NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

BIOLOGICALLY INSPIRED AUTOMATIC TARGET DETECTION, CLASSIFICATION, AND TRACKING

by

Eric Kim

September 2020

Thesis Advisor: Vinnie Monaco
Second Reader: Mathias N. Kolsch

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE BIOLOGICALLY INSPIRED AUTOMATIC TARGET DETECTION, CLASSIFICATION, AND TRACKING	**5. FUNDING NUMBERS**
6. AUTHOR(S) Eric Kim	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

The decision to use and deliver kinetic and/or non-kinetic fires has been and will forever be entwined into war. The U.S. military and specifically the Marines have been the masters of this process but to maintain this superiority, fire support needs to become more timely, discriminatory, lethal, and effective in today's data-saturated environment. Also, with the distributive nature of the modern operating environment, producing a SWaP-T-compatible solution is vital. To bridge these gaps, the author proposes to offload the target detection, classification, and tracking to a biologically inspired automated system composed of a Dynamic Vision Sensor and a spiking neural network running on neuromorphic hardware. Emphasis was placed on the spiking neural network algorithm development and building/evaluating the system. The author found that this approach could yield a system that will provide the warfighter with actionable information to improve the kill chain process while minimizing power consumption and time taken at the point of collection. The hope is that the research presented here will spur advances in the field of biologically inspired neuromorphic platforms that will produce timely, accurate, distilled, and actionable information to the end user to offload mundane/trivial tasks to allow for more decision time and space.

14. SUBJECT TERMS fire support, kill chain management, artificial intelligence, spiking neural networks, neuromorphic computing, computer vision, computer architectures	**15. NUMBER OF PAGES** 75
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

BIOLOGICALLY INSPIRED AUTOMATIC TARGET DETECTION, CLASSIFICATION, AND TRACKING

Eric Kim
Major, United States Marine Corps
BA, University of California, Berkeley, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2020

Approved by: Vinnie Monaco
Advisor

Mathias N. Kolsch
Second Reader

Gurminder Singh
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The decision to use and deliver kinetic and/or non-kinetic fires has been and will forever be entwined into war. The U.S. military and specifically the Marines have been the masters of this process but to maintain this superiority, fire support needs to become more timely, discriminatory, lethal, and effective in today's data-saturated environment. Also, with the distributive nature of the modern operating environment, producing a SWaP-T-compatible solution is vital. To bridge these gaps, the author proposes to offload the target detection, classification, and tracking to a biologically inspired automated system composed of a Dynamic Vision Sensor and a spiking neural network running on neuromorphic hardware. Emphasis was placed on the spiking neural network algorithm development and building/evaluating the system. The author found that this approach could yield a system that will provide the warfighter with actionable information to improve the kill chain process while minimizing power consumption and time taken at the point of collection. The hope is that the research presented here will spur advances in the field of biologically inspired neuromorphic platforms that will produce timely, accurate, distilled, and actionable information to the end user to offload mundane/trivial tasks to allow for more decision time and space.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AP	action potential
CNN	convolutional neural network
DNN	deep neural network
DVS	Dynamic Vision Sensor
FLOPS	floating point operations per second
FPS	frames per second
LIF	Leaky-Integrate-and-Fire
MTJ	magnetic tunnel junction
MNIST	Modified National Institute of Standards and Technology
PSP	post-synaptic potential
SNN	Spiking Neural Network
STEAM	Spike Time Encoded Addressable Memory
STDP	Spike Timing Dependent Plasticity
STICK	Spike Time Interval Computational Kernel
SWaP-T	size, weight, and power-time

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

To my dad,
Leland M. Garrison, MD:
Without you, none of this would have been possible.
“Quarta Matus”

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

With the increase of intelligence-gathering sources and point-of-collection data, the targeting process for kinetic fires has become task saturating for a human to process, leading to decision paralysis. Offloading the detection, classification, and tracking of objects to an autonomous system can make the kill chain more efficient, discriminatory, and allow the human in the loop to make pertinent, timely, and accurate decisions on the prosecution of targets. This will allow the human in the loop to focus on the decision making process with distilled and refined targeting data. Also, in the current and future operating environment that our troops will find themselves in, an autonomous system must not burden the end-user by having a large footprint, being computationally heavy, and power hungry. Future systems must be developed to operate in a size, weight, and power-time (SWaP-T) constrained environment to allow the operators on the ground the freedom of movement to out-pace the emerging threats.

This research investigates the possibility of utilizing state-of-the-art artificial intelligence, neuromorphic computing, and computer vision as a computer architecture to advance point-of-collection targeting data. Future applications of this proposed architecture could help advance intelligence-gathering, fire support, and kill chain management while not introducing an information processing burden on the end-user. The intent is to provide the foundational research and impetus to address the need for developing a scalable system and architecture that utilizes next generation Spiking Neural Network (SNN)s on neuromorphic hardware to conduct automated target detection, classification, and tracking in a SWaP-T constrained environment.

Three specific challenges of automating target detection, classification, and tracking are tackled in this work:

1. Conducting pattern recognition via a biologically-inspired algorithm. This comprises *sub-symbolic* processing on a neuromorphic platform.
2. Conducting *symbolic processing* on the same neuromorphic platform in the form of counting discrete elements.

3. Integrating sub-symbolic and symbolic processing within a common neuromorphic architecture.

One of the main contributions of this work is bridging the gap between pattern recognition via sub-symbolic processing and basic high-order cognitive operations through symbolic processing. Sub-symbolic processing takes an input, decomposes it into n-dimensional features, and provides an output based upon statistical values. Examples include Bayesian learning, deep learning, and traditional neural networks. On the other hand, symbolic processing takes a symbol or variable as an input, manipulates symbols based upon a predefined set of rules, and outputs a new expression based upon those rules. These are both implemented on a simulated neuromorphic architecture, specifically a SNN, as SNN-based chips have been found to offer reduced processing and power requirements which in turn leads to smaller and lighter hardware [1]–[3].

Offloading the detection, classification, and tracking of objects via visual spectrum sensors running on neuromorphic hardware is modeled. We propose that a Dynamic Vision Sensor (DVS) be used as an input device, neuromorphic hardware as the machinery to run on, and SNNs as the software and means to conduct sub-symbolic and symbolic processing. The emphasis of this work is on the training of the SNN to detect and classify an object, basic counting via symbolic processing utilizing Spike Time Interval Computational Kernel (STICK), and the linkages needed to integrate the two. Results are obtained on a simulated neuromorphic architecture compatible with several emerging platforms.

This thesis is organized as follows. We begin with a literature review and background research into the software and hardware components needed to build a biologically-inspired target detection system focusing on object detection, classification, and tracking. Software components include SNNs, unsupervised Spike Timing Dependent Plasticity (STDP), and STICK. The hardware aspects include neuromorphic computing, a means of simulation, and DVSs. We then describe algorithm development and system design. Next, we build the detection and classification model and describe the training process. Once a functioning baseline is achieved, symbolic processing is integrated. Finally, as a proof of concept, system integration is demonstrated via simulated inputs.

This study hopes to show to the military and intelligence communities the ascendancy of biologically inspired artificial intelligence, specifically SNNs, by providing a framework for

streamlining the targeting collection and kill chain process. Once adopted and tested, the results of this research could guide the development of devices that must operate in SWaP-T constrained environments where accurate detection, classification, and tracking of objects is conducted as quickly and reliably as possible while minimizing power consumption and error. This will provide the end-user with accurate and distilled information and more time to make a decision on the targeting and kill chain process, mitigating decision paralysis.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Literature Review

The intent of this literature review is to describe the foundational research into pursuing an end-to-end neuromorphic solution to automatic target detection, classification, and tracking. Beginning with the software development, the three main approaches to SNN learning are compared followed by an overview of symbolic processing within a SNN with STICK. Finally, the hardware component is addressed by discussing sensor options and leading neuromorphic hardware platforms. A thorough analysis of all possible methods of training and performing symbolic processing within a SNN is beyond the scope of this work. See [1], [4]–[6] for more information.

2.1 Spiking Neural Networks

The decision to utilize a biologically inspired platform is based upon the requirement of operating in a SWaP-T constrained environment. To maximize the efficiencies and optimizations gained in an end-to-end system, biologically inspired software should be fully implemented to run on neuromorphic hardware rather than relying on conventional computer architectures. Neuromorphic hardware and computing aims to build computers that are biologically plausible and inspired by the human brain. The impetus lies in gains in power efficiency, computational capacity, asynchrony, and parallelism [1]–[3]. The human brain operates at approximately 10 - 20 watts with a computational capacity from 10^{14} - 10^{28} floating point operations per second (FLOPS) compared to the typical personal computer running at 60 - 250 watts at 10^{13} FLOPS or the state of the art NVIDIA DGX-2 Deep Learning system running at 10,000 watts at 10^{15} FLOPS [7]–[9]. Asynchrony is achieved through event-based processing, which also allows for the hardware to be massively parallelizable [3].

To this extent, generation one perceptron and generation two feedforward or recurrent neural networks, to include state of the art convolutional neural network (CNN) were ruled out. See Figures 2.1 and 2.2, respectively, for details at the neuron level. SNNs represent the third-generation of neural network models that attempt to more closely mimic human neurobiology, therefore most fitting for this end-to-end application [1].

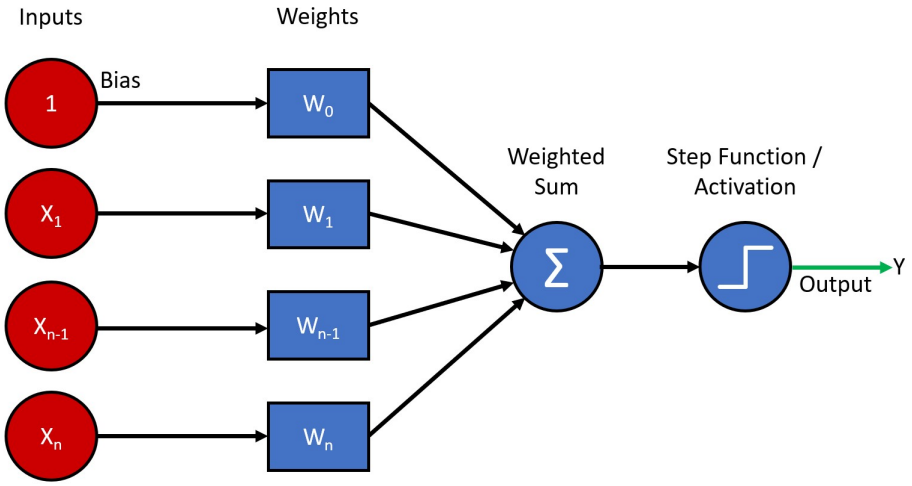

Figure 2.1. First-generation perceptron based upon the McCullock-Pitts neuron. Limited to binary classification and output.

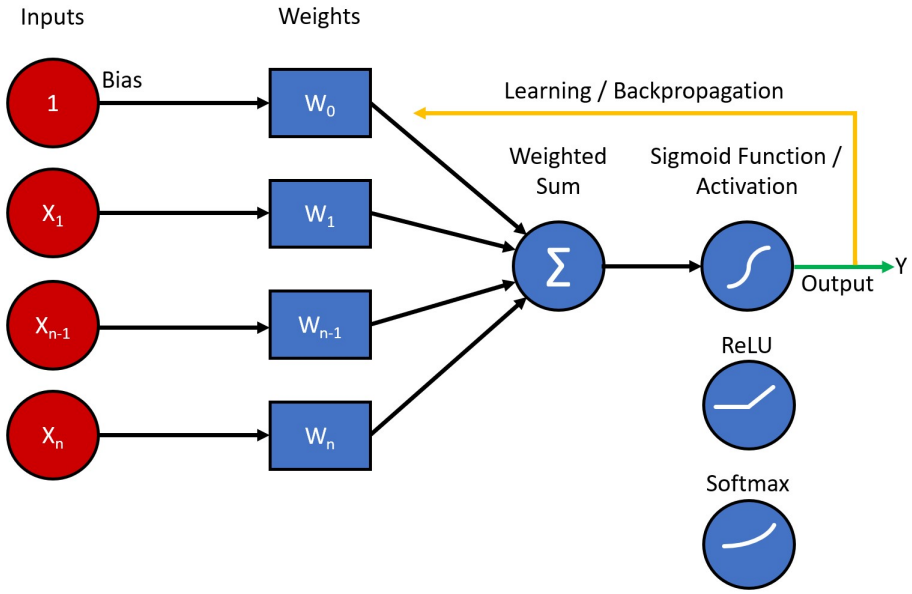

Figure 2.2. Second-generation neuron showing the addition of learning/backpropagation and three different activation functions that allow for analog/continuous set output.

The main difference between the previous generation neural networks and a SNN is the implementation of time into the model which affords neuronal and synaptic state maintenance

as well as asynchronous interactions within the network. Additionally, traditional generation one and two neural networks utilize a non-linear and continuous activation function and all neurons operate on the same clock cycle [3]. These additional features of SNNs, which more accurately model true mammalian neuronal dynamics, allow for a very important underlying attribute that inspires the direction of this research: a neuron only fires when a specific threshold is achieved. This key feature allows for decreased power consumption and event-based processing.

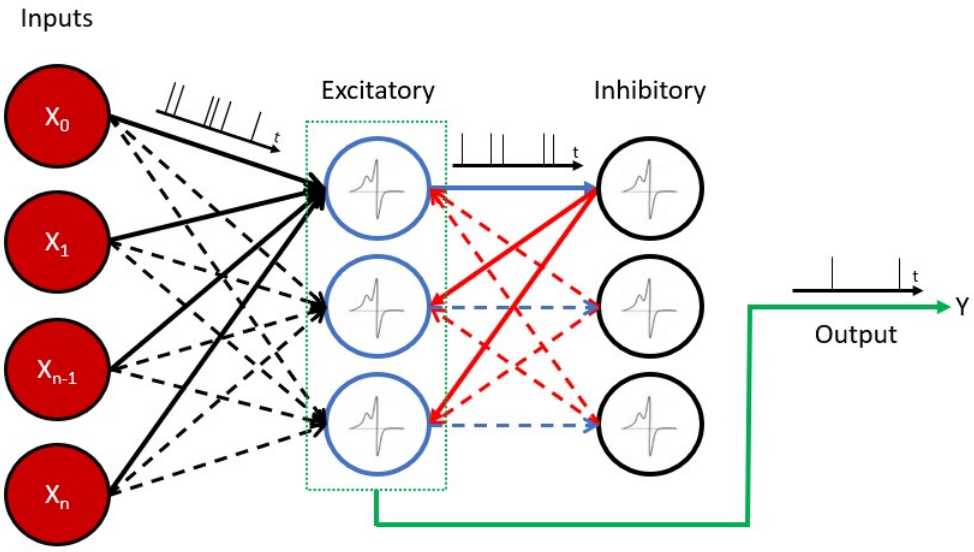

Figure 2.3. Third-generation SNN showing the addition of time, inputs encoded in spikes, and lateral inhibition.

Event-based processing and decreased power consumption go hand in hand. Unlike a rate-based neural network which requires all neurons to fire to produce an output, SNNs only require neurons for that particular stimulus and pathway to fire to produce an output while all the other neurons remain dormant. The advantages of event-based processing are demonstrated when considering visual input. A traditional frame-based video camera running at 30 hertz will provide 30 distinct inputs per second to a rate-based neural network which in turn needs to fire all neurons in the network 30 times. Conversely, a SNN will receive those same 30 inputs, but only fire the specific in-class neurons associated to the presented stimulus, only at the times when the stimulus is present, greatly reducing the computational load and power required. Energy efficiency is baked into the mammalian brain

through billions of years of evolutionary selection toward sending only the information that is required and at the lowest rate possible which in turn minimizes the length and diameter of axons as well as time of transmission [10].

While there exists a range of biologically plausible SNN models, one of the simplest is the Leaky-Integrate-and-Fire (LIF). This model describes the basic principle of how biological nervous systems network together. Presynaptic neurons send their output spike voltages to a postsynaptic neuron through a synapse. The function performed by the postsynaptic neuron is an approximate summation of the voltages due to the loss or leak of the voltage as it traverses the neurons and synaptic clefts. Finally, if enough spikes reach the postsynaptic neuron within a given time frame to push the membrane potential above a given threshold, it will fire and provide a voltage spike to its own postsynaptic neuron(s).

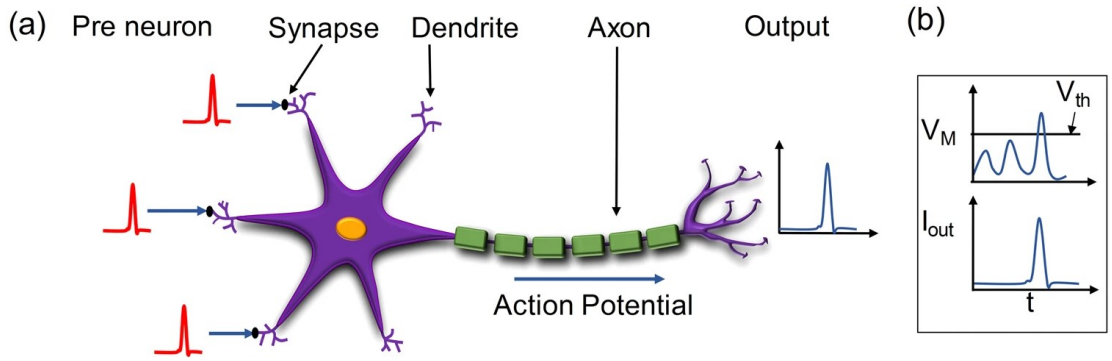

Figure 2.4. (**a**) Representation of a biological neuron from pre-neuron to output. (**b**) Output spiking of neuron with respect to the input threshold. Source: [11]

To quantify and take this big picture and idea of LIF and create a reproducible mathematical function, five main features of the biological neuronal system need to be further modeled. They are the action potential (AP), post-synaptic potential (PSP), firing threshold, refractoriness, and adaptation.

2.1.1 Action Potentials

An AP is a voltage spike created by the change in membrane potential of the target neuron based upon the rate and amplitude of input signals coming from presynaptic neurons and/or

sensors. A typical input signal, otherwise known as voltage pulse, has a duration of about 1 - 2 msec with a potential of about 1 mV. An AP, otherwise referred to as a spike, travels from the dendrite, through the neuronal soma, and down the axon where it reaches the axonal terminus at which point either a chemical and/or electrical signal is sent through the synapse to the targeted postsynaptic neuron. Dendrites act as an input function to the neuron, the soma as the neuron's computational entity, and the axon and axonal terminus as the output. The connection and space between the presynaptic neuron's axonal terminus and the postsynaptic neuron's dendrite comprise a synapse.

2.1.2 Post-Synaptic Potentials

As the name describes, this is the change in the membrane potential of the postsynaptic neuron at its input interface as it receives the output from the presynaptic neuron. This potential is graded as the signals from the presynaptic neuron(s) or sensory input(s) can be excitatory or inhibitory. It is a many-to-many relationship where a presynaptic neuron may have many connections to other neurons and a single postsynaptic neuron may have many input connections from different neurons. If the summation of the input signals at the post-synaptic membrane reach or exceed the firing threshold, an AP is generated in the postsynaptic neuron.

2.1.3 Firing Threshold

Also known as threshold potential, the firing threshold is the membrane voltage at which an AP is initiated. This corresponds to how much a neuron must depolarize, that is, change its voltage toward zero, from its resting membrane potential which is approximately -70 mV. When the input changes the post-synaptic potential and reaches the firing threshold, an all-or-none spiking depolarization occurs signifying an AP. A typical mammalian neuron has a firing threshold of -50 to -55 mV or 15 to 20 mV of depolarization from its resting membrane potential.

2.1.4 Refractory Period

This is the property of a neuron which limits the frequency of an AP within the neuron and makes it a discrete event. The refractory period of a neuron is the time it takes for the membrane to return to its resting membrane potential after a stimulus has created enough

depolarization to reach the firing threshold and initiate an AP. There are two types of refractory periods. Absolute refractory period is when no amount of stimuli will create another AP and corresponds to the depolarization and repolarization phases of the initial AP. Relative refractory period is during the hyperpolarization of the neuron as it passes and returns to its resting membrane potential. During this period, a subsequent AP can be initiate but will require a stronger stimulus to create a larger depolarization to reach the firing threshold.

2.1.5 Adaptation

This is the property of attenuating the response to a constant stimulus. Over time, at the neuronal level, a constant input at the post-synaptic membrane will have a decreasing effect on the frequency of initiated APs. This allows for the filtering of constant input so that it does not overwhelm the nervous system, tax resources, and enables focusing on other inputs from the environment. Adaptation also makes changes in the environment stand out more and helps focus sensory and processing attention to those changes.

2.1.6 Learning in SNNs

The intent of this thesis is to learn a particular optimal behavior to allow the efficient and accurate detection, classification, and tracking of targets. SNN learning is accomplished via three main modeling approaches. The order that they are presented follow the continuum toward a truly biologically inspired and plausible neural network, which is an overarching goal of this research.

Rate Base

This model of learning bridges the gap between generation two neural networks and SNNs. Learning is conducted with a traditional deep neural network (DNN) and later converted into a SNN. The conversion process takes the activations of the neurons in a traditional DNN and converts them into spiking rates as input into a SNN [3]. This method misses the mark of biological plausibility and relies on backpropagation and does not fully optimize the computational and power benefits of a pure SNN [2].

Supervised STDP

This next model of learning is a pure SNN utilizing the traditional supervised approach to learning which is analogous to supervised DNNs. Each neuron in the SNN has a teaching signal/supervisor neuron which provides feedback to the learning neurons. The biological process of STDP is used to reinforce neuronal connections in which the postsynaptic neuron fires after and within a short duration of the presynaptic neuron's firing [12]. While this approach more closely models biological neural networks by solely using a SNN framework, the supervisory neuron is an artificiality that is not biologically plausible [2].

Unsupervised STDP

This approach, which will be used in this research, utilizes a pure SNN and takes input without any supervisory signal or need for labeled data. The network learns the features of the input signal and adjusts the spike train path and weights. This allows the SNN to prune unnecessary neurons and pathways for the specific input, therefore optimizing computation and power requirements. As training/learning completes, input specific neural pathways are developed which then can be classified [2]. This is akin to showing a child the number "1" and teaching them it is "one."

2.2 Spike Timing Dependent Plasticity

STDP is one of the leading models that explain how biological learning occurs in the brain. Connections between neurons are strengthened or weakened by the timing of the postsynaptic response relative to the presynaptic input. Tightly time coupled input/output responses are strengthened while decoupled occurrence are weakened. If a presynaptic spike occurs right before the postsynaptic output spike, that presynaptic neuron's input is made stronger. The connections from presynaptic spikes that occur outside of a certain time threshold or after a postsynaptic output spike are made weaker. This allows for time dependent plasticity based upon when a spike is received at the postsynaptic dendrite. As time progresses and neuronal connections are strengthened, those weakened and unnecessary connections are pruned out and reduced to zero, leaving only the tightly time coupled connections. This creates a specific neuronal pathway based upon the original stimulus which is tightly coupled and minimizes response time.

As discussed in the SNN section, the main difference between traditional second-generation

DNNs and third-generation SNNs is the addition of time into the neuron model. Instead of using rate coding, averaging all the inputs into the neuron at a specific time step, pulse coding can be used which takes each individual presynaptic spike into account during a time frame. Therefore neurons do not fire at each time step, instead, only fire when enough presynaptic spikes are received during a certain time frame, seen as a pulse of spikes. This leads to greater computational and power efficiencies since a prescribed input will only activate the neurons in the pathway that was developed through the STDP learning process, instead of all neurons at all layers during a traditional second-generation DNN application [2].

STDP learning, as a model of biological neuronal synaptic weight adjustment, is unsupervised. It relies on the timing of the input spikes in relation to the output response to strengthen or weaken the synaptic connections, which, in the traditional sense of a DNN, adjusts the weights of the connections. This weight adjustment based upon spike timing is how learning occurs in an unsupervised manner with STDP. Backpropagation is not needed in this model as it is in traditional DNNs which require a supervisory signal to be backpropagated throughout the network of neurons to adjust connection weights and the forward propagating signal to the desired output. The lack of global error minimization through backpropagation and solely using local neuronal information via spike timing is much closer to true biologically plausible learning [2].

2.3 Symbolic Processing in Spiking Networks

To capitalize on the pattern recognition learned through STDP, a means for symbolic processing is required to provide contextual meaning to what is being observed. Symbolic processing is taking a symbol, that is, a variable, and manipulating it based upon a predefined set of rules or logic. Specifically to this research, the aim is to provide a means to count symbols. STICK offers a biologically plausible framework to conduct symbolic processing on provided inputs based upon foundational rules of arithmetic and logic [6]. This framework creates a group of specialized neurons to perform a specific task based upon these predetermined rules and/or logic. These task built neuronal groups can then be aggregated together to perform higher level symbolic processing tasks to fit the required need.

The basic premise behind STICK is encoding information in the timing between spikes and

storing those timings in the neuron models themselves based upon adjusting the membrane potential parameters. These timings are stored as transmission separations between a pair of two spikes [6]. Therefore, memory and computational processing is conducted at the same component, further reinforcing the latency benefits of neuromorphic computing and overcoming the von Neumann bottleneck. To mimic mammalian biology, timings are on the ms scale compared to the ns scale of traditional computer processors. Even with six orders of magnitude difference in precision, mammalian based timings have been proven ample enough to process relational, linear, and nonlinear operations and linear and nonlinear differential equations [6].

The STICK framework combines task specific neurons into a discrete group with predetermined synaptic connections and weights which form an independent component to conduct a basic individual task. These primitive groupings of neurons can be seen as miniature neural networks that are singular in task. The specific task is based upon the rules and logic built into the neurons and synaptic connections. An example of a primitive neural group could be a counter primitive where an input signal received by the *input* neuron adjusts the timing stored in the *incrementer* neuron by changing its membrane potential parameters. There is no learning phase for these primitive types as their task specific function is baked into their design. These basic neural groups have input and output neurons which allow them to form into larger neural networks of sub-components to accomplish higher order tasks. These inter-primitive synaptic connections can be static or dynamic to allow onboard learning and plasticity for truly unsupervised neuromorphic learning.

2.4 Neuromorphic Computing

Neuromorphic hardware attempts to model the brain's architecture by mimicking the actual cells and glia matter that comprise the physical aspects of the human brain. It is a shift away from the traditional von Neumann architecture of general purpose computer operation to provide a means for faster and more power efficient computation. This is analogous to the efforts and research in quantum computing to accelerate computation capacity. Neuromorphic computing looks to overcome the von Neumann bottleneck through biologically plausible hardware models that emulate the human brain and neuronal dynamics. It does this by asynchronous processing, known as event-driven processing and putting memory at each processing unit.

Neuromorphic hardware achieves biological plausibility through modeling each processing unit as a neuron and creating a network of neurons which can dynamically create linkages, that is, synapses, to generate a neuronal pathway. Processing units, or neurons, are grouped together in cores. The neurons in a core are typically fully connected to all other neurons within the same core. Each core is then fully connected to all other cores via intra-core connections. Neurons can adjust the weights of the synapses based upon the STDP process and only expend computational power when a neuron and/or neuronal pathway is needed or stimulated. Event-driven processing leads to massive energy consumption efficiencies and lower latency processing which all provide greater computational capacity.

Each hardware neuron is an independent computational unit that has dedicated memory associated with it. Having memory at each neuron overcomes the von Neumann bottleneck of fetching subsequent instructions from RAM and the associated latency of this retrieval due to the relative distance the signal must travel. Once STDP learning has occurred, the neuron has become purpose built and its required instructions are loaded into its dedicated memory [13]. This framework provides an optimized biologically plausible hardware architecture for SNNs to run on and be closer to human like general purpose learning [14].

Current examples of neuromorphic hardware include Intel's Loihi chip which contains 1024 neurons per core and 128 cores for over 131,000 neurons and 134 million synapses per chip. Each chip can then be integrated onto a board with the chips linked via synapses to create denser neuronal hardware such as Intel's Pohoiki Springs which has 768 Loihi chips for over 100 million neurons [13]. While this is almost three orders of magnitude less than the 86 billion neurons in a human brain, it is a substantial improvement toward biological plausibility.

Examples of state-of-the-art materials to advance hardware components and devices include magnetic tunnel junction (MTJ) and graphene microfabrication. MTJ is a spin-based memristor using magnesium oxide (MgO) to replicate mammalian neurons and synapses as shown in Figure 2.5 [15]. Graphene coupled lasers are being used as component transistors and resistors to fabricate smaller neuromorphic systems [15]. Figure 2.6 shows an implementation of a graphene excitable laser for pattern recognition.

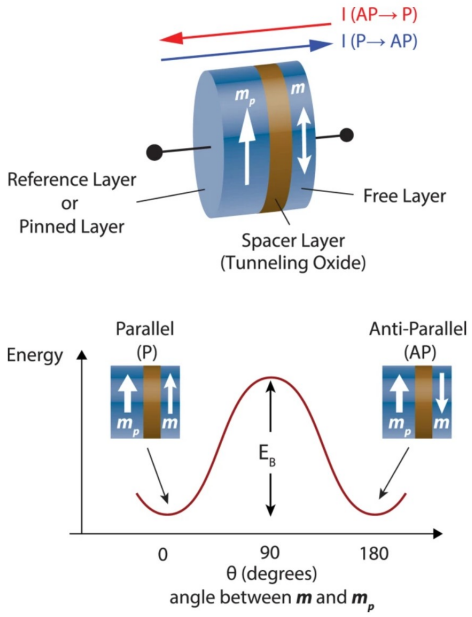

Figure 2.5. A MTJ consists of two magnetic layers sandwiching a spacer layer. While the magnetization direction of the reference layer is pinned, the magnetization of the free layer can be manipulated by an input charge current. The MTJ is characterized by two stable resistance states, namely the parallel (P) and anti-parallel (AP) configuration. The barrier height (E_B) causes the P and AP states of the MTJ to be thermally stable. Source: [16]

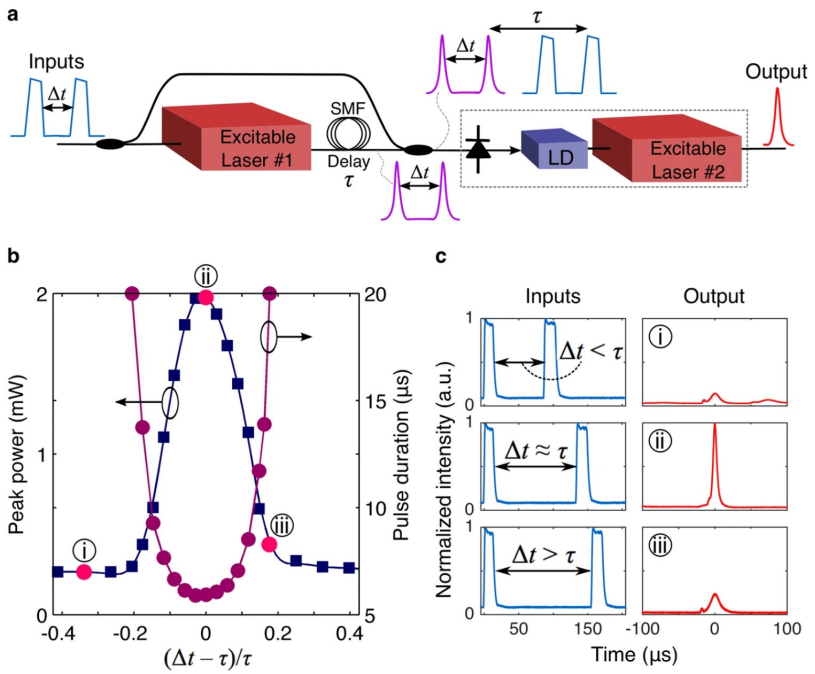

Figure 2.6. (**a**) Simple circuit with two cascaded graphene excitable lasers. (**b**) Measured output pulse peak power and pulse duration as a function of the time interval between the two input pulses. (**c**) Measured input and output waveforms at specific instances: (i) $\Delta t - \tau = -45\mu s$, (ii) $\Delta t \approx \tau = 135\mu s$ and (iii) $\Delta t - \tau = 35\mu s$. The output pulse energy is largest when $\Delta t \approx \tau$ showing the system only reacts to a specific spatiotemporal input pattern. Source: [17]

To fully take advantage of neuromorphic hardware and SNNs, sensory data in the form of spike pulses would be the optimal input, much like how the human eye communicates with the brain. DVSs attempt to more accurately model the human eye to benefit from the efficiencies it has over a traditional frame-based camera. These sensors are known as event-based sensors and exhibit the same power consumption efficiencies as neuromorphic event-driven processing as well as providing higher fidelity and lower latency [18].

Much like neuromorphic hardware is a biologically plausible departure from traditional von Neumann hardware to achieve greater performance in both computational capacity and power efficiency, event-based sensors such as the DVS achieve similar gains compared to traditional frame-based cameras and sensors. These frame-based sensors work by capturing

still photos by converting light energy into electrical energy at the Charged Couple Device and stores each individual frame digitally as a discrete event. These frames are captured in rapid succession, fast enough to create the illusion of fluid movement to the human eyes and brain. For human eyes and brain to perceive still photos in rapid progression as continuous, non-discreet movement, the sensor must capture at approximately 15 frames per second (FPS). Since each frame is being captured at a specified time step, information is lost or delayed between those capture points. Also, each frame is a discrete photo, therefore must be stored into memory even if nothing changed between frames. DVSs overcome these shortcomings through continuous event-based sensing, only sending changes in the environment, and sending those changes instantaneously.

DVSs ability to emulate biological visual pathways allows for higher fidelity output at reduced bandwidth, lower latency for near instantaneous output, continuous sensing, and reduced power requirements [18]–[20]. Current examples of DVSs include the iniVation DAVIS 346, Prophesee Gen 4 CD, and Samsung DVS-Gen4. All these have about three times the dynamic range of a traditional high-end frame-based camera which allows for detecting a greater range of light intensity [18]. This increased range provides finer details which would not have been detected by a traditional camera. The near continuous sensing of the DVS also detects missed events that fall in between the still photos that are captured by a frame-based camera. The DAVIS 346 is able to detect and transmit 12 million events per second with a latency of only 20 µs. Even though greater detail is detected and transmitted, the fact that only events, that is, changes in the environment, are sensed, allows for the DVS to be more power efficient and require less bandwidth to transmit [18].

With new technology there are drawbacks and limitations as well. A DVS's performance can be limited or hindered due to dramatic scene changes where the number of events outweighs the capacity of the DVS detection. Also a mobile application of the sensor could cause performance saturation as well due to the constant changing field-of-view. Large moving object relative to the DVS's field-of-view could also pose challenges in timely and accurate detection.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Design and Methodology

With creating an end-to-end system in mind, the architecture was designed from the ground up with only biologically inspired neuromorphic components. The model is tested via simulation as a proof of concept of building a fully biologically plausible system for target detection, classification, and tracking, performs near current technologies, and benefits from the efficiencies inherent in a biological system. To create and test this hypothesis, the mammalian visual pathway served as the foundation to base the proposed system. The Brian2 simulator was used in lieu of neuromorphic hardware to simulate a SNN. Finally, unsupervised STDP learning in the form of a SNN was used to conduct the pattern recognition while a STICK network was used for the symbolic processing. The novel and nontrivial aspect of this work is to build a symbolic processing SNN, determine the appropriate subset of neurons in the pattern recognition SNN, isolate them, and then connect the targeted network to the desired task specific symbolic STICK network. A commonly utilized image dataset was used to train and evaluate the pattern recognition performance of the system.

3.1 Approach

This work attempts to examine emergent technologies, integrate and apply them in a novel application, and establish a cause and effect relationship to show the benefits and need for further research. The research problem is composed of three main parts. First, this research shows that pattern recognition via unsupervised STDP learning is reproducible and performs near current technologies. Next, it demonstrates that the symbolic processing primitive of counting discrete events is accurate and possible on a STICK network. Finally, it indicates that symbolic and sub-symbolic processing and the integration of the two is achievable with SNNs. The end goal is to provide insights and the foundational work for aiding the development of courses of action for future research and applications.

To provide concrete evidence to support the hypotheses, quantitative experimental data is collected. Components are tested and measured independently to first show component effectiveness. After integration, measurements in an experimental environment are taken to

show the efficacy of the proposed system in detecting and classifying objects. Performance data is compared to already collected secondary data in the field of current technology to provide metrics for comparison. To control against variation and bias, experimental performance data is gathered by utilizing a common and widely used dataset while only varying the system conducting the target detection and classification. Tracking is not tested and left to future work.

The almost ubiquitous and widely used Modified National Institute of Standards and Technology (MNIST) handwritten digit dataset is used as a control mechanism for this experiment. It was chosen due to being well researched and utilized with a vast amount of quantitative secondary data on pattern recognition available for comparison. The MNIST dataset is composed of 70,000 labeled handwritten digit samples broken into a training set of 60,000 examples and the remaining 10,000 examples as the test set. The training set comprises examples from over 250 writers. All images have been normalized for size and centered. Each image is 28 x 28 pixels in standard 256 grayscale [21].

3.2 Model Simulation

To collect quantitative data for follow on analysis, the tools and procedures are discussed in detail for future reproducibility. The methods are broken down into three major categories. The initial step of pattern recognition through unsupervised STDP learning is discussed, then the symbolic processing via STICK, and finally the integration of the two. The Brian2 simulator running on Python is used to emulate the SNN and the neuromorphic hardware [22]. The pattern recognition and symbolic processing components are individually tested and analyzed. Upon successful results, they are integrated for the final test and evaluation.

3.2.1 Pattern Recognition

Pattern recognition comprises the target detection and classification aspects of the proposed system in question. To accomplish this, Peter Diehl and Matthew Cook's work [2] was reproduced to include all global variables and parameters, and then modified for this application. The base code can be found at: https://github.com/peter-u-diehl/stdp-mnist. The means of operationalizing a human neuron, synapse, learning, and neural network is formalized.

Neuron

The LIF and conductance models are used to model a neuron and synapse, respectively. The membrane voltage of the neuron is formulated as:

$$\tau \frac{dV}{dt} = (E_{\text{rest}} - V) + g_e \, (E_{\text{exc}} - V) + g_i \, (E_{\text{inh}} - V) \tag{3.1}$$

where:

Variable	Definition
τ	time constant
V	voltage in mV
E_{rest}	membrane potential at rest
g_e	conductance excitatory
g_i	conductance inhibitory
E_{exc}	equilibrium potential excitatory
E_{inh}	equilibrium potential inhibitory

Synapse

The conductance model is used for the synapse where an excitatory connection is described by:

$$\tau_{g_e} \frac{dg_e}{dt} = -g_e \tag{3.2}$$

where:

Variable	Definition
τ_{g_e}	time constant for excitatory neuron
g_e	conductance excitatory

An inhibitory synapse is described by:

$$\tau_{g_i} \frac{dg_i}{dt} = -g_i \tag{3.3}$$

where:

Variable	Definition
τ_{g_i}	time constant for inhibitory neuron
g_i	conductance inhibitory

Learning

Learning via unsupervised STDP is formulated by:

$$\Delta w = \eta \left(x_{\text{pre}} - x_{\text{tar}} \right) (w_{\text{max}} - w)^{\mu} \tag{3.4}$$

where:

Variable	Definition
Δw	change in weight
η	learning rate
x_{pre}	presynaptic trace
x_{tar}	presynaptic trace target value
w_{max}	maximum weight
μ	update dependence on previous weight

Neural Network

The SNN architecture is comprised of two layers. First is a 28 x 28 neuron input layer. Second is a 200-neuron processing layer separated into 100 excitatory neurons and 100 inhibitory neurons. Each input neuron is fully connected to all the excitatory neurons in the the second layer, represented as black directional arrows in Figure 3.3. An excitatory neuron is connected to its associated inhibitory neuron in an one-to-one directional fashion, represented as blue directional arrows in Figure 3.3. The inhibitory neuron is then connected to all the other excitatory neurons, expect for the source excitatory neuron, establishing the one-to-all minus one connection, represented as red directional arrows in Figure 3.3. These

inhibitory to excitatory neuron connections represent the lateral inhibition feedback to the excitatory neurons.

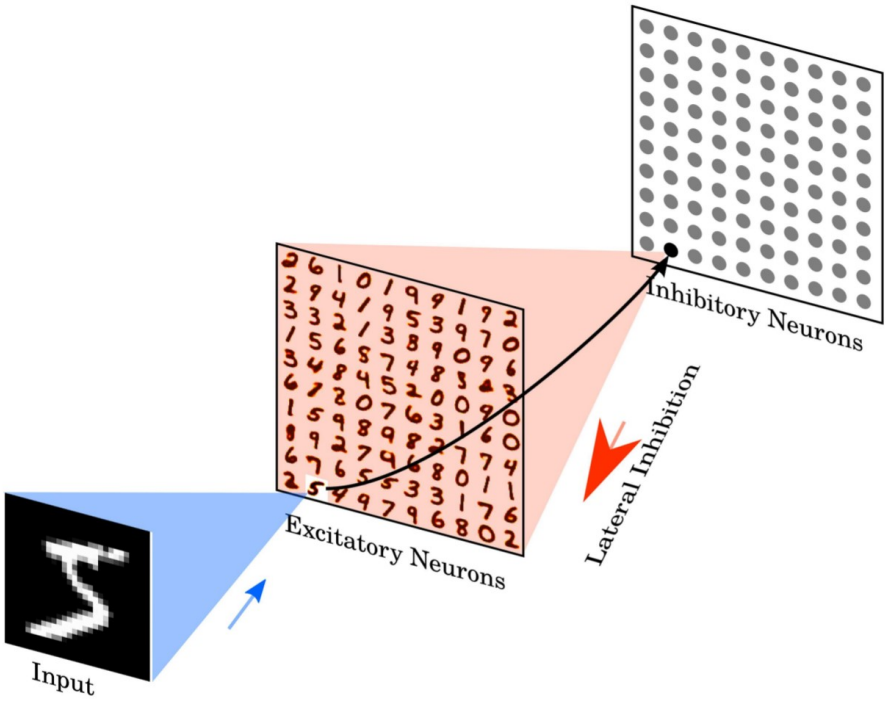

Figure 3.1. SNN architecture for unsupervised STDP learning showing the feedforward signal from input layer through the excitatory and inhibitory neurons and the resultant lateral inhibition feedback. Source: [2]

The MNIST training dataset is fed into the SNN as a Poisson-distributed spike-train with the rate of firing equaling the pixel intensity divided by four [2], [3]. Each image is presented to the network for 350 ms with a 150 ms gap to allow for the network to reset/decay back to a resting state. Once the unsupervised training is complete the learning rate is set to zero and the spiking thresholds for all neurons are fixed. The training set is fed through the network and each neuron is assigned the class which elicited the highest response [2]. Of note, labeled data is being used here. Classification accuracy is determined by presenting the test set, measuring the average response rate per class, and making a prediction based upon the class that has the highest firing rate.

A 400-neuron excitatory layer network was adapted and created for this research. After un-

supervised training is complete, the entire MNIST test set is used to generate a classification accuracy for comparison to Peter Diehl and Matthew Cook's original work.

All neurons, and their associated pathways, of the same label class is then connected to a *detector* neuron, which for this work was simulated. The *detector* neuron fires based upon presentation of that specific labeled stimulus. The *detector* neuron is then connected to the symbolic processing portion of the architecture, as represented in Figure 3.3.

3.2.2 Symbolic Processing

Symbolic processing comprises the higher level processing required to conduct quantitative analysis on objects detected and classified by the pattern recognition portion of the system. Xavier Lagorce and Ryad Benosman's work [6] on STICK was reproduced while maintaining all global variables and parameters and then modified to accomplish this task. His research shows the use of biologically plausible neuronal units in combination with precise spike timings can be the foundation of a Turing complete general purpose and compact computation system [6].

Neuron

A non-leaking neuronal model is used to create the computational unit of the STICK system:

$$\begin{cases} \tau_m \cdot \frac{\mathrm{d}V}{dt} = g_e + gate \cdot g_f \\ \frac{\mathrm{d}g_e}{dt} = 0 \\ \tau_f \cdot \frac{\mathrm{d}g_f}{dt} = -g_f \end{cases} \tag{3.5}$$

where:

Variable	Definition
τ_m	membrane time constant, set to 100 sec
V	voltage in mV
g_e	constant input current for synaptic event
$gate$	synapse gate signal
g_f	input synaptic event with exponential dynamics
τ_f	exponential time constant, set to 10 msec

Synapses

Four types of synapses are utilized in the STICK framework such that w = weight of the synapse:

- V-synapse: input directly alters the membrane potential voltage

$$V \leftarrow V + w$$

- g_e-synapse: input directly alters the constant input current

$$g_e \leftarrow g_e + w$$

- g_f-synapse: input directly alters the exponential input current

$$g_f \leftarrow g_f + w$$

- *gate*-synapse: gates the exponential input current where:

$$w = 1 \rightarrow gate = 1$$

$$w = -1 \rightarrow gate = 0$$

The volatile memory network developed in STICK was adapted for this application to create an accumulator network which is able to store a single scalar value and recall it based upon the time interval stored in the *acc* neuron [23]. Figure 3.2 shows the state diagram for the accumulator STICK network. Neurons that take external inputs are blue, internal network neurons are white, and output neurons are red. Only two types of synapses are required for this work: solid lines denote V-synapses while the dotted lines represent g_e-synapses.

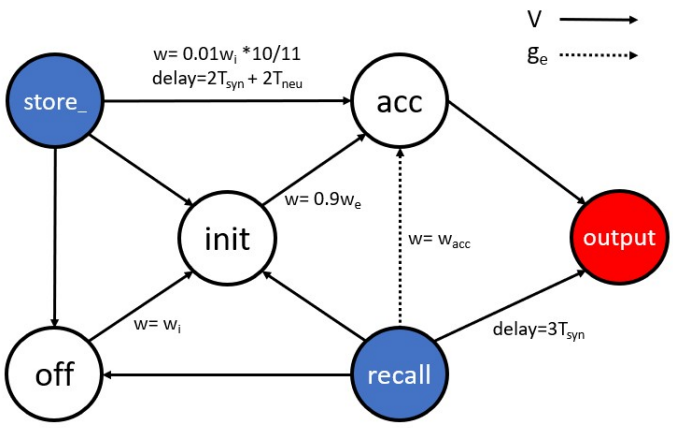

Figure 3.2. Self-initializing accumulator STICK network providing self-contained and independent symbolic processing.

To obtain the value stored within the accumulator network, a single spike is sent to the *recall* neuron. This will then send an input current change via a g_e-synapse to the *acc* neuron and a spike to the *output* neuron with a $3T_{syn}$ delay to cause the first *output* neuron spike to be triggered. The delay is to account for signal transmission time to provide an accurate Δt between the first and second *output* neuron spikes. Based upon the state of the membrane potential of the *acc* neuron, the input current change from the *recall* neuron's g_e-synapse will cause a constant rate of increase in the *acc* neuron's membrane potential until it reaches threshold. Therefore the membrane potential of the *acc* neuron will determine the time it takes for the *acc* neuron to spike and cause the second spike from the *output* neuron at the associated time delay from its first spike. This time delay, Δt, decodes to the value stored in the *acc* neuron based upon its membrane potential.

After successful functional testing of the accumulator network, it is then evaluated to determine the bounds of the network's capability to determine the minimum and maximum values stored and time resolution between events. Data will be represented through chronograms showing spikes and voltage changes over time per neuron.

3.2.3 Symbolic and Sub-symbolic Integration

Upon successful unsupervised training of the pattern recognition SNN to detect and classify written digits of the MNIST dataset and the accumulator network showing expected behavior of accurately counting discrete events, the two networks are then integrated to test and evaluate the efficacy of the joined system. To accomplish this, a *detector* neuron is added to the accumulator network to act as a bridge between the pattern recognition SNN and the accumulator network. The *detector* neuron is fully connected to all of the excitatory neurons of a single class from the pattern recognition network. An accumulator network for the digit zero has its own *detector* neuron which is then connected to the neurons in the pattern recognition excitatory layer that detect the digit zero. This is then repeated for all the classes of the pattern recognition network, each having its own distinct and separate *detector* neuron and accumulator network.

The signal from a 6400-neuron excitatory layer pattern recognition SNN is simulated based upon the findings in Peter Diehl and Matthew Cook's work which shows that approximately 17 neurons fire when a digit is shown, about 16 from in-class neurons and about 1 from out-of-class neurons [2]. Spike inputs are randomly generated to simulate within-class and out-of-class stimuli. For an in-class stimulus, excitatory neuron spikes are simulated with probability $P1 = 16/640$; for an out-of-class stimulus, neuron spike probability is $P2 = 1/5760$. These spikes are presented to 640 excitatory neurons signaling them to fire or not and send a spike to their associated *detector* neuron. If the *detector* neuron's threshold is met or exceeded, then the *detector* neuron sends a spike to the *store* neuron of the accumulator network, thus registering an event to be stored.

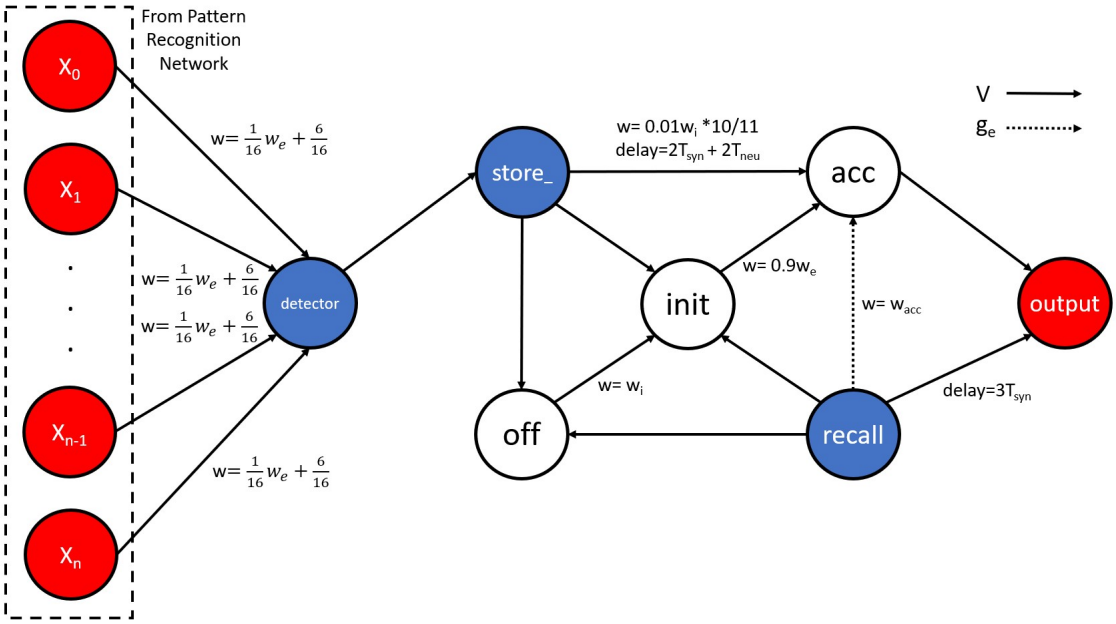

Figure 3.3. Connection between STDP pattern recognition SNN and self-initializing accumulator STICK network.

The synaptic weights between the 640 in-class neurons of the excitatory layer pattern recognition network and their detector neuron have the same weight over a V-synapse utilizing Equation 3.6. w_c is then adjusted to produce a 95% success rate. The single synapse between the *detector* and *store* neurons is a V-synapse with standard weight. Batch tests of 50, 100, 200, 500, 1000, and 2000 simulated digit presentation samples are preformed to calculate the average counting accuracy rates for both in-class and out-of-class digits and to show if any variability is present based upon batch size.

$$w_e = \frac{V_t / n_c}{P} + w_c \tag{3.6}$$

where:

Variable	Definition
w_e	weight of excitatory connection
V_t	threshold voltage in mV
n_c	number of in-class neurons
P	spiking probability
w_c	adjustable synaptic weight constant

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results

Due to the limited availability of neuromorphic hardware, a simulator was required to imitate the neuromorphic hardware environment on a general purpose computer. The Brian2 simulator provides the ability to run software based SNNs on traditional general purpose computers to experiment and conduct STDP learning [22]. The power and computing efficiencies are not realized due to the simulated environment and the von Neumann bottleneck constrained hardware.

Brian2 is a Python-based package that allows for formulaic definitions of neuromorphic dynamics and morphology. This allows the user to define specific neuron models, synaptic properties, and global parameters for tailored SNN applications such as STDP learning. Once these models are tested and validated through Brian2, they can then be applied to neuromorphic hardware for further test and evaluation to fully benefit from utilizing SNNs on neuromorphic hardware.

4.1 Pattern Recognition Network

A 400-neuron excitatory layer STDP network was built and tested to reproduce Peter Diehl and Matthew Cook's work [2]. The code base was modified to run on Python 3 and Brian2. Figure 4.1 (a) shows the resultant trained weights as a color gradient for the 400 neurons as a 2D receptive field. A classification accuracy of 91.43 was achieved with 9,143 correctly classified and 857 incorrectly classified MNIST digits which are shown in Figure 4.1 (b). This surpassed the 87.0 classification accuracy for a 400-neuron excitatory layer network reported in Peter Diehl and Matthew Cook's original work [2]. Two possible reasons for the improved performance may be attributed to optimizations in the Brain2 simulator over the Brain1 version used in the original work. First, in Brain1, some objects internal state could change when ran for the first time in comparison to Brain2 where the internal state of an object does not change and restores all needed variables and data structures for every run [24]. Second, the numerical integration method choice was left to Brain1 to decide in the original implementation whereas in Brain2, the Euler integration was used.

After training/learning has occurred, which took approximately 5 days to complete on a NVIDIA DGX Station (without the use of GPU acceleration), the neurons in the excitatory layer are tuned to spike for a certain class of digit based upon the pattern/weights of its receptive field. For the 400-neuron implementation reproduced in this work, approximately 40 neurons have learned to recognize a specific digit. Using the probabilities for a 6400-neuron excitatory layer STDP network presented in Peter Diehl and Matthew Cook's work, that would equate to only one neuron firing out of the 40 when an in-class stimulus is presented and a subthreshold response from out-of-class neurons. Analysis of the output from the test phase of the experiment showed that approximately seven neurons would spike out of the 400 excitatory neurons when a stimulus was presented. This suggests a non-linear relationship between the size of the excitatory neuron layer and the probability of a spike from a population of same class neurons being presented an in-class stimulus.

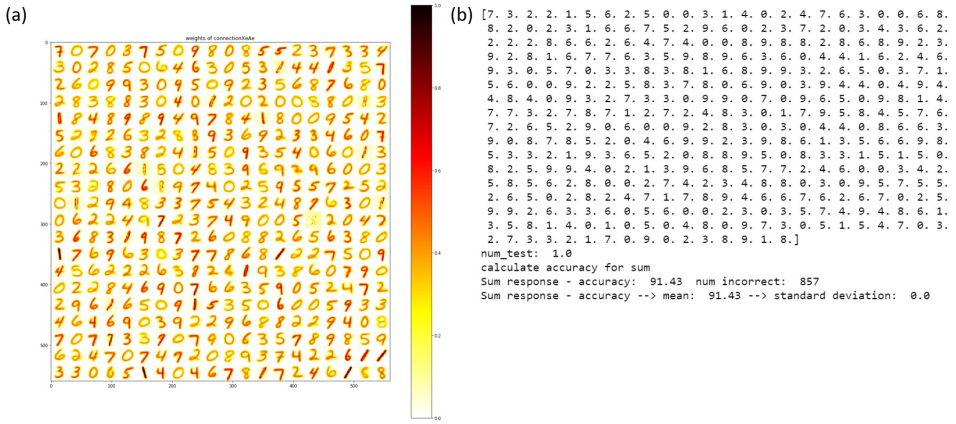

Figure 4.1. Results from 400-neuron excitatory layer STDP pattern recognition SNN. (**a**) 2D receptive fields of the 400-neuron excitatory layer arranged in a 20 x 20 matrix. Each digit represents a single neuron in the excitatory layer, and each pixel of the digit is a color representation of the synaptic weight of an input neuron to the excitatory neuron. There are 784 (28 x 28) input neurons per excitatory neuron, each representing a pixel of the MNIST image. (**b**) The raw output of the adapted code base showing the true labels of the 857 incorrectly classified digits and classification accuracy rate.

Figure 4.2 shows the normalized spiking rate for all 400 neurons in the excitatory layer when presented the set of in-class stimuli from the MNIST test dataset. The green and red lines signify a rate of 0.6 and 0.7, respectively. The rate equates to how often that particular

neuron fired when presented all of the images in a single class of the MNIST test dataset. For all 10 classes, at a rate of 0.6, six classes had only three neurons spike at or above the rate, which was also the minimum number of neurons to spike. The maximum number of neurons to spike at or above the rate of 0.6 was 10 neurons for the class zero. At a threshold rate of 0.7, three classes had only one neuron spike. The minimum and maximum number of neurons to spike at or above the threshold rate of 0.7 was zero and seven, respectively. Table 4.1 shows these results as well as the details of the other classes.

Figure 4.3 shows the histograms for the normalized spiking rates distribution for the 400-neuron excitatory layer STDP pattern recognition network broken down by individual classes. They show the sparsity of spiking rates for all classes. The first bin represents spiking rates from zero through 0.033. Class one had the maximum bin size of 373 neurons while class three had the minimum bin size of 319 neurons. Utilizing the results shown in Figures 4.2, 4.3, and Table 4.1 will assist in determining the neuronal class populations by segregating those neurons which have the highest rate of spiking per class based upon a threshold that can be determined experimentally.

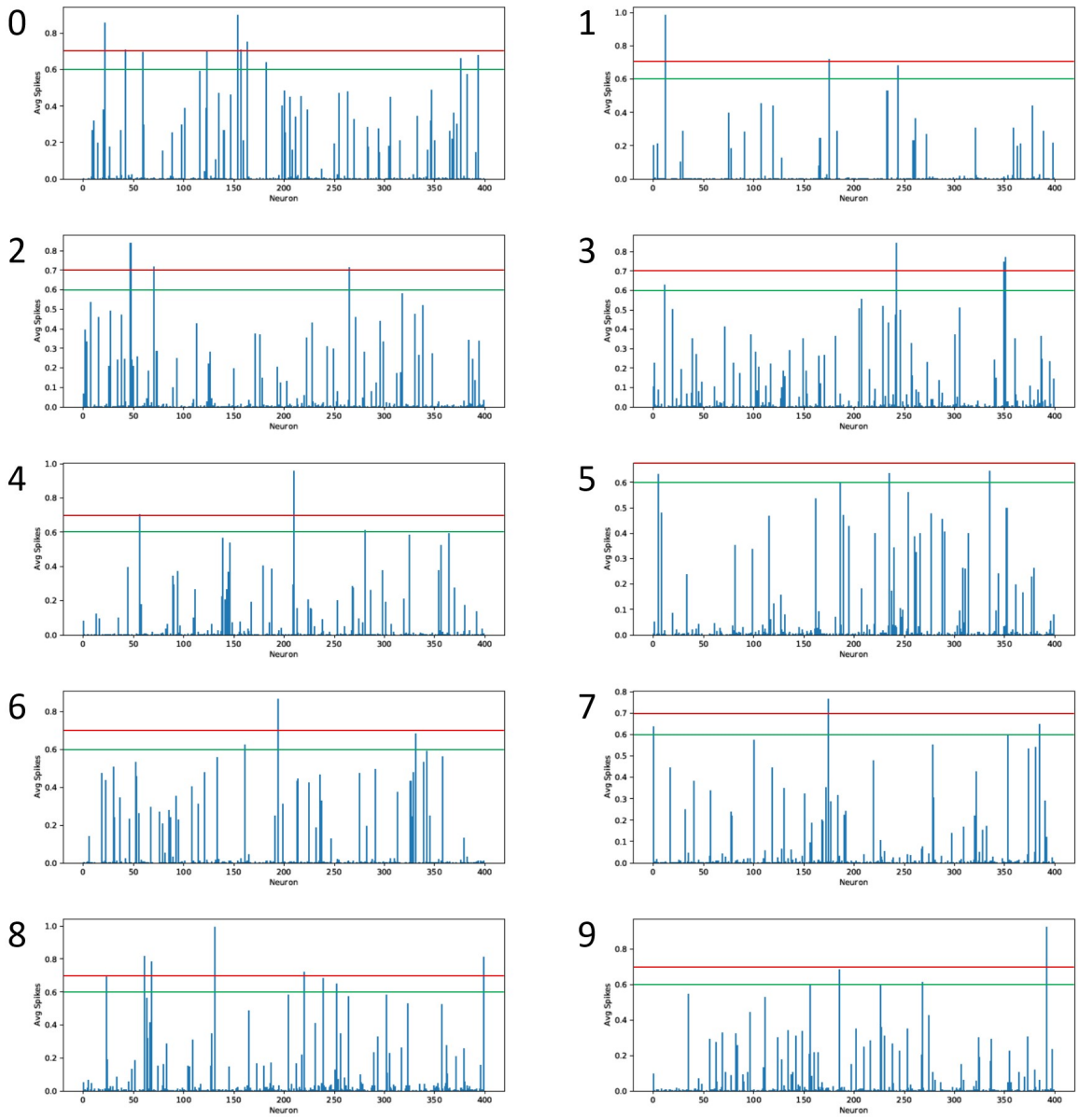

Figure 4.2. STDP pattern recognition neuron spiking averages by class showing the normalized average spiking rate versus which excitatory neuron in a 400-neuron excitatory layer. Green line showing a rate of 0.6, red line, 0.7.

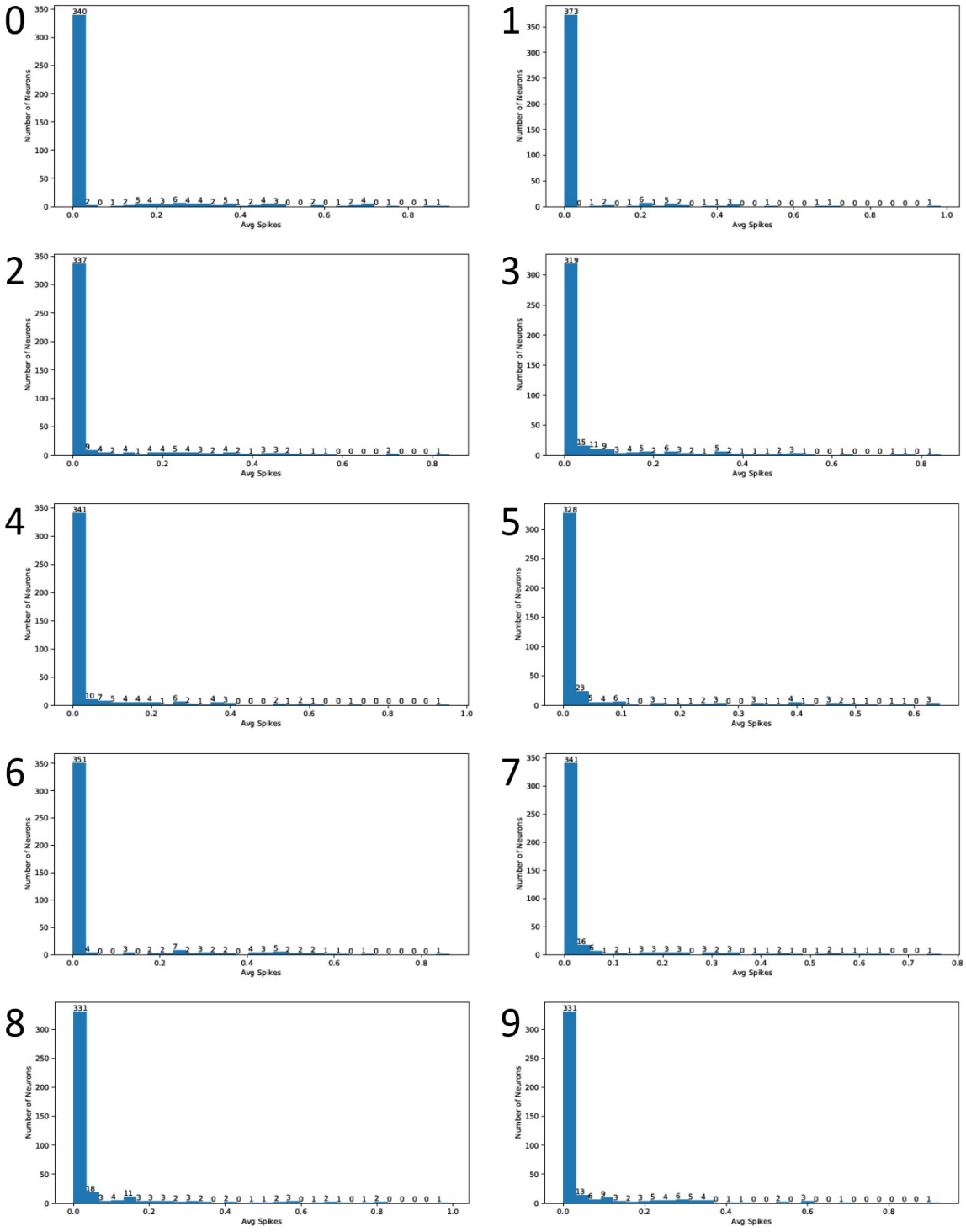

Figure 4.3. STDP pattern recognition neuron spiking histogram by class showing the number of neurons to spike versus the normalized averages in a 400-neuron excitatory layer. Bin counts are labeled on top.

Table 4.1. Number of excitatory layer neurons in STDP pattern recognition network to spike at or above 0.6 and 0.7 rate threshold by class. Rates are normalized based upon all presentations of in-class stimuli.

Rate \ Class	0	1	2	3	4	5	6	7	8	9
0.6	10	3	3	4	3	3	3	3	8	5
0.7	7	2	3	3	2	0	1	1	4	1

4.2 Symbolic Processing Network

A test harness was created to validate the efficacy of the self-initializing accumulator network to count discrete spike events which is a symbolic processing primitive. Input spikes are blue, internal network spikes are black, and output spikes are red. The green lines represent the voltage change of the neuron. Test results show that the accumulator network is able to utilize spiking time delays to count from zero through 100, inclusively, with 100% accuracy after 2000 iterations of randomly generated spike counts at randomly generated intervals. The significance of this is that a numerical value can be encoded into a time delay between two neuronal spikes. All instances have a recall initiated at 200 ms and a run-time of 314 ms. Figure 4.4 shows the chronograms for a single test of zero, one, and five spikes inputted into the *store* neuron, respectively.

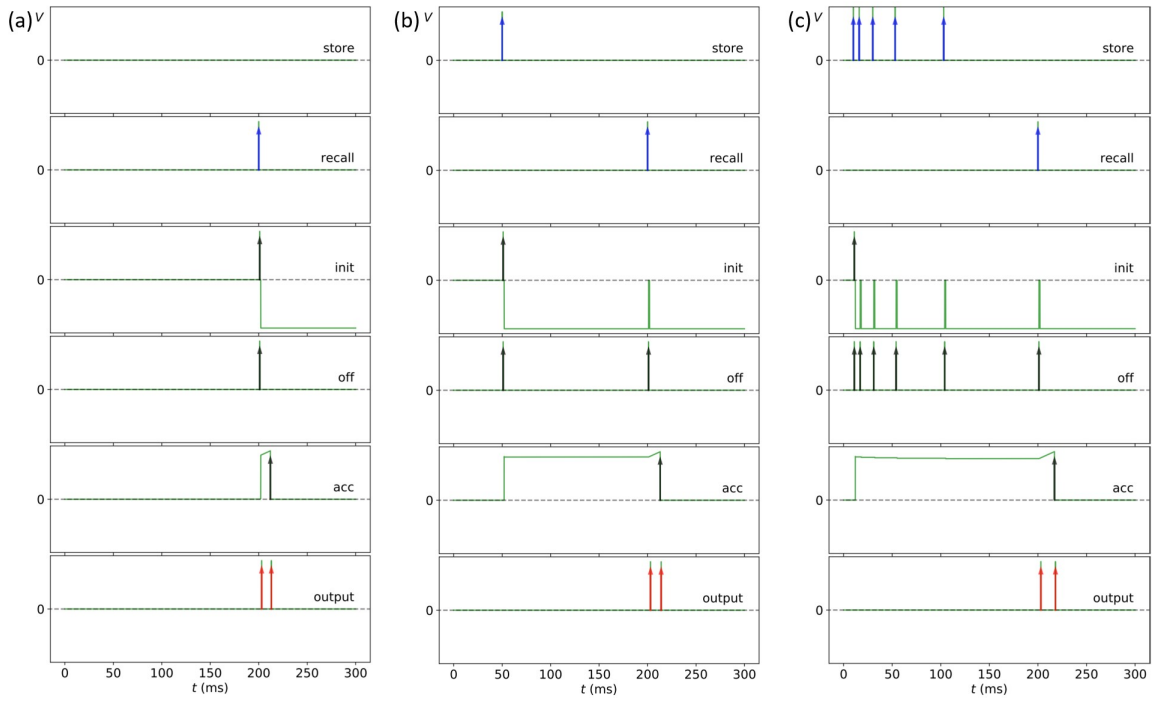

Figure 4.4. Self-initializing accumulator STICK network for symbolic processing of zero, one, and five spikes. (**a**) Chronogram showing neuronal voltage changes over time when no input spike is presented to the network. Notice the *init* neuron self-initializing the *acc* neuron when *recall* is activated. (**b**) Chronogram showing neuronal voltage changes over time when one input spike is presented to the network. Notice the *init* neuron self-initializing the *acc* neuron when *store* neuron is activated. (**c**) Chronogram showing neuronal voltage changes over time when five input spikes are presented to the network.

Looking at the *output* neuron, Figure 4.4 (a) shows a 10 ms difference between the first and second *output* neuron spikes which decodes for zero or a raw value of 0.00. Figure 4.4 (b) shows a 11 ms difference between the first and second *output* neuron spikes which decodes for one or a raw value of 0.01. Finally, Figure 4.4 (c) shows a 15 ms difference between the first and second *output* neuron spikes which decodes for five or a raw value of 0.05.

In its current form, the accumulator network can count/store values from zero to 100 utilizing biologically plausible neurodynamic values. This network is able to "self-initialize" by setting the *acc* neuron's membrane potential to $V_{threshold}$ - 10 mV, which decodes to

the value zero, no matter where the first input signal originates. Upon the first use of the accumulator network, if the first input is to the *recall* neuron, the network will self-initialize and output two spikes with a Δt that decodes for the value zero. If the first input to the network is to the *store* neuron, the *init* neuron will set the *acc* neuron to the membrane potential that decodes for zero, and then decrement the membrane potential by 0.1 mV which decodes for the value one. Each additional spike to the *store* neuron will decrement the *acc* neuron's membrane potential by 0.1 mv. Based upon how many spikes are received at the *store* neuron, an equivalent decrementation of the *acc* neuron's membrane potential will occur, therefore, accurately storing the desired value inputted to the network via the time it will take to induce a spike in the *acc* neuron.

4.2.1 Minimum Time Resolution

The minimum time resolution is the smallest interval with which the accumulator network can count two unique spike events. This value is necessary to determine the minimum spacing between two spikes so that predictable behavior is maintained within the network, specifically the accurate counting of discrete spikes. This will also influence the maximum value that is stored in the accumulator network which is detailed in the next section.

Results show the minimum time resolution to be greater than 1.0 ms. Figure 4.5 (a) shows two input spikes presented to the network at time 10.0 ms and 11.1 ms. All expected behavior is produced with two *output* neuron spikes spaced 12 ms apart which decodes for two or a raw value of 0.02. Figure 4.5 (b) shows two input spikes with 1.0 ms spacing causing an immediate and erroneous *output* neuron spike. The *recall* neuron spike at 200 ms causes the second *output* neuron spike for a total time difference of 187 ms between the first and second *output* neuron spikes. This causes an error due to being greater than T_{max} of 110 ms.

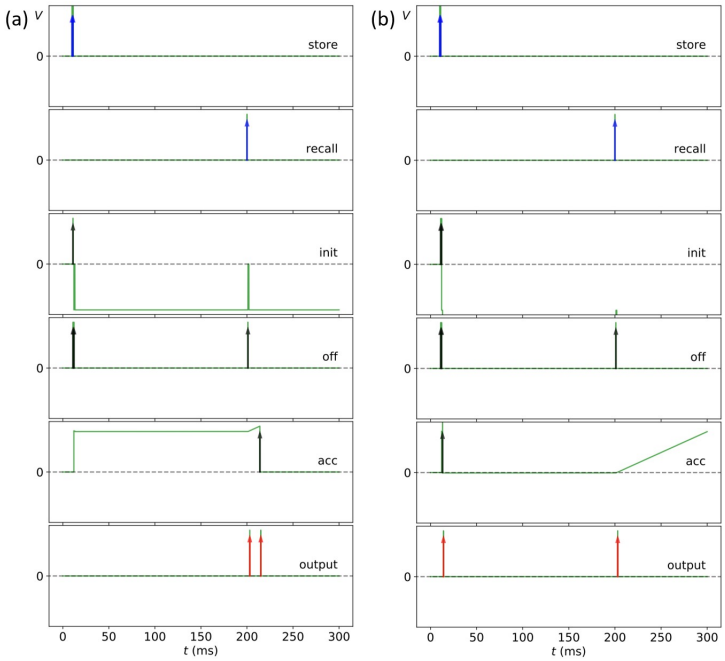

Figure 4.5. Self-initializing accumulator STICK with two inputs to *store* neuron at 1.1 ms and 1.0 ms spacing. Due to the time scale, the two spikes appear as one thicker arrow. (**a**) Chronogram showing neuronal voltage changes over time when two input spikes are presented to the network at time 10.0 ms and 11.1 ms and the resultant 12ms difference between the first and second *output* neuron spikes which decodes for 2. (**b**) Chronogram showing neuronal voltage changes over time when two input spikes are presented to the network at time 10.0 ms and 11.0 ms, where T_{min} = 1.0 ms. Notice an *output* neuron spike initiated at time 12 ms. which results in a 187 ms difference between the first and second *output* neuron spikes, far above T_{max} = 110 ms.

4.2.2 Maximum Counting Size

While holding all global parameters fixed to values that are congruent to mammalian neurodynamics, the maximum incrementing range was found to be from zero through 100 discrete events. This value is important for determining the base or radix for counting and storing values for future implementations and architectures utilizing this solution. With only six neurons, a base 100, or centesimal base system, can be implemented for storing and retrieving a count of numbers. Figure 4.6 shows 100 input spikes at 2 ms spacing and a Δt

of 110 ms between the first and second *output* neuron spikes which decodes for 100 or a raw value of 1.0. Figure 4.7 shows a single input spike at t_{recall} - 1.0 ms (at time 199.0 ms), which causes an error due to being within 1.0 ms of the recall spike.

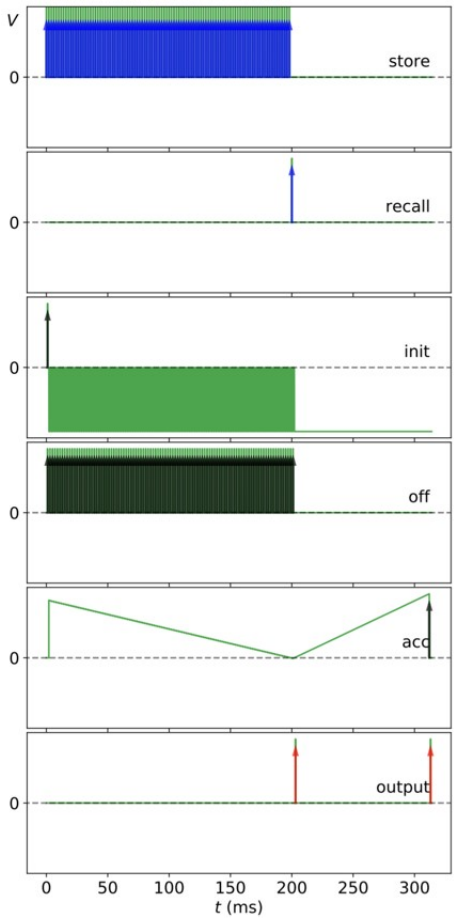

Figure 4.6. Self-initializing accumulator STICK chronogram showing neuronal voltage changes over time when 100 input spikes are presented to the *store* neuron. Note that the *acc* neuron has returned to its resting potential of zero mV. There is a 110 ms difference, which is T_{max}, between the first and second *output* neuron spikes that decodes for 100.

Figures 4.8 shows a single input spike at time zero and t_{recall} - 1.1 ms (at time 198.9 ms), respectively, which is the maximum window of time that a spike can be accurately detected by the network if the *recall* neuron is activated at time 200 ms. Recall activation was set to 200 ms to ensure the *acc* neuron did not hyperpolarize beyond its resting membrane

potential and to minimize the time between the two *output* neuron spikes so that human noticeable delays would not be apparent. Using only time spacing increments of 1 ms, a 2 ms separation is needed since the minimum time resolution must be greater than 1.0 ms, as shown in Section 4.2.2. Therefore, 100 events can be stored in the *acc* neuron plus one for the null state.

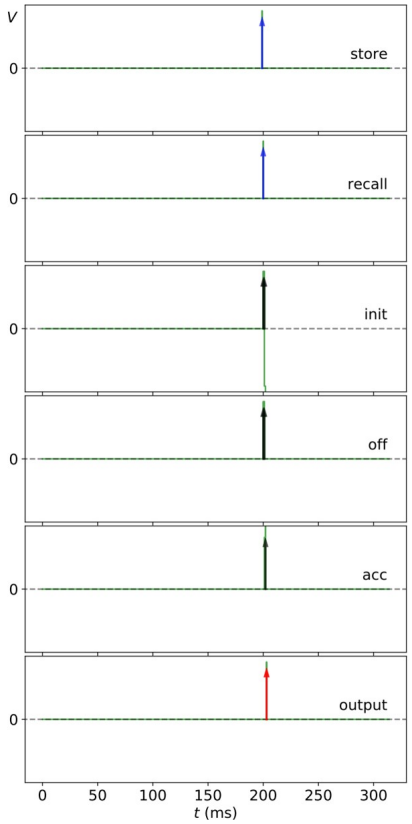

Figure 4.7. Self-initializing accumulator STICK chronogram showing neuronal voltage changes over time when one input spike is presented to the *store* neuron at t_{recall} - 1.0 ms which is 199.0 ms. Notice the erroneous behavior with the lack of two discrete *output* neuron spikes and the thicker *init* spike representing two *init* neuron spikes.

Two departures from human derived neurodynamic values and the resultant performance changes can be inferred from analysis of the used models. First, if the *acc* neuron's firing threshold, V_t, was doubled from 10 to 20 mV and all other values held the same, this would provide twice the range of possible values stored in the *acc* neuron. A drawback would

then be needing 400 ms to store all those values. A possible solution to that would be the second departure from biological values by decreasing the time for synaptic transmission, T_{syn}, from 1 ms to 0.5 ms. If integer values are still used for minimum time resolution, then 1 ms spacing can be used, therefore reducing the time required to store the 200 events back down to 200 ms.

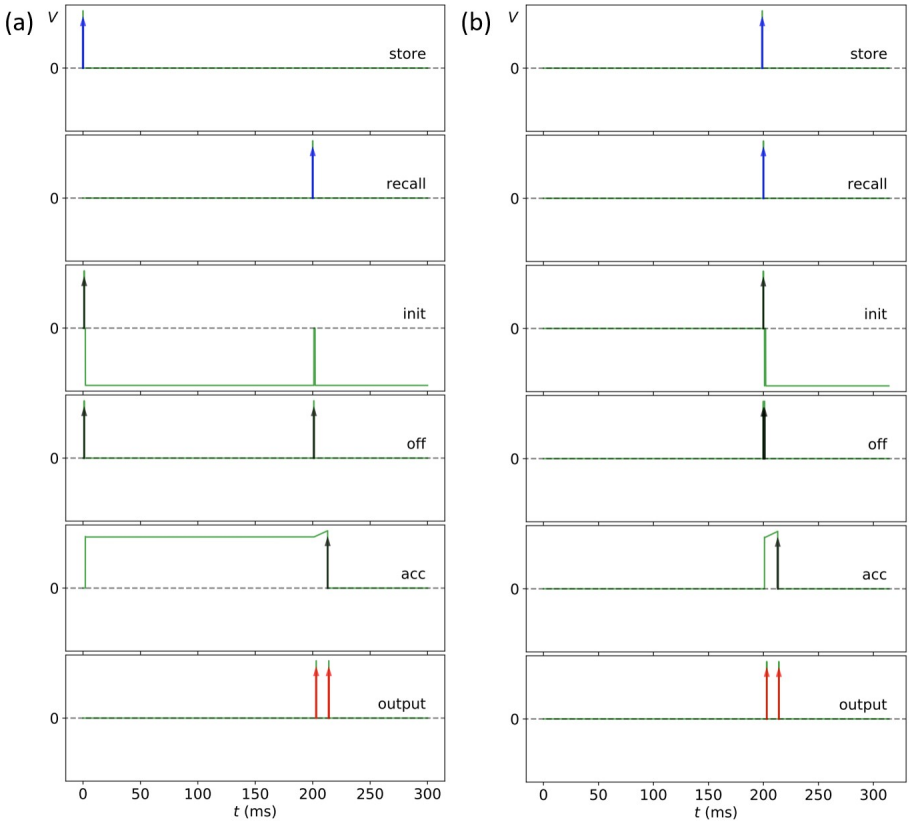

Figure 4.8. Self-initializing accumulator STICK with one input to *store* neuron at time zero and at t_{recall} - 1.1 ms. (**a**) Chronogram showing neuronal voltage changes over time when one input spike is presented to the network at time zero. (**b**) Chronogram showing neuronal voltage changes over time when one input spike is presented to the network at time 198.9 ms.

4.3 Symbolic and Sub-symbolic Processing Simulation

The results of this experimentation exhibited that simulating the pipeline from the pattern recognition output through the storing and recall of the symbolic processing network is

possible. 640 neurons were created to represent the excitatory neuron population for a single class within a 6400-neuron excitatory layer pattern recognition network. A many-to-one connection was created between these 640 neurons to their single dedicated *detector* neuron via a weighted V-synapse. The spiking rate of these excitatory neurons were mimicked by a randomly generated Bernoulli distribution and subsequently fed into the network as an array of zeros and ones where a one would represent a spike. When enough spikes reached the *detector* neuron, it would then send a single spike to the *store* neuron of the accumulator network which in turn would store a single event into the *acc* neuron by decrementing its membrane potential by 0.1 mV. A spike would then be sent to the *recall* neuron to verify the correct value stored in the accumulator network.

Table 4.2 shows the results of simulating the output of the pattern recognition SNN for a single class connected via a *detector* neuron to its associated class accumulator SNN as depicted in Figure 3.3. The simulation created the output spikes of a pattern recognition network of a single class based upon presentation of an in-class stimulus. The *detector* neuron's spiking threshold, V_t, was kept constant while the synaptic weight constant, w_c, was varied until a 95% true positive accuracy was achieved which we considered sufficient for this application. The simulation for out-of-class stimuli showed 100% true negative accuracy for all variations of synaptic weight constant and sample sizes. Even doubling the P2 probability from Section 3.2.3 to 2/5760 showed no change to the 100% true negative accuracy. Over a 40 times increase in probability was needed to cause a change in the true negative rate where $w_c = 6/16$. Larger weight constants for both true positive and true negative rates were not tested. Possible adverse effects to true negative rates could be expected. Figure 4.9 shows the plot for true positive rate versus increasing w_c.

Table 4.2. STDP pattern recognition and STICK accumulator symbolic processing integration simulation for spikes originating from the appropriate digit identification class neurons. The values represent the number of accurately counted spikes by the symbolic processing network based upon the simulated output of the pattern recognition SNN being presented an in-class stimulus.

Sample Size / w_c	50	100	200	500	1000	2000	Avg. Rate
0	26	46	102	246	507	1018	50.52
1/16	32	65	132	308	622	1251	63.48
2/16	39	70	149	367	724	1451	72.73
5/16	46	91	186	458	932	1857	92.73
6/16	47	97	191	482	957	1921	95.97

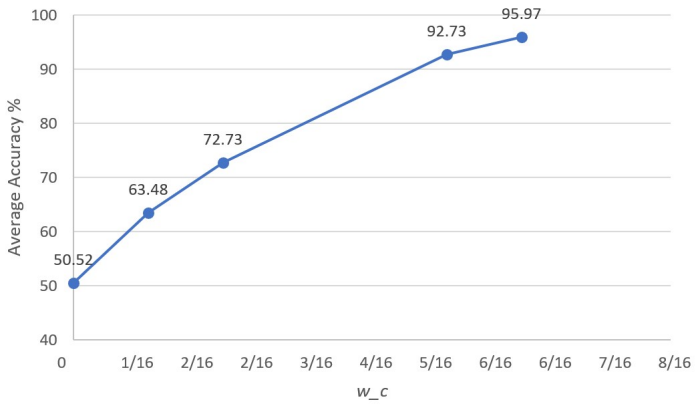

Figure 4.9. STDP pattern recognition and STICK accumulator symbolic processing integration simulation scatter plot showing true positive accuracy rate verses w_c.

CHAPTER 5:
Discussion

This work shows that sub-symbolic pattern recognition, symbolic counting, and the integration of the two are possible on a biologically inspired SNN. Each major portion of the research will be discussed, to include challenges and future suggested work on the topic.

5.1 Pattern Recognition Network

The results show that Peter Diehl and Matthew Cook's original work on unsupervised STDP learning via a SNN is portable, both to Python 3 and Brian2, with, in this particular case, better classification performance. Only 100 and 400-neuron excitatory layer networks were recreated, adapted, and tested leaving the 1600 and 6400-neuron excitatory layer networks to future works. With a classification accuracy of 91.43%, this is still far off from human performance as well as current state-of-the-art CNNs, as seen in Table 5.1. Nonetheless, this shows that unsupervised pattern recognition learning is possible via third generation SNNs with room to improve over the simple two layer network through the proposed use of skip-over connections, increased specialized layers, and localized lateral inhibition.

Table 5.1. MNIST classification accuracy benchmarks compared to re-creating a 400-neuron excitatory STDP pattern recognition SNN. Adapted from [2], [25], [26].

Model	Error	Accuracy
Human [25]	0.2	99.8
CNN [26]	1.68	98.32
ResNet [26]	0.84	99.16
DenseNet [26]	0.63	99.37
uSTDP(6400 neuron) [2]	5.0	95.0
uSTDP(400 neuron) [2]	13.0	87.0
This work(400 neuron)	8.57	91.43

5.2 Symbolic Processing Network

Utilizing a Brian2 implementation of STICK, the results presented show the capability of creating an independent SNN symbolic processing network to count discrete spikes as an input [6], [23]. The self-initializing accumulator network only requires six neurons to provide a plug-and-play SNN to conduct the symbolic processing primitive of counting. Comparing this compact network to a traditional von Neumann computer architecture of a CPU, memory, and secondary storage, a few benefits and drawbacks can be postulated. Possible benefits arise from the fact that computation, memory, and to a limited extent, secondary storage, is housed on the same component. In this work, the *acc* neuron performs all three above functions in a single neuron. The network can be further optimized by removing the *off* and *output* neurons but was left in this configuration for understandability and logical intra and inter network flow.

In the traditional view of computer operations being broken down in the four steps of fetch, decode, execute, and store, the first two steps are no longer necessary in this network. Instruction sets and data no longer need to be fetched and decoded prior to the execution of the instruction. Each neuron and synapse intrinsically hold all information needed to conduct its designed function. The execution step is just letting the neurons of the network fire based upon their neurodynamics, synaptic connections, and weights. The store step is the resultant membrane potential changes in the *acc* neuron based upon, and are byproducts of, the execution step. In more complex networks, synaptic weight changes would also be included. This then also means data does not have to go back out to memory, which is a costly time procedure, or the subsequent write to secondary memory. Other possible benefits include the non-volatileness of the values stored in the membrane potential and synaptic weights and the fact that memory no longer needs an address. Possible drawbacks include the mammalian neurodynamic values are in the range of *ms* vice current computer timings in the *ns*, or six orders of magnitude difference and the amount of neurons required to create a Turing complete solution.

The network interface was intended to be simple and scalable. It takes as an input zero to N spikes where N equals the maximum value needed to be stored by the network. The spike can originate from any neuron connected to the network's *store* neuron via a V-synapse as shown in Figure 3.3. More intricate connections can be devised to model more complex input configurations and behaviors such as integrating inhibitory signals into the LIF construct.

5.3 Symbolic and Sub-symbolic Processing Simulation

The simulation results show that connecting the unsupervised STDP pattern recognition SNN to the symbolic processing SNN via an intermediary *detector* neuron is a feasible solution for integrating sub-symbolic and symbolic processing on a common neuromorphic architecture. Experimentation shows that an additional weight constant, w_c, is required to obtain expected results. See Equation 3.6, Figure 3.3, and Table 4.2 for explanations and results.

An additional w_c equal to 6/16 is required for each synapse from the pattern recognition excitatory layer to the *detector* neuron. This equates to six additional excitatory spikes, or 37.5% more spikes over the average 16 spikes produced per presentation of an in-class stimulus, to induce the appropriate *detector* neuron response 95.97% of the time. This additional weight constant did not affect the out-of-class response. 100% true negative responses were recorded for the entire experimentation set for out-of-class stimuli.

This work is a departure from Peter Diehl and Matthew Cook's STDP MNIST work in two major ways. First, it treats each population of neurons in a given class separately. The neurons in a class are segregated from the other neurons in the excitatory layer and then connected to a specific and single *detector* neuron in a hard voting scheme. If enough spikes trigger the *detector* neuron to emit its own spike, then the associated accumulator network will store an event. Secondly, this work handles the learning and labeling of the classes all within the SNN compared to manually labeling the neuron's class between the training and test phases based upon the highest response to a given class when presented the training set [2].

5.4 Challenges

Through the conceptualizing, background exploration, design, and execution of this research, many challenges were discovered. Some surmountable, while others were bypassed due to limits on time, scope, and access to resources. The bypassed challenges are broken into their three major categories below.

5.4.1 Sub-symbolic Pattern Recognition

The challenges with the sub-symbolic pattern recognition portion of the SNN were implementing a truly unsupervised learning scheme and an autonomous means of segregating the excitatory neurons and pathways by class. The unsupervised STDP learning is much like showing a child the number "1" and telling them it is "one." The supervisory signal is the parent providing a verbal label to the pattern that the child sees. A possible means to achieve a truly unsupervised learning scheme may include being able to automatically differentiate and segregate the neurons and pathways by class. The current implementation does not allow for a simple way to pull the neurons, synapses, and weights by class to connect them to other networks such as the symbolic processing network created in this work.

5.4.2 Symbolic Processing

The major challenges in the symbolic processing SNN were determining the optimal behavior for the *acc* neuron and increasing the discrete range of values stored in the network. This work utilized a spike then leak approach to the *acc* neuron. This design supported a linear relationship between the decrementing membrane potential of the *acc* neuron based upon input spikes arriving at the *store* neuron and the increasing time it would take for the *acc* neuron to polarize back to threshold to induce the second spike to the *output* neuron. An alternative design could have let the *acc* neuron leak then spike. This would make initializing the network much easier, but then there would be an inverse relationship between the *acc* neuron membrane potential increasing and the decreasing time it would represent between the two *output* neuron spikes. The second major challenge for the accumulator network was maximizing the number of events stored within the network while maintaining biological neurodynamic values. Greater values could be stored in the network if synaptic transmission times were reduced or voltage sensitivity was increased.

5.4.3 Sub-symbolic and Symbolic Integration

There were two major challenges in the integration of the sub-symbolic and symbolic networks. First, using actual spiking outputs from the pattern recognition network and directly connecting them into the symbolic accumulator network remains an item for future work. Since a means of identifying a neuronal population specific to a class was not easily devised, the output of the sub-symbolic pattern recognition SNN was simulated and provided

as input into the symbolic processing accumulator network through an intermediary *detector* neuron. The means of connecting the excitatory neurons to the *detector* neuron was the next challenge. A fully connected/soft voting scheme was decided upon due to simulating a single class of excitatory neurons and approaching that population of neurons as a binary classifier. Finally, building to scale was not accomplished due to time constraints. As stated before, only one digit class within the system was simulated and tested. A multi-class system, each with its own *detector* neuron and associated accumulator network should be built to show the scalability of the architecture.

5.5 Future Work

Based upon the observed challenges, suggested future work includes:

- Determine whether a truly unsupervised learning method is possible. Although this work used a learning method labeled "unsupervised," an external learning signal was still used, albeit, only the testing phase of the learning process. A possibility could be integrating a self-modulating reward signal during learning and test phases.
- Build into or bolt on a means to self-segregate and identify neurons and pathways that associate to a class.
- Test whether the "spike then leak" method is optimal in comparison to a "leak to spike" behavior for the *acc* neuron.
- Adjusting global variables such as T_{syn}, T_{neu}, T_{min}, and dt to optimize minimum time resolution to maximize counting size.
- Based upon in-class neuronal segregation in the pattern recognition network, directly connect those neurons, post-learning, to its associated *detector* neuron to test integration performance with actual MNIST data.
- For connecting the two component networks, determine whether the fully connected/soft voting scheme researched here is the optimal connection type. Suggested alternatives include:
  - One-to-one/hard voting scheme
  - Adjusting the synaptic weights and/or types
  - Adjusting the connecting neuron's neurodynamics, such as firing threshold
- Laterally scale this work and the above suggestions to more than a single class instance such that all 10 pattern recognition networks are connected to their respective *detector*

neuron and accumulator network.

Additionally, the accumulator network can be made to increment and decrement at the same time to allow the network to count up and down. Finally, to fully test and show a complete proof of concept of an end-to-end system, a DVS should be used to provide real-time spiking sensor input and port the SNNs onto actual neuromorphic hardware.

5.6 Conclusion

Biologically inspired SNNs show great promise in advancing automatic target detection, classification, and tracking, especially in a SWaP-T constrained environment. Capitalizing on the previous SNN work done by Peter Diehl and Matthew Cook on unsupervised STDP learning for pattern recognition and Xavier Lagorce and Ryad Benosman's STICK framework for general purpose computation utilizing a SNN, this research was able to show that:

1. Sub-symbolic unsupervised STDP learning for pattern recognition of the MNIST dataset achieves greater than 91% accuracy on current Python 3 and the Brian2 simulator.
2. The symbolic processing primitive of counting discrete events is possible using the STICK framework; these structures form an independent SNN.
3. Via simulation, sub-symbolic and symbolic SNNs are integrated through a single neuron which can provide close to 96% accuracy for counting MNIST digits by class.

These three findings are foundational in the pursuit of a biologically inspired automatic target detection, classification, and tracking system which can operate in a SWaP-T constrained environment. They show that pattern recognition and basic counting of events are possible on a SNN and that they are able to be integrated as a single component. The future operator, from the Marine on the ground in combat, to the intelligence analyst at their terminal, will have to make timely, accurate, and reliable life or death decisions based upon a growing and inundating amount of raw sensor data. The potential benefits of third generation AI could help maintain our competitive edge. Reducing power requirements will allow for smaller, lighter, and faster sensing devices from self-deployable ground reconnaissance arrays to autonomous drone swarms while reducing the Warfighter's combat load therefore increasing endurance and time on station. The greater computational power and massive

parallelism will allow for greater precision and analytic powers. Finally, the event based sensing and processing, which allows for power and computational efficiencies, will reduce bandwidth requirements in our global combat networks.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997.

[2] P. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, vol. 9, Aug. 2015.

[3] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges," *Frontiers in Neuroscience*, vol. 12, p. 774, 2018.

[4] R. Mooney, J. Shavlik, G. Towell, and A. Gove, "An experimental comparison of symbolic and connectionist learning algorithms," in *International Joint Conference on Artificial Intelligence*, Detroit, MI, Aug. 1989, pp. 775–780.

[5] J. W. Shavlik, "Combining symbolic and neural learning," *Machine Learning*, vol. 14, no. 3, pp. 321–331, Mar. 1994.

[6] X. Lagorce and R. Benosman, "Stick: Spike time interval computational kernel, a framework for general purpose computation using neurons, precise timing, delays, and synchrony," *Neural Computation*, vol. 27, no. 11, pp. 2261–2317, Jul. 2015.

[7] F. Javed, Q. He, L. Davidson, J. Thornton, J. Albu, L. Boxt, N. Krasnow, M. Elia, P. Kang, S. Heshka, and D. Gallagher, "Brain and high metabolic rate organ mass: Contributions to resting energy expenditure beyond fat-free mass," *The American Journal of Clinical Nutrition*, vol. 91, pp. 907–912, Feb. 2010.

[8] N. Bostrom and A. Sandberg, "Whole Brain Emulation: A Roadmap," Future of Humanity Institute, Oxford University, Tech. Rep. 2008-3, 2008.

[9] "NVIDIA DGX-2 Datasheet," NVIDIA, 2019, accessed Jan. 15, 2020. Available: https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-1/dgx-2-datasheet-us-nvidia-955420-r2-web-new.pdf

[10] P. Sterling and S. Laughlin, *Principles of Neural Design*, 1st ed. London, England: MIT Press, 2015, ch. 3, pp. 51–55.

[11] H. Kalita, A. Krishnaprasad, N. Choudhary, S. Das, D. Dev, Y. Ding, L. Tetard, H.-S. Chung, Y. Jung, and T. Roy, "Artificial neuron using vertical mos 2/graphene threshold switching memristors," *Scientific Reports*, vol. 9, no. 1, pp. 1–8, Jan. 2019.

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

[13] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.

[14] T. Bohnstingl, F. Scherr, C. Pehle, K. Meier, and W. Maass, "Neuromorphic hardware learns to learn," *Frontiers in neuroscience*, vol. 13, p. 483, 2019.

[15] C. Schuman, T. Potok, R. Patton, J. Birdwell, M. Dean, G. Rose, and J. Plank, "A survey of neuromorphic computing and neural networks in hardware," *Neural and Evolutionary Computing*, May 2017.

[16] A. Sengupta, P. Panda, P. Wijesinghe, Y. Kim, and K. Roy, "Magnetic tunnel junction mimics stochastic cortical spiking neurons," *Scientific Reports*, vol. 6, no. 1, Jul. 2016.

[17] B. J. Shastri, M. A. Nahmias, A. N. Tait, A. W. Rodriguez, B. Wu, and P. R. Prucnal, "Spike processing with a graphene excitable laser," *Scientific Reports*, vol. 6, no. 1, Jan. 2016.

[18] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, no. 8, Apr. 2020.

[19] S. A. Baby, B. Vinod, C. Chinni, and K. Mitra, "Dynamic vision sensors for human activity recognition," in *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, 2017, pp. 316–321.

[20] L. Everding and J. Conradt, "Low-latency line tracking using event-based dynamic vision sensors," *Frontiers in Neurorobotics*, vol. 12, p. 4, 2018.

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[22] M. Stimberg, R. Brette, and D. Goodman. (2019, Aug.). "Brian 2, an intuitive and efficient neural simulator". *eLife*. [Online]. Available: 10.7554/eLife.47314

[23] J. V. Monaco, M. M. Vindiola, and R. Benosman, "STEAM: Spike time encoded addressable memory," presented at the 7th Neuro Inspired Computational Elements Workshop, Albany, NY, 2019.

[24] R. Brette, D. Goodman, and M. Stimberg. (2015, Jul. 16). Changes from Brian 1 - Brian 2. [Online]. Available: https://brian2.readthedocs.io/en/2.0rc/introduction/changes.html

[25] D. Keysers, "Comparison and combination of state-of-the-art techniques for handwritten character recognition: Topping the MNIST benchmark," *Computing Research Repository*, vol. 10, no. 22, Oct. 2007.

[26] F. Chen, N. Chen, H. Mao, and H. Hu, "Assessing four neural networks on handwritten digit recognition dataset (MNIST)," *Chuangxinban Journal of Computing*, vol. 11, no. 08, Jun. 2019.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California