



Racing to learn: statistical inference and learning in a single spiking neuron with adaptive kernels

Saeed Afshar^{1*}, Libin George², Jonathan Tapson¹, André van Schaik¹ and Tara J. Hamilton^{1,2}

¹ Bioelectronics and Neurosciences, The MARCS Institute, University of Western Sydney, Penrith, NSW, Australia

² School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, NSW, Australia

Edited by:

Chiara Bartolozzi, Italian Institute of Technology, Italy

Reviewed by:

Sadique Sheik, University of Zurich, ETH Zurich, Switzerland
Mattia Rigotti, IBM Research, USA
Stephen Nease, Cognitive Interaction Technology - Center of Excellence Bielefeld, Germany

*Correspondence:

Saeed Afshar, Bioelectronics and Neuroscience, The MARCS Institute, University of Western Sydney, Locked Bag 1797, Penrith, NSW 2751, Australia
e-mail: s.afshar@uws.edu.au

This paper describes the Synapto-dendritic Kernel Adapting Neuron (SKAN), a simple spiking neuron model that performs statistical inference and unsupervised learning of spatiotemporal spike patterns. SKAN is the first proposed neuron model to investigate the effects of dynamic synapto-dendritic kernels and demonstrate their computational power even at the single neuron scale. The rule-set defining the neuron is simple: there are no complex mathematical operations such as normalization, exponentiation or even multiplication. The functionalities of SKAN emerge from the real-time interaction of simple additive and binary processes. Like a biological neuron, SKAN is robust to signal and parameter noise, and can utilize both in its operations. At the network scale neurons are locked in a race with each other with the fastest neuron to spike effectively “hiding” its learnt pattern from its neighbors. The robustness to noise, high speed, and simple building blocks not only make SKAN an interesting neuron model in computational neuroscience, but also make it ideal for implementation in digital and analog neuromorphic systems which is demonstrated through an implementation in a Field Programmable Gate Array (FPGA). Matlab, Python, and Verilog implementations of SKAN are available at: http://www.uws.edu.au/bioelectronics_neuroscience/bens/reproducible_research.

Keywords: spiking neural network, neuromorphic engineering, spike time dependent plasticity, stochastic computation, dendritic computation, unsupervised learning

INTRODUCTION

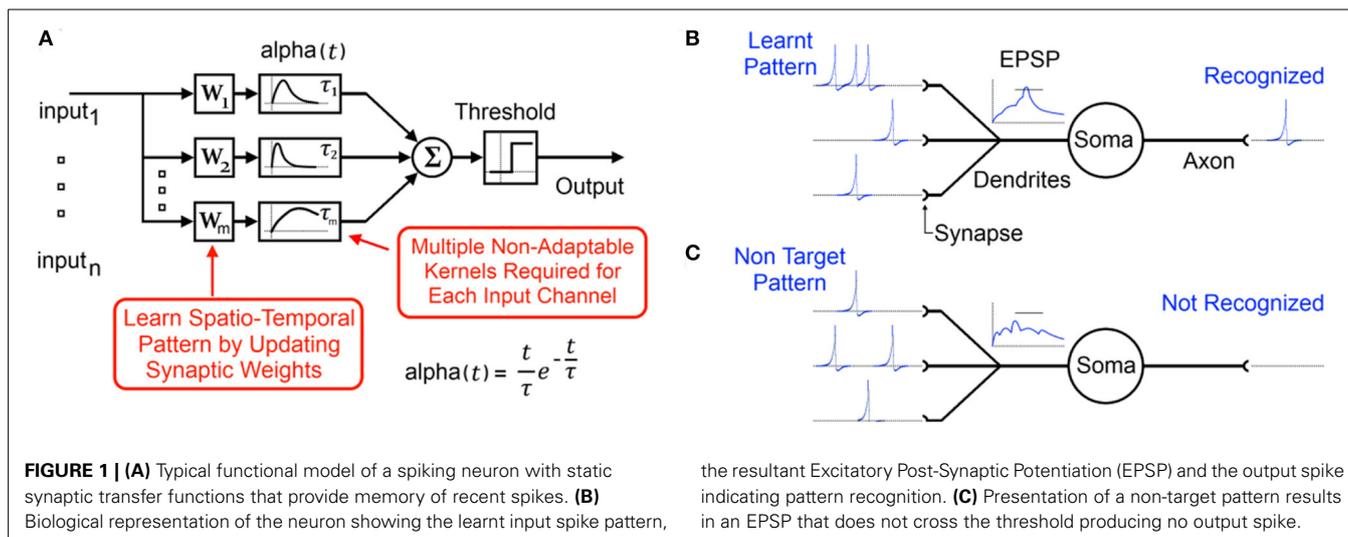
PRIOR WORK

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 (Herz et al., 2006; Llinas, 2008). 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 (Yegnanarayana, 1999). 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 (Levy and Baxter, 1996; Laughlin, 2001; Van Rullen and Thorpe, 2001). In contrast, 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 (Xu et al., 2012).

More recently, the greater efficiency, higher performance, and biologically realistic dynamics of temporal coding neural networks has motivated the development of synaptic weight adaptation schemes that operate on temporally coding Spiking Neural Networks (SNN) (Jaeger, 2001; Maass et al., 2002; Izhikevich, 2006; Kasabov et al., 2013; Tapson et al., 2013; Gütiç, 2014).

After proposition many of these models are followed soon by their implementation in neuromorphic hardware (Mitra et al., 2009; Indiveri et al., 2011; Beyeler et al., 2013; O'Connor et al., 2013; Chicca et al., 2014; Rahimi Azghadi et al., 2014). One of the problems faced by neuromorphic hardware engineers is the hardware inefficiency of many neural network algorithms. These algorithms are almost always initially designed for performance in a constraint free mathematical context with numerous all-to-all connected neurons and/or to satisfy some biological realism criteria, which create difficulties in hardware implementation.

Additionally in order for such spiking systems to combine temporal coding and weight adaptation, multiple synapses consisting of synaptic transfer functions (or synaptic kernels) as well as synaptic weights must be realized for every input channel as shown in **Figure 1**. With the aim of being biologically plausible, exponentially decaying functions are typically chosen as the synaptic kernel, which is then multiplied by the synaptic weight. Such functions and weights are quite complex and difficult to implement in simple scalable analog and digital hardware with even the simplest schemes requiring at least one multiplication operation at every synapse. The difficulty of realizing multipliers at the synapse and the large number of synapses used in most algorithms has motivated moves toward more scalable digital synapses (Merolla et al., 2011; Seo et al., 2011; Arthur et al., 2012; Pfeil et al., 2012), novel memristor based solutions (Indiveri et al., 2013; Serrano-Gotarredona et al., 2013) and second order solutions such as sparse coding (Kim et al., 2014), time multiplexing,



and Address Event Representation (AER) (Zamarreno-Ramos et al., 2013) where only one or a few instances of the complex computational units are realized and these are utilized serially. Despite the success of these approaches such serial implementations can sometimes introduce associated bottlenecks, which can detract from the main strength of the neural network approach: its *distributed* nature (Misra and Saha, 2010).

Rather than implement complex synaptic weight adaptation, other neuromorphic SNN implementations have, in the last 3 years, focused exclusively on adjustment of explicit propagation delays along the neural signal path and coincidence detection of input spikes to encode memory (Scholze, 2011; Sheik et al., 2012, 2013; Dowrick et al., 2013; Hussain et al., 2014; Wang et al., 2014). This discarding of synaptic weights and kernels significantly simplifies implementation and improves scalability. The disadvantage is that explicit delay learning schemes can produce “sharp” systems with poor tolerance for the dynamically changing temporal variance they inevitably encounter in applications where neuronal systems are expected to excel: noisy, dynamic, and unpredictable environments.

One of the features shared by all the preceding systems is that the kernels used for encoding temporal information are static as shown in **Figure 2**. However recent advances in neurophysiology have revealed that synapto-dendritic structures and their associated transfer functions are highly complex and adapt during learning in response to the statistical contexts of their stimulus environment (Losonczy et al., 2008; Yoshihara et al., 2009; Kasai et al., 2010a; Lee et al., 2012; Rochefort and Konnerth, 2012; Smith et al., 2013; Colgan and Yasuda, 2014). 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 function as 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 (Spruston, 2008; Silver, 2010; Harnett et al., 2012; Papoutsis et al., 2014). These effects represent novel dynamics with as yet unexplored emergent computational properties, which may potentially solve currently

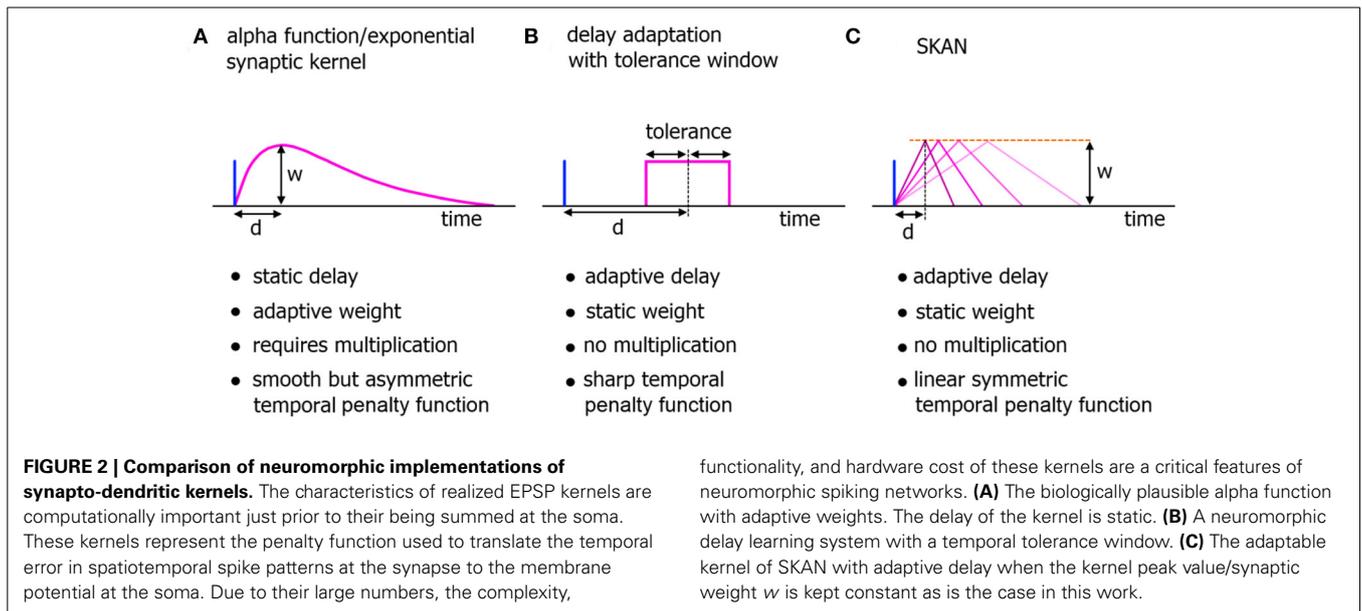
intractable problems in computational neuroscience (Bhatt et al., 2009; Shah et al., 2010). These dendritic adaptation effects have recently been modeled through large rule sets (Yu and Lee, 2003; Kasai et al., 2010b; Brunel et al., 2014) and in the neuromorphic field the use of dendrites for computation is beginning to be explored (Hsu et al., 2010; George et al., 2013; Ramakrishnan et al., 2013; Wang and Liu, 2013). However with biological realism as a major focus, many of the models carry significant extra complexity which can impede scalability.

NEURONS AS FUNCTIONAL MODELS OF DISTRIBUTED PROCESSING

In this paper the goal of 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 3 illustrates the basic elements of SNN algorithms as well as some useful information flow and storage restrictions (red), which, if adhered to at the neuron design stage, prove helpful during the physical 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, or systems that assume an external supervisor in real world contexts where such a signal is unlikely to be available.
2. **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 geometrically increasing number of connections, which is prohibitive both in hardware and in the brain (Topol et al., 2006; Bullmore and Sporns, 2012).



- 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 the domain of distributed processors such as GPUs and fall outside the neuromorphic scope.
- Multiplication: Multipliers are typically inefficient to implement in 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 number available on a hardware platform result in neural networks implemented with time-multiplexing. This, in turn, limits the size and the applications where this hardware is viable (Zhu and Sutton, 2003; Pfeil et al., 2012).

MATERIALS AND METHODS

The elements of Synapto-dendritic Kernel Adapting Neuron (SKAN) and its learning rule are defined in the first part of this section. In the second part, 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 4**.

Synapse/dendrite

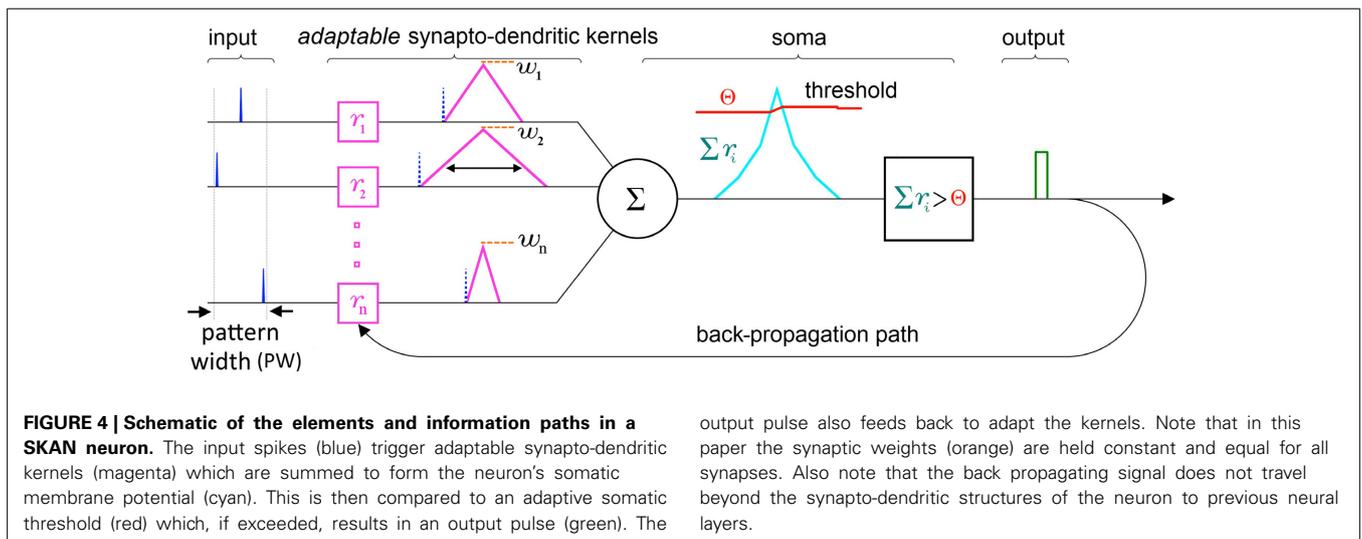
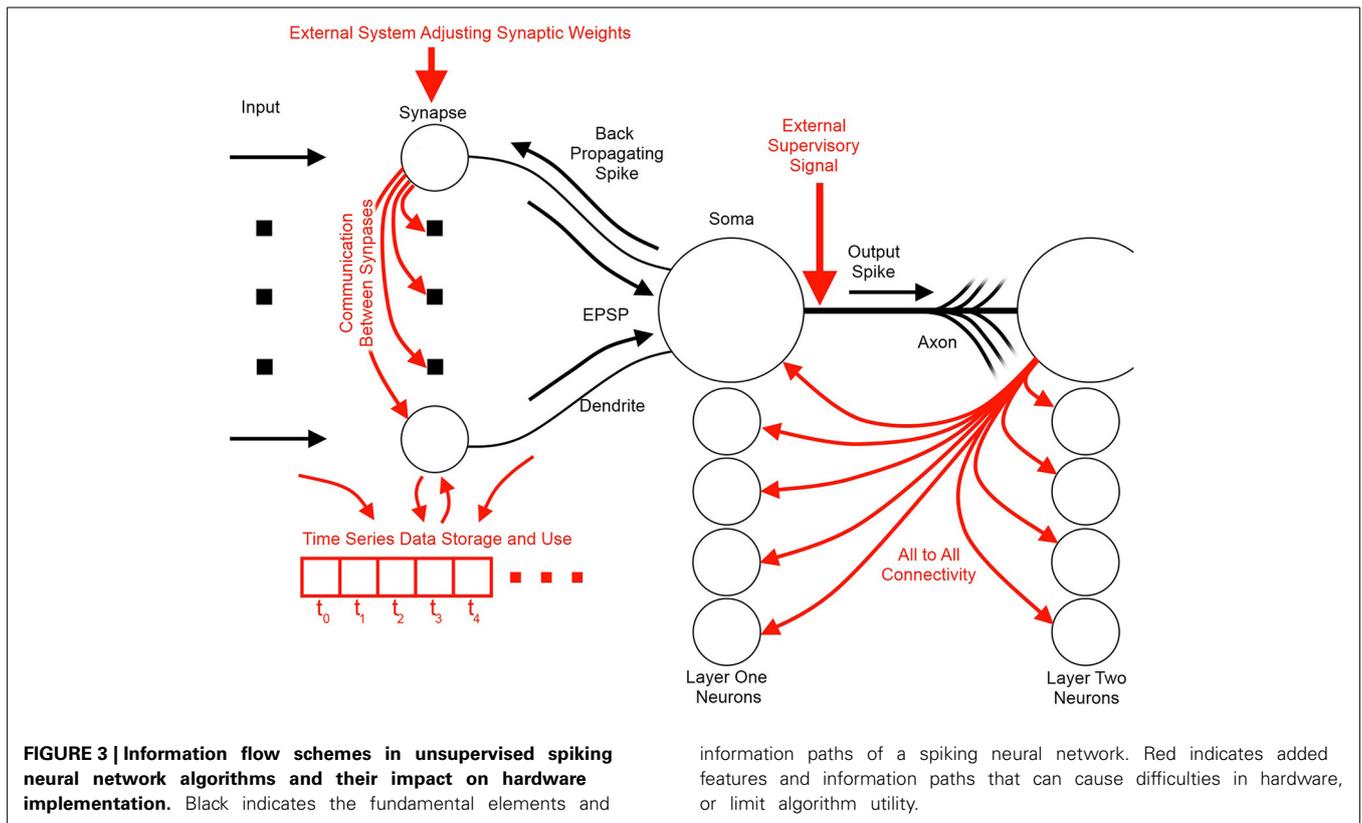
An incoming input spike initiates a simplified synapto-dendritic kernel at each input channel i . This kernel is controlled by a physiological process, p_i , and for simplicity is modeled as a ramp up and a ramp down sequence generated via an accumulator r_i with step size Δr_i . An input spike triggers 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 paper to simplify the algorithm. 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 toward 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 re-initiates the sequence. This simple conceptual sequence, which is analogous to a dendritically filtered neuronal EPSP, is illustrated in **Figure 5**.

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

$$p_i(t) = \begin{cases} 1 & \text{if } (u_i(t) = 1 \wedge p_i(t-1) = 0) \\ & \vee (p_i(t-1) = 1 \wedge r_i(t-1) < w_i) \\ -1 & \text{if } (p_i(t-1) = 1 \wedge r_i(t-1) \geq w_i) \\ & \vee (p_i(t-1) = -1 \wedge r_i(t-1) > 0) \\ 0 & \text{else} \end{cases} \quad (1)$$

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 (Sjöström et al., 2008). In this paper, 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.



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:

$$s(t) = \begin{cases} 1 & \text{if } \sum_i r_i(t) > \Theta(t - 1) \\ 0 & \text{else} \end{cases} \quad (2)$$

SKAN differs from most previous spiking neuron models in not resetting the membrane potential after spiking (see Denève, 2008; Tapson et al., 2013 for exceptions). This permits wide pulse widths at the neuron output $s(t)$. While such wide pulses do

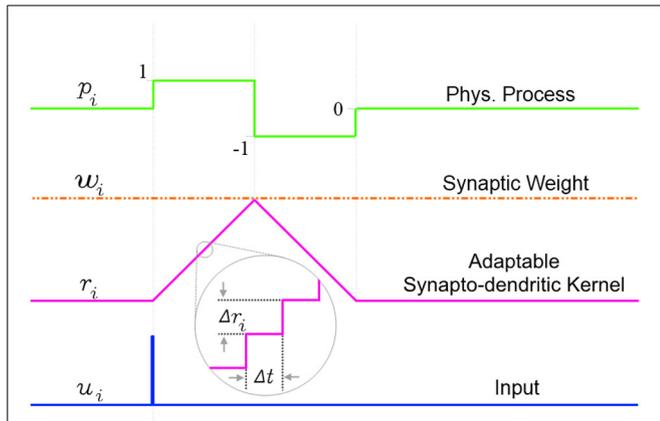


FIGURE 5 | The simplified adaptable synapto-dendritic kernel of SKAN. An input spike (blue) triggers the kernel’s ramp up ramp down sequence. The input spike sets a flag p_i representing a physical process to one (green). The flag causes an accumulator (magenta) to count up from zero by Δr_i at each 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.

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.

FEEDBACK MECHANISMS/LEARNING RULES

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:

$$\begin{bmatrix} r_i(t) \\ \Delta r_i(t) \end{bmatrix} = \begin{bmatrix} r_i(t-1) \\ \Delta r_i(t-1) \end{bmatrix} + p_i(t-1) \begin{bmatrix} \Delta r_i(t-1) \\ ddr \times s(t-1) \end{bmatrix} \quad (3)$$

The use of indirect evidence about the dynamic state of other dendrites in the form of the back propagating spike is a central feature

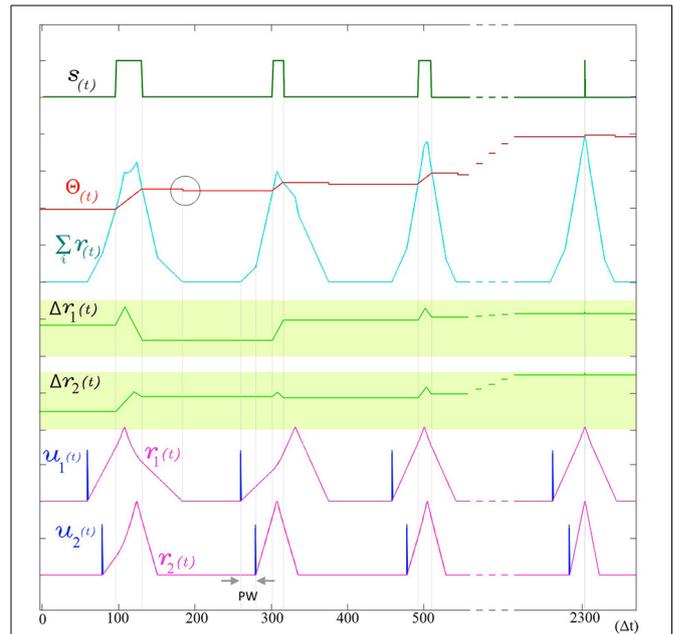


FIGURE 6 | The adaptation of SKAN. The kernels and the threshold of SKAN adapt in response to repeated spatio-temporal pattern presentations. For visual clarity the pattern only consists of the Inter-Spike Interval (ISI) across two input channels $u_i(t)$ such that the pattern width (PW) is equivalent 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 11th 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 gray circle.

in the operation of SKAN and enables the synchronization of all the neuron’s dendritic kernel peaks as shown in **Figure 6**.

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 6** and described in the first line of Equation 4.

$$\Theta(t) = \begin{cases} \Theta(t-1) + \Theta_{rise} & \text{if } \sum_i r_i(t) > \Theta(t-1) \\ \Theta(t-1) - \Theta_{fall} & \text{if } \sum_i r_i(t) = 0 \wedge \sum_i r_i(t-1) > 0 \\ \Theta(t-1) & \text{else} \end{cases} \quad (4)$$

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 6**. The counterbalancing 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 (Fontaine et al., 2014; Lee et al., 2014), 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.

Observing the first spike in a spike train or burst

As 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 for each input channel the neuron trains on the first spike of 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 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 spatio-pattern presented in a random sequence containing multiple patterns (Sofatzis et al., 2014a). This effect has been demonstrated in the context of visual processing where hand gestures were transformed to spatio-temporal patterns via a neuronal transform operation (Afshar et al., 2013) and processed by SKAN (Sofatzis et al., 2014b). **Figure 7** 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 be selected.

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. This effect, shown in **Figure 8**, is absent in the standard STDP schemes where improved response times comes at the cost of information loss. In STDP schemes the earliest spikes in a spatio-temporal pattern tend to be highly weighted while the later spike lose weight and have little effect on recognition (Masquelier et al., 2009). This behavior can be seen as advantageous if an assumption is made that the later

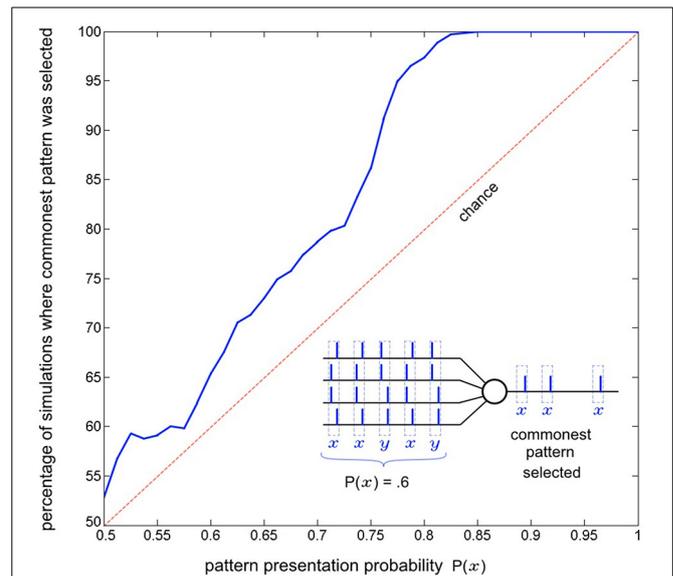


FIGURE 7 | Commonest pattern selection as a function of pattern presentation probability. The inset illustrates one simulation a 5 pattern long sequence where each pattern is sampled from two randomly initialized spatio-temporal patterns x and y , with probability $P(x) = 0.6$. In this particular simulation pattern x was selected by the neuron. The plot shows data resulting from the same experiment but with 1000 simulations of 300 pattern long sequences for each probability $P(x) = 0.5$ to 1. The graph shows that the likelihood of a pattern being selected rises with increasing presentation probability. For each simulation the output of the neuron for the second half of the sequence (150–300th pattern) was recorded and it was determined whether pattern x or pattern y had been selected. Also tested was whether both, or neither pattern was selected by the neuron at some point during the sequence (i.e., the neuron spiked at least once for both of the patterns or failed to spike for a pattern it had selected during the sequence). In the more than seven million pattern presentations ($1000 \times 150 \times 51$) neither of these occurred.

spikes in carry less information however in this is an assumption that cannot be made in general. In contrast SKAN's adaptable kernels reduce output spike latency with adaptation while still enabling every spike to affect the output. 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.

As shown in **Figure 8**, 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 8** 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.

$$\Delta r_{max} < w/PW \quad (5)$$

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

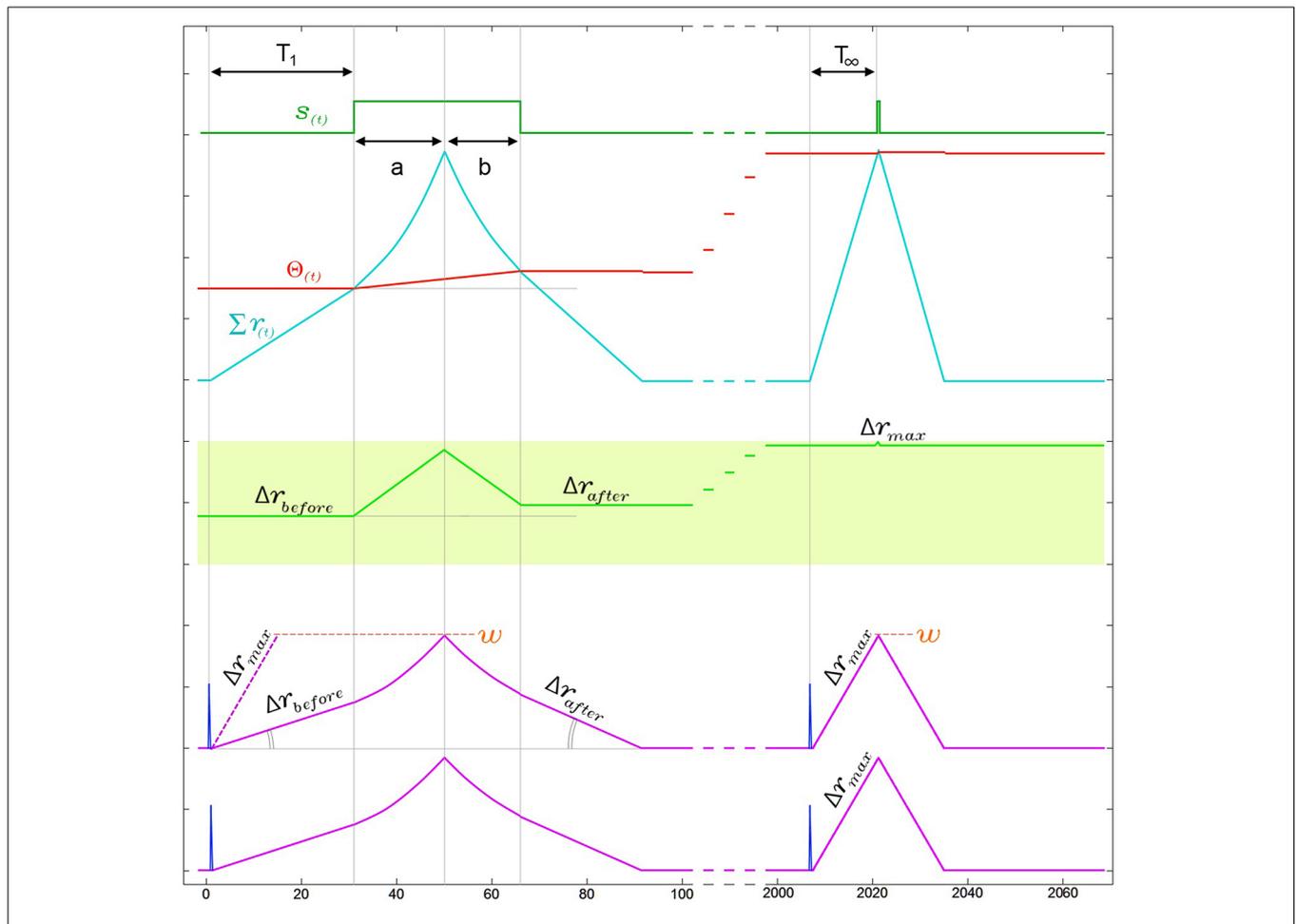


FIGURE 8 | Narrowing of kernels leads to improved response time during neuronal adaptation in a two input neuron. For visual clarity the neuron is presented with an $ISI = 0\Delta t$ pattern and the two kernels start with identical initial slopes $[\Delta r_1(0) = \Delta r_2(0) = \Delta r_{before}]$. 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 T_1 in the first presentation, improves until it reaches its minimal possible value $T_\infty \approx w/\Delta r_{max}$.

Evolution of the temporal receptive field in SKAN approximates statistical inference

Recent work has demonstrated the connection between synaptic weight adaptation and approximate probabilistic inference in the context of rate coding and spiking networks (Bastos et al., 2012; Boerlin et al., 2013; Pouget et al., 2013; Corneil et al., 2014; Kappel et al., 2014; Kuhlmann et al., 2014; Paulin and van Schaik, 2014; Tully et al., 2014), 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 ISI generating processes. **Figure 9** 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 increasing temporal jitter. 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 9**.

$$RF_{i=2}(\tau) = \sum_t \left(\sum_i r_i(\tau, t) - \Theta(\tau, t) \right) \times s(\tau, t) \quad (6)$$

where τ is the ISI being simulated.

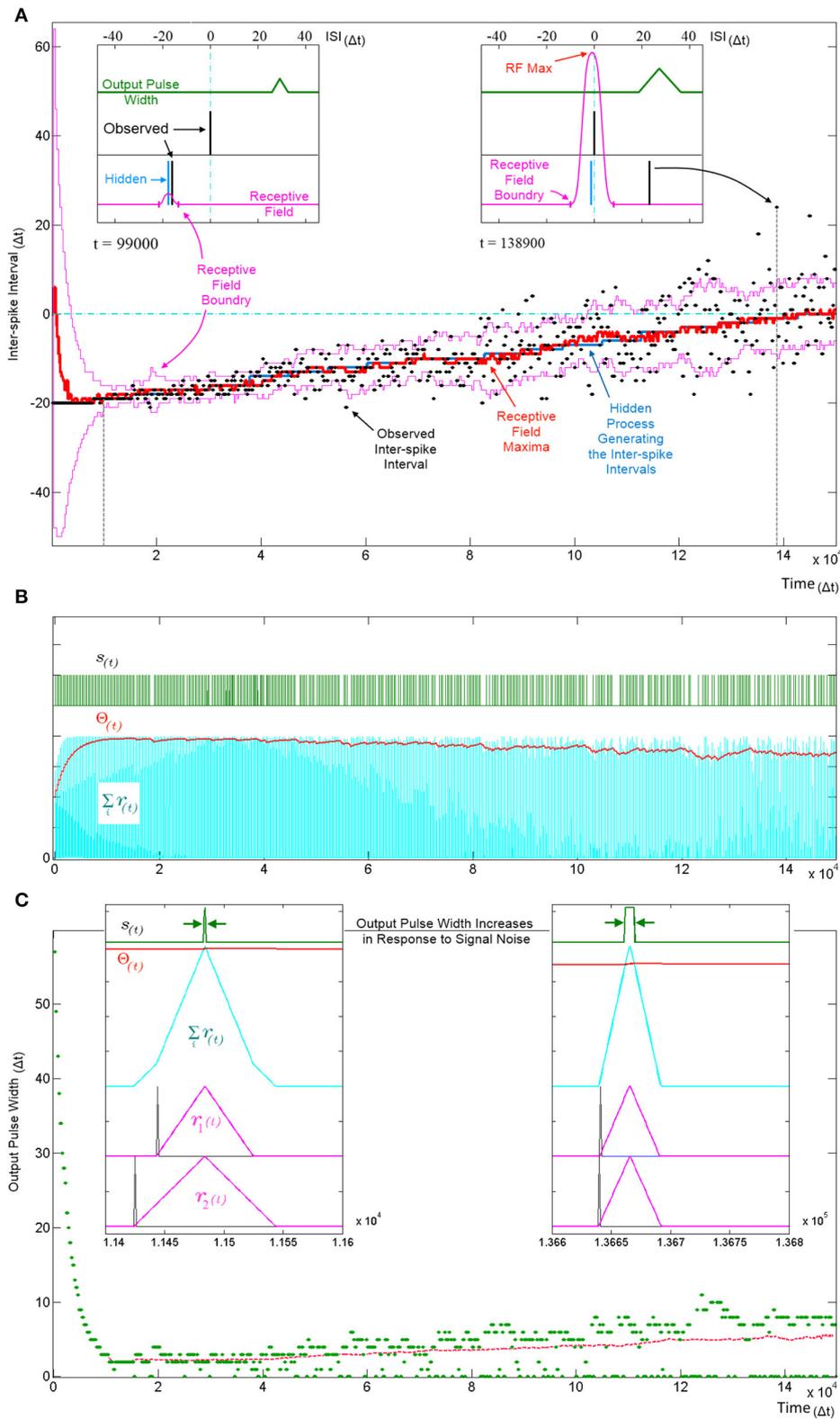


FIGURE 9 | Tracking a hidden ISI producing process and its variance. All three panels (A–C) show different aspects of the same simulation where a single SKAN learns statistics of a dynamic ISI across two input channels. (A) A hidden process (blue) moves from ISI = $-20 \Delta t$ to ISI = $0 \Delta t$. 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
(Continued)

FIGURE 9 | Continued

initialized kernels peak at $ISI = 6 \Delta t$. 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 $t = 138.9 \text{ k}\Delta t$). **(B)** Variables and behavior of the neuron throughout the simulation: after a rapid initial increase the threshold $\Theta(t)$ settles near the peak of the membrane potential $\Sigma r_i(t)$. As the noise increases linearly the threshold begins to

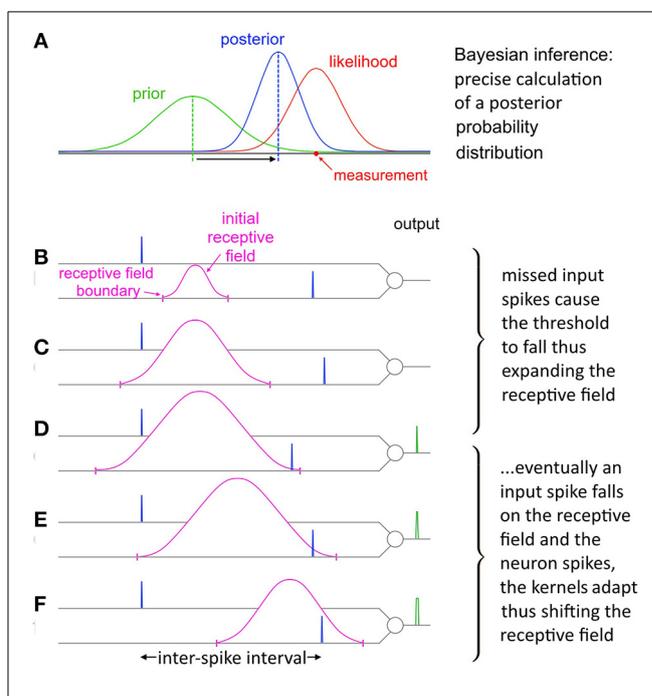
fall gradually. Missing output spikes in the $s(t)$ spike train correspond to input spikes that have landed outside the receptive field boundaries. **(C)** The output pulse width (green = observed, red = running average) increases with increasing signal noise. As with the receptive field size, the output 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.

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 9A 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 9C** 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 Θ_{fall} . When presented with noiseless patterns this fall would be balanced almost exactly by the threshold rise due to the Θ_{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 $\Theta_{rise}:\Theta_{fall}$, 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

**FIGURE 10 | 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.

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 10**.

LEARNING IN THE PRESENCE OF POISSON SPIKE NOISE AND MISSING TARGET SPIKES

In addition to robustness to temporal jitter in the put 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 (McDonnell and Ward, 2011; Hunsberger et al., 2014; Maass, 2014). In

the neuromorphic context the performance of neural network architectures in the presence of noise is well documented (Hamilton and Tapson, 2011; Hamilton et al., 2014; Marr and Hasler, 2014). 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 11** illustrates how different signal to noise ratios can affect SKAN's ability to learn an embedded spatio-temporal spike pattern.

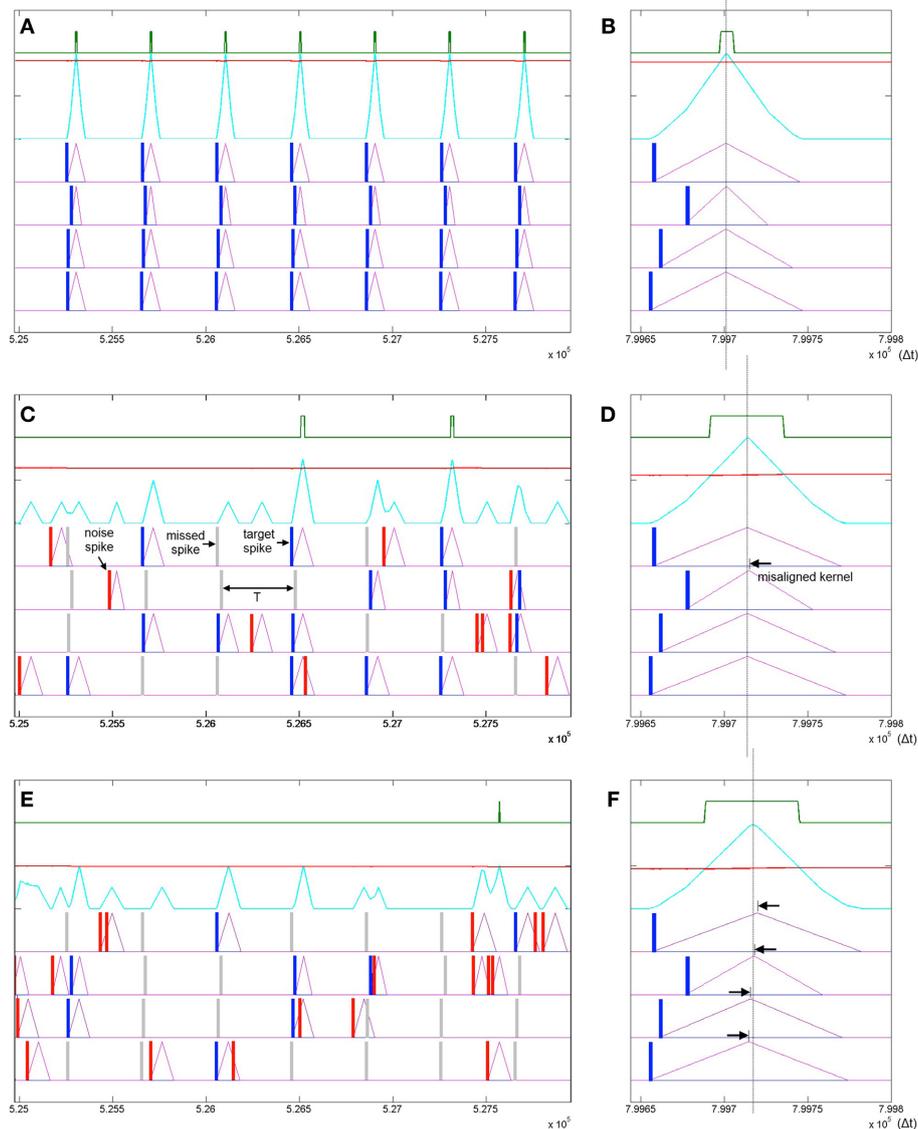


FIGURE 11 | Learning spatio-temporal spike patterns in the presence of both Poisson spike noise and missing target spikes. Panel (A) shows the presentation of seven patterns in the middle of a simulation sequence with a noiseless environment. The kernels are highly adapted ($\Delta r_2 = \Delta r_{max}$), 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 pattern (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 $P(\text{signal}) = 0.5$ and the Poisson rate is also half such that $P(\text{noise}) = 0.5/T$. 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.

To quantify the performance of SKAN in the presence of Poisson noise and missing target spikes a series of simulations each comprising of 2000 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 12**.

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 (Gupta and Long, 2009; Nessler et al., 2013), 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 [$s_n(t - 1) = 0$].

$$s_n(t) = \begin{cases} 1 & \text{if } \sum_i r_{n,i}(t) > \Theta_n(t - 1) \\ & \wedge (inh(t - 1) = 0 \vee s_n(t - 1) = 1) \\ 0 & \text{else} \end{cases} \quad (7)$$

$$inh(t) = \begin{cases} inh_{max} & \text{if } \bigcup_n s_n(t) = 1 \\ inh(t - 1) - inh_{decay} & \text{if } inh(t - 1) > 0 \\ 0 & \text{else} \end{cases} \quad (8)$$

As shown in **Figure 13** 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 13** the decay only begins at the end of the pulse making the inhibitory signal operate as

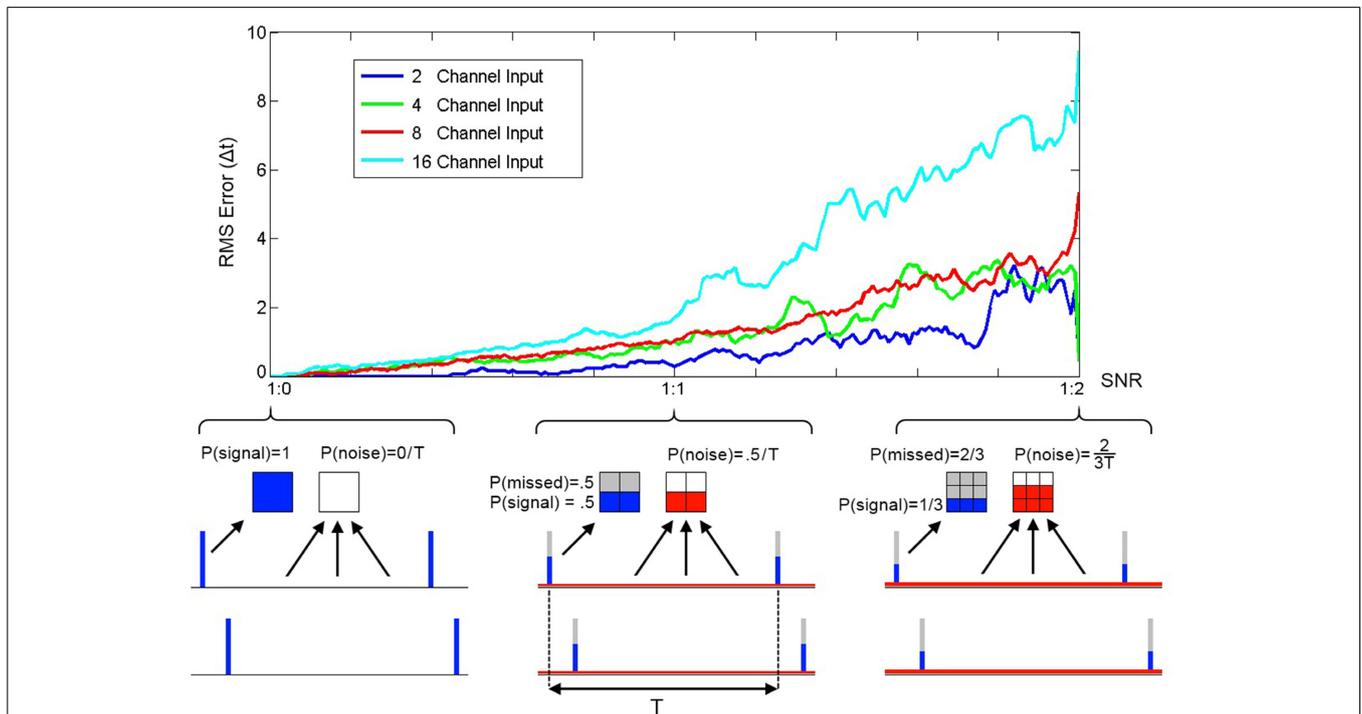
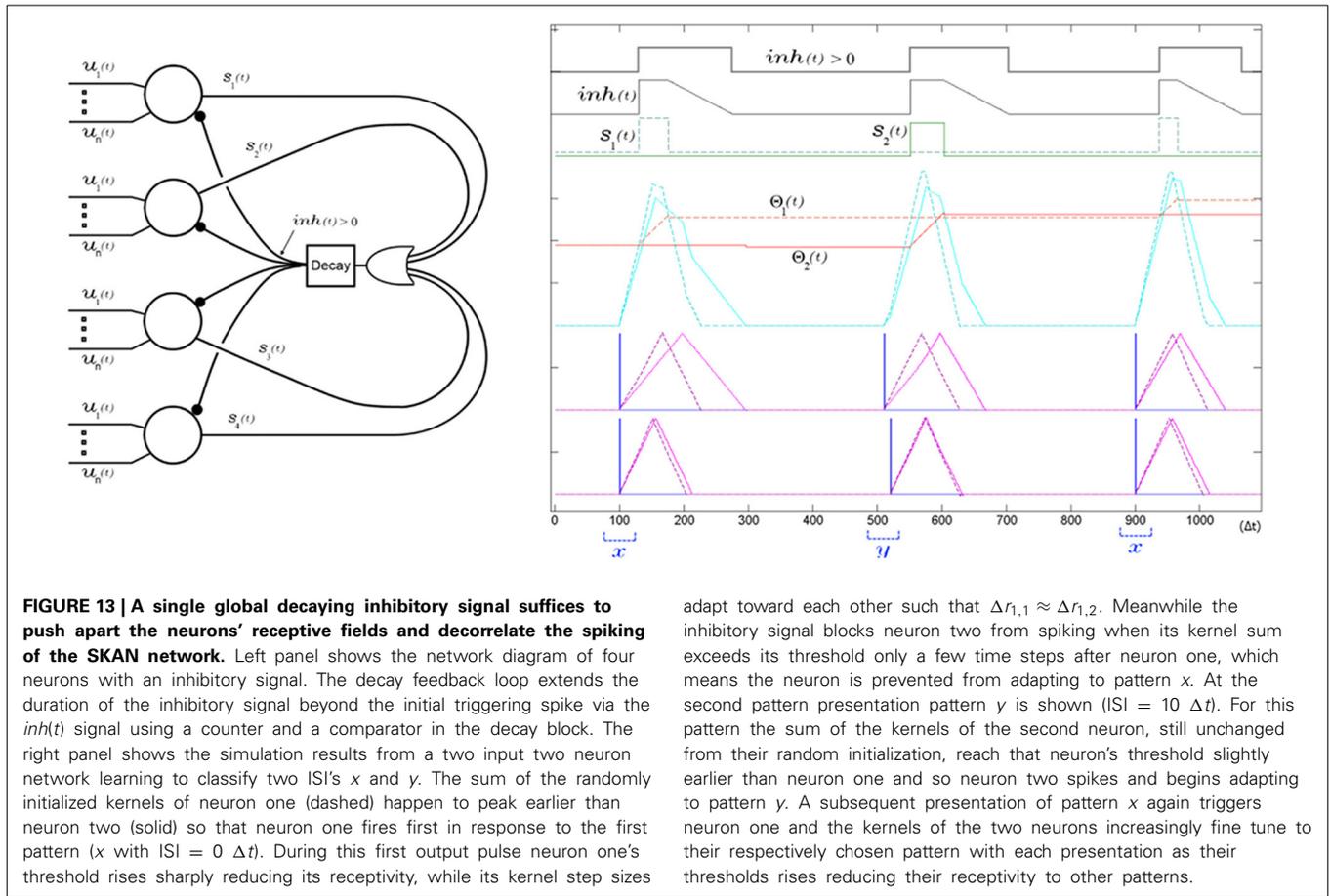


FIGURE 12 | RMS error between receptive field maxima and target spike patterns as a function of spike signal to noise ratio. The three bottom panels show the spike probability distributions at three points along the SNR axis. The signal spikes (blue), missed spikes (gray), and noise spikes (red) are 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.



adapt toward 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 ($ISI = 10 \Delta t$). 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 to their respectively chosen pattern with each presentation as their thresholds rises reducing their receptivity to other patterns.

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.

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 multi-neuron 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.

$$\Theta_n(t) = \begin{cases} \Theta(t-1) + \Theta_{rise} & \text{if } \sum_i r_i(t) > \Theta(t-1) \\ & \wedge (inh(t-1) = 0 \vee s_n(t-1) = 1) \\ \Theta(t-1) - \Theta_{fall} & \text{if } (\sum_i r_i(t) = 0 \wedge \sum_i r_i(t-1) > 0) \\ & \wedge inh(t-1) = 0 \\ & \vee (s(t) = 0 \wedge s(t-1) = 1) \\ \Theta(t-1) & \text{else} \end{cases} \quad (9)$$

Such a global inhibitory signal has been utilized in LIF neurons (Afshar et al., 2012; Tapson and van Schaik, 2012) and synaptic weight STDP neurons as a means of decorrelating neuronal firing patterns (Masquelier et al., 2009; Habenschuss et al., 2013). 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 (Afshar et al., 2012; Tapson and van Schaik, 2012) 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.

$$\begin{aligned} \text{total connections} &= (\text{number of input channel} + 2) \\ &\times \text{number of neurons} \end{aligned} \quad (10)$$

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.

RESULTS

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.

Hardware efficiency through 1-to-1 neuron to pattern allocation at the local level

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 14**.

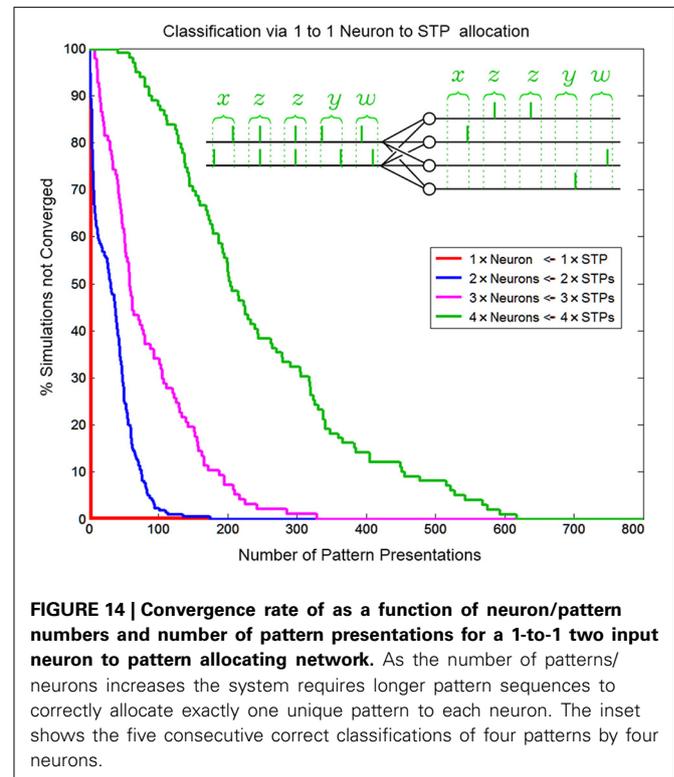


Table 1 | Parameter values used for all results.

Parameter	Value	Description
ddr	1	Change in the kernel step size. Higher value results in faster adaptation; lower values are more robust to noise
$w(r_{max})$	10,000	Maximum kernel height (synaptic weight)
r_{min}	0	The kernel signal $r(t)$ saturates at zero
$\Delta r_{i,n}(t=0)$	$100 \times (1 + \text{rand})$	Initial kernel step size (For each input i to each neuron n) 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}	$40 \times \text{inputs}$	Rise in threshold during output spike, where inputs is the number of input channels per neuron
Θ_{fall}	$100 \times \text{inputs}$	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: $inh_{max}/inh_{decay} = \min[\Delta r_{i,n}(t=0)]$
T	$400 \Delta t$	Time between pattern presentations

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. This means that attempting to test SKANs in the traditional neural network sense by switching off a network's adaptation mechanisms would disable the system. 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 20 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 14**. 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 14**. Simulations were terminated once a network had converged. The number of consecutive patterns was chosen as 20 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

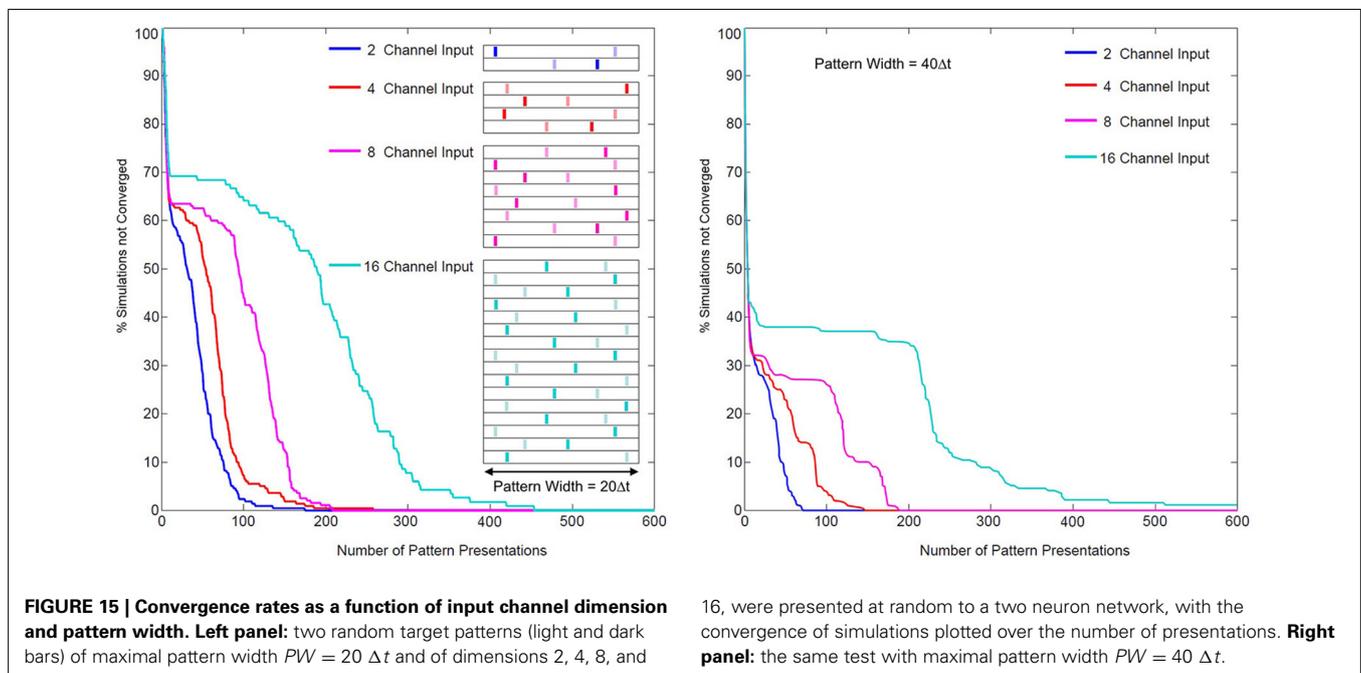
The problem of coordinating multiple synapses for unsupervised neuronal classification in SNN models, whether through simply learning synaptic weights or through more complex pathways, is difficult (Jimenez Rezende and Gerstner, 2014). In SKAN the hybrid synapto-dendritic kernel adaptation produces convergence profiles shown in **Figure 15**. These results show how the convergence profiles of SKAN change with the number of active

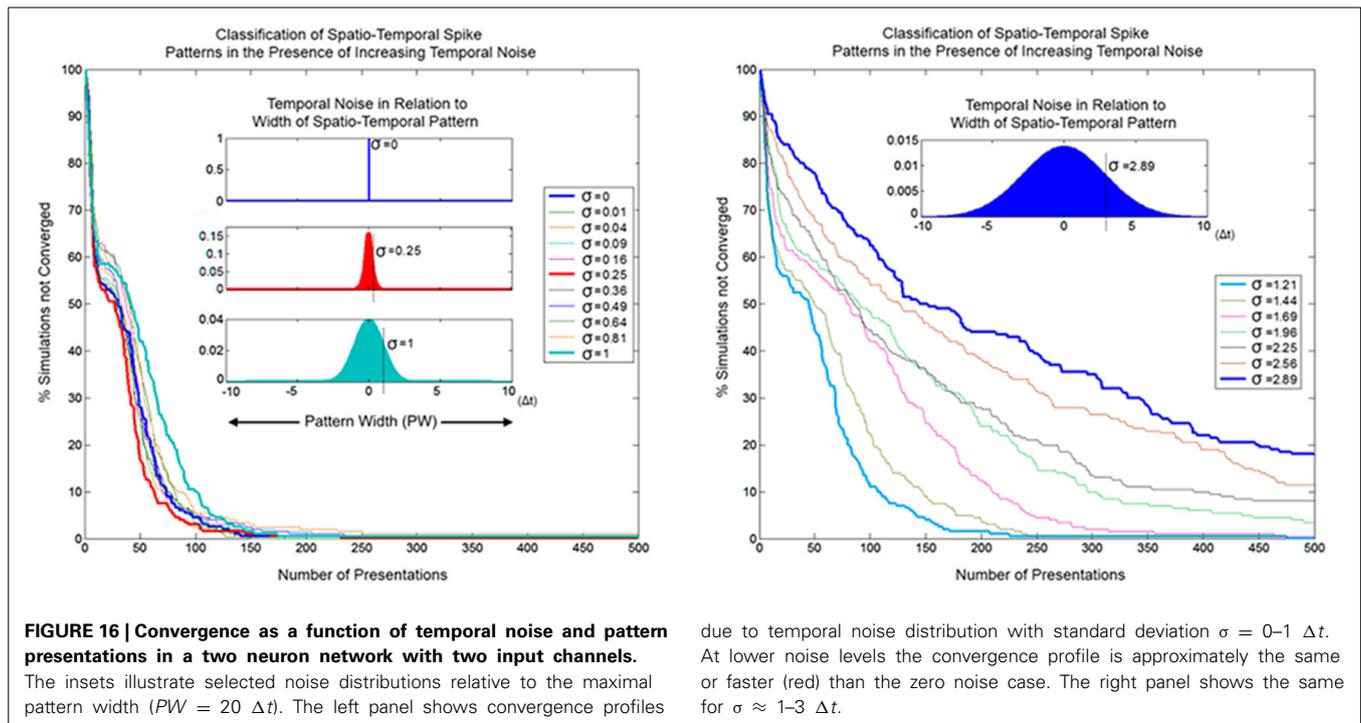
input channels. Additionally, the right panel in **Figure 15** shows the effect of increasing the resolution of the spatio-temporal pattern. Doubling the number of time steps in the maximal width of the target pattern PW , results in improved convergence profiles.

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 16** where two neurons act as two Kalman filters with shared inputs attempting to learn the statistics of two noisy but distinct ISI generating processes.

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 scheme, the robustness of SKAN's learning algorithm to temporal noise 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.

IMPLEMENTATION IN FPGA

SKAN was implemented in an Altera Cyclone-V GX FPGA, a low-end FPGA containing 77,000 programmable logic elements (LEs). The functions of SKAN were programmed based on the equations described in the earlier sections, written using the Verilog hardware description language, with no optimization techniques employed. A key feature of this design is that no multipliers are required: SKAN is executed entirely using simple summation and logical operations only, thus significantly reducing computational complexity and hardware resources. Registers are used to store required design parameters. **Table 2** shows the utilization of the FPGA in terms of registers, adaptive logic modules (ALMs) and the percentage of resources used for SKAN modules containing different number of synapses. From this we can see that SKAN is efficient in its usage of hardware resources. Results from the FPGA are identical to the simulated results as integers were used for both the simulations and the FPGA and therefore no approximations were required. Integers were used to avoid floating point operations, thereby reducing computation. An efficient use of hardware resources, reduced computational effort, and its ease of implementation make SKAN an attractive neuromorphic solution in terms of both cost and performance.

Table 2 | Altera Cyclone V FPGA resource usage for a SKAN neuron with different number of synapses.

No. of synapses	FPGA resource usage		
	Single bit registers	Adaptive logic modules (ALMs)*	Usage percentage (%)
1	48	189	0.6
2	72	297	1.0
4	121	501	1.7
8	218	922	3.2
16	411	1580	5.4

*An ALM is equivalent to 2.65 logic elements (LEs).

DISCUSSION

As outlined in the introduction a limiting factor in many neuromorphic systems is the large number of complex synapses which require multipliers and high connectivity networks required for robust performance. A simple solution to this challenge has been to physically implement one or a few instances of these complex elements and use time multiplexing and AER to generate larger virtual networks. The kernels of SKAN which do not require multipliers allow more synapses to be physically realized in hardware while their adaptability means that better performance can be achieved using fewer synapses. Furthermore, the time based operation of the neurons reduces the required connectivity. This potentially allows entire networks to be physically implemented in hardware. Such small or medium sized networks whose behaviors have been described in this report can then be cascaded or multiplexed to form larger networks. Such solutions could potentially occupy a middle ground between fully hardware implemented

networks with limited connectivity but high bandwidth and single neuron realizations with high connectivity and limited operating speeds. While the focus of this introductory report is on characterization of small non-optimized SKAN networks, preliminary work on the application of the architecture to larger, more difficult recognition tasks such as unsupervised learning of the MNIST dataset has not revealed any limits to the capabilities of larger, more optimized networks. Future work will focus on comparison of SKAN networks to other neural network solutions on established datasets, comparison of the inference capabilities of the neuron to optimal probabilistic estimators and the investigation of the combined effects of adaptation of SKAN's kernels and the adaptation of its synaptic weight parameter w which allows encoding of synaptic signal to noise ratios for each input channel.

CONCLUSION

In this paper we have presented the SKAN, a neuromorphic implementation of a spiking neuron that performs statistical inference and unsupervised learning and spatio-temporal spike pattern classification. The use of simple adaptable kernels was shown to represent an efficient 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. Finally it was shown that SKAN is hardware efficient and easily implemented on an FPGA.

REFERENCES

- Afshar, S., Cohen, G. K., Wang, R. M., van Schaik, A., Tapson, J., Lehmann, T., et al. (2013). The ripple pond: enabling spiking networks to see. *Front. Neurosci.* 7:212. doi: 10.3389/fnins.2013.00212
- Afshar, S., Kavehei, O., van Schaik, A., Tapson, J., Skafidas, S., and Hamilton, T. J. (2012). "Emergence of competitive control in a memristor-based neuromorphic circuit," in *The 2012 International Joint Conference on Neural Networks (IJCNN)* (Brisbane, QLD), 1–8.
- Arthur, J. V., Merolla, A., Akopyan, F., Alvarez, R., Cassidy, A., Chandra, S., et al. (2012). "Building block of a programmable neuromorphic substrate: a digital neurosynaptic core," in *Proceedings of the International Joint Conference on Neural Networks* (Brisbane, QLD).
- Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., and Friston, K. J. (2012). Canonical microcircuits for predictive coding. *Neuron* 76, 695–711. doi: 10.1016/j.neuron.2012.10.038
- Beyeler, M., Dutt, N. D., and Krichmar, J. L. (2013). Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule. *Neural Netw.* 48, 109–124. doi: 10.1016/j.neunet.2013.07.012
- Bhatt, D. H., Zhang, S., and Gan, W. B. (2009). Dendritic spine dynamics. *Annu. Rev. Physiol.* 71, 261–282. doi: 10.1146/annurev.physiol.010908.163140
- Boerlin, M., Machens, C. K., and Denève, S. (2013). Predictive coding of dynamical variables in balanced spiking networks. *PLoS Comput. Biol.* 9:e1003258. doi: 10.1371/journal.pcbi.1003258
- Brunel, N., Hakim, V., and Richardson, M. J. E. (2014). Single neuron dynamics and computation. *Curr. Opin. Neurobiol.* 25, 149–155. doi: 10.1016/j.conb.2014.01.005
- Bullmore, E. T., and Sporns, O. (2012). The economy of brain network organization. *Nat. Rev. Neurosci.* 13, 336–349. doi: 10.1038/nrn3214
- Chicca, E., Stefanini, F., Bartolozzi, C., and Indiveri, G. (2014). Neuromorphic electronic circuits for building autonomous cognitive systems. *Proc. IEEE* 99, 1–22. doi: 10.1109/JPROC.2014.2313954
- Colgan, L., and Yasuda, R. (2014). Plasticity of dendritic spines: subcompartmentalization of signaling. *Annu. Rev. Physiol.* 76, 365–385. doi: 10.1146/annurev-physiol-021113-170400
- Corneil, D. S., Neftci, E., Indiveri, G., and Pfeiffer, M. (2014). *Learning, Inference, and Replay of Hidden State Sequences in Recurrent Spiking Neural Networks*. Salt Lake City, UT.
- Denève, S. (2008). Bayesian spiking neurons I: inference. *Neural Comput.* 20, 91–117. doi: 10.1162/neco.2008.20.1.91
- Dowrick, T., Hall, S., and McDaid, L. (2013). A simple programmable axonal delay scheme for spiking neural networks. *Neurocomputing* 108, 79–83. doi: 10.1016/j.neucom.2012.12.004
- Fontaine, B., Peña, J. L., and Brette, R. (2014). Spike-threshold adaptation predicted by membrane potential dynamics *in vivo*. *PLoS Comput. Biol.* 10:e1003560. doi: 10.1371/journal.pcbi.1003560
- George, S., Hasler, J., Koziol, S., Nease, S., and Ramakrishnan, S. (2013). Low power dendritic computation for wordspotting. *J. Low Power Electron. Appl.* 3, 73–98. doi: 10.3390/jlpea3020073
- Gupta, A., and Long, L. N. (2009). Hebbian learning with winner take all for spiking neural networks. *Neural Netw.* 81, 1054–1060. doi: 10.1109/IJCNN.2009.5178751
- Gütig, R. (2014). To spike, or when to spike? *Curr. Opin. Neurobiol.* 25, 134–139. doi: 10.1016/j.conb.2014.01.004
- Habenschuss, S., Puh, H., and Maass, W. (2013). Emergence of optimal decoding of population codes through STDP. *Neural Comput.* 25, 1371–1407. doi: 10.1162/NECO_a_00446
- Hamilton, T. J., Afshar, S., van Schaik, A., and Tapson, J. (2014). Stochastic electronics: a neuro-inspired design paradigm for integrated circuits. *Proc. IEEE* 102, 843–859. doi: 10.1109/JPROC.2014.2310713
- Hamilton, T. J., and Tapson, J. (2011). "A neuromorphic cross-correlation chip," in *Proceedings - IEEE International Symposium on Circuits and Systems* (Rio de Janeiro), 865–868.
- Harnett, M. T., Makara, J. K., Spruston, N., Kath, W. L., and Magee, J. C. (2012). Synaptic amplification by dendritic spines enhances input cooperativity. *Nature* 491, 599–602. doi: 10.1038/nature11554
- Herz, A. V. M., Gollisch, T., Machens, C. K., and Jaeger, D. (2006). Modeling single-neuron dynamics and computations: a balance of detail and abstraction. *Science* 314, 80–85. doi: 10.1126/science.1127240
- Hsu, C. C., Parker, A. C., and Joshi, J. (2010). "Dendritic computations, dendritic spiking and dendritic plasticity in nanoelectronic neurons," in *Midwest Symposium on Circuits and Systems* (Seattle, WA), 89–92.
- Hunsberger, E., Scott, M., and Eliasmith, C. (2014). The competing benefits of noise and heterogeneity in neural coding. *Neural Comput.* 26, 1600–1623. doi: 10.1162/NECO_a_00621
- Hussain, S., Basu, A., Wang, R. M., and Julia Hamilton, T. (2014). Delay learning architectures for memory and classification. *Neurocomputing* 138, 14–26. doi: 10.1016/j.neucom.2013.09.052
- Indiveri, G., Linares-Barranco, B., Hamilton, T. J., van Schaik, A., Etienne-Cummings, R., Delbruck, T., et al. (2011). Neuromorphic silicon neuron circuits. *Front. Neurosci.* 5:73. doi: 10.3389/fnins.2011.00073
- Indiveri, G., Linares-Barranco, B., Legenstein, R., Deligeorgis, G., and Prodromakis, T. (2013). Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology* 24:384010. doi: 10.1088/0957-4484/24/38/384010
- Izhikevich, E. M. (2006). Polychronization: computation with spikes. *Neural Comput.* 18, 245–282. doi: 10.1162/089976606775093882
- Jaeger H. (2001). *The Echo State Approach to Analyzing and Training Recurrent Neural Networks*. Technical Report, GMD Report 148, GMD-German National Research Institute for Computer Science.
- Jimenez Rezende, D., and Gerstner, W. (2014). Stochastic variational learning in recurrent spiking networks. *Front. Comput. Neurosci.* 8:38. doi: 10.3389/fncom.2014.00038
- Kappel, D., Nessler, B., and Maass, W. (2014). STDP installs in winner-take-all circuits an online approximation to hidden markov model learning. *PLoS Comput. Biol.* 10:e1003511. doi: 10.1371/journal.pcbi.1003511
- Kasabov, N., Dhoble, K., Nuntalid, N., and Indiveri, G. (2013). Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition. *Neural Netw.* 41, 188–201. doi: 10.1016/j.neunet.2012.11.014
- Kasai, H., Fukuda, M., Watanabe, S., Hayashi-Takagi, A., and Noguchi, J. (2010a). Structural dynamics of dendritic spines in memory and cognition. *Trends Neurosci.* 33, 121–129. doi: 10.1016/j.tins.2010.01.001
- Kasai, H., Hayama, T., Ishikawa, M., Watanabe, S., Yagishita, S., and Noguchi, J. (2010b). Learning rules and persistence of dendritic spines. *Eur. J. Neurosci.* 32, 241–249. doi: 10.1111/j.1460-9568.2010.07344.x

- Kim, J. K., Knag, P., Chen, T., and Zhang, Z. (2014). Efficient hardware architecture for sparse coding. *IEEE Trans. Signal Process.* 62, 4173–4186. doi: 10.1109/TSP.2014.2333556
- Kuhlmann, L., Hauser-Raspe, M., Manton, J. H., Grayden, D. B., Tapson, J., and van Schaik, A. (2014). Approximate, computationally efficient online learning in Bayesian spiking neurons. *Neural Comput.* 26, 472–496. doi: 10.1162/NECO_a_00560
- Laughlin, S. B. (2001). Energy as a constraint on the coding and processing of sensory information. *Curr. Opin. Neurobiol.* 11, 475–480. doi: 10.1016/S0959-4388(00)00237-3
- Lee, K. F. H., Soares, C., and Béique, J. C. (2012). Examining form and function of dendritic spines. *Neural Plasticity* 2012:704103. doi: 10.1155/2012/704103
- Lee, K. F. H., Soares, C., and Béique, J.-C. (2014). Tuning into diversity of homeostatic synaptic plasticity. *Neuropharmacology* 78, 31–37. doi: 10.1016/j.neuropharm.2013.03.016
- Levy, W. B., and Baxter, R. A. (1996). Energy efficient neural codes. *Neural Comput.* 8, 531–543. doi: 10.1162/neco.1996.8.3.531
- Llinas, R. (2008). Neuron. *Scholarpedia* 3:1490. doi: 10.4249/scholarpedia.1490
- Losonczy, A., Makara, J. K., and Magee, J. C. (2008). Compartmentalized dendritic plasticity and input feature storage in neurons. *Nature* 452, 436–441. doi: 10.1038/nature06725
- Maass, W. (2014). Noise as a resource for computation and learning in networks of spiking neurons. *Proc. IEEE* 102, 860–880. doi: 10.1109/JPROC.2014.2310593
- Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* 14, 2531–2560. doi: 10.1162/089976602760407955
- Marr, B., and Hasler, J. (2014). Compiling probabilistic, bio-inspired circuits on a field programmable analog array. *Front. Neurosci.* 8:86. doi: 10.3389/fnins.2014.00086
- Masquelier, T., Guyonneau, R., and Thorpe, S. J. (2009). Competitive STDP-based spike pattern learning. *Neural Comput.* 21, 1259–1276. doi: 10.1162/neco.2008.06-08-804
- McDonnell, M. D., and Ward, L. M. (2011). The benefits of noise in neural systems: bridging theory and experiment. *Nat. Rev. Neurosci.* 12, 415–426. doi: 10.1038/nrn3061
- Merolla, P., Arthur, J., Akopyan, F., Imam, N., Manohar, R., and Modha, D. S. (2011). “A digital neuromorphic core using embedded crossbar memory with 45nm” in *Proceedings of the Custom Integrated Circuits Conference* (San Jose, CA).
- Misra, J., and Saha, I. (2010). Artificial neural networks in hardware: a survey of two decades of progress. *Neurocomputing* 74, 239–255. doi: 10.1016/j.neucom.2010.03.021
- Mitra, S., Fusi, S., and Indiveri, G. (2009). Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI. *IEEE Trans. Biomed. Circuits Syst.* 3, 32–42. doi: 10.1109/TBCAS.2008.2005781
- Nessler, B., Pfeiffer, M., Buesing, L., and Maass, W. (2013). Bayesian computation emerges in generic cortical microcircuits through Spike-Timing-Dependent Plasticity. *PLoS Comput. Biol.* 9:e1003037. doi: 10.1371/journal.pcbi.1003037
- O’Connor, P., Neil, D., Liu, S.-C., Delbruck, T., and Pfeiffer, M. (2013). Real-time classification and sensor fusion with a spiking deep belief network. *Front. Neurosci.* 7:178. doi: 10.3389/fnins.2013.00178
- Papoutsis, A., Kastellakis, G., Psarrou, M., Anastasakis, S., and Poirazi, P. (2014). Coding and decoding with dendrites. *J. Physiol. Paris* 108, 18–27. doi: 10.1016/j.jphysparis.2013.05.003
- Paulin, M. G., and van Schaik, A. (2014). *Bayesian Inference with Spiking Neurons*. arXiv preprint arXiv:1406.5115
- Pfeil, T., Potjans, T. C., Schrader, S., Potjans, W., Schemmel, J., Diesmann, M., et al. (2012). Is a 4-bit synaptic weight resolution enough? – constraints on enabling spike-timing dependent plasticity in neuromorphic hardware. *Front. Neurosci.* 6:90. doi: 10.3389/fnins.2012.00090
- Pouget, A., Beck, J. M., Ma, W. J., and Latham, P. E. (2013). Probabilistic brains: knowns and unknowns. *Nat. Neurosci.* 16, 1170–1178. doi: 10.1038/nn.3495
- Rahimi Azghadi, M., Iannella, N., Al-Sarawi, S. F., Indiveri, G., and Abbott, D. (2014). Spike-based synaptic plasticity in silicon: design, implementation, application, and challenges. *Proc. IEEE* 102, 717–737. doi: 10.1109/JPROC.2014.2314454
- Ramakrishnan, S., Wunderlich, R., Hasler, J., and George, S. (2013). Neuron array with plastic synapses and programmable dendrites. *IEEE Trans. Biomed. Circuits Syst.* 7, 631–642. doi: 10.1109/TBCAS.2013.2282616
- Rocheffort, N. L., and Konnerth, A. (2012). Dendritic spines: from structure to *in vivo* function. *EMBO Rep.* 13, 699–708. doi: 10.1038/embor.2012.102
- Scholze, S. (2011). VLSI implementation of a 2.8 Gevent/s packet-based AER interface with routing and event sorting functionality. *Front. Neurosci.* 5:117. doi: 10.3389/fnins.2011.00117
- Seo, J. S., Brezzo, B., Liu, Y., Parker, B. D., Esser, S. K., Montoyo, R. K., et al. (2011). “A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons,” in *Proceedings of the Custom Integrated Circuits Conference* (San Jose, CA).
- Serrano-Gotarredona, T., Prodromakis, T., and Linares-Barranco, B. (2013). A proposal for hybrid memristor-CMOS spiking neuromorphic learning systems. *IEEE Cir. Syst. Mag.* 13, 74–88. doi: 10.1109/MCAS.2013.2256271
- Shah, M. M., Hammond, R. S., and Hoffman, D. A. (2010). Dendritic ion channel trafficking and plasticity. *Trends Neurosci.* 33, 307–316. doi: 10.1016/j.tins.2010.03.002
- Sheik, S., Chicca, E., and Indiveri, G. (2012). “Exploiting device mismatch in neuromorphic VLSI systems to implement axonal delays,” in *The 2012 International Joint Conference on Neural Networks (IJCNN)* (Brisbane), 1–6.
- Sheik, S., Pfeiffer, M., Stefanini, F., and Indiveri, G. (2013). “Spatio-temporal spike pattern classification in neuromorphic systems,” in *Biomimetic and Biohybrid Systems*, eds N. F. Lepora, A. Mura, H. G. Krapp, P. F. M. J. Verschure, and T. J. Prescott (Heidelberg: Springer), 262–273. doi: 10.1007/978-3-642-39802-5_23
- Silver, R. A. (2010). Neuronal arithmetic. *Nat. Rev. Neurosci.* 11, 474–489. doi: 10.1038/nrn2864
- Sjöström, P. J., Rancz, E. A., Roth, A., and Häusser, M. (2008). Dendritic excitability and synaptic plasticity. *Physiol. Rev.* 88, 769–840. doi: 10.1152/physrev.00016.2007
- Smith, S. L., Smith, I. T., Branco, T., and Häusser, M. (2013). Dendritic spikes enhance stimulus selectivity in cortical neurons *in vivo*. *Nature* 503, 115–120. doi: 10.1038/nature12600
- Sofatzis, R. J., Afshar, S., and Hamilton, T. J. (2014a). “The synaptic kernel adaptation network,” in *The Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)* (Melbourne, VIC).
- Sofatzis, R. J., Afshar, S., and Hamilton, T. J. (2014b). “Rotationally invariant vision recognition with neuromorphic transformation and learning networks,” in *The Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)* (Melbourne, VIC).
- Spruston, N. (2008). Pyramidal neurons: dendritic structure and synaptic integration. *Nat. Rev. Neurosci.* 9, 206–221. doi: 10.1038/nrn2286
- Tapson, J. C., Cohen, G. K., Afshar, S., Stiefel, K. M., Buskila, Y., Wang, R. M., et al. (2013). Synthesis of neural networks for spatio-temporal spike pattern recognition and processing. *Front. Neurosci.* 7:153. doi: 10.3389/fnins.2013.00153
- Tapson, J., and van Schaik, A. (2012). “An asynchronous parallel neuromorphic ADC architecture,” in *2012 IEEE International Symposium on Circuits and Systems (Seoul)*, 2409–2412. doi: 10.1109/ISCAS.2012.6271783
- Topol, A. W., La Tulipe, D. C., Shi, L., Frank, D. J., Bernstein, K., Steen, S. E., et al. (2006). Three-dimensional integrated circuits. *IBM J. Res. Dev.* 50, 491–506. doi: 10.1147/rd.504.0491
- Tully, P. J., Hennig, M. H., and Lansner, A. (2014). Synaptic and nonsynaptic plasticity approximating probabilistic inference. *Front. Synaptic Neurosci.* 6:8. doi: 10.3389/fnsyn.2014.00008
- Van Rullen, R., and Thorpe, S. J. (2001). Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex. *Neural Comput.* 13, 1255–1283. doi: 10.1162/08997660152002852
- Wang, R. M., Hamilton, T. J., Tapson, J. C., and van Schaik, A. (2014). A mixed-signal implementation of a polychronous spiking neural network with delay adaptation. *Front. Neurosci.* 8:51. doi: 10.3389/fnins.2014.00051
- Wang, Y., and Liu, S. C. (2013). Active processing of spatio-temporal input patterns in silicon dendrites. *IEEE Trans. Biomed. Circuits Syst.* 7, 307–318. doi: 10.1109/TBCAS.2012.2199487
- Xu, W., Morishita, W., Buckmaster, S., Pang, Z. P., Malenka, R. C., and Südhof, T. C. (2012). Distinct neuronal coding schemes in memory revealed by selective erasure of fast synchronous synaptic transmission. *Neuron* 73, 990–1001. doi: 10.1016/j.neuron.2011.12.036
- Yegnanarayana, B. (1999). *Artificial Neural Networks*. New Delhi: Prentice-Hall India.
- Yoshihara, Y., De Roo, M., and Muller, D. (2009). Dendritic spine formation and stabilization. *Curr. Opin. Neurobiol.* 19, 146–153. doi: 10.1016/j.conb.2009.05.013

- Yu, Y., and Lee, T. (2003). Dynamical mechanisms underlying contrast gain control in single neurons. *Phys. Rev. E* 68:011901. doi: 10.1103/PhysRevE.68.011901
- Zamarreno-Ramos, C., Linares-Barranco, A., Serrano-Gotarredona, T., and Linares-Barranco, B. (2013). Multicasting mesh AER: a scalable assembly approach for reconfigurable neuromorphic structured AER systems. Application to ConvNets. *IEEE Trans. Biomed. Circuits Syst.* 7, 82–102. doi: 10.1109/TBCAS.2012.2195725
- Zhu, J., and Sutton, P. (2003). FPGA implementations of neural networks - A survey of a decade of progress. *Lect. Notes Comput. Sci.* 2778, 1062–1066. doi: 10.1007/978-3-540-45234-8_120

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 14 August 2014; accepted: 05 November 2014; published online: 25 November 2014.

Citation: Afshar S, George L, Tapson J, van Schaik A and Hamilton TJ (2014) Racing to learn: statistical inference and learning in a single spiking neuron with adaptive kernels. Front. Neurosci. 8:377. doi: 10.3389/fnins.2014.00377

This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience.

Copyright © 2014 Afshar, George, Tapson, van Schaik and Hamilton. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.