Active Dendrites Enable Efficient Continual Learning in Time-To-First-Spike Neural Networks

Pes, Lorenzo; Luiken, Rick; Corradi, Federico; Frenkel, Charlotte

DOI
10.1109/AICAS59952.2024.10595872

Publication date
2024

Document Version
Final published version

Published in
2024 IEEE 6th International Conference on AI Circuits and Systems, AICAS 2024 - Proceedings

Active Dendrites Enable Efficient Continual Learning in Time-To-First-Spike Neural Networks

Lorenzo Pes[1,2⋆], Rick Luiken[1], Federico Corradi[1] and Charlotte Frenkel[2]
[1] Electrical Engineering Department, Eindhoven University of Technology, Eindhoven, The Netherlands
[2] Microelectronics Department, Delft University of Technology, Delft, The Netherlands
⋆ *Correspondence to l.pes@tue.nl*

2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS) | 979-8-3503-8363-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/AICAS59952.2024.10595872

***Abstract*—While the human brain efficiently adapts to new tasks from a continuous stream of information, neural network models struggle to learn from sequential information without catastrophically forgetting previously learned tasks. This limitation presents a significant hurdle in deploying edge devices in real-world scenarios where information is presented in an inherently sequential manner. Active dendrites of pyramidal neurons play an important role in the brain's ability to learn new tasks incrementally. By exploiting key properties of time-to-first-spike (TTFS) encoding and leveraging its high sparsity, we present a novel spiking neural network (SNN) model enhanced with active dendrites. Our model can efficiently mitigate catastrophic forgetting in temporally-encoded SNNs, which we demonstrate with an end-of-training accuracy across tasks of 88.3% on the test set using the Split MNIST dataset. Furthermore, we provide a novel digital hardware architecture that paves the way for real-world deployment in edge devices. Using a Xilinx Zynq-7020 SoC FPGA, we demonstrate a 100-% match with our quantized software model, achieving an average inference time of 37.3 ms and an 80.0% accuracy.**

***Index Terms*—Spiking Neural Networks (SNNs), Neuromorphic Computing, Continual Learning, Time-To-First-Spike (TTFS), Active Dendrites, Field Programmable-Gate Arrays (FPGAs)**

I. Introduction

As humans experience the physical world, they demonstrate the innate ability to sequentially learn new tasks without forgetting how to perform previously learned ones. For example, consider the consecutive learning experiences of a human as depicted in Fig. 1a, from the initial steps of walking to the ability to drive a bike without falling, and finally to the more complex task of driving a car. Humans do not forget to walk or ride a bike when learning to drive a car. In stark contrast, as illustrated in Fig. 1b, machine learning (ML) models typically struggle to learn new tasks sequentially without forgetting previously learned tasks [1]. To mitigate this problem, conventional training methodologies based on *error backpropagation* (BP) [2] and *stochastic gradient descent* (SGD) [3], rely on examples of different tasks being presented in an interleaved fashion, as in Fig. 1c.

However, while this approach is at the core of today's state-of-the-art performance of ML models on pattern recognition [4], object detection in images and videos [5], [6], natural language processing [7], [8], and speech recognition [9], real-world scenarios at the edge mostly rely on information being presented in a sequential fashion. To deploy ML techniques in such use cases without suffering from *catastrophic forgetting* [1], various approaches have been proposed and can be categorized into three groups: *regularization-based*, attempting to prevent changes in parameters that are important for a previously-learned task [10], [11]; *architectural*, which use a subset of parameters for each new task [12], [13]; and *replay-based*, where data from previous tasks is presented again to the network while new tasks are being trained [14], [15].

Beyond suffering from catastrophic forgetting, current ML systems still lag orders of magnitude behind their biological counterparts in terms of energy efficiency [16]. In an attempt to better approach brain's efficiency, spiking neural networks (SNNs) are an increasingly popular network model. Various spike coding schemes have been investigated to represent information, with the most common ones being:

- *Rate coding* – Information is encoded in the instantaneous frequency of spike streams. This coding scheme is popular in SNN models thanks to its robustness and ease of use, where precision can be achieved at the expense of sparsity.
- *Time-to-first-spike (TTFS) coding* – Information is encoded into the spike time from an initial observation reference, where the more important the information, the earlier the spike. Often combined with the assumption that each neuron spikes at most once, TTFS coding outlines significant energy savings in SNN hardware as it allows optimizing for sparsity [17].

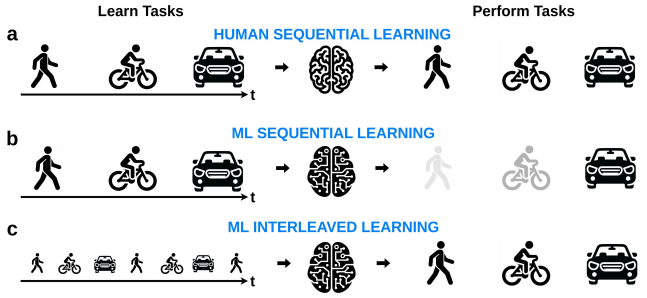

Fig. 1. **Human versus conventional ML learning**. **a** The human brain can learn new tasks in a sequential order without forgetting previous ones. **b** Training ML models in a sequential order leads to catastrophic forgetting. **c** Interleaving samples when training ML models avoids catastrophic forgetting.

Nevertheless, TTFS-based networks are typically considered difficult to train due to the *dead neurons* problem [18], where inactive neurons do not contribute to the learning process. This issue, combined with catastrophic forgetting, is currently hindering the adoption of TTFS-encoded SNNs for adaptive edge computing based on streaming data. In this work, we solve this challenge by introducing the concept of *active dendrites* [13] in TTFS-encoded SNNs. Active dendrites, coupled with a gating mechanism, allow for a dynamic selection of different sub-networks for different tasks, which mitigates catastrophic forgetting by avoiding overwriting previous knowledge. Interestingly, the dead neurons problem of TTFS networks can be exploited to perform this gating mechanism intrinsically. We demonstrate these findings by showcasing a test accuracy of 88.3% in sequentially training tasks based on the Split MNIST dataset. Additionally, we propose a digital hardware architecture for TTFS-encoded SNNs enhanced with active dendrites, which can perform inference with an average time of 37.3 ms while fully matching the results from our quantized software model.

II. Background Material

A. Time-to-first-spike neural networks

In light of their compelling performance in terms of inference time and power consumption, TTFS neural networks sparked a strong interest in the field of *neuromorphic computing* [19]–[22]. While most previous approaches encounter problems of non-differentiable spike functions [19] and exploding gradients [20], [21], Zhang *et al.* in [18] propose an elegant solution to these issues by defining the membrane potential evolution of a neuron as

$$V_j^l(t) = \begin{cases} \sum_i^I W_{ij}^l (t - t_i^{l-1}), & \text{if } t > t_i^{l-1} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where V_j^l is the membrane potential of neuron j in layer l, t is the simulation timestep, t_i^{l-1} is the spike time of the pre-synaptic neuron i in the preceding layer $l-1$ and W_{ij}^l is the synaptic strength connecting neuron i to neuron j. Specifically, $i \in \mathbb{R}^{\mathbb{I}}$ and $j \in \mathbb{R}^{\mathbb{J}}$, with I being the total number of neurons in layer $l-1$ and J the total number of neurons in layer l.

When a pre-synaptic neuron i emits a spike, the membrane potential of the post-neuron j linearly integrates the synaptic connection W_{ij}^l as depicted in Fig. 2a. During the integration process, if the membrane potential of neuron j crosses the *threshold voltage*, i.e V_{th}, a spike is emitted as

$$S_j^l(t) = \begin{cases} 1, & \text{if } V_j^l(t) \geq V_{th} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The time at which this event occurs, that is, t_j^l, is called the *spike time* and can be found by plugging $V_j^l(t_j) = V_{th}$ into (1), resulting in

$$t_j^l = \frac{V_{th} + \sum_i^I W_{ij}^l t_i^l}{\sum_i^I W_{ij}^l}, \quad \text{if } t > t_i^{l-1}. \tag{3}$$

If the spike time of a neuron occurs after the maximum observation windows defined by T_{max}, the neuron is considered *dead*, carrying no information and consequently having a gradient of zero during error BP. According to Zhang *et al.* in [18], the required gradients for applying error backpropagation are

$$\frac{\partial t_j^l}{\partial W_{ij}^l} = \frac{\partial t_j^l}{\partial V(t_j^l)} \frac{\partial V(t_j^l)}{\partial W_{ij}^l} = \frac{t_i^{l-1} - t_j^l}{\sum_i^I W_{ij}^l}, \tag{4}$$

$$\frac{\partial t_j^l}{\partial t_i^{l-1}} = \frac{\partial t_j^l}{\partial V(t_j^l)} \frac{\partial V(t_j^l)}{\partial t_i^{l-1}} = \frac{W_{ij}^l}{\sum_i^I W_{ij}^l}. \tag{5}$$

B. Active dendrites in artificial neural networks

Conventional artificial and spiking neuron models are historically rooted in the *point neuron* model, which assumes a linear impact of all synapses on the membrane potential [23]. This model lacks the architectural organization and dynamics found in the most abundant neurons in the cerebral cortex: pyramidal neurons, illustrated in Fig. 2b. Synaptic connections of a neuron are located on dendritic branches, which are referred to as proximal or distal when they are close or far from the neuron body (soma), respectively. Furthermore, dendrites can be of two types: basal or apical, depending on whether they connect to the soma or the apex of the neuron. While synapses located in proximal dendrites are believed to linearly scale the input signal of neighboring neurons, distal dendrites, also referred to as *active dendrites*, perform non-linear local integration of the input

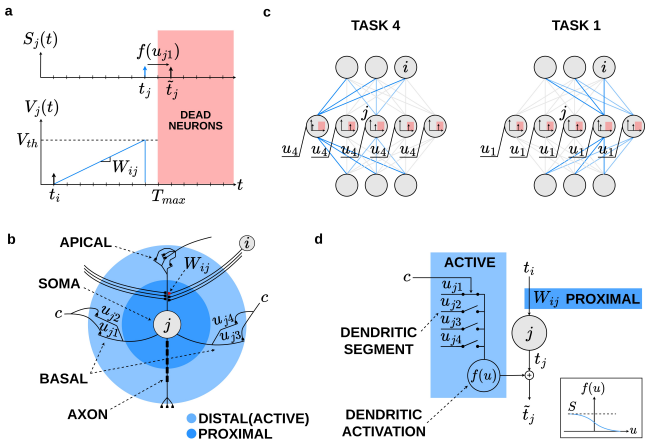

Fig. 2. **Neuron model and network architecture. a** Bottom: linear integration of synaptic strength W_{ij} following a pre-synaptic spike at t_i. Top: dendritic modulation of the spike time delay. **b** Illustration of a pyramidal neuron. **c** Selection of different sub-networks for different tasks based on the activity of dendritic segments. Dead neurons are used to efficiently implement a gating mechanism, with outgoing synaptic connections shown in gray (connections from active neurons are shown in blue). **d** Proposed neuron model and dendritic activation function.

signals [24]. Specifically, basal active dendrites process contextual information from neighboring cortical areas [25] and modulate the activity of the soma in a context-dependent manner [26].

Inspired by these ideas, Iyer *et al.* in [13] demonstrate that enhancing an artificial neuron model with the processing capabilities of active dendrites mitigates the problem of catastrophic forgetting by fostering the emergence of different subnetworks for different tasks. Specifically, the authors introduce a modulation effect on the feedforward activation y of a neuron j to mimic the behavior of active dendrites as

$$\tilde{y_j} = y_j \times \sigma(\max_j u_j^T c), \tag{6}$$

where, $\tilde{y_j}$ is the modulated activation, $\sigma(\cdot)$ is a sigmoidal dendritic activation function, u_j is a dendritic segment and c is the *context vector*. In a continual learning setup, the context vector is changed upon the start of a new task and a different dendritic segment is selected accordingly. Based on the value of the selected segment, the activity of a neuron can be down-modulated or remain unchanged. To foster the emergence of different subnetworks for different tasks, Iyer et al. [13] propose to add a *k*WTA layer after each layer enhanced with active dendrites. This layer selects the *k* neurons with the highest activation and gates the others. In this manner, only a small subset of neurons in the layer is activated for a given task. Furthermore, during the backward pass, only the synaptic strengths and the active dendritic segment of the winning neurons are updated. By following this approach, different subnetworks emerge for each task, thereby reducing interference between tasks and mitigating catastrophic forgetting.

III. Proposed Active-Dendrites Algorithmic and Hardware Framework for TTFS-Encoded Networks

A. Neuron model and network architecture

To mitigate the problem of catastrophic forgetting in SNNs while exploiting the sparsity of TTFS encoding, our neuron model enhances the model of [18] with a simplified version of the active dendrites

proposed in [13]. As explained in Section II-B, active dendrites modulate the activity of artificial neurons in a context-dependent manner. However, while artificial neurons encode information in real-valued numbers, TTFS neurons encode information in the spike time. Thus, to modulate the activity of a TTFS neuron, we introduce a dendritic-dependent spike time delay mechanism as follows:

$$\tilde{t_j^l} = t_j^l + f(u_{jn}^l), \tag{7}$$

where $\tilde{t_j^l}$ is the dendritic-modulated spike time, t_j^l is the spike time defined in Eq. (3), u_{jn}^l is the dendritic segment selected for task n, and $f(\cdot)$ is the dendritic activation function defined as

$$f(u) = \frac{S}{1+e^u}, \tag{8}$$

where S is a hyperparameter that controls the strength of the dendritic delay. It ensures that negative dendrites increase delay, while positive dendrites reduce delay.

The minimum number of dendritic segments for each neuron must match the number of tasks, that is, $u_j \in \mathbb{R}^{\mathbb{N}}$, where N is the total number of tasks. Depending on the task being performed, the context vector c connects a different dendritic segment to the activation function. As illustrated in Fig. 2c, if task $n = 4$ is performed, dendrite u_4 is connected to the activation function. Using this approach, we generate a similar context-dependent modulation effect as the one proposed in [13] and expressed in Eq. (6).

However, as opposed to the approach from Iyer et al. that necessitates a dedicated *k*WTA layer, we can exploit dead neurons in our TTFS-encoded network to intrinsically implement a gating mechanism. Indeed, a dead neuron is equivalent to a real value of zero in a network of artificial neurons, acting as a gating mechanism similar to a *k*WTA layer. Moreover, the dendritic-dependent spike time delay mechanism can push neurons that otherwise would have been active in the dead zone, thereby forming a dynamic context-dependent gating mechanism. A pictorial representation of the proposed neuron model and the activation function is provided in Fig. 2d.

Following the introduction of active dendrites, the model contains two learnable parameters $\Theta = (W^l, u^l)$, which are modified by BP to minimize a spike-time-based cross-entropy loss function as proposed in [18]. With the new model equations, the gradients expressed in Section II-A thus become

$$\frac{\partial t_j^l}{\partial W_{ij}^l} = \frac{\partial t_j^l}{\partial V(t_j^l)} \frac{\partial V(t_j^l)}{\partial W_{ij}^l} = \frac{t_i^{l-1} - t_j^l + f(u_{jn}^l)}{\sum_i^I W_{ij}^l}, \tag{9}$$

$$\frac{\partial t_j^l}{\partial u_{jn}^l} = \frac{\partial t_j^l}{\partial V(t_j^l)} \frac{\partial V(t_j^l)}{\partial u_{jn}^l} = \frac{f'(u_{jn}^l)}{\sum_i^I W_{ij}^l}, \tag{10}$$

where Eq. (10) defines the direction of steepest descent for the dendritic segment u_{jn}^l. Note that the downstream gradient expressed in Eq. (5) remains unaffected by the introduction of active dendrites.

B. Digital hardware architecture

The proposed architecture is inspired by Gyro [27], a digital event-driven architecture supporting multiple fully-connected layers of spiking neurons, as depicted in Fig. 3a for three layers. Each layer of Gyro consists of a *memory module* storing synaptic connections between adjacent layers, a *neural cluster* containing parallel instances of the neuron processing unit (NPU), *input/output queues* storing the address of spiking pre-synaptic and post-synaptic neurons, and *control modules* implemented as finite-state-machines (FSMs)

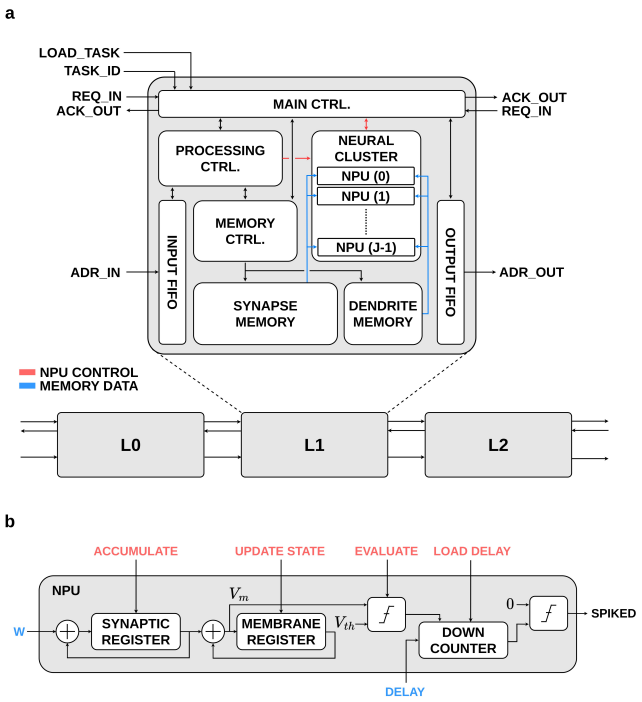

Fig. 3. **Digital Hardware Architecture. a** Architecture of a layer containing J parallel neuron processing units (NPUs). **b** Block diagram of an NPU implementing the dynamics of the proposed neuron model. The synaptic register accumulates the synaptic strength (i.e., W) received at each timestep. The membrane register stores the membrane voltage of the neuron. It integrates the value stored in the synaptic register at the end of each timestep. The down counter is loaded with the dendritic delay of the current task. Following a threshold crossing event, it starts decrementing at each timestep. When its value reaches zero, it raises the *SPIKED* flag.

controlling the communication between layers and the interaction between modules within a layer.

Adjacent layers communicate through a 4-phase handshake protocol orchestrated by the main controller. At each timestep, the address of spiking pre-synaptic neurons is pushed to the input queue of the succeeding layer, followed by a request signal to initiate processing. In response to this request, the processing controller pops the addresses and triggers the memory controller to access the synapse memory at the received locations. The addresses of the synaptic memory are aligned with the indices of pre-synaptic neurons and each address contains all the synaptic weights connecting a pre-synaptic neuron to all postsynaptic ones. The synaptic memory has a word width of $J \times Q_s$, with J the number of post-synaptic neurons and Q_s the signed fixed-point precision of synaptic parameters. This memory organization ensures that the membrane potential of each NPU can be updated in parallel after one pre-synaptic spike.

Unlike Gyro, which implements a leaky-integrate-and-fire neuron model as an NPU, our architecture implements the dendrite-enhanced neuron model proposed in Section III-A. To achieve this goal, we propose a novel NPU, illustrated in Fig. 3a, and incorporate an additional memory module, named *DENDRITE MEMORY*, for storing the dendritic delay of each NPU in a layer. The dendrite memory has a depth equivalent to the number of tasks, where each address has a width of $J \times Q_d$, with Q_d being the unsigned fixed-point precision of the dendritic delay. This memory organization ensures that the dendritic delays for all neurons can be loaded concurrently.

Specifically, when a layer receives the *TASK_ID* indicating the current task along with a *LOAD_TASK* signal, the main controller triggers the memory controller to access the dendrite memory. The memory output is then directed to the delay input port of each NPU and loaded into the corresponding down counter.

IV. Results

A. Software simulations

Our model's performance was evaluated in a single-head task-incremental scenario using the Split MNIST dataset, a popular benchmark for continual learning [28]. It consists in sequentially training a neural network to solve 5 different tasks. Each task requires the network to discriminate between two consecutive digits of the MNIST dataset, e.g. 0 and 1, 2 and 3, etc. (Fig. 4a). Each digit is temporally encoded by transforming the pixel intensities I_i into spike times as $t_i^{input} = T_{max}(I_{max} - I_i)/I_{max}$, where $I_{max} = 256$ and $T_{max} = 450$. To evaluate the effectiveness of our solution, we conducted three experiments:

1) *Interleaved without dendrites*: Establishing the upper performance bound using a TTFS-encoded network with interleaved task presentation (i.e., no catastrophic forgetting).
2) *Sequential without dendrites*: Establishing the lower performance bound on the same TTFS-encoded network but with sequential task presentation (i.e., catastrophic forgetting).
3) *Sequential with dendrites*: Incorporating active dendrites in the TTFS-encoded network and presenting tasks in sequential order.

The network architecture of experiments (1) and (2) is 784-403-403-2, while the network architecture in experiment (3) is 784-400-400-2, with all neurons in the hidden layers enhanced with active dendrites. The additional neurons in the hidden layers of the first two experiments ensure that all experiments have an equivalent number of learnable parameters. Similarly to previous works [28], we use the average test accuracy across all tasks at the end of training as a performance metric. All experiments use the Adam optimizer with a learning rate of 3e-4 for all learnable parameters and $S = 4$.

Each experiment was repeated for 5 different seeds and the results averaged. A summary of the performance for each experiment is provided in Fig. 4b. Experiment (3) shows a reduction of 8.7 accuracy points from the upper bound, while experiment (2) shows a more substantial reduction of 27.6 accuracy points. For a clearer visualization of the proposed solution's effectiveness in mitigating catastrophic forgetting, in Fig. 4c we plot the test accuracy for each task over the training duration. This figure illustrates the capability of the model enhanced with active dendrites to retain information on previous tasks as new tasks are being added.

B. FPGA implementation

The proposed hardware architecture is implemented in the programmable logic of a Xilinx Zynq-7020 SoC FPGA. For monitoring, controlling and configuration purposes, the hardware architecture is connected to the processing system (PS) of the SoC using two AXI buses. The AXI buses are controlled from a Python environment running on the PS. Once an output spike is generated, the output address and spike time is written back to the memory of the PS.

To limit the required on-chip memory, the synaptic weights and dendritic delays need to be quantized. Following full-precision training on a GPU cluster, the synaptic weights of the experiment (3) are quantized to 4-bit signed fixed-point, that is, $Q_s = 4$ while the dendritic delays are quantized to 8-bit unsigned fixed-point, that is $Q_d = 8$. We use 11-bit signed fixed-point to represent the membrane potentials of the neurons. We deploy the quantized network in both software simulation and in the FPGA. For the Split MNIST dataset, the FPGA implementation and the software simulation both achieve 80.0% accuracy. The FPGA implementation matches the software simulation for all tasks and all samples, showcasing an average inference time of 37.3 ms for each image. Our design uses 93.2% of LUTs, 35.3 % of flip-flops and 29.3 % of BRAMs.

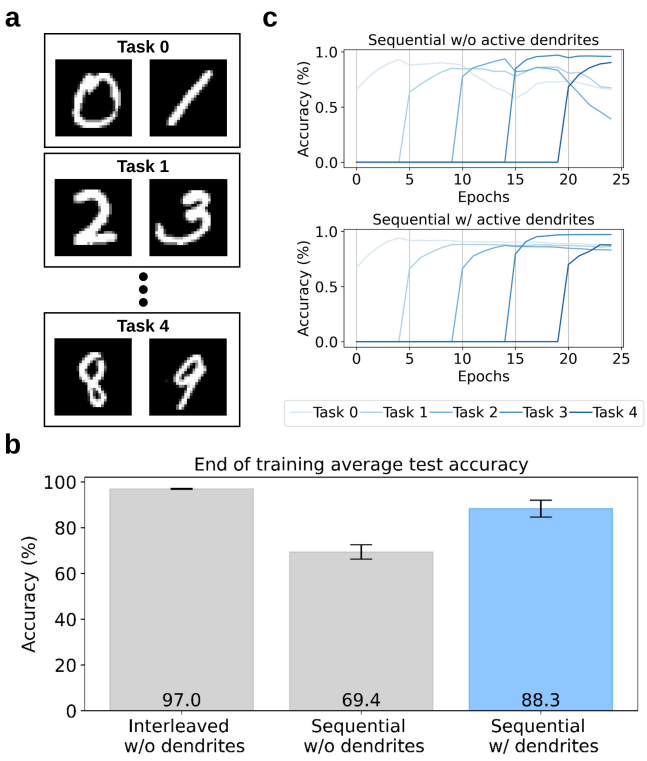

Fig. 4. **Split-MNIST setup and results. a** Example of three distinct tasks, each aiming to differentiate between two digits. **b** Test accuracy for each task over training time for the model without active dendrites (top) and with active dendrites (bottom). Note that a new task is introduced every 5 training epochs. **c** Average accuracy across tasks at the end of training for three experiments.

V. Conclusions

In this paper, we have introduced a novel TTFS-encoded SNN model enhanced with active dendrites, which can mitigate the problem of catastrophic forgetting. We demonstrated competitive performance on the standard Split MNIST dataset, showcasing an end-of-training accuracy of 88.3% across all tasks. Specifically, the network enhanced with active dendrites shows a reduction of only 8.7 accuracy points from the upper bound, while the same model without active dendrites shows a reduction of 27.6 accuracy points. Additionally, we proposed a novel digital hardware architecture that paves the way toward the deployment of continual-learning devices at the edge. Our proposed architecture has an average inference time of 37.3 ms and a test accuracy of 80.0% when deployed on a Xilinx Zynq-7020 SoC FPGA.

Acknowledgment

This publication is funded in part by the project NL-ECO: Netherlands Initiative for Energy-Efficient Computing (with project number NWA. 1389.20.140) of the NWA research programme Research Along Routes by Consortia which is financed by the Dutch Research Council (NWO).

REFERENCES

[1] M. McCloskey *et al.*, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[2] D. E. Rumelhart *et al.*, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[3] H. Robbins *et al.*, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[4] O. I. Abiodun *et al.*, "Comprehensive review of artificial neural network applications to pattern recognition," *IEEE access*, vol. 7, pp. 158 820–158 846, 2019.

[5] A. Krizhevsky *et al.*, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[6] Z.-Q. Zhao *et al.*, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[7] T. Brown *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[8] T. Mikolov *et al.*, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[9] S. Schneider *et al.*, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[10] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[11] F. Zenke *et al.*, "Continual learning through synaptic intelligence," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup *et al.*, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 3987–3995. [Online]. Available: https://proceedings.mlr.press/v70/zenke17a.html

[12] N. Y. Masse *et al.*, "Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization," *Proceedings of the National Academy of Sciences*, vol. 115, no. 44, pp. E10 467–E10 475, 2018.

[13] A. Iyer *et al.*, "Avoiding catastrophe: Active dendrites enable multi-task learning in dynamic environments," *Frontiers in neurorobotics*, vol. 16, p. 846219, 2022.

[14] H. Shin *et al.*, "Continual learning with deep generative replay," *Advances in neural information processing systems*, vol. 30, 2017.

[15] G. M. Van de Ven *et al.*, "Brain-inspired replay for continual learning with artificial neural networks," *Nature communications*, vol. 11, no. 1, p. 4069, 2020.

[16] C. Frenkel *et al.*, "Bottom-up and top-down approaches for the design of neuromorphic processing systems: Tradeoffs and synergies between natural and artificial intelligence," *Proceedings of the IEEE*, 2023.

[17] ——, "A 28-nm convolutional neuromorphic processor enabling online learning with spike-based retinas," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1–5.

[18] M. Zhang *et al.*, "Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 5, pp. 1947–1958, 2021.

[19] S. M. Bohte *et al.*, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1-4, pp. 17–37, 2002.

[20] H. Mostafa, "Supervised learning based on temporal coding in spiking neural networks," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 7, pp. 3227–3235, 2017.

[21] S. R. Kheradpisheh *et al.*, "Temporal backpropagation for spiking neural networks with one spike per neuron," *International Journal of Neural Systems*, vol. 30, no. 06, p. 2050027, 2020.

[22] I.-M. Comşa *et al.*, "Temporal coding in spiking neural networks with alpha synaptic function: learning with backpropagation," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 10, pp. 5939–5952, 2021.

[23] L. Lapicque, "Recherches quantitatives sur l'excitation electrique des nerfs," *J Physiol Paris*, vol. 9, pp. 620–635, 1907.

[24] J. Hawkins *et al.*, "Why neurons have thousands of synapses, a theory of sequence memory in neocortex," *Frontiers in neural circuits*, p. 23, 2016.

[25] Y. Yoshimura *et al.*, "Properties of horizontal and vertical inputs to pyramidal cells in the superficial layers of the cat visual cortex," *Journal of Neuroscience*, vol. 20, no. 5, pp. 1931–1940, 2000.

[26] N. Takahashi *et al.*, "Active dendritic currents gate descending cortical outputs in perception," *Nature Neuroscience*, vol. 23, no. 10, pp. 1277–1285, 2020.

[27] F. Corradi *et al.*, "Gyro: A digital spiking neural network architecture for multi-sensory data analytics," in *Proceedings of the 2021 Drone Systems Engineering and Rapid Simulation and Performance Evaluation: Methods and Tools Proceedings*, 2021, pp. 9–15.

[28] G. M. van de Ven *et al.*, "Three types of incremental learning," *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1185–1197, 2022.