Low-Power Real-Time Sequential Processing with Spiking Neural Networks

Chamika Mihiranga Liyanagedera, Manish Nagaraj, Wachirawit Ponghiran, and Kaushik Roy
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47906 USA
cliyanag,mnagara,wponghir,kaushik@purdue.edu

2023 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-5109-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISCAS46773.2023.10181703

***Abstract*—The biological brain is capable of processing temporal information at an incredible efficiency. Even with modern computing resources, traditional learning-based approaches are struggling to match its performance. Spiking neural networks that "mimic" certain functionalities of the biological neural networks in the brain is a promising avenue for solving sequential learning problems with high computational efficiency. Nonetheless, training such networks still remains a challenging task as conventional learning rules are not directly applicable to these bio-inspired neural networks. Recent efforts have focused on novel training paradigms that allow spiking neural networks to learn temporal correlations between inputs and solve sequential tasks such as audio or video processing. Such success has fueled the development of event-driven neuromorphic hardware that is specifically optimized for energy-efficient implementation of spiking neural networks. This paper highlights the ongoing development of spiking neural networks for low-power real-time sequential processing and the potential to improve their training through an understanding of the information flow.**

I. Introduction

Artificial Neural Networks (ANN) have garnered much attention recently and have evolved from simple Multi Layer Perceptrons (MLPs) to full blown Deep Neural Networks (DNNs) over the last decade. The amazing prospects of such Artificial Intelligence (AI) have been demonstrated many times over through projects like AlphaGo [1] and Deepfake AI [2]. However, regardless of how impressive they seem, modern AI is still incapable of imitating the processing power of a biological brain. This difference can be attributed, among others, to the functional and structural differences between artificial and biological neural networks. As a consequence, numerous recent research studies have been focused on more biologically plausible neural networks, such as Spiking Neural Networks (SNN) [3], as an effort to bridge this computational gap. Spiking neurons though not identical, share many similarities with biological neurons, including electrical pulse driven communication [4] and internalized short term memory. These properties of SNNs have paved a pathway for a new generation of AI that can overcome the limitations of traditional ANNs.

The biological brain is exceptionally good at processing temporal information [5]. Even the simplest task of catching a ball involves processing continuous visual cues captured by the retina to control perfectly timed complex motor functions for proper hand-eye coordination. Standard feed-forward ANNs are inefficient at such tasks, as they are incapable of handling temporal information effectively. Efforts have been directed toward Recurrent Neural Networks (RNNs) [6]–[8] and Long Short Term Memories (LSTMs) [9]–[12] for processing temporal information at the expense of increased computational and training overhead. SNNs, on the other hand, are equipped to handle such temporal information more efficiently, without the assistance of complex computational units or recurrent connectivity owing to their internalized memory called the *membrane potential*. Additionally, spiking neurons can learn to establish an association between previous inputs through a tunable parameter called the *leakage*. Similar to the biological neurons, spiking neurons communicate via binary values $\{0,1\}$ representing the absence or presence of electric pulses or spikes [4]. Hence, important information is encoded not only in the spike frequency but also in the timing of the spikes. Due to these properties, SNNs are well suited for solving sequential problems such as speech recognition [13] and video processing [14].

The AlphaGo AI required 1MW of power to beat Lee Sedol in a five-game Go match in March 2016, which is 50,000 times the average power consumption of a human brain of 20W [15]. This exceptional power efficiency of the biological brain can be partly attributed to its event driven nature. SNNs adapt similar event driven computations into their architectures which allows them to be significantly compute efficient than their ANN counterpart. Unlike ANNs, where every neuron activates simultaneously, a spiking neuron activates only when an input spike is received and certain internal conditions are satisfied. This allows SNNs to have sparse activity throughout the network, leading to significant energy savings, and making them suitable for resource-constrained applications.

Even with these benefits, the adaptation of SNNs has been a bit sluggish until recent years due to challenges in training Deep SNNs. Training techniques that employ gradient-descent based back propagation are not directly applicable to SNNs due to their non-differentiable activation function. Converting a trained ANN to SNN has been a popular option [16]–[18], but this renders the network insensitive to temporal information. One popular approach is to use Backpropagation Through Time (BPTT) technique with surrogate gradients for

This work was supported in part by, Center for Brain-inspired Computing (C-BRIC), a DARPA sponsored JUMP center, Semiconductor Research Corporation (SRC), National Science Foundation, and IARPA MicroE4AI.

training SNNs [19]–[22]. This technique has been successfully employed for training SNNs for sequential tasks like optical flow prediction [23]. Another common approach for training SNNs is to employ a local learning rule like the Spike-Time-Dependant Plasticity (STDP) that arises from a biological observation [24]. Since the learning rule relies on a causal correlation between inputs and outputs of spiking neurons, it does not require gradient calculation. Multi-layer SNNs have been trained to distinguish directions in real-time through this technique [25].

In this article, we revisit the physics of spiking neurons, popular techniques on how SNNs can be trained for sequential tasks, different applications that benefit from the sequential processing capabilities of SNNs, and some evidence of low-power sequential processing on SNN hardware.

II. Dynamics of Spiking Neurons

Aiming to mimic the behavior of biological neurons, spiking neurons incorporate the concept of the internal state into their computational model. In a popular spiking neuron model [26] called the leaky integrate-and-fire (LIF) neuron, this internal state is typically referred to as the *membrane potential* $V(t)$. Suppose $X_i(t)$ are inputs from a neighboring neurons $i \in \{1, 2, ..., a\}$ to the LIF neuron at time t as shown in Fig. 1(a). The membrane potential is increased by the influence of these inputs depending on the connection strength of each input to the LIF neuron denoted by w_i. Then, an output of "1" representing a neuron spiking activity is generated if the membrane potential exceeds a defined threshold γ. Otherwise, the spiking neuron produces an output of "0" to represent no neuronal activity. Upon spike generation, the membrane potential reduces by the threshold amount to reflect a depotentiation of the biological neuron. Mathematically, the spiking neuron can be described using the following equation:

$$dV(t)/dt = -V(t)/\tau + \sum_{i=1}^{a} w_i X_i(t) - \gamma Y(t) \qquad (1)$$

where $Y(t)$ is an output of the spiking neuron at time t. Note that membrane potential decays exponentially according to a constant τ in an absence of inputs, thus it behaves like a leaky integrator of the inputs.

However, obtaining the membrane potential by directly solving the differential equation above at every t is impractical. Discretizing spiking neuron operations over small time-steps leads to a closed-form approximation as follows:

$$V[n] = \beta V[n-1] + \sum_{i=1}^{a} w_i X_i[n] - \gamma Y[n-1] \qquad (2)$$

$$Y[n] = \mathrm{UnitStep}(V[n] - \gamma) \qquad (3)$$

where n represents an index of simulation time-step, β is a decaying factor for the membrane potential, and $\mathrm{UnitStep}(\cdot)$ denotes a unit step function. For a group of spiking neurons that receive the same inputs and form a neuron layer as

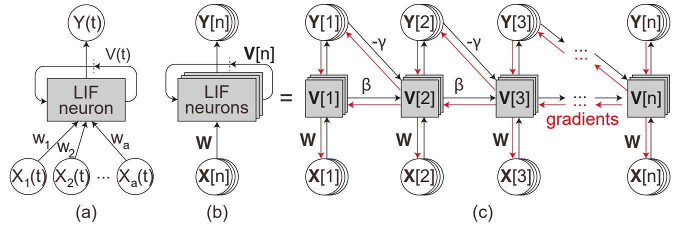

Fig. 1. (a) An example of LIF neuron which receives $X_i(t)$ as inputs and produces $Y(t)$ as an output. w_i denotes the weight of each connection to the neuron. (b) Diagram representing operations of spiking neurons that form a neuron layer in a discrete time. The neurons receive input vector $\mathbb{X}[n]$ and produce output vector $\mathbb{Y}[n]$ at time-step n. (c) Equivalent representation when the operations are unrolled into n time-steps.

illustrated in Fig. 1(b), their operations can be expressed using a similar set of equations:

$$\mathbb{V}[n] = \beta \mathbb{V}[n-1] + \mathbb{X}[n]\mathbb{W} - \gamma \mathbb{Y}[n-1] \qquad (4)$$

$$\mathbb{Y}[n] = \mathrm{UnitStep}(\mathbb{V}[n] - \gamma) \qquad (5)$$

where $\mathbb{X}[n] = (X_1[n], X_2[n], ..., X_b[n])$ represents inputs in the form of a vector and $\mathbb{Y}[n] = (Y_1[n], Y_2[n], ..., Y_b[n])$ represents the output vector with each element $Y_j[n]$ indicating an output of neuron j in the group at time-step n. Connection matrix $\mathbb{W}$ consists of element $\mathbb{W}_{ij}$ in row i and column j that indicates a connection strength between input i and spiking neuron j.

The presence of internalized memory is a characteristic SNNs share with networks like RNNs and LSTMS. However, what sets SNNs apart from the rest is their simple internal memory update mechanism and pulse like binary communication inspired by biological neurons. This simplified internal state update mechanism, though renders training more complicated, implies smaller number of parameters compared to LSTMs. Binary communication replaces the weight multiplication with simple addition operations, granting significant power savings when the inputs are zero. The benefits of having these differences are quantified in some hardware implementations discussed in Section V.

III. SNNs Training Methodology

Since the membrane potential of spiking neurons depends on the inputs at the recent time-step and the membrane potential at the previous time-step, we can recursively apply Eq. 4-5 for output computation. Operations of the spiking neurons can be unfolded in time by duplicating the computation graph of the spiking neurons for n time-steps (see Fig. 1(c)). Then the gradients can be computed and propagated backward through multiple time-steps using the BPTT technique.

Nonetheless, the BPTT algorithm cannot be simply applied due to the non-differentiability of the unit step function. The derivative of the unit step function is zero almost everywhere, thus deterring gradients from flowing backward and rendering the stochastic gradient descent technique ineffective for training SNNs. A popular and effective remedy proposed in the literature is to replace the ill-defined derivation of unit step function with the derivative of a continuously differentiable function [19]–[22]. Further analysis of the computation graph revives complex back-propagation paths to the membrane potential at each time-step (see red lines Fig. 1(c)). Several works have proposed techniques to make training numerically stable [27]–[30].

Rather than relying on the gradient for computing synaptic weight updates, another approach to train SNNs is to employ a local learning rule inspired by biological observations. For instance, with the STDP rule, the synaptic weights are adjusted based on the precise timing of the pre-and post-synaptic spikes. If there are incoming spikes to the neurons which shortly lead to desired output spikes, synaptic connections between inputs and outputs are strengthened. Otherwise, the connection strength is reduced. This method allows local information to update synaptic weights without the need for a global learning signal. Local learning rules similar to STDP has been proposed and demonstrated to learn precise correlations between input and output spikes on a synthetic task [19]. Nevertheless, training SNNs to capture temporal information better remains to be an active research problem and numerous progress has been made through understanding information flow in the computational graph.

IV. Solving Sequential Problems with SNNs

Classification of sequential data like audio and video has been a problem of interest in the machine learning community as it requires drawing temporal correlations between information that arrives at different times. This calls for neural networks to retain a memory of the past for making an accurate decision later, and thus RNNs and LSTMs have become a common choice for effectively handling the problem [9]–[12]. In recent years, improvements in SNN training methodologies leading to better sequential learning has the potential to make SNNs a preferred alternative, especially for low-power computing. Furthermore, there are domains that have unique feature representations and need more efficient ways to process them, such as the outputs of an event-based camera. Some works showcasing the advantages of SNNs in solving sequential tasks are presented below.

A. Keyword Spotting and Word Recognition

Owing to the popularity of virtual assistant technologies, Key-Word Spotting (KWS) is an emerging essential task in speech recognition. KWS searches for and identifies triggering cues (such as "Hey Siri,") for edge devices. This requires the algorithm to be continuously performed and needs to have a low computational complexity to sustain continuous operation. The authors in [31] proposed using SNNs as an alternative to traditional ANNs for KWS. They explored two different approaches of encoding: one where information is encoded in the firing rate of the spiking neurons, and another where information is encoded in the relative timing of the spiking neuron outputs. In both cases, they observed that the accuracy of identifying keywords on the TIMIT speech dataset [32] remained the same as ANNs while SNNs reduced the computational cost by 84% and memory requirement from $O(N^2)$ to $O(N)$ where N is the number of the neurons.

The authors in [33] leveraged the inherent recurrence found in spiking neurons to perform word recognition on a much bigger LibriSpeech speech dataset [34]. The authors modified the network of LIF neurons to allow multi-bit outputs (in contrast to the traditional binary outputs) to overcome the vanishing gradient problem in training SNNs. Their proposed SNNs had $2\times$ fewer parameters compared to LSTMs and required $10\times$ fewer expensive multiplication operations compared to GRUs while maintaining similar accuracy metrics as the other two networks.

B. Action Recognition from Videos

As SNNs are modeled to closely mimic biological neural networks, they are sometimes crafted to replicate a certain functionality of the brain to help neuroscientists gain a better understanding. Such constructed models were shown in some works to be helpful for recognizing actions from videos. For instance, the authors in [14] created SNNs to mimic part of the visual cortex, specifically V1 (primary visual cortex) and MT (medial temporal) regions of the human brain. Through their analysis, they observed that the synchronicity between spiking neurons and their mean firing rate correlates with temporal information sequentially presented in the video inputs. The motion maps generated from the constructed SNNs were then shown to contain essential information for recognizing actions.

In a separate effort, the authors in [35] trained SNNs with explicit recurrent connections to perform action recognition from videos. The proposed SNNs also have a mechanism to perform temporal pooling which helps drawing attention to certain important time-steps. Their results show that the SNNs are more robust to noisy frames than tradition RNNs while having a lower computational complexity.

C. Processing Event-based Camera Outputs

There has been substantial interest in developing learning algorithms for emerging bio-inspired event-based cameras. Unlike a traditional frame-based camera that captures the light intensity during a fixed interval, these new vision cameras measure changes in the light intensity at each pixel (so-called events), leading to a camera with a high operating frequency and low power consumption. Because the events are generated only at the contour during a movement and are sparse, SNNs are typically the preferred network for processing the events. We discuss two applications of SNNs for processing sequential event-based camera outputs, namely optical flow prediction and object detection & tracking.

Optical flow estimation is an important task in vision-based navigation as it provides a comprehensive understanding of a relative motion in the scene. Essentially, optical flow is an estimation of the $2D$ motion field which can be obtained by computing the spatial and temporal variations of pixel intensities. Being able to draw correlations from events that arrive sequentially is important to ensure high-quality optical flow prediction. The author in [23] proposed a hybrid neural network architecture called Spike-FlowNet to estimate the optical flow of the scene using only the event-based camera outputs. The network consists of an SNN encoder followed by an ANN decoder. The authors showed that the proposed network outperformed the ANN model in terms of computational efficiency and prediction accuracy on the MVSEC dataset [36]. This highlighted the capability of SNNs in capturing temporal correlation from naturally sparse sequential data.

The authors in [37] further improved the event-based optical flow estimation by modifying the existing training pipeline to better suit SNNs. As the optical flow estimation have been traditionally predicted by ANNs from the frame-based camera outputs, a similar training pipeline was adapted by the previous work. In this work, the authors reformulated the training loss to improve convexity and showed that the width of the surrogate gradient and weight initialization are crucial to accomplish a better optical flow estimation. This underscored the importance of information theory to suggest appropriate adjustment on the existing training methodology for emerging SNNs.

Orthogonal to the aforementioned works, authors in [38] suggested combining information from two different vision sensors for even more accurate optical flow estimation. They argued that since events only contain a limited amount of spatial information, frames obtained from a traditional camera can augment missing spatial details. The proposed hybrid architecture called the Fusion-FlowNet comprised of two branches: an SNN branch which took events as inputs and an ANN branch which took frames as inputs. These branches were later merged into an intermediate representation for optical flow estimation. The authors expected the SNN branch to capture the temporal information from events and the ANN branch to capture the spatial information necessary for improving the prediction quality. They reported a reduction in arithmetic operations compared to the ANN model while showing that the proposed fusion network estimated optical flows with similar quality.

Similar to the task of optical flow, object detection & tracking plays an important role in vision-based navigation platforms. In addition to maintaining a high accuracy of detection, it is essential for tracking the object with little to no latency with minimalenergy expense. The authors in [39] showed that by replacing max-pooling layers with spiking neurons in traditional object detection pipelines, the computational complexity of those pipelines can be reduced.

V. Evidences of Low-Power Sequential Processing with SNNs

Though the intrinsic recurrence of SNNs enables sequential information processing, it introduces an additional data structure to be stored and additional computations to be updated. Naively implementing SNNs on Von-Neumann architecture, commonly used for the feed-forward ANN inference, introduces large amount of memory accesses, leading to increased energy and power consumption. Thus, efficient computation with SNNs calls for a specifically designed hardware that minimizes the memory transactions for updating internal states and is optimized for sparse computations.

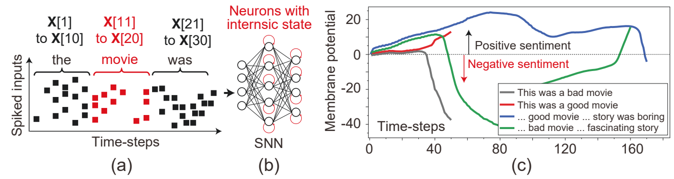

Fig. 2. (a) For the sentiment analysis, each word in a review is converted to a binary spike train before being fed to SNN. (b) Multi-layer SNN that is trained and mapped to the SNN hardware. (c) Changes in the membrane potential of the output spiking neuron at the last layer with time-steps.

An in-memory computing macro catering for SNN inference has been proposed in [40] to demonstrate the benefit of SNNs for sentiment analysis on the IMDB reviews. Here each word in a review was sequentially fed to determine whether the review was positive or negative. Fig. 2(c) depicts changes in the membrane potential of the output spiking neuron at the last layer with time-steps. The SNN implementation led to a 97.4% reduction in energy-delay product compared to the LSTM implemented on an LSTM accelerator. Due to simpler recurrent dynamics, SNN had 8.5× lower trainable parameters than LSTM. Although each word was translated to spike trains for SNN operation over multiple time-steps, the implementation achieved 5.6× higher energy efficiency per inference due to the sparse communication.

A neuromorphic research chip with computing macros and local memories for SNNs, called *Intel Loihi*, has also been developed [41] and used to evaluate SNNs for the compute efficiency and runtime on KWS application [42]. Here two-layer SNNs were trained and mapped to the hardware while equivalent ANNs were trained and mapped to a Von-Neumann architecture CPU and GPU fabricated on a similar technology. The results showed the improvement in both inference speed and energy consumption when using SNNs for real-time KWS.

VI. Conclusion

SNNs are artificial neural networks constructed with certain distinctive characteristics and are more bio-plausible than standard deep networks. They were traditionally tools for neuroscientists to decipher operations of the brain but later received more attention from machine learning practitioners due to their binary communication scheme that simplifies power-hungry multiplication into simple addition operations. Intrinsic memory of the SNNs in the form of cell membrane potential, further offers an ability to capture temporal correlation from their inputs and to solve sequential learning problems. Understanding the dynamics of spiking neurons through a computational graph has provided insight into various training methodologies, both gradient-based and gradient-free techniques. Successful training methodologies later led to the application of SNNs on several sequential tasks; some of which are discussed in this article. Two hardware implementations underlining the effectiveness of low-power real-time sequential processing with SNNs are discussed. Both showed superior compute efficiency and memory requirement over traditional RNNs and LSTMs, hence suggesting further opportunities with the bio-plausible model for computing.

References

[1] J. X. Chen, "The evolution of computing: Alphago," *Computing in Science & Engineering*, vol. 18, no. 4, pp. 4–7, 2016.

[2] M. Westerlund, "The emergence of deepfake technology: A review," *Technology Innovation Management Review*, vol. 9, no. 11, 2019.

[3] K. Yamazaki, V.-K. Vo-Ho, D. Bulsara, and N. Le, "Spiking neural networks and their applications: A review," *Brain Sciences*, vol. 12, no. 7, p. 863, 2022.

[4] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: opportunities and challenges," *Frontiers in neuroscience*, p. 774, 2018.

[5] F. Johansson, G. Hesslow, and J. F. Medina, "Mechanisms for motor timing in the cerebellar cortex," *Current opinion in behavioral sciences*, vol. 8, pp. 53–59, 2016.

[6] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee, 2013, pp. 6645–6649.

[7] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*. PMLR, 2014, pp. 1764–1772.

[8] P. Wijesinghe, C. Liyanagedera, and K. Roy, "Biologically plausible class discrimination based recurrent neural network training for motor pattern generation," *Frontiers in neuroscience*, vol. 14, p. 772, 2020.

[9] J. S. P. Giraldo and M. Verhelst, "Laika: A 5uw programmable LSTM accelerator for always-on keyword spotting in 65nm CMOS," in *ESSCIRC 2018-IEEE 44th European Solid State Circuits Conference (ESSCIRC)*. IEEE, 2018, pp. 166–169.

[10] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 3285–3292.

[11] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *European conference on computer vision*. Springer, 2016, pp. 816–833.

[12] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks," *Biomedical signal processing and control*, vol. 47, pp. 312–323, 2019.

[13] P. Wijesinghe, G. Srinivasan, P. Panda, and K. Roy, "Analysis of liquid ensembles for enhancing the performance and accuracy of liquid state machines," *Frontiers in neuroscience*, vol. 13, p. 504, 2019.

[14] M.-J. Escobar, G. S. Masson, T. Vieville, and P. Kornprobst, "Action recognition using a bio-inspired feedforward spiking network," *International journal of computer vision*, vol. 82, no. 3, pp. 284–301, 2009.

[15] V. Balasubramanian, "Brain power," *Proceedings of the National Academy of Sciences*, vol. 118, no. 32, p. e2107022118, 2021.

[16] C. M. Liyanagedera, A. Sengupta, A. Jaiswal, and K. Roy, "Stochastic spiking neural networks enabled by magnetic tunnel junctions: From nontelegraphic to telegraphic switching regimes," *Physical Review Applied*, vol. 8, no. 6, p. 064017, 2017.

[17] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: VGG and residual architectures," *Frontiers in neuroscience*, vol. 13, p. 95, 2019.

[18] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[19] F. Zenke and S. Ganguli, "Superspike: Supervised learning in multilayer spiking neural networks," *Neural computation*, vol. 30, no. 6, pp. 1514–1541, 2018.

[20] G. E. F. Bellec, D. Salaj, A. Subramoney, R. Legenstein, and W. Maass, "Long short-term memory and learning-to-learn in networks of spiking neurons," in *Advances in Neural Information Processing Systems: NeurIPS*, 2018.

[21] S. B. Shrestha and G. Orchard, "SLAYER: spike layer error reassignment in time," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 1419–1428.

[22] W. He, Y. Wu, L. Deng, G. Li, H. Wang, Y. Tian, W. Ding, W. Wang, and Y. Xie, "Comparing SNNs and RNNs on neuromorphic vision datasets: similarities and differences," *Neural Networks*, vol. 132, pp. 108–120, 2020.

[23] C. Lee, A. K. Kosta, A. Z. Zhu, K. Chaney, K. Daniilidis, and K. Roy, "Spike-FlowNet: event-based optical flow estimation with energy-efficient hybrid neural networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 366–382.

[24] G.-q. Bi and M.-m. Poo, "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type," *Journal of neuroscience*, vol. 18, no. 24, pp. 10 464–10 472, 1998.

[25] J. P. Dominguez-Morales, Q. Liu, R. James, D. Gutierrez-Galan, A. Jimenez-Fernandez, S. Davidson, and S. Furber, "Deep spiking neural network model for time-variant signals classification: a real-time speech recognition approach," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[26] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[27] F. Zenke and T. P. Vogels, "The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks," *Neural computation*, vol. 33, no. 4, pp. 899–925, 2021.

[28] N. Rathi and K. Roy, "DIET-SNN: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[29] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian, "Incorporating learnable membrane time constant to enhance learning of spiking neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2661–2671.

[30] B. Yin, F. Corradi, and S. M. Bohté, "Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks," *Nature Machine Intelligence*, vol. 3, no. 10, pp. 905–913, 2021.

[31] B. U. Pedroni, S. Sheik, H. Mostafa, S. Paul, C. Augustine, and G. Cauwenberghs, "Small-footprint spiking neural networks for power-efficient keyword spotting," in *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2018, pp. 1–4.

[32] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, p. 27403, 1993.

[33] W. Ponghiran and K. Roy, "Spiking neural networks with improved inherent recurrence dynamics for sequential learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022, pp. 8001–8008.

[34] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[35] W. Wang, S. Hao, Y. Wei, S. Xiao, J. Feng, and N. Sebe, "Temporal spiking recurrent neural network for action recognition," *IEEE Access*, vol. 7, pp. 117 165–117 175, 2019.

[36] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3d perception," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2032–2039, 2018.

[37] J. Hagenaars, F. Paredes-Vallés, and G. De Croon, "Self-supervised learning of event-based optical flow with spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7167–7179, 2021.

[38] C. Lee, A. K. Kosta, and K. Roy, "Fusion-FlowNet: Energy-efficient optical flow estimation using sensor fusion and deep fused spiking-analog network architectures," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6504–6510.

[39] G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor, and R. Benosman, "HFirst: A temporal approach to object recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 10, pp. 2028–2040, 2015.

[40] A. Agrawal, M. Ali, M. Koo, N. Rathi, A. Jaiswal, and K. Roy, "Impulse: A 65-nm digital compute-in-memory macro with fused weights and membrane potential for spike-based sequential learning tasks," *IEEE Solid-State Circuits Letters*, vol. 4, pp. 137–140, 2021.

[41] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[42] P. Blouw, X. Choo, E. Hunsberger, and C. Eliasmith, "Benchmarking keyword spotting efficiency on neuromorphic hardware," in *Proceedings of the 7th annual neuro-inspired computational elements workshop*, 2019, pp. 1–8.