UC Santa Cruz

UC Santa Cruz Electronic Theses and Dissertations

Title

Memristive Spiking Neural Network for Neuromorphic Computing

Permalink

https://escholarship.org/uc/item/0n46h0g9

Author

Zhou, Peng

Publication Date

2022

Copyright Information

This work is made available under the terms of a Creative Commons Attribution-NonCommercial-ShareAlike License, availalbe at https://creativecommons.org/licenses/by-nc-sa/4.0/

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

MEMRISTIVE SPIKING NEURAL NETWORK FOR NEUROMORPHIC COMPUTING

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL AND COMPUTER ENGINEERING

by

Peng Zhou

December 2022

The Dissertation of Peng Zhou
is approved:

Professor Sung-Mo "Steve" Kang, Chair

Professor Jason K. Eshraghian, Co-Chair

Professor Donald M. Wiberg

Peter Biehl
Vice Provost and Dean of Graduate Studies

Copyright © by

Peng Zhou

2022

Table of Contents

List of Figures

List of Tables

Abstract

Memristive Spiking Neural Network for Neuromorphic Computing

by

Peng Zhou

This dissertation is dedicated to using Memristive Spiking Neural Networks (MSNNs) for deep learning tasks such as image classification, visual associative memory tasks such as pattern recognition, and auditory cortex processing tasks such as sound localization (SL). The image classification model consists entirely of memristive neurons and memristive synapses utilizing deep learning plausible supervised learning rules. The pattern recognition fully MSNNs consists of memristive neurons and memristive synapses harnessing biologically plausible unsupervised learning rules. SL MSNNs emulate biological brain functionality with volatile memristive synapses.

By developing a minimal circuit element memristive neuron – Memristive Integrate-and-Fire (MIF) neuron – with commercially accessible memristors, we are able to demonstrate large-scale fully MSNNs and apply the supervised Backpropagation Through Time (BPTT) algorithm to train networks, achieving state-of-the-art accuracy for several datasets. In addition, using a memristive unsupervised learning rule based on a continuously evolving alpha function membrane potential, we are also able to train large-scale fully MSNNs with Spiking-Time-Dependent Plasticity (STDP) and achieve high accuracies. Moreover, a volatile memristor is used for mimicking Short-Term Depression (STD) synapses such that we can simulate SL

networks and pinpoint the direction of the sound coming from.

The major contributions reported in this proposal include:

- Development of the MIF neuron SPICE-level model;
- Validation of the SPICE-level MIF neuron model to a Python-based simulation of large-scale MSNNs;
- Simulation of a small-scale neuron network consisting of a presynaptic, a memristive synapse, and a postsynaptic MIF neuron to generate the STDP learning window;
- Abstraction of memristive STDP with alpha functions;
- Simulation of a large-scale, fully MSNN consisting of MIF neurons and memristive synapses for unsupervised learning in Python;
- Development of the MIF neuron numerical integration model;
- Validation of the numerical integration model in a behavioral simulation of large-scale MSNNs;
- Verification of the forward Euler numerical integration method to simplify the training process when compared to more complicated numerical methods;
- Training fully MSNNs by directly applying the gradient descent learning algorithm to the MIF neuron numerical integration model;
- Modeling a volatile memristor that exhibits Short-Term Plasticity (STP) based on real synapses;

- Mimicking a real brain auditory cortex Spiking Neural Network (SNN) with the proposed memristor model.

Acknowledgments

First and foremost, I would like to express my deepest appreciation to my advisor, Prof. Sung-Mo Kang, for being caring, loving, kind, and supportive throughout this journey. I am very fortunate to receive his guidance for five years of research. He usually gave me a relatively general direction and let me explore topics I felt passionate about. If our exploration went well, he would encourage me to dive deeper; if we encountered problems, he would be very patient, give me generous feedback, and guide me through challenges. In addition, his achievement in research, industry, and university admin work inspired me how to make decisions in my career paths and how to contribute to this society the most. He is also very hospitable in taking us to his favorite Chinese restaurant whenever our lab has visitors. He and Mia are always altruistic and care about my life here.

I could not have undertaken this journey without my co-advisor, Prof. Jason Eshraghian. Prof. Kang introduced me to Jason when he was a postdoc at the University of Michigan, and I was immediately impressed by his knowledge of devices, circuits, and algorithms. He mainly worked on memristive circuit design during his Ph.D. pursuit, and he was adept at learning spiking neural network algorithms and developing snnTorch during the pandemic. I had the unique privilege to receive both his visionary and detailed guidance for each idea about research, each step of our projects, and each workshop and paper he recommended. His enthusiasm about neuromorphic computing, his brilliance for conducting cool and rigorous research, and his Australian humor during our meetings makes me believe research is a fun adventure about unraveling mysteries.

I gave my sincere thanks to Prof. Donald M. Wiberg, who served on both my advancement and defense committees. He is the first professor I know in the ECE department, and he introduced me to Prof. Kang when I mentioned my interest in the brain and neuroscience. He was supportive each time I approached him and gave me full encouragement about my research.

I also thank Prof. Shiva Abbaszadeh. She mentored me to work on applications of neuromorphic computing and deep learning. Her hardworking toward research is outstanding. She is always strong, persistent, and open-minded about new research ideas.

I would also like to thank Prof. Bai-Sun Kong, who visited Prof. Kang's lab and mentored me to work on memristive spiking neural networks. I was a first-year graduate student, and Prof. Kong mentored me with extreme patience. During his visit, he came to the lab every day to read papers, discuss ideas, and have meetings with us. His diligence and dedication motivated me to continue working in this area.

It has been a pleasure working with my lab mates, Donguk Choi and Jieun Kim. Donguk had worked in the semiconductor industry for many years, and he is excellent in several areas, such as semiconductor processes and mathematics. Jieun worked closely with me on spiking neural network simulations. She is intelligent and reliable. We had worked the whole night many times to help each other with simulations.

Many thanks to friends both in USA and China. I made many good friends in the USA while keeping connections with friends who have 15 hours timezone difference from here. They gave strong support when I met tough challenges in

both research and life, especially after the outbreak of the COVID-19 pandemic. Special thanks to friends who are like family and always give me love, trust, and help whenever I need it.

Finally, words cannot express my gratitude to my family, mom, and dad, for their love and support in pursuing my Ph.D. overseas while missing me every day. And my cousins, aunts, uncles, and grandparents, who care for me in a big family and feel proud of me.

Chapter 1

Introduction

1.1 Neuromorphic Computing

Neuromorphic computing is guided by neural dynamics, and takes advantage of the brain's parallel computing, sparse encoding, and power efficiency. It encompasses the broad span of brain-inspired materials, devices, analog, digital, mixed-mode analog/digital VLSI, computer architectures and systems, software, and algorithms to mimic neural networks in the nervous system. The origin of neuromorphic computing can be traced back to the 1980s when Carver Mead developed bio-inspired microelectronics [103, 104]. He noticed the similarity between the dynamics of biological neurons and the transistors in the sub-threshold regime, which can use analog circuits to emulate biological neural networks, and notably led to the design of the first silicon retina [100]. Taking inspiration from biological neural networks ushered in a new computing paradigm where computing and memory are merged together.

The brain is widely considered to be the most complex 1.4 kg mass in the

universe, consisting of 10^{11} neurons and 10^{15} synapses with extremely complex connections and layers. The neuron is regarded as the basic unit of computing in the brain, where large numbers of neurons are interconnected via synapses. Information is transferred between neurons as action potentials, or spikes, which communicate voltage pulses from a presynaptic neuron to postsynaptic neurons. The strength of the spike is weighted by the strength of the synapse. While the brain and conventional computers both serve as processing systems in a very general way, it is clear that their core architecture and operation are different, which encourages a large group of multidisciplinary researchers to explore the idea of brain-inspired computing, also known as neuromorphic computing. While neuroscientists still do not fully understand all the details of the brain's structure, operation, and dynamics, a variety of abstract neuron and synapse models have been proposed and validated to support the pursuit of neuromorphic computing.

Neuromorphic computing also attracts much attention as Moore's law is pushed to the limit. Creating new computer architectures to overcome the von Neumann bottleneck is a promising approach to improving computation capabilities. Inspiration is derived from the human brain since neurons and synapses both exhibit storage and computation functionality, whereas conventional computing paradigms separate memory and processing into two disparate parts. This von Neumann architecture also brings another bottleneck known as the memory wall. The rate of improvement of processing in central processing units (CPUs) has exceeded that of memory accesses, such that much time is spent in retrieving data from memory which limits the performance of the system. Additionally, the brain is extremely

energy efficient for the same number of calculations when compared to conventional computers. The main architecture of neural networks in the brain consists of neurons and synapses where neuron spiking and synaptic plasticity are among the most important dynamics in the network. Drawing the benefits from spikes and synapses and moving them into modern computer architectures provides an opportunity to improve hardware scale in terms of compactness and power consumption.

The promise of neuromorphic computing has been recognized across both industry and academia, and has grown in importance with the increasing importance of deep neural networks (DNNs) over the past decade. It has been investigated by several companies and research teams that try to make neuromorphic computing units through various approaches, including Loihi [44], SpiNNaker [63], NeuroGrid [28], TrueNorth [106], BrainScaleS [128], DYNAPs [109], and so on [33, 62]. It is a multidisciplinary approach, and many seemingly disparate fields have been involved including physics and material science, electrical engineering, software engineering, algorithm, biology, neuroscience, cognitive science, and so on.

1.2 Memristor

The memristor device, considered the fourth fundamental element in electrical engineering, was introduced 50 years ago. This history is relatively brief when compared to the centuries-old resistor, capacitor, and inductor. Memristor was first postulated by Chua in 1971 [37], and generalized by Chua and Kang in 1976 [39]. A few decades later, the research and development of memristor circuits and systems were catapulted following the fabrication of a nanoscale memristor array by HP lab

that was reported in Nature in 2008 [138]. Since then, a variety of materials and devices based on different physical and chemical characteristics have been proposed and fabricated as memristors such as electrochemical metallization, valence change, thermochemical memory cells, and phase change. Recently, they have become commercially available as resistive random-access memories (RRAM) [1, 3]. The applications of memristors are promising for bio-inspired neuromorphic computing due to their similarity to plastic synapses of low-power consumption, 3D integration with silicon, and high density. It is the most popular and ubiquitous device-level element in neuromorphic systems [130]. Although many neuromorphic systems are fabricated based on conventional complementary metal–oxide–semiconductor (CMOS) technology, CMOS technology faces several bottlenecks when applied to novel neuromorphic systems. As each neuron has up to 10^4 synaptic interconnects, it requires a large amount of circuit, memory, and energy for CMOS to emulate the various synaptic plasticity which plays a key role in learning in this massive system. It encourages physics, materials, and device researchers to explore new technology beyond CMOS, especially nanoscale devices resembling biological synapses.

1.3 Spiking Neural Network

Spiking Neural Networks (SNNs), the third generation of the neural network following the first generation and the second Artificial Neural Networks (ANNs), aim to bridge the gap between the biological brain and computational unit in the neural network architecture level [33]. The first generation ANNs is relatively simple, consist of only two fully connected layers, and provide a binary output. The

second generation of neural networks replaced the output with a continuous analog value and can be trained using the gradient backpropagation algorithm. This architecture has both feedforward and backpropagation paths and can be utilized in a multi-layered neural network, which also refers to Deep Neural Networks (DNNs). Among the most famous types of DNNs are Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Generally speaking, CNNs are deployed for computer vision tasks, and RNNs are used for sequential or temporal data, such as in audio processing which relies on the memory of temporal events. DNNs are state-of-the-art in various areas, and many Python packages have been developed including Keras [2], TensorFlow [12], and PyTorch [7]. However, compared with a biological brain, the performance of DNNs in the terms of required data, speed, and power consumption is unsatisfactory, which results in the interest of biological plausibility and drawing more inspiration from the brain. Therefore, SNNs emulate the neuronal dynamics observed by neuroscientists and carefully mimic the neuron and synapse models. SNNs leverage spiking activity, which is one of the most significant features captured in neuronal dynamics in the brain. Many SNN Python simulation packages have been developed including Brian2 [136], Nest [65], Nengo [26], PyNN [45], NEURON [74].

Currently, neuroscience and machine learning have vastly different approaches to learning. Additionally, with the support of Graphic Processing Units (GPUs) and Compute Unified Device Architecture (CUDA), DNNs have achieved impressive results for a large number of applications. Therefore, some efforts have been devoted to linking the brain and accelerated computing by integrating modern

DNN approaches with SNNs [102]. By transferring the biologically plausible neuron model to an RNN-like computational graph and surrogate gradient descent methods [111], SNNs can take advantage of feedforward and Backpropagation Through Time (BPTT) [161]. Some Python packages developed to harness autodifferentiation in SNNs include snnTorch [11], Rockpool [8], sinabs [9], norse [6], and so on. Additionally, another approach was taken by converting the ANN to SNN after training by rate coding [127], which is often achieved with the SNN conversion toolbox (SNN-TB) [10] or temporal coding [137].

Chapter 2

Background Description of Memristor

In 1971, Chua noticed that there is a missing fundamental circuit element based on arguments of symmetry [37]. The resistor formed a relationship between voltage and current. The inductor formed the link between magnetic flux and current. And the capacitor connected charge and voltage. There should be a relation to link between the charge and flux, and Chua hypothesized that the memristor could fill the void as shown in Figure 2.1 [4]. As its name reveals, a resistor having memory - memory resistor - memristor, is a device where flux can alter charge and vice versa.

2.1 Memristor

A memristor has the following criteria: [59]

- passivity
- two-terminal device

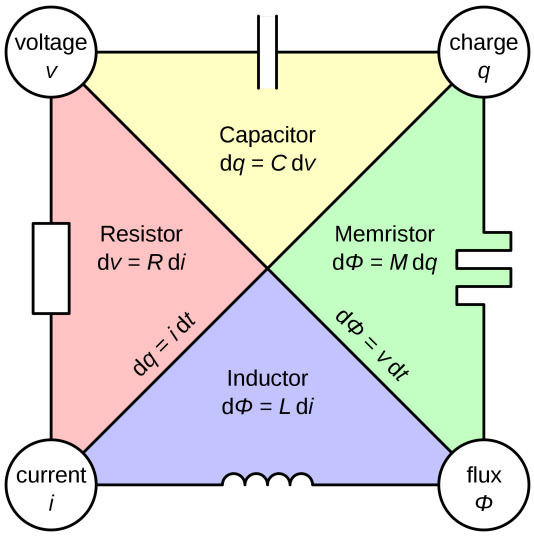


Figure 2.1: Resistor, capacitor, inductor, and memristor relations.

- pinched hysteresis loop in the V-I plane under bipolar periodic input
- crossing at the origin.

Memristor can be defined as charge-dependent:

$$M(q) = \frac{\mathrm{d}\phi}{\mathrm{d}q} \tag{2.1}$$

or flux-dependent:

$$M(\phi) = \frac{\mathrm{d}q}{\mathrm{d}\phi} \tag{2.2}$$

Charge is the time integral of an electric current, and magnetic flux is the time integral of an electric voltage. Substituting the flux as the time integral of the voltage, and charge as the time integral of current, the Equation 2.1 and 2.2 can be derived as:

$$M(q(t)) = \frac{\mathrm{d}\phi/\mathrm{d}t}{\mathrm{d}q/\mathrm{d}t} = \frac{V(t)}{I(t)} \tag{2.3}$$

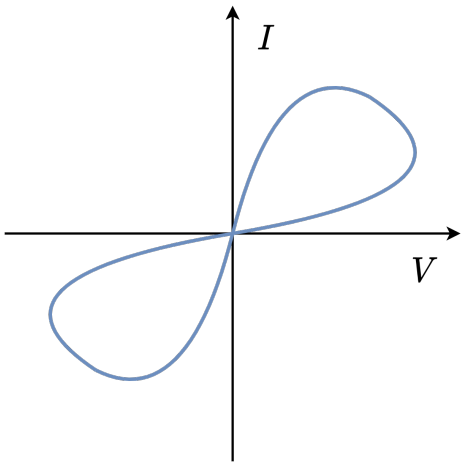


Figure 2.2: A typical memristor IV curve.

$$M(\phi(t)) = \frac{\mathrm{d}q/\mathrm{d}t}{\mathrm{d}\phi/\mathrm{d}t} = \frac{I(t)}{V(t)} \tag{2.4}$$

In Equation 2.3, memristor works as a resistor and in Equation 2.4, memristor works as a conductor. A typical V-I curve of memristor is shown in Figure 2.2.

2.2 Memristive Device and System

In 1976, the concept has been generalized to the 'memristive device and system' [39]. The generalized definition of memristor is:

$$\dot{x} = f(x, u, t) \tag{2.5a}$$

$$y = g(x, u, t)u \tag{2.5b}$$

where x is the n^{th} order state variable, and $\dot{x}$ is its time-derivative.

For a current-controlled memristive device:

$$\dot{x} = f(x, i, t) \tag{2.6a}$$

$$v = g(x, i, t) \cdot i \tag{2.6b}$$

For a voltage-controlled memristive device:

$$\dot{x} = f(x, v, t) \tag{2.7a}$$

$$i = g(x, v, t) \cdot v \tag{2.7b}$$

In 2008, Hewlett-Packard lab finally linked the memristive device to a titanium dioxide (TiO_2) solid-state circuit element, which attracted significant attention from multiple research fields. It is worth to note although they named the device memristor, it is actually a memristive device, which later makes the two concepts exchangeable. This discovery has encouraged device researchers to overcome the limitation of CMOS technology by using resistive memory; the circuit researcher to use it as a logic, analog computing element, or memory storage such as resistive random-access memory (RRAM); neuromorphic computing researchers to deploy it as a 'synapse' and 'neuron', and algorithm researchers to model its dynamics and utilize it in SNNs.

Although multiple models and devices have been proposed, a memristor should meet three requirements: (i) nonlinearity, (ii) continuous temporal dynamics, and (iii) a strictly monotonically increasing flux-charge relation [59].

Chapter 3

Memristive Synapse

The brain can compute a tremendous number of tasks, and the core units of the brain are generally viewed to be neurons and synapses. In the nervous system, a synapse is a structure that can pass an electrical or chemical signal from a presynaptic neuron to a postsynaptic neuron. For a chemical synapse, when there is a spike in the presynaptic neuron, this signal will stimulate the release of a chemical called neurotransmitter. After some time, the neurotransmitter will migrate to bind with the receptors in the membrane of the postsynaptic neuron. Different neurotransmitters have different effects such as excitatory or inhibitory effects on the postsynaptic neurons. Many receptors contain different ion channels, which can permit ions to flow between the inside and outside of the membrane, and it can cause the Excitatory Postsynaptic Current (EPSC) or Inhibitory Postsynaptic Current (IPSC).

Synapses have plasticity. In neuronal systems, the strength of synapses increases or decreases according to spiking in presynaptic or postsynaptic neurons,

referred to as synaptic plasticity. In fact, memory is represented by the interconnection of a large neural network. That is to say, the synapse weight is the key to learning and memory. Different types of synaptic plasticity include Long-Term Plasticity (LTP) [25] and Short-Term Plasticity (STP) [179]. Synapses may strengthen or weaken and exhibit memory retention over a relatively long time, called Long-Term Facilitation (LTF) or Long-Term depression (LTD), respectively. If the change is within a relatively short time, then it is referred to as Short-Term Facilitation (STF) or Short-Term Depression (STD).

3.1 Non-Volatile Memristor for Long-Term Plasticity

Synaptic LTP has already inspired the training process in DNN, which concatenates all synapse weights as a large multi-dimensional matrix and finds the optimal weight matrix through error backpropagation. However, the mechanism and learning rules in neuroscience are not the same. One of the most famous learning rules in neuroscience is called Hebbian rule [73]. The shortest summary of this rule is: neurons which 'fire together, wire together' [133]. Hebbian rule can be interpreted as a rate model defined by the neuron spiking rate. It is a local rule, and it requires neurons to be simultaneously active. The general model for this local rule can be defined as [64]:

$$\frac{\mathrm{d}w_{ij}}{\mathrm{d}t} = F(w_{ij}, M, v_j^{pre}, v_i^{post}) \tag{3.1}$$

where w_{ij} is the synaptic weight, M is the effect of neuromodulator, v_j^{pre} is the presynaptic neuron firing rate, and v_i^{post} is the postsynaptic neuron firing rate.

These variables are all locally available at the site of a synapse.

A Taylor expansion of Equation 3.1 with respect to the rates is:

$$\frac{\mathrm{d}w_{ij}}{\mathrm{d}t} = a_0(w_{ij}, M) + a_1(w_{ij}, M)^{pre} v_j^{pre} + a_1(w_{ij}, M)^{post} v_i^{post} + a_2(w_{ij}, M)^{corr} v_j^{pre} v_i^{post} + ... \tag{3.2}$$

where a_0 is the effect when no spike occurs at both pre- and postsynaptic neurons. a_1^{pre} is the expansion coefficient when spikes occur only at presynaptic neurons. a_1^{post} is the expansion coefficient when spikes occur only at postsynaptic neurons. a_2^{corr} is the expansion coefficient of the joint activity when spikes occur at both pre- and postsynaptic neurons. There are more terms of higher orders v_j^{pre} and v_i^{post} which are represented by '...'. The coefficients are all dependent on parameters w_{ij} and M. According to different parameters and conditions, Hebbian rule can be LTF or LTD. It is worth noting that, Hebbian rule is a set of learning rules rather than a specific fixed rule.

Another popular learning rule is called Spike-timing-dependent plasticity (STDP). With the STDP process, the synaptic weight will increase or decrease based on the time difference between the pre- and postsynaptic spikes. The total weight change from neuron j to neuron i w_{ij} is defined as:

$$\Delta w_{ij} = \sum_n \sum_f W(t_i^n - t_j^f) \tag{3.3}$$

where t_i^n denotes the spike times of postsynaptic neuron i and t_j^f indicates the spike time of presynaptic neuron j. n and f counts the pre- and postsynaptic spikes. $W(x)$ is the 'learning window' of the STDP function.

Many variations of STDP exist, and one of the most common types is

pair-based, and is defined as:

$$W_+(x) = A_+(w)e^{\dashv|\Delta t|/\tau_+} \quad at \quad t_{post} \quad for \quad t_{pre} < t_{post} \tag{3.4a}$$

$$W_-(x) = A_-(w)e^{\dashv|\Delta t|/\tau_-} \quad at \quad t_{pre} \quad for \quad t_{post} < t_{pre} \tag{3.4b}$$

where $|\Delta t| = \left|t_{post} - t_{pre}\right|$, t_{post} is the time of the postsynaptic spike and t_{pre} is the time of the presynaptic spike. Usually, $A_+(w)$ is positive, $A_-(w)$ is negative, and they may depend on the current synaptic weight. $W_+(x)$ is for LTF and $W_-(x)$ is for LTD.

Let us introduce $S_j = \sum_f \delta(t - t_j^f)$ and $S_i = \sum_n \delta(t - t_i^n)$ representing the spike trains of pre- and postsynaptic neurons. The pair-based STDP rule of Equation 3.4 can be implemented by:

$$\frac{\mathrm{d}x_j}{\mathrm{d}t} = -\frac{x_j}{\tau_+} + \sum_f \delta(t - t_j^f) \tag{3.5a}$$

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = -\frac{y_i}{\tau_-} + \sum_n \delta(t - t_i^n) \tag{3.5b}$$

where t_j^f indicates the spike time of the presynaptic neuron and t_i^n denotes the spike times of the postsynaptic neuron. x_j and y_i can be interpreted as a trace that each pre- and postsynaptic spike leaves, respectively. The trace x_j and y_i corresponds to the $e^{\dashv|\Delta t|/\tau_+}$ and $e^{\dashv|\Delta t|/\tau_-}$ in the Equation 3.4

Thus, we can update the Equation 3.4 as:

$$\frac{\mathrm{d}w_{ij}}{\mathrm{d}t} = A_+(w_{ij})x_i(t)\sum_n \delta(t - t_i^n) + A_-(w_{ij})y_i(t)\sum_f \delta(t - t_j^f) \tag{3.6}$$

where the first term on the right side denotes the pre-before-post effect and the second term indicates the post-before-pre effect.

Interestingly, for independent Poisson inputs, STDP models are related to rate models [64], and it can be defined as one of the rate models as:

$$\frac{\mathrm{d}w_{ij}}{\mathrm{d}t} = \int_0^{+\infty} W(-s)\epsilon(s)\mathrm{d}s \cdot v_j^{pre} + \int_{-\infty}^{+\infty} W(s)\mathrm{d}s \cdot v_j^{pre} v_i^{post} \tag{3.7}$$

where the $\int_0^{+\infty} W(-s)\epsilon(s)\mathrm{d}s$ in the first term of the right side is the integral over the 'causal' part of the learning window, also known as the 'pre-before-post' relation. $\epsilon(s)$ for $s > 0$ describes the time course of a Postsynaptic Potential (PSP).

Comparing Equation 3.7 with Equation 3.2, we find that we have two terms defining the coefficient $a_1(w_{ij}, M)^{pre}$ and $a_2(w_{ij}, M)^{corr}$ while other terms are all zero.

Non-volatile memristors can operate as LTP synapses as the memristance will not change after each update. In a fully connected neural network, the number of synapses between two layers is $m \cdot n$ where m is the number of neurons in the previous layer, and n is the number of neurons in the next layer as the Figure 3.1 presents. Therefore, non-volatile memristors have been used in crossbar arrays for vector-matrix multiplication utilizing Kirchhoff's current law as Figure 6.2 shows.

Gradient backpropagation and STDP are entirely different mechanisms for learning. Generally, the former is well-adopted in the DNN area while the latter is popular in the SNN area. Both of them can be implemented with memristors. For backpropagation, memristive crossbar arrays can be utilized for neural network training [71] or just inference alone [14]. The training requires an external control unit which adds more overhead than Figure 6.2 shows. The STDP rule described

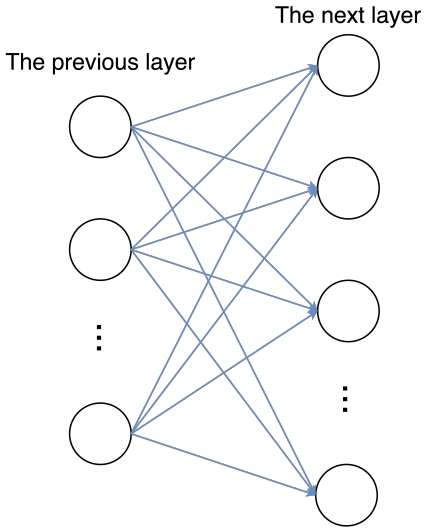


Figure 3.1: Two layers of neural network

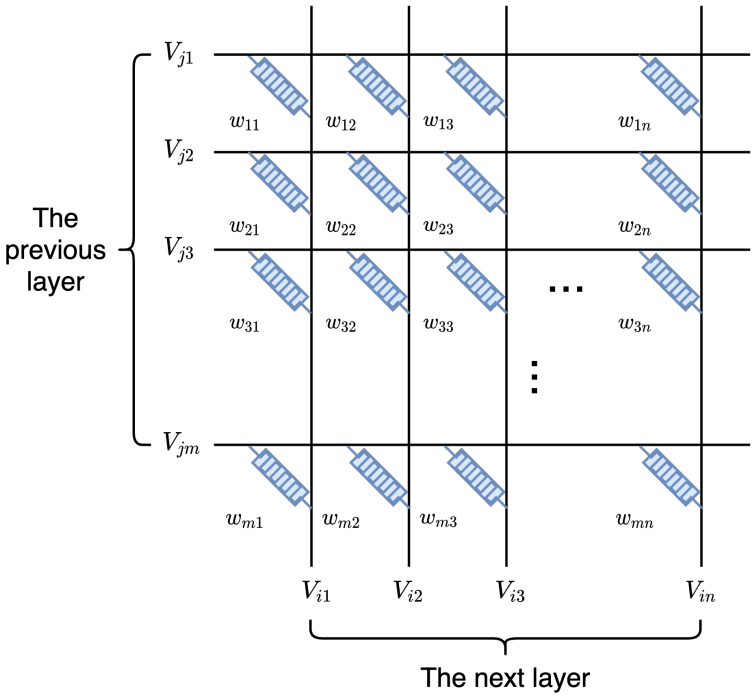


Figure 3.2: A memristive crossbar representing the weight matrix.

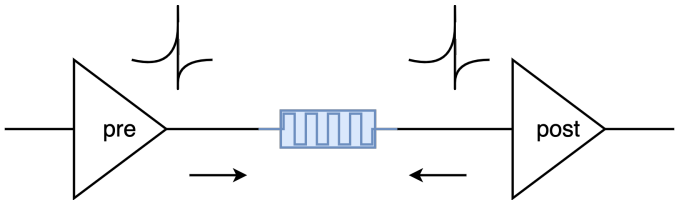


Figure 3.3: Memristor between pre- and postsynaptic neurons.

above can also be achieved by using memristors [101], and can be verified using SPICE models [167]. As Figure 3.3 shows, a memristor is interposed between two neurons, where the presynaptic and postsynaptic spikes will generate the voltage difference across the memristor and then cause the memristance update. Moreover, the time difference between presynaptic and postsynaptic spikes will cause different changes in memristance.

3.2 Volatile Memristor for Short-Term Plasticity

The majority of plastic synapse emulation relies on non-volatile memristors for LTP, while few researchers focus on volatile memristors to mimic STP synapses. The volatile memristors are also significant since the STP synapses play profound roles in neural information processing such as motion detection, speech recognition, and working memory [66]. Different from LTP, STP only relies on the spiking activity of the presynaptic neuron. Assume fraction P_{rel} denotes the neurotransmitter released by the presynaptic neuron. Then the synaptic weight is directly related to P_{rel}, and both STF and STD can be modeled with the dynamics of P_{rel}. STF can be defined as:

$$\frac{\mathrm{d}P_{rel}}{\mathrm{d}t} = \frac{P_{rel} - P_0}{\tau_F} + f_F(1 - P_{rel}) \sum_f \delta(t - t^f) \tag{3.8}$$

where t^f is the time for each presynaptic spike. P_0 is the resting value of P_{rel}. τ_F is a time constant governing the recovery process. f_F controls the degree of facilitation.

Analogously, STD can be defined as:

$$\frac{\mathrm{d}P_{rel}}{\mathrm{d}t} = \frac{P_{rel} - P_0}{\tau_D} + f_D(1 - P_{rel}) \sum_f \delta(t - t^f) \tag{3.9}$$

where t^f is the time for each presynaptic spike. P_0 is the resting value of P_{rel}. τ_D is a time constant governing the recovery process. f_D controls the degree of depression.

Volatile memristors can emulate the STF or STD as the memristance change only retains a short time in volatile memristors. Usually, STF and STD are measured by Paired-Pulse Facilitation (PPF) and Paired-Pulse Depression (PPD). The PPF and PPD index, as a parameter to evaluate the strength of PPF or PPD, is defined as $\frac{A_2}{A_1}$, where A1 and A2 are the absolute amplitudes of the EPSC or the IPSC by two successive presynaptic spikes. In 2018, a kind of volatile memristor was proposed and the PPF and PPD indexes were measured [139]. This is a two-terminal single-layered molybdenum disulfide (MoS_2) device whose conductance change is realized by Joule heating.

3.3 Application of Volatile Memristive Synapse in Sound Localization

3.3.1 Introduction

Among the SNN with STP synapses in the brain, the sound localization (SL) system with STD synapses that can extract Interaural Time Difference (ITD) is a well-established mechanism. We demonstrated an SNN that can detect the sound source direction and achieved human-level SL accuracy using the volatile memristor model [139].

SL [129] is the ability to identify the direction of a sound source requiring the most precise temporal processing in the brain, which is an essential survival feature for predators and prey. The brain locates the sound with several cues, including 'monaural spectral cues' and 'binaural cues'. Interaural Time Difference (ITD) works as 'binaural cues' in an auditory cortex identifying the sound source along the azimuth [68]. For an off-centered sound source, there is a timing difference between the sound waves arriving at both ears, which defines the amount of ITD. The Jeffress model [80] describes an ITD-based cortical mechanism for locating a sound source direction. Short-Term Depression (STD), which temporally weakens synaptic connectivity by decreasing the Excitatory Postsynaptic Current (EPSC), plays an important role in SL in the auditory cortex [15, 41, 78, 89]. Recently, 2D materials have been found to be efficient for mimicking the behavior of synapses with STD, and have shown feasibility for implementing a primitive form of SL [139].

We presented a biologically plausible SNN based on the 2D material achiev-

ing human-level SL accuracy. A two-terminal single-layered MoS_2 device whose conductance change is realized by Joule heating is used for efficiently implementing a plastic synapse with STD. The Jeffress model is adapted to mimic the auditory cortex performing SL. Alpha synaptic current and Leaky Integrate-and-Fire (LIF) neuron models are adopted to make our neuronal behavior as realistic as possible. Lateral inhibition, one of the basic functions of biological neurons, is used for clearer SL with improved energy efficiency. Combining all of the above, our biomimetic SNN achieves human-level SL accuracy of 1-degree resolution when multiple input spikes with random variation in spike timing are received.

3.3.2 Models

3.3.2.1 Memristor Model

Memristors can imitate the PPD [139]. PPD is measured as the ratio of the magnitudes of the two EPSC in two successive presynaptic spikes. Let A_1, A_2 be the amplitude of the first EPSC and the second EPSC. Then the ratio $\frac{A_2}{A_1}$ defines PPD, and the value of PPD varies with the separation time of these two successive presynaptic spikes. The value of PPD is always smaller than 1, that is, $\frac{A_2}{A_1} < 1$. In SL SNN, synapses have STD plasticity, and we use PPD to model STD. The value of PPD approaches 1 as the separation time increases. The PPD function is described by:

$$PPD(t) = 1 - \left(1 - U \cdot PPD(t_0^-)\right) \cdot e^{-\frac{t-t_0}{\tau}} \quad \text{for } t \geq t_0 \geq 0 \tag{3.10}$$

where $PPD(t)$ denotes the depression effect. The first spike arrives at time t_0,

and t is the separation time of the second spike. t_0^- is the time just before the spike. $U \cdot PPD(t_0^-)$ refers to the value of the remaining PPD value after reduction. $PPD(t)$ starts at 1. Thus, for the first spike, $PPD(t_0^-) = 1$. τ is the time constant determining how fast the PPD approaches 1. Note that in Brian2, neuronal and synaptic models need to be illustrated by mathematical equations in ODE format, on which our simulation is based. The PPD behavior of the memristor can be modeled by ODE format in Brian2 as:

$$\tau \cdot \frac{\mathrm{d}PPD(t)}{\mathrm{d}t} = 1 - PPD(t) \tag{3.11a}$$

$$PPD(t^+) \leftarrow U \cdot PPD(t^-) \quad \text{upon spiking at } t = t_0 \tag{3.11b}$$

$$\text{for } t \geq t_0 \geq 0$$

where t^- is the time just before t, and t^+ is the time just after t.

The PPD data [139] used for emulation is shown in Table 3.1. The PPD value increases as the separation time increases. When the separation time is large enough, PPD approaches 1. In Equation 3.10 and 3.11, we need to calculate U and τ using the PPD data in Table 3.1. After curve fitting, U is identified to be 0.6668 and τ 183.8 (ms). Thus, Equation 3.10 can be rewritten as:

$$PPD(t) = 1 - \left(1 - 0.6668 \cdot PPD(t_0^-)\right) \cdot e^{-\frac{t-t_0}{183.8}} \qquad \text{for } t \geq t_0 \geq 0 \tag{3.12}$$

Table 3.1: Measured PPD data from a memristor having STD.

Separation time between two spikes (ms)	**Paired-pulse depression (PPD)**
10	0.651
25	0.770
50	0.783
100	0.844
250	0.892
500	0.949
1000	0.953
2000	0.971

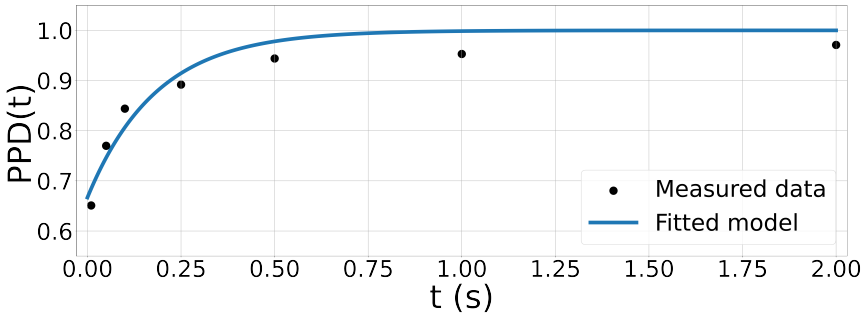


Figure 3.4: Short-Term Depression (STD) recovery curve based on PPD data. Black dots are the measured PPD data, and the blue curve denotes the curve fitting result. $t_0 = 0$.

And Equation 3.11 can be updated as:

$$183.8 \cdot \frac{\mathrm{d}PPD(t)}{\mathrm{d}t} = 1 - PPD(t) \tag{3.13a}$$

$$PPD(t^+) \leftarrow 0.6668 \cdot PPD(t^-) \quad \text{upon spiking at } t = t_0 \tag{3.13b}$$

for $t \geq t_0 \geq 0$

The weight recovering curve after the first spike based on PPD data is shown in Figure 3.4.

3.3.2.2 Neuron and Synapse Model

The LIF model [64, 88] is most widely used for analyzing neuronal dynamics. The equivalent circuit model for a generalized LIF neuron is shown in Figure 3.5. In this model, a resistor R in series with a DC source V_{rest}/V_{reset} is connected in parallel with a capacitor C. A postsynaptic neuron receives a synaptic current $I(t)$ generated by presynaptic spikes. A portion of current $I(t)$ flowing into C makes the membrane potential $V(t)$ increase. The charge leakage is through resistor R. When $V(t)$ increases to reach a threshold, the neuron generates a spike. After spiking,

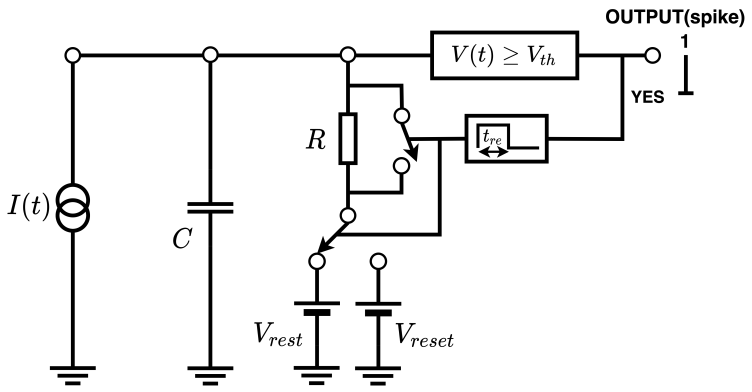


Figure 3.5: Schematic diagram of the LIF electrical model.

$V(t)$ is reset to V_{reset}. If $I(t)$ is absent, the voltage across C is settled eventually at V_{rest} representing the cell's resting potential. During the refractory period t_{re}, a neuron is incapable of spiking. The membrane potential dynamics is described by Equation 3.14:

$$\tau_m \frac{\mathrm{d}V(t)}{\mathrm{d}t} = -\big(V(t) - V_{rest}\big) + RI(t) \tag{3.14}$$

where $V(t)$ is the membrane potential $\tau_m = RC$ is the membrane time constant. V_{rest} is the resting potential and the initial value of $V(t)$ when $I(t) = 0$.

The model of the synaptic current used here is an alpha function [160]. It describes a presynaptic spiking effect, each of which has two phases short rise and long decay. The synaptic current following a synaptic spike is described by Equation 3.15 [124]:

$$I(t) = I_0 \cdot u(t - t_0) \cdot \left(\frac{t - t_0}{\tau_\alpha}\right) \cdot e^{1 - \frac{t - t_0}{\tau_\alpha}} \quad \text{for } t \geq t_0 \geq 0 \tag{3.15}$$

where $I(t)$ describes the synaptic current following a synaptic spike at t_0. τ_a is

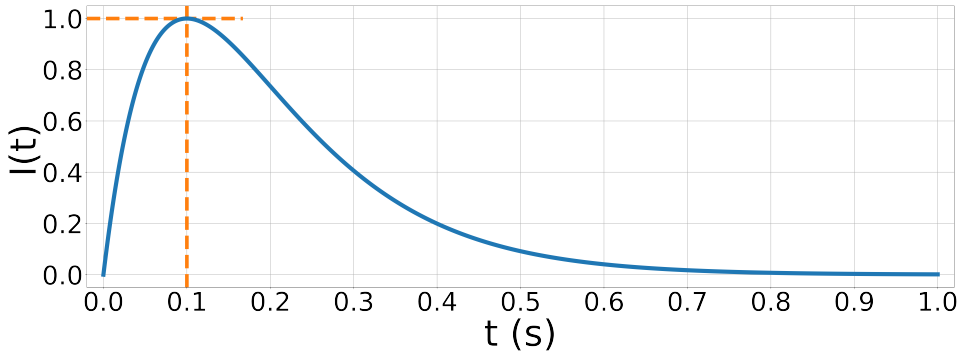


Figure 3.6: Alpha function ($\tau_\alpha = 0.1$ s, $I_0 = 1$, $t_0 = 0$).

the time constant governing the exponential behavior of the current waveform. An example of current-based excitatory alpha synapse behavior evoked by a spike's arrival is shown in Figure 3.6.

Equation 3.15 can be found as the solution of a second-order ODE system below:

$$\tau_\alpha \cdot \frac{\mathrm{d}x(t)}{\mathrm{d}t} = -x(t) \tag{3.16a}$$

$$\tau_\alpha \cdot \frac{\mathrm{d}I(t)}{\mathrm{d}t} = x(t) - I(t) \tag{3.16b}$$

$$x(t^+) \leftarrow x(t^-) + I_0 \cdot e \quad \text{upon spiking at } t = t_0 \tag{3.16c}$$

for $t \geq t_0 \geq 0$

Here $x(t)$ is an intermediate variable. After each presynaptic spike, $x(t)$ of a non-plastic synapse increases by $I_0 \cdot e$ and affects the synaptic current $I(t)$. After a presynaptic spike, $I(t)$ increases to reach the maximum value I_0 at time $t - t_0 = \tau_\alpha$. When there are multiple presynaptic spikes, the synaptic current is given by:

$$I(t) = I_0 \cdot \Sigma_k u(t - t_k) \cdot (\frac{t - t_k}{\tau_\alpha}) \cdot e^{1 - \frac{t - t_k}{\tau_\alpha}}$$

$$\text{where } k = 0, 1, 2, 3, ..., (n-1) \tag{3.17}$$

$$\text{for } t \geq 0, t_{n-1} > t_{n-2} > ... > t_0 \geq 0$$

where t_k is the time of $(k+1)^{th}$ synaptic spike. The step function in Equation 3.17 is to model a causal behavior in which each term related to the spike at t_k must be valid for $t \geq t_k$. Equation 3.17 is the solution of the second-order ODE below:

$$\tau_\alpha \cdot \frac{\mathrm{d}x(t)}{\mathrm{d}t} = -x(t) \tag{3.18a}$$

$$\tau_\alpha \cdot \frac{\mathrm{d}I(t)}{\mathrm{d}t} = x(t) - I(t) \tag{3.18b}$$

$$x(t^+) \leftarrow x(t^-) + I_0 \cdot e \quad \text{upon spiking at } t = t_k \tag{3.18c}$$

where $k = 0, 1, 2, 3, ..., (n-1)$

for $t \geq 0, t_{n-1} > t_{n-2} > ... > t_0 \geq 0$

Let us now consider the case in which synapses have STD. When two consecutive spikes arrive within a short period, the amount of synaptic current generated by the second spike is less than that of the first. The degree of reduction generated by two consecutive spikes is set by PPD, as described by Equation 3.10 and 3.11. Thus, upon the arrival of the second spike, I_0 is multiplied by the value of PPD. The spiking neuron and STD synapse dynamics are described in the following evolution: the first spike occurs at t_0. When the first presynaptic spike arrives at the time t_0, $PPD(t_0^-) = 1$. $I_0 \cdot e \cdot PPD(t_0^-)$ needs to be added to $x(t)$,

which affects the synaptic current $I(t)$. As the postsynaptic neuron's input, $I(t)$ affects the postsynaptic neuron's membrane potential $V(t)$. After the first spike, $PPD(t)$ immediately sets to U and tends to recover to 1. When the second spike arrives at t_1, $I_0 \cdot e \cdot PPD(t_1^-)$ is added to $x(t)$. The full recovery time of $PPD(t)$ is relatively short, which is an important characteristic of STP. The depression happens as $PPD(t) < 1$. Therefore, the STD synapse dynamics modeled by $PPD(t)$ and $I(t)$ evoked by two spikes at t_0 and t_1 is given by:

$$PPD(t) = \begin{cases} 1 & t \in [0, t_0] \\ 1 - (1 - U \cdot PPD(t_0^-)) \cdot e^{-\frac{t-t_0}{\tau}} & t \in [t_0, t_1] \\ 1 - (1 - U \cdot PPD(t_1^-)) \cdot e^{-\frac{t-t_1}{\tau}} & t \in [t_1, \infty) \end{cases} \tag{3.19}$$

Note that $PPD(t_0^-) = 1$.

$$I(t) = \begin{cases} 0 & t \in [0, t_0] \\ I_0 \cdot PPD(t_0^-)(\frac{t-t_0}{\tau_\alpha}) \cdot e^{1-\frac{t-t_0}{\tau_\alpha}} & t \in [t_0, t_1] \\ I_0 \cdot PPD(t_0^-)(\frac{t-t_0}{\tau_\alpha}) \cdot e^{1-\frac{t-t_0}{\tau_\alpha}} + I_1 \cdot PPD(t_1^-)(\frac{t-t_1}{\tau_\alpha}) \cdot e^{1-\frac{t-t_1}{\tau_\alpha}} & t \in [t_1, \infty] \end{cases} \tag{3.20}$$

Equation 3.19 is the solution of ODE below:

$$\tau \cdot \frac{\mathrm{d}PPD(t)}{\mathrm{d}t} = 1 - PPD(t) \tag{3.21a}$$

$$PPD(t^+) \leftarrow U \cdot PPD(t^-) \quad \text{upon spiking at } t = t_0, t_1 \tag{3.21b}$$

for $t \geq 0, t_1 \geq t_0 \geq 0$

And Equation 3.20 is the solution of the ODE below:

$$\tau_\alpha \cdot \frac{\mathrm{d}x(t)}{\mathrm{d}t} = -x(t) \tag{3.22a}$$

$$\tau_\alpha \cdot \frac{\mathrm{d}I(t)}{\mathrm{d}t} = x(t) - I(t) \tag{3.22b}$$

$$x(t^+) \leftarrow x(t^-) + I_0 \cdot e \cdot PPD(t^-) \quad \text{upon spiking at at } t = t_0, t_1 \tag{3.22c}$$

for $t \geq 0, t_1 \geq t_0 \geq 0$

When there are n spikes, the STD synaptic current is described by:

$$PPD(t) = 1 - \Sigma_k \left(u(t - t_k) - u(t - t_{k+1})\right) \cdot \left(1 - U \cdot PPD(t_k^-)\right) \cdot e^{-\frac{t-t_k}{\tau}} \tag{3.23a}$$

$$I(t) = I_0 \cdot \Sigma_k u(t - t_k) \cdot PPD(t_k^-) \cdot (\frac{t - t_k}{\tau_\alpha}) \cdot e^{1-\frac{t-t_k}{\tau_\alpha}} \tag{3.23b}$$

$$\text{where } k = 0, 1, 2, 3, ..., (n-1)$$

$$\text{for } t \geq 0, t_{n-1} > t_{n-2} > ... > t_0 \geq 0$$

where t_k is the time of the $(k+1)^{th}$ spike and $PPD(t_0^-) = 1$. The step function term $\left(u(t - t_k) - u(t - t_{k+1})\right)$ in Equation 3.23 is a window function to set the valid range of each PPD term. Equation 3.23 is the solution of a set of ODEs below:

$$\tau \cdot \frac{\mathrm{d}PPD(t)}{\mathrm{d}t} = 1 - PPD(t) \tag{3.24a}$$

$$\tau_\alpha \cdot \frac{\mathrm{d}x(t)}{\mathrm{d}t} = -x(t) \tag{3.24b}$$

$$\tau_\alpha \cdot \frac{\mathrm{d}I(t)}{\mathrm{d}t} = x(t) - I(t) \tag{3.24c}$$

$$x(t^+) \leftarrow x(t^-) + I_0 \cdot e \cdot PPD(t^-) \quad \text{upon spiking at } t = t_k \tag{3.24d}$$

$$PPD(t^+) \leftarrow U \cdot PPD(t^-) \quad \text{upon spiking at } t = t_k \tag{3.24e}$$

where $k = 0, 1, 2, 3, \ldots, (n-1)$

for $t \geq 0, t_{n-1} > t_{n-2} > ... > t_0 \geq 0$

Figure 3.7 shows the difference in postsynaptic neuron membrane potential between non-plastic and STD synapses. Figure 3.7(a)(c)(e) shows the case of non-plastic synaptic $x(t)$, $I(t)$, and postsynaptic neuron $V(t)$. Figure 3.7(b)(d)(f) shows the STD synaptic $x(t)$, $I(t)$, and postsynaptic neuron $V(t)$. The behavior shown in these two cases, both evoked by three consecutive presynaptic spikes at t_0, t_1, t_2. In Figure 3.7(a) and (c), the increase of $x(t)$ and $I(t)$, generated by three spikes, are the same. In Figure 3.7(b) and (d), the second and third increase is less than the first one due to PPD. Since the time difference of $t_2 - t_1$ is large, the third increase of $x(t)$ approaches back to the original value because the synaptic weight is recovering, but it is still smaller than the first increase as $PPD(t)$ is yet to fully settle. The synaptic current generated by the second spike in Figure 3.7(d) is less than that in Figure 3.7(c). Therefore, in Figure 3.7(e), a spike occurs, but there is no spike in Figure 3.7(f).

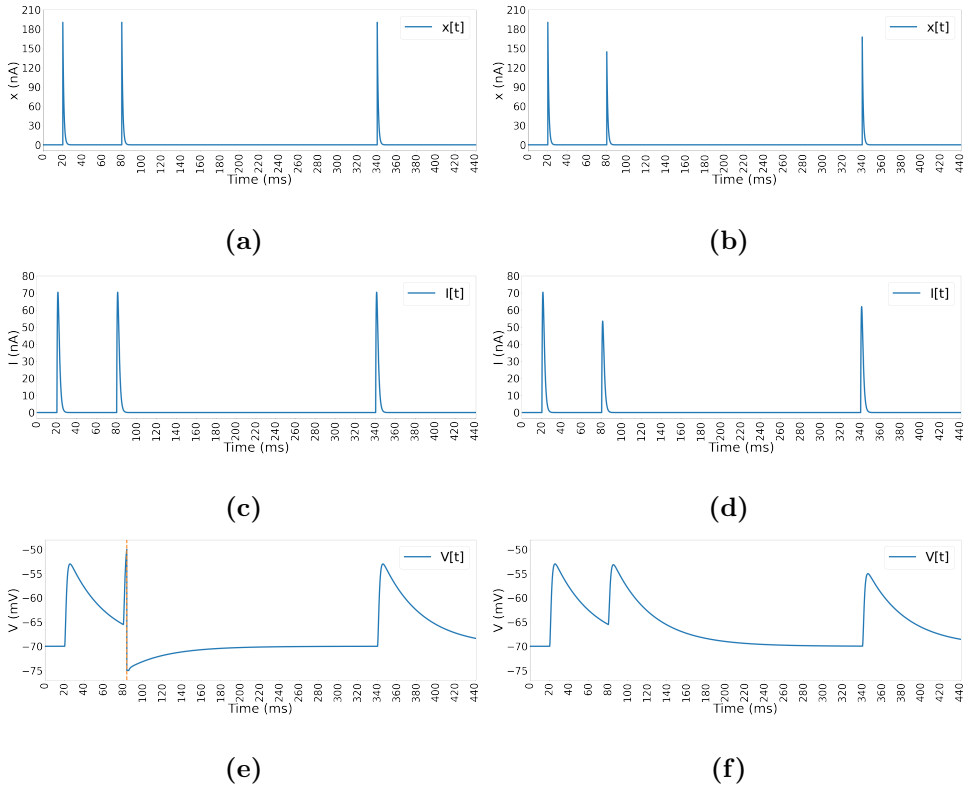


Figure 3.7: (a)(c)(e): The $x(t)$ and $I(t)$ of a non-plastic synapse evoked by three consecutive presynaptic spikes, and the $V(t)$ of the postsynaptic neuron. (b)(d)(f): The $x(t)$ and $I(t)$ of the synapse with STD evoked by three consecutive presynaptic spikes, and the postsynaptic $V(t)$.

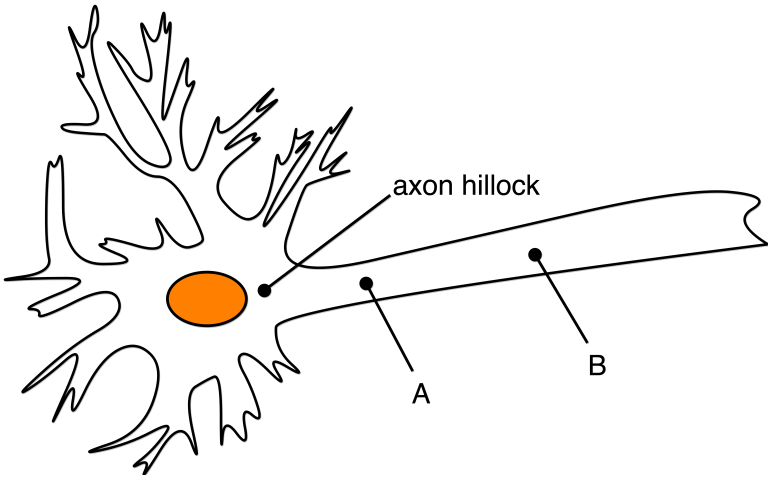


Figure 3.8: Axon delay model.

3.3.2.3 Axon Delay Model

The axon delay line model accounts for the propagation delay of electrical signals in axons shown in Figure 3.8. Point A is a point after the axon hillock where action potential originates, and point B is a point before an axon terminal. The propagation time delay between the A and B is the distance d between these two points divided by the speed s of the spiking signal [24], i.e., $t_{delay} = \frac{d}{s}$.

3.3.3 Sound Localization Neural Network

3.3.3.1 Neural Network Architecture

In the Jeffress model [80], the mechanism of SL is described with delay lines and coincidence-detecting neurons in the nucleus laminaris (NL) of a bird. The axon delay line models the propagation delay of spikes traveling through the axonal pathway. The spiking of coincidence-detecting neurons provides information on the directions of the sound source. Due to different traveling distances in dif-

ferent neurons, the spike train will arrive at neurons at different times. When two spike trains from the cochleae on both sides reach a specific output neuron simultaneously, the neuron generates the maximal number of spikes. The index of this neuron indicates the angle of the sound source. In this work, we call it the 'winner' neuron. This process is called coincidence detection, and the output neurons are coincidence-detecting neurons. The spiking rate of the input to ITD SL SNN will change when the sound intensity varies. STD is adjusted so that the number of spikes in output neurons is independent of the input spiking rate [41]. Based on this method, we propose an SL MSNN shown in Figure 3.9 to mimic the ITD SL in the auditory cortex. It is designed to cover the circular range of 180° with 1° resolution. The MSNN has three types of neurons: 2 input neurons, 181 output neurons, and 181 inhibitory neurons, and four types of synapses: excitatory synapses with no plasticity from input spike trains, excitatory synapses with STD from input neurons, excitatory synapses with no plasticity from output neuron, and inhibitory synapses from the inhibitory neuron. The first set of excitatory synapses is used to send input spike trains to input neurons. We have 2 such synapses. The second set of excitatory synapses with STD is used to send spike trains from input neurons to output neurons. We have 362 of such synapses. The third set of excitatory synapses sends spike trains from output neurons to corresponding inhibitory neurons. We have 181 of such synapses. The fourth set of inhibition synapses sends spike trains from the inhibitory neurons to neighborhood output neurons. The inhibitory neurons are used for lateral inhibition, an ability of firing neurons to inhibit neighboring neurons' activity [132, 168]. Each inhibitory neuron receives a spiking signal from

the corresponding output neuron, and can inhibit up to eight neighboring neurons, including four neurons to the left, and four neurons to the right. We define the neuron index starting with the one closest to the left input neuron. For example, inhibitory neuron 4 receives spike trains from output neuron 4, and inhibits output neurons 0, 1, 2, 3, and 5, 6, 7, 8. The corner neurons, such as inhibitory neurons 0 and 180 can only inhibit four neurons, inhibitory neurons 1 and 179 can only inhibit five neurons, inhibitory neurons 2 and 178 can only inhibit six neurons, and inhibitory neurons 3 and 177 can only inhibit seven neurons. The SL MSNN dynamics is as follows. Firstly, we provide the input neurons' spike trains with different spiking rates to input neurons. Two input neurons then generate their own spike trains that travel through their own axons and reach 181 output neurons from both ends of the network. If the timing of the spike trains from input neurons is such that it causes a set of spikes at some output neurons, the spike trains generated by the output neurons are transmitted to their corresponding inhibitory neurons. Finally, the inhibitory neurons suppress neighboring output neurons to which the output neurons are connected. We represent the propagation delay t_{delay} between the closest output neuron pair along the axonal pathway by the unit axon delay Δ. In our MSNN, lateral inhibition is significant for highlighting the winner neuron. The winner neuron will spike the highest number of times and inhibit the neighbors from increasing their numbers of spikes. Due to the 'cone of confusion' [123], we locate the sound source within 180 degrees, where a total of 181 output neurons work as coincidence detectors to achieve 1-degree resolution from 0° to 180°. Following the signal transmission sequence in the SL MSNN, we name the synapses relaying the

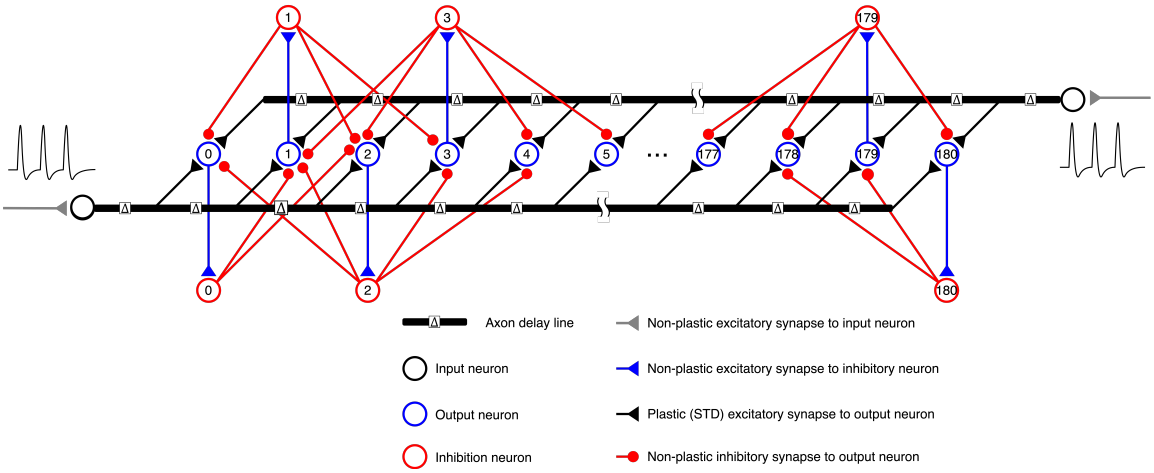


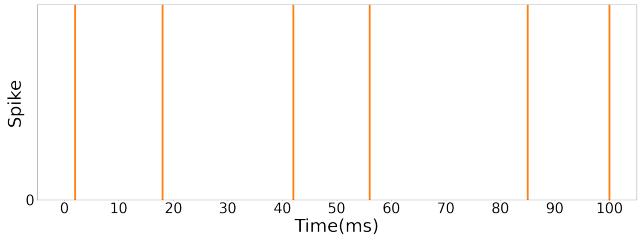
Figure 3.9: Sound localization neural network.

input spike trains to input neurons Synapse-I, the synapses from input neurons to output neurons Synapse-II, the synapses from output neurons to inhibitory neurons Synapse-III, the synapses from inhibitory neurons to output neurons Synapses-IV, V, VI, VII, where Synapse-IV is for the 1-degree neighbors, Synapse-V for 2-degree, Synapse-VI for 3-degree, and Synapse-VII for 4-degree.

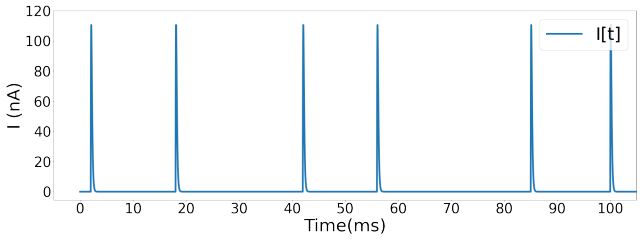
3.3.3.2 Sound Localization Memristive Spiking Neural Network Operation

Figure 3.10 shows how the input neuron in Figure 3.9 functions. Figure 3.10(a) shows a spike train to the input neuron. The mean input spiking rate is 50 spikes/s, and there are some random variations. Figure 3.10(b) shows the synaptic current into the input neuron generated by the input spike train. As we discussed earlier, the shape of the input synaptic current is modeled by the alpha function waveform. In Figure 3.10(c), the blue waveform shows the Excitatory Postsynaptic

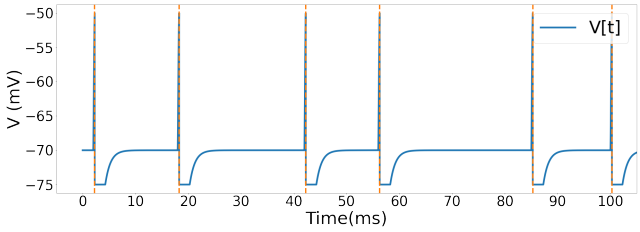
Potential (EPSP). When it reaches the threshold, it will generate a spike which is indicated by the orange dotted line. The activities of rest, integrate, fire, and reset are shown in Figure 3.10(c), where the membrane voltage rests at -70 mV, and integrates with the rising input synaptic current. When the input current rises, and the membrane potential reaches its threshold voltage at -50 mV, the neuron generates a spike. The membrane potential resets to -75 mV immediately after the spike and gradually returns to the resting potential.



(a)



(b)

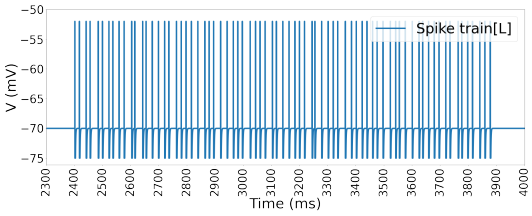


(c)

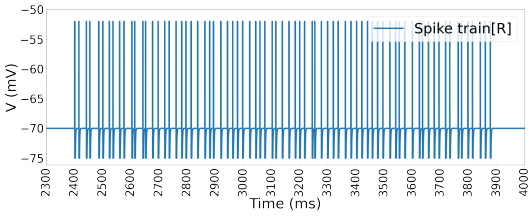
Figure 3.10: Spiking behavior of the input neuron. (a) Input spike train. (b) Synaptic current. (c) Membrane potential.

Figure 3.11 is an illustration of how the winner neuron in our SL MSNN works. The spike trains from the left and right input neurons propagate to all output neurons through their individual axon tracks. Figure 3.11(a) and (b) show the spike trains from the left and right input neurons propagated to the winner neuron, which is the output neuron 120 in this simulation. And Figure 3.11(c) presents the synaptic current to the output neuron 120, and Figure 3.11(d) illustrates the membrane potential of the output neuron 120, which has eleven spikes. The output neuron 120 receives the spike trains from the left input neuron and the right input neuron simultaneously, making it spike many times. The amplitude of the synaptic current is depressed. Therefore the EPSP is also decreased after the first EPSP due to STD in synapses. The output neuron 120 is inhibited due to a decrease in the synaptic current when its neighborhood neurons spike. However, this inhibition level is much weaker compared to the inhibition it provides to its neighborhood neurons.

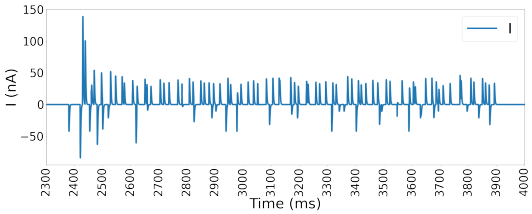
Figure 3.12 presents the membrane potentials of output neurons whose indices are 0, 45, 90, 120, and 150. The simulation is the same as Figure 3.11 so that the output neuron 120 shown in Figure 3.12(d) has the same performance as shown in Figure 3.11(d). The blue curve shows the membrane potential, and each orange-dotted line indicates a spike. From Figure 3.12(a) to (e), we see that two membrane potential waveforms invoked by two input spike trains approach each other, overlap, and separate from each other as the output neuron index increases. The two spike trains do not reach all neurons at the same time due to different delays. They arrive simultaneously only at the winner neuron, where two spike trains have the maxi-



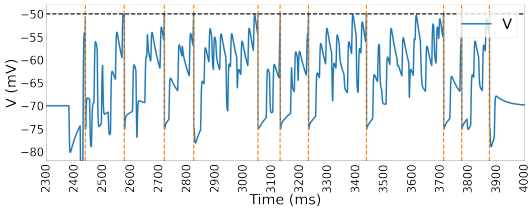
(a)



(b)



(c)



(d)

Figure 3.11: (a) The spike train propagated to output neuron 120 from the left input neuron. (b) The spike train propagated to output neuron 120 from the right input neuron. (c) The synaptic current to output neuron 120. (d) The membrane potential of the output neuron 120. Two spike trains reach output neuron 120 at the same time, producing 10 spikes.

mum overlap. The farther a neuron is away from the winner neuron, the greater the time difference is between the two spike trains. The greater the time difference between the two spike trains transmitted to a neuron, the less the probability that the neuron can integrate charge to reach the threshold. The membrane activity is near-symmetric when two neurons' indices have the same absolute difference from the winner neuron. They are not precisely symmetric due to the randomness in input spike trains. In the simulation shown in Figure 3.12, two spike trains arrive simultaneously at neuron 120, generating ten spikes. Other output neurons cannot integrate membrane potential large enough to generate as many spikes as the winner neuron does. As shown in Figure 3.12(a), the maximum arrival time difference of the two spike trains is 4800 ms at neuron 0, which is given by ITD of 1200 ms and the propagation time of 180×20 ms from the opposite side. At neuron 0, the membrane potential waveform gets excited twice due to two spike trains, one from the left side and the other from the right side, which arrive with a large time separation. At neuron 45 (Figure 3.12(b)), the two potential waveforms are separated by a shorter time. At neuron 90 (Figure 3.12(c)), the two potential waveforms overlapped. Furthermore, at neuron 120 (Figure 3.12(d)), the two potential waveforms are fully overlapped, generating a set of spikes. At neuron 150 (Figure 3.12(e)), the two potential waveforms separate and have only a small overlap. As shown in Figure 3.12(c) and (e), output neurons 90 and 150 have a near-symmetric membrane potential. We can see and infer from the 5 figures that the two voltage waveforms would meet and separate gradually as the index increases. In SL MSNN, the winner neuron index is determined by

$$Index = 90 \pm \frac{ITD}{2\Delta} \tag{3.25}$$

When the left input neuron first receives an input, ITD is positive; when the right input neuron first receives an input, ITD is negative. The factor of 2 in the denominator comes from the one-degree shift which corresponds to twice Δ, where Δ means the unit axon delay. In Figure 3.12, we set the left input neuron to receive its input 300 ms ahead of the right input neuron with Δ of 20 ms. Then, by Equation 3.25, the index of the winner neuron is $90 + 1200/(2 \times 20) = 120$. This verifies that the SL MSNN works correctly.

To achieve low power consumption, the system implementation needs to minimize the total number of spikes while still achieving coincidence detection. Thus, our design goal was set to a small number of spikes while having accurate and precise ITD SL neural network dynamics. Based on the experimental results, we chose the parameters listed in Table 3.2.

Table 3.2: Sound localization neural network simulation parameters.

Parameter	**Value**
Resting potential	-70 mV
Reset potential	-75 mV
Threshold	-50 mV
Input neuron membrane resistance	1 M
Input neuron membrane capacitor	1 nF
Continued on next page	

Table 3.2 – continued from previous page

Parameter	Value
Output neuron membrane resistance	4 M
Output neuron membrane capacitor	10 nF
Inhibitory neuron membrane resistance	1 M
Inhibitory neuron membrane capacitor	1 nF
Unit time delay	20 ms
Refractory period	2 ms
U	0.6668
τ	183.8 ms
Synapses-I I_0	110 nA
Synapses-II I_0	70 nA
Synapses-III I_0	110 nA
Synapses-IV I_0	40 nA
Synapses-V I_0	30 nA
Synapses-VI I_0	20 nA
Synapses-VII I_0	10 nA
Synapses-I τ	0.1 ms
Synapses-II τ	1 ms
Synapses-III τ	0.1 ms
Synapses-IV τ	1 ms
Synapses-V τ	1 ms

Continued on next page

Table 3.2 – continued from previous page

Parameter	Value
Synapses-VI τ	1 ms
Synapses-VII τ	1 ms

3.3.3.3 Sound Localization Memristive Spiking Neural Network Evaluation

In order to demonstrate the functionality of our SL system and to compare the contribution of different configurations, we configured four different settings for the system: (a) with neither STD nor lateral inhibition, (b) with only lateral inhibition, (c) with only STD, and (d) with both STD and lateral inhibition.

Figure 3.13 shows the number of spikes of output neurons for receiving the sound from 90 degrees apart from the left ear with three types of system settings. In this case, the winner neuron should be 90 degree, and it should have the maximum number of spikes. Each result contains nine sets of simulations using different input spiking rates ranging from 10 to 50 spikes/s and having the same length of 1500 ms. The input spike train has some variance due to its random behavior, and thus the interval is not exactly uniform. In Figure 3.13(a), we implemented neither STD nor lateral inhibition. With this setting, the SL MSNN cannot recognize the sound source for most of the input spiking rate, failing to achieve a 1-degree resolution. The numbers of spikes are large, ranging from 15 to 75. Additionally, the number

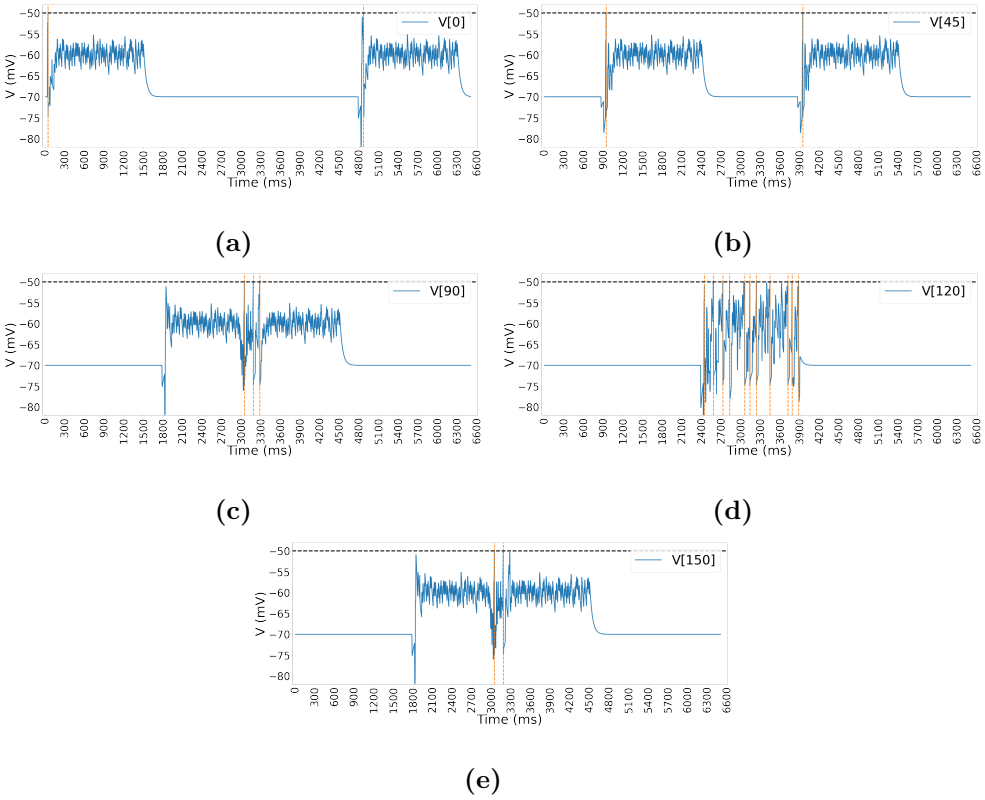


Figure 3.12: Membrane potential and spiking behavior of the output neurons with (a) index 0, (b) index 45, (c) index 90, (d) index 120, and (e) index 150. Indexes start with neurons closing to the left input.

of spikes depends on the input spiking rate. For each neuron index, a larger input spiking rate will generate a larger number of spikes, and a smaller input spiking rate will evoke a smaller number of spikes. In comparison. Figure 3.13(b) shows the result with only lateral inhibition. With this setting, the resolution is not increasing as the output neurons near the left and right input neurons are wrongly highlighted. The number of spikes among all output neurons decreases because of inhibitory connections. Figure 3.13(c) presents the result with only STD and it shows that only the winner neuron spikes the maximum number of spikes. Therefore, we can locate the sound direction within a 1-degree resolution by finding the winner neuron. Moreover, the number of spikes for the winner neuron sharply decreases to 15-18 spikes, and the number of spikes is no longer dependent on the input spiking rates. (i.e., the numbers of spikes are within a small range for the same neuron index with different input spiking rates). However, the numbers of spikes of the output neurons are still relatively large. Additionally, the number of spikes for the winner neuron is not much larger than those of other neurons, which may cause some problems as input spike trains to SL MSNN have random behavior. Figure 3.13(d) shows the results with both STD and lateral inhibition. In this case, we can also find only the winner neuron spikes the maximum number of spikes, so that we can locate the sound direction within 1-degree resolution according to the spiking neuron index. Besides, the winner neuron generates a much larger number of spikes compared with other neurons. With only STD, the winner neuron's number of spikes ranges from 15-18, and neuron 91's number of spikes ranges from 3-17. With both STD and lateral inhibition, the winner neuron's number of spikes ranges from

8-12, and neuron 91 number of spikes ranges from 0-4. That is to say, the winner neuron is highlighted, and other neurons are inhibited. It also helps reduce power consumption when compared to the previous cases. Comparing Figure 3.13(a) with (c), we can see the effect of the depression achieved by STD. STD can help locate the winner neuron, and it can drastically reduce the number of spikes so that the power consumption is decreased. Additionally, for different input spiking rates, the effect of STD makes neurons generate spikes in a small range of numbers. We can infer the same observation by comparing Figure 3.13(b) with (d). By comparing Figures 3.13(c) and (d), we can observe the effect of lateral inhibition on the spiking performance. With lateral inhibition, as the winner neuron has a larger number of spikes, the inhibition by the winner neuron is stronger than other neurons, and thus the winner neuron is highlighted. We cannot infer the same observation by comparing Figure 3.13(a) with (b) so that we know lateral inhibition is not working without STD. Therefore, after the inter-comparison of Figure 3.13, we can make the following conclusions:

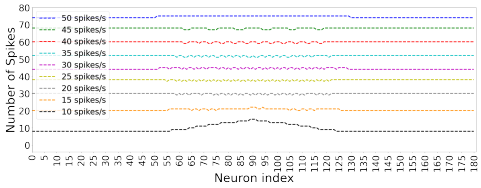
- Without STD and lateral inhibition, we cannot identify the direction. The number of spikes is large, and it depends on the input spiking rates.
- With only lateral inhibition, we cannot identify the direction. The number of spikes is still relatively large, and it depends on the input spiking rates.
- With only STD, we can locate the neuron and achieve the 1-degree resolution with the maximum number of spikes. Additionally, the number of spikes is sharply decreased and is no longer dependent on the input spiking rates.

- With both STD and lateral inhibition, we can locate the winner neuron and achieve the 1-degree resolution with the maximum number of spikes. The numbers of spikes in output neurons are not dependent on the input spiking rates, and output neurons spike with a relatively smaller number. Moreover, the winner neuron is highlighted compared with other neurons.

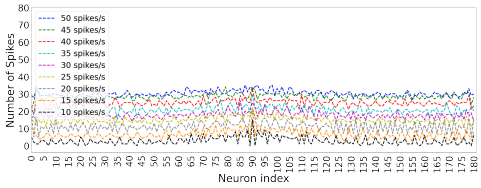
To evaluate the accuracy of the whole system, we investigated the sound signal from various degrees. We chose the 90, 120, 150, and 180 degrees as the representative cases to simplify the simulation process. For sounds from 120, 150, and 180 degrees as shown in Figures 3.14, 3.15, 3.16, we used three types of MSNN settings with input spiking rates ranging from 10 to 50 spikes/s, and the same length of 1500 ms. The input spike trains have different randomness for different sound sources. As shown in Figures 3.13(d)-3.16(d), our SL MSNN achieved the 1-degree resolution for all four sound source degrees. The system level configurations used in this simulation are the same for all different sound source degrees, including the neuron model, synapse model, Δ value, and other relative parameters. Although we chose 90, 120, 150, and 180 degrees as the representative input, the verification of these cases can be extended to other inputs.

3.4 Chapter Summary

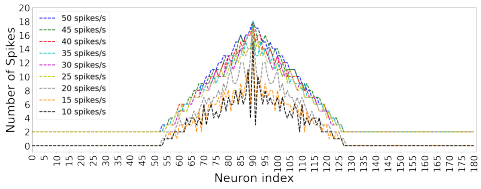
In this chapter, we illustrated that non-volatile memristors can emulate synapses with long-term plasticity, and volatile memristors can emulate synapses with short-term plasticity. We showed the application of a non-volatile memristor to an SL MSNN which can locate the sound source with a 1-degree resolution.



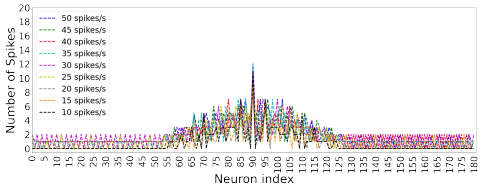
(a)



(b)

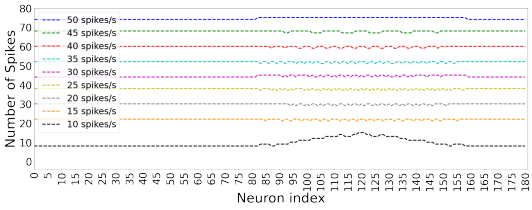


(c)

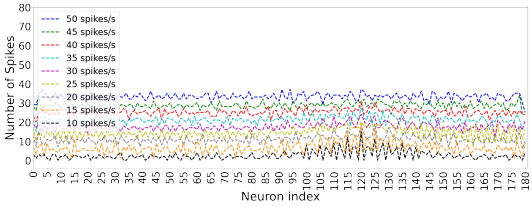


(d)

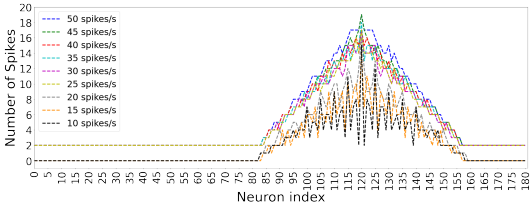
Figure 3.13: Spiking performance of output neurons with different settings receiving the sound from 90 degree. Input spike rates are 10-50 spikes/s. (a) Implementing neither STD nor lateral inhibition. (b) Implementing only lateral inhibition. (c)Implementing only STD. The winner neuron's number of spikes ranges from 15-18. Neuron 89's number of spikes ranges from 4-17 and neuron 91's number of spikes ranges from 3-17. (d) Implementing both STD and lateral inhibition. Winner neuron 90 number of spikes ranges from 8-12. Neuron 89's number of spikes ranges from 0-5 and neuron 91's number of spikes ranges from 0-4.



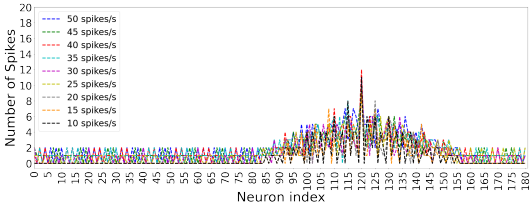
(a)



(b)

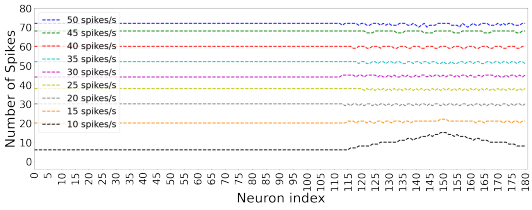


(c)

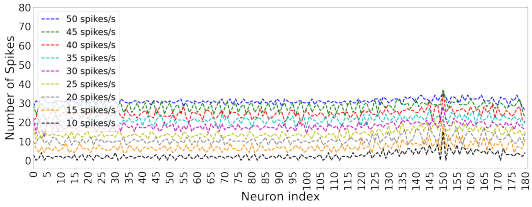


(d)

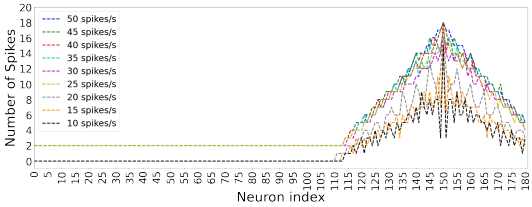
Figure 3.14: Spiking performance of output neurons with different settings receiving the sound from 120 degree. Input spike rates are 10 50 spikes/s. (a) Implementing neither STD nor lateral inhibition. (b) Implementing only lateral inhibition. (c) Implementing only STD. (d) Implementing both STD and lateral inhibition.



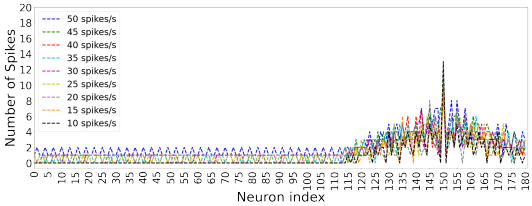
(a)



(b)

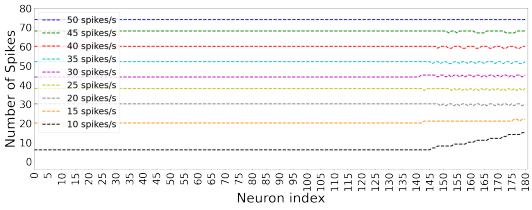


(c)

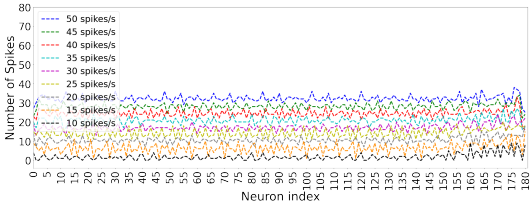


(d)

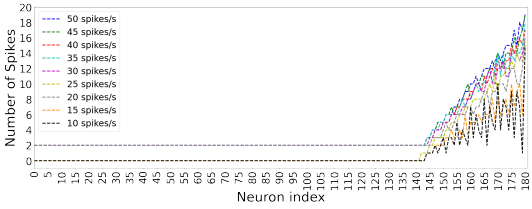
Figure 3.15: Spiking performance of output neurons with different settings receiving the sound from 150 degree. Input spike rates are 10 50 spikes/s. (a) Implementing neither STD nor lateral inhibition. (b) Implementing only lateral inhibition. (c) Implementing only STD. (d) Implementing both STD and lateral inhibition.



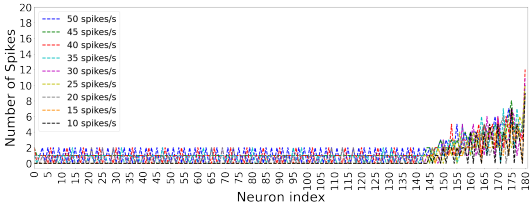
(a)



(b)



(c)



(d)

Figure 3.16: Spiking performance of output neurons with different settings receiving the sound from 180 degree. Input spike rates are 10 50 spikes/s. (a) Implementing neither STD nor lateral inhibition. (b) Implementing only lateral inhibition. (c) Implementing only STD. (d) Implementing both STD and lateral inhibition.

The STD synapses in the MSNN are simulated by volatile memristors. In order to minimize power consumption while achieving correct functionality, the number of spikes in the system is kept low. Currently, the volatile memristor modeled in this work has a relatively long recovery time so it may not be suitably used for a very accurate ITD SNN. Fortunately, there are more and more volatile memristors being proposed [152]. Therefore, a higher time resolution SNN may be implemented by quicker recovery volatile memristors in the future, and a practical application such as self-navigation may be possible.

Chapter 4

Memristive Neuron

4.1 Hodgkin–Huxley Model

There are multiple neuron models that describe the complicated dynamics of biological neurons, among which the Hodgkin–Huxley (HH) model [76] is one of the most pervasive models with a set of nonlinear differential equations for accurately approximating the electrical signals of neurons. The HH neural model is shown in Figure 4.1(a) where what they called time-varying nonlinear conductor $R_{Na}(G_{Na})$ and R_K (G_K) model the sodium and potassium channels, linear conductor $R_L(G_L)$ simulates leak channels, and C models the membrane of a neuron. The equations of the HH model are shown as follows:

$$c\frac{\mathrm{d}V_m}{\mathrm{d}t} = I_C(t) + \sum_k I_k(t) \tag{4.1}$$

where V_m is the membrane potential. $\sum_k I_k(t)$ is the sum of the ionic currents flow into the neuron, which can be formulated by three ion currents as:

$$\sum_k I_k = C_m \frac{\mathrm{d}V_m}{\mathrm{d}t} + G_K n^4 (V_m - V_K) + G_{Na} m^3 (V_m - V_{Na}) + G_L (V_m - V_L) \quad (4.2a)$$

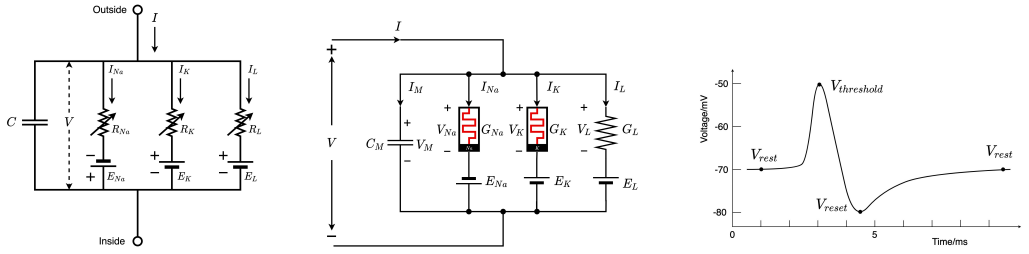
$$\frac{\mathrm{d}n}{\mathrm{d}t} = \alpha_n(V_m)(1-n) - \beta_n(V_m)n \quad (4.2b)$$

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m(V_m)(1-m) - \beta_m(V_m)m \quad (4.2c)$$

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \alpha_h(V_m)(1-h) - \beta_h(V_m)h \quad (4.2d)$$

where three parameters V_K, V_{Na}, and V_L are the reversal potentials. α_i and β_i are rate constants for the i-th ion channel, depending on membrane potential. $\bar{G}_K$, $\bar{G}_{Na}$, and $\bar{G}_L$ are the maximal value of the conductance. n, m, and h are dimensionless quantities between 0 and 1 associated with three ion channels.

The HH model is reduced by setting the leakage channel conductance to $G_L = 0$ for providing the best fit for human cardiac action potentials [112]. Chua and Kang proved that G_{Na} and G_K are memristors [39] and Chua et al. showed the memristive circuit [38] for that.



(a) **(b)** **(c)**

Figure 4.1: Hodgkin-Huxley neuron model (Hodgkin and Huxley, 1952). (a) An equivalent circuit for the HH models [76]. (b) An equivalent circuit for the memristive HH model [38]. (c) An action potential waveform showing rest-, threshold- and reset- potentials.

4.2 Leaky Integrate-and-Fire Model

The LIF [93,135] is one of the most broadly used simplified neuron models. The simple equivalent model is shown in Figure 4.2(a). In this model, a resistor R in series with a DC source V_{rest}/V_{reset} is connected in parallel with a capacitor C. A postsynaptic neuron receives a synaptic current $I(t)$ generated by presynaptic spikes. A portion of current $I(t)$ flowing into C makes the membrane potential $V(t)$ increase. The charge leakage is through resistor R. When $V(t)$ increases to reach a threshold value, the neuron generates a spike. After spiking, $V(t)$ is reset to V_{reset}. If $I(t)$ is absent, the voltage across C is settled eventually at V_{rest} representing the cell's resting potential. During the refractory period t_0, a neuron is incapable of spiking. Figure 4.2(c-d) illustrates the LIF neuron dynamics for the case of a DC input current and a zero rest and reset potential (E_{reset}=E_{rest}=0) [140].

The membrane potential dynamics before reaching the threshold is described by Equation 4.3.

$$\tau_m \frac{dV(t)}{dt} = -(V(t) - V_{rest}) + RI(t) \tag{4.3}$$

where $V(t)$ is the membrane potential $\tau_m = RC$ is the membrane time constant called 'leaky integrator'. V_{rest} is the resting potential and the initial value of $V(t)$ when $I(t) = 0$.

The input synaptic current $I(t)$ can be described by a time-varying alpha function, although different functions such as 'Instantaneous Rise and Single-Exponential Decay', 'Biexponential functions', 'saw tooth', and 'pulse function' can

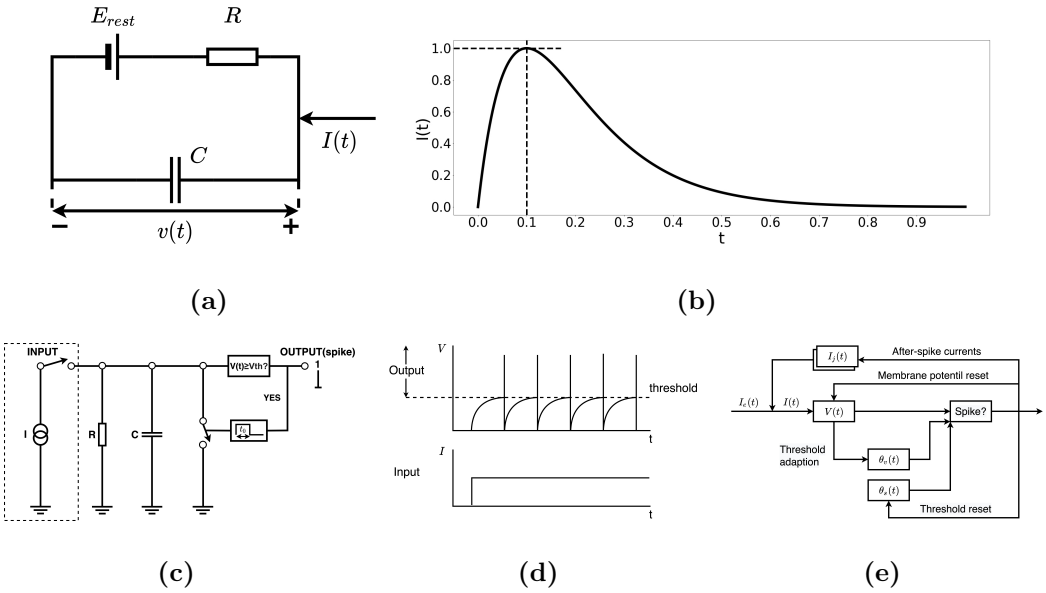


Figure 4.2: The LIF neuron model. (a) Schematic diagram of the LIF electrical model. (b) Input current in the form of an alpha function (τ_{alpha}=0.1, I_0=1) (c) Controlled spiking in the LIF model with a comparison of membrane potential and threshold at each time step. When a spike is triggered, a voltage-controlled switch discharges C for a duration of the refractory period t_0. Reproduced with permission from Tal and Schwartz, 1997 [140]. (d) A simulation of constant firing frequency for DC current input in Figure 4.2(c) in which t_0 is hidden. DC input current and output spikes are both shown. (e) A generalized LIF model with threshold control. Figure from Teeter et al. (2018) reproduced under a CC BY 4.0 license [144].

be used as alternatives. The alpha synaptic current is modeled by Equation 4.4, and the plot is shown in Figure 4.2(b).

$$I(t) = I_0 e((t - t_0)/\tau_\alpha) e^{-\frac{t}{\tau_\alpha}} \text{ for } t > 0 \tag{4.4}$$

where τ_α is the time constant governing the exponential behavior of the current and t_0 is when the spike happens. After a presynaptic spike, $I(t)$ increases to reach the maximum value I_0 in time $t - t_0 = \tau_\alpha$.

However, in Figure 4.2(a), there is no circuit for reset behavior when reaching the threshold. The inequality $V \geq V_{threshold}$ must be evaluated using an active circuit such as a comparator. Each time the threshold is reached, the membrane potential must be reset as in Figure 4.2(d). Thus, the LIF model's generalized version requires additional overhead, as depicted in Figure 4.2(e). In the generalized LIF model, reset behavior requires external control to pull down the membrane potential below the resting potential. This external control requires complex, active circuits consisting of MOS transistors, resistors, and even a Silicon-Controlled Rectifier (SCR), which is a drawback as it consumes much more power and area in a neuromorphic system.

4.3 Memristive Integrate-and-Fire Model

In order to overcome the drawback in conventional neuron models, memristor technologies can be harnessed to emulate biological neurons with the motivation of low energy consumption and high packing density. The HH model has been emulated by utilizing two memristors in parallel with two capacitors respectively

mimicking two channels, which are coupled to each other with a resistor. There is also an input resistor and an output impedance consisting of a resistor and a capacitor [117]. The LIF neuron has been reported by using a diffusive/volatile memristor in parallel with a capacitor, and the neuron was used in a fully memristive neural network [158].

Although some memristor-based neuron models have been proposed, there is a high barrier to accessing experimental data and hardware for prototyping memristive circuits. Many state-of-the-art experimental demonstrations have relied on specialized fabrication processes that cannot be reproduced by using off-the-shelf memristors. This is compounded by the limited choice of commercially available, low-cost, discretely packaged memristors known to be highly sensitive and stochastic in behavior. The challenge in developing hardware prototypes has made it difficult to perform experimental validation.

Beyond neural network acceleration, the design of a solid-state brain that can harness the neural code has received increasing attention as a way to handle huge sensory input data without being thwarted by the von Neumann bottleneck. The solid-state brain mimics the structure of the cerebral cortex by connecting neurons with a large fan-out of variable synapses. It is conjectured the neurons and synapses can be realized with memristors integrated with a CMOS process. However, the design of a large-scale solid-state brain remains elusive in neuromorphic computing due to the enormous overhead associated with mimicking the surface area of neural tissue, constrained power consumption, routing, and massive parallelism of synaptic connections.

We attempt to overcome several of these barriers by introducing the MIF model circuit for neurons. Firstly, we demonstrate the generation of spiking neuronal signals using our MIF and MIF2 models. The broad span of literature on memristive neuron circuits and neuristors is typically limited by either being simulation-only idealized studies, or otherwise relying on device-dependent characteristics not readily accessible to those in the broader research community. With similar neuron circuits presented in the past, we demonstrate the operation of the MIF and MIF2 circuits without the need for specialized fabrication processes that make our results reproducible by an amateur in hardware prototyping on a tight budget. We develop a circuit-theoretic foundation of our models that mimic the passive membrane model [93], and by extension, the model circuits demonstrate minimal complexity and integration area. An energy analysis is conducted based on our own experiments, showing that a scaled-up system with the same order of neurons as in the human brain would consume an approximately equivalent amount of power. We present our results with the hope that it fosters more experimental demonstrations of memristive circuits and systems.

4.3.1 MIF: Version 1

In place of the resistor R in the LIF model shown in Figure 4.2, the first version of the MIF model in Figure 4.3 uses a single memristor. This model does not require external adaptive control, to discharge the capacitor when the action potential reaches $V_{threshold}$. The MIF model is device-agnostic; as long as the energy supplied is sufficient for bipolar resistance switching, the circuit will be capable

of action potential generation. This is a significant improvement in view of the integration complexity and packing density, since the implementation of an adaptive control circuitry requires the CMOS substrate to be connected via holes to higher-layer memristors. Prior implementations of similar neuristor circuits are typically dependent upon restrictive volatile characteristics, with relaxation times far smaller than any RC delay present in the neuron [51, 117]. The MIF neuron model generalizes previous volatile memristor-based designs to reproduce the dynamics of biological neurons more closely.

Practical considerations of cycle-to-cycle variation and the ratio of pre- and post-switching resistances will serve to alter the spike shape. For a MIF circuit with a volatile memristor, a DC E_{rest}-source is sufficient to generate spikes under a sufficient input current. For instance, under DC current input of an appropriate magnitude, a MIF circuit, particularly with locally active memristors [17], can generate oscillation of spiking signals. If a non-volatile memristor with a negative reset voltage is present in MIF, then a suitable voltage pulse must be added to the DC E_{rest}-battery to reset the memristor back to its initial off state after emitting a spike. We can formulate a state equation for the MIF model in Figure 4.3(a) in terms of the voltage $V(t)$ across the capacitor C (equivalently, the membrane potential), the memristor resistance (or memristance) R_M, the voltage source E_{rest}, and the input current $I(t)$:

$$C\frac{V(t)}{dt} = -\frac{V(t) - E_{\text{rest}}}{R_{\text{M}}(x, V(t))} + I(t) \tag{4.5}$$

Here, $R_M(x,V(t))$ is a voltage-controlled memristance, where x represents the internal state of the device. Equation 4.6 describes a digital resistive switch (see Figure

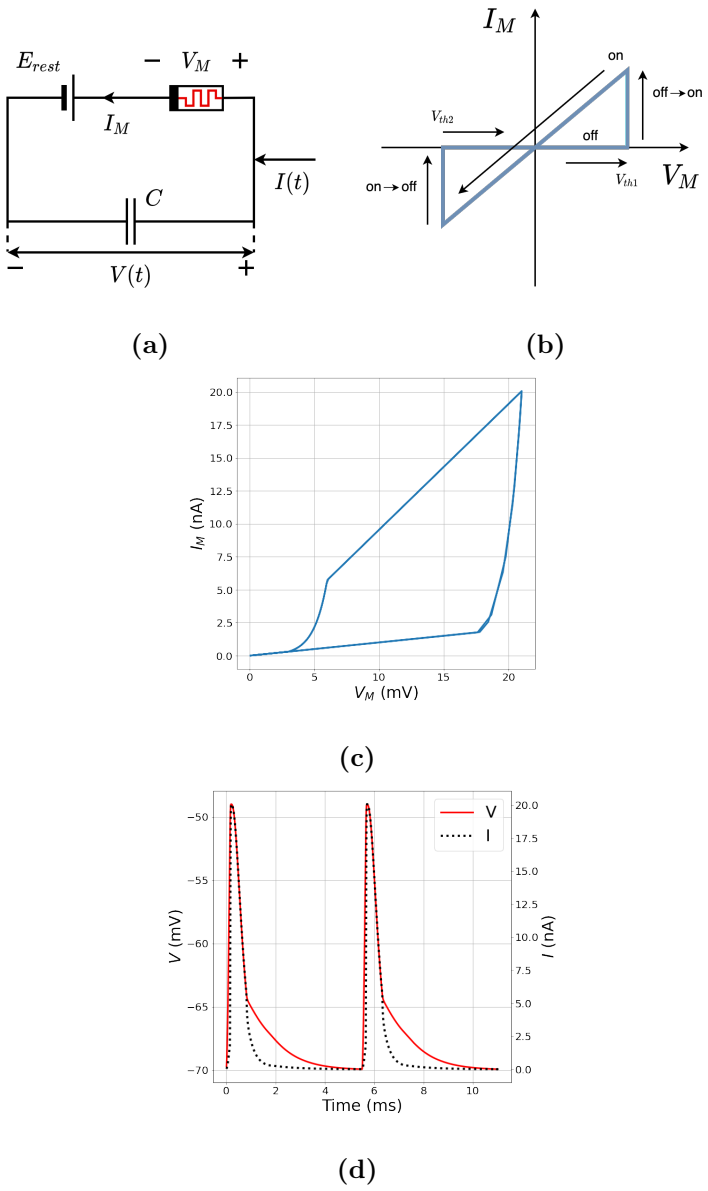


Figure 4.3: The MIF neuron model: version 1. (a) The MIF model replaces the R of the LIF model with one memristor. (b) I-V curve of a simple memristor with resistance switching. (c) The measured I-V curve of the memristor selector that is volatile and used to obtain the MIF simulation results in (d). In Figure 4.3(d), I(t) is the current described by Equation 4.4.

4.3(b) for an example of a simple I-V characteristic chosen for simple analysis). Note that, due to the difference between the device set and reset thresholds, we denote the memristor set voltage as V_{th1}, and the memristor reset voltage as V_{th2} in Figure 4.3(b). Furthermore, V_M is the potential across the memristor and $V_{threshold}$ is the membrane threshold voltage, which is distinct from both the memristor set and reset thresholds. The voltage across the memristor is the difference between the membrane potential and DC potential $V_M = V(t)$-E_{rest}. And as such, we can state that for the memristor with a simple resistance-switching I-V characteristic shown in Figure 4.3(b),

$$\begin{aligned} R_M &= R_{off} \text{ for } V_{th2} < V_M < V_{th1} \text{ as } V_M \text{ increases,} \\ R_M &= R_{on} \text{ for } V_{th2} < V_M < V_{th1} \text{ as } V_M \text{ decreases.} \end{aligned} \tag{4.6}$$

Assuming an initial membrane potential $V(0) = V_{rest}$, $V(t)$ may increase due to an incoming current spike. Accordingly, V_M will increase as long as the memristor state does not switch. Therefore, while $V_{rest} \leq V(t) \leq V_{threshold}$, for the case of Figure 4.3(b), Equation 4.5 can be recast as

$$C\frac{V(t)}{dt} = -\frac{V(t) - E_{rest}}{R_{off}} + I(t). \tag{4.7}$$

Equation 4.7 is valid in the case where the memristor features linear I-V characteristics for both on and off states. For nonlinear analog memristors, we use computer simulation for accurate circuit analysis. We used the Cadence/SPECTRE simulator for a detailed analysis of the MIF circuit with nonlinear I-V characteristics of a volatile selector shown in Figure 4.3(c) [61].

Figure 4.3(d) shows that as the capacitor voltage $V(t)$ rises and reaches $\mathrm{V_{threshold}}$, the voltage across the memristor V_M reaches the set-voltage $\mathrm{V_{th1}}$. When the memristor enters the set-state, R_M switches from $\mathrm{R_{off}}$ to $\mathrm{R_{on}}$. Following this change, the voltage across the capacitor starts to fall toward $\mathrm{V_{rest}}$, initially with a time constant $\mathrm{R_{on}C}$, and later $\mathrm{R_{off}C}$ since $\mathrm{V_{th2}}$ is non-negative and small in this volatile selector device. As the voltage across the memristor V_M falls towards $\mathrm{V_{th2}}$ close to zero, R_M switches back from $\mathrm{R_{on}}$ to $\mathrm{R_{off}}$. After $I(t)$ decreases to zero, the membrane potential settles at $\mathrm{V_{rest}}$. If a non-volatile memristor is used, then the DC $\mathrm{E_{rest}}$-source must be supplemented by simply adding a reset voltage pulse.

4.3.1.1 Minimality of the MIF Model

The MIF model is universally minimal in terms of the number of circuit elements and the integration area because no other model simpler than the MIF model composed of two passive circuit elements, C and memristor, and a DC voltage source E_{rest}*, can reproduce a spiking waveform featuring the two voltage levels of* V_{rest} *and* $V_{\mathrm{threshold}}$*. The MIF circuit can also generate oscillatory spiking signals for a locally active memristor with S-shape I-V characteristics under a DC input current of an appropriate magnitude.*

The MIF model with a locally active memristor can be driven to switch on and off continually due to the membrane potential, thus generating oscillations under a DC current stimulus. In the MIF circuit, even with a strictly passive memristor, a spiking voltage is generated by the memristor's resistance (R_M)-switching controlled by the charge build-up on the capacitor, which raises the memristor voltage V_M

to the set-voltage V_{th1}, and switches R_M to R_{on} for discharging with a fast time constant. If the memristor is volatile, it is autonomously reset after its recovery time and reaches its lowest level V_{rest}, reducing the memristor voltage V_M to V_{rest} - E_{rest} allowing the cycle to repeat, with the next voltage spike generated by the MIF circuit when a subsequent input current pulse turns the memristor on again. Even when the input current is DC, the MIF circuit with a locally active memristor can autonomously generate an oscillatory spiking output voltage waveform, which is comparable to Figure 4.2(d) in the generalized LIF model with additional control circuitry depicted in Figure 4.2(e). It should be noted that the MIF circuit does not require any additional control circuit. For instance, in the MIF circuit topology, the use of a locally active memristor with S-shaped I-V characteristics, i.e., a neuristor, has been used for the generation of oscillatory spikes under DC input current [117].

When a non-volatile memristor is used, as explained earlier, the DC E_{rest}-source must be supplemented by an additive narrow voltage pulse at an appropriate time with an amplitude larger than V_{th2}, where V_{th2} is the negative reset-voltage of the non-volatile memristor. No simpler circuit can produce a such action potential. Prior works require circuits with much more complex circuitry. Furthermore, the use of memristors consumes minimal layout area compared to the use of a large resistor.

4.3.2 MIF2: Version 2

In the typical waveform of the action potential from Figure 4.1(c), the action potential $V(t)$ starts at V_{rest}, rises to $V_{threshold}$, then resets to $V_{reset} < V_{rest}$,

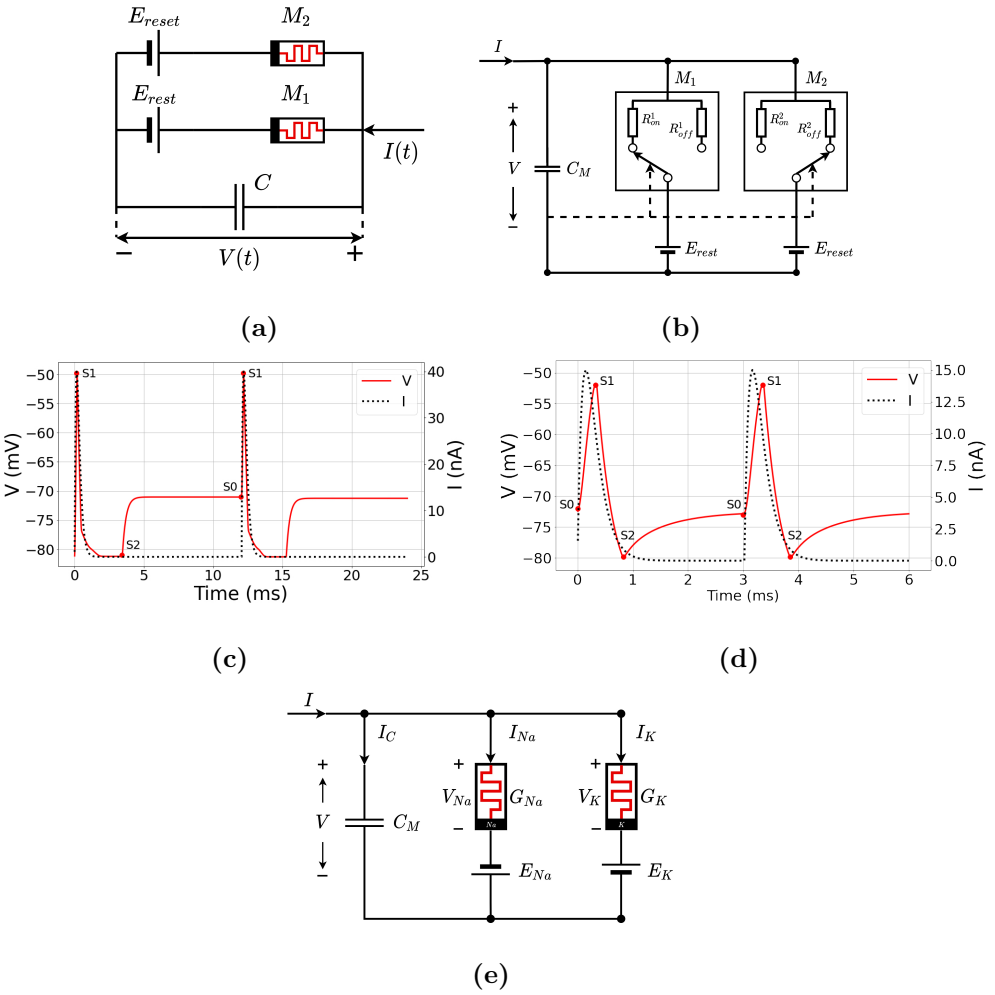


Figure 4.4: The MIF neuron model: version 2. (a) MIF2 model with two memristors and two DC voltage sources. (b) An illustration of MIF2 in which resistance switching of two memristors are controlled by V. (c) SPECTRE simulation results. (d) MIF2 simulation with the tuning of the model parameters. (e) Memristive HH model that is a reduced HH model with a symbolic representation of two conductive channels as memristors. In other words, in this memristive HH model the conductance equations remain the same as those in the HH model.

slowly increases and settles at V_{rest}. It thus traverses three critical voltage levels. The need for a sub-rest voltage level stems from the refractory period during neuronal information transmission. hyperpolarization suppresses the impact of an incoming stimulus on the neuron during the refractory period, and by driving the neuron to a voltage below the rest potential, it effectively raises the relative threshold, which the membrane capacitance voltage needs to attain for the generation of an action potential. While this may seem counterproductive, it prevents any stimulus already sent through the axon from triggering a backpropagating action potential in the neuron body. Therefore, hyperpolarization assures unidirectional signal transmission.

The MIF model in Figure 4.3(a) cannot simulate all three transitions between V_{rest}, $V_{threshold}$ and V_{reset}. Another voltage source E_{reset} is required. Therefore, we propose the MIF2 model shown in Figure 4.4(a). The use of a second memristor in series with a DC voltage source E_{reset} is derived from a circuit-theoretic view and the fabrication and area density considerations. The MIF2 circuit is experimentally verified in the following sections, which demonstrate the generality of the new memristor-based neuron design, including the frequency-dependent memristor signature.

Topologically, the model is identical to the reduced HH model in Figure 4.1(b) with $G_L = 0$ S [112]. The two DC voltage sources, E_{rest} and E_{reset}, correspond to the E_K and E_{Na} voltage sources in the HH model. The current paths through the two memristors correspond to the G_K and G_{Na} channels in the HH model. In this regard, the MIF2 circuit can be considered a macro-model of the reduced HH

model. This observation is consistent with the sequential opening of voltage-gated Na^+ and K^+ channels during action potential generation, which is described in detail by Kandel *et al.* [82]. To summarize, the MIF2 circuit is devised so as to accurately reproduce the analog waveform of an action potential, and to capture the main mechanisms behind the generation of a biological neuronal spike. While the MIF2 model draws inspiration from the HH model, it is not equivalent.

The following equation describes the MIF2 model, where R_{M1} and R_{M2} denote the resistances of the memristors connected to E_{rest} and E_{reset}, respectively:

$$C\frac{V(t)}{dt} = -\frac{V(t) - E_{\text{rest}}}{R_{M1}(x_1, V(t))} - \frac{V(t) - E_{\text{reset}}}{R_{M2}(x_2, V(t))} + I(t). \tag{4.8}$$

Equation 4.8 is valid for three different phases of the action potential waveform. As $V(t)$ goes through different stages, the state $x_{j,j=1,2}$ of one of the two memristors may change accordingly. The key transitions in the capacitor voltage are:

1. $V_{\text{rest}} - V_{\text{threshold}}$
2. $V_{\text{threshold}} - V_{\text{reset}}$
3. $V_{\text{reset}} - V_{\text{rest}}$

In each phase of the waveform $V(t)$ shown in Figure 4.4(c) and (d), memristor M_1 (M_2) may switch between the off-resistance state and the on-resistance state, depending upon the bias voltage across it, i.e., the difference between $V(t)$ and the DC voltage level E_{rest} (E_{reset}). Table 4.1 summarizes the state changes of the two memristors.

Table 4.1: States of the two memristors over the three phases of the membrane potential

Waveform Transition Phase	**M1 ($\mathbf{E}_{\text{rest}}$)**	**M2 ($\mathbf{E}_{\text{reset}}$)**	**Switching State**
V_{rest} - $V_{\text{threshold}}$	off	off	S0
$V_{\text{threshold}}$ - V_{reset}	off	on	S1
V_{reset} - V_{rest}	on	off	S2

For correct operation, MIF2 must satisfy the following requirements: after the peak potential at $V_{\text{threshold}}$ the membrane potential hyperpolarizes down to V_{reset}. At reset, the membrane potential V_{reset} is constrained by the DC reset potential E_{reset} and the potential across the memristor V_{M2}:

$$V(t) = V_{\text{reset}} = E_{\text{reset}} + V_{M2} \tag{4.9}$$

For the memristor M2 to reset, the voltage across it must be less than V_{th2}. Thus, the following constraint is imposed:

$$V_{\text{th2}} > V_{\text{reset}} - E_{\text{reset}} \tag{4.10}$$

In the next phase, the membrane potential rises toward V_{rest} from V_{reset}. The memristor M2, connected to the E_{reset}-source, must be off ($R_{M2} = R_{\text{off}}$) to avoid blocking the pull-up effort by M1 toward the E_{rest} level.

This autonomous spiking behavior can be achieved using volatile memristors appropriately biased via DC voltage sources and choosing a proper value for the membrane capacitance. The simulation results are shown in Figure 4.4(c).

$V(t)$ rises to the peak value, dips to V_{reset}, and then rises to V_{rest}, which is behaviorally correct. As qualitatively described earlier, in its dynamical evolution $V(t)$ goes through three distinct voltage levels, namely $V_{threshold}$, V_{reset}, and V_{rest}, as governed by the resistance switching phenomena occurring in the two memristors, as marked by S0, S1, S2 switching state in Figure 4.4(c).

Some circuit parameters can be tuned to ease the abrupt changes in the capacitor voltage $V(t)$, as shown in Figure 4.4(d), which displays a spiking voltage waveform strikingly similar to that in Figure 4.1(c). In this case, the memristor M1 remained in its off state. For the case with non-volatile memristors, a narrow voltage pulse need to be added to the E_{reset}-voltage source. Figure 4.4(d) shows simulation results of the MIF2 circuit for the case where M1 and M2 are identical volatile memristors, but not required to be so. The two DC voltage sources can be tuned to mimic the action potential waveform closely. With E_{reset} and E_{rest} values respectively set to -80mV and -65mV, the membrane potential displayed a rest potential of -72.5 mV, and a reset potential of -80 mV. For $V_{threshold}$ = -52 mV, the memristor M1, connected serially to the E_{rest}-battery, remains in the off state.

4.3.2.1 Minimality of the MIF2 Model

The MIF2 model consisting of one capacitor, and two memristors M1 and M2, connected to DC voltage sources E_{rest} and E_{reset}, respectively, is minimal in terms of the number of circuit elements and the integration area in generating the spiking neuronal signal waveform with three voltage levels, specifically V_{rest}, $V_{threshold}$, and V_{reset}. The MIF2 circuit can generate oscillatory spiking signals for

locally active memristors under a DC input current of an appropriate magnitude.

The MIF2 model, with two memristors, resembles the memristive HH model with battery-driven sodium and potassium channels. The MIF2 can be deemed a macro-model of the memristive HH model. It is topologically identical to the reduced memristive HH model shown in Figure4.4(e) [112].

The pair of voltage sources, E_{rest} and E_{reset}, in MIF2 reflect the rest- and reset potentials and correspond to E_K and E_{Na} of the HH model. Previously, it was shown that for the generation of a membrane potential waveform that does not dip to V_{reset}, the proposed MIF circuit was minimal in view of the number of circuit elements and the integration area. To include hyperpolarization dipping, the inclusion of one additional memristor is necessary. Otherwise, the drop to the V_{reset} level cannot be replicated without complex external control, as described in Tal and Schwartz [140]. Although a resistor may be chosen in lieu of a memristor to generate a new voltage level, such a choice cannot achieve the minimal integration area, nor replicate the spike-rate dependency of memristor I-V characteristics, the signature of memristors. This justifies that MIF2 is minimal in terms of the number of circuit elements and the integration area density. The two memristors in the MIF2 model can be volatile or non-volatile. The E_{rest} and E_{reset} sources can be purely DC for the case of volatile memristors. If a non-volatile memristor is used as M2, a reset pulse must be added to E_{reset}.

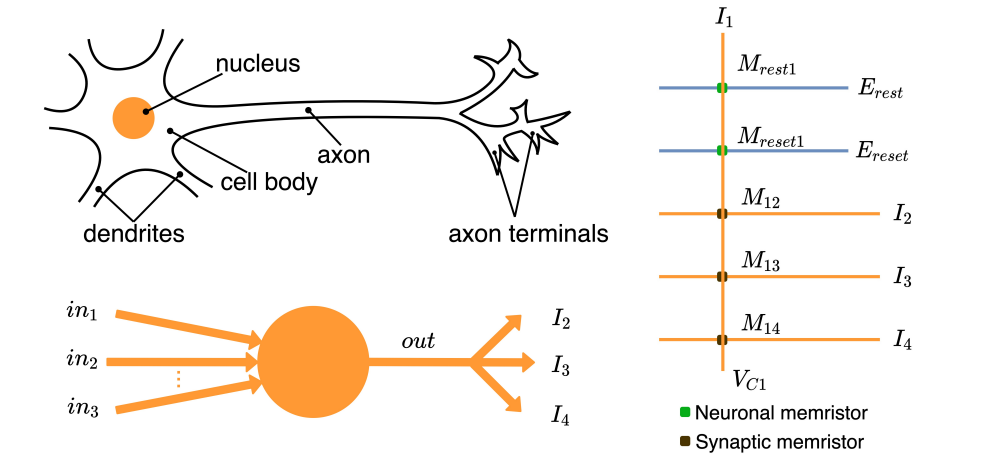
4.3.3 Discussion and Analysis

The pursuit for VLSI implementation of SNNs dates back to Mead's early work [104] and has been expanded since then [146]. A highly compact physical implementation of the LIF model was introduced by using a multitude of MOSFETs, resistors, and SCRs [125]. This section shows the most compact physical realization of the MIF circuit model from Figure 4.3(a). As discussed earlier, the use of nanoscale memristors provides the highest packing density as opposed to the use of resistors and other complex switching circuitry. It also should be noted that the pulse-rate (frequency) dependency of memristors in MIF models cannot be mimicked by using resistors.

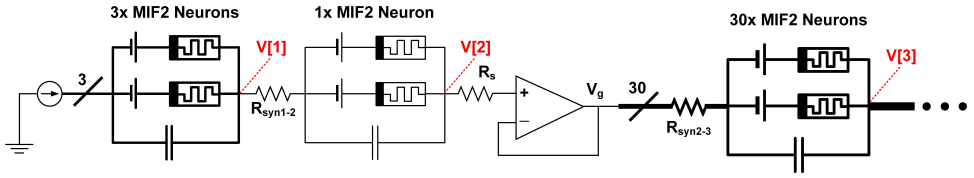
Interestingly, our parameter sweep analysis shows that resistive variation (which is typically the most dominant variation) does not have a significant effect on cycle-to-cycle variations. On the other hand, the reset and rest times are highly sensitive to the threshold voltage V_{th1}. This provides insight to device researchers that reducing the threshold voltage V_{th1} variation will prove to be more important for memristive neurons.

4.3.3.1 VLSI Implementation of Memristive Solid-State Brain

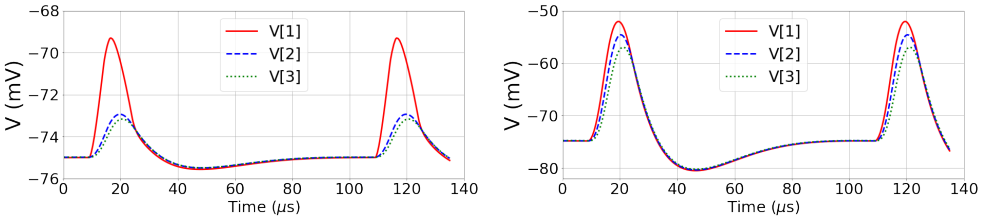
An idealization of the neuron is shown in Figure 4.5(a), as part of a general network representation (for example, in Sherman, 2004 [134]). A layout of the MIF2 neuron with a fan-out of three synapses is shown in Figure 4.5(b). The vertical metal line has a finite bit-line capacitance C_{bit}. This vertical line also sources the input current I_1.


nucleus
axon
cell body
dendrites
axon terminals
in_1
in_2
in_3
out
I_2
I_3
I_4
I_1
M_{rest1}
E_{rest}
M_{reset1}
E_{reset}
M_{12}
I_2
M_{13}
I_3
M_{14}
I_4
V_{C1}
Neuronal memristor
Synaptic memristor

(a) (b)


3x MIF2 Neurons
1x MIF2 Neuron
30x MIF2 Neurons
V[1]
V[2]
V[3]
3
30
R_{syn1-2}
R_{syn2-3}
R_s
V_g

(c)


V (mV)
Time (μs)
V[1]
V[2]
V[3]

(d) (e)

Figure 4.5 ***(previous page)*****:** Neural network implementation using the MIF model and synaptic memristors (represented by $R_{syn1-2,2-3}$). (a) A simple neuronal schematic with a fan-in and fan-out of three each. (b) The layout of the neuronal schematic in (a). V_{c1} is the voltage across the membrane capacitance in the MIF2 circuit excited by I_1. I_{2-4} are the fan-out currents of the neuron circuit. (c) Circuit topology of MIF2-based neuronal network with a high fan-out of 30 using a voltage follower interposed between neurons and synapses. (d) Simulation result of (c) without buffering shows that loading from subsequent stages attenuates the spike amplitude. V[1], V[2], and V[3] reflect the output voltage for each subsequent stage, i.e., the 1st, 2nd, and 3rd neural layer, respectively. (e) Simulation results of (c) with buffering shows that high fan-outs are possible owing to the high input/low output impedance of the voltage follower.

The resulting V_{c1} is the membrane voltage. M_{rest1} is the memristor in series with E_{rest}, and M_{1j}, j=2,3,4, is the synaptic memristor between the axon terminal connected to neuron 1 and the j^{th} fan-out neuron. Memristors are formed vertically at the cross-points. This layout shows that the entire neuronal network in Figure 4.5(a) can be laid out in a single column, thus taking up a small area which translates into a high on-chip packing density. More complex neural networks can be implemented in a vertically stacked 3D structure as in the brain. For the layout of the MIF2 circuit, a thin horizontal line for E_{reset} is inserted next to the E_{rest}-line, or vertically on top of it. In the following sub-sections, a simple analysis in terms of surface area and power consumption of a solid-state brain composed of our MIF neurons will be presented.

The high fan-out of biological neurons has been a long-standing challenge to emulate in silicon. The large capacitive loading reduces the slew rate of the neuronal signals, although not an issue where slower, biological timescales are used. Rather, the bigger issue is signal attenuation. With respect to neuronal circuits with high fan-outs, we have simulated a MIF2 circuit with three layers of neurons connected via synapses. The circuit topology is shown in Figure 4.5(c), where R_I = 0.1KΩ and R_{syn} = 1KΩ for all synapses. Here, for simplicity, we use linear resistors for synapses. The second layer is set to drive 30 neurons connected via synapses, where a voltage follower is used to buffer spike from the presynaptic MIF2 neuron. The follower is effectively used as a voltage-controlled voltage source (VCVS), which mimics the chemically driven replication of an action potential along the axon [82].

SPICE simulations of a MIF2 fully-connected network without buffering

are shown in Figure 4.5(d). Signal attenuation in a deeper layer is drastic, which motivates the decoupling of layers using a buffer. In many SNNs, spike timing rather than the spike waveform is the mechanism for neural encoding. In such cases, a simple digital spike read-out circuit would be appropriate, which would typically be implemented by a current-mode sense amplifier (or alternative arrangement of cross-coupled inverters) [20]. Obtaining a high fan-out of analog signals requires an analog buffer, with simulation results shown in Figure 4.5(e). Signal integrity is maintained through 3 layers of MIF2 neurons, even without buffering between the first and second layers.

Additionally, if we consider the design of a sparsely connected SNN, a typical neuron has a similar number of input and output synaptic connections which creates balanced fan-in and fan-out characteristics. Taking into account the relatively low-frequency spiking nature of SNNs, the necessary buffer circuitry has relaxed design considerations to ensure signal integrity through a deep network.

4.3.3.2 Surface Area Comparison

To benchmark area consumption, we consider the total surface area of a solid-state brain composed of MIF2 neurons, synapses, and interconnects. We can consider the following known facts about the human brain [147].

- Number of neurons = 10^{11}
- Number of synapses = 10^{14} – 10^{15}
- Surface area (median) = 2,400cm^2

- Volume occupied (median) = 1,050 cm^3

Figure 4.5(b) indicates that both the neuron and the synapse can be laid out using the same area of $4F^2$, where F represents the minimum feature size for the width of horizontal and vertical lines used in the crossbar array. For instance, at the 5 nm technology F = 5 nm, the minimum pitch of vertical lines and horizontal lines would be 2F, thus requiring an area of $4F^2$ for a neuron or a synapse with vertical stacking. Based on state-of-the-art reports on VLSI implementations of neuromorphic circuits, it can be conservatively assumed that, due to overhead such as interconnects, sensing and peripheral circuitry, and other requirements, $4F^2$ is multiplied by a factor of 5 [84,151]. Thus, each instance of a neuron or a synapse has an area of $20F^2$ allocated on-chip as a conservative estimate. In a simple estimation, the total area required to realize all neurons, synapses, DC voltage source lines, and ground lines is:

- Area = $20F^2 \times (10^{11} + 10^{15})$
- For F = 5nm, the area required would be:
- Area = $20 \times (5 \times 10^{-9})^2 \times (10^{11} + 10^{15})$ = 5,000 cm^2

By comparison, physiologically, the median surface area of the brain is 2,400 cm^2, which is smaller by a factor of 2.1. On the other hand, with F = 3.5 nm, the area becomes approximately the same. Thus, memristor technology integrated with an aggressively scaled fabrication process node can potentially enable the circuit realization of a solid-state brain within a surface area equal to that of a human brain. If the need for more complex circuitry demanded the availability of additional chip

area, multiple stacked layers could be used so as to prevent any further enlargement of the surface area. Since the biological brain has a 3D structure, the memristive solid-state brain can also be constructed vertically across a dozen stacked layers, with over approximately 0.3 cm thickness, which is comparable to the physiological thickness of the median human brain, where 1,050 cm^3 / 2,400 cm^2 = 0.44 cm.

4.3.3.3 Power Analysis

Equally important is the power consumption of the entire brain network. For this analysis, we need to evaluate the power of each neuron, of each synapse, and of each wire connecting array inputs to array outputs via neurons and synapses. Considering MIF neurons for simplicity, the average power consumption per neuron can be estimated by considering how the membrane potential builds up from V_{rest} to $V_{threshold}$ over the charge-up period t_{clamp}:

$$P_{\text{clamp}} = \frac{1}{t_{\text{clamp}}} \int_0^{t_{\text{clamp}}} I(t)V(t)\text{dt} \tag{4.11}$$

It is estimated that the human brain consumes 12-20 W for an assembly of about 10^{11} neurons and 10^{15} synapses. where n represents the number of synapses per neuron, an order of magnitude estimate can be derived as follows [47]:

$$\begin{aligned} P_{\text{brain}} &= N_{\text{neurons}} \times n \times V_{\text{spikes}} \times I_{\text{spike}} \times T_{\text{duration}} \times f_{\text{spike}} \\ &= 10^{11} neurons \times 10^{4} syn/neuron \times 10^{-1} V \times 10^{-10} A \times 10^{-3} sec \times 1Hz \\ &= 10W \end{aligned} \tag{4.12}$$

More precisely, calculations based on caloric intake place the upper ceiling estimate at 23.3 W [70] and 12.6 W as the floor estimate [118] which leads to the often cited 20 W value. A simple calculation of the power consumption per spike per neuron or synapse gives P_n = 10fW. To achieve low power consumption on the order of fW, a small I_0 in the pA range is needed. I_0 can be increased significantly if the brain power is assumed to depend mainly on the neuron power consumption. Then, the power budget per neuron can be increased by a few orders of magnitude.

As an illustrative example, we consider a case with R_{off} = 50 GΩ, C = 0.1 pF, E_{rest} = -70 mV, V_{rest} = -60 mV, $V_{threshold}$ = 30mV, I_0 = 2.5 pA. We find $\tau = R_{off}C$ = 5 ms and t_{clamp} = 7.63 ms. Therefore:

For simple analysis, with $V(t)$ approximated by using a triangular function, P_{clamp} can be estimated to be

$$P_{\text{clamp}} = I_0 \frac{V_{\text{threshold}} - V_{\text{rest}}}{2} \tag{4.13}$$

P_{clamp} is calculated to be P_{clamp} = 2.5 pA × 0.5(30+60) mV = 112.5 fW. The average power in the discharge period $P_{discharge}$ would be smaller by at least one order of magnitude since the memristor resistance is R_{on}, lower than R_{off} by at least one order of magnitude. Thus, the average power consumption P_n of MIF neuron over one period is

$$P_n = P_{\text{clamp}}(1 + R_{\text{on}}/R_{\text{off}})t_{\text{clamp}}/T_{\text{period}} \tag{4.14}$$

For the above example, if the total duration of one cycle of spiking and subsequent rest is T_{period} = 20 ms, and R_{on}/R_{off} = 0.01, then by Equation 4.14, P_n

= 112.5 fW × (1+0.01) × (7.63 ms / 20 ms) = 47.2 fW.

If a few parameters are changed, specifically to R_{off} = 50 MΩ, C = 10 pF, E_{rest} = -70 mV, V_{rest} = - 60 mV, $V_{threshold}$ = 30 mV, I_0 = 2.5 nA, then τ = 0.5 ms and t_{clamp} = 0.763 ms, thus P_{clamp} = 112.5 pW from (22), and P_n = 112.5 pW × 0.763 ms / 20 ms = 4.29 pW, which is higher by two orders of magnitude with respect to its value in the previous case, where I_0 was lower by three orders of magnitude. However, if neurons are assumed to dominate the power consumption, then an nA-range current would be considered reasonable.

It can be stated that:

$$P_n = f(I_0, C, R_{\text{on}}, R_{\text{off}}, E_{\text{rest}}, E_{\text{reset}}, V_{\text{rest}}, V_{\text{threshold}}). \tag{4.15}$$

To reduce the power consumption, the voltage swing $V_{threshold}$ – V_{rest} should be kept small. Then $R_{off}I_0$ can be lowered accordingly, which allows a smaller I_0. This observation is useful for the design of MSNNs, especially to specify memristor parameters such as R_{on}, R_{off}, V_{th1}.

If all neurons are assumed to spike within the same temporal window, the total power consumption for neurons and synapses per spiking period, $P_{n\&s}$, can be estimated as:

$$P_{n\&s} = 47.2 \times 10^{-15}(10^{11} + 10^{15})W \approx 47.2W, \tag{4.16}$$

and therefore, the total energy $E_{n\&s}$ <47.2 W × 20ms = 944 mJ.

The power consumed by neurons and synapses is estimated to be about 2/3 of the total brain power and about half of $P_{n\&s}$ is consumed by interconnects.

Thus, when all neurons and synapses are assumed to spike within a single cycle, it is reasonable to estimate that the brain power consumption P_{Brain}, with an average probability of spiking (generally estimated to be 30%), is

$$P_{\mathrm{Brain}} = 0.3 \times 1.5 P_{n\&s} = 21.2W. \tag{4.17}$$

This is in the ballpark of 20 W.

4.4 Chapter Summary

In this chapter, the introduction of MIF and MIF2 models is poised to enable MSNNs to be the most compact and least power consuming, comparable to the human brain with a median total surface area of 2,400 cm^2 and approximately 20 W power consumption. Both versions of the MIF models are based on passive memristors, but can generate oscillatory spiking signals with particular memristors. The MIF2 model and the reduced memristive HH model [112] are topologically identical, with parameters that are physiologically comparable. Our circuit-theoretic models are general, thus allowing a broader class of memristors to be implemented as MIF and MIF2 neurons.

The operating mechanisms of both MIF and MIF2 models are critically dependent on the resistance-switching nature of the memristors, for both their volatile and non-volatile realizations. Thus, any memristor with stable off- and on- resistances and attainable set and reset voltages can be used in the model circuits. For the generation of a spike train, we have shown how to excite the MIF and MIF2 neurons with an input train of current pulses of sufficient amplitude. With a locally

active memristor, MIF and MIF2 circuits can generate a train of spiking signals under a constant DC current stimulus of an appropriate magnitude.

As shown in a crossbar layout for both neurons and synapses, on-chip neural networks can be laid out compactly. It is estimated that by using a 3.5 nm technology, the entire solid-state brain could be laid out within a surface area comparable to that of the human brain. The estimate for the surface area was made for a single layer implementation and additionally for a multi-layered architecture with a total thickness lower than the thickness of the median human brain. The MIF model enables a systematic estimation of the spiking power. With some simplifying assumptions, the estimated total power consumption based on the MIF model is in the 20 W ballpark, as frequently cited in the literature. The realization of a memristive solid-state brain would become a tantalizing possibility with further advancements in nanoscale fabrication technologies for memristive systems.

Chapter 5

Memristive Spiking Neural Network with Unsupervised Learning

5.1 Background

Neuromorphic computing is guided by the rich neural dynamics present in the brain, the highly parallelized nature of neural computation, and the sparse encoding of data as spikes, in pursuit of optimizing memory and computation for energy efficiency. The pervasive von Neumann architecture disaggregates memory and computation which leaves much room for improvement for threads with a deterministic set of instructions. This deficiency has spurred the development of a variety of neuromorphic computing systems [28,33,44,62,63,106,109,113,128], which integrate spiking neurons and simplified synaptic models onto a silicon substrate. In almost all instances, synaptic weights are stored in random access memory (RAM), thus moving memory closer to a processor. But memory access and computation remain as two separate steps, which does not address the cost of data communication and

memory access, which impose the most overhead in both neuromorphic and general purpose computing systems.

The memristor has been presented as an option for merging the computation and memory substrates [37, 39, 138]. What was once a theoretical postulate proposed by Chua in 1971 [37] and generalized by Chua and Kang in 1976 [39] has become a commercially available technology that can be integrated in the back-end-of-the-line (BEOL) of modern CMOS processes [1, 3]. Their non-volatile retention capacity is often likened to synapses [32, 56, 81, 119, 131], and their threshold-switching characteristics are occasionally exploited as a spiking neuron model [58, 117, 170, 171]. For example, the modulation of device resistance has been correlated to synaptic plasticity, where achieving short-term plasticity (STP) and long-term plasticity (LTP) using memristors is as simple as applying programming pulse trains [153]. The benefits of memristive neurons and synapses arise from CMOS-compatibility, high-density, nanoscale vertical integration, and their ability to directly implement biological features, as opposed to requiring several arithmetic steps as with transistor-only circuits [43]. The use of SNNs as opposed to modern deep learning paradigms has shown significant energy and latency benefits as a result of activation sparsity, spike-based representations of data, and event-based data processing [19, 55, 57, 172, 174], and can be harnessed using the growing infrastructure to support SNN simulations [26, 45, 65, 74, 136].

Much of the prior work on MSNNs is constrained to using either memristive synapses *or* memristive neurons. The work in [158] demonstrated a fully integrated memristive system that emulated both synapses and spiking neuron models

to achieve simple pattern recognition tasks. The result is demonstrated using an in-house fabricated diffusive memristor, which is not readily accessible to the broader research community. In contrast to that, we wonder if it is possible to design a system with a fully memristive approach that shows the capacity to learn pattern recognition tasks using memristive neurons and synapses. Especially, We would like to achieve this by using models of commercially available, low-cost memristors [107]. Therefore, we integrate the memristive synapse with the STDP learning capability described in Chapter 3 and MIF/MIF2 neuron model illustrated in Chapter 4, such that we can simulate an MSNN with the unsupervised learning rule.

5.2 Methods

5.2.1 Memristive Integrate-and-Fire Model

The proposed MIF circuit is shown in Figure 5.1. It is characterized by the differential equations below:

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \frac{I - G_1(v - E_{rest}) - G_2(v - E_{reset})}{C} \tag{5.1a}$$

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = \frac{1}{\tau_1}\left(\frac{1 - x_1}{1 + e^{\frac{v_{on1} - (v - E_{rest})}{k_{th}}}} - \frac{x_1}{1 + e^{\frac{(v - E_{rest}) - v_{off1}}{k_{th}}}}\right) \tag{5.1b}$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = \frac{1}{\tau_2}\left(\frac{1 - x_2}{1 + e^{\frac{v_{on2} - (v - E_{rest})}{k_{th}}}} - \frac{x_2}{1 + e^{\frac{(v - E_{rest}) - v_{off2}}{k_{th}}}}\right) \tag{5.1c}$$

$$G_1 = \frac{x_1}{R_{on1}} + \frac{1 - x_1}{R_{off1}} \tag{5.1d}$$

$$G_2 = \frac{x_1}{R_{on2}} + \frac{1 - x_2}{R_{off2}} \tag{5.1e}$$

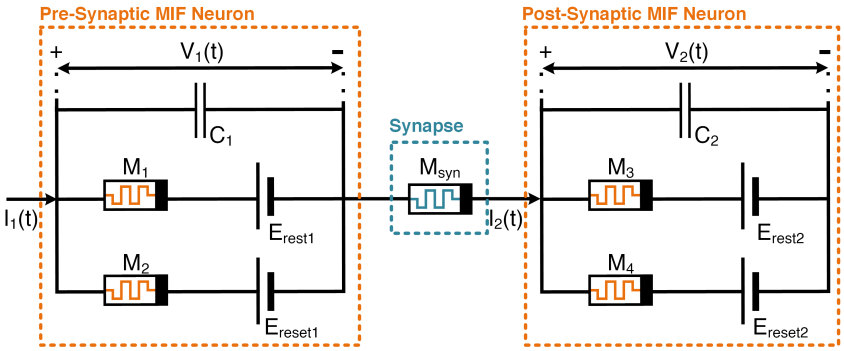


Figure 5.1: Presynaptic and postsynaptic memristive neurons with a trainable memristive synapse interposed between the two. The neuron model is a MIF neuron circuit, consisting of two memristors M_1 and M_2, connected to DC voltage sources E_{rest} and E_{reset}, in parallel with a capacitor C. Voltage spikes generated by the MIF neuron propagate through the synapse, and trigger an input current to the postsynaptic MIF neuron, which in turn will generate spikes.

where G_1 and G_2 are the memductances, x_1 and x_2 are a pair of internal states, τ_1 and τ_2 are time constants governing the rate of change in internal states, k_{th} is the effective thermal voltage. They are the characteristic variables of M_1 and M_2, respectively. This system of equations mirrors several prominent SPICE memristor models, and has been used to emulate the commercially available Knowm memristor [107].

To show the dynamics of the above system, we apply an alpha input current, with the resultant memristor internal states and voltage waveforms shown below in Figure 5.2:

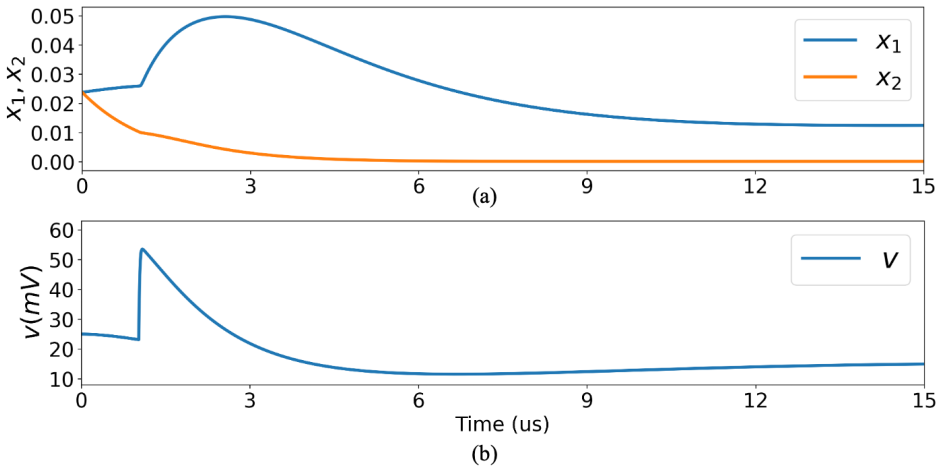


Figure 5.2: Simulation result of MIF receiving an alpha input current.

(a) Internal states x_1, x_2. (b) Voltage response v.

The alpha current is modeled by the Equation 5.2:

$$\tau_{syn}\frac{\mathrm{d}I}{\mathrm{d}t} = a - I \tag{5.2a}$$

$$\tau_{syn}\frac{\mathrm{d}a}{\mathrm{d}t} = -a + W_j \cdot \sum_f \delta(t - t_j^f) \tag{5.2b}$$

where W_j is a weighted synaptic current generated between presynaptic neuron j and associated postsynaptic neuron. $\sum_f \delta(t - t_j^f)$ indicates the total number of spikes emitted by presynaptic neuron j, and I is the input current.

The parameters used in this model are listed in Table 5.3:

5.2.2 Memristive Synapse with STDP

The Spike-timing-dependent plasticity (STDP) learning rule modulates synaptic weights based on the time difference of pre- and postsynaptic spike arrivals [29]. This type of learning rule can be achieved by using memristors [101,131], and can be verified using SPICE-level models [167]. As Figure 5.1 shows, a synap-

Table 5.1: MIF circuit parameters

Parameter	**Value**	**Parameter**	**Value**
E_{rest}	0 mV	E_{reset}	50 mV
C	100 pF	k_{th}	15 mV
v_{off_1}, v_{off_2}	0 mV	v_{on_1}, v_{on_2}	100 mV
R_{off_1}, R_{off_2}	0.1 MΩ	R_{on_1}, R_{on_2}	1 kΩ
τ_1, τ_2	1 μs		

tic memristor is interposed between two neurons, where the pre- and postsynaptic spikes will generate a voltage across the memristor that causes the memristance to be updated. Moreover, the time difference between pre- and postsynaptic spikes will modulate the changes in the memristance of the synapse.

In the memristive synapse with the STDP learning mechanism, the weight change will increase rapidly and then decrease slowly when the time difference of a pre- and postsynaptic spike increases from zero. The weight change of a memristive synapse is determined by the voltage across the memristor. The largest voltage will result in the largest weight change, and occurs when one of the pre- or postsynaptic neurons reaches the threshold level, while the other is at the reset voltage level. Therefore, the learning window of memristive STDP will rise with a fast time constant, followed by a slow decay. Thus, it is reasonable to model the memristive synapse behavior with an alpha learning window as:

To model the memristive synapse, we proposed the following model:

$$W_{pre}(x) = U_{pre} \cdot \frac{|\Delta t|}{\tau_{pre}} \cdot e^{-\frac{|\Delta t|}{\tau_{pre}}} \tag{5.3a}$$

$$\text{at} \quad t_{post} \quad \text{for} \quad t_{pre} < t_{post}$$

$$W_{post}(x) = U_{post} \cdot \frac{|\Delta t|}{\tau_{post}} \cdot e^{-\frac{|\Delta t|}{\tau_{post}}} \tag{5.3b}$$

$$\text{at} \quad t_{pre} \quad \text{for} \quad t_{post} < t_{pre}$$

where $\mathrm{U_{pre}}$, $\mathrm{U_{post}}$, τ_{pre}, τ_{post} are fitted constants, and $\Delta \mathrm{t}=\mathrm{t_{post}}\text{-}\mathrm{t_{pre}}$. Generally, the curve in the learning window will take an alpha shape in both the positive and negative planes of the x-axis.

This type of STDP can be modeled with a system of differential equations defined in Equation 5.4:

$$\tau_{pre}\frac{\mathrm{d}A_{pre}}{\mathrm{d}t} = T_{pre} - A_{pre} \tag{5.4a}$$

$$\tau_{pre}\frac{\mathrm{d}T_{pre}}{\mathrm{d}t} = T_{pre} + U_{pre} \cdot \sum_{f} \delta(t - t_j^f) \tag{5.4b}$$

$$\tau_{post}\frac{\mathrm{d}A_{post}}{\mathrm{d}t} = T_{post} - A_{post} \tag{5.4c}$$

$$\tau_{post}\frac{\mathrm{d}T_{post}}{\mathrm{d}t} = -T_{post} + U_{post} \cdot \sum_{n} \delta(t - t_i^n) \tag{5.4d}$$

$$W_j \leftarrow W_j + A_{post} \quad \text{upon presynaptic spike} \tag{5.4e}$$

$$W_j \leftarrow W_j + A_{pre} \quad \text{upon postsynaptic spike} \tag{5.4f}$$

where f indicates the total number of spikes emitted by the presynaptic neuron j, and n defines the total number of spikes emitted by the postsynaptic neuron i. $\mathrm{U_{pre}}$ is typically positive, while $\mathrm{U_{post}}$ is usually negative. W_j is the weighted synaptic

current. Upon the arrival of a presynaptic spike, W_j immediately increases by the amount of $\mathrm{A_{post}}$. $\mathrm{T_{pre}}$ induces a rise by the amount of $\mathrm{U_{pre}}$, which then evolves according to Equation 5.4(b). This in turn modulates $\mathrm{A_{pre}}$ in 5.4(a). Upon the arrival of a postsynaptic spike, W_j immediately increases by the amount of $\mathrm{A_{pre}}$. $\mathrm{T_{post}}$ increases by the amount of $\mathrm{U_{post}}$, and then affects $\mathrm{A_{post}}$ in a similar manner to the impact from a presynaptic spike.

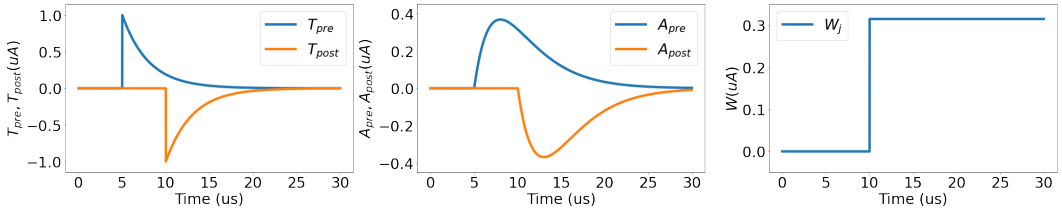


Figure 5.3: An example of the proposed memristive STDP with $\tau_{\mathrm{pre}} = \tau_{\mathrm{post}}$=3 μs, $\mathrm{U_{pre}}$=1 μA, $\mathrm{U_{post}}$=-1 μA. (a) $\mathrm{T_{pre}}$ and $\mathrm{T_{post}}$ in Equation 5.4 are determined by a pre- and a post-synaptic spike, respectively. (b) $\mathrm{A_{pre}}$ and $\mathrm{A_{post}}$ in Equation 5.4 are determined by a pre- and a post- synaptic spike, respectively. (c) The weight is updated according to Equation 5.4.

When there are multiple pre- and postsynaptic spikes, the weight should be updated according to Equation 5.5.

$$\frac{\mathrm{d}w_{ij}}{\mathrm{d}t} = U_{pre}A_{pre}\sum_{n}\delta(t-t_i^n) + U_{post}A_{post}\sum_{f}\delta(t-t_j^f) \tag{5.5}$$

where the first term on the right side denotes the pre-before-post effect and the second term indicates the post-before-pre effect. Usually, U_{pre} is positive and U_{post} is negative. An example of a weight update due to multiple pre- and postsynaptic neuron spikes is shown in Figure 5.4.

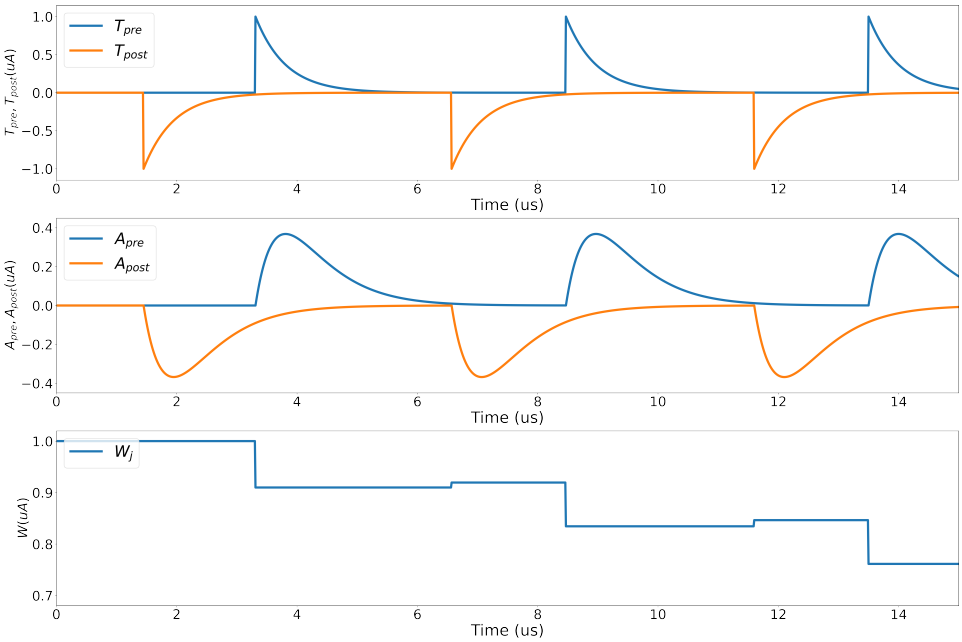


Figure 5.4: An example of weight update with multiple pre- and postsynaptic spikes.

5.2.3 Models Evaluation

We modeled a simple recurrent neural network as described in [49] to evaluate the functionality of the model. The network has 9 input neurons and 9 MIF neurons. There are 1-to-1 connections between the input and MIF layer. Within the MIF layer, neurons are fully connected with each other. Thus, there are 72 synapses as Figure 5.5 shows. For this neural network, we only used positive synapses.

We have two input patterns, one is 0, 4, 8 with high intensity while the other inputs have low intensity. The other pattern applies a high intensity to inputs 2, 4, 6, while the other inputs have low intensity. These two input patterns are transferred to input spike trains by rate coding. Generally, a higher intensity will be transferred to a Poisson input spike train with a higher spiking rate. By feeding

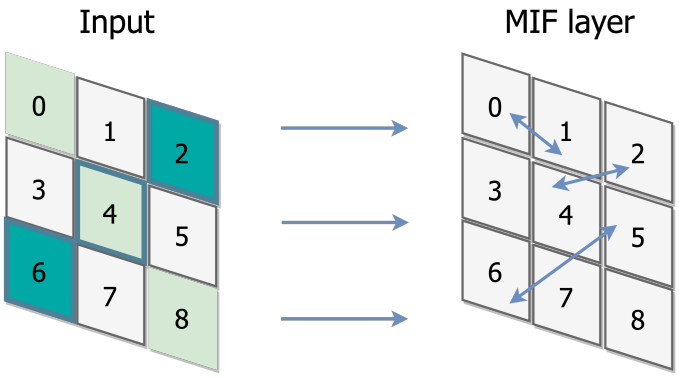


Figure 5.5: The architecture network.

two patterns successively to the neural network, the weight matrix is able to update according to the input patterns. A general view of the connections between 9 neurons after training is shown in Figure 5.6, which shows the connections between neurons 0, 4, 8, and 2, 4, 6 are enhanced.

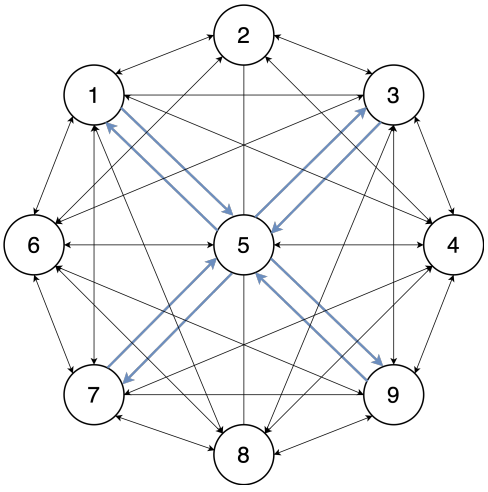


Figure 5.6: A general view of the connections inside the MIF neural layer after training.

Figure 5.7(a) and (b) also present the connection of this SNN. The results after training are shown in Figure 5.7(c), which is similar to the results in [49].

Therefore, the use of the MIF model in an SNN is validated.

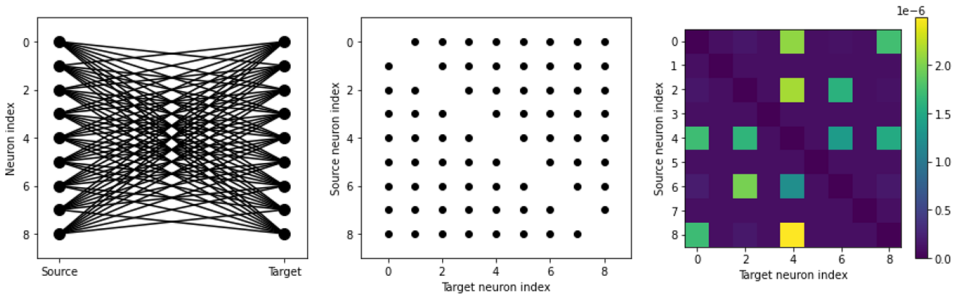


Figure 5.7: The results after training.

The parameters used in this network are listed in Table 5.2:

Table 5.2: Parameters in the MIF SNN for model evaluation.

Parameter	**Value**	**Parameter**	**Value**
w_{PE}	$2 \cdot e$ mA	I_{max}	$4 \cdot e$ uA
w_{EI}	$2 \cdot e$ uA	E_{rest}	-50 mV
w_{IE}	$2 \cdot e$ uA	E_{reset}	-100 mV
w_{EE}	$0.1 \cdot e$ uA	v_{rest}	-75 mV
v_{off}	5 mV	k_{th}	0.6 * 25 mV
v_{on}	110 mV	th_{MIF}	-50 mV
R_{on}	1 k	re_{MIF}	4 ms
R_{off}	0.1 M	τ_{alpha}	0.5 ms
C_{MIF}	100 pF	τ	1 ms
U_{pre}	0.1 A	τ_{pre}	5 ms

Continued on next page

Table 5.2 – continued from previous page

Parameter	Value	Parameter	Value
U_{post}	-0.05 A	τ_{post}	5 ms
Num_P	9	single train time	1000 ms
Num_E	9	rest train time	10 ms
defaultclock.dt	0.1 ms		

5.2.4 Type-1 MSNN for Memory Retrieval

Now we can implement a more complicated fully MSNN. The neural network architecture we implement with the fully memristive neuron and synapse models consists of three layers [49]. The input layer applies a Poisson spike train which encodes the input patterns via rate encoding. The second layer is a MIF excitatory layer. The third layer is an inhibitory layer and includes feedback connections to the excitatory layer. The architecture is shown in Figure 5.8. The input to the excitatory layer consists of excitatory synapses with fixed weights between the input layer and the MIF excitatory layer, with one-to-one connections. The output of the excitatory layer includes trainable (via STDP) excitatory recurrent synaptic connections, in addition to fixed-weight excitatory synapses between the MIF excitatory layer and the inhibitory layer, also with one-to-one connections. The inhibitory layer includes inhibitory feedback synapses with fixed weights between the inhibitory layer and the MIF excitatory layer with one-to-(all-1) connections

(not including the single MIF neuron connected to the inhibitory neuron). In this neural network, the numbers of neurons in all three layers are identical, and STDP is enabled within the MIF excitatory layer. The input patterns consist of 32×32 pixels, such that the total number of neurons in each layer is 1024. Each pattern generates a Poisson spike train of 35 μs duration, followed by a 15 μs refractory period without any input such that each input pattern does not affect the network dynamics upon arrival of the next input pattern. Each input spike will generate an alpha current where $\tau_{\text{syn}} = 10ns$. The threshold of each MIF neuron is set to 25 mV. As in the MIF neuron, there is no extra control circuit to force reset, and we set a refractory of 3 μs to prevent duplicating spikes during the simulation. We find the weight increase has a more significant effect when compared with the weight decrease, and it also enables the network to learn faster. Therefore, we set $\text{U}_{\text{pre}} = 10\mu A$ and $\text{U}_{\text{post}} = -0.1\mu A$. Note that $\text{U}_{\text{pre,post}}$ refers to an internal state variable that is modulated by the current-based neuron model, and is in dimensions of amperes. The recurrent connections are initialized with randomly weighted synaptic currents between 0 to 0.2 μA. The exact weight values between the excitatory layer and the inhibition layer do not have a large effect, which enables the inhibitory neurons to trigger a spike following output from excitatory neurons. On the other hand, the weights between input-to-excitatory layers and the inhibitory feedback mechanism both need to be carefully tuned to be neither too weak, nor too strong, to prevent complete suppression of downstream spikes, or excessive reinforcement of firing. We chose the fixed input to the excitatory weight to be 50 μA, and the fixed inhibitory layer feedback weight to be 20 μA.

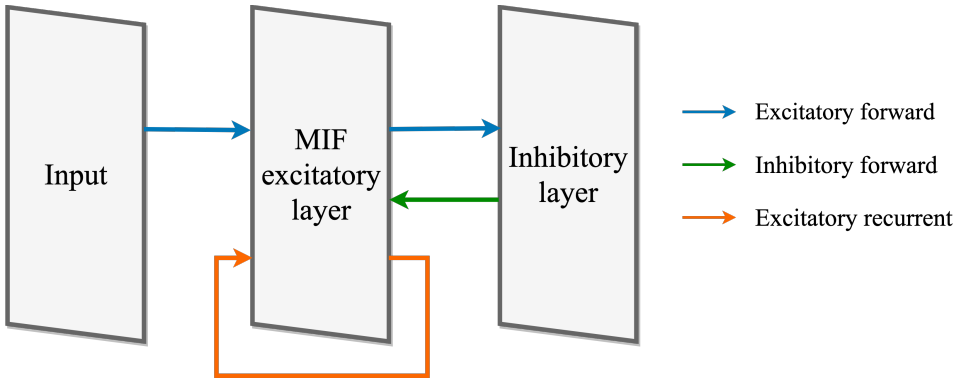


Figure 5.8: The MSNN architecture, which consists of the input layer, MIF excitatory layer, and the inhibitory layer. The blue arrows denote the excitatory synaptic forward connections, the orange arrow indicates excitatory synaptic recurrent connections, and the green arrow shows the inhibitory synaptic forward connection.

5.2.5 Type-2 MSNN for Pattern Recognition

In this section, we will introduce another MIF SNN architecture. Inspired by [50], the sequence of the MSNN for pattern recognition is identical to the memory retrieval network, in that we use a Poisson input layer, an excitatory layer, and an inhibitory layer. To improve unsupervised learning, we modify the nature of the synaptic connections. STDP-based learning is enabled between the input layer and the excitatory MIF layer with all-to-all connections. As before, each excitatory neuron connects to one inhibitory neuron and the inhibitory layer uses inhibitory synapses with fixed weights connected to the excitatory layer. In this network, no recurrent connections are present due to the lack of temporal dependencies in the pattern recognition task. The number of neurons between the excitatory and inhibitory layers must be identical, though they no longer need to match the input

layer. The architecture is shown in Figure 5.9. We use 1024 neurons in the input layer, and 320 neurons each in the excitatory and inhibitory layers, where STDP takes place between the input and the excitatory layers. We set $U_{pre} = 10nA$, $U_{post} = -0.1nA$, and the fixed inhibitory layer feedback weight to 200 μA. The weighted synaptic current between the input and excitatory layers is randomly initialization between 0 and 12 μA based on the empirical evaluation. Additionally, a random 0-2 μs delay between the input layer and the excitatory layer is assigned to avoid excessive simultaneous spiking, thus mitigating the exploding weights problem. All other neurons and input parameters are identical to the memory retrieval tasks.

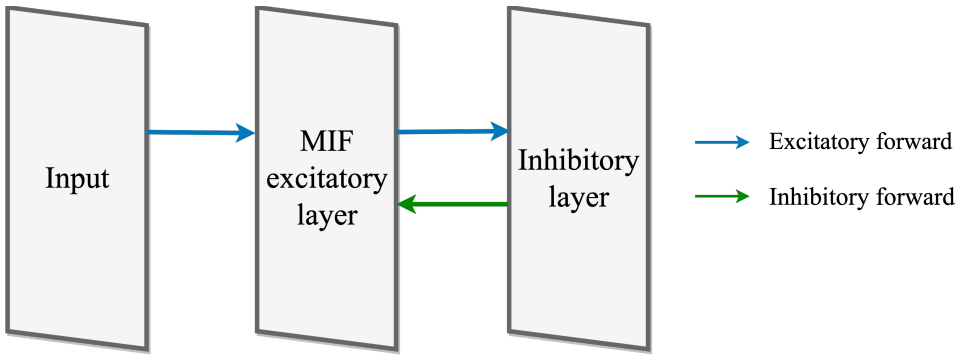


Figure 5.9: The second MSNN architecture.

5.3 Results

5.3.1 Type-1 MSNN for Memory Retrieval Result

We provided four different patterns to the MSNN successively for each iteration, and simulated across multiple epochs. This is to coarsely emulate how

biological systems experience real-world data in a batch size of '1' in online learning systems. The Poisson input layer converts each pixel intensity of the input pattern into a spiking probability of each neuron. We used four different patterns, as shown in the top row of Figure 5.10. The weight matrix is updated with the STDP rule. The training phase consists of 20 epochs for all patterns, corresponding to 80 iterations.

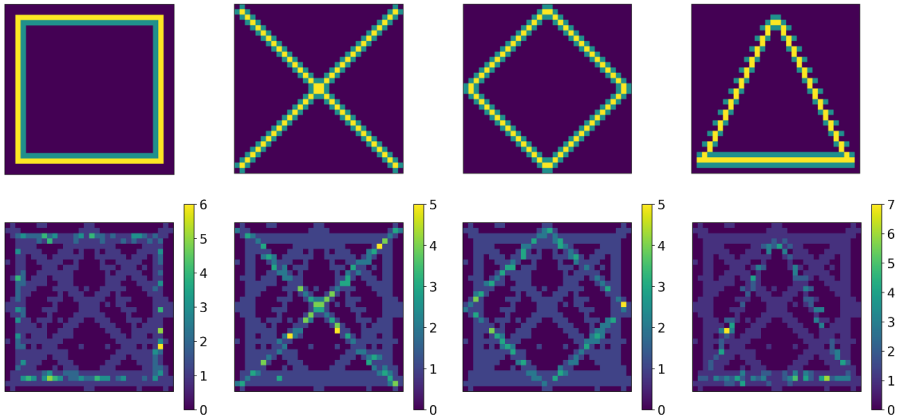


Figure 5.10: Top row: the four input spiking patterns (1:Square, 2: Cross, 3: Diamond, 4: Triangle) applied to the input of the MSNN. Bottom row: the number of spikes in the excitatory layer during the test stage, upon receiving these four input patterns.

After training, the same four patterns are successively applied to the MIF. The MSNN is expected to 'recall' the patterns as the connections have ideally made associations in the form of synaptic memory. When one of the patterns is applied to the MSNN again, other patterns are also recalled with less intensity, which is related to 'memory retrieval', illustrated in the bottom row of Figure 5.10. In order to evaluate memory retrieval performance, we calculate the percentage of overlapping

spike count with incorrect patterns, with the results shown in Figure 5.11. We find that when an input pattern is provided to the network, its resemblance to the spiking behavior with the target behavior is at a maximum. This is depicted in Figure 5.11, where the values along the diagonal are the largest. As shown in the bottom row of Figure 5.10, for each input pattern, the network continues to recall features of the other three patterns, which is a result of low-resistance pathways that cause overlap with other patterns, but with less resemblance to the target pattern.

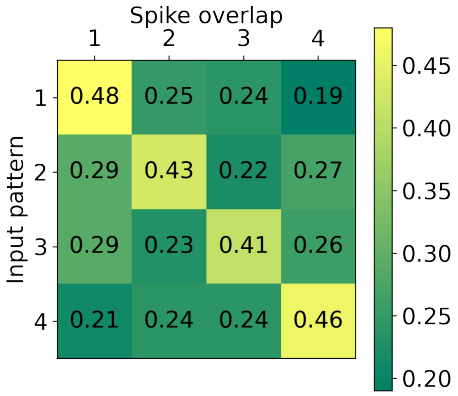


Figure 5.11: The resultant heatmap shows the memory retrieval resemblance for four patterns, which are 1:Square, 2: Cross, 3: Diamond, 4: Triangle.

The parameters used in this network are listed in Table 5.3:

Table 5.3: Parameters in the type-1 MSNN.

Parameter	Value	Parameter	Value
w_{PE}	50 A	τ_1	1 s
w_{EE}	0.2 A	th_{MIF}	25 mV
w_{EI}	3 mA	re_{MIF}	3 us
w_{IE}	20 A	U_{pre}	10 A
I_{max}	100 A	U_{post}	1 nA
E_{rest}	0 mV	τ_2	1 s
E_{reset}	50 mV	τ_e	0.01 s
v_{rest}	0 mV	τ_i	0.02 s
k_{th}	0.6 * 25 mV	C_{MIF}	100 pF
v_{on_1}	100 mV	v_{on_2}	100 mV
v_{off_1}	0 mV	v_{off_2}	0 mV
R_{on_1}	1 k	R_{on_1}	1 k
R_{off_1}	0.1 M	R_{off_1}	0.1 M
C_M	1 pF	τ_{pre}	2 us
R_M	10	τ_{post}	2 us
Num_P	1024	Inh_{th}	25 mV
Num_E	1024	Inh_{τ}	0.01 us
single train time	35 us	Inh_{reset}	0 mV
rest train time	15 us	Inh_{rest}	0 mV

Continued on next page

Table 5.3 – continued from previous page

Parameter	Value	Parameter	Value
defaultclock.dt	0.01 us	Inh_{re}	3 us

5.3.2 Type-2 MSNN for Pattern Recognition Result

During training, we applied the four different patterns (Figure 5.10) to the MSNN successively, then accumulated and recorded the number of spikes in each output MIF neuron for each pattern. During the training process, each MIF neuron is randomly assigned a pattern and updated after every 5 epochs. This allows each MIF neuron to be associated with a pattern such that it may be used to encode one of four patterns during the test stage. After assignments are updated, we use the current weights to test the accuracy of the network. During the test stage, the four patterns are passed to the network for 40 epochs, and the total numbers of spikes are counted for each pattern for all output excitatory neurons. The neuron with the highest spike count is deemed as the winner neuron for this input pattern. The accuracy evolution during training is shown in Figure 5.12, where a total accuracy of 97.5% is attained at the 70th iteration, shown by the dashed orange line. Early stopping is applied here, as further training causes excessive reinforcement of high-conductance pathways.

The parameters used in this network are listed in Table 5.4:

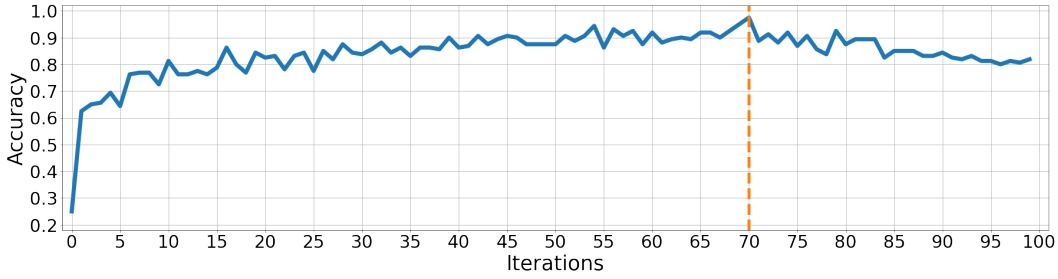


Figure 5.12: Accuracy across multiple iterations.

Table 5.4: Parameters in the second MIF SNN.

Parameter	Value	Parameter	Value
w_{EE}	0.2 A	th_{MIF}	25 mV
w_{EI}	3 mA	re_{MIF}	3 us
w_{IE}	200 A	U_{pre}	10 A
I_{max}	30 A	U_{post}	-0.1 nA
C_{MIF}	100 pF	τ_1	1 s
E_{rest}	0 mV	τ_2	1 s
E_{reset}	50 mV	τ_e	0.01 s
v_{rest}	0 mV	τ_i	0.02 s
k_{th}	0.6 * 25 mV		
v_{on_1}	100 mV	v_{on_2}	100 mV
v_{off_1}	0 mV	v_{off_2}	0 mV
R_{on_1}	1 k	R_{on_1}	1 k
R_{off_1}	0.1 M	R_{off_1}	0.1 M

Continued on next page

Table 5.4 – continued from previous page

Parameter	Value	Parameter	Value
C_M	1 pF	τ_{pre}	2 us
R_M	10	τ_{post}	2 us
$PE_{minDelay}$	0 ms	$PE_{maxDelay}$	2 us
Num_P	1024	Inh_{th}	25 mV
Num_E	320	Inh_{τ}	0.01 us
single train time	35 us	Inh_{reset}	0 mV
rest train time	15 us	Inh_{rest}	0 mV
defaultclock.dt	0.01 us	Inh_{re}	3 us

5.4 Chapter Summary

In this chapter, we proposed two different types of fully MSNNs for unsupervised learning, based on a MIF neuron model together with memristive synapses. The synapses and neuron models are designed using SPICE-level memristor models to relate circuit-level plausibility with the biological plausibility of spiking neurons. We demonstrated memory retrieval and pattern recognition across the four input patterns with the type-1 MSNN, and validated the potential of our fully-memristive approach across both tasks with the type-2 MSNN. Although we presented the results for four input patterns, they present an early validation that encourages future development. It is the first step towards designing fully MSNNs, with the

long-term aim of building a large-scale memristive brain. Future work will address more challenging tasks, such as testing memory retrieval for incomplete patterns, and increasing dataset complexity for the multi-pattern classification problems. Explorations of new learning paradigms that rely on error propagation may need to be integrated together with the STDP update rule to enable the success of more complex tasks. The proposed framework will render circuit-level implementations using a broad class of memristors to perform neuromorphic computing.

Chapter 6

Memristive Spiking Neural Network with Supervised Learning

6.1 Background

SNNs impose neuroscience-inspired constraints on modern deep learning algorithms, and have accordingly demonstrated significant improvements in run-time efficiency. By moving from full precision and fixed precision activations of artificial neuron models over to temporally-encoded data representations captured by spiking neurons, neuromorphic hardware has shown significant savings in energy consumption and latency [19, 143, 154, 164, 172, 173].

The broad success of error backpropagation to train deep learning models has ushered in a plethora of related training algorithms adapted for SNNs, most of which are guided by surrogate gradient descent to overcome the non-differentiability of discrete spikes [111, 161]. This proliferation of SNN usage is complemented by the development of modular deep learning programming packages that have optimized

autodifferentiation for CUDA acceleration [8, 9, 11, 72, 115].

In parallel to these advances in training SNNs, the past decade has seen huge strides in brain-inspired devices, circuits, and architectures that integrate neuronal dynamics to improve the hardware integration of SNNs and its constituent parts. Memristors and resistive RAM (RRAM) make up an immense part of such exploratory research in SNN implementation as they are a natural bridge between SNN algorithms and accelerators [37, 39]. They have been widely employed as both synapses and as spiking neurons.

At the ionic level, memristive synapses have been integrated into systems that naturally implement the spike-timing-dependent-plasticity (STDP) update rule using higher-order device dynamics [81,98,131]. An alternative use of ion-driven dynamics is when implementing the memristor as a neuron, where nonlinear conductance evolution gives rise to abrupt switching that can be used to emit sudden voltage spikes. This approach is typically coupled with capacitive integration, and has been referred to as a 'neuristor' [21,46,97,117], and a 'Memristive Integrate-and-Fire' (MIF) neuron [69,83,173]. Similarly, membrane leakage in biological neurons can be implemented using resistive dissipation as in neuristors, or via volatile ionic drifting dynamics observed in single devices [178] and also in nanowire networks [75,99,175].

Moving up to the architectural level, RRAM has been shown as a promising candidate for Compute-In-Memory (CIM) architectures due to their ability to parallelize matrix-vector multiplication independently of time complexity when integrated as large-scale, modular arrays [32, 53, 54, 96, 171]. Rather than neurons, memristive synapses map neural network weights to device conductances. In general, RRAM

CIM architectures are intended to be trained offline with weights mapped on-chip for inference and deployment. As such, RRAM synapses should be stationary and only used for weight read-out. Higher-order dynamical behaviors of memristors are abstracted away, and treated as non-idealities.

An additional challenge with RRAM-based CIM is the cost of communicating analog current signals along lengthy bit-lines and conversion into the digital domain. These issues have spurred the use of binary activations in the form of spike-based CIM accelerators which have been shown to alleviate the burdens of mixed-signal computation, by removing the need for large Analog-to-Digital Convertor (ADC) data conversion [56].

The majority of deep learning acceleration using memristors can be classified into one of the above cases: memristive neurons, memristive synapses that learn via associative learning, and CIM accelerators. A small set of designs have combined memristive neurons and memristive synapses together [142,158]. Doing so is a commendable feat, as the natural switching dynamics of memristive systems are relied on to achieve data-driven tasks. The cost of allowing hardware to behave naturally is that a designer can no longer rely on synchronous, clock-driven processing, and is subject to fault injections that are a result of nonlinear ionic dynamics. Letting the natural dynamics of memristive hardware 'teach itself' exacerbates the challenges of training MSNNs. This challenge has limited the demonstration of fully MSNNs to unsupervised learning tasks that have been shown to solve simple, low-dimensional pattern recognition via local learning rules (typically STDP) and associative learning. These include the classification of several characters and numbers (such as a

subset of the MNIST dataset).

We have shown that a fully MSNN based on the MIF2 model can emulate unsupervised learning (STDP) in Chapter 5. It would be very intriguing to think if the MIF2 model can also be used in supervised learning. And it is valuable to validate if the MIF/MIF2 model can be used in SNNs with a deep learning framework.

There is an incredibly broad span of work that integrates memristors with brain-inspired architectures, from low-level analog action potential emulation [83], to discrete spiking dynamics [56], up to non-spiking CIM processors [32]. We focus our background on prior work that uses nonlinear dynamics in memristive neurons together with memristive synapses, with an associated demonstration of synaptic optimization to achieve a data-driven outcome.

6.1.1 Fully MSNNs

Fully MSNNs refer to arrays that utilize nonlinear switching dynamics of memristors to trigger action potentials, and are coupled with memristive weights that are used as neural network parameters. The 8×8 crossbar array presented in [158] successfully integrates a fully MSNN, including memristive synapses and neurons. The synaptic array is trained using unsupervised STDP to classify four letters in a 24-pixel grid. While the task achieved is considerably simple, the fully memristive experimental demonstration sets the stage to build up new training methods.

The work in [85] uses half-wave rectification interposed between crossbar

arrays to process the ReLU activation in the analog domain. While not fully memristive, nor a 'spiking' network, it demonstrates a pivotal example of successively passing analog activations between RRAM crossbars without intermediate data conversion, in a manner akin to how our analog action potentials are transmitted between layers. This training process relies on gradient-based optimization where device non-idealities are injected during the forward pass. In doing so, a test set accuracy of 93.63% is obtained on the MNIST dataset.

In Refs. [150] and [162], convolutional SNNs with memristors are used, where both used pretrained non-spiking networks that are mapped or converted into the spiking domain. Both networks obtained competitive accuracy on the MNIST dataset, though did not demonstrate performance on more complex, real-world data. This may be due to the large differences between the networks that were trained and the MSNN that was implemented. In Ref. [51], a dense MSNN is adopted using a similar approach to what is used here, and as such, has minimal hardware requirements at run-time. The training process translates the switching dynamics of the memristive neuron into a firing rate, and may be the reason why a relatively low accuracy of 83.2% was achieved on the MNIST dataset. A more detailed comparison is illustrated in Section V, subsection B.

Almost all of these works offer compelling demonstrations using in-house fabricated arrays, either with standalone crossbars or back-end-of-the-line (BEOL) integrated arrays with foundry-made chips. In contrast, we have aimed to make our work as device-agnostic as presently possible by using commercially available Knowm memristors and their corresponding model [107].

Although Knowm memristors are not known for their reliability, their metastable switching dynamics are accounted for within the gradient calculation step, as it forms part of the computational graph. In contrast, Kiani *et al.* use the memristors in the forward-pass only, as their devices are not intended to be reprogrammed during inference, and thus their method does not require switching to generate spiking dynamics. Gradients can therefore be deterministically calculated partially off-chip [85], whereas our approach harnesses memristive dynamics in the forward-pass computation in our network. As such, our MSNN approach can leverage the benefits of spike-based processing, such as sparse processing and lower data collision rates.

6.1.2 Memristive Learning Frameworks

To ease and emulate the training process of memristive networks, a variety of valuable frameworks have been developed each addressing various niches. These include MemTorch [92], NeuroSim [35], and the IBM Analog Hardware Acceleration Kit [121], which implement non-spiking networks that adopt mixed-signal bit-line charge/current accumulation/summation processing. In these simulators, memristive dynamics are accounted for during weight updates, and are otherwise fixed during inference. To complement these tools, NeuroPack [77] specifically targets the simulation of spiking networks, where memristive dynamics are also factored in during the weight update process, and fixed during inference. Spiking dynamics are triggered by pulse-based input voltages.

The closest relation to our proposed work comes from Demirag *et al.* who

offer a software implementation of a real-time recurrent learning variant [27, 163] using phase change device models to train an MSNN [48].

Much like these simulators, our approach *MEMprop* integrates SPICE-level memristor models into the training process. But distinct from these simulators and training methods, we utilize arrays of MIF neuristors that trigger self-induced action potentials during the forward-pass computation. To emit spikes, the state of a memristor must evolve and ultimately switch to provide sudden discharge pathways which triggers spikes. Therefore, the dynamical characteristics of a memristor will alter the gradient itself, whereas all prior approaches account for dynamics in the weight update step when reprogramming memristive synapses.

Our approach applies the backpropagation algorithm directly to SPICE-level neuristor models that emit analog action potentials as a result of charge accumulation and metastable threshold switching. That is to say, the memristive dynamics in SPICE models are not fixed during the forward-pass, but rather, they are dynamically extracted during the forward-pass to fire neuristor-induced spikes. The BPTT algorithm is applied *directly* to SPICE models, thus integrating memristive neurons as part of the gradient calculation step, rather than isolating their dynamics to synaptic weight updates as in prior implementations. This offers a dynamical and totally new way to train fully memristive networks that uses state-based dynamics as part of the gradient calculation step.

In terms of hardware implementation, the conventional use of RRAM in circuits often requires significant overhead to convert analog currents into digital voltages and consumes a large amount of power [32]. In many instances, the power/area

demands of the ADCs and Digital-to-Analog Converter (DAC) far exceed the overhead brought on by RRAM, offsetting the advantages of memristors. In contrast, our spike-based approach eliminates the need for ADCs and DACs such that the cost of peripheral circuits is substantially reduced.

6.1.3 Error Backpropagation Through SPICE models

Our proposed approach enables us to scale up the complexity of learnable tasks in fully MSNNs by directly applying gradient descent to the nonlinear state evolution of memristive neurons and synapses. Both neurons and synapses in biological neural networks are modeled using memristors. The MIF neuron model is designed to achieve distinct depolarization, hyperpolarization, and repolarization voltage phases with a minimal set of circuit elements. Memristive synapses act as interconnects between layers of neurons.

To train the fully memristive network, we propose MEMprop, an application of error backpropagation to large-scale networks derived from SPICE circuit models. This enables dynamical, time-varying memristive neurons to learn and thus achieve much higher accuracy on data-driven tasks than has been previously reported with MSNNs. By relying on the analog spiking characteristics that naturally occur in the MIF neuron model, the non-differentiability of spike-based activations is completely avoided. This means MEMprop does not rely on surrogate gradient techniques that are commonly used to train SNNs, which calculates biased gradient estimators to circumvent the dead neuron problem [111]. To promote the broad accessibility of our methods, we use SPICE models of commercially available, low-cost

memristors and demonstrate the efficacy of MEMprop in a supervised deep learning framework.

6.2 Methods

6.2.1 Memristive Integrate-and-Fire Model

The neuron model adopted in our MSNN is the MIF neuron model depicted in Chapter 4 Figure 4.4a. For easy referral, we include the Figure 6.1 of the MIF circuit in this chapter as well. Qualitatively, given a positive current injection $I(t)$, the membrane potential $v(t)$ will rise up from the neuron's resting potential E_{rest}. Once a sufficiently large electric field builds up across the memristor M_2, it switches on, effectively shorting the output to E_{reset} which charges back up to equilibrium, E_{rest}.

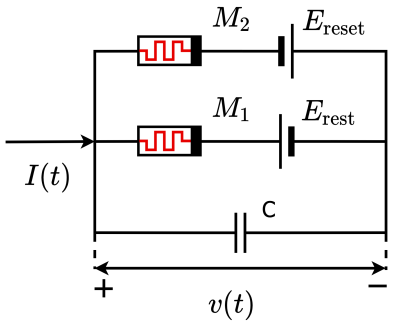


Figure 6.1: (a) A MIF neuron consists of two memristors M_1 and M_2, connected to DC voltage sources E_{rest} and E_{reset}, in parallel with a capacitor C. The MIF neuron is provably minimal in generating a membrane potential traversing from a rest voltage level to a threshold voltage level, then to a reset voltage level, and then back to the rest potential when a current pulse is applied.

The memristor model used is based on the generalized metastable switch

(MSS) model [107], which has been used to accurately describe a large range of possible devices. In this model, an MSS is an idealized element that switches with a given probability between two states as a function of its voltage and temperature. A memristor is modeled by a collection of MSSs, which determines the state-time dynamics that lead to non-volatile characteristics. x characterizes the internal state as a variable normalized between 0 and 1, as determined by the switching states of all MSSs.

Formally, the governing dynamics of the MIF neuron are characterized by the system of differential equations in Table 6.1. The membrane potential v is dependent on G_1 and G_2, the device conductances of $M1$ and $M2$, the MIF capacitance C, and E_{rest} and E_{reset}, which are voltage biases in the MIF circuit.

G_1 and G_2 are dependent on x_1 and x_2, a pair of variables governing the internal state of each device, where R_{on} and R_{off} are the on/off resistances.

The rates of change of x_1 and x_2 are dependent on the state evolution time constants τ_1 and τ_2, the thermal voltage V_{th} at room temperature, and the volatility constant $k_{\text{v}} \in [0, 1]$ in the range of 0 to 1. As $k \to 0$, the more retentive the device is.

6.2.2 Neural Network Layout

When accounting for the hardware implementation of a fully MSNN, the voltage response of the MIF neuron must drive the memristive synapses, and is correspondingly weighted by the synaptic conductances. This can be fully integrated into a crossbar according to Equation (6.1) below:

$$\mathbf{I} = \mathbf{G} \times \mathbf{V} \tag{6.1}$$

$$\begin{bmatrix} I_0 \\ I_1 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} w_{00} & w_{01} & \cdots & w_{0m} \\ w_{10} & w_{11} & \cdots & w_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n0} & w_{n1} & \cdots & w_{nm} \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_m \end{bmatrix}$$

$$I_n = \sum_{i=0}^{m} v_i \times w_{ni} \tag{6.2}$$

where $\mathbf{I}$ is the output current vector, $\mathbf{V}$ is the input voltage vector, and $\mathbf{G}$ is the conductance matrix of the crossbar. The output current vector is generated via bit-line current summation as shown in Equation (6.2), and directly drives the input of the next MIF neuron layer. Resistive loading may attenuate the output current in subsequent stages, but this can be accounted for using a scaling factor, or otherwise buffered [83, 155, 156].

Figure 6.2 shows a small-scale schematic of a fully MSNN with five MIF neurons and a 3×2 memristive crossbar, which receives five fan-in input currents I_{in0} - I_{in2} and generates three fan-out output currents. E_{rest} and E_{reset} are the voltage sources in the MIF circuit. C_{BL} is the bit-line parasitic capacitance generated by the metal line, or otherwise by the drain-bulk capacitance of select transistors, and is used as the membrane capacitance in the MIF circuit.

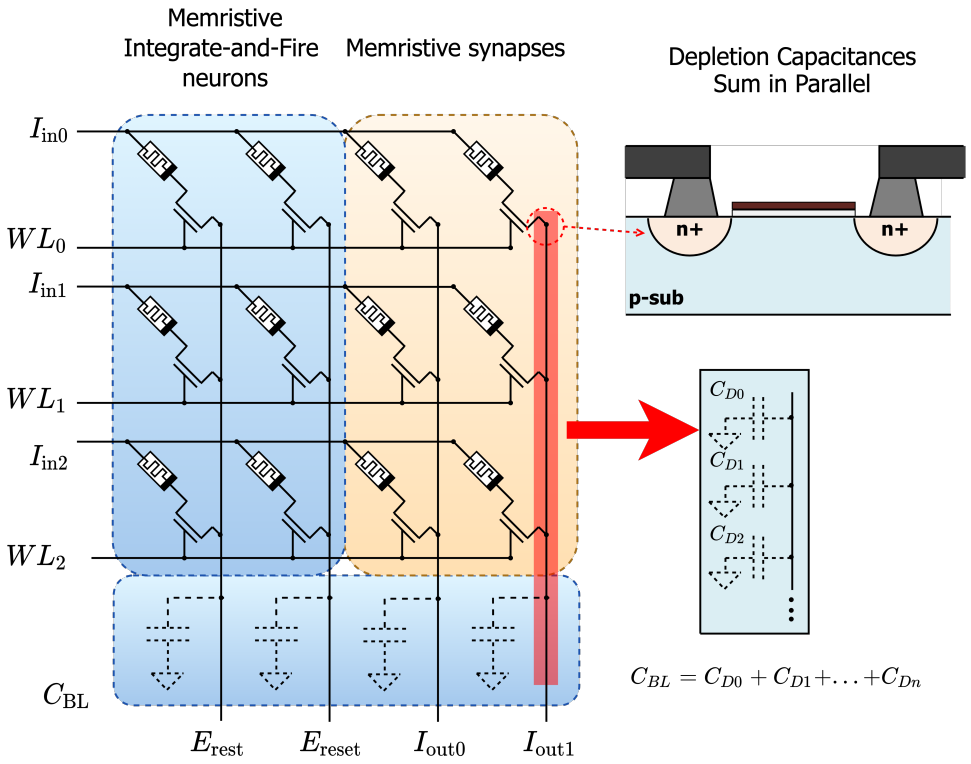


Figure 6.2: Schematic of a fully MSNN. The blue-shaded segment depicts memristive neurons, and the orange-shaded segment includes memristive synapses

6.2.3 Forward-mode Solution of MIF Model

In order to train a network of MIF neurons and synapses using gradient descent, the differential equations representing the MIF circuit dynamics (middle column, Table 6.1) are recast into discrete-time form (left column, Table 6.1). In doing so, the memristive dynamics can be captured in a computational graph that evolves over time, much like a recurrent neural network (RNN).

In practice, SPICE-kind simulators use a variety of differential equation solvers, such as the backward Euler method, and the 4th-order Runge-Kutta method (RK4). For compatibility with the BPTT algorithm, we solve the differential equations using the forward Euler method, which provides an explicit representation of the next time step using present-time dynamics. The rich dynamics of the MIF neuronal network are now accounted for in the MSNN, unrolled in time such that gradient descent can be used to optimize the memristive synapses as a function of the MIF evolution.

Many neural coding studies represent spike trains as a summation of time-shifted Dirac delta pulses $\sum_n \delta(t - t_j^n)$. As such a model of spikes is an idealization. We use the spike train to modulate a time-continuous, alpha input current I modeled by the equation in Table 6.2 to be compatible with real, physical systems. In this series of equations, a is an internal state variable, τ_{syn} is a time constant that determines the shape of the alpha current, W_i is the synaptic weight between a presynaptic neuron i and its associated postsynaptic neuron. It is commonly regarded that such alpha waveforms correspond to the response from biological neurons in the sensory periphery that respond with graded potentials [52].

Table 6.1: MIF model differential equations vs numerical integration.

Variable	Continuous Time Derivative	Discrete Time Solution
v	$\frac{\mathrm{d}v}{\mathrm{d}t} = \frac{I - G_1(v - E_{\mathrm{rest}}) - G_2(v - E_{\mathrm{reset}})}{C}$	$v^{t+1} = \frac{I_t - G_1^t(v^t - E_{\mathrm{rest}}) - G_2^t(v^t - E_{\mathrm{reset}})}{C} + v^t$
G_1	$G_1 = \frac{x_1}{R_{\mathrm{on1}}} + \frac{1-x_1}{R_{\mathrm{off1}}}$	$G_1^{t+1} = \frac{x_1^{t+1}}{R_{\mathrm{on1}}} + \frac{1-x_1^{t+1}}{R_{\mathrm{off1}}}$
G_2	$G_2 = \frac{x_1}{R_{\mathrm{on2}}} + \frac{1-x_2}{R_{\mathrm{off2}}}$	$G_2^{t+1} = \frac{x_2^{t+1}}{R_{\mathrm{on2}}} + \frac{1-x_2^{t+1}}{R_{\mathrm{off2}}}$
x_1	$\frac{\mathrm{d}x_1}{\mathrm{d}t} = \frac{1}{\tau_1}\left(\frac{1-x_1}{1+\exp(\frac{v_{\mathrm{on1}} - (v - E_{\mathrm{rest}})}{V_{\mathrm{th}} \cdot k_{\mathrm{v}}})} - \frac{x_1}{1+\exp(\frac{(v - E_{\mathrm{rest}}) - v_{\mathrm{off1}}}{V_{\mathrm{th}} \cdot k_{\mathrm{v}}})}\right)$	$x_1^{t+1} = \frac{1}{\tau_1}\left(\frac{1-x_1^t}{1+\exp(\frac{v_{\mathrm{on1}} - (v^t - E_{\mathrm{rest}})}{V_{\mathrm{th}} \cdot k_{\mathrm{v}}})} - \frac{x_1^t}{1+\exp(\frac{(v^t - E_{\mathrm{rest}}) - v_{\mathrm{off1}}}{V_{\mathrm{th}} \cdot k_{\mathrm{v}}})}\right) + x_1^t$
x_2	$\frac{\mathrm{d}x_2}{\mathrm{d}t} = \frac{1}{\tau_2}\left(\frac{1-x_2}{1+\exp(\frac{v_{\mathrm{on2}} - (v - E_{\mathrm{reset}})}{V_{\mathrm{th}} \cdot k_{\mathrm{v}}})} - \frac{x_2}{1+\exp(\frac{(v - E_{\mathrm{reset}}) - v_{\mathrm{off2}}}{V_{\mathrm{th}} \cdot k_{\mathrm{v}}})}\right)$	$x_2^{t+1} = \frac{1}{\tau_2}\left(\frac{1-x_2^t}{1+\exp(\frac{v_{\mathrm{on2}} - (v^t - E_{\mathrm{reset}})}{V_{\mathrm{th}} \cdot k_{\mathrm{v}}})} - \frac{x_2^t}{1+\exp(\frac{(v^t - E_{\mathrm{reset}}) - v_{\mathrm{off2}}}{V_{\mathrm{th}} \cdot k_{\mathrm{v}}})}\right) + x_2^t$

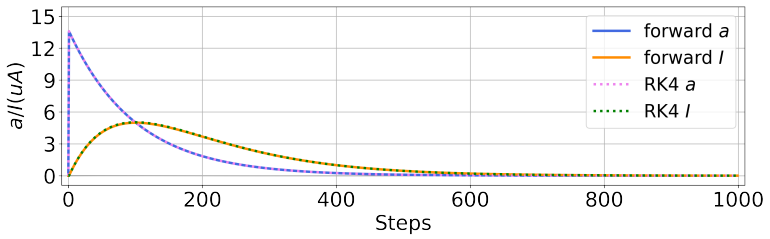
Table 6.2: Alpha current differential equations vs numerical integration.

Variable	Continuous Time Derivative	Discrete Time Solution
I	$\tau_{\text{syn}} \frac{\mathrm{d}I}{\mathrm{d}t} = a - I$	$I^{t+1} = \frac{a^t - I^t}{\tau_{\text{syn}}} + I^t$
a	$\tau_{\text{syn}} \frac{\mathrm{d}a}{\mathrm{d}t} = -a + W_i \cdot \sum_n \delta(t - t_i^n)$	$a^{t+1} = -\frac{a^t + W_i \cdot \sum_n \delta(t - t_i^n)}{\tau_{\text{syn}}} + a^t$

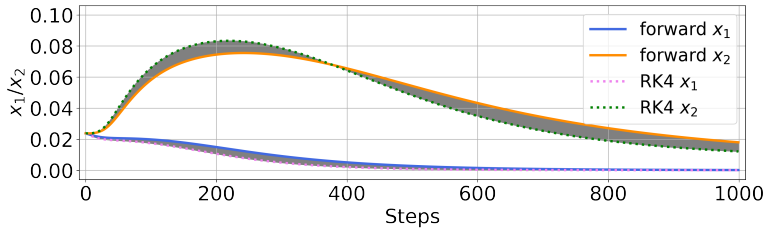
6.2.4 MIF Single Neuron Simulation

To verify the accuracy of the forward Euler method, we conduct a single neuron simulation using Python shown in Figure 6.3 across 1,000 time steps, which provides ample time for the membrane potential v to traverse from spiking V_{th}, to the reset potential E_{reset}, back to the resting potential E_{rest}. Each time step is of duration 0.01 ms over a total duration of 10 ms.

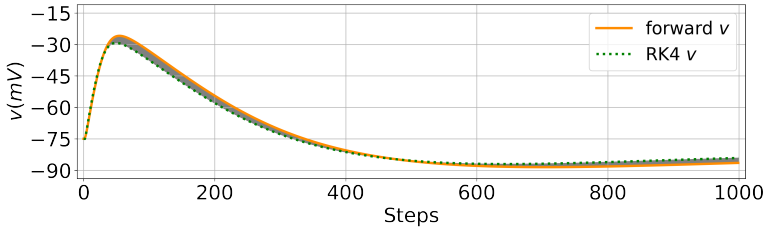
The parameters used in the simulation are listed in Table 6.3. The resting potential, charge integration, thresholding, and reset dynamics are closely reproduced in a SPICE simulator that uses the RK4 solver, which verifies the solution generated by adopting the forward Euler method. Deep learning is known to be tolerant to fault injections [91,116], and so the tradeoff between the numerical integration accuracy and the training complexity should be considered. We adopted the forward Euler method which is able to balance the two. Technically, our training method is expected to generalize to other numerical integration methods as well, but with added computational complexity.



(a)



(b)



(c)

Figure 6.3: Simulation results of MIF neuron solved using forward Euler numerical integration. The results match the quantitative dynamics solved by a SPICE simulator. The difference between the two methods is marked by grey area. (a) Alpha synaptic current dynamics. (b) Internal states x_1, x_2. (c) Voltage response v.

Table 6.3: MIF circuit parameters

Parameter	**Value**	**Parameter**	**Value**
v_{off_1}, v_{off_2}	5 mV	v_{on_1}, v_{on_2}	110 mV
R_{off_1}, R_{off_2}	0.1 MΩ	R_{on_1}, R_{on_2}	1 kΩ
τ_1, τ_2	1 ms	τ_{syn}	0.64 ms
E_{rest}	0 mV	E_{reset}	50 mV
V_{th}	25 mV	k_{v}	0.6
C	100 pF		

6.2.5 BPTT in MSNNs

The discrete-time solution in Table 6.1 is illustrated as a directed, acyclic graph in Figure 6.4, where time flows from left to right. The MIF circuit parameters are color-coded to show how each electrical characteristic impacts the others at the next time step.

To train a network, a loss function is calculated using the membrane potential v of the output layer at each step. Note that the adjoint method in [36] is not adopted here, as an intermediate state is required to calculate loss and guide training for each time step. We are concerned with the spiking output at all time steps, and not just the final state of the system which makes BPTT a more optimal choice. In our example network shown in Figure 6.4 with 10 output neurons, the predicted MIF neuron is expected to spike most frequently by aiming to increase the membrane potential across time steps, while the incorrect target should be sup-

pressed. As the membrane dynamics are continuous, a fully analog MSNN can be trained without surrogate gradients, as has become ubiquitous in deep SNNs trained via error backpropagation [57, 111].

The BPTT algorithm iteratively applies the chain rule from the output back to the leaf nodes, w, to determine the update direction to optimize the network. Prior demonstrations treat w as the device conductance. In our case, we also do this, but additionally generate spiking behaviors using naturally occurring MIF dynamics rather than applying a hard threshold to the membrane potential. Not only is this a more biorealistic representation, but it can be fully integrated using RRAM crossbars.

6.2.6 Method Evaluation

To validate the idea of using MIF neurons in a deep learning framework, we used a 4-layer neural network. In the input layer, there are $28 \times 28 = 784$ input units. The first hidden layer consists of 100 neurons followed by a ReLU activation, and the second hidden layer contains 10 neurons followed by a ReLU activation. Finally, the output layer consists of 10 MIF neurons. Fully connected (all-to-all) weights are used between each layer.

The forward Euler solution of each MIF neuron was defined in PyTorch v1.10.1 in Python 3.8, where the autodifferentiation framework was used to keep track of gradients of all forward-mode computations on-the-fly. For training and testing, we used the MNIST dataset [94] which consists of 70,000 samples of handwritten digits. During the training process, 1,000 time steps are simulated for each

input image sample. Over the 1,000 steps, the input pixel intensity is fed to the network only at steps 0, 400, and 800 to promote sparse network activity. The negative log-likelihood loss is applied to the membrane potential, i.e., $v(t)$ in the MIF circuit in Figure 6.1, at each time step. In other words, the goal of the network is to maximize the voltage across the MSNN circuit of the correct class. The total loss is summed prior to backpropagating the error through the SPICE memristor model. The Adam optimizer is utilized as it performs well on both recurrent networks and stochastic problems [87]. For the Adam optimizer, the learning rate is set to 10^{-4}, and the initial decay rates for first and second order moments are set to $\beta_1 = 0.9$, $\beta_2 = 0.999$. The learnable weights of dense layers are initialized by sampling from a uniform distribution $U(-\sqrt{k}, \sqrt{k})$, where $k = \frac{1}{N_i}$ and N_i is the number of input features to layer i. The MSNN is trained for 40 epochs using a batch size of 200 samples. Accuracy is evaluated at every 10 training iterations. For the MNIST dataset, the training and testing accuracies across multiple epochs are shown in Figure 6.5. 97.58% accuracy is achieved for the total MNIST test set. For a more challenging task using real-world data, we also train our MSNN on the Fashion-MNIST dataset, where we obtain 75.26% testing accuracy. The Fashion-MNIST dataset is rarely demonstrated using analog MSNNs, which highlights the potential for MSNNs to be capable of moving beyond simple MNIST classification.

We have validated this idea by training a neural network with the last layer as MIF neurons. It is intriguing to show if we can train a neural network in which hidden layers consist of MIF neurons as well.

An overview of this approach is shown in Figure 6.4.

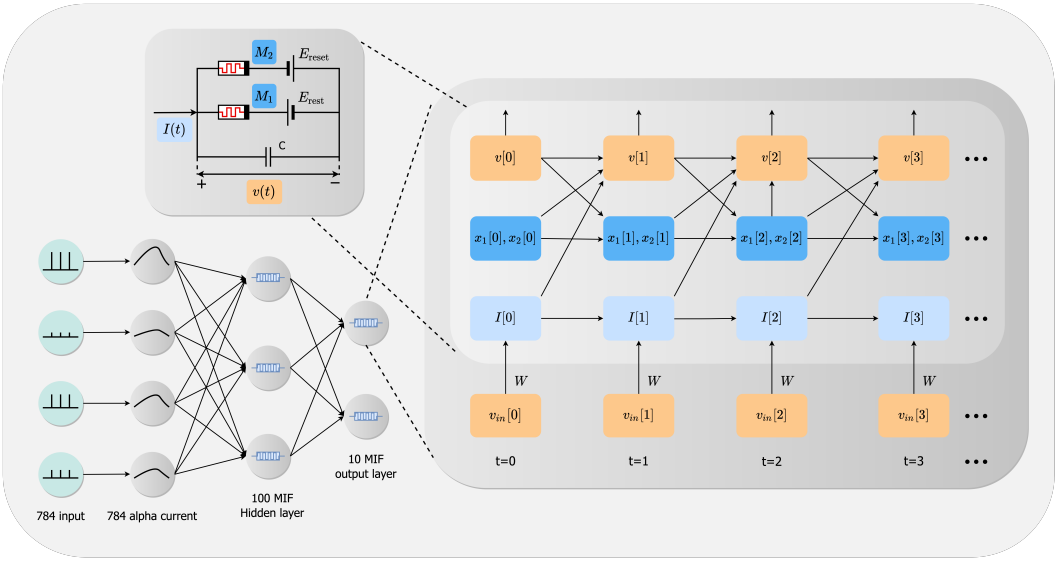
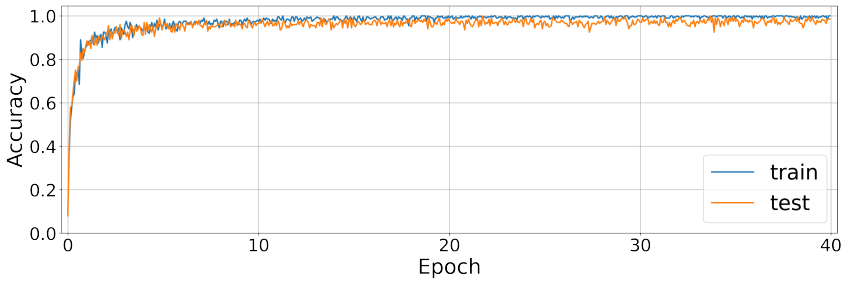


Figure 6.4: An overview of our MEMprop approach. The MSNN architecture and the resulting computational graph consist of memristive dynamics. In this approach, both neurons and synapses in biological neural networks are modeled using memristors. We emulate neurons using a MIF circuit that utilizes SPICE models of commercially available, low-cost memristors. Memristive synapses act as interconnects between layers of neurons. Errors are backpropagated directly through SPICE circuit models of memristive neurons such that higher-order device dynamics are fully utilized in the learning process. Memristive dynamics are broken down into a series of composable, differentiable functions and used during gradient descent. We used a lightweight 3-layer dense SNN with 100 hidden MIF neurons and benchmarked it on several datasets.

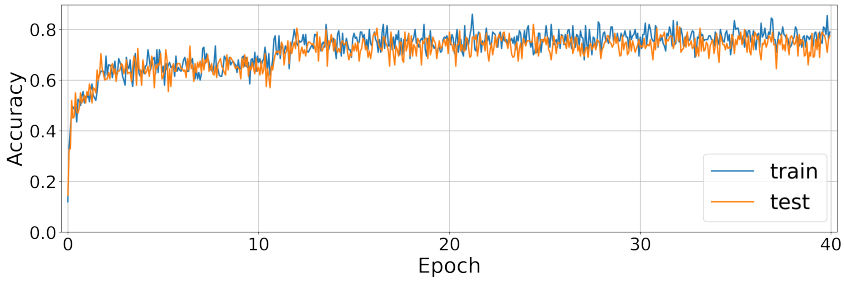
6.3 Experimental Results

6.3.1 Network Architecture

To validate the use of MIF neurons in both hidden layers and output layers of a deep learning framework, we used a lightweight 3-layer dense SNN with



(a)



(b)

Figure 6.5: Accuracy across epochs for training and testing processes for (a) MNIST dataset (b) Fashion-MNIST dataset.

100 hidden MIF neurons (bottom-left of Figure 6.4). Each MSNN is simulated for a duration of 10 ms over a span of 1,000 discrete time steps.

The input layer consists of a number of input features (784 for the static datasets; 2,048 for a downsampled neuromorphic dataset). An alpha current generator with an amplitude weighted by the input pixel intensity (Table 6.2), weighted by memristive synapses, inducing 100 MIF neurons to fire into another set of memristive synapses, terminated by the output layer of MIF neurons. To account for circuit loading effects, the current injected into each MIF layer is attenuated by a scaling factor defined in the hyperparameters.

6.3.2 Datasets

Three datasets of increasing difficulty are used to assess the MSNN: MNIST, FashionMNIST, and the DVS128 Gesture datasets. Despite being considered a 'solved' problem for quite some time now, it was often the case that the MNIST dataset was the most challenging problem that could be solved by previously reported MSNNs. In most cases, a subset or simplified alternative would be used to demonstrate pattern recognition. Here, we demonstrate for the first time that RRAM arrays that rely on internal dynamics are still capable of processing more challenging problems and real-world datasets, namely, FashionMNIST and neuromorphic data.

The MNIST [94] and the slightly more challenging FashionMNIST datasets [165] are used to test the performance of the MSNN on temporally static data. Both datasets consist of 60,000 28×28 greyscale images in the training set, and 10,000

images in the test set, with 10 output classes of handwritten digits (MNIST) and clothing items/accessories (FashionMNIST/FMNIST). During the training process, the alpha current inject is applied at every 100 steps to promote sparse network activity.

For a more challenging test case, the DVS128 Gesture dataset [13] is used as a neuromorphic baseline to test the MSNN's performance on event-driven data filmed with an event-based camera [114]. Each sample only processes sufficient changes in luminance, and consists of 11 different output classes of hand gestures, such as clapping, arm rotation, and air guitar. Each sample is downsampled to a resolution of $32 \times 32 \times 2$, where the channel depth of 2 accounts for on and off spikes (positive and negative luminescence changes). The training data is integrated to fit within 100 discrete time steps, and the testing data is integrated within 360 time steps, to account for GPU memory constraints.

6.3.3 Training Process

The loss used is the negative log-likelihood of the membrane potential $v(t)$ (Figure 6.4) of the output layer. The network objective is to increase the voltage across the MIF neuron associated with the correct class. The total loss across time steps is summed prior to backpropagating the error through the SPICE memristor model using the Adam optimizer [87]. The MSNNs are each trained for 50 epochs.

6.3.4 Results

The final test set accuracy is measured across 50 epochs over 5 trials for each dataset with early stopping applied, and the average result is shown in Figure 6.6 with error bars.

The average accuracy on the MNIST dataset reaches 93.08%, which is very high among fully MSNN considering that almost no required peripheral circuitry for hardware implementation and far exceeds other MSNN baselines that have been previously reported. Although it remains somewhat below non-memristive baselines which can obtain greater than 97% on the MNIST dataset using a similar architecture, such models are typically computed using fixed- and full-precision arithmetic rather than depending on naturally arising memristive dynamics.

Where our approach tends to shine is on real-world data that goes beyond the simplicity of handwritten digit recognition. Our fully MSNN performance holds for the FashionMNIST dataset, where we obtained an average of 84.77%, and also for the DVS128 Gesture dataset, with 82.63%. This provides the first successful demonstration of training an MSNN on a neuromorphic dataset by relying on naturally occurring device dynamics for spike emission.

6.4 Discussion and Conclusion

6.4.1 Area, Power, and Latency

With our fully MSNN approach, one of the main advantages is that we avoid the need for ADCs, and subsequently, we do not need to multiplex or serialize

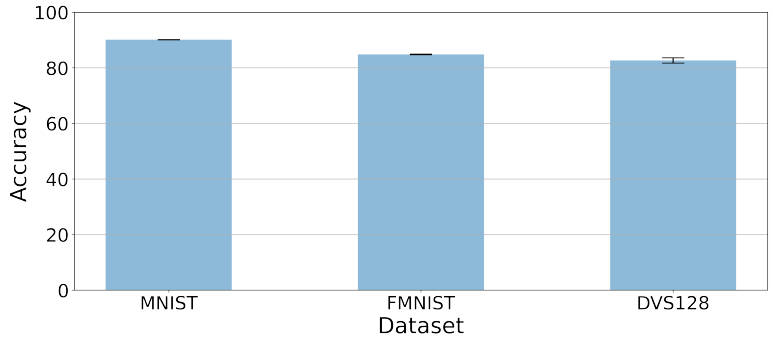


Figure 6.6: Test set accuracies with error bars over 5 trials. The mean accuracy and standard deviation for each dataset are (1) MNIST: $\bar{x}$ =93.08, σ =0.07 (2) FMNIST: $\bar{x}$ =84.77, σ =0.13 (3) DVS128: $\bar{x}$ =82.63, σ =0.95.

data. To quantify the potential improvement in terms of power, area, and latency, we assume a dense 3-layer architecture of 784-100-10 neurons. Such a network requires 110 MIF neurons and 79.4k memristive synapses. For each synaptic weight, a pair of devices are required to implement current subtraction to represent both positive and negative weights. Given the array structure in Figure 6.2, a total of 0.16M RRAM cells are required, which needs 10 RRAM tiles of 128×128 in size.

To provide an area estimation at a 65-nm technology node, the size of each RRAM tile with 1T1R cells is 2.77×10^{-3} mm^2, and a complete array will occupy 2.77×10^{-2} mm^2 [151]. If a bit-line current summation approach was adopted for a dense ANN with 8-bit networks, a current-mode SAR ADC each occupies 3×10^{-3} mm^2, where 4 ADCs are shared across the column wires for each crossbar. In such a case, the improvement of our approach without converting and serializing activations by adopting a spike-based approach is a factor of 5.33×.

Estimating the power consumption of asynchronous, spike-based workloads requires profiling the spiking activity in the network for a given dataset. Using the

MNIST dataset, we found that an average of 2% of neurons are spiking at any given time in the network. This may be modulated to promote varying degrees of excitation in the network, for example, by applying a spike-dependent regularization term to the objective function. The average resistance in our network is close to the midpoint between the on/off resistances: $R_{\text{off}} = 100$ kΩ, $R_{\text{on}} = 1000$ Ω, average $R_{\text{ave}} = 50.5$ kΩ. The voltage across each device when emitting a spike follows a sharp peak of $v_{\text{on}} = 110$ mV, a refractory period where the potential drops down to $v_{\text{off}} = 5$ mV, and the average value assuming the spike is linearized for the window of interest is $v_{\text{ave}} = 57.5$ mV can give an estimate of the average power

$$P_{\text{ave}} = \frac{v_{\text{ave}}^2}{R_{\text{ave}}} \times N_{\text{cells}} \times 2\%, \tag{6.3}$$

which is then numerically evaluated for a single 128×128 tile to be 21.45 μW, and 0.21 mW when accounting for all 10 tiles. When accounting for data conversion overhead where each 8-bit ADC consumes 2×10^{-4} W, the total power consumption increases to 8.21 mW. Our approach can offer an improvement of 38.30×.

We estimate latency by driving ADCs at an operating frequency of 40MHz. With 4 ADCs converting 64 column currents (assuming 2 bit-lines per activation), 32 data-lines are grouped to share a driving DAC which generates an 8-bit serialized input operating across 8 distinct cycles, evaluating to input data latency of 6.4μs. At the output, 32 bit-lines each share an ADC where 16 column currents must be converted, increasing the latency by 0.4μs. Further assuming that data conversion dominates latency, this means that it takes 6.8μs per vector-matrix multiplication (VMM). A 1,000 time step simulation, as used in our network, will extend this to a

latency of 6.8 ms. Additionally, each input is fed as an alpha current signal with a measurable latency of around 0.64 ms, giving the total latency for the conventional approach of 7.44 ms, as opposed to MEMprop which only requires the additional switching time of the memristors (on the order of 100s of nanoseconds added to the alpha current latency). In our approach, currents and voltages are generated without the need for conversion or serialization, so there is no added ADC latency. The alpha current injection and response already account for RC delay through the memory array.

A summary of the above-mentioned comparison is shown in Table 6.4.

Table 6.4: Power, area, and latency improvement in our MSNN

Aspect	**Our method**	**Mixed-Signal**	**Improvement**
Area	2.77 mm^2	14.77 mm^2	5.33×
Power	0.21 mW	8.21 mW	38.30×
Latency	0.64 ms	7.44 ms	11.63x

6.4.2 Comparison

A comparative analysis against other memristive networks is provided in Table 6.5. Where metrics are not provided in the original sources (e.g., chip area, power consumption, and inference latency), we have made an estimate where sufficient data has been provided to extrapolate these values. The closest MSNNs to our work are briefly described below.

In Ref. [150], a convolutional neural network with analog neurons and digital spiking at the input layer is adopted. Each analog neuron consists of four transmission gates, seven operational amplifiers, a comparator, and a memristor. While this is substantially more complex than the MIF neuron, the large overhead per neuron is offset by using time-division multiplexing access (TDMA) to treat a single physical neuron as multiple algorithmic neurons.

Wijesinghe *et al.* also adopt a convolutional SNN using stochastic switching to implement probabilistic firing, which is a natural fit for devices that switch based on random processes [162]. A non-spiking ANN is first trained and subsequently converted to an SNN. This approach is known to perform optimally for static datasets on reasonably deep SNNs, but it sets an upper limit of performance such that the SNN accuracy will not surpass the accuracy of the ANN. Furthermore, ANN to SNN conversion is yet to show the success on neuromorphic datasets [57]. The total area of the design is reported to be 3 mm^2, and the latency for a single spike is 0.21 μs.

In Ref. [51], a 3-element neuron is constructed, consisting of a resistor, a memristor, and a capacitor, which is closest in spirit to our own approach here. The neuron is parameterized to fit a LIF neuron model, which relies on hard-thresholded spike generation which is a non-differentiable function. The problem is circumvented by utilizing surrogate gradient descent, and results in a test set accuracy of 83.2%. This comparatively lower accuracy highlights the challenges of adopting neuristor-like dynamics, and the difficulty of mapping conventional training methods, both supervised (surrogate gradient descent) and unsupervised (STDP), to dynamical

Table 6.5: Comparison among fully MSNNs

Method	Architecture	Neuron No. of elements	Refrac-tory	Neuron feature	Training Method	MNIST	Complexity of Additional Control Logic
[150]	1Mem-C-M-C-M-1024F-10F	4(5)+33T	✓	Mixed-signal	LIF rate fit TDMA	97.1%	high
[162]	784-C-M-C-M-10F	5+2T	✓	Digital	Stochastic switch fit ANN converts SNN	~96%	high
[51]	784-100F-10F	3	✓	Analog	LIF shape fit surrogate gradient	83.2%	low
Our work	784-100F-10F	3(5)	✗	Analog	MIF direct train MEMprop	93.2%	low

T: transistor. [2] Mem: memristor. [3] C: convolutional layer. [4] M: max pooling layer. [5] F: fully connected layer.

Table 6.6: Test set accuracies for the MNIST, FashionMNIST, and DVS128 Gesture Datasets.

Dataset	**Batch**	**LR**	**Best**	**Avg.** $(n = 5)$	σ
MNIST	128	1e-4	**93.18**	**93.08**	0.07
FMNIST	128	1e-4	**84.79**	**84.77**	0.13
DVS-128	16	1e-4	**83.21**	**82.63**	0.95

LR: learning rate η. [2] σ: standard deviation.

RRAM arrays. We expect the lack of a direct correspondence between thresholded LIF neurons and memristive neuron spiking is what resulted in the accuracy degradation, though it sets the stage for developing new training methods that account for dynamical, continuous-time switching into the computational graph of the network. In absence of power consumption metrics, we apply the same assumptions as in our MSNN architecture, namely, that 2% of the network is active at any given time, and estimate the total power consumption is 0.63 W.

No surrogate gradients or associated approximations are required due to the continuous-time nature of the spiking behavior in our experiments. The absence of ADCs/DACs in conventional full-precision/fixed-precision bit-line current summation approaches reduces the computational overhead. This is due to the use of *MEMprop*. More details of our experiments are listed in Table 6.6.

6.4.3 Concluding Remarks

To the best of our knowledge, this is the first work that has mapped low-level, analog SPICE dynamics directly into an acyclic computational graph that is compatible with the backpropagation algorithm. Directly mapping the behavior of dynamical RRAM arrays into an autodifferentiation framework has enabled us to achieve promising performance on real-world datasets on fully MSNNs, beyond simple pattern recognition and handwritten digit classification. By harnessing a blend of the intrinsic behaviors of memristors and the sparse network activity of SNNs, the power consumption and latency of our approach surpass those of all other similar methods without compromising on accuracy.

6.5 Chapter Summary

In this chapter, we push beyond the classical limits of fully MSNNs, and demonstrate competitive performance on real-world datasets that go beyond static pattern recognition using an MSNN that includes both memristive synapses and memristive neurons. We achieve this by combining higher-order memristive dynamics with the gradient optimization process. While fault injections during weight updates have been accounted for in the past, this is the first time nonlinear memristive dynamics are included within the computational graph used in the gradient calculation process, enabling more precise fine-tuning of network parameters.

The action potentials generated by memristive neurons are a result of nonlinear ion-driven dynamics, which are functionally equivalent to a chain of nonlinear

operations in a computational graph. As each spike is a result of naturally arising physics-driven phenomena rather than discrete switching in digitally-composed SNNs, all functions are fully differentiable and the gradient is well-defined. Therefore, there is no need to resort to surrogate gradient approaches that have become ubiquitous when training SNNs. The voltage action potential is scaled by memristive synapses, and the resultant current drives downstream layers of memristive neurons.

A summary of our contributions is provided below:

- We present a fully MSNN utilizing both memristive neurons and synapses, and is shown to achieve state-of-the-art accuracy for lightweight dense network architectures on real-world datasets. These datasets are more complex than what has been demonstrated in the past on MSNNs while using naturally arising physics-driven phenomena in RRAM;
- We propose *MEMprop*, the application of error backpropagation directly to SPICE circuit models of memristive neurons such that higher-order device dynamics are fully utilized in the learning process. Memristive dynamics are broken down into a series of composable, differentiable functions and used during gradient descent;
- MSNNs are shown to be trainable without the need for gradient approximations around hard thresholds, such as with surrogate gradient descent, and
- The need for ADCs is amortized by relying entirely upon analog current injection at the input, and analog action potentials at the output.

In doing so, the rich dynamics of nonlinear devices can be fully leveraged in larger systems, in a way that moves far beyond applying fault injection and variation into the training process. We achieve state-of-the-art accuracy for lightweight dense network architectures on several static and neuromorphic datasets, which pushes the fringe of what previous MSNNs have shown to be capable of.

6.6 Conclusions

This dissertation investigates Memristive Spiking Neural Network (MSNN) -based architectures and algorithms, including memristive neurons, memristive synapses, MSNNs that can process auditory signals, fully MSNNs that can learn with Spiking-Time-Dependent Plasticity (STDP) and Backpropagation Through Time (BPTT). A minimum circuit Memristive Integrate-and-Fire (MIF) model was developed in a SPICE-level simulator and ported to Python for large-scale neural network simulation.

In chapter one, neuromorphic computing, memristors, and SNNs are introduced as the background of our work.

In chapter two, we dive deeper to show more general concepts behind the memristor – i.e., memristive devices and systems.

In chapter three, we demonstrated how non-volatile memristors can be used to mimic synapses with Long-Term Plasticity (LTP), which often occurs in learning phenomena and volatile memristors can be used to mimic synapse with Short-Term Plasticity (STP). We utilized a volatile memristor to implement STP synapse and emulated a sound localization (SL) MSNN. Alpha synaptic current and

Leaky Integrate-and-Fire (LIF) neuron models are adopted to make our neuronal behavior as realistic as possible. Lateral inhibition, one of the basic functions of biological neurons, is used for clearer SL with improved energy efficiency. We evaluated the localization accuracy by feeding the different frequencies of sound signals to the MSNN, which achieved a 1-degree resolution.

In chapter four, the Hodgkin–Huxley (HH) model and LIF models are described, followed by the proposal of our MIF model. The MIF1 model is used for neuronal signaling with two voltage levels: the spike-peak, and the rest potential. The second model, MIF2, is also presented, which promotes local adaptation by accounting for a third refractory voltage level during hyperpolarization. We show both compact models are minimal in terms of the number of circuit elements and integration area. Using the MIF and MIF2 models, we postulate the design of a memristive solid-state brain with an estimation of its surface area and power consumption. Analytical projections show that a memristive solid-state brain could be realized within (i) the surface area of the median human brain, 2,400 cm2, (ii) the same volume of the median human brain, and (iii) a total power budget of approximately 20 W using a 3.5 nm technology. Distinct from the past decade of memristive neuron literature, our benchmarks are attained using generic commercially available memristors that are reproducible using off-the-shelf components. We expect this work can promote more experimental demonstrations of memristive circuits that do not rely on prohibitively expensive fabrication processes.

In chapter five, fully MSNNs consisting of MIF neurons and memristive STDP synapses are elaborated. We present a fully MSNN consisting of physically-

realizable memristive neurons and memristive synapses to implement an unsupervised STDP learning rule. The system is fully memristive in that both neuronal *and* synaptic dynamics can be realized by using memristors. The neuron is implemented using the SPICE-level MIF model, which consists of a minimal number of circuit elements necessary to achieve distinct depolarization, hyperpolarization, and repolarization voltage waveforms. The proposed MSNN uniquely implements STDP learning by using cumulative weight changes in memristive synapses from the voltage waveform changes across the synapses, which arise from the presynaptic and postsynaptic spiking voltage signals during the training process. Two types of MSNN architectures are investigated: 1) a biologically plausible memory retrieval system, and 2) a multi-class classification system. Our circuit simulation results verify the MSNN's unsupervised learning efficacy by replicating biological memory retrieval mechanisms, and achieving 97.5% accuracy in a 4-pattern recognition problem in a large-scale discriminative MSNN.

In chapter six, fully MSNNs consist of MIF neurons and memristive STDP synapses are illustrated. We present *MEMprop*, the adoption of gradient-based learning to train fully MSNNs. Our approach harnesses intrinsic device dynamics to trigger naturally arising voltage spikes. These spikes emitted by memristive dynamics are analog in nature, and thus fully differentiable, which eliminates the need for surrogate gradient methods that are prevalent in the Spiking Neural Network (SNN) literature. Memristive neural networks typically either integrate memristors as synapses that map offline-trained networks, or otherwise rely on associative learning mechanisms to train networks of memristive neurons. We instead apply the

BPTT training algorithm directly on analog SPICE models of memristive neurons *and* synapses. Our implementation is fully memristive, in that synaptic weights and spiking neurons are both integrated on resistive RAM (RRAM) arrays without the need for additional circuits to implement spiking dynamics, e.g., Analog-to-Digital Converters (ADCs) or thresholded comparators. As a result, higher-order electrophysical effects are fully exploited to use the state-driven dynamics of memristive neurons at run time. By moving towards non-approximate gradient-based learning, we obtain highly competitive accuracy among previously reported lightweight dense fully MSNNs on several benchmarks.

Bibliography

[1] eflash. https://www.tsmc.com/english/dedicatedFoundry/technology/specialty/eflash.

[2] Keras. https://https://keras.io.

[3] Mad200. https://mycmp.fr/nvm-mad200/.

[4] memristor. https://en.wikipedia.org/wiki/Memristor.

[5] Neuronal dynamics. https://www.epfl.ch/labs/lcn/~gerstner/NeuronalDynamics-MOOCall.html.

[6] norse. https://norse.github.io/norse.

[7] Pytorch. https://https://pytorch.org.

[8] rockpool. https://rockpool.ai.

[9] sinabs. https://sinabs.ai.

[10] snntoolbox. https://snntoolbox.readthedocs.io.

[11] snntorch. https://snntorch.readthedocs.io.

[12] Tensorflow. https://www.tensorflow.org.

[13] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7243–7252, 2017.

[14] Aayush Ankit, Izzat El Hajj, Sai Rahul Chalamalasetti, Geoffrey Ndu, Martin Foltin, R Stanley Williams, Paolo Faraboschi, Wen-mei W Hwu, John Paul Strachan, Kaushik Roy, et al. Puma: A programmable ultra-efficient memristor-based accelerator for machine learning inference. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 715–731, 2019.

[15] Haroon Anwar, Xinping Li, Dirk Bucher, and Farzan Nadim. Functional roles of short-term synaptic plasticity with an emphasis on inhibition. *Current opinion in neurobiology*, 43:71–78, 2017.

[16] Alon Ascoli, Ahmet Samil Demirkol, Ronald Tetzlaff, Stefan Slesazeck, Thomas Mikolajick, and Leon Chua. On local activity and edge of chaos in a namlab memristor. *Frontiers in Neuroscience*, 15:233, 2021.

[17] Alon Ascoli, Stefan Slesazeck, Hannes Mähne, Ronald Tetzlaff, and Thomas Mikolajick. Nonlinear dynamics of a locally-active memristor. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(4):1165–1174, 2015.

[18] Alon Ascoli, Ronald Tetzlaff, Sung-Mo Kang, and Leon O Chua. Theoretical foundations of memristor cellular nonlinear networks: A drm 2-based method

to design memcomputers with dynamic memristors. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(8):2753–2766, 2020.

[19] Mostafa Rahimi Azghadi, Corey Lammie, Jason K Eshraghian, Melika Payvand, Elisa Donati, Bernabe Linares-Barranco, and Giacomo Indiveri. Hardware implementation of deep network accelerators towards healthcare and biomedical applications. *IEEE Transactions on Biomedical Circuits and Systems*, 14(6):1138–1159, 2020.

[20] R J Baker. Resistive memory element sensing using averaging.

[21] Lin Bao, Jiadi Zhu, Zhizhen Yu, Rundong Jia, Qifeng Cai, Zongwei Wang, Liying Xu, Yanqing Wu, Yuchao Yang, Yimao Cai, et al. Dual-gated mos2 neuristor for neuromorphic computing. *ACS Applied Materials & Interfaces*, 11(44):41482–41489, 2019.

[22] Felix Christian Bauer, Dylan Richard Muir, and Giacomo Indiveri. Real-time ultra-low power ecg anomaly detection using an event-driven neuromorphic processor. *IEEE Transactions on Biomedical Circuits and Systems*, 13(6):1575–1582, 2019.

[23] Bruce P Bean. The action potential in mammalian central neurons. *Nature Reviews Neuroscience*, 8(6):451–465, 2007.

[24] Mark Bear, Barry Connors, and Michael A Paradiso. *Neuroscience: Exploring the brain*. Jones & Bartlett Learning, LLC, 2020.

[25] Mark F Bear and Robert C Malenka. Synaptic plasticity: Ltp and ltd. *Current opinion in neurobiology*, 4(3):389–399, 1994.

[26] Trevor Bekolay, James Bergstra, Eric Hunsberger, Travis DeWolf, Terrence C Stewart, Daniel Rasmussen, Xuan Choo, Aaron Voelker, and Chris Eliasmith. Nengo: a python tool for building large-scale functional brain models. *Frontiers in neuroinformatics*, 7:48, 2014.

[27] Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature Communications*, 11(1):1–15, 2020.

[28] Ben Varkey Benjamin, Peiran Gao, Emmett McQuinn, Swadesh Choudhary, Anand R Chandrasekaran, Jean-Marie Bussat, Rodrigo Alvarez-Icaza, John V Arthur, Paul A Merolla, and Kwabena Boahen. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5):699–716, 2014.

[29] Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, 18(24):10464–10472, 1998.

[30] Jens Blauert. *Spatial hearing: the psychophysics of human sound localization.* MIT press, 1997.

[31] Geoffrey W Burr, Robert M Shelby, Abu Sebastian, Sangbum Kim, Seyoung

Kim, Severin Sidler, Kumar Virwani, Masatoshi Ishii, Pritish Narayanan, Alessandro Fumarola, et al. Neuromorphic computing using non-volatile memory. *Advances in Physics: X*, 2(1):89–124, 2017.

[32] Fuxi Cai, Justin M Correll, Seung Hwan Lee, Yong Lim, Vishishtha Bothra, Zhengya Zhang, Michael P Flynn, and Wei D Lu. A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations. *Nature Electronics*, 2(7):290–299, 2019.

[33] Luis A Camuñas-Mesa, Bernabé Linares-Barranco, and Teresa Serrano-Gotarredona. Neuromorphic spiking neural networks and their memristor-cmos hardware implementations. *Materials*, 12(17):2745, 2019.

[34] Gal Chechik, Isaac Meilijson, and Eytan Ruppin. Synaptic pruning in development: a computational account. *Neural computation*, 10(7):1759–1777, 1998.

[35] Pai-Yu Chen, Xiaochen Peng, and Shimeng Yu. Neurosim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(12):3067–3080, 2018.

[36] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[37] Leon Chua. Memristor-the missing circuit element. *IEEE Transactions on Circuit Theory*, 18(5):507–519, 1971.

[38] Leon Chua, Valery Sbitnev, and Hyongsuk Kim. Hodgkin–huxley axon is made of memristors. *International Journal of Bifurcation and Chaos*, 22(03):1230011, 2012.

[39] Leon O Chua and Sung Mo Kang. Memristive devices and systems. *Proceedings of the IEEE*, 64(2):209–223, 1976.

[40] Circuits Multi-Projects (CMP). Mad200 memory advanced demonstrator 200mm., 2018.

[41] Daniel L Cook, Peter C Schwindt, Lucinda A Grande, and William J Spain. Synaptic depression in the localization of sound. *Nature*, 421(6918):66–70, 2003.

[42] Taiwan Semiconductor Manufacturing Corporation. eflash, 2020.

[43] Erika Covi, Stefano Brivio, Alexander Serb, Themis Prodromakis, Marco Fanciulli, and Sabina Spiga. Analog memristive synapse in spiking networks implementing unsupervised learning. *Frontiers in neuroscience*, 10:482, 2016.

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

[45] Andrew P Davison, Daniel Brüderle, Jochen M Eppler, Jens Kremkow, Eilif Muller, Dejan Pecevski, Laurent Perrinet, and Pierre Yger. Pynn: a common interface for neuronal network simulators. *Frontiers in neuroinformatics*, 2:11, 2009.

[46] Javier Del Valle, Pavel Salev, Yoav Kalcheim, and Ivan K Schuller. A caloritronics-based mott neuristor. *Scientific Reports*, 10(1):1–10, 2020.

[47] Tobi Delbruck. Neuromorphic AI, Tutorial Session, IEEE International Conf. on Artificial Intelligence Circuits and Systems, Hsinchu, Taiwan, 2019.

[48] Yigit Demirag, Charlotte Frenkel, Melika Payvand, and Giacomo Indiveri. Online training of spiking recurrent neural networks with phase-change memory synapses. *arXiv preprint arXiv:2108.01804*, 2021.

[49] Nick Diederich, Thorsten Bartsch, Hermann Kohlstedt, and Martin Ziegler. A memristive plasticity model of voltage-based stdp suitable for recurrent bidirectional neural networks in the hippocampus. *Scientific reports*, 8(1):1–12, 2018.

[50] Peter U Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience*, 9:99, 2015.

[51] Qingxi Duan, Zhaokun Jing, Xiaolong Zou, Yanghao Wang, Ke Yang, Teng Zhang, Si Wu, Ru Huang, and Yuchao Yang. Spiking neurons with spatiotem-

poral dynamics and gain modulation for monolithically integrated memristive neural networks. *Nature communications*, 11(1):1–13, 2020.

[52] Jason K Eshraghian, Seungbum Baek, Jun-Ho Kim, Nicolangelo Iannella, Kyoungrok Cho, Yong Sook Goo, Herbert HC Iu, Sung-Mo Kang, and Kamran Eshraghian. Formulation and implementation of nonlinear integral equations to model neural dynamics within the vertebrate retina. *International Journal of Neural Systems*, 28(07):1850004, 2018.

[53] Jason K Eshraghian, Kyoungrok Cho, and Sung Mo Kang. A 3-d reconfigurable rram crossbar inference engine. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2021.

[54] Jason K Eshraghian, Sung-Mo Kang, Seungbum Baek, Garrick Orchard, Herbert Ho-Ching Iu, and Wen Lei. Analog weights in reram dnn accelerators. In *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 267–271. IEEE, 2019.

[55] Jason K Eshraghian and Wei D Lu. The fine line between dead neurons and sparsity in binarized spiking neural networks. *arXiv preprint arXiv:2201.11915*, 2022.

[56] Jason K Eshraghian, Xinxin Wang, and Wei D Lu. Memristor-based binarized spiking neural networks: Challenges and applications. *IEEE Nanotechnology Magazine*, 2022.

[57] Jason K Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz,

Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D Lu. Training spiking neural networks using lessons from deep learning. *arXiv preprint arXiv:2109.12894*, 2021.

[58] Jason Kamran Eshraghian, Kyoungrok Cho, Ciyan Zheng, Minho Nam, Herbert Ho-Ching Iu, Wen Lei, and Kamran Eshraghian. Neuromorphic vision hybrid rram-cmos architecture. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(12):2816–2829, 2018.

[59] Jason Kamran Jr Eshraghian. *Retinal and Neural Dynamics using Memristor-CMOS Architectures*. PhD thesis, The University of Western Australia, 2019.

[60] Bertrand Fontaine, Dan FM Goodman, Victor Benichoux, and Romain Brette. Brian hears: online auditory processing using vectorization over channels. *frontiers in Neuroinformatics*, 5:9, 2011.

[61] Elliot J Fuller, Scott T Keene, Armantas Melianas, Zhongrui Wang, Sapan Agarwal, Yiyang Li, Yaakov Tuchman, Conrad D James, Matthew J Marinella, J Joshua Yang, et al. Parallel programming of an ionic floating-gate memory array for scalable neuromorphic computing. *Science*, 364(6440):570–574, 2019.

[62] Steve Furber. Large-scale neuromorphic computing systems. *Journal of neural engineering*, 13(5):051001, 2016.

[63] Steve B Furber, Francesco Galluppi, Steve Temple, and Luis A Plana. The spinnaker project. *Proceedings of the IEEE*, 102(5):652–665, 2014.

[64] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[65] Marc-Oliver Gewaltig and Markus Diesmann. Nest (neural simulation tool). *Scholarpedia*, 2(4):1430, 2007.

[66] Abed Ghanbari, Aleksey Malyshev, Maxim Volgushev, and Ian H Stevenson. Estimating short-term synaptic plasticity from pre-and postsynaptic spiking. *PLoS computational biology*, 13(9):e1005738, 2017.

[67] Dan FM Goodman and Romain Brette. The brian simulator. *Frontiers in neuroscience*, 3:26, 2009.

[68] Benedikt Grothe. New roles for synaptic inhibition in sound localization. *Nature Reviews Neuroscience*, 4(7):540–550, 2003.

[69] Song Hao, Xinglong Ji, Shuai Zhong, Khin Yin Pang, Kian Guan Lim, Tow Chong Chong, and Rong Zhao. A monolayer leaky integrate-and-fire neuron for 2d memristive neuromorphic networks. *Advanced Electronic Materials*, 6(4):1901335, 2020.

[70] Leslie A Hart. *How the brain works*. Basic Books, 1975.

[71] Raqibul Hasan, Tarek M Taha, and Chris Yakopcic. On-chip training of memristor crossbar based multi-layer neural networks. *Microelectronics Journal*, 66:31–40, 2017.

[72] Hananel Hazan, Daniel J Saunders, Hassaan Khan, Devdhar Patel, Darpan T Sanghavi, Hava T Siegelmann, and Robert Kozma. Bindsnet: A machine learning-oriented spiking neural networks library in python. *Frontiers in Neuroinformatics*, page 89, 2018.

[73] Donald Hebb. 0. the organization of behavior, 1949.

[74] Michael L Hines and Nicholas T Carnevale. Neuron: a tool for neuroscientists. *The neuroscientist*, 7(2):123–135, 2001.

[75] Joel Hochstetter, Ruomin Zhu, Alon Loeffler, Adrian Diaz-Alvarez, Tomonobu Nakayama, and Zdenka Kuncic. Avalanches and edge-of-chaos learning in neuromorphic nanowire networks. *Nature Communications*, 12(1):1–13, 2021.

[76] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.

[77] Jinqi Huang, Spyros Stathopoulos, Alex Serb, and Themis Prodromakis. Neuropack: An algorithm-level python-based simulator for memristor-empowered neuro-inspired computing. *arXiv preprint arXiv:2201.03339*, 2022.

[78] John R Hughes. Post-tetanic potentiation. *Physiological reviews*, 38(1):91–113, 1958.

[79] Javier Iglesias and Alessandro EP Villa. Neuronal cell death and synaptic pruning driven by spike-timing dependent plasticity. In *International Conference on Artificial Neural Networks*, pages 953–962. Springer, 2006.

[80] Lloyd A Jeffress. A place theory of sound localization. *Journal of comparative and physiological psychology*, 41(1):35, 1948.

[81] Sung Hyun Jo, Ting Chang, Idongesit Ebong, Bhavitavya B Bhadviya, Pinaki Mazumder, and Wei Lu. Nanoscale memristor device as synapse in neuromorphic systems. *Nano letters*, 10(4):1297–1301, 2010.

[82] Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven Siegelbaum, A James Hudspeth, and Sarah Mack. *Principles of Neural Science*, volume 4. McGraw-hill New York, 2000.

[83] Sung Mo Kang, Donguk Choi, Jason K Eshraghian, Peng Zhou, Jieun Kim, Bai-Sun Kong, Xiaojian Zhu, Ahmet Samil Demirkol, Alon Ascoli, Ronald Tetzlaff, and Leon O Chua. How to build a memristive integrate-and-fire model for spiking neuronal signal generation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2021.

[84] Sung Mo Kang, Robert H Krambeck, H-FS Law, and Alexander D Lopez. Gate matrix layout of random control logic in a 32-bit cmos cpu chip adaptable to evolving logic design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2(1):18–29, 1983.

[85] Fatemeh Kiani, Jun Yin, Zhongrui Wang, J Joshua Yang, and Qiangfei Xia. A fully hardware-based memristive multilayer neural network. *Science Advances*, 7(48):eabj4801, 2021.

[86] Sungho Kim, Chao Du, Patrick Sheridan, Wen Ma, ShinHyun Choi, and Wei D

Lu. Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity. *Nano Letters*, 15(3):2203–2211, 2015.

[87] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[88] Christof Koch and Idan Segev. *Methods in neuronal modeling: from ions to networks*. MIT press, 1998.

[89] Hiroshi Kuba, Konomi Koyano, and Harunori Ohmori. Synaptic depression improves coincidence detection in the nucleus laminaris in brainstem slices of the chick embryo. *European Journal of Neuroscience*, 15(6):984–990, 2002.

[90] Corey Lammie and Mostafa Rahimi Azghadi. Memtorch: A simulation framework for deep memristive cross-bar architectures. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2020.

[91] Corey Lammie, Jason K Eshraghian, Wei D Lu, and Mostafa Rahimi Azghadi. Memristive stochastic computing for deep learning parameter optimization. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(5):1650–1654, 2021.

[92] Corey Lammie, Wei Xiang, Bernabé Linares-Barranco, and Mostafa Rahimi Azghadi. Memtorch: An open-source simulation framework for memristive deep learning systems. *Neurocomputing*, 485:124–133, 2022.

[93] Louis Lapique. Recherches quantitatives sur l'excitation electrique des nerfs

traitee comme une polarization. *Journal of Physiology and Pathololgy*, 9:620–635, 1907.

[94] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[95] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[96] Yunning Li. Analog computing using 1t1r crossbar arrays. 2018.

[97] Hyungkwang Lim, Vladimir Kornijcuk, Jun Yeong Seok, Seong Keun Kim, Inho Kim, Cheol Seong Hwang, and Doo Seok Jeong. Reliability of neuronal information conveyed by unreliable neuristor-based leaky integrate-and-fire neurons: a model study. *Scientific Reports*, 5(1):1–15, 2015.

[98] Chih-Yang Lin, Jia Chen, Po-Hsun Chen, Ting-Chang Chang, Yuting Wu, Jason K Eshraghian, John Moon, Sangmin Yoo, Yu-Hsun Wang, Wen-Chung Chen, et al. Adaptive synaptic memory via lithium ion modulation in rram devices. *Small*, 16(42):2003964, 2020.

[99] Alon Loeffler, Ruomin Zhu, Joel Hochstetter, Adrian Diaz-Alvarez, Tomonobu Nakayama, James M Shine, and Zdenka Kuncic. Modularity and multitasking in neuro-memristive reservoir networks. *Neuromorphic Computing and Engineering*, 1(1):014003, 2021.

[100] Misha Mahowald. The silicon retina. In *An Analog VLSI System for Stereoscopic Vision*, pages 4–65. Springer, 1994.

[101] Gabriel Maranhão and Janaina Gonçalves Guimarães. Low-power hybrid memristor-cmos spiking neuromorphic stdp learning system. *IET Circuits, Devices & Systems*, 15(3):237–250, 2021.

[102] Adam H Marblestone, Greg Wayne, and Konrad P Kording. Toward an integration of deep learning and neuroscience. *Frontiers in computational neuroscience*, 10:94, 2016.

[103] Carver Mead. *Analog VLSI and neural systems*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[104] Carver Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636, 1990.

[105] Carver Mead and Mohammed Ismail. *Analog VLSI implementation of neural systems*, volume 80. Springer Science & Business Media, 2012.

[106] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

[107] Timothy W Molter and M Alexander Nugent. The generalized metastable switch memristor model. In *CNNA 2016; 15th International workshop on cellular nanoscale networks and their applications*, pages 1–2. VDE, 2016.

[108] Don Monroe. Neuromorphic computing gets ready for the (really) big time, 2014.

[109] Saber Moradi, Ning Qiao, Fabio Stefanini, and Giacomo Indiveri. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps). *IEEE transactions on biomedical circuits and systems*, 12(1):106–122, 2017.

[110] Manu V Nair. *Memristive Crossbar Arrays for Machine Learning Systems*. PhD thesis, The University of Manchester (United Kingdom), 2015.

[111] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[112] Denis Noble. A modification of the hodgkin—huxley equations applicable to purkinje fibre action and pacemaker potentials. *The Journal of physiology*, 160(2):317–352, 1962.

[113] Garrick Orchard, E Paxon Frady, Daniel Ben Dayan Rubin, Sophia Sanborn, Sumit Bam Shrestha, Friedrich T Sommer, and Mike Davies. Efficient neuromorphic signal processing with loihi 2. In *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 254–259. IEEE, 2021.

[114] Lichtsteiner Patrick, Christoph Posch, and Tobi Delbruck. A 128×128 120 db

15μ s latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43:566–576, 2008.

[115] Christian Pehle and Jens Egholm Pedersen. Norse - A deep learning library for spiking neural networks, January 2021. Documentation: https://norse.ai/docs/.

[116] Lillian Pentecost, Alexander Hankin, Marco Donato, Mark Hempstead, Gu-Yeon Wei, and David Brooks. Nvmexplorer: A framework for cross-stack comparisons of embedded non-volatile memories. *arXiv preprint arXiv:2109.01188*, 2021.

[117] Matthew D Pickett, Gilberto Medeiros-Ribeiro, and R Stanley Williams. A scalable neuristor built with mott memristors. *Nature materials*, 12(2):114–117, 2013.

[118] Harvard Health Publishing. Calories burned in 30 minutes for people of three different weights, 2018.

[119] Mostafa Rahimi Azghadi, Ying-Chen Chen, Jason K Eshraghian, Jia Chen, Chih-Yang Lin, Amirali Amirsoleimani, Adnan Mehonic, Anthony J Kenyon, Burt Fowler, Jack C Lee, et al. Complementary metal-oxide semiconductor and memristive hardware for neuromorphic computing. *Advanced Intelligent Systems*, 2(5):1900189, 2020.

[120] Mostafa Rahimiazghadi, Corey Lammie, Jason Kamran Eshraghian, Melika Payvand, Elisa Donati, Bernabe Linares-Barranco, and Giacomo Indiveri.

Hardware implementation of deep network accelerators towards healthcare and biomedical applications. *IEEE Transactions on Biomedical Circuits and Systems*, 2020.

[121] Malte J Rasch, Diego Moreda, Tayfun Gokmen, Manuel Le Gallo, Fabio Carta, Cindy Goldberg, Kaoutar El Maghraoui, Abu Sebastian, and Vijay Narayanan. A flexible and fast pytorch toolkit for simulating training and inference on analog crossbar arrays. In *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 1–4. IEEE, 2021.

[122] Nitin Rathi, Priyadarshini Panda, and Kaushik Roy. Stdp-based pruning of connections and weight quantization in spiking neural networks for energy-efficient recognition. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(4):668–677, 2018.

[123] Michael Risoud, J-N Hanson, Fanny Gauvrit, C Renard, P-E Lemesre, N-X Bonne, and Christophe Vincent. Sound source localization. *European annals of otorhinolaryngology, head and neck diseases*, 135(4):259–264, 2018.

[124] A Roth and MCW Van Rossum. Modeling synapses. computational modeling methods for neuroscientists, 2009.

[125] MJ Rozenberg, O Schneegans, and P Stoliar. An ultra-compact leaky-integrate-and-fire model for building spiking neural networks. *Scientific Reports*, 9(1):1–7, 2019.

[126] Marek Rudnicki, Oliver Schoppe, Michael Isik, Florian Völk, and Werner Hemmert. Modeling auditory coding: from sound to spikes. *Cell and Tissue Research*, 361(1):159–175, 2015.

[127] Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11:682, 2017.

[128] Johannes Schemmel, Daniel Brüderle, Andreas Grübl, Matthias Hock, Karlheinz Meier, and Sebastian Millner. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1947–1950. IEEE, 2010.

[129] Jan Schnupp, Israel Nelken, and Andrew King. *Auditory neuroscience: Making sense of sound*. MIT press, 2011.

[130] Catherine D Schuman, Thomas E Potok, Robert M Patton, J Douglas Birdwell, Mark E Dean, Garrett S Rose, and James S Plank. A survey of neuromorphic computing and neural networks in hardware. *arXiv preprint arXiv:1705.06963*, 2017.

[131] Teresa Serrano-Gotarredona, Timothée Masquelier, Themistoklis Prodromakis, Giacomo Indiveri, and Bernabe Linares-Barranco. STDP and STDP variations with memristors for spiking neuromorphic learning systems. *Frontiers in neuroscience*, 7:2, 2013.

[132] Shihab A Shamma. Speech processing in the auditory system ii: Lateral inhibition and the central processing of speech evoked activity in the auditory nerve. *The Journal of the Acoustical Society of America*, 78(5):1622–1632, 1985.

[133] Carla J Shatz. The developing brain. *Scientific American*, 267(3):60–67, 1992.

[134] S M Sherman, R W Guillery, and Shepherd G M. *Synaptic organization of the brain*. Thalamus. Oxford University Press: Oxford, 2004.

[135] RB Stein and Alan Lloyd Hodgkin. The frequency of nerve action potentials generated by applied currents. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 167(1006):64–86, 1967.

[136] Marcel Stimberg, Romain Brette, and Dan FM Goodman. Brian 2, an intuitive and efficient neural simulator. *Elife*, 8:e47314, 2019.

[137] Christoph Stöckl and Wolfgang Maass. Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. *Nature Machine Intelligence*, 3(3):230–238, 2021.

[138] Dmitri B Strukov, Gregory S Snider, Duncan R Stewart, and R Stanley Williams. The missing memristor found. *Nature*, 453(7191):80–83, 2008.

[139] Linfeng Sun, Yishu Zhang, Geunwoo Hwang, Jinbao Jiang, Dohyun Kim, Yonas Assefa Eshete, Rong Zhao, and Heejun Yang. Synaptic computation enabled by joule heating of single-layered semiconductors for sound localization. *Nano letters*, 18(5):3229–3234, 2018.

[140] Doron Tal and Eric L Schwartz. Computing with the leaky integrate-and-fire neuron: logarithmic computation and multiplication. *Neural Computation*, 9(2):305–318, 1997.

[141] Hongwei Tan, Sayani Majumdar, Qihang Qin, Jouko Lahtinen, and Sebastiaan van Dijken. Mimicking neurotransmitter release and long-term plasticity by oxygen vacancy migration in a tunnel junction memristor. *Advanced Intelligent Systems*, 1(2):1900036, 2019.

[142] Zhiri Tang, Yanhua Chen, Shizhuo Ye, Ruihan Hu, Hao Wang, Jin He, Qijun Huang, and Sheng Chang. Fully memristive spiking-neuron learning framework and its applications on pattern recognition and edge detection. *Neurocomputing*, 403:80–87, 2020.

[143] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. Deep learning in spiking neural networks. *Neural Networks*, 111:47–63, 2019.

[144] Corinne Teeter, Ramakrishnan Iyer, Vilas Menon, Nathan Gouwens, David Feng, Jim Berg, Aaron Szafer, Nicholas Cain, Hongkui Zeng, Michael Hawrylycz, et al. Generalized leaky integrate-and-fire models classify multiple neuron types. *Nature Communications*, 9(1):1–15, 2018.

[145] Ronald Tetzlaff. *Memristors and Memristive Systems*. Springer, 2013.

[146] Chetan Singh Thakur, Jamal Lottier Molin, Gert Cauwenberghs, Giacomo Indiveri, Kundan Kumar, Ning Qiao, Johannes Schemmel, Runchun Wang,

Elisabetta Chicca, Jennifer Olson Hasler, et al. Large-scale neuromorphic spiking array processors: A quest to mimic the brain. *Frontiers in Neuroscience*, 12:891, 2018.

[147] Roberto Toro, Michel Perron, Bruce Pike, Louis Richer, Suzanne Veillette, Zdenka Pausova, and Tomáš Paus. Brain size and folding of the human cerebral cortex. *Cerebral Cortex*, 18(10):2352–2357, 2008.

[148] Roshni Uppala. *Simulating large scale memristor based crossbar for neuromorphic applications*. PhD thesis, University OF Dayton, 2015.

[149] M Mitchell Waldrop. The chips are down for moore's law. *Nature News*, 530(7589):144, 2016.

[150] JJ Wang, SG Hu, XT Zhan, Q Yu, Z Liu, Tu Pei Chen, Y Yin, Sumio Hosaka, and Y Liu. Handwritten-digit recognition by hybrid convolutional neural network based on hfo2 memristive spiking-neuron. *Scientific Reports*, 8(1):1–7, 2018.

[151] Qiwen Wang, Xinxin Wang, Seung Hwan Lee, Fan-Hsuan Meng, and Wei D Lu. A deep neural network accelerator based on tiled rram architecture. In *2019 IEEE International Electron Devices Meeting (IEDM)*, pages 14–4. IEEE, 2019.

[152] Ruopeng Wang, Jia-Qin Yang, Jing-Yu Mao, Zhan-Peng Wang, Shuang Wu, Maojie Zhou, Tianyi Chen, Ye Zhou, and Su-Ting Han. Recent advances of

volatile memristors: Devices, mechanisms, and applications. *Advanced Intelligent Systems*, 2(9):2000055, 2020.

[153] Wei Wang, Wenhao Song, Peng Yao, Yang Li, Joseph Van Nostrand, Qinru Qiu, Daniele Ielmini, and J Joshua Yang. Integration and co-design of memristive devices and algorithms for artificial intelligence. *Iscience*, page 101809, 2020.

[154] Xiangwen Wang, Xianghong Lin, and Xiaochao Dang. Supervised learning in spiking neural networks: A review of algorithms and evaluations. *Neural Networks*, 125:258–280, 2020.

[155] Xiao-Yuan Wang, Chuan-Tao Dong, Peng-Fei Zhou, Sanjoy Kumar Nandi, Shimul Kanti Nath, Robert G Elliman, Herbert Ho-Ching Iu, Sung-Mo Kang, and Jason K Eshraghian. Low-variance memristor-based multi-level ternary combinational logic. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2022.

[156] Xiao-Yuan Wang, Peng-Fei Zhou, Jason K Eshraghian, Chih-Yang Lin, Herbert Ho-Ching Iu, Ting-Chang Chang, and Sung-Mo Kang. High-density memristor-cmos ternary logic family. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(1):264 – 274, 2021.

[157] Yan Wang, Ziyu Lv, Qiufan Liao, Haiquan Shan, Jinrui Chen, Ye Zhou, Li Zhou, Xiaoli Chen, Vellaisamy AL Roy, Zhanpeng Wang, et al. Synergies of electrochemical metallization and valance change in all-inorganic perovskite

quantum dots for resistive switching. *Advanced Materials*, 30(28):1800327, 2018.

[158] Zhongrui Wang, Saumil Joshi, Sergey Savel'ev, Wenhao Song, Rivu Midya, Yunning Li, Mingyi Rao, Peng Yan, Shiva Asapu, Ye Zhuo, et al. Fully memristive neural networks for pattern classification with unsupervised learning. *Nature Electronics*, 1(2):137–145, 2018.

[159] Rainer Waser and Masakazu Aono. Nanoionics-based resistive switching memories. *Nanoscience And Technology: A Collection of Reviews from Nature Journals*, pages 158–165, 2010.

[160] Eric W. Weisstein. From mathworld–a wolfram web resource. `https://mathworld.wolfram.com/AlphaFunction.html`.

[161] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[162] Parami Wijesinghe, Aayush Ankit, Abhronil Sengupta, and Kaushik Roy. An all-memristor deep spiking neural computing system: A step toward realizing the low-power stochastic brain. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(5):345–358, 2018.

[163] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989.

[164] Timo Wunderlich, Akos F Kungl, Eric Müller, Andreas Hartel, Yannik Strad-

mann, Syed Ahmed Aamir, Andreas Grübl, Arthur Heimbrecht, Korbinian Schreiber, David Stöckel, et al. Demonstrating advantages of neuromorphic computation: a pilot study. *Frontiers in Neuroscience*, 13:260, 2019.

[165] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[166] Chris Yakopcic. *Memristor device modeling and circuit design for read out integrated circuits, memory architectures, and neuromorphic systems*. University of Dayton, 2014.

[167] Chris Yakopcic, Tarek M Taha, Guru Subramanyam, and Robinson E Pino. Generalized memristive device spice model and its application in circuit design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(8):1201–1214, 2013.

[168] Steven Yantis and Richard A Abrams. *Sensation and perception*. Worth Publishers New York, NY, 2014.

[169] Wei Yi, Kenneth K Tsang, Stephen K Lam, Xiwei Bai, Jack A Crowell, and Elias A Flores. Biological plausibility and stochasticity in scalable vo 2 active memristor neurons. *Nature Communications*, 9(1):1–10, 2018.

[170] Xumeng Zhang, Wei Wang, Qi Liu, Xiaolong Zhao, Jinsong Wei, Rongrong Cao, Zhihong Yao, Xiaoli Zhu, Feng Zhang, Hangbing Lv, et al. An artifi-

cial neuron based on a threshold switching memristor. *IEEE Electron Device Letters*, 39(2):308–311, 2017.

[171] Yang Zhang, Zhongrui Wang, Jiadi Zhu, Yuchao Yang, Mingyi Rao, Wenhao Song, Ye Zhuo, Xumeng Zhang, Menglin Cui, Linlin Shen, et al. Brain-inspired computing with memristors: Challenges in devices, circuits, and systems. *Applied Physics Reviews*, 7(1):011308, 2020.

[172] Peng Zhou and Shiva Abbaszadeh. Towards real-time machine learning for anomaly detection. In *2020 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–3. IEEE.

[173] Peng Zhou, Dong-Uk Choi, Jason K Eshraghian, and Sung-Mo Kang. A fully memristive spiking neural network with unsupervised learning. *arXiv preprint arXiv:2203.01416*, 2022.

[174] Peng Zhou, Jason K Eshraghian, Dong-Uk Choi, and Sung-Mo Kang. Spiceprop: Backpropagating errors through memristive spiking neural networks. *arXiv preprint arXiv:2203.01426*, 2022.

[175] Ruomin Zhu, Alon Loeffler, Joel Hochstetter, Adrian Diaz-Alvarez, Tomonobu Nakayama, Adam Stieg, James Gimzewski, Joseph Lizier, and Zdenka Kuncic. Mnist classification using neuromorphic nanowire networks. In *International Conference on Neuromorphic Systems 2021*, pages 1–4, 2021.

[176] Xiaojian Zhu, Chao Du, YeonJoo Jeong, and Wei D Lu. Emulation of synaptic metaplasticity in memristors. *Nanoscale*, 9(1):45–51, 2017.

[177] Xiaojian Zhu, Jihang Lee, and Wei D Lu. Iodine vacancy redistribution in organic–inorganic halide perovskite films and resistive switching effects. *Advanced Materials*, 29(29):1700527, 2017.

[178] Xiaojian Zhu, Qiwen Wang, and Wei D Lu. Memristor networks for real-time neural activity analysis. *Nature communications*, 11(1):1–9, 2020.

[179] Robert S Zucker and Wade G Regehr. Short-term synaptic plasticity. *Annual review of physiology*, 64(1):355–405, 2002.

Appendix A

Authorship Declaration: Co-authored Publications

This dissertation contains research articles that have been published/submitted as listed below:

- Sung Mo Kang, Donguk Choi, Jason K. Eshraghian, **Peng Zhou**, Jieun Kim, Bai-Sun Kong, Xiaojian Zhu, Ahmet Samil Demirkol, Alon Ascoli, Ronald Tetzlaff, Wei. D. Lu, and Leon O. Chua. "How to build a memristive integrate-and-fire model for spiking neuronal signal generation." in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 12, pp. 4837-4850, Dec. 2021, doi: 10.1109/TCSI.2021.3126555.

  Location in dissertation: Chapter 4

- **Peng Zhou**, Donguk Choi, Jason K. Eshraghian, and Sung-Mo Kang. "A Fully Memristive Spiking Neural Network with Unsupervised Learning." 2022 IEEE International Symposium on Circuits and Systems (ISCAS), 2022, pp.

634-638, doi: 10.1109/ISCAS48785.2022.9937309.

Location in dissertation: Chapter 5

- **Peng Zhou**, Jason K. Eshraghian, Donguk Choi, and Sung-Mo Kang. "Gradient-based Neuromorphic Learning on Dynamical RRAM Arrays." in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2022, doi: 10.1109/JETCAS.2022.3224071.

  Location in dissertation: Chapter 6

- **Peng Zhou**, Donguk Choi, Sung-Mo Kang, and Jason K. Eshraghian. "Back-propagating Errors Through Memristive Spiking Neural Networks." Submitted to 2023 IEEE International Symposium on Circuits and Systems (ISCAS).

  Location in dissertation: Chapter 6