Investigation of Synapto-dendritic Kernel Adapting Neuron Models and their use in Spiking Neuromorphic Architectures

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Science

University of Western Sydney

by

Saeed Afshar

June 2015

Table of Contents

Contents	
Table of Contents	3
List of Tables	4
List of Figures	4
Abstract	6
Acknowledgements	6
Chapter 1: Introduction	8
Neuromorphic implementation of Spiking Neural Networks	8
Neurons as spiking, distributed processors	10
Too many synapses	11
Chapter 2: SKAN at the single neuron scale	16
SKAN building blocks	16
Synapse/Dendrite:	16
Soma:	
Feedback mechanisms/Learning rules:	
Synapto-dendritic kernel slope adaption	
Threshold adaptation	
Single SKAN Dynamics	20
Observing the first spike in a spike train or burst	ark not defined.
Selecting to learn the commonest spatio-temporal patterns	20
SKAN response time improves with adaptation without information loss	21
Evolution of the temporal receptive field in SKAN	23
Learning in the presence of Poisson spike noise and missing target spikes	
Chapter 3: SKAN at the network scale	
Multi-SKAN classifier	
Online unsupervised spatio-temporal spike pattern classification	
1-to-1 neuron to pattern allocation	
Classification performance as a function of spatio-temporal pattern dimension	
Classification in the presence of temporal noise	
Chapter 4: SKAN and STDP	

Synaptic weights as independent encoders of input noise	.36
Simplified weight update rule	.37
Synaptic weight normalization without division	. 38
STDP and SKAN combine to produce synaptic encoding of afferent Signal to Noise Ratio (SNR)	.42
Recognition performance on noise corrupted spatio-temporal patterns	.45
Chapter 5: Hardware implementation and application	.47
Implementation in FPGA hardware	.47
Example application: Unsupervised feature learning using a camera with noise corrupted pixels	.48
Conclusion	.50
References	.51

List of Tables

Table 1. Parameter values used for all SKAN results.	32
Table 2. Altera Cyclone V FPGA resource usages	47

List of Figures

Figure 1. Using weighted synaptic kernels to learn spatio-temporal spike patterns	8
Figure 2. Information flow schemes in unsupervised spiking neural network algorithms	10
Figure 3. Comparison of neuromorphic implementations of synapto-dendritic kernels	12
Figure 4. The inter-neuronal transfer function	15
Figure 5. Schematic of the elements and information paths in a SKAN neuron	16
Figure 6. The simplified adaptable synapto-dendritic kernel of SKAN	17
Figure 7. The adaptation of SKAN	19
Figure 8. Commonest pattern selection as a function of pattern presentation probability	21
Figure 9. Narrowing of kernels leads to improved response time during neuronal adaptation	22
Figure 10. Tracking a hidden ISI producing process and its variance	25
Figure 11. Evolution of SKAN's receptive field in response to input	26
Figure 12. Learning spatio-temporal spike patterns in the presence of noise.	27
Figure 13. Learning error as a function of spike signal to noise ratio	28
Figure 14. Learning using a single global decaying inhibitory signal	30
Figure 15. Convergence rate of as a function of neuron/pattern numbers	33
Figure 16. Convergence rates as a function of input channel dimension and pattern width	34
Figure 17. Convergence as a function of temporal noise	35

Figure 18. Comparison of STDP rules	37
Figure 19. Comparison of fixed point division and shift based normalization.	40
Figure 20. RMS error of shift based normalization	41
Figure 21. Synaptic weight adaptation in a neuron with three synaptic inputs	43
Figure 22. Mean normalized synaptic weights <i>w</i> _i as a function of noise spike rate	44
Figure 23. Evolution of synaptic weights of a sixteen input neuron	45
Figure 24. Enhanced recognition performance via synaptic signal to noise ratio encoding	46
Figure 25. Unsupervised learning of common features concurrent with SNR encoding synaptic we	ights
on noise corrupted spatio-temporal patterns	49

Abstract

The motivation for this thesis is idea that *abstract*, adaptive, hardware efficient, inter-neuronal transfer functions (or kernels) which carry information in the form of postsynaptic membrane potentials, are the most important (and erstwhile missing) element in neuromorphic implementations of Spiking Neural Networks (SNN). In the absence of such abstract kernels, spiking neuromorphic systems must realize very large numbers of synapses and their associated connectivity. The resultant hardware and bandwidth limitations create difficult tradeoffs which diminish the usefulness of such systems.

In this thesis a novel model of spiking neurons is proposed. The proposed Synapto-dendritic Kernel Adapting Neuron (SKAN) uses the adaptation of their synapto-dendritic kernels in conjunction with an adaptive threshold to perform unsupervised learning and inference on spatio-temporal spike patterns. The hardware and connectivity requirements of the neuron model are minimized through the use of simple accumulator-based kernels as well as through the use of timing information to perform a winner take all operation between the neurons. The learning and inference operations of SKAN are characterized and shown to be robust across a range of noise environments.

Next, the SKAN model is augmented with a simplified hardware-efficient model of Spike Timing Dependent Plasticity (STDP). In biology STDP is the mechanism which allows neurons to learn spatiotemporal spike patterns. However when the proposed SKAN model is augmented with a simplified STDP rule, where the synaptic kernel is used as a binary flag that enable synaptic potentiation, the result is a synaptic encoding of afferent Signal to Noise Ratio (SNR). In this combined model the neuron not only learns the target spatio-temporal spike patterns but also weighs each channel independently according to its signal to noise ratio. Additionally a novel approach is presented to achieving homeostatic plasticity in digital hardware which reduces hardware cost by eliminating the need for multipliers.

Finally the behavior and potential utility of this combined model is investigated in a range of noise conditions and the digital hardware resource utilization of SKAN and SKAN + STDP is detailed using Field Programmable Gate Arrays (FPGA).

Acknowledgements

I would like to thank Dr. Tara Julia Hamilton for her unwavering support without which this thesis would not have been possible. I would also like to thank Professors Jonathon Tapson, Philip de Chazal and Andre van Schaik for their helpful discussions and help in editing the thesis. I would like to thank Richard Sofatzis, Libin George and Chetan Thakur for their interest in the SKAN model, and for putting in the time to develop their own hardware implementations of the model.

Statement of originality

The intellectual content of this thesis, the SKAN algorithm, algorithm simulation results, hardware implementation, results and writing is entirely the product of my own work and all assistance received in preparing this thesis has been acknowledged.

Chapter 1: Introduction

Neuromorphic implementation of Spiking Neural Networks

Real neurons, the electrically excitable cells of the Eumetazoan, constitute an extremely diverse intractably complex community whose dynamic structures and functions defy all but the broadest generalizations [1][2]. In order to minimize this complexity, the field of Artificial Neural Networks (ANN) has traditionally modeled neurons as deterministic, centrally clocked elements which operate on real valued signals [3][4]. These signals represent neuronal rate coding where the spiking rate of a neuron encodes useful information and the adjustment of synaptic weights results in learning. This scheme, while mathematically amenable incurs a significant energy cost by discarding the rich temporal information available in the real signals used by neurons to communicate, since many more spikes per input channel are required to transmit a real valued rate value as opposed to a single spike per channel needed for transmission of temporally encoded inter-spike interval information[5][6][7]. In contrast to the standard ANN model, the highly optimized, low power, portable signal processing and control system that is the brain, readily uses temporal information embedded in the input signals and internal dynamics of its stochastic heterogeneous elements to process information [8].



Figure 1. Using weighted synaptic kernels to learn spatio-temporal spike patterns. a) Typical functional model of a spiking neuron with synapses whose transfer functions or kernels do not adapt as a result of learning. These *static* synaptic transfer function act as memory elements providing information about recent spikes to the soma. While the adaptation of the synaptic strength or weight of the kernels allows learning of input spike patterns. b) Biological representation of the neuron showing the learnt input spike pattern, the resultant Excitatory Post-Synaptic Potentiation (EPSP) and the output spike indicating pattern recognition. c) Presentation of a non-target pattern results in an EPSP that does not cross the threshold producing no output spike. Figure from [9].

More recently, the greater efficiency, higher performance and biologically realistic dynamics of temporal coding neural networks has motivated the development of novel Spiking Neural Networks (SNN). These include unsupervised neural network algorithms such as polychronous networks [10], echo state machines [11] liquid state machines [12], sparse coding models [13] and spike-based Radial Basis Function (RBF) networks [14] which combine feedback, axonal or dendritic delays or synaptic kernels with STDP or related learning rules to learn particular input spike patterns. Supervised algorithms such as the online learning algorithms, spikeProp [15], ReSuMe [16] and the tempotron [17] use STDP in combination with a supervisory spiking signal to learn mappings from input spike trains to output spike trains, whereas the Synaptic Kernel Inverse Method [18] performs the same mapping through calculating a linear solution to a higher dimensional hidden layer projection. In addition some learning

algorithms such as the deSNN which combines rank order learning with spike time learning operate in both supervised and unsupervised modes [19]. At a higher level of analysis models such the Hierarchical Temporal Memory attempt to replicate the complexity of the neocortex by combining spatio-temporal pooling coincidence detection and Bayesian networks [20]. As shown in Figure 1 these models typically adapt the value of synaptic weights with static synaptic kernels to learn a particular spatio-temporal pattern. Many of the neuron models proposed by computational neuroscientist are later implemented in neuromorphic hardware [19][21][22][23][24][25][26]. However these algorithms are almost always initially designed not for their suitability in hardware but for performance in a constraint free mathematical context with numerous all-to-all connected neurons and/or to satisfy some biological realism criteria. This disassociation between algorithm design and the hardware design can result in difficulties in the hardware implementation phase where the original full-scale model is typically simplified to the point where it can be implemented in hardware while still retaining its functionality.

Neuromorphic implementations of spiking neural networks can be broadly be divided by whether the system performs learning on-chip or offline, whether synapses are implemented using digital or analog circuits and whether the learning is supervised or unsupervised [27]. Neuromorphic platforms such as IBM's true north chip [28] can train large scale neural networks offline on predefined datasets . These trained networks are then used for inference on new data. This offline learning approach can greatly simplify hardware implementation through removal of all learning circuits however it also results in a system that cannot train on new real world data after system deployment which may be necessary in some applications. Among platforms with on-chip learning the choice between analog, digital or mixed signal implementations offers a variety of tradeoffs including design simplicity, power-efficiency and scalability. See [24] for a detailed review. Additionally recent advances in the field of memristive devices has allowed the development of memristor-based crossbar architectures which promise high density low power implementation of complex synaptic dynamics [29]. While all synaptic models and update rules including the SKAN model presented in this thesis can potentially be realized using digital, analog, memristive or mixed implementations, this thesis focuses primarily on digital implementations.

Finally the use of supervisory signals such as those used in spikeProp, ReSuMe and the tempotron allows these models to generate mappings from prescribed inputs to prescribed outputs. While the supervised learning approach taken in these models together with the rise of the deep learning paradigm has been the central driver for recent advances in machine learning, unsupervised learning rules such as the SKAN model presented in this thesis are likely to be a key component in future developments since many biological neuronal systems are hypothesized to operate in this mode [30].

The last decade has seen significant progress in the implementation of unsupervised spiking neural networks. In [31] a simplified RBF-based spiking neural network proposed in [14] is implemented in FPGA hardware which uses an unsupervised delay adaptation learning rule with 10-bit synapses and an integrate and fire somatic model. In this system new neurons are added as novel patterns are presented to the system. Here novel patterns are defined as those which do not generate activity in the network within a predefined time interval. This RBF-based model has been used in a range of applications such as control and visual processing [31][32]. In [33] and [34], transposable crossbar SRAM arrays are used to implement high density binary synapses for use in classification and associative memory tasks. While

implementation of the synapse as a simple binary latch enables the realization of large numbers of synapses in hardware, the reduced synaptic resolution also reduces overall network capacity [33]. In [21] a sparse coding spiking neural network is presented and demonstrated as an image feature extractor. This implementation uses 13-bit synapses during learning, and 8-bit synapses for inference. This system leverages the low spike rate generated through the sparse coding algorithm to implement efficient inter-neuronal communication while keeping spike collision rates low enough so as not to affect inference. Of these previously proposed models implemented in digital hardware, the system in [31] is most comparable to the proposed SKAN model in that it is also designed to operate primarily in the temporal coding domain. However this system uses standard integrate and fire neurons with preset threshold levels, and timing windows and as such does not exhibit the complex set of behaviors and functionalities exhibited by the SKAN systems.

Neurons as spiking, distributed processors

One of the motivations of this thesis is to attempt to bridge the gap between neuron model and hardware design, making hardware constraints considerations central during the algorithm design stage. In this context the goal of computational performance in hardware motivates a change in focus from claims of accurate modeling of computation in biological neurons to exploiting the computational power of artificial but biologically inspired neurons. These are herein defined as a set of simple distributed informational processing units that communicate through binary valued pulses (spikes), receive inputs from multiple input channels (synapses and dendrites), and have a single output channel (axon).



Figure 2. Information flow schemes in unsupervised spiking neural network algorithms and their impact on hardware implementation. Black indicates the fundamental elements and information paths of a spiking neural network. Red indicates added features and information paths that can cause difficulties in hardware, or limit algorithm utility. Figure from [9].

Figure 2 illustrates the basic elements of spiking neural network algorithms as well as some useful information flow and storage restrictions (red), which, if adhered to at the neuron model design stage, prove helpful during the hardware implementation stage. These restrictions include:

- 1. Self-contained: In a self-contained system, no external controlling system is required for the system to function. Examples of systems that are *not* self-contained include synapses that require adjustment via an external controller.
- Scalable connectivity: Systems that require all-to-all connectivity between the neurons or where the synapses or dendrites directly communicate their weights or potentials to each other are not hardware scalable or biologically possible. All-to-all connected neurons require a quadratically increasing number of connections, which is prohibitive both in hardware and in the brain [35][36].
- 3. Storage of time series data: Systems whose processing units require large segments of their time series data to be stored and be accessible for later processing in the fashion of standard processors require a significant amount of on-site memory not possible in biological systems and would add significant complexity to neuromorphic hardware. Furthermore, such systems overlap with the domain of distributed processors such as GPUs and fall outside the neuromorphic scope.
- 4. Multiplication: Multipliers are typically inefficient to implement in digital hardware and are limited in standard digital solutions such as Field Programmable Gate Arrays (FPGAs) and Digital Signal Processors (DSPs). Their computational inefficiency and their limited availability on hardware platforms result in neural networks being implemented using time-multiplexing techniques. This, in turn, limits the system scale and the applications where this hardware is viable or the speed at which the system can operate [37][38].

While the algorithm restrictions listed above can simplify hardware implementation the central limiting factor in the realization of neuromorphic systems is the need for the realization of large numbers of interconnected synapses.

Too many synapses

Synapses are by far the most numerous computational elements in the brain and in neuromorphic systems. Due to their large numbers, the return on investment on synapses, i.e., how much functional computation they perform versus how much hardware resources they take up, becomes a defining feature of any neural system whether evolved or engineered [39][40][41]. Thus the extraction of the most functionality from the fewest, simplest synapses is often a central focus for the neuromorphic engineer [42][43].

In the context of spiking neuromorphic systems the synapse serves three essential functions. The first is simply to form a connection from one neuron to the next. The second is to generate a memory of the input spike via the synaptic kernel and the third is to weigh this kernel such that when it is added to other similarly weighted synaptic kernels, the resulting summation, called the somatic membrane potential, is a useful signal, encoding functionally relevant information such as how well an input spatio-temporal pattern matches those commonly seen in the past.

While the realization and use of these distributed kernel-based processing units has evidently been mastered by evolution, despite significant recent progress, our best engineered systems still find the large-scale realization of these three functions challenging. The first and simplest function of the synapse, that of acting as the connection between neurons represents the greatest hardware challenge. Limitations in network bandwidth or connectivity are often the most serious obstacles restricting full utilization of neuromorphic hardware resources. Innovative approaches such as time multiplexing [41] and Address Event Representation (AER) [44][45] can create virtual all-to-all connected networks, however, these advantages come at the expense of reduced operating speed.



Figure 3. Comparison of neuromorphic implementations of synapto-dendritic kernels. The characteristics of realized Excitatory Post-Synaptic Potential (EPSP) kernels are computationally important just prior to being summed at the soma. These kernels represent the cost function used to translate the temporal error in spatiotemporal spike patterns at the synapse to the somatic membrane potential. Due to the large number of synapses neural network systems require, the complexity, functionality, and hardware cost of these kernels is a critical feature of neuromorphic spiking networks. a) An exponentially decaying kernel typically used for modelling synapses. Here the kernel response and thus the delay to the peak of the kernel is non adaptive (static) in response to inputs while the strength or weight of the kernel adapts as a result of learning. b) A simplified synaptic model with the kernel modelled as a binary delayed window. Here *w* is a binary zero or one weight. c) The SKAN model described in this thesis. Figure from [46].

The second function of the synapse, that of spreading an input spike's energy over time can be realized via a range of synaptic kernels with varying levels of complexity, hardware cost and computational utility. Two extremes include an exponentially decaying kernel shown in Figure 3 (a) which is typically used to model biological synapses [24], and the simple delay learning approach with a binary kernel and a temporal tolerance window shown in Figure 3 (b) [47]. In the ideal synaptic model, a real-valued synaptic alpha function is multiplied by a real-valued synaptic weight with the later adapting to input spikes to model synaptic weight adaptation. However the cost of implementing this synaptic kernel in large numbers in digital hardware is substantial as it requires the realization of a multiplier at each synapse. In addition, the alpha function does not model the computationally useful peak delay adaptation effects observed in biology [48] which necessitates the realization of multiple synapses for learning arbitrary delays.

The third function of the synapse, that of weighing the kernel, requires another multiplication operation between the synaptic weight and the instantaneous value of the EPSP kernel. For real-valued synapses and kernels the hardware cost of this multiplication operation can be prohibitive [42]. At the other end of the spectrum, rather than implement complex synaptic weight adaptation, other neuromorphic implementations of spiking networks have focused on the adjustment of explicit propagation delays along the neural signal path to encode memory [47][49][50]. Here the energy of the spike is spread via a binary valued tolerance window as shown Figure 3 (b). This discarding of synaptic weights significantly simplifies implementation and allows more synapses to be realized. The down side is that explicit window-based delay learning schemes can produce "sharp" systems with lower tolerance for the dynamically changing temporal variance they inevitably encounter in applications where neuronal systems are expected to excel: noisy, dynamic and unpredictable environments [51]. In addition while use of these simplified kernels allow more synapses to be realized, limited network bandwidth can sometimes mean that these larger numbers cannot actually be fully utilized.

The feature shared by all previously proposed neuromorphic models of spiking neural networks is that the synaptic kernels used for encoding temporal information are static. However recent advances in neurophysiology have revealed that synapto-dendritic structures which include the synapse, dendritic tree and dendritic trunk and their associated transfer functions are highly complex and adapt during learning in response to the statistics their stimulus environment [52]. Some of the earliest investigations of the input-output behavior of dendritic trees revealed them to be active computational elements exhibiting complex nonlinear operations such as 'dendritic cluster sensitivity' where excitatory synaptic inputs concentrated on particular regions of dendritic trees generate stronger neuronal response relative to diffuse excitation [53]. Over the next two decades advances in imaging and neurophysiological recoding techniques have enabled detailed investigation of dendritic spines which are small outgrowths residing on dendritic trees. Due to their critical position between synaptic inputs and the soma they are considered to be integral to neuronal processing and come in a variety of shapes, sizes and population densities suggesting a high level of functional diversity [54]. Dendritic spines have been shown to enhance cooperative interaction among multiple inputs, by enhancing interaction among co-active spines and increasing nonlinear integration at the dendritic tree [55]. Furthermore through NMDAR regulation of potassium channels dendritic spines enable localized modification of dendritic nonlinearities in response to correlated inputs at the dendrite in a branch-specific manner [56]. The plasticity of dendritic spines in terms of density and shape has not only been shown to modify the transfer function from synapse to soma but also affect long and short term synaptic plasticity, as well as affect high level memory and cognition. In addition disorders in dendritic spine functioning have been implicated in psychiatric disorders [57].

These discoveries are significant in the context of the computational power of even single biological neurons. Whereas in the traditional neuron model synapto-dendritic structures functioned as simple weights and cables connecting one soma to the next, the recent findings have demonstrated a wide range of signal integration and processing occurring along the signal path, which confers considerable computational power to single neurons [58][59]. These effects represent novel dynamics with as yet unexplored emergent computational properties, which may potentially solve currently intractable

problems in computational neuroscience [60][61]. These dendritic adaptation effects have recently been modelled through large rule sets [62][63][64] and in the neuromorphic field the use of dendrites for computation is beginning to be explored [65][66][67][68]. However with biological realism as a major focus, many of the models carry significant extra complexity which can impede scalibity. In contrast while this work does not attempt to model the recently discovered neurophysiological mechanism underlying synaptic and dendritic adaptation mechanisms, it does utillize the underlying principle that when combined these effects can realize an arbitrary inter-neuronal transfer function with many degrees of freedom and that this adaptability can be leveraged for the implementation of more computationally efficient neuron designs.



Figure 4. The inter-neuronal transfer function is made up of many dynamic synaptic and dendritic couplings yet these couplings can be abstracted to a single adaptable transfer function or kernel which can be implemented efficiently in hardware. The simplification of inter-neuronal coupling as a single kernel significantly reduces hardware and connectivity requirements in comparison to modelling every synapse and dendritic tree. Although such a simple model discards a large amount of detail, the functionality of the neuron to learn and recognize spatio-temporal spike patterns is preserved.

In this context, the Synapto-dendritic Kernel Adapting Neuron (SKAN) model, which is the focus of this thesis and is shown Figure 3 (c), uses fully adaptable yet simple accumulator based kernels. Here it should be noted that the kernels of SKAN do not directly model individual synapses, which have kernels that are approximately static i.e. do not adapt their transfer functions. Prior to the recent discoveries of dendritic kernel adaptation, the standard neurophysiological model of spike pattern detection has been primarily that of synaptic efficacy adaptation through STDP in combination with random synaptic propagation delays, where the temporal relation of pre- and postsynaptic spikes determines the change in synaptic efficacy [69]. There have been many proposed STDP models such as the early models by Gerstner et al in the auditory system [70] as well as more complex triplet spike STDP [71]. These models and others have revealed important neuro-computational principles which arise as a result of STDP such as the tuning of neurons to the earliest spikes in a spike train [72]. As shown in Figure 4, the kernels of SKAN can be interpreted as simplified models of multiple synapses as well as the dendritic tree on which they reside, with the assumption that the function of the whole system is to create a mapping from the

afferents to the soma such that commonly presented spatio-temporal patterns are preferentially transmitted. While the kernel adaptation modelled in SKAN could potentially be approximated by a biologically realistic model of multiple synapses and dendrites under conventional STDP rules, the SKAN model is an attempt to model the entire synapse to soma coupling system in as simplified and hardware amenable way possible while still preserving its functionality. The investigation of the SKAN model and its extension which form the basis of this thesis have been presented in [9] and [46] respectively.

Chapter 2: SKAN at the single neuron scale

The elements of SKAN and its learning rule are defined in this chapter. In the next chapter, the dynamical behaviors of SKAN are described.

SKAN building blocks

At the single neuron level, SKAN consists of a combined synapto-dendritic kernel adaptation and a homeostatic soma with an adapting threshold as shown in Figure 5.



Figure 5. Schematic of the elements and information paths in a SKAN neuron. The input spikes (blue) trigger adaptable synapto-dendritic kernels (magenta) which are summed to form the neuron's somatic membrane potential (cyan). This is then compared to an adaptive somatic threshold (red) which, if exceeded, results in an output pulse (green). The output pulse also feeds back to adapt the kernels. Note that in this chapter the synaptic weights (orange) are held constant and equal for all synapses. Also note that the back propagating signal does not travel beyond the synapto-dendritic structures of the neuron to previous neural layers. Figure from [9].

Synapse/Dendrite:

An incoming input spike initiates a simplified synapto-dendritic kernel at each input channel *i*. This kernel is controlled by a physiological process, p_i , and for simplicity is modeled as a ramp up and a ramp down sequence generated via an accumulator r_i with step size Δr_i . An input spike triggers p_i , starting the first phase where the accumulator ramps up at each time step Δt by Δr_i until it reaches a maximum value w_i which represents the synaptic weight, and which is kept constant throughout this chapter to simplify the algorithm. Note that in this work only positively weighted synapses are considered. After r_i reaches w_i , the process switches from the ramp up phase, $p_i=1$, to a ramp down phase, $p_i=-1$, which causes the accumulator to count down at each time step towards zero with the same step size Δr_i , until it reaches

zero, turning off the physiological process, $p_i=0$. It will stay in this state until a new incoming spike reinitiates the sequence. This simple conceptual sequence, which is analogous to a dendritically filtered neuronal EPSP, is illustrated in Figure 6.



Figure 6. The simplified adaptable synapto-dendritic kernel of SKAN. An input spike (blue) triggers the kernel's ramp up ramp down sequence. The input spike sets a flag p_i representing a physical process to one (green). The flag causes an accumulator (magenta) to count up from zero by Δr_i at each time step until it reaches w_i (constant orange dotted line), after which the flag is set to negative one, which causes the accumulator to count back down to zero, at which point the flag returns to zero completing the sequence. The value of the accumulator represents the synapto-dendritic kernel, i.e., the post-synaptic potential, which travels to the soma and is summed with other kernels to produce the somatic membrane potential. Figure from [9].

The state of the ramp up ramp down flag sequence is described by Equation 1:

Equation 1:

$$p_{i}(t) = \begin{cases} 1 & \text{if } \left(u_{i}(t) = 1 \land p_{i}(t-1) = 0 \right) \lor \left(p_{i}(t-1) = 1 \land r_{i}(t-1) < w_{i} \right) \\ -1 & \text{if } \left(p_{i}(t-1) = 1 \land r_{i}(t-1) \ge w_{i} \right) \lor \left(p_{i}(t-1) = -1 \land r_{i}(t-1) > 0 \right) \\ 0 & \text{else} \end{cases}$$

The *w* parameter in SKAN has similarities to the weight by which a synaptic kernel is multiplied in standard synaptic STDP neuron models and neuromorphic circuits, but with the advantage of not requiring any multipliers, which are otherwise required at every synapse in hardware implementations. The adjustment of *w* in SKAN, via standard synaptic STDP schemes would allow synaptic prioritization and/or the closing off of inactive or noisy channels. The combined effects of dendritic structure and synaptic weight plasticity has only recently begun to be explored, but early evidence points to significant computational power of such a combined system [73]. In this chapter, however, in order to clearly demonstrate the stand-alone capabilities of SKAN's synapto-dendritic kernel adaptation mechanism, the synaptic weight parameter of *w* is held constant and is identical for all synapses. In chapter 4, we explore the consequences of allowing *w* to be learned.

Soma:

At the soma the synapto-dendritic kernels are summed together. This summed term is analogous to the membrane potential of a biological neuron. Along with the membrane potential the soma uses a dynamic threshold voltage parameter $\Theta(t)$ and as long as the membrane potential exceeds threshold, the soma spikes, setting the binary s(t) from 0 to 1 as described in Equation 2:

Equation 2

$$s(t) = \begin{cases} 1 & \text{if } \sum_{i} r_i(t) > \Theta(t-1) \\ 0 & \text{else} \end{cases}$$

SKAN differs from most previous spiking neuron models in not resetting the membrane potential after spiking (see [18], [74] for exceptions). This permits wide pulse widths at the neuron output s(t). While such wide pulses do not resemble the canonical form of the single spike, they are analogous to concentrated spike bursts and play a significant part in the functioning of SKAN. The wide pulses which result from not resetting the membrane potential allow multiple learning events per input spatio-temporal pattern as opposed to the single learning event which would occur if the membrane potential was reset at the onset of the output spike.

Kernel and threshold adaptation:

Synapto-dendritic kernel slope adaption

One of the central elements of SKAN is the feedback effect of the output pulse s(t) on each of the synapto-dendritic kernels. Here s(t) is analogous to the back propagating spike signal in biological neurons which travels back up the dendrites toward the synapses and is responsible for synaptic STDP.

The logic of the kernel adaptation rule is simple; if a particular dendrite is in the ramp up phase $p_i=1$ and the back propagation signal s(t) is active, the soma has spiked and this particular kernel is late to reach its peak, meaning that the other kernels have cooperatively forced the membrane potential above the threshold while this kernel has yet to reach its maximum value w_i . In response, the ramp's step size Δr_i is increased by some small positive value *ddr* for as long as the output pulse is high (s(t)=1) and the kernel is in the ramp up phase. Similarly if a kernel is in the ramp down phase $p_i = -1$ when the back propagation signal is high, then the kernel peaked too early, having reached w_i and ramping down before the neuron's other kernels. In this case the ramp step size Δr_i is decreased by *ddr*. Equation 3 describes this simple kernel adaptation rule:

Equation 3

$$\begin{bmatrix} r_{i\,(t)} \\ \Delta r_{i\,(t)} \end{bmatrix} = \begin{bmatrix} r_{i\,(t-1)} \\ \Delta r_{i\,(t-1)} \end{bmatrix} + p_{i\,(t-1)} \begin{bmatrix} \Delta r_{i\,(t-1)} \\ ddr \times s_{(t-1)} \end{bmatrix}$$

The use of indirect evidence about the dynamic state of other dendrites in the form of the back propagating spike is a central feature in the operation of SKAN and enables the synchronization of all the neuron's dendritic kernel peaks as shown in Figure 7.



Figure 7. The adaptation of SKAN. The kernels and the threshold of SKAN adapt in response to repeated spatio-temporal pattern presentations. For visual clarity the pattern only consists of the Inter-Spike Interval (ISI) across two input channels $u_i(t)$ such that the pattern width (PW) is equivalent to the ISI. By the third presentation of the pattern the kernels have captured the ISI information. With each subsequent presentation the threshold $\Theta(t)$ increases making the neuron more selective as the kernel step sizes $\Delta r_i(t)$ increase making the kernels narrower. As a result each pattern presentation increases the neuron's confidence about the underlying process producing the ISI's, narrowing the neuron's receptive field around the target ISI and producing a smaller output pulse s(t) until, by the 11^{th} presentation ($t=2300 \Delta t$), the Θ_{rise} during the output spike and Θ_{fall} balance each other such that the $\Theta_{before} \approx \Theta_{after}$. The soma output spike s(t) is now a finely tuned unit delta pulse which indicates high certainty. When the membrane potential returns to zero, the neuron's threshold falls as indicated by the grey circle at the top left of the figure. Figure from [9].

Threshold adaptation

The threshold of SKAN is adaptive and changes under two conditions: when the neuron outputs a spike and when the membrane potential returns to zero.

At every time step during an output pulse s(t)=1 the threshold increases by Θ_{rise} . This increase in the threshold is analogous to the frequency adaptation effect seen in neurons, which creates a feedback

loop reducing the ability of the neuron to spike. Similarly in SKAN, the higher threshold reduces the likelihood and duration of an output pulse. This effect is shown in Figure 7 and is described in the first line of Equation 4.

Equation 4

$$\Theta(t) = \begin{cases} \Theta(t-1) + \Theta_{rise} & \text{if } \sum_{i} r_i(t) > \Theta(t-1) \\ \Theta(t-1) - \Theta_{fall} & \text{if } \sum_{i} r_i(t) = 0 \ \land \sum_{i} r_i(t-1) > 0 \\ \Theta(t-1) & \text{else} \end{cases}$$

The post spike decrease in threshold Θ_{fall} operates in opposition to the Θ_{rise} term. The returning of the membrane potential $\Sigma r_i(t)$ to zero causes a decrease in the threshold by Θ_{fall} as described by the second line of Equation 4 and shown in Figure 7. The counter balancing effect produced by the Θ_{fall} and Θ_{rise} in SKAN is a highly simplified version of the complex mechanisms underlying spike-threshold and frequency adaption in biological neurons [75], [76], where excited neurons eventually reach an equilibrium state through homeostatic processes such that the average spike frequency of neurons with a constant input tends asymptotically toward a non-zero value as $t \rightarrow \infty$.

This simple rule set describes all the elements of a single SKAN.

Single SKAN Dynamics

In this section the dynamics emerging from SKAN's rule are discussed for the single neuron case.

Detecting the onset of a spike or burstAs described in the first line of Equation 1 the ramp up phase of the kernel at channel *i* is only initiated if a spike arrives at the channel (u_i =1) while and the kernel is inactive (p_i =0). As a result while the *i*th kernel is active no further input spikes are observed. This has the effect that each input channel adapts its kernel to the leading spike in a spike train or burst. For the case where the spike train or burst is of shorter duration than the total duration of the kernel, the behavior of the neuron is identical to one where the burst is replaced by a single input spike arriving at the start of the burst. The effect of more general Poisson noise spikes is described later in this section.

Selecting to learn the commonest spatio-temporal patterns

As a single neuron, SKAN has previously been shown to select and learn the most common spatiopattern presented in a random sequence containing multiple patterns [77]. This effect has been demonstrated in the context of visual processing where hand gestures were transformed to spatiotemporal patterns via a neuronal transform operation [78] and processed by SKAN [79]. Figure 8 shows the performance of a four input neuron as a function of spatio-temporal pattern probability. The graph shows that the neuron's selection of commonest pattern is significantly above chance such that for sequences with P(x)>0.85 only the more common pattern will selected.



Figure 8. Commonest pattern selection as a function of pattern presentation probability. The inset illustrates one simulation of a 5 pattern long sequence where each pattern is sampled from two randomly initialized spatio-temporal patterns *x* and *y*, with probability P(x)=0.6. In this particular simulation pattern *x* was selected by the neuron. The plot shows the percentage of simulations where the neuron selected pattern *x*, as a function of its presentation probability P(x)=0.5 to 1. The result shows that the likelihood of a pattern being selected rises with increasing presentation probability. For each simulations the neuron received a sequence of 300 pattern presentation and the output of the neuron for the second half of the sequence $(150^{\text{th}} - 300^{\text{th}})$ pattern) was recorded. If during the second half of the sequence the neuron spiked if and only if pattern *x* was presented the neuron was deemed to have selected the pattern. Figure from [9].

SKAN response time improves with adaptation without information loss

In addition to the kernel adaptation and increasing threshold effect, the response time of SKAN, i.e., the time from the last arriving input spike in a pattern to the neuron's output spike, decreases with every pattern presentation as shown in Figure 9. In STDP schemes the earliest spikes in a spatio-temporal pattern tend to be highly weighted while the later spikes have little effect on recognition [84]. This behavior can be seen as advantageous if an assumption is made that the later input spikes carry less information however in this is an assumption that cannot be made in general. For spatio-temporal spike patterns where late spikes carry relevant information the exponential decay of the STDP curve means this later information acquires a lower synaptic weight and is in effect ignored regardless of information content. In contrast SKAN's adaptable kernels reduce output spike latency with adaptation while still enabling every spike to affect the output equally. This effect proves critical in the context of a multi-SKAN competitive network, where the best-adapted neuron is also always the fastest neuron to spike.



Figure 9. Narrowing of kernels leads to improved response time during neuronal adaptation in a two input neuron. For visual clarity the neuron is presented with an ISI=0 Δt pattern and the two kernels start with identical initial slopes ($\Delta r_1(0) = \Delta r_2(0) = \Delta r_{before}$). In the region under the output pulse, r(t) is the second integral of the constant ddr and therefore follows time symmetric parabolic paths (a) and (b) as it rises and falls. However due to the threshold rise which also occurs during the output pulse, the output pulse is not symmetric around the r(t) peak, such that the parabolic ramp down phase (b) is shorter than the parabolic ramp up phase (a). As a result of this asymmetry Δr_{after} is larger than Δr_{before} . This effect increases the kernel's slope Δr with each pattern presentation, narrowing the kernels until Δr reaches Δr_{max} . As a result of this narrowing, the response time of the neuron from last arriving input spike to the rising edge of the output, which is T₁ in the first presentation, improves until it reaches its minimal possible value T_∞ ≈ $w /\Delta r_{max}$. Figure from [9].

As shown in Figure 9, the combination of the kernel and threshold adaptation rules of SKAN increases Δr and decreases the response time between the last arriving input spike and the rising edge of the output spike with each presentation. If this increase is left unchecked Δr will increase until it equals w at which point the kernels take the shape of a single pulse such that $T_{\infty}=1\Delta t$. To prevent this Δr must saturate at Δr_{max} as shown in Figure 9 with Δr_{max} limited by Equation 5. This restriction ensures that the kernel of the first spike in an input pattern cannot return to zero before the last spike in the pattern arrives enabling all kernels to converge due to feedback from the same output signal.

Equation 5

$$\Delta r_{max} < w / PW$$

where *PW* is the maximal pattern width of the target pattern.

Evolution of the temporal receptive field in SKAN

Recent work has demonstrated the connection between synaptic weight adaptation and approximate probabilistic inference in the context of rate coding and spiking networks [80], [81], [82], [83], [84], [85], [86], [87], where typically the state of binary hidden variables are inferred from noisy observations using a large number of neurons. In this section we show that synapto-dendritic kernel adaptation enables a single neuron to make statistical inferences not about binary hidden variables but about hidden interspike interval (ISI) generating processes. Note that in this context the inter-spike intervals may be negative indicating a reversed spike order. Figure 10 illustrates the evolution of the temporal receptive field of a neuron with two inputs as the neuron attempts to learn the statistics of an underlying process that produces ISIs with linearly increasingly temporal jitter i.e. an ISI generating process with a Gaussian spike time probability distribution function whose variance increases linearly over time. The receptive field of the neuron describes the amount by which the membrane potential $\Sigma r_i(t)$ exceeds the threshold $\Theta(t)$ as a function of the input spike pattern times of $u_i(t)$. For the simple two input case illustrated, the receptive field is a scalar function of the one-dimensional ISI. In order to calculate the receptive field, following each pattern presentation the neuron's new parameters (Δr_i and Θ) were saved and the neuron was simulated repeatedly using these saved parameters for every possible ISI given the maximum pattern width PW. For each simulation the summation in Equation 6 was calculated at the end of the simulation resulting in the receptive fields shown in Figure 10.

Equation 6

$$RF_{i=2}(\tau) = \sum_{t} \left(\sum_{i} r_{i}(\tau, t) - \Theta(\tau, t) \right) \times s(\tau, t)$$

where τ is the ISI being simulated.

The ISI at which the receptive field expression above is at its maximum (RF Max) indicates the ISI for which the neuron is most receptive and may be interpreted as the ISI expected by the neuron. Similarly the ISI boundary where the receptive field expression goes to zero is the limit to the range of ISI's expected by the neuron. An ISI falling outside the receptive field boundaries results in no spike and no adaptation but simply reduces the neuron's confidence and can be viewed as outlier.



Figure 10. Tracking a hidden ISI producing process and its variance. All three panels (a, b and c) show different aspects of the same simulation where a single SKAN learns statistics of a dynamic ISI across two input channels. a) A hidden process (blue) moves from ISI=-20 Δt to ISI=0 Δt . The process begins with no temporal jitter noise, such that the observed ISI's (black dots) equal the hidden process ($\sigma=0 \Delta t$) and the blue hidden process is covered by the observed black dots. At t=0, the sum of the neuron's randomly initialized kernels peak at ISI=6 Δt . As the kernels adapt and capture the ISI information, the receptive field maximum (red line) converges on to the observed ISIs. This causes the threshold to rise shrinking the receptive field to a minimum size (left inset $t = 9.9 \text{ k}\Delta t$). At this stage receptive field boundaries (pink lines) lie very close on either side of the hidden process' mean value. As the simulation continues the noise in the ISI producing process is deliberately increased linearly with time. The neuron continues to follow the process but with every spike that doesn't land on the receptive field maximum the threshold falls slightly which increases the receptive field size and increases the neuron's receptivity to ever more unlikely observations (right inset t=138.9 k Δ t). b) Variables and behavior of the neuron throughout the simulation: after a rapid initial increase the threshold O(t) settles near the peak of the membrane potential $\Sigma r_i(t)$. As the noise increases linearly the threshold begins to fall gradually. Missing output spikes in the s(t) spike train correspond to input spikes that have landed outside the receptive field boundaries. c) The output pulse width (green = observed, red = running average) increases with increasing signal noise. As with the receptive field size, the output pulse width is initially large. As more patterns are observed, the threshold rises and settles just below the peak of the membrane potential and the pulse width reaches a minimum width of 1-2 Δt . At this low noise level ($\sigma \approx 0.5 \Delta t$) there are no missing outputs, such that all pulse widths are above zero. As the noise increases, more ISIs land away from the receptive field maximum and some fall completely outside the receptive field, decreasing the threshold, which results in wider output pulse width whenever observed ISI's do land near the receptive field maximum. The dashed magenta line tracks the mean spike width, which also increases with noise. This illustrates that the mean output pulse width of SKAN is a reliable correlate of input noise level. Figure from [9].

Figure 10 a) shows SKAN's receptive fields tracking the statistics of a moving ISI generating process with dynamic noise levels with a high level of accuracy such that the blue line indicating the hidden process is barely visible from under the red line marking the receptive field maximum. Figure 10 c) shows the neuron transmitting wider output or bursts with increasing noise. In addition, increasing ISI noise causes a growing gap between the envelope of the pulse widths and the running average of the pulse widths. This increasing gap is critical to the operation of the neuron, as it is caused by missed pattern presentations, i.e., patterns that produce no output pulse because of the presented noisy pattern being too dissimilar to the one the neuron has learnt and expects. The effect of a missed pattern is a fall in the neuron's threshold by O_{fall} . When presented with noiseless patterns this fall would be balanced almost exactly by the threshold rise due to the O_{rise} term in Equation 4 during the output pulse. However, without the output spike there is a net drop in threshold. Yet this lower threshold also makes the neuron more receptive to noisier patterns creating a feedback system with two opposing tendencies which:

- 1. Progressively narrows kernels around the observed input pattern while shrinking the neuron's receptive field by raising the threshold.
- 2. Expands the receptive field in response to missed patterns by reducing the threshold while allowing the kernels to learn by incorporating ever less likely patterns.

The balance between these two opposing tendencies is determined by the ratio Θ_{rise} : Θ_{fall} , which controls how responsive the neuron is to changing statistics. With a stable noise level SKAN's dynamics always move toward an equilibrium state where the neuron's tendency to contract its receptive field is precisely balanced by the number of noisy patterns not falling at the receptive field maximum. This heuristic strategy results in the receptive field's maximum and extent tracking the expected value of the input ISI's and their variance respectively as shown in Figure 11.



Figure 11. Evolution of SKAN's receptive field in response to input. a) Total resultant change in SKAN's receptive field after multiple pattern presentations. b) SKAN with a small initial receptive field which does not match the ISI distribution. The input spike lands outside the receptive field boundaries. c) As more ISI's fall outside the small receptive field the threshold falls and the receptive field expands, but without shifting the position of its maximum value. d) An ISI just falls on to the greatly expanded receptive field producing an output spike. e) The output spike causes the SKAN kernels to adapt shifting the receptive field toward the true position of the underlying process. f) As more and more ISI's fall closer to the receptive field maximum wider output pulses are produced which adapt the kernels faster shifting the receptive field more rapidly while the resultant rise in the threshold contracts the receptive field. With enough observations the receptive field would eventually become centered on the input ISI distribution with the receptive field boundaries tracking the ISI's distribution. Figure from [9].

Learning in the presence of Poisson spike noise and missing target spikes

In addition to robustness to temporal jitter in the input pattern an important feature of neural systems is their performance in the presence of Poisson spike noise. Recent work has highlighted that unlike most engineered systems where noise is assumed to degrade performance, biological neural networks can often utilize such noise as a resource [88], [89], [90]. In the neuromorphic context the performance of neural network architectures in the presence of noise is well documented [51], [91], [92]. To test SKAN's potential performance in stochastic real world environments, the combined effects of extra noise spikes as well as missing target spikes needs to be tested. Figure 12 illustrates how different signal to noise ratios can affect SKAN's ability to learn an embedded spatio-temporal spike pattern.



Figure 12. Learning spatio-temporal spike patterns in the presence of both Poisson spike noise and missing target spikes. Panel (a) shows the presentation of seven patterns in the middle of a simulation sequence with a noiseless environment. The kernels are highly adapted ($\Delta r_2 = \Delta r_{max}$), the threshold is high and the output spikes are narrow indicating high certainty. Panel (b) shows the result of a final noiseless test pattern at the end of the simulation showing in detail that the kernels resulting from the test pattern peak at the same time. Panel (c) shows the same interval of the same simulation as panel (a) but with a 1:1 signal to noise ratio where the probability of a target spike being deleted is half or P(signal)=0.5 and the Poisson rate is also half such that P(noise) = 0.5/T. Panel (d) shows the result of a noiseless test pattern presentation at the end of the simulation. The increased level of noise has resulted in an incorrect ramp step (Δr_2) such that the r_2 kernel peaks slightly late (black arrow). Panel (e) shows a simulation with a 1:2 signal to noise ratio. Panel (f) shows that the high noise level has resulted in slight misalignment of all four kernels. Figure from [9].

To quantify the performance of SKAN in the presence of Poisson noise and missing target spikes a series of simulations each comprising of two thousand pattern presentations were performed. At the end of each simulation the RMS error between the neuron's receptive field maxima and the random target pattern was measured and is shown in Figure 13.



Figure 13. RMS error between receptive field maxima and target spike patterns as a function of spike signal to noise ratio. The three bottom panels show the spike probability distributions at three points along the SNR axis. The signal spikes (blue), missed spikes (grey), and noise spikes (red) are illustrated for the three cases of 1:0, 1:1 and 1:2 signal to noise ratios. The mean spike rate was maintained at 1 spike per channel per time period between pattern presentations T. At the completion of a simulation with one thousand pattern presentations the RMS error between the resulting receptive field maxima and the target spatio-temporal pattern was calculated. As the plot illustrates the error increases with noise and simulations of neurons with more input channels resulted in higher error. Figure from [9].

These results in this chapter show that even with the presence of significant corruptive environmental noise (additive noise spikes as well as non-target patterns), and noise in the signal (temporal noise in the signal as well as missing spikes), the neuron is able to develop a robust model of the target spatio-temporal spike pattern.

Chapter 3: SKAN at the network scale

Multi-SKAN classifier

In order to extend a single learning neuron to a classifier network it is important that different neurons learn different patterns. Ideally a neuron in a layer should not be in anyway affected by the presentation of a pattern that another neuron in the same layer has already learnt or is better placed to learn.

As outlined in Equation 3, SKAN adapts its kernels only during an output pulse. This rule is particularly conducive to competitive learning such that the simple disabling of the neuron's spiking ability disables all learning. Whereas previously proposed algorithms utilize multi neuron Winner-Take-All layers with real valued rate based inhibitory signals to prevent correlated spiking and maximize the network learning capacity [93], [94], in a SKAN network a simple global inhibitory OR gate serves the same function. The reason a simple binary signal can be used here is that in a SKAN network the best-placed neuron for any pattern will be the fastest neuron to spike. This allows a layer of neurons with shared inputs to learn to recognize mutually exclusive spatio-temporal patterns. To this end, Equation 2, describing the neuron's output, is replaced by Equation 7 (underlined terms added). The addition of a global decaying inhibitory signal as described in Equation 8, act on all neurons to disable any rising edge at the output. This means that neuron *n* can only *initiate* an output spike *s_n* if no other neuron has recently spiked, i.e., the inhibitory signal is inactive (*inh*(*t*-1)=0) and it can only *continue* spiking if it was already spiking in the last time step (*s_n*(*t*-1)=0).

Equation 7

$$s_{n\left(t\right)} = \begin{cases} 1 & \text{if } \sum_{i} r_{n,i\left(t\right)} > \Theta_{n\left(t-1\right)} \ \land \ \left(inh(t-1) = 0 \ \lor \ s_{n\left(t-1\right)} = 1\right) \\ 0 & \text{else} \end{cases}$$

Equation 8

$$inh(t) = \begin{cases} inh_{max} & \text{if } \bigcup_n s_n(t) = 1\\ inh(t-1) - inh_{decay} & \text{if } inh(t-1) > 0\\ 0 & \text{else} \end{cases}$$

As shown in Figure 14 and described in Equation 8, the inhibitory signal is realized via an OR operation on the output of all neurons, and a decaying behavior which keeps the inhibitory signal active for a period of time after a neuron has spiked to prevent spiking by other neurons. After the output spike ends, this feedback loop decays from inh_{max} by inh_{decay} at each time step until reaching zero at which point the global inhibitory signal turns off allowing any neuron to spike. As shown in Figure 14 the decay only begins at the end of the pulse making the inhibitory signal operate as a global peak detector which stays at inh_{max} for the duration of the pulse, ensuring that the inhibitory signal robustly suppresses spiking activity for a wide range of potential output pulse widths.



Figure 14. A single global decaying inhibitory signal suffices to push apart the neurons' receptive fields and decorrelate the spiking of the SKAN network. Left panel shows the network diagram of four neurons with an inhibitory signal. The decay feedback loop extends the duration of the inhibitory signal beyond the initial triggering spike via the *inh(t)* signal using a counter and a comparator in the decay block. The right panel shows the simulation results from a two input two neuron network learning to classify two ISI's x and y. The sum of the randomly initialized kernels of neuron one (dashed) happen to peak earlier than neuron two (solid) so that neuron one fires first in response to the first pattern (x with ISI=0 Δt). During this first output pulse neuron one's threshold rises sharply reducing its receptivity, while its kernel step sizes adapt towards each other such that $\Delta r_{1,1} \approx \Delta r_{1,2}$. Meanwhile the inhibitory signal blocks neuron two from spiking when its kernel sum exceeds its threshold only a few time steps after neuron one, which means the neuron is prevented from adapting to pattern x. At the second pattern presentation pattern y is shown (ISI=10 Δt). For this pattern the sum of the kernels of the second neuron, still unchanged from their random initialization, reach that neuron's threshold slightly earlier than neuron one and so neuron two spikes and begins adapting to pattern y. A subsequent presentation of pattern x again triggers neuron one and the kernels of the two neurons increasingly fine tune themselves to their respectively chosen pattern as with each presentation their thresholds rise. This fine-tuning process reduces the receptivity of the neurons to each other's patterns. Figure from [9].

As with the single neuron output rule, the single neuron threshold adaptation rule of Equation 4 can be modified to Equation 9 (underlined terms added) to utilize the global inhibitory signal for the multineuron case. This modification prevents a neuron's threshold being affected by the presentation of patterns that another neuron is better adapted to. The addition of the underlined terms in the first line of Equation 9 means that a neuron's threshold can only rise when its membrane potential exceeds its threshold and the inhibitory signal is not already active, or if the neuron itself spiked in the previous time step. The fall in the threshold is similarly conditioned on the neuron having spiked *before* the global inhibitory signal was activated, such that only the very best adapted neuron, i.e., the one that generated the inhibitory signal in the first place, adapts its threshold.

Equation 9

$$\Theta_{n}(t) = \begin{cases} \Theta(t-1) + \Theta_{rise} & \text{if } \sum_{i} r_{i}(t) > \Theta(t-1) \xrightarrow{\wedge} (inh(t-1) = 0 \lor s_{n}(t-1) = 1) \\ \Theta(t-1) - \Theta_{fall} & \text{if } \left(\sum_{i} r_{i}(t) = 0 \land \sum_{i} r_{i}(t-1) > 0 \xrightarrow{\wedge} inh(t-1) = 0\right) \lor (s(t) = 0 \land s(t-1) = 1) \\ \Theta(t-1) & \text{else} \end{cases}$$

Such a global inhibitory signal has been utilized in LIF neurons [95], [96] and synaptic weight STDP neurons as a means of decorrelating neuronal firing patterns [97], [98]. Here, however, its use is subtly different from both. Although in LIF and synaptic STDP architectures and in SKAN a global inhibitory signal results in the decorrelation of output spikes, in the purely synaptic weight adapting schemes the neuron's response time remains static and does not improve with adaptation and in the LIF networks [95], [96] there is no lasting adaptation at all. SKAN's improved response time due to kernel adaptation and the global inhibitory signal realize a positive feedback mechanism absent in previous models. In a SKAN network a neuron's small initial advantage for a pattern results in a slightly earlier output spike. This output spike globally inhibits all other neurons, which in turn results in exclusive adaptation to the pattern by the first spiking neuron. This further improves that neuron's response time for the pattern and increases the likelihood of the neuron being the first to spike due to a subsequent presentation of the same pattern, even in the presence of temporal jitter. Thus, the adaptation of SKAN's kernels and thresholds, together with the global inhibitory network, mean that the neuron whose initial state is closest to the presented pattern will be the first to respond and prevent all other neurons adapting to this pattern. This effectively "hides" the pattern from the other neurons and allows unsupervised spike pattern classification by the network as whole as demonstrated in the proceeding results sections.

There are two important constraints adhered to by the preceding modification of the SKAN rules. The first constraint is that the required connectivity does not increase combinatorially with the number of neurons as described in Equation 10 since the only feedback path is from the single global inhibitory signal.

Equation 10

total connections = (number of input channels + 2) × number of neurons

The second constraint is that no complex central controller is required for arbitration between the neurons. In competitive neural network schemes where a neuron's fitness is expressed as a real value from each neuron to a Winner-Take-All network, multiple bits (connections in hardware) are required to transport this information. Alternatively rate based systems encode such real valued signal over time in their spike rate which are then utilized by a corresponding rate based Winner-Take-All system. But in SKAN these requirements are reduced. Since a neuron's latencies correlates with its adaptation to a target pattern, the neurons do not need to report a real value but only a single bit. This mode of operation can be interpreted as either a connectivity saving or as a speed saving with respect to alternative multi-bit or rate based systems respectively. Furthermore because of the robustness of the system, checking for, or prevention of, simultaneous output spikes is not necessary. Random initial heterogeneities in the neurons' parameters and/or noise in their signals is enough to eliminate the need for central control by pushing the neurons away from input space saddle points toward their stable non-overlapping receptive fields.

Online unsupervised spatio-temporal spike pattern classification

In the following sections the classification performance of SKAN is tested in several ways. For these tests equally likely spatio-temporal spike patterns, each with one spike per channel per presentation were

presented in random sequences to the SKAN network. Table 1 details the parameters used in all the tests. These parameters were deliberately chosen for non-optimized performance so as to try to mimic the use of the system in the wild by a non-expert user. Examples of available optimizations include: higher *ddr* values which result in faster converging systems, reduced $\Theta_{rise} / \Theta_{fall}$ ratio for improved robustness to noise, increased $\Delta r_{max} / \Delta r_{i,n}(t=0)$ ratio and increased pattern widths for enhanced pattern selectivity.

Parameter	Value	Description		
ddr	1	Change in the kernel step size. Higher value results in		
		faster adaptation; lower values are more robust to noise.		
w(r _{max})	10000	Maximum kernel height (Synaptic weight).		
r _{min}	0	The kernel signal $r(t)$ saturates at zero.		
$\Delta r_{i,n}(t=0)$	100×(1+ rand)	Initial kernel step size (For each input <i>i</i> to each neuron <i>n</i>).		
		The randomized initialization allows different neurons to		
		learn different patterns.		
Δr_{max}	400	Maximum kernel step size.		
$r_{i,n}(t=0)$	0	Initial kernel value.		
Θ_{rise}	40 × inputs	Rise in threshold during output spike, where inputs is the		
		number of input channels per neuron.		
Θ_{fall}	100× inputs	Fall in threshold due to input spikes, where inputs is the		
		number of input channels per neuron.		
inh _{max}	100	Initial value of the inhibitory countdown .		
inh _{decay}	1	Step size of the inhibitory countdown. As a rule of thumb		
		use: $inh_{max}/inh_{decay} = min(\Delta r_{i,n} (t=0)).$		
Т	400 <i>∆t</i>	Time between pattern presentations		

Table 1. Parameter values used for all SKAN results.

1-to-1 neuron to pattern allocation

Through temporal competition a local network of mutually inhibiting SKANs can efficiently distribute limited neural resources in a hardware implementation to observed spatio-temporal patterns as is demonstrated in Figure 15.



Figure 15. Convergence toward a stable one to one neuron to pattern allocation as a function of neuron/pattern numbers and number of pattern presentations for a 1-to-1 two input neuron to pattern allocating network. As the number of patterns/neurons increases the system requires longer pattern sequences to correctly allocate exactly one unique pattern to each neuron. The inset shows the five consecutive correct classifications of four patterns by four neurons. Figure from [9].

Similar to biological systems, in a SKAN network there is no supervisor switching the network from a training mode to a testing mode so there is no distinction between learning and recognition. Thus to test the network's performance 1000 simulations were generated for each instance of the network, with up to 800 pattern presentations each. The network was considered to have converged to a stable solution when twenty consecutive patterns were correctly classified by the network, i.e., with a single neuron responding per spatiotemporal pattern. This is illustrated in the inset of Figure 15. Correct classification was defined as the case where a neuron spikes if and only if its target pattern is presented and where the neurons consistently spike for the same learnt target pattern. Also, a single neuron should spike once for each input pattern and no extra output spikes occur. The percentage of simulations that had not converged to correct classification was recorded as a function of the number of patterns presented, and is shown in Figure 15. Simulations were terminated once a network had converged. The number of consecutive patterns was chosen as twenty to reduce the likelihood that the observed "correct" response of the network was due to chance.

Classification performance as a function of spatio-temporal pattern dimension.

The problem of coordinating multiple synapses for unsupervised neuronal classification in spiking neural network models, whether through simply learning synaptic weights or through more complex pathways, is difficult [99]. In SKAN the hybrid synapto-dendritic kernel adaptation produces convergence profiles shown in Figure 16. These results show how the convergence profiles of SKAN change with the number of active input channels. Additionally, the right panel in Figure 16 shows the effect of increasing the

resolution of the spatio-temporal pattern. Doubling the number of time steps in the maximal width of the target pattern *PW*, results in improved convergence profiles.



Figure 16. Convergence rates as a function of input channel dimension and pattern width. Left panel: two random target patterns (light and dark bars) of maximal pattern width $PW = 20 \Delta t$ and of dimensions 2, 4, 8 and 16, were presented at random to a two neuron network, with the convergence of simulations plotted over the number of presentations. Right panel: the same test with maximal pattern width $PW = 40 \Delta t$. Figure from [9].

Classification in the presence of temporal noise

In order for SKAN to operate as an effective classifier competing neurons must balance the requirements of selectivity and generalization. In the spatio-temporal context, generalization takes the form of temporal jitter noise. In this context neurons must recognize patterns closest to their learnt target pattern despite the presence of temporal noise, while not recognizing other similarly noise corrupted patterns that are closer to the target patterns learnt by other neurons. Furthermore, the neurons should not expect the learning phase to be any less noisy than the testing phase or even for there to be any such distinct separation between learning and recognition. As well the neurons should maintain their correct learning and recognition behavior across a wide range of noise levels and they should ideally do so without the requirement for external adjustment of their parameters. SKAN satisfies all these requirements. The classification performance of SKAN is robust to temporal jitter noise as illustrated in Figure 17 where two neurons with shared inputs attempt to learn the statistics of two noisy but distinct ISI generating processes.



Figure 17. Convergence as a function of temporal noise and pattern presentations in a two neuron network with two input channels. The insets illustrate selected noise distributions relative to the maximal pattern width (PW = 20 Δt). The left panel shows convergence profiles due to temporal noise distribution with standard deviation $\sigma = 0.1 \Delta t$. At lower noise levels the convergence profile is approximately the same or faster (red) than the zero noise case. The right panel shows the same for $\sigma \approx 1.3 \Delta t$. Figure from [9].

Because of the constant adaptation of the neurons, moderate levels of temporal noise with standard deviation up to $\sigma = 0.25 \Delta t$, which is 1/80th of the pattern width, either do not affect or actually *improve* SKAN performance. With high temporal noise levels, i.e., with a standard deviation that is 1/20th the width of the pattern ($\sigma=1 \Delta t$), the convergence profile is still similar to that of the noiseless case. Such levels of temporal noise can disable a conventional processor and even some neural networks. Even at the extreme, with noise that has a standard deviation more than 1/7th of the pattern width, some simulations still result in the neurons correctly classifying the separate ISI sources.

As a temporal coding neuron model, the robustness of SKAN to temporal noise shown in the previous results is critical for potential real-world applications, where the ability to operate (and degrade gracefully) in noisy, dynamic environments is favored over ideal performance in ideal noise free circumstances. Furthermore the fact that the model maps neurons and spatio-temporal patterns in a one to one manner allows implementation of small yet robust single layer spiking networks in hardware.

Chapter 4: SKAN and STDP

Synaptic weights as independent encoders of input noise

As detailed in the previous chapter in addition to SKAN's kernel adaptation, which captures the temporal information of input spatio-temporal spike patterns, the neuron's threshold also adapts: At every time step during an output pulse the threshold rises and every time the membrane potential returns to zero the threshold falls. As the neuron spikes more, its threshold rises, making the neuron ever more selective for the pattern that triggered it and narrowing its spatio-temporal receptive field. Conversely, unrecognized input patterns, which do not cause an output spike, reduce the threshold, making the neuron more receptive to new patterns. Through this feedback mechanism the neuron automatically maintains a balance between selectivity and generalization in response to the statistics of its environment.

In this way the threshold level of SKAN is an encoding of the overall input noise. However this means that the neuron model has only a single measure for the noisiness of its inputs and thus implicitly assumes that all its inputs have an equal noise level. This assumption can indeed be made in many applications yet there are also circumstances where the heterogeneity of input noise is significant. An illustrative example is an event based neuromorphic experimental camera such as the event based Dynamic Vision Sensor (DVS) camera [100] where a very small number of pixels <1% can have a very high rate of Poisson corrupted spikes. These noisy inputs and their erroneous high activity can play havoc with downstream event based tracking and recognition algorithms. In such an application an unsupervised feature extractor that was also able to weigh its input channels based on their signal to noise ratio is useful. This serves as a motivation for and test of the augmentation of SKAN with a simplified Spike Timing Dependent Plasticity (STDP) rule. By augmenting SKAN with STDP the neuron model is enhanced so that each synapse adapts independently to its own noise environment with noisier input channels acquiring a lower synaptic weights and noiseless inputs acquiring higher synaptic weights. Results show that this system significantly outperforms the statically weighted SKAN model in environments where the inputs have different noise levels.

Simplified weight update rule



Figure 18. Comparison of the classical STDP synaptic weight update curve with simplified kernel based STDP. a) In the classical STDP update rule, a synapse has its highest weight increase if it receives a presynaptic input spike just prior to a postsynaptic output spike. Conversely a postsynaptic output spike that precedes a presynaptic input spike causes the greatest decrease in synaptic weight. Both effects decay with longer inter-spike intervals. b) In the simplified model used in this thesis, the adaptable EPSP kernel not only learns the commonest pre/post synaptic spike interval, but also doubles as a flag that enables the increase in synaptic weights in the event of a postsynaptic spike. Figure from [46].

In biology the rules governing synaptic weight adaptation vary enormously both in degree and in type across species, brain regions, synapse types, cell types, within individual cells, over short time scales, and as a function of organism development [101]. However, the STDP rule shown in Figure 18 (a) is by far the best studied synaptic plasticity rule in neuroscience today due to its reproducibility and neurocomputational utility in selectively strengthening synapses such that given a large enough number of heterogeneous synapses with different intrinsic delays, any arbitrary spatio-temporal pattern can be learnt by a single neuron [102]. As a consequence, the faithful modeling of this rule in hardware is now a major focus in neuromorphic engineering [24]. In this model of synaptic plasticity the strengthening or weakening of synaptic transmission efficiency is typically modelled by an exponential decaying function of the inter-spike interval between the pre- and post- synaptic spikes as shown Figure 18 (a). As with the smooth synaptic alpha function, such accurate modelling of neurobiological processes can incur additional hardware costs while providing little computational improvement compared to even highly simplified models [93]. Therefore, as was the case with SKANs simplified kernels, in this chapter, the classic STDP rule is replaced with the simplified weight update rule shown in Figure 18 (b). The rule is designed so that it reuses the same signals and flags that are already present in the static weight SKAN system, such that the presynaptic input spike $u_i(t)$, which triggers the EPSP kernel $r_i(t)$, also triggers a binary weight adjustment flag $d_i(t)$. If a back-propagating postsynaptic output spike, s(t), arrives while this flag is high, then the synaptic weight $w_i(t)$ is increased by w_{rise} and the flag is reset to zero. Alternatively if the membrane potential $\Sigma r_i(t)$, returns to zero before an output spike arrives, then the synaptic weight is decreased by w_{fall} and the flag is returned to zero. These two rules are described in Equation 11 and Equation 12.

Equation 11

$$w_i(t) = \begin{cases} w_i(t-1) + w_{rise}, & \text{if } d_i(t-1) \land \downarrow s(t) \\ \\ w_i(t-1) - w_{fall}, & \text{if } d_i(t-1) \land \downarrow \Sigma r_i(t) \end{cases}$$

Equation 12

$$d_i(t) = \begin{cases} 1, & \text{if} & u_i(t) \\ 0, & \text{if} & \downarrow \Sigma r_i(t) \lor \downarrow s(t) \end{cases}$$

Where $\downarrow s(t)$ is the falling edge of the postsynaptic output spike, $\downarrow \Sigma r_i(t)$ is the return of the membrane potential to zero ($\Sigma r_i(t-1) > 0 \land \Sigma r_i(t) = 0$) and $u_i(t)$ is the presynaptic input spike.

As a result of Equation 11 and Equation 12, every time the membrane potential rises due to input spikes, the synaptic weights of the activated inputs either rise in response to the resultant output spike or they fall when the membrane potential returns to zero.

This simplification of the STDP model significantly reduces hardware costs. By using the EPSP kernel as a binary flag of adaptable duration, the need for realization of the exponentially decaying function of Figure 18 (a) is eliminated and the use of the constant update terms w_{rise} and w_{fall} replaces the addition of two arbitrary values, $w_i(t)$ and $\Delta w_i(t)$, which would otherwise be required at each synapse and which is significantly more costly in terms of hardware resources in comparison to constant terms which can be hardwired.

Additionally unlike in classical STDP, in the proposed model, if an output spike were to be triggered just after the membrane had returned to zero (say by an external stimulator) there would be no change to the synaptic weight, but in the normal operation of the neuron, this simplification of the model does not affect the overall behavior. Similarly, in the SKAN model multiple input spikes that arrive within a short time or in bursts are covered by the EPSP kernel of the leading spike and are invisible to the system. This reduction is arguably desirable as it correlates well to real world event driven stimuli where relative stimulus onset times across afferents carry salient information.

Synaptic weight normalization without division

An additional layer of complexity arises through the need for synaptic homeostasis which is required to keep the synaptic weights within some limited dynamic range while preserving their *relative* strengths. In the biological context a number of homeostasis models have been proposed [103][104]. The common feature of these models is divisive normalization where the strength of all synapses in a neuron are rescaled via division by a global signal which keeps all synaptic weights within their physiological dynamic range while preserving their relative transmission efficiency. The implementation of this normalization operation in digital hardware again involves multiplication operation at each synapse.

In this section we propose a novel digital approach to this problem that eliminates the need for this multiplication. The synaptic weights and other variables in the neuron all of which are implemented using unsigned integers are normalized not via multiplication but bit shift operations. Here, instead of normalizing the synaptic weights such that the max of the weights, $\max(w_i(t))$, or the sum of the weights, $\sum w_i(t)$, is clamped to a specific value, the max signal is allowed to vary within the top half of a digital range, updated by the weight update rule of equation 1 and equation 2. When the update rule pushes the max($w_i(t)$) signal beyond this digital range such that the *overflow*($w_i(t)$) signal goes high for *any* of the synapses, then all the neuron's signals, i.e., the weights $w_i(t)$, EPSP kernels $r_i(t)$, EPSP kernel step-sizes $\Delta r_i(t)$ and the threshold $\Theta(t)$, are right shifted, (division by 2). Conversely if the signal falls below half its range such that the most significant bit of all the synapses is zero, then all the signals of the neuron are left shifted (multiplication by 2). These two conditions simplify to Equation 13 and Equation 14.

Equation 13

 $\bigcup_{i} overflow(w_{i}(t)) \rightarrow right shift (w_{i}(t), r_{i}(t), \Delta r_{i}(t), \Theta(t))$

Equation 14

 $\bigcap_{i} MSB(w_{i}(t)) = \mathbf{0} \rightarrow left shift \ (w_{i}(t), r_{i}(t), \Delta r_{i}(t), \boldsymbol{\Theta}(t))$



Figure 19. Comparison of fixed point division and shift based normalization for encoding the relative strengths of synaptic weights within an 8-bit dynamic range. The two synapses with weights w_1 and w_2 are updated by two independent non-zero mean stochastic processes. The panels on the left show the encoded weights while the panels on the right show the relative weights with the strongest synapse normalized to 1 (calculated via floating point division). a) The original weights of w_1 and w_2 with no bound on their dynamic range: $w_i(t+1) = w_i(t) + \Delta w_i(t)$. b) The true relative strengths of the original synaptic weights. c) Limiting the two weights to between 0-255 (8-bit unsigned integer) via fixed point division: $w_i(t+1) = w_i(t) + \Delta w_i(t) / \max_i(w_i(t) + \Delta w_i(t))$. The black and white bars indicate the top and bottom half of the digital range (0-127 and 128-255). d) Relative strength of the bounded 8-bit synaptic weights. The error plot shows RMS error with respect to the original relative strengths shown in b). e) 8-bit Shift based normalization showing the stronger signal w_1 triggering shifts in both synapses as it exceeds the bounds of the top half of its digital range: $w_i(t+1)=f(w_i(t) + \Delta w_i(t))$. Where f() denotes equations 3 and 4. f) The relative synaptic strengths encoded via shift based normalization and the associated error. Figure from [46].

The fact that *all* neuronal parameters, w_i (t), r_i (t) and O(t) are also shifted means that the neuron is essentially not affected by the shift operations. As demonstrated in Figure 19 for a two synapse neuron, the overall effect of the shifting operations described is to continuously generate more dynamic range such that all weights become normalized while the max signal remains within the range described by Equation 15:

Equation 15

$$2^{b-1} \le \max(w_i(t)) \le 2^b - 1$$

where *b* is the number of bits used to represent the synaptic weights.



Figure 20. RMS error of shift based normalization with respect to normalization via double-precision floating-point division. Random synaptic weight updates with $P(w_{rise}) = P(w_{fall})$ were performed on a simulated 16 synapse neuron with synaptic bitwidth of 8 to 16. The RMS error of the relative weights of the shift based synapses was calculated against synapses which were normalized via double-precision division. Increasing bit-width in the shift based synapses resulted in lower error but even at the lower bit-lengths the relative order of the synaptic weights followed the floating-point implementation. Figure from [46].

Figure 20 shows the error introduced in the relative strength of synaptic weights through the use of shift based normalization with varying bit-widths. In addition to the quantization noise introduced, an important edge case occurs in the 'right shift neuron values' operation, which requires a design decision in terms of any weak synapses which go to zero. One option is to not allow any weights to go to zero. This can be implemented either by checking all bits of every synapse and setting to 1 any that go to zero or simply by assuming the LSB of all synapses is set to high without any zero checking. Another option in dealing with synaptic weights that go to zero is to disable them completely. This can potentially allow reallocation of the synapses to other neurons. An application of this option is discussed in chapter 5.

STDP and SKAN combine to produce synaptic encoding of afferent Signal to Noise Ratio (SNR)

Given static synaptic weights $w_i(t)=w_i(t_0)$, the simple kernel adaptation of SKAN can perform unsupervised learning of common spatio-temporal patterns in noisy environments as detailed in chapter 2 and 3. When this static weight model of SKAN is combined with the synaptic weight update and the normalization operation of the previous section, the neuron not only finds and learns the most common spatio-temporal pattern, but additionally adjusts its synaptic weights independently to compensate for the signal-to-noise ratio of individual afferents and thus improves recognition performance. In this section the corruptive noise is defined as additive noise spikes generated by a homogeneous Poisson process with rate λ/T . The signal represents spatio-temporal spike patterns which are presented every T time steps as shown in Figure 21. The SNR is thus defined as the ratio of the target pattern presentation rate 1/T, and the noise process rate λ/T which results in Equation 16:

Equation 16

$\textit{SNR} = 1:\lambda$

To demonstrate the SNR encoding effect, consider the case where the neuron is presented with repeated spatio-temporal spike patterns that are received via noise corrupted afferents. After several pattern presentations the neuron's kernels 'see through' this noise and adapt their slopes $\Delta r_i(t)$ so that they align with the pattern. This is because the noise is uncorrelated with the pattern and it is just as likely to increase the slopes as it is to decrease them such that the noise is averaged out, leaving only the target pattern for the kernels to train on. Figure 21 shows what happens next for a case where one of the three afferents is corrupted by SNR=1:1, that is, where the probability of the presence of a Poisson noise spike during any time period equals the probability of a target spike belonging to the target spatio-temporal pattern.



Figure 21. Synaptic weight adaptation in a neuron with three synaptic inputs. The first input channel (afferent₁) is noisy (SNR=1:1) while the other two input channels (afferent_{2:3}) are noise free. Input spikes trigger both the kernel $r_i(t)$ and the weight update flag $d_i(t)$. The output spike s(t) causes a rise in the synaptic weight of all synapses by w_{rise} , while the return of the membrane potential to zero induces a fall by w_{fall} , in the activated synapses with $d_i(t)=1$. During the time interval shown the noisy afferent with weight w_1 experiences two weight falls and one weight rise while the two noise free afferents experience only one weight fall and one weight rise. As a result of many such adjustments w_1 falls to its steady state value of approximately half of w_2 and w_3 . The lower weighted synapses contribute less and less to the membrane potential. The full dynamic ranges of the synapses are indicated by the grey regions marked afferent_{1:3}. Figure from [46].

Noise spikes, being uncorrelated with the target pattern and with each other, typically arrive on their own or in such a way that their EPSPs are not enough to push the membrane potential $\Sigma r_i(t)$ past the threshold $\Theta(t)$ to cause a postsynaptic output spike. Such noise spikes do however reduce their respective synaptic weight by w_{fall} . In the case where a clean target spatio-temporal pattern arrives without any neighboring noise, all the synaptic weights are increased equally by w_{rise} . The combined result of these changes is that synapses that receive target input spikes more often accumulate higher and higher weights while synapses with greater noise spikes have their weights pushed down. Over time the forces pushing a synaptic weight up (postsynaptic spikes following presynaptic spikes), and down (presynaptic noise spikes without a postsynaptic output spike) come into balance with each other and

the weight normalization produced by the neuron's shift operations. If the Poisson processes generating the noise spikes is homogeneous, i.e. constant over time, then the synaptic weights converge to a steady state value which encodes (and compensates for) the relative signal to noise ratio of each afferent as shown in Figure 22.



Figure 22. Mean normalized synaptic weights w_i as a function of noise spike rate showing the synaptic weights of a three synapse neuron encoding the *relative* level of input SNR such that the noisier afferents receive a lower weighting than less noisy afferents. Here SNR=1: λ_i and λ_i is the noise spike generation rate at the *i*th input. The neuron was receiving a random spatio-temporal target pattern corrupted with varying levels of Poisson noise for each of the three input channels. In all panels the Poisson noise rate across the first two channels, λ_1 and λ_2 , was kept constant while λ_3 was swept from SNR=0 to SNR=1 and the average steady state value of the synaptic weights, w_1 , w_2 and w_3 are plotted. Figure from [46].

To demonstrate how the synaptic weights of a neuron evolve to their steady state over time, a sixteen input neuron with half its afferents corrupted with noise is shown in Figure 23. Here, the neuron's input and kernel signals have been removed to more clearly show the relative synaptic weight encoding over time. Additionally the plot shows the neuron's membrane potential $\Sigma r_i(t)$ leveling off at a low steady state value with lower noise and more consistent output spikes S(t).



Figure 23. Time series plot showing the evolution of synaptic weights of a sixteen input neuron from an initial equal value their steady state. Half the inputs have a noise spike rate $\lambda_{9:16}=1/2$, (SNR=1:1/2) while the other half are noise free $\lambda_{1:8}=0$. The equally sized rectangles indicate the equal normalized dynamic range of the synaptic weights while the sixteen indexed plots show the weight of each synapse $w_i(t)$. The noise free synapses, $w_{1:8}(t)$ are all nearly equal and at the top of the normalized dynamic range. The weight of the noisy synapses $w_{9:16}(t)$ fall to approximately half that of the noise free synapses where they remain in a steady state in response to the noise environment. By lowering the synaptic weight of the noisy afferents the neuron reduces their contribution to the somatic membrane potential making the later a less noisy and thus more useful signal during recognition. Figure from [46].

Recognition performance on noise corrupted spatio-temporal patterns

To quantify the recognition performance of the neuron under various noise regimes, the recognition error of the synaptic weight adapting neuron was measured against a neuron without synaptic weigh adaptation. Both neurons were presented with a random sequence one thousand patterns populated by two random spatio-temporal patterns. For each test some of the input channels were noise corrupted at varying levels in the same manner as illustrated in Figure 23. The neurons were given one hundred presentations of this noisy randomized data stream within which to perform unsupervised learning of one of the two random patterns after which the pattern for which the neuron spiked most was designated as its target pattern. In the following nine hundred presentations the error in recognition was measured, defined as the number of missed target patterns plus false positive output spikes divided by the total number of target presentations. Figure 24 demonstrates the effect of the SNR

encoding synapse, where over a wide range of noise environments the neuron effectively removes all corrupting input noise and delivers near perfect unsupervised learning and recognition performance. An interesting feature of the neuron is the initial increase in error at the low noise level for the weight adapting neuron, where a minimum noise threshold must be reached to trigger the weight adaptation system. For these tests the w_{rise}/w_{fall} ratio was deliberately chosen to clearly illustrate this behavior. This initial rise can be brought down by choosing a larger w_{fall} term making the neuron more aggressive in terms of shutting down noisy afferents.



signal to noise ratio per noisy channel

Figure 24. Enhanced recognition performance via synaptic signal to noise ratio encoding. (top) Recognition error as a function of increasing noise in a kernel adapting neuron with static synaptic weights. The four plots demonstrate increasing recognition error rate both as the number of noisy channels increase and as the SNR per noisy channel deteriorates. Note that as long as the noise corrupted channels are few in number the static SKAN can provide a moderate level of unsupervised recognition performance. (bottom) The same noise regime being applied to the same neuron this time with the dynamically adaptive synaptic weights (note the change in scale for the vertical axis). After perfect performance in the noiseless environments the error rates raise rapidly (SNR = 1:0 - 1:0.25). The reason for this initial rise in error is that the relative level of noise is simply too low to trigger the neuron's weight adaptation system such that the recognition profile is almost the same as for the static weighted neuron. As the noise level increases the neuron's SNR encoding system switches off the noisy channels and the recognition performance returns to near perfect. At very high noise levels (SNR < 1:3), the error rate begins to rise again, this time because the neuron's learning of its "target pattern" during the unsupervised learning period begins to deteriorate. Figure from [46].

Chapter 5: Hardware implementation and application

In this chapter the FPGA resource usage of SKAN and the combined SKAN + STDP models are detailed and results from an example application of the combined model is presented.

Implementation in FPGA hardware

To evaluate the hardware requirements of the proposed combined SKAN-STDP system. The system was implemented in an Altera Cyclone-V GX FPGA, a low-end FPGA containing 77,000 programmable logic elements (LEs) in Verilog hardware description language which was generated using Mathworks Simulink HDL coder. There were no approximations used in the FPGA implementation as the ideal neuron model is deterministic and was designed using unsigned integer operation. Accordingly all signals in the FPGA and software implementations were determined to be identical. Figures in this thesis were generated using software simulations of the model which was identical to that realized in hardware. The model's synaptic and somatic parameters were implemented using 12-bit and 20-bit unsigned integers respectively which is at the high end of synaptic signal precision [42]. The hardware usage of the two proposed models is presented in Table 2 for the one neuron case.

Synapses/Neuron		1	2	4	8	16
Single bit Registers	SKAN	38	62	110	206	398
	SKAN + STDP	62	102	182	342	662
Logic utilization (in ALM)	SKAN	69	102	179	358	693
	SKAN + STDP	136	285	521	1004	2030

Table 2. Altera Cyclone V FPGA resource usage for the static weight SKAN and dynamic weight SKAN models, with varying number of synapses. Table from [46]

*An Adaptive Logic Module (ALM) is equivalent to 2.65 logic elements.

For the static weighted SKAN system additional synapses require 24 single bit registers and approximately 50 ALMs each. For the combined SKAN-STDP system additional synapses require 40 single bit registers and approximately 150 ALMs each. To provide a comparison to the resource usage of the 12-bit SKAN and SKAN-STDP synapses, a single 12-bit unsigned multiplier and its input registers were synthesized on the same device. The resource usage of the multiplier consisted of 24 single bit registers and 65 ALMs. While utilizing approximately twice the hardware resources of a single multiplier of the same precision, the SKAN-STDP synapse is capable of learning an arbitrary spike delay, generating a membrane potential at the soma, encoding its input signal to noise ratio in its synaptic weight and normalizing the dynamic range of its signals.

In this context of hardware resource efficiency, a direct comparison of SKAN to other neuromorphic hardware solutions is complicated by the multi-subsystem nature of such systems where the resource requirements for the unsupervised feature extractor subsystem is difficult to establish in isolation. In addition the computational tasks performed by the feature extractor subsystem tends to vary for each solution and is difficult to quantify. Even in closed loop, supervised neuromorphic systems where the need for comparison has motivated recent progress on hardware performance benchmarking, the development of unbiased results is challenging [105]. Here the digital multiplier serves as an indirect method of comparison of the presented SKAN implementation to other digital implementations of

online synaptic learning in spiking neuromorphic systems. Since the realization of fixed point synaptic kernels weighted by fixed point synaptic weights would require at least a synaptic multiplication operation, the resource requirements of the digital multiplier serves as an approximate lower bound on the resources required to achieve the same synaptic resolution.

Example application: Unsupervised feature learning using a camera with noise corrupted pixels

Cameras can often suffer from noisy pixels and experimental or neuromorphic cameras are especially prone to this problem. Cameras such as the event based DVS camera [100] can suffer from faulty pixels which generate noisy streams of pulses where there should be no activation and this can have a detrimental effect on upstream recognition systems. Such faults can require on-going examination of the camera by an expert user in order to detect and remove such noise corrupted pixels. Here a synaptic SNR encoding neuron is particularly useful in being able to simultaneously perform both the noise removal and the online unsupervised feature learning task.

To demonstrate the concurrent learning of common features in a sequence of spatio-temporal patterns and block noisy inputs, the neuron was presented with a subset of the zero images in the MNIST handwritten digit dataset which is commonly used for training image processing systems [106]. The pixels in the dataset images were directly mapped to the synaptic inputs of the neuron. However since the neuron receives spikes as inputs, the pixel values needed to be converted to spikes. The simplest approach, which was used here, involved mapping intensity to spike latency, with the brightest pixels arriving first and the darkest arriving last. This transformation was performed by a simple one to one mapping, however, neuromorphic approaches such as use of distributed integrate and fire neurons can also be used to convert real value signals to spike times [95].

To simulate the faulty camera a group of pixel in the central region of the image were corrupted with random levels of noise (1:1 < SNR < 1:3) as shown in Figure 25 (a). Here the design choice regarding zero weights referred to in chapter 4 was implemented, where once a synaptic weight reached zero value, the synapse, and therefore the pixel, was disabled. This resulted in a system where the noise corrupted pixels were all disabled after at most 189 training images as shown in Figure 25 (d), leaving only the noise free channels for the kernels to train on and generating the receptive field of the neuron shown in Figure 11 (b). Note that the aggressiveness of pixel removal system can be controlled via the w_{rise} / w_{fall} ratio.



Figure 25. Unsupervised learning of common features concurrent with SNR encoding synaptic weights on noise corrupted spatio-temporal patterns encoding handwritten zero digits from the MNIST dataset using a single neuron. (a) The input space with corrupted pixels highlighted. (b) The final receptive field of the neuron after exposure to the MNIST zeros. Pixels with higher probability of being dark are more likely to generate late spikes which a correctly trained neuron should encode in the form of narrow kernels or high kernel slopes $\Delta r_i(t)$. Conversely, pixels more likely to be bright should be encoded by lower kernel slopes, as is seen. As a result, the further an input image is from this 'model' of a zero, the weaker the response of the neuron to the image. (c) The final synaptic weights of the neuron showing the disabled pixels. (d) The evolution of the synaptic weights over time. The neuron correctly weighted all the equally noiseless pixels equally high while weakening the weight of the noisy channels until they reached their minimum value at which point they are disabled. Figure from [46].

The disabled synapses shown in Figure 25 can potentially be reused, making SNR encoding synapses not only useful in terms of enhancing the performance of downstream signal processing systems as demonstrated in Figure 24, but in reconfigurable systems could also enable the potential reallocation of hardware resources to other tasks. This would allow more efficient hardware use in the context of the noise present in the sensors and in the environment.

Discussion

In the work presented in this thesis the neurons operate continuously without a mechanism to terminate learning. While this combined learning and recognition mode of operation can be useful in some real world environments it contrasts to most machine learning approaches where learning and recognition are separate tasks and where an external arbiter is typically tasked with halting the learning process after a satisfactory solution has been found. As demonstrated empirically in the Chapter 2, given an ergodic input the dynamics of the kernels and somatic threshold are sufficient to allow a network to learn most common spatio-temporal patterns in their environment while the adaptable threshold ensures that the output spike rate and spike width moves toward a stable solution over time. However for applications where a non-adaptive or static solution is required the addition of external supervisory controllers would be required to halt kernel and threshold adaptation. In future work the conditions under which learning and adaptation should be halted will be investigated.

While in this thesis the behavior SKAN model has been tested using a range of synthetically generated spatio-temporal spike pattern tests, it is difficult to compare these behaviors to results from standard machine learning approaches which are benchmarked using numerical datasets and not spatio-temporal spike patterns. In addition, as an unsupervised learning algorithm, SKAN is best suited to operate as a feature extraction subsystem in a larger multilayer recognition system together with downstream supervised learning algorithms. Future work will focus on the development and benchmarking of multilayer hardware implemented spiking neural networks in which SKAN operates as a feature extractor. This larger multilayer unsupervised-supervised network can more readily be compared in performance to traditional machine learning methods on new spike-based benchmark datasets developed by the neuromorphic community.

The SKAN model implemented here in digital hardware is potentially also suitable for implementation in analog VLSI neuromorphic systems. Although the SKAN model attempts to collapse the functionality of multiple synapses into single smart adaptive synaptic kernels using digital accumulators, these accumulators would still constitute the most hardware resource intensive elements of the system simply by virtue of their numbers in most network applications. As such the replacement of these digital elements with simple RC circuits can potentially drastically reduce hardware requirements.

Conclusion

In this thesis we have presented the SKAN, a neuromorphic implementation of a spiking neuron that performs unsupervised learning and inference on spatio-temporal spike pattern classification. The use of simple adaptable kernels was shown to represent an effective solution to hardware realized neural networks without the need for multipliers while SKAN operation was shown to be robust in the presence of noise allowing potential applications in noisy real-world environments.

Next an extended model of the synapto-dendritic kernel adapting neuron, with a simplified STDP synaptic weight update rule was presented and shown to perform concurrent unsupervised learning of commonly presented spatio-temporal patterns and synaptic encoding of afferent signal to noise ratio. In addition a novel shift based digital normalization algorithm was introduced which allowed synaptic

homeostasis or weight normalization without the need for a fixed-point division operation. While the neuron model is a simplified abstraction of highly complex synaptic, dendritic and somatic processes, its adaptive kernels permit the efficient functional modeling of neurons learning of spatio-temporal spike patterns in the presence of varying levels of noise. The signal to noise encoding synapses were shown to compensate for afferents corrupted with noise spikes resulting in improved learning and recognition performance across a range of noise environments with relevance to neuromorphic engineering applications such as bio-inspired visual processors. The proposed neuron models were shown to be hardware efficient and suitable for implementation on FPGA. The implementation of the neuron model in digital hardware showed that neuron's synapses have hardware usage requirements comparable to a single digital multiplier while being able to generate complex computationally useful behaviors.

References

- [1] A. V. M. Herz, T. Gollisch, C. K. Machens, and D. Jaeger, "Modeling single-neuron dynamics and computations: a balance of detail and abstraction.," *Science*, vol. 314, pp. 80–85, 2006.
- [2] R. Llinas, "Neuron," *Scholarpedia*, vol. 3, no. 8, p. 1490, Aug. 2008.
- [3] D. E. Rumelhart, J. L. McClelland, and R. J. Williams, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, no. 1. 1986.
- [4] B. Yegnanarayana, ARTIFICIAL NEURAL NETWORKS (Google eBook). PHI Learning Pvt. Ltd., 2009.
- [5] R. Van Rullen and S. J. Thorpe, "Rate Coding Versus Temporal Order Coding: What the Retinal Ganglion Cells Tell the Visual Cortex," *Neural Comput.*, vol. 13, no. 6, pp. 1255–1283, Jun. 2001.
- [6] S. B. Laughlin, "Energy as a constraint on the coding and processing of sensory information," *Current Opinion in Neurobiology*, vol. 11. pp. 475–480, 2001.
- [7] W. B. Levy and R. A. Baxter, "Energy Efficient Neural Codes." MIT Press 238 Main St., Suite 500, Cambridge, MA 02142-1046 USA journals-info@mit.edu, 04-Apr-2008.
- [8] W. Xu, W. Morishita, P. S. Buckmaster, Z. P. Pang, R. C. Malenka, and T. C. Südhof, "Distinct Neuronal Coding Schemes in Memory Revealed by Selective Erasure of Fast Synchronous Synaptic Transmission," *Neuron*, vol. 73, pp. 990–1001, 2012.
- [9] S. Afshar, L. George, J. Tapson, A. van Schaik, and T. J. Hamilton, "Racing to Learn : statistical inference and learning in a single spiking neuron with adaptive kernels.," *Front. Neurosci.*, vol. 8, p. 377, 2014.
- [10] E. M. Izhikevich, "Polychronization: computation with spikes.," *Neural Comput.*, vol. 18, no. 2, pp. 245–82, Mar. 2006.
- [11] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks," in *GMD-Forschungszentrum Informationstechnik Report 148*, 2001, vol. 148, p. 43.
- [12] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: a new framework for neural computation based on perturbations.," *Neural Comput.*, vol. 14, pp. 2531– 2560, 2002.
- [13] J. Zylberberg, J. T. Murphy, and M. R. DeWeese, "A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of V1 simple cell receptive fields," *PLoS Comput. Biol.*, vol. 7, no. 10, pp. 1–33, 2011.

- [14] S. M. Bohte, H. La Poutré, and J. N. Kok, "Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks," *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 426–435, 2002.
- [15] S. M. Bohte, J. N. Kok, and H. La Poutré, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, pp. 17–37, 2002.
- [16] F. Ponulak, "ReSuMe-new supervised learning method for Spiking Neural Networks," in *International Conference on Machine Learning, ICML*, 2005.
- [17] R. Gütig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing-based decisions.," *Nat. Neurosci.*, vol. 9, no. 3, pp. 420–8, 2006.
- [18] J. C. Tapson, G. K. Cohen, S. Afshar, K. M. Stiefel, Y. Buskila, R. M. Wang, T. J. Hamilton, and A. van Schaik, "Synthesis of neural networks for spatio-temporal spike pattern recognition and processing.," *Front. Neurosci.*, vol. 7, p. 153, Jan. 2013.
- [19] N. Kasabov, K. Dhoble, N. Nuntalid, and G. Indiveri, "Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition," *Neural Networks*, vol. 41, pp. 188– 201, 2013.
- [20] D. George and J. Hawkins, "Towards a mathematical theory of cortical micro-circuits.," *PLoS Comput. Biol.*, vol. 5, no. 10, p. e1000532, 2009.
- [21] P. Knag, J. K. Kim, T. Chen, and Z. Zhang, "A Sparse Coding Neural Network ASIC With On-Chip Learning for Feature Extraction and Encoding," *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 1070–1079, Apr. 2015.
- [22] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. Boahen, "Neuromorphic silicon neuron circuits.," *Front. Neurosci.*, vol. 5, p. 73, 2011.
- [23] S. Mitra, S. Fusi, and G. Indiveri, "Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI," *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, pp. 32–42, 2009.
- [24] M. Rahimi Azghadi, N. Iannella, S. F. Al-Sarawi, G. Indiveri, and D. Abbott, "Spike-Based Synaptic Plasticity in Silicon: Design, Implementation, Application, and Challenges," *Proc. IEEE*, vol. 102, pp. 717–737, 2014.
- [25] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic Electronic Circuits for Building Autonomous Cognitive Systems," *Proc. IEEE*, pp. 1–22, 2014.
- [26] M. Giulioni, F. Corradi, V. Dante, and P. del Giudice, "Real time unsupervised learning of visual stimuli in neuromorphic VLSI systems.," *Sci. Rep.*, vol. 5, p. 14730, Jan. 2015.
- [27] F. Walter, F. Röhrbein, and A. Knoll, "Neuromorphic implementations of neurobiological learning algorithms for spiking neural networks.," *Neural Netw.*, vol. 72, pp. 152–167, Aug. 2015.
- [28] P. A. Merolla, J. V Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science (80-.).*, vol. 345, no. 6197, pp. 668–673, 2014.
- [29] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.*, vol. 10, no. 4, pp. 1297–1301, 2010.

- [30] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Oct. 2014.
- [31] H. Zhuang, K.-S. Low, and W.-Y. Yau, "A Pulsed Neural Network With On-Chip Learning and Its Practical Applications," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 34–42, Feb. 2007.
- [32] H. Zhuang, K.-S. Low, and W.-Y. Yau, "Multichannel Pulse-Coupled-Neural-Network-Based Color Image Segmentation for Object Detection," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3299– 3308, Aug. 2012.
- [33] J. Seo, B. Brezzo, Y. Liu, B. D. Parker, S. K. Esser, R. K. Montoye, B. Rajendran, J. a. Tierno, L. Chang, D. S. Modha, and D. J. Friedman, "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," 2011 IEEE Cust. Integr. Circuits Conf., pp. 1–4, 2011.
- [34] J. Seo and M. Seok, "Digital CMOS neuromorphic processor design featuring unsupervised online learning," in 2015 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), 2015, pp. 49–51.
- [35] A. W. Topol, D. C. La Tulipe, L. Shi, D. J. Frank, K. Bernstein, S. E. Steen, A. Kumar, G. U. Singco, A. M. Young, K. W. Guarini, and M. leong, "Three-dimensional integrated circuits," *IBM Journal of Research and Development*, vol. 50. pp. 491–506, 2006.
- [36] E. Bullmore and O. Sporns, "The economy of brain network organization," *Nature Reviews Neuroscience*. 2012.
- [37] T. Pfeil, T. C. Potjans, S. Schrader, W. Potjans, J. Schemmel, M. Diesmann, and K. Meier, "Is a 4-Bit Synaptic Weight Resolution Enough? – Constraints on Enabling Spike-Timing Dependent Plasticity in Neuromorphic Hardware," *Frontiers in Neuroscience*, vol. 6. 2012.
- [38] J. Zhu and P. Sutton, "FPGA Implementations of Neural Networks A Survey of a Decade of Progress," *Lect. Notes Comput. Sci.*, vol. 2778, pp. 1062–1066, 2003.
- [39] J. Schemmel, D. Briiderle, A. Griibl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 2010, pp. 1947–1950.
- [40] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker Project," *Proc. IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.
- [41] R. Wang, T. J. Hamilton, J. Tapson, and A. van Schaik, "A compact reconfigurable mixed-signal implementation of synaptic plasticity in spiking neurons," in 2014 IEEE International Symposium on Circuits and Systems (ISCAS), 2014, pp. 862–865.
- [42] T. Pfeil, T. C. Potjans, S. Schrader, W. Potjans, J. Schemmel, M. Diesmann, and K. Meier, "Is a 4-bit synaptic weight resolution enough? - constraints on enabling spike-timing dependent plasticity in neuromorphic hardware.," *Front. Neurosci.*, vol. 6, p. 90, Jan. 2012.
- [43] P. P. San, S. Hussain, and A. Basu, "Spike-timing dependent morphological learning for a neuron with nonlinear active dendrites," in 2014 International Joint Conference on Neural Networks (IJCNN), 2014, pp. 3192–3196.
- [44] B. Zhao, R. Ding, S. Chen, B. Linares-Barranco, and H. Tang, "Feedforward Categorization on AER Motion Events Using Cortex-Like Features in a Spiking Neural Network.," *IEEE Trans. neural networks Learn. Syst.*, vol. PP, no. 99, p. 1, Oct. 2014.
- [45] S. Mitra, S. Fusi, and G. Indiveri, "Real-Time Classification of Complex Patterns Using Spike-Based

Learning in Neuromorphic VLSI.," IEEE Trans. Biomed. Circuits Syst., vol. 3, no. 1, pp. 32–42, Feb. 2009.

- [46] S. Afshar, L. George, C. S. Thakur, J. Tapson, A. van Schaik, P. de Chazal, and T. J. Hamilton, "Turn Down That Noise: Synaptic Encoding of Afferent SNR in a Single Spiking Neuron.," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 2, pp. 188–96, Apr. 2015.
- [47] S. Hussain, A. Basu, R. M. Wang, and T. Julia Hamilton, "Delay learning architectures for memory and classification," *Neurocomputing*, vol. 138, pp. 14–26, 2014.
- [48] Y. Buskila, J. W. Morley, J. Tapson, and A. van Schaik, "The adaptation of spike backpropagation delays in cortical neurons.," *Front. Cell. Neurosci.*, vol. 7, p. 192, Jan. 2013.
- [49] R. M. Wang, T. J. Hamilton, J. C. Tapson, and A. van Schaik, "A mixed-signal implementation of a polychronous spiking neural network with delay adaptation.," *Front. Neurosci.*, vol. 8, p. 51, Jan. 2014.
- [50] T. Dowrick, S. Hall, and L. McDaid, "A simple programmable axonal delay scheme for spiking neural networks," *Neurocomputing*, vol. 108, pp. 79–83, May 2013.
- [51] T. J. Hamilton, S. Afshar, A. van Schaik, and J. Tapson, "Stochastic Electronics: A Neuro-Inspired Design Paradigm for Integrated Circuits," *Proc. IEEE*, vol. 102, no. 5, pp. 843–859, May 2014.
- [52] R. A. Silver, "Neuronal arithmetic.," Nat. Rev. Neurosci., vol. 11, pp. 474–489, 2010.
- [53] B. W. Mel, "Synaptic integration in an excitable dendritic tree.," J. Neurophysiol., vol. 70, no. 3, pp. 1086–101, 1993.
- [54] N. L. Rochefort and A. Konnerth, "Dendritic spines: from structure to in vivo function," *EMBO reports*, vol. 13. pp. 699–708, 2012.
- [55] M. T. Harnett, J. K. Makara, N. Spruston, W. L. Kath, and J. C. Magee, "Synaptic amplification by dendritic spines enhances input cooperativity," *Nature*. 2012.
- [56] A. Losonczy, J. K. Makara, and J. C. Magee, "Compartmentalized dendritic plasticity and input feature storage in neurons.," *Nature*, vol. 452, pp. 436–441, 2008.
- [57] H. Kasai, M. Fukuda, S. Watanabe, A. Hayashi-Takagi, and J. Noguchi, "Structural dynamics of dendritic spines in memory and cognition," *Trends in Neurosciences*, vol. 33. pp. 121–129, 2010.
- [58] A. Papoutsi, G. Kastellakis, M. Psarrou, S. Anastasakis, and P. Poirazi, "Coding and decoding with dendrites," *Journal of Physiology Paris*, vol. 108. pp. 18–27, 2014.
- [59] N. Spruston, "Pyramidal neurons: dendritic structure and synaptic integration.," *Nat. Rev. Neurosci.*, vol. 9, pp. 206–221, 2008.
- [60] M. M. Shah, R. S. Hammond, and D. A. Hoffman, "Dendritic ion channel trafficking and plasticity," *Trends in Neurosciences*, vol. 33. pp. 307–316, 2010.
- [61] D. H. Bhatt, S. Zhang, and W.-B. Gan, "Dendritic spine dynamics.," Annu. Rev. Physiol., vol. 71, pp. 261–282, 2009.
- [62] Y. Yu and T. Lee, "Dynamical mechanisms underlying contrast gain control in single neurons," *Phys. Rev. E*, vol. 68, no. 1, p. 011901, Jul. 2003.
- [63] N. Brunel, V. Hakim, and M. J. E. Richardson, "Single neuron dynamics and computation," *Current Opinion in Neurobiology*, vol. 25. pp. 149–155, 2014.
- [64] H. Kasai, T. Hayama, M. Ishikawa, S. Watanabe, S. Yagishita, and J. Noguchi, "Learning rules and persistence of dendritic spines," *Eur. J. Neurosci.*, vol. 32, pp. 241–249, 2010.

- [65] S. George, J. Hasler, S. Koziol, S. Nease, and S. Ramakrishnan, "Low Power Dendritic Computation forWordspotting," *J. Low Power Electron. Appl.*, vol. 3, pp. 73–98, 2013.
- [66] S. Ramakrishnan, R. Wunderlich, J. Hasler, and S. George, "Neuron array with plastic synapses and programmable dendrites," *IEEE Trans. Biomed. Circuits Syst.*, vol. 7, pp. 631–642, 2013.
- [67] Y. Wang and S. C. Liu, "Active processing of spatio-temporal input patterns in silicon dendrites," *IEEE Trans. Biomed. Circuits Syst.*, vol. 7, pp. 307–318, 2013.
- [68] C. C. Hsu, A. C. Parker, and J. Joshi, "Dendritic computations, dendritic spiking and dendritic plasticity in nanoelectronic neurons," in *Midwest Symposium on Circuits and Systems*, 2010, pp. 89–92.
- [69] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, "Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs," *Science (80-.).*, vol. 275, no. 5297, pp. 213–215, 1997.
- [70] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner, "A neuronal learning rule for submillisecond temporal coding.," *Nature*, vol. 383, no. 6595, pp. 76–81, 1996.
- [71] P. J. Sjöström, G. G. Turrigiano, and S. B. Nelson, "Rate, timing, and cooperativity jointly determine cortical synaptic plasticity," *Neuron*, vol. 32, no. 6, pp. 1149–1164, 2001.
- [72] R. Guyonneau, R. VanRullen, and S. J. Thorpe, "Neurons tune to the earliest spikes through STDP.," *Neural Comput.*, vol. 17, no. 4, pp. 859–879, 2005.
- [73] P. J. Sjöström, E. A. Rancz, A. Roth, and M. Häusser, "Dendritic excitability and synaptic plasticity.," *Physiol. Rev.*, vol. 88, pp. 769–840, 2008.
- [74] S. Deneve, "Bayesian spiking neurons I: inference.," *Neural Comput.*, vol. 20, pp. 91–117, 2008.
- [75] B. Fontaine, J. L. Peña, and R. Brette, "Spike-Threshold Adaptation Predicted by Membrane Potential Dynamics In Vivo," *PLoS Comput. Biol.*, vol. 10, p. e1003560, 2014.
- [76] K. F. H. Lee, C. Soares, and J.-C. Béïque, "Tuning into diversity of homeostatic synaptic plasticity.," *Neuropharmacology*, vol. 78, pp. 31–7, Mar. 2014.
- [77] R. J. Sofatzis, S. Afshar, and T. J. Hamilton, "The Synaptic Kernel Adaptation Network," in *The Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014.
- [78] S. Afshar, G. K. Cohen, R. M. Wang, A. Van Schaik, J. Tapson, T. Lehmann, and T. J. Hamilton, "The ripple pond: enabling spiking networks to see.," *Front. Neurosci.*, vol. 7, pp. 1–12, 2013.
- [79] Sofatzis, R.J., S. Afshar, and T. J. Hamilton, "Rotationally Invariant Vision Recognition with Neuromorphic Transformation and Learning Networks," in *The Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014.
- [80] A. Pouget, J. M. Beck, W. J. Ma, and P. E. Latham, "Probabilistic brains: knowns and unknowns.," *Nat. Neurosci.*, vol. 16, pp. 1170–8, 2013.
- [81] A. M. Bastos, W. M. Usrey, R. A. Adams, G. R. Mangun, P. Fries, and K. J. Friston, "Canonical Microcircuits for Predictive Coding," *Neuron*, vol. 76. pp. 695–711, 2012.
- [82] M. Boerlin, C. K. Machens, and S. Denève, "Predictive Coding of Dynamical Variables in Balanced Spiking Networks," *PLoS Comput. Biol.*, vol. 9, 2013.
- [83] D. Kappel, B. Nessler, and W. Maass, "STDP Installs in Winner-Take-All Circuits an Online Approximation to Hidden Markov Model Learning," *PLoS Comput. Biol.*, vol. 10, 2014.
- [84] D. S. Corneil, E. Neftci, G. Indiveri, and M. Pfeiffer, "Learning, Inference, and Replay of Hidden

State Sequences in Recurrent Spiking Neural Networks," 2014.

- [85] P. J. Tully, M. H. Hennig, and A. Lansner, "Synaptic and nonsynaptic plasticity approximating probabilistic inference.," *Front. Synaptic Neurosci.*, vol. 6, p. 8, 2014.
- [86] L. Kuhlmann, M. Hauser-Raspe, J. H. Manton, D. B. Grayden, J. Tapson, and A. van Schaik, "Approximate, computationally efficient online learning in Bayesian spiking neurons.," *Neural Comput.*, vol. 26, no. 3, pp. 472–96, Mar. 2014.
- [87] M. G. Paulin and A. van Schaik, "Bayesian Inference with Spiking Neurons," Jun. 2014.
- [88] E. Hunsberger, M. Scott, and C. Eliasmith, "The Competing Benefits of Noise and Heterogeneity in Neural Coding.," *Neural Comput.*, pp. 1–24, May 2014.
- [89] W. Maass, "Noise as a Resource for Computation and Learning in Networks of Spiking Neurons," 2014.
- [90] M. D. McDonnell and L. M. Ward, "The benefits of noise in neural systems: bridging theory and experiment.," *Nat. Rev. Neurosci.*, vol. 12, pp. 415–426, 2011.
- [91] B. Marr and J. Hasler, "Compiling probabilistic, bio-inspired circuits on a field programmable analog array.," *Front. Neurosci.*, vol. 8, p. 86, Jan. 2014.
- [92] T. J. Hamilton and J. Tapson, "A neuromorphic cross-correlation chip," in *Proceedings IEEE* International Symposium on Circuits and Systems, 2011, pp. 865–868.
- [93] B. Nessler, M. Pfeiffer, L. Buesing, and W. Maass, "Bayesian Computation Emerges in Generic Cortical Microcircuits through Spike-Timing-Dependent Plasticity," *PLoS Comput. Biol.*, vol. 9, 2013.
- [94] a Gupta and L. N. Long, "Hebbian learning with winner take all for spiking neural networks," 2009 Int. Jt. Conf. Neural Networks, vol. 81, pp. 1054–1060, 2009.
- [95] S. Afshar, O. Kavehei, A. van Schaik, J. Tapson, S. Skafidas, and T. J. Hamilton, "Emergence of competitive control in a memristor-based neuromorphic circuit," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 1–8.
- [96] J. Tapson and A. van Schaik, "An asynchronous parallel neuromorphic ADC architecture," in 2012 *IEEE International Symposium on Circuits and Systems*, 2012, pp. 2409–2412.
- [97] S. Habenschuss, H. Puhr, and W. Maass, "Emergence of optimal decoding of population codes through STDP.," *Neural Comput.*, vol. 25, no. 6, pp. 1371–407, Jun. 2013.
- [98] T. Masquelier, R. Guyonneau, and S. J. Thorpe, "Competitive STDP-based spike pattern learning.," *Neural Comput.*, vol. 21, no. 5, pp. 1259–76, May 2009.
- [99] D. Jimenez Rezende and W. Gerstner, "Stochastic variational learning in recurrent spiking networks.," *Front. Comput. Neurosci.*, vol. 8, p. 38, 2014.
- [100] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.
- [101] J. Sjöström and W. Gerstner, "Spike-timing dependent plasticity," *Scholarpedia*, vol. 5, no. 2, p. 1362, Feb. 2010.
- [102] H. Markram, W. Gerstner, and P. J. Sjöström, "Spike-timing-dependent plasticity: a comprehensive overview.," *Front. Synaptic Neurosci.*, vol. 4, p. 2, Jan. 2012.
- [103] G. G. Turrigiano, "The self-tuning neuron: synaptic scaling of excitatory synapses.," *Cell*, vol. 135, no. 3, pp. 422–35, Oct. 2008.

- [104] E. Marder and J.-M. Goaillard, "Variability, compensation and homeostasis in neuron and network function.," *Nat. Rev. Neurosci.*, vol. 7, no. 7, pp. 563–74, Jul. 2006.
- [105] A. Diamond, T. Nowotny, and M. Schmuker, "Comparing Neuromorphic Solutions in Action: Implementing a Bio-Inspired Solution to a Benchmark Classification Task on Three Parallel-Computing Platforms," *Front. Neurosci.*, vol. 9, Jan. 2016.
- [106] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits." [Online]. Available: http://yann.lecun.com/exdb/mnist.