Spiking Neural Encoding Schemes and STDP Training Algorithms for Edge Computing

Honghao Zheng
Department of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA, USA
zhenghh@vt.edu

Yang Yi
Department of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA, USA
yangyi8@vt.edu

Abstract

To enhance real-time data processing, edge computing is utilized in a wider and wider range of applications. For the areas that require large bandwidth and low latency, edge computing even becomes a must. For instance, in the communication area, spectrum sharing within multiple users requires high accuracy of spectrum using prediction as well as low latency. For such tasks, neuromorphic computing, especially spiking neural networks (SNNs), can be a potential method because of its power and silicon area efficiency. In this paper, we have discussed various kinds of spiking neural encoding schemes and their integrated circuit (IC) implementations. We have also summarized the pair-based STDP and the triplet-based STDP learning rule, their mathematical models, and the triplet-based reconfigurable circuit implementation. The Pytorch simulation of different encoding schemes working with two STDP rules for the MNIST and a dynamic spectrum sensing dataset is also presented. It shows that multiplexing ISI-phase encoder can achieve at most 8.9% higher accuracy than other encoders, and TSTDP provides 2.7% higher accuracy than PSTDP for the MNIST dataset. What's more, for the task of spectrum sensing for edge computing, the multiplexing encoding is also 4.3% more accurate, and TSTDP is 0.3% more accurate for the spectrum utilization prediction.

Keywords: spiking neural encoding, STDP training, spectrum sensing, integrated circuit

ACM Reference Format:
Honghao Zheng and Yang Yi. 2023. Spiking Neural Encoding Schemes and STDP Training Algorithms for Edge Computing. In *The Eighth ACM/IEEE Symposium on Edge Computing (SEC '23), December 6–9, 2023, Wilmington, DE, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3583740.3626816

SEC '23, December 6–9, 2023, Wilmington, DE, USA

ACM ISBN 979-8-4007-0123-8/23/12.
https://doi.org/10.1145/3583740.3626816

1 Introduction

Edge computing has revolutionized how data is processed, enabling real-time processing capabilities closer to the data source and significantly reducing data transmission times and energy consumption. Rather than converting data between sensors and storage or storage and processor, processing data closer to the data source will eliminate most of the latency between them. What's more, the bandwidth of the whole system can also be highly improved with the short route of the data transmission. Last but not least, local data processing also helps with data security and robustness. Without the procedure of transmitting and processing data on the cloud, information can gain more privacy and have fewer errors [3]. With the abovementioned advantages, a lot of applications, especially the Internet of Things (IoTs) are utilizing this concept to improve their performance.

To implement data processing closer to the data source, the processor's power and area limitation is relatively more severe than other computing units in the cloud or data center. Thus, compared with the conventional von Neumann structure, artificial intelligence computing units are more potential candidates for such situations. Motivated by the structure of biological neural systems, SNN was introduced as a substitute for conventional Artificial Neural Networks (ANN) because they more accurately replicate the functioning of biological neural structures [6]. In SNNs, information is conveyed only when the membrane potential surpasses a specific threshold, leading to the transmission of information as spikes. This unique characteristic, energy efficiency, and parallel processing capabilities make SNNs a viable option for handling tasks that require extensive data processing, such as image analysis. For instance, Intel's SNN processor, *Loihi* [5], is capable of categorizing objects in a 3D environment while consuming only 0.001 times the energy of a conventional computer.

In neuromorphic computing systems, analogous to biological neural systems, signals are shown as spikes. Hence, a spike encoder is a crucial component of a neuromorphic computing system. To comprehensively grasp the workings of spiking information processing, it is imperative to investigate the neural encoding schemes. Such schemes involve the transformation of raw sensory data into a series of spike trains, which downstream processing units can then interpret [11]. Broadly, there are two primary categories of encoding schemes: rate encoding and temporal encoding. Rate encoding is a method that associates input data with the number of spikes observed within a specified time frame. Its straightforward implementation makes rate encoding the preferred choice in software and hardware applications. However, this approach has a significant limitation - it results in low data density because it only utilizes the firing rate to transmit information while disregarding the temporal patterns of spikes [14]. Conversely, temporal encoding captures information using the timing patterns of spikes, thereby leveraging both the firing rate and the timing of spikes for information representation.

For the effective execution of applications based on neural networks, it is crucial to have efficient training algorithms and synaptic circuits. Numerous algorithms, such as surrogate gradients and spike-timing-dependent plasticity (STDP), have been explored for training SNNs. Beyond the fundamental STDP rule exists a more sophisticated triplet STDP (TSTDP) variant [10]. This rule considers a sequence of spikes instead of just a single pair, enabling a more precise emulation of intricate biological neural processes. Additionally, even within the pair-based STDP (PSTDP) rules, there exist variants beyond the asymmetric rule.

In the area of communication, the utilization of spectrum bands has always been an important topic since the spectrum bands that can be used for a certain system are limited [4, 8]. The suboptimal spectrum utilization efficiency results from the presence of unused subcarriers. To address this problem, it is essential for secondary users to access those under-utilized subcarriers. This necessitates monitoring the spectrum utilization of primary users via spectrum sensing.

The major contributions of this work are summarized as follows:

- We have investigated and summarized different encoding schemes and implemented and discussed their integrated circuit (IC) schematics.
- Both the PSTDP and TSTDP training algorithms and TSTDP reconfigurable IC implementations are also introduced
- Simulations of different encoding schemes and training algorithms for the spectrum-sharing communication application as well as the image classification tasks are also executed. The multiplexing encoding scheme has achieved 4.3% of better accuracy in the application of spectrum prediction compared to other encoding schemes. What's more, the triplet STDP learning rule has been proven to have a 0.3% higher prediction correct rate.

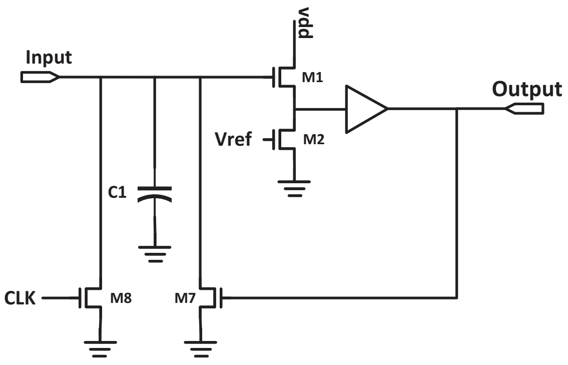

Figure 1. Circuit schematic of rate encoder.

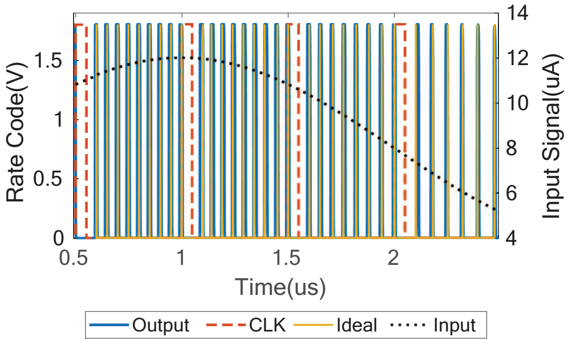

Figure 2. Simulation and ideal results of rate encoder.

2 Spiking Neural Encoding Schemes and Their IC Implementations

2.1 The Rate Encoder Circuit and Simulation Result

Rate encoding, Time to First Spike (TTFS) encoding, and Interspike Interval (ISI) encoding rank among the top three prevalent spiking codes. With rate encoding, input data is translated into a spike rate within a defined sampling window, meaning that a larger input corresponds to a higher number of spikes in that window. Owing to its straightforward nature, rate encoding is the most commonly adopted code. However, its oversight of the temporal patterns in the spikes results in a less efficient information transfer [1].

The schematic of the rate encoder is depicted in Fig. 1. Once the CLK signal resets the voltage over the membrane capacitor C1 via the switch transistor M8, an encoding window is initiated. The voltage over the membrane capacitor C1 rises as the input current is input. If the membrane voltage surpasses the reference voltage Vref, a spike is emitted through the buffer. This emitted spike also activates the switch transistor M7, resetting the membrane voltage to its baseline, thereby restarting the integration process. There's a linear correlation between the number of spikes in the sampling window and the input current. Fig. 2 demonstrates the ideal result of the rate encoding and the simulation result

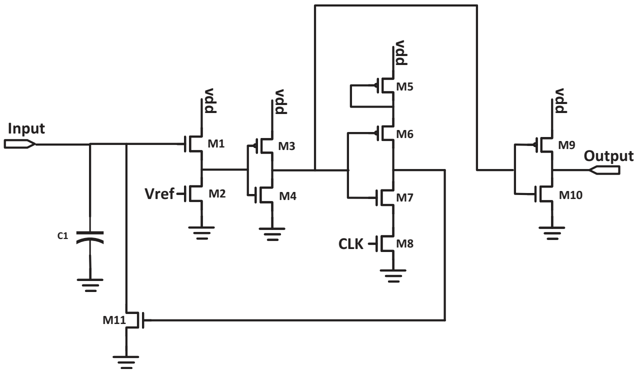

Figure 3. Circuit schematic of TTFS encoder.

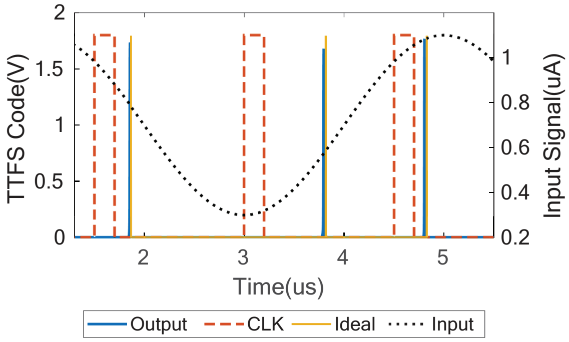

Figure 4. Simulation and ideal results of TTFS encoder.

of the rate encoder circuit. Similar relations have been observed. A higher input current results in more spikes within the sampling window, whereas a lower input current leads to fewer spikes.

2.2 The TTFS Encoder Circuit and Simulation Result

As the schematic of the TTFS encoder shown in Fig. 3 [2], when the CLK signal comes, it resets the membrane voltage using the switch transistor M11 and initiates the charge integration process. As the voltage across the membrane capacitor C1 increases, the voltage at the transistor M1's source also rises, controlled by Vref. Once this voltage surpasses the threshold voltage of the inverter made up of M3 and M4, the output transitions to a digitally high state. The four-transistor clock-controlled inverter immediately sends a high feedback signal to switch M11, resetting the membrane voltage to its baseline. Consequently, the encoder's output showcases a singular spike rather than a prolonged high digital square wave. Furthermore, the feedback signal remains high until the next CLK signal, ensuring only one spike appears within a given sampling window. Fig. 4 illustrates an inverse proportion between the time difference and the input current. As the input increases, the spike approaches the CLK signal more closely. The simulated outcome closely aligns with the ideal output.

2.3 The ISI Encoder Circuit and Simulation Result

This section discusses the ISI encoder's parallel structure [15]. While incorporating more neurons in the ISI encoder increases spikes within a single encoding window, it simultaneously elevates power usage and expands the design footprint. Consequently, we'll focus on the two-neuron parallel structure of the ISI encoder circuit here. Fig. 5 depicts the encoder's schematic. Both neurons operate under the same CLK signal and share an identical encoding window. Given that their input currents are matched, their charge integration rates align. The distinguishing factor between these neurons lies in their varied reference voltages, causing them to generate spikes at distinct moments. Subsequently, an OR gate amalgamates these two spikes, producing a dual-spike train, thereby translating the input data into time intervals between spikes. As demonstrated in Fig. 6, the ISI encoder's simulation outputs match the encoder's ideal outputs, indicating the circuit's functionality has achieved the ISI encoder's requirements.

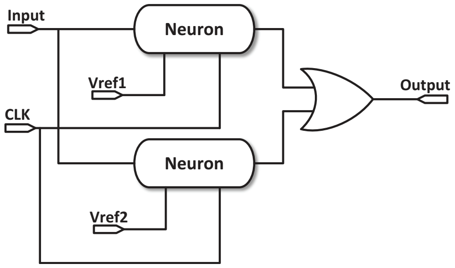

Figure 5. Circuit schematic of ISI encoder.

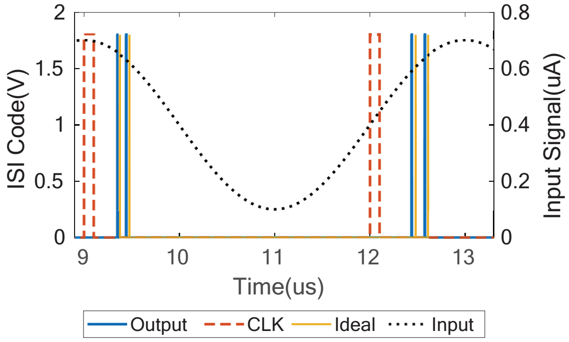

Figure 6. Simulation and ideal results of ISI encoder.

2.4 The TTFS-phase Encoder Circuit and Simulation Result

Originally identified in biological neural systems, multiplexing encoding schemes combine various neural codes—particularly

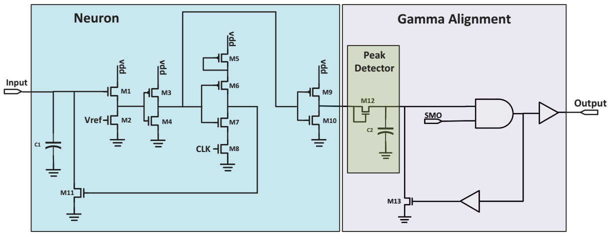

Figure 7. Circuit schematic of TTFS-phase encoder.

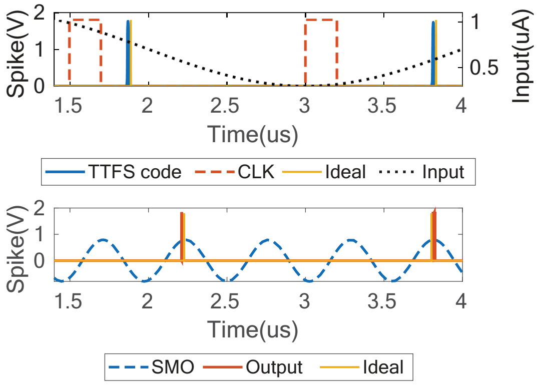

Figure 8. Simulation and ideal results of TTFS-phase encoder.

those operating on distinct time scales—to enhance data capacity. The multiplexing TTFS-phase encoding technique adjusts the TTFS-encoded spikes to align with the immediate local maximum of their respective SMOs. Our design, illustrated in Fig. 7 [17], incorporates a single channel, leading the TTFS-phase encoder to use only one SMO. Firstly, the signals will be processed by a Neuron block, also known as the TTFS encoder, to provide TTFS-coded spikes. To facilitate the spike-shifting operation, a gamma alignment block is integrated. Within this block, an upcoming spike is captured and sustained by a peak detector. The spike voltage remains maintained across the capacitor through a diode-linked transistor. Subsequently, as the local maximum of the SMO is reached, an AND gate releases a spike, which, after stabilization by a buffer, is emitted. Simultaneously, this spike actuates the switch transistor, resetting the held voltage to its baseline, where it remains until the arrival of the next spike.

Fig. 8 displays the ideal and simulated signal flows within the TTFS-phase encoder. The figure's upper section portrays the TTFS encoding function, whereas the lower section outlines the gamma alignment procedure. Once processed by the TTFS neuron, the current signal transitions into spikes. Within the gamma alignment block, these TTFS spikes are then repositioned to align with the subsequent local peak of the SMO.

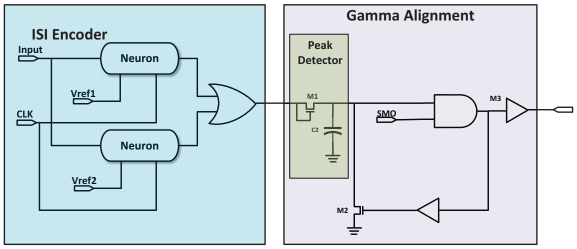

Figure 9. Circuit schematic of ISI-phase encoder.

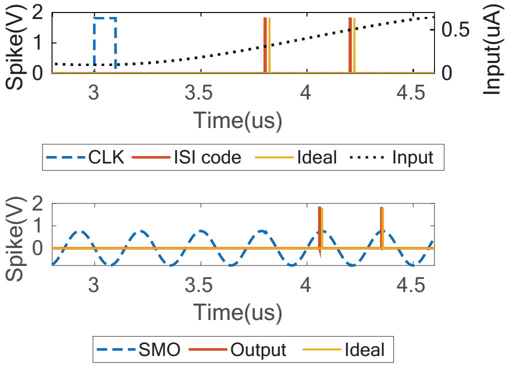

Figure 10. Simulation and ideal results of ISI-phase encoder.

2.5 The ISI-phase Encoder Circuit and Simulation Result

Much like the TTFS-phase encoder, the ISI-phase encoder incorporates both an ISI encoder and a gamma alignment block [16], as shown in Fig. 9. The gamma alignment block's role is to align the expanded spikes with the SMO's local peak for the spike train within a single sampling window. The ISI encoder uses two neurons, producing two spikes within one encoding window. As a result, the SMO frequency ought to be increased. If not, there's a risk that both spikes within the same encoding window might align with the same local peak, resulting in a single spike in the sampling window. Fig. 10 depicts that the simulation results are very close to the ideal ISI-phase encoded spike trains.

3 STDP Training Algorithms of Online Training for Edge Computing

To meet the demands of edge computing, neuromorphic computing systems must possess online training capabilities. By incorporating online training, neural networks can bypass the conventional offline training typically conducted on cloud servers. In SNNs, a range of algorithms are employed for training purposes. Notably, STDP stands out as a promising choice. This algorithm adjusts the synaptic weights based on the relative timing of spikes.

3.1 Pair-based STDP Learning Algorithm

The asymmetric pair-based STDP rule is the most direct and widely adopted example [13]. In this particular rule, weights are increased when the spike time aligns with the direction of spike propagation, a phenomenon known as long-term potentiation (LTP). Conversely, if the post-neuron spike occurs before the pre-neuron spike, indicating a weaker relationship between the two neurons, the synaptic weight is reduced, a process termed long-term depression (LTD). The correlation between weight alteration and the time difference is described as follows:

$$\Delta W = \begin{cases} A^{+}e^{-(t_{post}-t_{pre})/\tau}, & t_{post}-t_{pre} > 0 \\ -A^{-}e^{(t_{post}-t_{pre})/\tau}, & t_{post}-t_{pre} < 0. \end{cases} \tag{1}$$

Let t_{pre} and t_{post} represent the pre-neuron and post-neuron firing times, respectively. A^{+} and A^{-} denote the peak values of potentiation and depression, with the same magnitude but opposite signs. The time constant, denoted by τ, determines the decay rate for both potentiation and depression. Notably, the potentiation and depression values exhibit an exponential relationship with the differences in spike timings. This ensures that closely timed spikes significantly influence the weight, while those spaced farther apart exert a minimal impact on the weight.

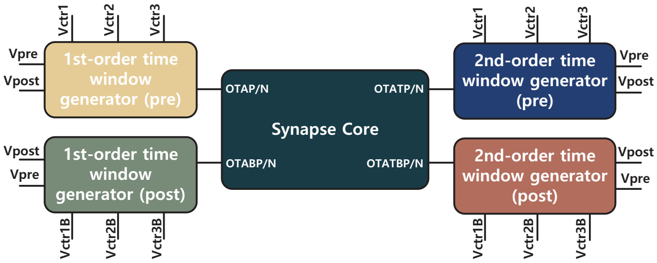
Vctr1
Vctr2
Vctr3
Vpre
Vpost
1st-order time window generator (pre)
Vpost
Vpre
1st-order time window generator (post)
Vctr1B
Vctr2B
Vctr3B
OTAP/N
OTATP/N
Synapse Core
OTABP/N
OTATBP/N
Vctr1
Vctr2
Vctr3
2nd-order time window generator (pre)
Vpre
Vpost
2nd-order time window generator (post)
Vpost
Vpre
Vctr1B
Vctr2B
Vctr3B

Figure 11. Triplet reconfigurable STDP circuit structure.

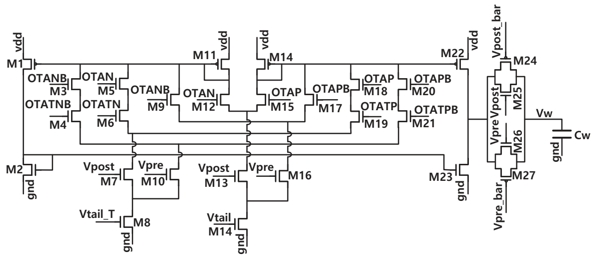

Figure 12. Circuit schematic of the synapse core.

3.2 Triplet-based STDP Learning Algorithm and TSTDP Reconfigurable Circuit Implementation

As discussed above, the Pair STDP rule (PSTDP) considers two spikes and modifies the synaptic weight based on their time difference. In contrast, the Triplet STDP rule (TSTDP) factors in three spikes [7]. The spike combinations could be either pre-post-pre or post-pre-post. These spike combinations illustrate how the timing differences between spikes are used to adjust synaptic weights. Different from the pair-based STDP training algorithm, the triplet-based STDP training algorithm represents a more complicated math model:

$$\Delta W = \begin{cases} A_{1}^{+}e^{(\frac{-\Delta t_1}{\tau_1^{+}})} + A_{2}^{+}e^{(\frac{-\Delta t_2}{\tau_2^{+}})}e^{(\frac{-\Delta t_1}{\tau_1^{+}})} \\ -A_{1}^{-}e^{(\frac{\Delta t_1}{\tau_1^{-}})} - A_{2}^{-}e^{(\frac{-\Delta t_3}{\tau_2^{-}})}e^{(\frac{\Delta t_1}{\tau_1^{-}})} \end{cases} \tag{2}$$

where A_1 and A_2 denote the potentiation and depression parameters, respectively, while Δt_1 signifies the time difference between the pre-spike and post-spike. As for Δt_2, it is defined as $t_{post}(n)$ˇ$t_{post}(n-1)$, representing the time interval between two consecutive post spikes. Here, n stands for a specific time step, and n − 1 indicates the immediately preceding time step. In a similar vein, Δt_3 illustrates the time gap between two pre-spikes. When comparing the TSTDP mathematical model with the PSTDP equation, it becomes evident that the TSTDP encompasses higher-order terms in both its depression and potentiation formulas.

In our quest to harness the strengths of the TSTDP learning rule across various applications, we have introduced an innovative STDP learning circuit. This circuit seamlessly integrates diverse STDP learning rule shapes into the triplet STDP framework. By doing so, it capitalizes on the inherent advantages of TSTDP, such as its frequency effects, making it adaptable for a range of applications. As demonstrated in Fig. 11, Our group has designed the triplet reconfigurable STDP circuit [18]. The reconfigurable triplet STDP circuit is governed by six digital signals, facilitating a switch among various learning rule shapes. These signals are classified into controls for pre-spikes and post-spikes. The circuit design comprises five distinct sections. Two sections focus on generating time windows for weight changing caused by two spikes, while another two are dedicated to weight changing caused by three spike trains. The circuit's fifth part termed the synapse core, leverages voltage signal flows from the preceding four sections to modify the synapse's weight.

The time window generators will take account in both the pre-and post-spikes as well as the control signal and output OTA inputting signals. While the 1st-order generator uses the current spikes, the two 2nd-order generators will hold the pre- and post-spikes for one clock cycle, respectively, and then utilize the spikes for OTA signal emitting. As depicted in Fig. 12, the synapse core gets the OTA signals from the generators and adjusts the weight.

The OTA signals from the window generators come to the synapse core in pairs. The two signals in the pair will have voltage differences at the spiking time. With this difference, the synapse core will be able to either inject or drain charge from the weight capacitor at the time of spikes. Moreover, the Vtail and Vtail_T can be used to adjust the charge amount

Table 1. Accuracy of different encoding schemes training with both PSTDP and TSTDP learning algorithms for MNIST and dynamic spectrum sensing datasets

MNIST	Rate	TTFS	ISI	TTFS-phase	ISI-phase
PSTDP	87.2%	88.9%	91.8%	92.5%	93.8%
TSTDP	87.6%	91.1%	92.8%	93.2%	96.5%
Spectrum	Rate	TTFS	ISI	TTFS-phase	ISI-phase
PSTDP	87.4%	89.4%	90.2%	90.3%	92.3%
TSTDP	88.3%	89.9%	90.1%	91.0%	92.6%

injected or drained from Cw. Thus, the circuit can achieve different ways of adjusting weight voltage.

4 Performance of Different Encoding Schemes Working with Both STDP Training Algorithms

To utilize the SNN classification in edge computing, especially in communication applications, it has to be proven to have high task efficiency and accuracy. To evaluate the efficacy of various encoding schemes working with both STDP training algorithms, we implemented spiking neural networks with distinct encoders and trained by both STDP rules using PyTorch and the SpykeTorch simulator. Our experiments employed the MNIST dataset along with the spectrum sensing dataset. The latter was initially introduced in a dynamic spectrum-sharing system to address the spectrum scarcity challenges faced by 5G networks. Although Multiple-Input, Multiple-Output Orthogonal Frequency-Division Multiplexing (MIMO-OFDM) technologies enhance spectral efficiency, unused subcarriers still result in less-than-optimal spectrum utilization. To circumvent this, secondary users must tap into these under-utilized subcarriers. This demands monitoring the spectrum consumption of primary users via spectrum sensing. Studies have confirmed that employing spiking neural networks to predict under-utilized spectrum bands is a notably energy-efficient approach [9, 12].

Table II has demonstrated the classification accuracy of various encoders working with both algorithms for the two datasets. For the MNIST dataset, the multiplexing ISI-phase encoding scheme achieves the highest accuracies with pair-based STDP and triplet-based STDP learning rules. Compared with other encoding schemes, the ISI-phase encoding yields at most 6.6% and 8.9% accuracy working with PSTDP and TSTDP, respectively. What's more, it also shows that the TSTDP learning can offer 2.7% higher accuracy than the PSTDP rule.

As for the dynamic spectrum sensing dataset, the ISI-phase encoder achieves 92.6% accuracy when collaborating with TSTDP, which is 0.3% higher than PSTDP and 4.3% higher than other encoders. It proves that the multiplexing encoding schemes can achieve better classification performance and the triplet-based STDP learning is able to provide higher accuracies than conventional STDP training algorithm for both image classification and communication applications. With that, the need to transmit data to the center server and wait for the processing can be avoided. Since the data are processed close to its source, the latency of the spectrum switching is drastically reduced, resulting in a much more efficiently used spectrum band.

5 Conclusion

In this paper, we have summarized two key features of the SNN for the application of communication systems for edge computing. One is the spike neural encoding and the other is the STDP training algorithm. For the spiking encoding, we have discussed various kinds of encoding schemes and their integrated circuit (IC) implementations. The results indicate that while rate encoding is simple, its data capacity is limited. In contrast, temporal codes offer increased data capacity but lack noise robustness. Multiplexing encoding schemes strike a balance by delivering high data capacity and robustness; however, their complexity results in significant power and area expenses. We have also talked about two various STDP training algorithms. One is the pair-based STDP rule, and the other is the triplet-based STDP learning rule. The mathematical models of the algorithms are explained, and the circuit implementation of the triplet-based STDP circuit with the reconfigurable feature is also demonstrated in the paper. The Pytorch simulation of different encoding schemes working with two STDP rules for the MNIST and a dynamic spectrum sensing dataset is also presented. It shows that multiplexing ISI-phase encoder can achieve at most 8.9% higher accuracy than other encoders and TSTDP provides 2.7% higher accuracy than PSTDP for the MNIST dataset. What's more, for the task of spectrum sensing for edge computing, the multiplexing encoding is also 4.3% more accurate, and TSTDP is 0.3% more accurate for the spectrum utilization prediction. This result has proved that SNNs can be potential candidates for communication applications, especially with multiplexing encoding and trained by TSTDP rules. Consequently, there's no longer a need to send data to a central server and await processing. As data is processed closer to its origin, the latency associated with spectrum switching diminishes significantly, leading to a more optimized use of the spectrum band.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation (NSF) under Grant CCF-1750450, Grant ECCS-1731928, Grant ECCS-2128594, Grant ECCS- 2314813, and Grant CCF-1937487.

References

[1] Daniel Auge, Julian Hille, Etienne Mueller, and Alois Knoll. 2021. A survey of encoding techniques for signal processing in spiking neural networks. *Neural Processing Letters* 53, 6 (2021), 4693–4710.

[2] Kangjun Bai, Qiyuan An, Lingjia Liu, and Yang Yi. 2019. A training-efficient hybrid-structured deep neural network with reconfigurable memristive synapses. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28, 1 (2019), 62–75.

[3] Keyan Cao, Yefan Liu, Gongjie Meng, and Qimeng Sun. 2020. An overview on edge computing research. *IEEE access* 8 (2020), 85714–85728.

[4] Hao-Hsuan Chang, Hao Song, Yang Yi, Jianzhong Zhang, Haibo He, and Lingjia Liu. 2018. Distributive dynamic spectrum access through deep reinforcement learning: A reservoir computing-based approach. *IEEE Internet of Things Journal* 6, 2 (2018), 1938–1948.

[5] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro* 38, 1 (2018), 82–99.

[6] Samanwoy Ghosh-Dastidar and Hojjat Adeli. 2009. Spiking neural networks. *International journal of neural systems* 19, 04 (2009), 295–308.

[7] Julijana Gjorgjieva, Claudia Clopath, Juliette Audet, and Jean-Pascal Pfister. 2011. A triplet spike-timing–dependent plasticity model generalizes the Bienenstock–Cooper–Munro rule to higher-order spatiotemporal correlations. *Proceedings of the National Academy of Sciences* 108, 48 (2011), 19383–19388.

[8] Lingjia Liu, Runhua Chen, Stefan Geirhofer, Krishna Sayana, Zhihua Shi, and Yongxing Zhou. 2012. Downlink mimo in lte-advanced: Su-mimo vs. mu-mimo. *IEEE Communications Magazine* 50, 2 (2012), 140–147.

[9] Lingjia Liu, Jianzhong Zhang, and Zhouyue Pi. 2012. Inter-cell interference avoidance for downlink transmission. US Patent 8,238,954.

[10] Jean-Pascal Pfister and Wulfram Gerstner. 2006. Triplets of spikes in a model of spike timing-dependent plasticity. *Journal of Neuroscience* 26, 38 (2006), 9673–9682.

[11] Edmund T Rolls and Alessandro Treves. 2011. The neuronal encoding of information in the brain. *Progress in neurobiology* 95, 3 (2011), 448–490.

[12] Rubayet Shafin, Lingjia Liu, Vikram Chandrasekhar, Hao Chen, Jeffrey Reed, and Jianzhong Charlie Zhang. 2020. Artificial intelligence-enabled cellular networks: A critical path to beyond-5G and 6G. *IEEE Wireless Communications* 27, 2 (2020), 212–217.

[13] Chenyuan Zhao, Qiyuan An, Kangjun Bai, Bryant Wysocki, Clare Thiem, Lingjia Liu, and Yang Yi. 2019. Energy efficient temporal spatial information processing circuits based on STDP and spike iteration. *IEEE Transactions on Circuits and Systems II: Express Briefs* 67, 10 (2019), 1715–1719.

[14] Chenyuan Zhao, Bryant T Wysocki, Yifang Liu, Clare D Thiem, Nathan R McDonald, and Yang Yi. 2015. Spike-time-dependent encoding for neuromorphic processors. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 12, 3 (2015), 1–21.

[15] Chenyuan Zhao, Yang Yi, Jialing Li, Xin Fu, and Lingjia Liu. 2017. Interspike-interval-based analog spike-time-dependent encoder for neuromorphic processors. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25, 8 (2017), 2193–2205.

[16] Honghao Zheng, Kang Jun Bai, and Yang Yi. 2023. Enabling a New Methodology of Neural Coding: Multiplexing Temporal Encoding in Neuromorphic Computing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 31, 3 (2023), 331–342. https://doi.org/10.1109/TVLSI.2023.3234514

[17] Honghao Zheng, Nima Mohammadi, Kangjun Bai, and Yang Yi. 2021. Low-power Analog and Mixed-signal IC Design of Multiplexing Neural Encoder in Neuromorphic Computing. In *2021 22nd International Symposium on Quality Electronic Design (ISQED)*. 154–159. https://doi.org/10.1109/ISQED51717.2021.9424267

[18] Honghao Zheng and Yang Yi. 2023. Enhancing SNN Training Performance: A Mixed-Signal Triplet Reconfigurable STDP Circuit with Multiplexing Encoding. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–5. https://doi.org/10.1109/ISCAS46773.2023.10181729