the square wave. It contains fundamental harmonic and third harmonic at least. The sinusoidal noise is defined as follows

$$n_1 = \sin(t) \tag{29}$$

where t=0:π/180:28π, and n_1 represents the generated sinusoidal signal.

The Gaussian noise in this study is expressed as

$$n_2 = \sigma\square\sqrt{-2\ln(1-t_1)}\cos 2\pi t_2 + \mu \tag{30}$$

where t_1 and t_2 are two random independent variables that follows the uniform distribution with [0,1]. The variable n_2 is the generated Gaussian noise signal.

Besides, the square-wave noise signal is calculated by the following equation:

$$\begin{aligned} n_3 &= sin(x)+1/3*sin(3*x)+1/5*sin(5*x) \\ &+1/7*sin(7*x)+1/9*sin(9*x) \\ &+1/11*sin(11*x)+1/13*sin(13*x) \\ &+1/15*sin(15*x)+1/17*sin(17*x) \\ &+1/19*sin(19*x)+1/21*sin(21*x) \end{aligned} \tag{31}$$

where x=0:π/180:28π and n_3 represents the generated square-wave signal.

As shown in Fig. 9(b), learning performance using the synaptic feedback with the symmetric weights is better than the random feedback weights. In addition, the synaptic feedback based on the symmetric weights with square-wave noise can further improve the learning performance of the proposed algorithm.

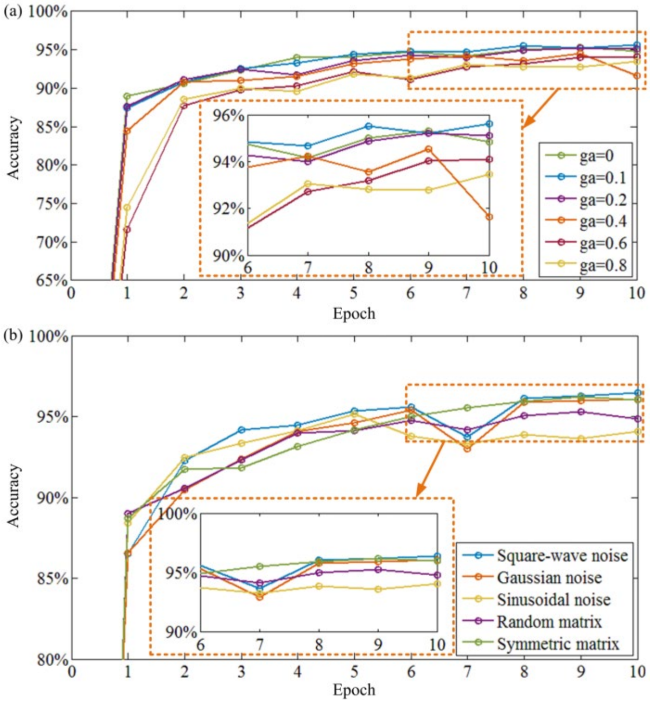

Fig. 9. Performance analysis for the effects of the apical attenuation and synaptic feedback weights. (a) The effect of the dendritic segregation on learning performance. (b) The effect of the synaptic feedback.

In order to explore the effects of the sparseness of the synaptic feedback weight on the learning performance, we use different levels of sparseness to test the recognition accuracy. As shown in Fig. 10(a), the high-level sparseness of the synaptic weight with 5% and 10% sparseness will induce a reduction of the learning performance, while the learning accuracy of the other sparseness conditions are consistent with each other. The network using weight sparseness with amplification by the corresponding times is also investigated, for example, 10% sparseness with 10-time amplification, 20% sparseness with 5-time amplification. Fig. 10(b) reveals that the learning performance is also lower in the conditions of 10% sparseness and 12.5% sparseness in spite of the amplification. The situation of 16.7% sparseness with amplification by 6 times results in the best learning performance compared to other groups, which means that appropriately strong sparseness will facilitate the learning of the proposed algorithm. Fig. 10(c) shows that the learning performance of the sparseness weight with amplification is better than the sparseness processing without amplification. It reveals that the compensation for the loss of the weight sparseness can improve the learning capability of the proposed algorithm, and the sparse feedback can provide a information signal that is sufficient for credit assignment during learning.

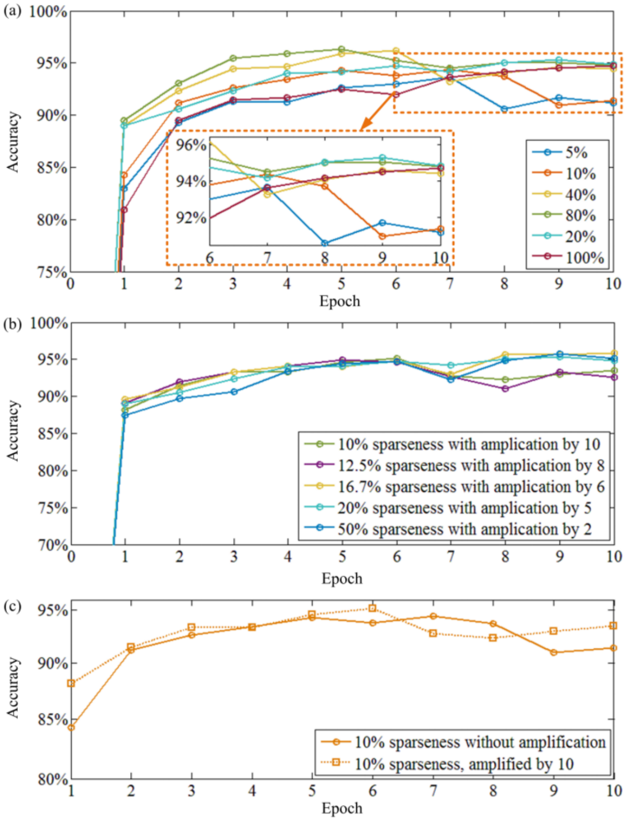

Fig. 10. Performance analysis of the effects of the feedback sparseness on the learning performance. (a) The effect of the network connection sparseness on the learning performance across 10 epochs. (b) Effects of weight magnitudes for learning with sparse weights.

The throughput of NADOL at nominal voltage for on-chip learning is compared with a high-performance GPU NVIDA Titan X as shown in Fig. 11. It shows that the presented NADOL architecture can be 6.2× faster computation in comparison with GPU training. The real-time computation evaluation shows that the presented system converges 4× faster than GPU training.

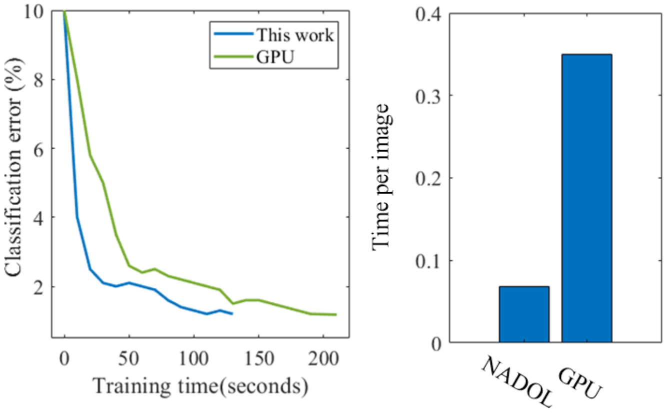

Fig. 11. Throughput of the learning in comparison with a single Titan-X GPU.

In this paper, our main focus is on establishing a neurally-inspired online learning architecture using dendritic learning mechanisms with strong biological plausibility. We aim to demonstrate the training efficiency of relatively small-scale networks. However, to prove the effectiveness of NADOL, we further examined its performance on more complex datasets. It is important to note that the dendritic learning mechanisms currently employed in NADOL cannot be applied to convolutional neural networks, resulting in a loss of learning accuracy on complex datasets. In the future, we will investigate improved NADOL architectures for convolutional networks, although this may sacrifice low-power performance in pursuit of higher learning accuracy, providing more options for neurally-inspired learning architectures. Although these results lag far behind state-of-the-art classification models that utilize convolutional neural networks and less biologically plausible spiking neural networks, they still demonstrate that the proposed NADOL architecture can provide sufficient classification performance as an alternative to traditional AI neural networks and improve the learning performance of networks with the same architecture that do not utilize multi-compartment models.

V. Discussions

Guided by brain-inspired "spiking" computational frameworks, neuromorphic computing represents a pivotal solution for brain-inspired machine intelligence, offering the promise of achieving artificial intelligence while substantially reducing the energy demands of computing platforms. This study introduces an innovative neuromorphic architecture, named NADOL, designed to address the critical challenge of spike-driven online learning, a bottleneck in the realm of neuromorphic computing and embedded artificial intelligence. As illustrated in Fig. 1, NADOL leverages dendritic processing mechanisms, enabling learning through dendritic nonlinear computations. A comprehensive overview of the architecture is provided in Fig. 2 to Fig. 5, elucidating the mapping strategy from the biological dendritic network to the spiking neuron model and ultimately to the neuromorphic architecture. The study utilizes the PLA method to further optimize the NADOL algorithm, as depicted in Fig. 6. Furthermore, Fig. 7 underscores NADOL's superiority in terms of training energy overhead (%)—a testament to its energy-efficient solution for neuromorphic online learning. Fig. 8 delves into the learning capacity with varying neuron numbers in the hidden layer, emphasizing the importance of a scalable and efficient architecture for implementing larger-scale SNN models for dendritic learning. The study then delves deeper into NADOL's dendritic learning architecture, investigating the impacts of dendritic segregation and different synaptic feedback strategies on NADOL's learning capability, as exemplified in Fig. 10. Hardware resource utilization is meticulously detailed in Table III, with a series of cost functions presented to evaluate NADOL's hardware performance. Fig. 11 serves as an evaluation of NADOL's hardware architecture, marking a significant stride toward spike-driven learning in neuromorphic engineering.

The major contributions of NADOL can be distilled into three key facets. First and foremost, NADOL offers a neuromorphic perspective on brain-inspired intelligence. It incorporates a novel semi-supervised learning mechanism that exploits spatiotemporal event representations. Secondly, NADOL represents a novel neuromorphic architecture characterized by high learning capacity and low power consumption. SNNs on NADOL leverage timing information, bestowing them with the inherent advantages of sparsity and efficiency in spiking dynamics. Lastly, NADOL serves as a bridge between neuroscience and brain-inspired intelligence, employing a neuromorphic approach. The integration of dendritic learning, which enhances spike-based learning through the addition of dendritic connections as an additional hyperparameter, opens intriguing avenues for elevating the intelligence level of neuromorphic systems.

One of the most remarkable merits of neuromorphic computing is its energy efficiency. NADOL utilizes fewer than 20kSynOps events, primarily generated between the input and hidden layers. The energy consumption of a synaptic operation on BiCoSS hovers around 25pJ. In contrast, single spike classification on conventional neuromorphic systems consumes approximately 500pJ, underscoring the remarkable efficiency of NADOL, especially in comparison to GPU-based platforms. It is worth noting that in this study, we implemented the hardware SNN model using fixed-point representation, which yielded slightly lower accuracy when juxtaposed with software neural networks employing floating-point number representations. Nevertheless, there exists a trade-off between accuracy and hardware resource cost. Additionally, our proposed system can be scaled up through the utilization of the well-established time-multiplexing technique. However, this expansion necessitates FPGA chips with augmented memory resources to accommodate the proposed SNN on a single chip.

Another notable advantage of neuromorphic engineering is its alignment with the biological mechanisms of the real brain. Efficient learning hinges on the ability to assign contributions to behaviors for each neuron, a conundrum known as the credit assignment problem. In hierarchical networks with multiple processing stages, distinguishing credit-related activities from non-credit-related activities via synaptic plasticity rules can be daunting when credit signals are integrated with other input signals. Herein, the spatial layout and nonlinear dynamics of the dendrite structure play a pivotal role in disentangling credit signals from other inputs. Evidence suggests that top-down feedback signals are integrated in the distal apical dendrites in cortical pyramidal neurons, profoundly influencing neural spiking and synaptic plasticity. This underscores the utility of distal apical dendrites in resolving the credit assignment problem within the human brain.

While previous works have explored supervised learning with the backpropagation algorithm based on SNN models and achieved impressive results in unsupervised learning with STDP algorithms, these models lack hardware implementations in the existing literature. Therefore, we refrain from direct comparisons with NADOL, given the absence of hardware implementations for these models to the best of our knowledge.

It is important to note that the maximum accuracy attained by our neuromorphic system with dendritic learning falls below the accuracy achieved by supervised deep learning methods employing convolutional layers. This discrepancy can be attributed to the limitations of the two-layer network architecture we employed. Nonetheless, one of the primary goals of neuromorphic hardware is to uncover the operational mechanisms of biological neural systems. We opted against using a convolutional structure due to its lack of biological plausibility. Dendritic learning, on the other hand, aligns with biologically plausible learning mechanisms that can occur within the brain.

Real biological neurons are not singular compartments; they possess intricate dendrites that integrate various signals at different positions through nonlinear processing methods, yielding essential functions. One viable solution to the credit assignment problem is to segregate credit signals into dendritic compartments. This approach ensures that credit signals remain distinct from ongoing computations, driving unique spiking activities dedicated to transmitting credit information. Therefore, dendritic processing emerges as a crucial element in resolving the credit assignment problem in a biologically plausible manner.

While assuming that all neurons contain a single compartment simplifies mathematical modeling and analysis, it diverges from the structure of the biological brain. Deep learning models introduce a separate pathway outside the trained network, a construct absent in the human brain. Recent computational research underscores the potential significance of independent compartments and nonlinear dendrites in addressing the credit assignment problem within a biologically plausible framework. Consequently, it is imperative for the neuromorphic computing and brain-inspired intelligence communities to continue exploring and harnessing dendritic learning within SNNs.

VI. Conclusions

In this study a novel neuromorphic architecture NADOL is presented using the biologically plausible learning rules, which has the energy efficient on-line learning capability and shows a classification performance superior in comparison with other neuromorphic approaches. Specifically, the results are achieved by optimizing the proposed algorithm, dedicated parallel processing on BiCoSS system, and utilizing the sparse spike-driven encoding within the SNN framework. A comprehensive analysis is performed, considering the effects of neuron number in hidden layer, dendritic segregation, feedback connective forms, and sparseness methods on the learning capability of NADOL. It shows that NADOL outperforms other solutions, and has higher learning efficiency compared to GPU platform. The presented study can be applied in kinds of fields, including autonomous embedded robots, internet of things, brain-machine interfaces, and experimental neuroscience platforms. It is also a critical approach towards the further comprehension of biological mechanisms underlying online learning in human brain.

Acknowledgment

The authors would like to thank the anonymous editor and reviewers for their constructive comments and suggestions.

References

[1] B. J. Lansdell and K. P. Kording, "Towards learning-to-learn," *Current Opinion in Behavioral Sciences*, vol. 29, pp. 5-50, 2019.

[2] R. S. Sutton and A. G. Barto, "Reinforcement Learning," *A bradford book*, vol. 15, no. 7, pp. 665-685, 1998.

[3] Littman and L. Michael, "Reinforcement learning improves behaviour from evaluative feedback," *Nature*, vol. 521, no. 7553, pp. 445-451, 2015.

[4] Y. Lecun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436, 2015.

[5] K. P. Kording and P. Knig, "Supervised and Unsupervised Learning with Two Sites of Synaptic Integration," *J. Comput. Neurosci.*, vol. 11, no. 3, pp. 207-215, 2001.

[6] B. V. Benjamin, P. Gao, E. McQuinn, et al., "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proc. IEEE*, vol. 102, no. 5, pp. 699-716, 2014.

[7] P. Thomas, A. Grübl and J. Sebastian, "Six networks on a universal neuromorphic computing substrate," *Front. Neurosci.*, vol. 7, no. 7, pp. 11, 2013.

[8] J. Park, T. Yu and S. Joshi, "Hierarchical Address Event Routing for Reconfigurable Large-Scale Neuromorphic Systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2408-2422, 2017.

[9] E. Chicca, F. Stefanini and C. Bartolozzi, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proc. IEEE*, vol. 102, no. 9, pp. 1367-1388, 2014.

[10] P. A. Merolla, J. V. Arthur and R. Alvarez-Icaza, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668-673, 2014.

[11] N. Qiao, H. Mostafa, and F. CorradiR, "A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses," *Front. Neurosci.*, vol. 9, no. 141, 2015.

[12] G. Indiveri, Linares-Barranco, Bernabé and T. J. Hamilton, "Neuromorphic Silicon Neuron Circuits," *Front. Neurosci.*, vol. 5, no. 73, pp. 73, 2011.

[13] W. Florian, R. Florian and K. Alois, "Neuromorphic implementations of neurobiological learning algorithms for spiking neural networks," *Neural Netw.*, vol. 72, pp. 152-167, 2015.

[14] S. Yang, J. Wang, X. Hao, et al., "BiCoSS: Toward Large-Scale Cognition Brain With Multigranular Neuromorphic Architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, 2021.

[15] S. Yang, J. Wang and B. Deng, et al., "Real-Time Neuromorphic System for Large-Scale Conductance-Based Spiking Neural Networks," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 1-14, 2018.

[16] S. Yang, B. Deng, J. Wang, et al., "Design of hidden-property-based variable universe fuzzy control for movement disorders and its efficient reconfigurable implementation," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 2, pp. 304-318, 2018.

[17] S. Yang, B. Deng and J. Wang, et al., "Scalable Digital Neuromorphic Architecture for Large-Scale Biophysically Meaningful Neural Network With Multi-Compartment Neurons," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 148-162, 2019.

[18] S. Yang, J. Wang and S. Li, et al., "Cost-efficient FPGA implementation of a biologically plausible dopamine neural network and its application," *Neurocomputing*, vol. 177 pp. 274-289, 2016.

[19] M. B. Milde, B. Hermann and A. Dietmüller, "Obstacle Avoidance and Target Acquisition for Robot Navigation Using a Mixed Signal Analog/Digital Neuromorphic Processing System," *Front. Neurosci.*, vol. 11, pp. 28, 2017.

[20] C. H. Tsai, W. J. Yu, W. H. Wong and C.-Y. Lee, "A 41.3/26.7 pJ per neuron weight RBM processor supporting on-chip learning/inference for IoT applications", *IEEE J. Solid-State Circuits*, vol. 52, no. 10, pp. 2601-2612, Oct. 2017.

[21] D. Ma et al., "Darwin: A neuromorphic hardware co-processor based on spiking neural networks", J. Syst. Architecture, vol. 77, pp. 43-51, 2017.

[22] T. P. Lillicrap, D. Cownden and D. B. Tweed, "Random synaptic feedback weights support error backpropagation for deep learning," *Nat. Commun.*, vol. 7, pp. 13276, 2016.

[23] Satoshi, Manita and Takayuki, "A Top-Down Cortical Circuit for Accurate Sensory Perception," *Neuron*, vol. 86, no. 5, pp. 1304-1316, 2015.

[24] L. Qianli, Z. L. Joel and P. Tomaso, "How Important is Weight Symmetry in Backpropagation?" *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[25] M. W. Spratling and M. H. Johnson, "A feedback model of perceptual learning and categorization," *Vis. Cogn.*, vol. 13, no. 2, pp. 129-165, 2006.

[26] Y. Lecun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436, 2015.

[27] V. Mnih, K. Kavukcuoglu, D. Silver, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, 2015.

[28] V. D. M. Laurens and G. Hinton, "Visualizing Data using t-SNE," *J. Machine Learn. Res.*, vol. 9, no. 2605, pp. 579-2605, 2008.

[29] S. K. Esser, P. A. Merolla and J. V. Arthur, "Convolutional Networks for Fast, Energy-Efficient Neuromorphic Computing," *Proc Natl Acad. USA*, vol. 113, no. 41, pp. 11441-11446, 2016.

[30] W. Muñoz, R. Tremblay and D. Lenstein, "Layer-specific modulation of neocortical dendritic inhibition during active wakefulness," *Science*, vol. 355, no. 6328, pp. 954-959, 2017.

[31] C. Schmidt-Hieber, G. Toleikyte and L. Aitchison, "Active dendritic integration as a mechanism for robust and precise grid cell firing," *Nat. Neurosci.*, vol. 20, no. 8, pp. 1114-1121 , 2017.

[32] N. Takahashi, T. G. Oertner and P. Hegemann, "Active cortical dendrites modulate perception," *Science*, vol. 354, no. 6319, pp. 1587-1590, 2016.

[33] A. Amravati, S. B. Nasir, S. Thangadurai, I. Yoon, and A. Raychowdhury, “A 55nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots,” *IEEE ISSCC*, pp. 124–125, Feb. 2018.

[34] S. Gonugondla, M. Kang, and N. Shanbhag, “A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training,” *IEEE ISSCC*, pp. 490–491, Feb. 2018.

[35] H. Asgari, B. M. -N. Maybodi, M. Payvand and M. R. Azghadi, "Low-Energy and fast spiking neural network for context-dependent learning on FPGA", *IEEE Trans. Circuits Syst. II-Exp. Briefs*, vol. 67, no. 11, pp. 2697-2701, Nov. 2020.

[36] M. Gholami, E. Z. Farsa and G. Karimi, "Reconfigurable field-programmable gate array-based on-chip learning neuromorphic digital implementation for nonlinear function approximation", *Int. J. Circuit Theory Appl.*, vol. 49, no. 8, pp. 2425-2435, Jun. 2021.

[37] B. Deng, Y. Fan, J. Wang and S. Yang, "Reconstruction of a Fully Paralleled Auditory Spiking Neural Network and FPGA Implementation," *IEEE Trans Biomed. Circ. Syst.*, vol. 15, no. 6, pp. 1320-1331, Dec. 2021.

[38] Q. Wang, Y. Li, B. Shao, et al., "Energy efficient parallel neuromorphic architectures with approximate arithmetic on FPGA," *Neurocomputing*, vol. 221, pp. 146-158, 2017.

[39] N. B. Fred, B. Peter and L. Jiabo, "A 3.43TOPS/W 48.9pJ/pixel 50.1nJ/classification 512 analog neuron sparse coding neural network with on-chip learning and classification in 40nm CMOS," *Symposium on VLSI Circuits. IEEE*, pp. 5-8, 2017.

[40] M. Suzuki and M. E. Larkum, "Dendritic calcium spikes are clearly detectable at the cortical surface," *Nat. Commun.*, vol. 8, no. 1, pp. 276, 2017.

[41] J. Guerguiev, T. P. Lillicrap and B. A. Richards, "Towards deep learning with segregated dendrites," *eLife*, vol. 6, 2017.

[42] J. Bono and C. Clopath, "Modeling somatic and dendritic spike mediated plasticity at the single neuron and network level," *Nat. Commun.*, vol. 8, no. 1, pp. 706, 2017.

[43] S. Yang, B. Chen, “SNIB: Improving Spike-Based Machine Learning Using Nonlinear Information Bottleneck,” *IEEE Trans. Syst. Man Cybern. Syst.*, 2023.

[44] H. Mostafa, “Supervised learning based on temporal coding in spiking neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3227-3235, 2017.

[45] S. R. Kheradpisheh, & T. Masquelier, “S4NN: temporal backpropagation for spiking neural networks with one spike per neuron,” *Intern. J. Neural Syst.*, vol., 30, no. 6, pp. 2050027, 2020.

[46] S. Gonugondla, M. Kang and N. Shanbhag, "A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training", *IEEE ISSCC Dig. Tech. Papers*, pp. 490-491, Feb. 2018.

[47] A. Amravati, S. B. Nasir, S. Thangadurai, I. Yoon and A. Raychowdhury, "A 55nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots", *IEEE ISSCC Dig. Tech. Papers*, pp. 124-125, Feb. 2018.

[48] J. Park, J. Lee and D. Jeon, "A 65-nm neuromorphic image classification processor with energy-efficient training through direct spike-only feedback", *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 108-119, Jan. 2020.

[49] E. Z. Farsa et al., "A low-cost high-speed neuromorphic hardware based on spiking neural network", *IEEE Trans. Circuits Syst. II*, vol. 66, no. 9, pp. 1582-1586, Sep. 2019.

[50] S. Y. Bonabi et al., "FPGA implementation of a biological neural network based on the Hodgkin-Huxley neuron model", *Front. Neurosci.*, vol. 8, no. 379, pp. 1-12, 2014.

[51] D. Neil and S. Liu, "Minitaur an event-driven FPGA-based spiking network sebrator", *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 22, no. 12, pp. 2621-2628, Dec. 2014.

[52] H. Wang et al., "TripleBrain: A Compact Neuromorphic Hardware Core With Fast On-Chip Self-Organizing and Reinforcement Spike-Timing Dependent Plasticity," *IEEE Trans Biomed. Circ. Syst.*, vol. 16, no. 4, pp. 636-650, Aug. 2022.

[53] Q. Wang, Y. Li, B. Shao, S. Dey, and P. Li, “Energy efficient parallel neuromorphic architectures with approximate arithmetic on FPGA,” *Neurocomputing*, vol. 221, pp. 146–158, 2017.

[54] S. Li, Z. Zhang, R. Mao, J. Xiao, L. Chang, and J. Zhou, “A fast and energy-efficient SNN processor with adaptive clock/event-driven computation scheme and online learning,” *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 68, no. 4, pp. 1543–1552, Apr. 2021.

[55] J. Wu et al., “Efficient design of spiking neural network with STDP learning based on fast CORDIC,” *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 68, no. 6, pp. 2522–2534, Jun. 2021.

[56] Z. He et al., “A low-cost FPGA implementation of spiking extreme learning machine with On-chip reward-modulated STDP learning,” *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 69, no. 3, pp. 1657 – 1661, Mar. 2022.

Shuangming Yang received his M.S. degree and Ph.D. degree from Tianjin University, Tianjin, China in 2016 and 2020 respectively. He is currently an assistant professor in the School of Electrical and Information Engineering, Tianjin University. His research interests include neuromorphic engineering, neural system modeling, neural engineering, brain-inspired computing, and machine learning. He is currently a Review Editor for *Frontiers in Neuroscience*.

Haowen Wang received the bachelor degree in control science and engineering from Xi ’ an university of technology, Xi’an, China, in 2022. She is currently pursuing for a master ’ s degree at Tianjin University, Tianjin, China. Her current research interest is brain-inspired computing.

Yanwei Pang received the Ph.D. degree in electronic engineering from the University of Science and Technology of China, Hefei, China, in 2004. He is currently aProfessor with Tianjin University, China, and alsothe Founding Director of the Tianjin Key Laboratoryof Brain Inspired Intelligence Technology (BIIT),China. His research interests include object detectionand image recognition, in which he has published150 scientific articles, including 40 IEEE TRANS-ACTIONS articles and 30 top conferences (e.g.,CVPR, ICCV, and ECCV) papers. He is an Associate Editor of both IEEE transactions on neural networks and learning systems (TNNLS) and Neural Networks (Elsevier) and a Guest Editor of Pattern Recognition Letters.

Mostafa Rahimi Azghadi (S’07–M’14, SM’19) received the Ph.D. degree in electrical and electronic engineering from The University of Adelaide, Australia. From 2012 to 2014, he was a Visiting Ph.D. Student with the Neuromorphic Cognitive System Group, Institute of Neuroinformatics, University and Swiss Federal Institute of Technology (ETH), Zurich, Switzerland. He is currently a Lecturer with the College of Science and

Engineering, James Cook University, Townsville, Australia, where he researches neuromorphic engineering and brain-inspired architectures. He was a recipient of several national and international awards and scholarships, such as the Queensland Young Tall Poppy Science Award in 2017 and the South Australia Science Excellence Awards in 2015.

Bernabe Linares-Barranco (M'90, S'06, F'10) received the B.S. degree in electronic physics, the M.S. degree in microelectronics, and a first Ph.D. degree in 1990 from the University of Sevilla, Sevilla, Spain, in 1986, 1987, and 1990, respectively, and a second Ph.D. degree from Texas A&M University, College Station, TX, USA, in 1991. He currently is Full Professor of Research and serves as Director of the Institute since February 2018. His recent interests are in Address-Event-Representation VLSI, real-time AER vision sensing and processing chips, memristor circuits, and extending AER to the nanoscale. He has received two IEEE Transactions Best Paper Awards, and has been an Associate Editor of the IEEE transactions on circuits and systems-II, the IEEE transactions on neural networks, and Frontiers in neuromorphic engineering. From 2011 to 2013, he was the Chair of the IEEE Circuits and Systems Society Spain Chapter, and became an IEEE Fellow in 2010.