2024 IEEE 67th International Midwest Symposium on Circuits and Systems (MWSCAS) | 979-8-3503-8717-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/MWSCAS60917.2024.10658798

Visual Analysis of Leaky Integrate-and-Fire Spiking Neuron Models and Circuits

Sara Sedighi
Department of Electrical and Computer Engineering
Boise State University
Boise, ID 83725, United States
sarasedighi@u.boisestate.edu

Farhana Afrin
Department of Electrical and Computer Engineering
Boise State University
Boise, ID 83725, United States
farhanaafrin@u.boisestate.edu

Elonna Onyejegbu
Department of Electrical and Computer Engineering
Boise State University
Boise, ID 83725, United States
elonnaonyejegbu@u.boisestate.edu

Kurtis D. Cantley
Department of Electrical and Computer Engineering
Boise State University
Boise, ID 83725, United States
kurtiscantley@boisestate.edu

***Abstract*—Emulating biologically plausible online learning in spiking neural networks (SNNs) will enable the next generation of energy-efficient neuromorphic architectures. While software leads the way in terms of exploring various Machine Learning (ML) algorithms and applications, bridging the gap between hardware (devices and circuits) and software is crucial to accurately predict network properties, especially at large scale. This work compares behavior of a spiking neuron circuit simulated with Cadence Spectre to a Python model implemented with a custom spiking neuron model. The results demonstrate that the two exhibit the same spiking characteristics over a range of parameter values, confirming that the more versatile Python model indeed has a hardware equivalent.**

Keywords— Spiking neural network, Threshold dynamics, decay rate, LIF neuron

I. Introduction

Spiking Neural Networks (SNN) hold promise for energy-efficient computing due to their unique properties. Sparse firing, where neurons only transmit information when activated, inherently reduces energy consumption, while simultaneously increasing density of information flow. Additionally, the spiking nature of SNNs facilitates simpler hardware designs compared to traditional networks. Finally, SNNs exhibit robustness to noise, making them well-suited for resource-constrained devices and edge computing applications where power efficiency is paramount [1,2]. Evidence of SNN's capabilities in complex tasks like image classification, natural language processing, and reinforcement learning SNNs show promise for bridging the gap between neuroscience and artificial intelligence, making them a highly relevant topic for continued exploration [3-5].

In SNNs, two-terminal nonvolatile memory elements called memristors can be used for energy-efficient emulation of biological synaptic plasticity [6]. Electronic spiking neural networks comprised of memristors with Spike-Timing-Dependent Plasticity (STDP) learning rules have been shown to excel at processing temporal information, making them suitable for tasks involving time-series data, sequence learning, and temporal pattern recognition [7]. Electronic memristive neural networks with simple dynamic circuit elements like R(t) elements can be trained to mimic neuro-biological classical and non-classical STDP learning behaviors [8,9]. These CMOS-based networks can learn spatio-temporal patterns without supervision [10]. Among various neuron models, Leaky integrate-and-fire (LIF) neurons have emerged as a compelling building block for SNNs due to their simplicity, effectiveness, and biological plausibility [3,4]. However, to develop fully functioning neuromorphic hardware, it is required to incorporate the deep learning model into the hardware models.

In this paper, we employ the basic deep-learning model of SNN, incorporating the LIF neuron model and STDP learning rule to explore the impact of various SNN parameters on the training of the neural networks in neuromorphic hardware. Furthermore, a comparison with the hardware model strengthens the validity of the ML model. The analytical results reveal that the ML model is compatible with the hardware LIF model and thus can be employed for further R(t)-based SNN.

II. Python-Based Simulation of LIF Neurons

The leaky integrate-and-fire (LIF) model, an old concept dating back to 1907, offers a simple yet effective framework for understanding how most neurons process information. A leaky capacitor represents the neuron's cell membrane, accumulating voltage (membrane potential) from incoming ionic currents. The main equation governing this dynamic can be expressed as [11]

$$\tau_m \times \frac{dV}{dt} = -(V - E_L) + I \qquad (1)$$

Where τ_m is the membrane time constant, V is the voltage, E_L is the resting potential, and I is the input current. As current charges the membrane, the voltage rises. If it reaches a critical threshold V_{th}, the neuron fires, voltage is reset to a lower value V_{reset} and held there for a short time τ_{ref} mimicking the real neuron's refractory period [12-14].

In this work, to simplify the analysis of the model's behavior and focus on the core concepts, we utilize a step input current with a constant amplitude denoted by w. This injects a constant current into the model, neglecting the more complex and realistic time-varying currents present in biological neurons. Additionally, we introduce the parameter β as a representative of the decay rate of the voltage over time. β is defined as the reciprocal of the membrane time constant τ_m ($\beta = 1 - \tau_m^{-1}$) [11]. τ_m is the time constant of the membrane potential. This means that a larger tau corresponds to a smaller beta and vice versa .This simplifies the notation and allows for easier exploration of the model's dynamics under different decay rates.

Though fairly simple, the leaky integrate-and-fire (LIF) model captures the essence of how neurons process information. The leaky capacitor in the model fills with charge over time, i.e. current, and reaches a critical point called threshold voltage. This overflow of charges triggers a "spike" (action potential) and a brief rest period (refractory period), mimicking the behavior of a common type of neuron in the brain. The LIF model elegantly captures key features: 1) the summation of incoming currents over time and space, influencing when the neuron "fires"; 2) the refractory period, ensuring realistic firing rates; and 3) the natural leakage of voltage throughout, reflecting passive ion fluxes. While simplified, the LIF model's computational efficiency and ability to qualitatively predict diverse neuronal behaviors make it a powerful tool for exploring the world of neural information processing.

III. Results & Discussions

We present a comparative analysis of spiking neuron firing frequencies obtained from Python simulations and device-level simulation of the hardware model. This analysis aims to validate the accuracy of our simulations by focusing on key factors influencing firing frequency (threshold, input amplitude, and decay rate). Discrepancies will be investigated to identify potential areas for model refinement, ultimately bridging the gap between simulation and real-world spiking neuron behavior.

A computational investigation of LIF neuron dynamics was conducted using a custom Python model. This model encompassed biophysically relevant parameters including spike threshold, reset potential, membrane time constant, refractory period, and various input signal characteristics. The simulation framework allowed for systematic manipulation of these parameters, enabling the exploration of their influence on neuronal firing frequency. The generated data was subsequently employed to construct visualizations that elucidate the relationships between these parameters and the resulting firing patterns.

For the hardware implementation, we follow the CMOS LIF circuit based on the neuron design by Carver Mead [15], which is extensively used with different types of memory devices for pattern recognition and has been demonstrated in hardware [10,16-18]. The test circuit of our hardware model is simulated with TSMC 180nm technology as presented in Fig. 1.

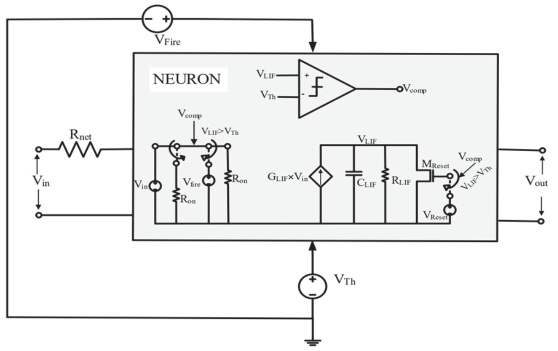

Fig. 1: The test circuit of the hardware model, with the gray block representing the decoupled neuron model: R_{net}=10k, V_{Fire}=1V. The comparator circuit is designed in CMOS based on the general-purpose comparator designed by Jacob, Baker [19].

In this paper, for computational efficiency, the CMOS-based design can be simplified into a decoupled model in which the leaky integration and firing take place independently as shown in Fig. 1. The input current is governed by the scaling parameter G_{LIF} which is comparable to the parameter w in the Python model, the leakage parameter C_{LIF} and R_{LIF} are analogous to decay rate, β. The threshold voltage can be tuned in this decoupled neuron model same as the python model. For the comparative analysis in this paper, we applied time-dependent voltage as input to the test circuit analogous to the input for the python-model and the output spike, V_{out} is 1V pulse of 1ms pulse width.

Fig. 2 delves into the relationship between threshold and input stimulus and its subsequent impact on spiking frequency. Lowering the firing threshold for any given stimulus magnitude results in significantly higher spiking frequency. Lowering the threshold unleashes a surge in frequency, regardless of input strength. This upward climb continues, with even the same threshold whispering higher frequencies for stronger input amplitudes.

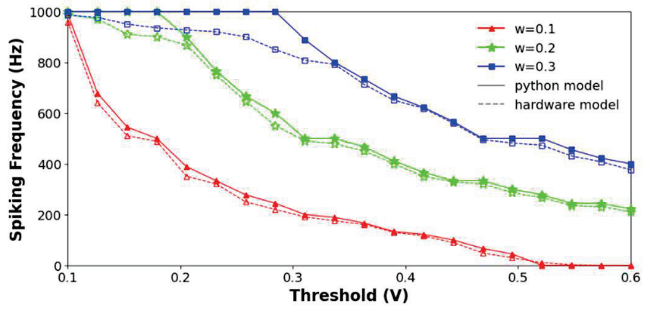

Fig. 2: Effect of threshold tuning controls on spiking frequency with different input amplitudes and fixed decay rate: Python vs. Hardware.

In Fig. 3, we observe a direct relationship between increasing input amplitude and spike frequency, evident across three distinct threshold values. Notably, for any given input amplitude, a lower threshold results in a higher spike frequency, highlighting the threshold's role as a sensitivity control for neuronal firing.

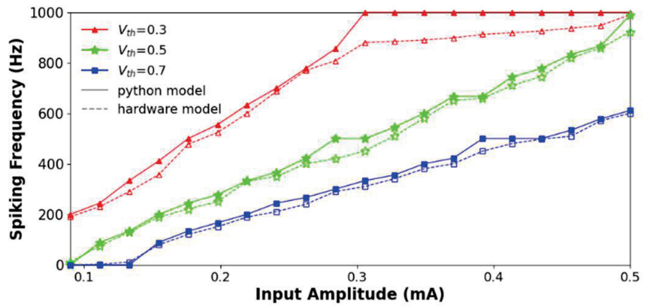

Fig. 3: Effect of input amplitude controls on spiking frequency with different threshold and fixed decay rate: Python vs. Hardware.

Similar to the effect of input amplitude, Fig. 4 demonstrates a positive correlation between increasing decay rate and output spike frequency. This holds true across all three threshold values presented. For a fixed input amplitude, a lower threshold once again reveals a higher spike frequency, reiterating its function as a critical modulator of neuronal responsiveness. This suggests that both higher decay rates and lower thresholds promote faster firing patterns in the model.

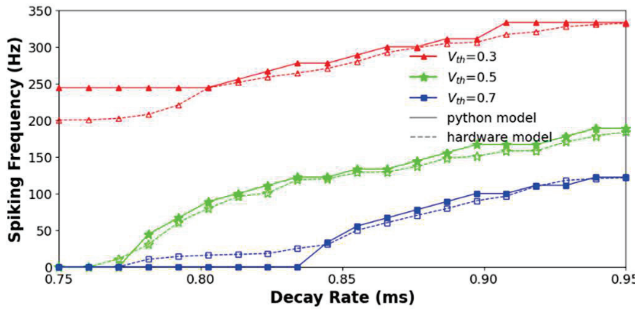

Fig. 4: Effect of decay rate controls on spiking frequency with different threshold and fixed input amplitude: Python vs. Hardware.

According to the plots in Fig. 2-4, Python simulations and hardware measurements of spike frequency across varying thresholds, input amplitudes, and decay rates exhibit strong agreement in their overall trends, showcasing an inverse relationship between threshold and frequency, a positive correlation between amplitude and frequency, and an increase in frequency with increasing decay rate. However, discrepancies exist in exact values, potentially stemming from real-world hardware imperfections, modeling simplifications, or parameter mismatches. While confirming the overall validity of the simulations, these deviations highlight the need for further refinement of models and consideration of hardware limitations, and deeper investigation into the specific sources of variation for improved accuracy and hardware translation based on these simulations.

Fig. 5 portrays the dynamic interplay between three distinct input currents and a neuron's spiking behavior within the LIF model. Each input current, visualized as a distinct trace over time, exerts its influence on the neuron's internal state, represented by its membrane potential. The threshold, depicted as a horizontal line, acts as a critical boundary for generating spikes. It simulates the behavior of an LIF neuron model with multiple input currents. It visualizes the input currents, membrane potential dynamics, and spike occurrences over time, offering insights into the neuron's response to varying input stimuli.

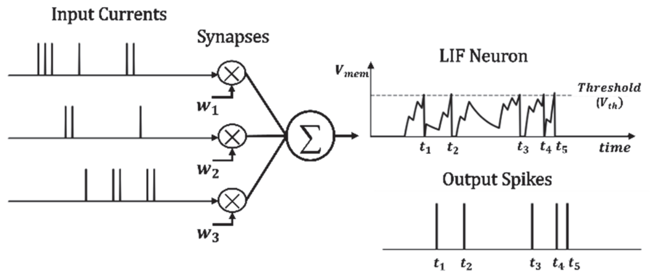

Fig. 5: Leaky Integrate and Fire (LIF) neuron dynamics: Pre-spikes, weighted by synaptic strength, integrate as current influx. Membrane potential decays exponentially. Upon crossing the firing threshold, the neuron fires a post-spike and resets [20].

Fig. 5 shows how LIF neurons work. Incoming spikes, modulated by synaptic weights, create a current that affects the post-neuron. The post-neuron's membrane potential integrates this current, leaking over time. When it surpasses a threshold, the neuron fires a spike and resets. No bias term is used.

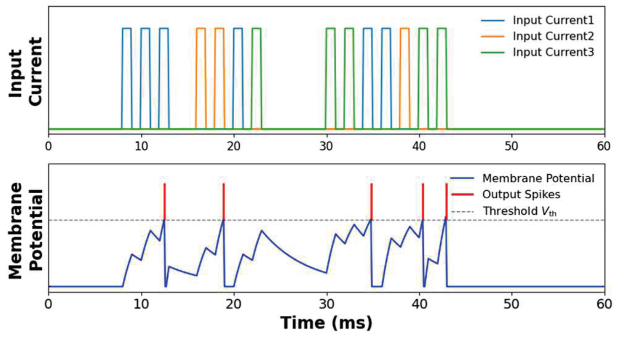

Fig. 6: Membrane potential response of a LIF neuron to three different input currents: top: input currents, bottom: membrane potential and output spikes.

Fig. 6 simulates a leaky integrate-and-fire (LIF) neuron model, which mimics the behavior of biological neurons. It incorporates the charging and discharging dynamics of the neuron's membrane potential in response to input currents. When the input current is nonzero, the membrane potential charges, and when it reaches a threshold, the neuron fires a spike. Upon input current cessation, the membrane potential gradually discharges. Importantly, when the membrane potential crosses the threshold, indicating a spike occurrence, it resets to a resting potential, similar to the refractory period observed in biological neurons. This behavior provides a realistic representation of neuron dynamics and offers insights into its response patterns under varying input conditions.

Building upon the fundamental LIF model introduced in the previous section (Fig. 5), this subsection explores how strategic adjustments to key parameters can influence the model's ability to capture non-zero input currents with improved fidelity. Specifically, we investigate the impact of modifying the threshold potential, input amplitude, and leak time constant (represented by the decay rate, τ) on the resulting spike patterns in Fig. 7.

By carefully manipulating these parameters, we can modulate the neuron's sensitivity to input currents and its propensity to generate spikes and simultaneously ensure the membrane potentials between both versions are also congruent.. For instance, lowering the threshold potential makes the neuron more excitable, requiring less input current to trigger a spike. Conversely, increasing the threshold potential renders the neuron less excitable, demanding a stronger input current to elicit a spike. Similarly, adjusting the input amplitude directly impacts the level of current injected into the neuron, influencing the rate and timing of spike generation. Finally, the leak time constant, τ, determines the rate at which the membrane potential decays back to its resting state after a spike. A shorter τ leads to faster decay and potentially sharper spikes, while a longer τ results in slower decay and potentially broader spikes.

Through controlled manipulation of these parameters, we can achieve a refined level of control over the neuron's spiking behavior. This enables us to tailor the LIF model to more accurately capture the dynamics of biological neurons under diverse input conditions, ultimately leading to a more comprehensive understanding of neuronal information processing.

A key challenge in training SNNs lies in the non-differentiable nature of the spiking events themselves. To overcome this hurdle, some SNN training algorithms leverage the concept of membrane potential. Membrane potential represents the voltage buildup within a neuron, acting as a precursor to the generation of a spike. By treating membrane potentials as differentiable signals, even with inherent noise introduced by the discontinuous spikes, these algorithms enable the application of error backpropagation. This approach closely resembles the training methods used in conventional deep networks [21].

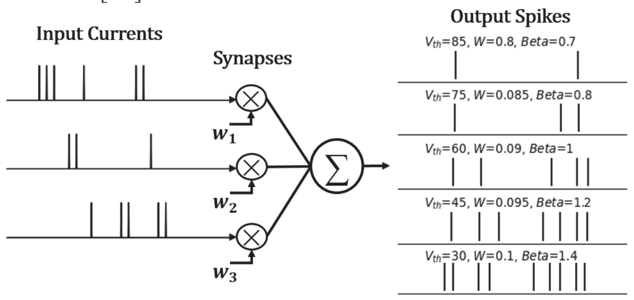
Input Currents
Synapses
Output Spikes
V_{th}=85, W=0.8, $Beta$=0.7
V_{th}=75, W=0.085, $Beta$=0.8
V_{th}=60, W=0.09, $Beta$=1
V_{th}=45, W=0.095, $Beta$=1.2
V_{th}=30, W=0.1, $Beta$=1.4
w_1
w_2
w_3
$\sum$

Fig. 7: Parameter Modulation for Input Signal Replication in LIF Neuron: Left panel: Input Currents 1, 2, and 3, Middle Panel: Summation of Input Currents 1, 2, and 3, Bottom Panel: Spike Generation with Parameter Variations.

IV. Conclusions

The impact of altering the threshold, firing rate, and input amplitude on the behavior of the LIF neuron model has been analyzed and compared with a hardware simulation. This analysis can help to tune the network parameters and provide a close match between hardware simulation and the python model for utilization in larger R(t)-based memristive SNNs performing pattern detection and classification. Properly tuned neuron parameters provide the opportunity to rapidly study various learning algorithms at a larger scale via simulation and subsequently deploy those algorithms in hardware. While the simulation models may not provide an exact match due to process variations in a physical circuit, it is expected that synaptic weights will adjust accordingly such that accuracy is maintained at the network level.

Acknowledgments

Funding for this work is provided by the National Science Foundation award 1751230 and the US Department of Energy award DE-SC0023391.

References

[1] M. Pfeiffer and T. Pfeil, "Deep Learning with Spiking Neurons: Opportunities and Challenges," Frontiers in Neuroscience, vol. 12, pp. 68–73, October 2018.

[2] C.D. Schuman, S.R. Kulkarni, M. Parsa et al., "Opportunities for neuromorphic computing algorithms and applications," Nat Comput Sci, vol. 2, pp. 10-19, 2022.

[3] W. Gerstner, (2014), "Neuronal dynamics: From single neurons to networks and brain function," Cambridge University Press.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[4] E. Izhikevich, (2006), "Dynamical systems in neuroscience," MIT Press.

[5] A. Pfeiffer and T. Standage, (2018), "How spiking neural networks can benefit from hardware acceleration. Nature Machine Intelligence," 1(1), 26-34.

[6] S.G. Dahl, R.C. Ivans and K.D. Cantley, "Effects of memristive synapse radiation interactions on learning in spiking neural networks," SN Appl. Sci., vol. 3, pp. 555, 2021.

[7] PJ Sjöström, GG Turrigiano and SB. Nelson, "Rate timing and cooperativity jointly determine cortical synaptic plasticity," Neuron, vol. 32, pp. 1149-1164, Dec 2001.

[8] F. Afrin and K. D. Cantley, "R(t)-Based Spike-Timing-Dependent Plasticity in Memristive Neural Networks," 2023 IEEE Workshop on Microelectronics and Electron Devices (WMED), Boise, ID, USA, 2023, pp. 1-4.

[9] F. Afrin and K. D. Cantley, "Investigating R(t) Functions for Spike-Timing-Dependent Plasticity in Memristive Neural Networks," 2023 IEEE 66th International Midwest Symposium on Circuits and Systems (MWSCAS), Tempe, AZ, USA, 2023, pp. 659-663.

[10] R. C. Ivans, S. G. Dahl, and K. D. Cantley, "A Model for R(t) Elements and R(t) -Based Spike-Timing-Dependent Plasticity with Basic Circuit Examples," IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 10, pp. 4206-4216, Oct. 2020.

[11] J. K. Eshraghian et al., "Training Spiking Neural Networks Using Lessons From Deep Learning," in Proceedings of the IEEE, vol. 111, no. 9, pp. 1016-1054, Sept. 2023.

[12] L. Lapicque, "Quantitative investigations of electrical nerve excitation treated as polarization," 1907. Biol Cybern. 2007 Dec;97(5-6):341-9. doi: 10.1007/s00422-007-0189-6. PMID: 18046573.

[13] N. Brunel, van MC. Rossum, "Lapicque's 1907 paper: from frogs to integrate-and-fire," Biol Cybern. 2007 Dec;97(5-6):337-9.

[14] P. Dayan and L.F. Abbott, (2001), "Theoretical neuroscience: Computational and mathematical modeling of neural systems," The MIT Press.

[15] C. Mead, "Analog VLSI and neural systems," 1st ed. Addison-Wesley Longman Publishing Co., Inc., 1989.

[16] K. D. Cantley, A. Subramaniam, H. J. Stiegler, R. A. Chapman, and E. M. Vogel, "Spike timing-dependent synaptic plasticity using memristorsand nano-crystalline silicon tft memories," in *Proc. 11th IEEE Int. Conf. Nanotechnol.*, Portland, OR, USA, Aug. 2011, pp. 421–425.

[17] Del Valle J, Salev P, Kalcheim Y, Schuller IK. A caloritronics-based Mott neuristor. Sci. Rep. 2020 Mar 9;10(1):4292.

[18] A. Natarajan and J. Hasler, "Implementation of Synapses with Hodgkin Huxley Neurons on the FPAA, " 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 2019, pp. 1-5.

[19] Baker, R. J. CMOS: Circuit Design, Layout, and Simulation: Third Edition, 2010.

[20] C. Lee, SS. Sarwar, P. Panda, G. Srinivasan, K. Roy, "Enabling Spike-Based Backpropagation for Training Deep Neural Network Architectures", Front Neurosci. 2020 Feb 28;14:119.

[21] JH. Lee, T. Delbruck and M. Pfeiffer "Training Deep Spiking Neural Networks Using Backpropagation", Front Neurosci. 2016 Nov 8;10:508.