A Spiking Neural Network System for Robust Sequence Recognition

Qiang Yu, *Student Member, IEEE*, Rui Yan, *Member, IEEE*, Huajin Tang, *Member, IEEE*, Kay Chen Tan, *Fellow, IEEE*, and Haizhou Li, *Fellow, IEEE*

Abstract—This paper proposes a biologically plausible network architecture with spiking neurons for sequence recognition. This architecture is a unified and consistent system with functional parts of sensory encoding, learning, and decoding. This is the first systematic model attempting to reveal the neural mechanisms considering both the upstream and the downstream neurons together. The whole system is a consistent temporal framework, where the precise timing of spikes is employed for information processing and cognitive computing. Experimental results show that the system is competent to perform the sequence recognition, being robust to noisy sensory inputs and invariant to changes in the intervals between input stimuli within a certain range. The classification ability of the temporal learning rule used in the system is investigated through two benchmark tasks that outperform the other two widely used learning rules for classification. The results also demonstrate the computational power of spiking neurons over perceptrons for processing spatiotemporal patterns. In summary, the system provides a general way with spiking neurons to encode external stimuli into spatiotemporal spikes, to learn the encoded spike patterns with temporal learning rules, and to decode the sequence order with downstream neurons. The system structure would be beneficial for developments in both hardware and software.

Index Terms—Encoding, pattern recognition, sequence recognition, spiking neural networks (SNNs), temporal learning.

I. INTRODUCTION

S PIKING neurons have received significant attentions during the past several decades as they are more biologically plausible and computationally powerful than traditional neurons such as perceptrons [1]–[3]. Like biological neurons, spiking neurons are capable of processing spikes (also called action potentials), which are believed to be the principal feature of information transmission in neural systems [4].

With these building units of spiking neurons, the brain presents remarkably cognitive abilities, including recognition, navigation, and memory [5]. Sequence recognition is also one of these cognitive abilities, and it refers to the ability to detect and recognize the temporal order of discrete elements occurring in sequence. Such sequence decoding operations are required for processing temporally complex stimuli, such as speech, where important information is embedded in patterns over time [6]–[9]. However, the biophysical mechanisms by which neural circuits detect and recognize sequences of external stimuli are poorly understood.

Sequence information processing is a general problem that the brain needs to solve. Several approaches with the design of traditional artificial neural network structures [10], [11] have been considered and implemented for processing temporal information. The functionality of the brain for sequence recognition is mimicked through the artificial structures. However, these neural structures do not consider the building units of spiking neurons. Recognizing the sequences of external stimuli with spiking features in the brain still remains an open question. Numerous studies have put efforts separately to computational mechanisms with spiking neurons, where some focus on neural representations of the external information [12] while the others focus on the internal processing of either upstream or downstream neurons [2], [3], [7], [13]–[18]. Recognizing the sequence of incoming stimuli at a systematic level of view has not been established. Such a systematic system should incorporate several functional parts, including neural coding, learning, and decoding. With these functionalities integrating together, the system could process information from levels of upstream encoding neurons to levels of downstream decoding neurons.

Neural coding refers to the scheme by which the information of external stimuli is represented by spikes. Among different coding schemes, the rate code and the temporal code are most widely studied [12], [19]. The rate code is a traditional coding scheme, assuming that most, if not all, information about the stimulus is contained in the firing rate of the neuron. Although there are few doubts as to the relevance of the rate code, it neglects the extra information embedded in the temporal structure. Recently, increasing evidence shows that neural systems convey information through the precise timing of spikes, which supports the hypothesis of the temporal code [12], [20], [21]. Examples of experimental observations include the retina [22], the lateral geniculate nucleus [23], and the visual cortex [24]. Regardless of these recent biological findings, the temporal code can offer significant

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

2   IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS

computational advantages over the rate code, such as information capacity and efficiency [19], [25]. According to the temporal code, the complex external stimuli can thus drive different sensory neurons to spike at different time, with this fact forming a spatiotemporal representation of the external stimuli. However, due to the complexity of processing the spatiotemporal spike patterns, the development of efficient learning algorithms is demanded.

Some evolutionary approaches have been used to train the spiking neural networks (SNNs) by optimizing network parameters through genetic algorithms [26]. Although such evolutionary approaches would provide high performance in several specific tasks [26], they are extremely time consuming to train and are very sensitive to parameter values and initialization [27], [28]. Considering the biological realism, an entire family of spike-timing-dependent plasticity (STDP) rules [29], [30] have been widely studied recently, as well as their extensions for practical applications [27], [28]. An STDP characterizes synaptic changes in terms of the temporal contiguity of the presynaptic and postsynaptic spikes, with causal pairing order inducing long-term potentiation, and with the reversed order leading to long-term depression [21], [29]. However, neurons implementing STDP will not always guarantee convergence during training [27], [28], and a suitable balance of many parameters is required [31].

Some other learning algorithms [2], [3], [13]–[16] have also been proposed for spiking neurons to process spatiotemporal spike patterns. The tempotron rule [3] has shown how a neuron can give a binary response to precise-spike spatiotemporal patterns. However, the tempotron is unable to produce precise-timing spikes that can further serve as the information-carrying input for subsequent neurons. By contrast, the SpikeProp rule [2] can train the neuron to perform a spatiotemporal classification by emitting single spikes at the desired firing time. Nevertheless, in its original form, the SpikeProp rule cannot train the neuron to reproduce multiple-spike trains. Several learning algorithms, such as ReSuMe [13], Chronotron [14], Spike Pattern Association Neuron [15], and Precise-Spike-Driven (PSD) [16], have been proposed to address this issue. Among these four rules, without complex error calculation, the PSD rule is simple and efficient from the computational point of view, and yet biologically plausible [16]. In the classification of spatiotemporal patterns, the PSD rule can even outperform the efficient tempotron rule [16]. Moreover, the PSD rule is not limited to the classification, but can also train the neuron to associate the spatiotemporal spike patterns with the desired spike trains.

Recently, a new decoding scheme with spiking neurons has been proposed to describe how downstream neurons with dendritic bistable plateau potentials can perform the decoding of spike sequences [7], [17]. The transition dynamics of this downstream decoding network is demonstrated to be equivalent to that of a finite-state machine (FSM). This decoding scheme has the same computational power as the FSM, which has been widely used in engineering systems, such as natural language processing and speech recognition [32]. The decoding is capable of recognizing an arbitrary number of spike sequences [17]. However, as a part of a whole system, this decoding only describes the behavior of the downstream neurons. How the upstream neurons behave and communicate with the downstream neurons remains unclear.

In this paper, a unified and consistent system with spiking neurons is proposed for sequence recognition. To the best of our knowledge, this is the first attempt to consider a spiking system for sequence recognition with functionalities of sensory coding, learning, and decoding. This paper helps to reveal the systematic neural mechanisms considering all the processes of sensory coding, learning, and downstream decoding. Such a system bridges the gap between these independently studied processes. The system is integrated in a consistent scheme by processing precise-timing spikes, where temporal coding and learning are involved. The sensory coding describes how an external information is converted into neural signals. Through learning, the neurons adapt their synaptic efficacies for processing the input neural signals. The decoding describes how the output neurons extract information from the neural responses. The sequence recognition of the proposed biologically plausible system is realized through the combination of item recognition and sequence order recognition. Identifying the input stimuli is an essential step required before recognizing the sequence order. The proposed system is robust to noisy sensory input and it is also insensitive to changes in the intervals between input stimuli within a certain range. The experiments present SNNs as a paradigm, which can be used for recognizing sequences of incoming stimuli.

The rest of this paper is organized as follows. In Section II, detailed descriptions are presented about the methods used in our integrated system. Section III shows the performances of our system through numerical simulations. Detailed investigation and analysis on different parts of the system are presented. The ability of the applied temporal learning rule is isolated for investigation using the EXCLUSIVE OR (XOR) benchmark task. Then, a practical optical character recognition (OCR) task is applied to investigate the functionality of our system on item recognition. The performance of the spike sequence decoding system is investigated using a synthetic sequence of spikes. Finally, the ability of the whole system on item sequence recognition is demonstrated. Discussions about our system are presented in Section IV. Finally, the conclusion is drawn in Section V.

II. INTEGRATED NETWORK FOR SEQUENCE RECOGNITION

In this section, the whole system for sequence recognition is described, as well as the corresponding schemes used in different subsystems.

A. Rationale of the Whole System

The whole system model contains three functional parts, including sensory encoding, learning, and decoding (Fig. 1). They are the essential parts for a system of spiking neurons to fulfill the sequence recognition task.

In order to utilize the spiking neurons for processing external information, the first step is to get the data into them, where proper encoding methods are required [33], [34].

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

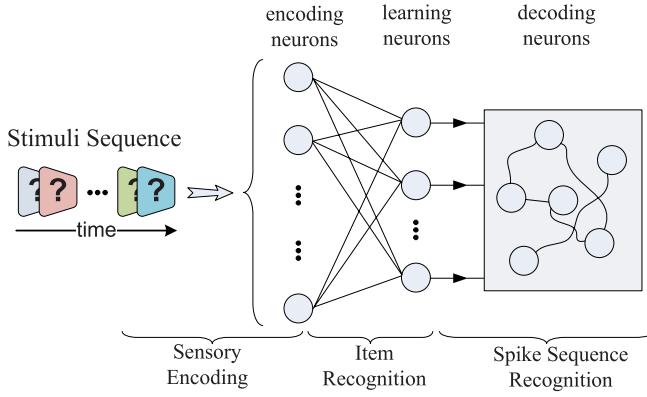YU *et al.*: SNN SYSTEM FOR ROBUST SEQUENCE RECOGNITION

3

Fig. 1. Schematic of the system structure for sequence recognition. The system contains three functional parts, which are used for sensory encoding, item recognition, and spike sequence recognition, respectively. The encoding neurons convert the external stimuli into spatiotemporal spike patterns. The learning neurons would recognize the content of each input item based on the corresponding spatiotemporal spike pattern. The sequence order of the input stimuli would be recognized through the decoding neurons.

The components of the stimuli are connected to the encoding neurons. These encoding neurons are used to generate spatiotemporal spike patterns which represent the external stimuli. For example, each item in the sequence (denoted as "?" in Fig. 1) will be converted into a particular form of spike patterns by the encoding neurons. For our choice, single spikes are used, with each encoding neuron firing only once within the presence of an input item. The scheme of single spikes is simple and efficient, which would potentially facilitate computing speed since fewer spikes are involved in the computation [33], [35].

Before the recognition of the sequence order, another important step is to recognize each input item. We call this recognition process as item recognition. Without successful item recognition, it is impossible to detect the order of the incoming items since each one will be an unknown item to the system. The learning neurons in our system are used to perform the item recognition. These learning neurons are trained to perform the recognition on the encoded spike patterns that sent from the encoding neurons. During the training, the synaptic efficacies of the learning neurons are adapted to memorize the items. After training, the item recognition memory is stored in the synaptic efficacies, and thus the learning neurons can be applied in the system structure. Whenever a new incoming item comes into the system, the learning neurons can properly make a decision based on previously obtained memory.

After successful item recognition, the final step is to recognize the specific sequence of the incoming items. The learning neurons send spikes to the decoding neurons, where each spike represents a decision result for an input item. The target of the decoding neurons is to successfully recognize the spike sequence order of the learning neurons, and we call this recognition process as spike sequence recognition. The memory of the sequence order is stored in the connection structure of the decoding neurons.

Therefore, as is shown in Fig. 1, our system will process the incoming items in three main steps: 1) each item is first converted into spike patterns through the encoding neurons; 2) the encoded pattern will be recognized by the learning neurons, and a decision spike will be sent to the following network; and 3) the desired sequence order will be recognized by the decoding neurons. In order to make the whole system function properly, the three subsystems need to communicate consistently. The functionalities of the three subsystems are encoding, item recognition, and spike sequence recognition. With all these functionalities combined together, the whole system could perform the item sequence recognition. The schemes used in each subsystem are presented as follows.

B. Neural Encoding Method

Neural encoding considers how to generate a set of specific activity patterns that represent the information of external stimuli. The specific activity patterns considered in this paper are in a spatiotemporal form where the precise timing of spikes is used for carrying information. Any encoding methods, that can generate spatiotemporal spike patterns with spikes being sparsely distributed within the time window, could be a proper choice for the encoding in our system. Here, we present a phase encoding method, and use it as our encoding part in the system due to its high spatial and temporal selectivity [36].

An increasing body of evidence shows that action potentials are related to the phases of the intrinsic subthreshold membrane potential oscillations (SMOs) [37]–[39]. These observations support the hypothesis of a phase code [36], [40], [41]. Such a coding method can encode and retain information with high spatial and temporal selectivity [36]. Following the coding methods presented in [36] and [41], we propose a new phase encoding method. In the coding methods of [36] and [41], the artificial steps of alignment and compression are questionable for biological implementation. Differently, we provide a more biologically plausible coding scheme with a clearer picture about how spikes would be generated. This would be helpful for further implementing the coding scheme on the hardware systems. Our encoding mechanism is presented in Fig. 2.

As can be seen from Fig. 2, each encoding unit contains a positive neuron (Pos), a negative neuron (Neg), and an output neuron (E_{out}). The coding units compose the encoding neurons in Fig. 1. Each encoding unit is connected to an input signal x and an SMO. The potentials of the Pos and Neg neurons are the summation of x and SMO. The direction of the summation is determined by the polarities of synapses. Whenever the membrane potential first crosses the threshold (θ_E), the neuron will fire a spike. In order to utilize single spikes, the neuron is allowed to fire only once within the whole oscillation period T. This can be implemented through resetting the neuron's potential to prevent it from firing again within T. Due to the strong connections, the firing of either the Pos neuron or the Neg neuron will immediately cause a spike from the E_{out} neuron. The SMO for the ith encoding unit is described as

$$\text{SMO}_i = M\cos(\omega t + \phi_i) \tag{1}$$

where M is the magnitude of the SMO, $\omega = 2\pi/T$ is the phase angular velocity, and ϕ_i is the initial phase. ϕ_i is defined as

$$\phi_i = \phi_0 + (i-1)\cdot\Delta\phi \tag{2}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

4

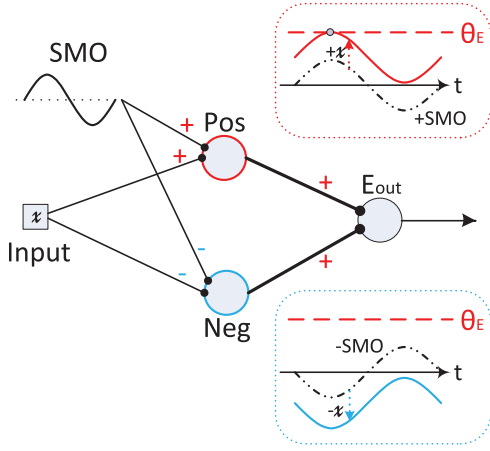IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS

Fig. 2. Phase encoding method with one encoding unit being presented. An encoding unit is composed of a positive neuron (Pos), a negative neuron (Neg), and an output neuron (E_{out}). The encoding unit receives signals from an input (x) and an SMO. The value of x can be negative as well as positive. Rectangle boxes: potential dynamics of the Pos and Neg neurons. The polarities of the synapses, being either positive or negative, are denoted by $+$ and $-$, respectively. Whenever the membrane potential crosses the threshold (θ_E), the neuron (Pos or Neg) will fire a spike immediately. E_{out} is strongly connected to the Pos and Neg neurons, where the firing of either the Pos neuron or the Neg neuron will immediately cause a spike from the E_{out} neuron.

where $\phi_0 = 0$ is the reference phase and $\Delta\phi$ is the phase difference between nearby encoding units. We set $\Delta\phi = 2\pi/N_{en}$ where N_{en} is the number of encoding units.

C. Item Recognition With the PSD Rule

The PSD rule [16] is recently proposed for processing spatiotemporal spike patterns. This rule is not only able to train the neurons to associate spatiotemporal spike patterns with desired spike trains, but also able to train the neurons to perform the classification of spatiotemporal patterns. As the PSD rule is simple and efficient, we use it to train the learning neurons in the proposed system for item recognition.

We choose the leaky integrate-and-fire (LIF) model to be the neuron model of our learning neurons. The dynamics of each neuron evolves according to the following:

$$\tau_m \frac{dV_m}{dt} = -(V_m - E) + (I_{syn} + I_{ns})R_m \qquad (3)$$

where V_m is the membrane potential, $\tau_m = R_m C_m$ is the membrane time constant, $R_m = 1$ MΩ and $C_m = 10$ nF are the membrane resistance and capacitance, respectively, E is the resting potential, I_{ns} and I_{syn} are the background current noise and synaptic current, respectively. When V_m exceeds a constant threshold V_{thr}, the neuron is said to fire, and V_m is reset to V_{reset} for a refractory period $t_{ref} = 3$ ms. We set $E = V_{reset} = 0$ mV and $V_{thr} = E + 18$ mV for clarity.

The synaptic current is in the form of

$$I_{syn}(t) = \sum_i w_i \cdot I^i_{PSC}(t) \qquad (4)$$

where w_i is the synaptic weight of the ith afferent neuron, and I^i_{PSC} is the unweighted postsynaptic current from the

corresponding afferent

$$I^i_{PSC}(t) = \sum_{t^j} K(t - t^j)H(t - t^j) \qquad (5)$$

where t^j is the time of the jth spike from the ith afferent neuron. $H(t)$ is the Heaviside function, and K denotes a normalized kernel as

$$K(t - t^j) = V_0 \cdot \left(\exp\left(\frac{-(t - t^j)}{\tau_s} \right) - \exp\left(\frac{-(t - t^j)}{\tau_f} \right) \right) \qquad (6)$$

where V_0 is a normalization factor such that the maximum value of the kernel is 1, τ_s and τ_f are the slow and fast decay constants, respectively, and their ratio is fixed at $\tau_s/\tau_f = 4$. We choose $\tau_s = 10$ ms in this paper.

The PSD rule is derived from the common Widrow–Hoff rule by applying spike convolution on the input spike train [16]. By choosing a convolving kernel, I^i_{PSC} can be used as the eligibility trace for the weight adaptation. The learning rule is in the form of

$$\frac{dw_i(t)}{dt} = \eta[s_d(t) - s_o(t)]I^i_{PSC}(t) \qquad (7)$$

where η is the learning rate. $s_d(t)$ and $s_o(t)$ are the desired and the actual output spike trains, respectively. The spike train is described as

$$s(t) = \Sigma_f \delta(t - t^f) \qquad (8)$$

where t^f is the firing time of the fth spike, and $\delta(x)$ is the Dirac function: $\delta(x) = 1$ (if $x = 0$) or 0 (otherwise).

By integrating (7), we get

$$\Delta w_i = \eta \int_0^\infty [s_d(t) - s_o(t)]I^i_{PSC}(t)dt$$
$$= \eta \left[\sum_g \sum_f K(t^g_d - t^f_i)H(t^g_d - t^f_i) \right.$$
$$\left. - \sum_h \sum_f K(t^h_o - t^f_i)H(t^h_o - t^f_i) \right]. \qquad (9)$$

This equation describes a trial learning where the weight modification is performed at the end of the pattern presentation. We use (9) to train the learning neurons for item recognition. After training, the learning neurons would successfully recognize the spatiotemporal spike patterns that sent from the encoding neurons.

In order to measure and analyze the similarity between two spike trains, the same distance measurement with a form of the van Rossum metric [42] is used as in [16]. The distance can be written as

$$\text{Dist} = \frac{1}{\tau} \int_0^\infty [f(t) - g(t)]^2 dt \qquad (10)$$

where τ is a free parameter (we set $\tau = 10$ ms), $f(t)$ and $g(t)$ are the filtered signals of the two spike trains that are considered for distance measurement. The filter function is described as in (6).

The spike distance can reflect the dissimilarity between the desired and actual spike trains.
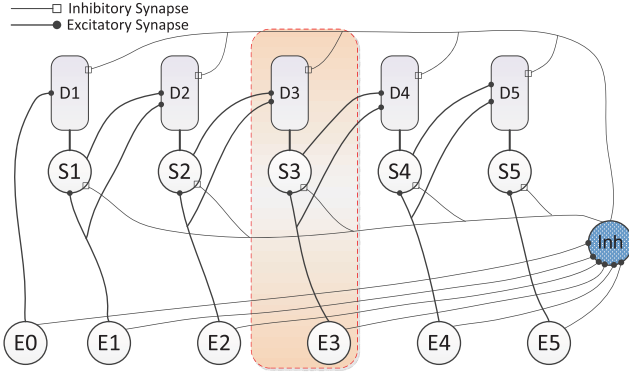
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

YU *et al.*: SNN SYSTEM FOR ROBUST SEQUENCE RECOGNITION

5

Fig. 3. Neural structure for spike sequence recognition. E0–5: excitatory input neurons. S1–5 and D1–5: soma and dendrite, respectively. Inh: global inhibitory neuron. Dashed box: basic building block for scaling the network to sequences of different size.

D. Spike Sequence Decoding Method

In this part, we describe the sequence decoding method used for the decoding neurons in our system. A network of neurons with dendritic bistable plateau potentials can be used to recognize spike sequences [17]. Based on this idea, we build our decoding system as presented in Fig. 3. This decoding network can recognize a specific sequence order of the spike inputs from the excitatory input neurons. The sequence scale of this network could be easily modified through adding or deleting the basic building blocks as in Fig. 3.

In Fig. 3, the dendrites have transient bistable plateau potentials, which produce the UP and DOWN states of the soma. The transitions between the UP and DOWN states are controlled by the feedforward excitation from the input neurons, feedforward inhibition from the global inhibitory neuron, as well as the lateral excitations between the neurons in the network. There is only one global inhibitory neuron that receives input from the excitatory input neurons and sends inhibition to all the dendrite and soma. The inhibitory neuron will send a spike with a short delay after receiving an excitatory input. At the beginning of the dendrites entering the plateau potentials, they are transiently stable for a time and thus resistant to the inhibition during this time. In order to make a soma fire, there are two necessary conditions: 1) the soma is in the UP state and 2) the corresponding excitatory neuron fires. The first condition reflects the correct sequential inputs from the past and the second condition requires that the current input must be the desired one in the sequence. More detailed descriptions about the dynamics of Fig. 3 can be seen in Section III-C.

The dynamics of the membrane potential of the soma is described as

$$\tau_{\text{sm}} \frac{dV_{\text{sm}}}{dt} = -(V_{\text{sm}} - E_r) + g_{\text{ds}}(V_{\text{dr}} - V_{\text{sm}})$$
$$+ I_s + I_A + I_{\text{ns}} \quad (11)$$

where V_{sm} and V_{dr} denote the potential of the soma and the dendrite, respectively; $\tau_{\text{sm}} = 20$ ms is the membrane time constant; $E_r = -70$ mV is the resting membrane potential; $g_{\text{ds}} = 0.35$ is the conductance from the dendrite to the soma; I_s is the synaptic current on the soma; I_A is the A-type potassium current; and I_{ns} is a background current, and is set to zero here.

The A-type potassium current [43], [44] is activated near the resting potential and inactivated at more depolarized potentials. I_A in the soma is given by

$$I_A = -g_A \cdot a_\infty \cdot V_{\text{sm}}^3 \cdot b(t) \cdot (V_{\text{sm}} - E_K) \quad (12)$$

where $g_A = 10$ is the conductance; $E_K = -90$ mV is the reversal potential of the potassium current; a_∞ and $b(t)$ are the activation and inactivation variables, respectively, and they are given by

$$a_\infty = \frac{1}{1 + \exp(-(V_{\text{sm}} + 70)/5)} \quad (13)$$

$$\tau_A \frac{db}{dt} = -b + \frac{1}{1 + \exp((V_{\text{sm}} + 80)/6)} \quad (14)$$

where $\tau_A = 5$ ms is a time constant.

The synaptic current on the soma is given by

$$I_s = -g_{As} \cdot (V_{\text{sm}} - E_E) - g_{Gs} \cdot (V_{\text{sm}} - E_I) \quad (15)$$

where g_{As} and g_{Gs} are the alpha-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid (AMPA) and gamma-amino-butyric-acid (GABA) synaptic conductance, respectively. The AMPA and GABA synaptic conductance mediate synaptic excitation and inhibition, respectively. $E_E = 0$ mV and $E_I = -75$ mV are the reversal potential of excitatory and inhibitory synapses, respectively.

The dynamics of the membrane potential of the dendrite is described as

$$\tau_{\text{dr}} \frac{dV_{\text{dr}}}{dt} = -(V_{\text{dr}} - E_r) + g_{\text{sd}} \cdot (V_{\text{sm}} - V_{\text{dr}}) + I_{\text{dr}} \quad (16)$$

where $\tau_{\text{dr}} = 10$ ms is the time constant of the dendrite; $g_{\text{sd}} = 0.05$ is the conductance from the soma to the dendrite; I_{dr} is the synaptic current on the dendrite, and is given by

$$I_{\text{dr}} = -g_{Ad} \cdot (V_{\text{dr}} - E_E) - g_{Gd} \cdot (V_{\text{dr}} - E_I)$$
$$- \frac{g_{Nd} \cdot V_{\text{dr}}}{1 + \exp(-(V_{\text{dr}} + 30)/5)} \quad (17)$$

where g_{Ad} and g_{Gd} are the AMPA and GABA synaptic conductance, respectively; g_{Nd} is the N-methyl-D-aspartate (NMDA) synaptic conductance that is responsible for the transient bistable plateau potential.

An incoming spike arrives at a synapse with strength G will cause changes on synaptic conductance g: $g \rightarrow g + G$. On the dendrite, a spike to an excitatory synapse will cause $g_{Ad} \rightarrow g_{Ad} + G$ and $g_{Nd} \rightarrow g_{Nd} + 5G$. Without incoming spikes, all the synaptic conductance will decay exponentially. The decay time constants for both the AMPA and GABA conductance are 5 ms. For the NMDA conductance, the decay time constant is 150 ms. g_{Nd} is not allowed to exceed 10 due to a saturation.

The inhibitory neuron could be modeled as a single compartment quadratic LIF neuron [7], [17] such that it can respond with a short latency to an excitatory spike input. Different from this complex model, we simplify the model by setting the inhibitory neuron to spike once with a delay of 2 ms at each input spike. This choice makes the model simple and the functionality of the global feedforward inhibition remains the same.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

6

IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS

TABLE I
XOR PROBLEM DESCRIPTION

XOR Input	Encoded Spike Input (ms)	Desired Response
(0, 0)	(0, 0)	Fire
(0, 1)	(0, 10)	Silence
(1, 0)	(10, 0)	Silence
(1, 1)	(10, 10)	Fire

III. NUMERICAL SIMULATIONS

In this section, several experiments are presented to demonstrate the characteristics of our model. Through simulations, we investigate the abilities of our system mainly for item recognition and sequence recognition. A correct recognition on the input items is an essential step for further recognizing the sequence.

First, in Section III-A, the XOR problem is used to preliminarily analyze the classification ability of the temporal learning rule on spike patterns. Through the XOR task, we want to isolate the PSD rule for testing before applying it in our system. In Section III-B, we present the ability of our system for item recognition. A set of optical characters with images of digits 0–9 are used. Section III-C shows the performance of our spike sequence decoding subsystem where the downstream neurons could recognize a specific spike sequence. Finally, in Section III-D, the functionalities of both the item recognition and the spike sequence recognition are combined together for the item sequence recognition, which shows the performance of the whole system.

A. Learning Performance Analysis of the PSD Rule

In this part, we isolate the PSD rule from the whole system for testing with the XOR task. The XOR problem is a linearly nonseparable task, and it is a benchmark widely used for investigating the classification ability of SNNs recently [2], [45]–[47]. Thus, we also use the XOR problem to investigate the performance of the PSD rule first.

Similar to the setup in [2], we directly map the XOR inputs to spike times, with the symbol 0/1 being associated with a spike at 0/10 ms. Table I shows the input spike patterns and desired response for the XOR task. Noteworthily, the two patterns (0–0 ms and 10–10 ms) are actually identical without considering the delay of 10 ms. In order to have causal response to both inputs, a reasonable choice for the output spike should be later than 10 ms. Therefore, we use the PSD to train the neuron to fire at 12 and 22 ms for pattern (0, 0) and (1, 1), respectively. The neuron is also trained to be silent for patterns of (0, 1) and (1, 0). The PSD learning parameters are set as $\eta = 0.01$ and $\tau_s = 10$ ms.

Fig. 4 shows the PSD rule can train the neuron to solve the XOR task successfully. When patterns of (0, 1) and (1, 0) present, the neuron keeps silent without firing. For patterns of (0, 0) and (1, 1), the neuron will fire a spike as desired. However, it would be problematic if we want the neuron to fire a same desired spike train for patterns in the same group. This is due to the low dimensionality of the XOR task. Both the temporal and spatial dimensions are very limited (2 here).
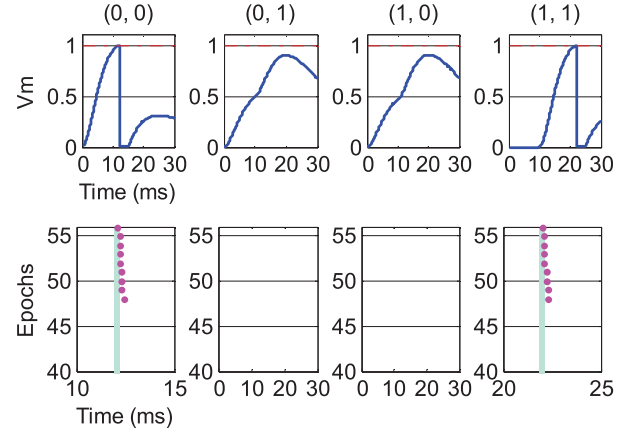

Fig. 4. Performance of the PSD rule on the XOR task with direct spike mapping. Top panel: membrane potentials of the neuron after training. Bottom panel: spike time of the neuron along the training. Shaded bars: desired spike time.

In order to train the neuron to fire multiple precise-timing spikes, the dimensionality of the problem should be enhanced.

Similar to the setup in [13] and [47], we randomly generate two homogeneous poisson spike trains with a firing rate of 50 Hz in a time window of 200 ms. These two spike trains represent XOR inputs of 0 and 1, respectively, and they are used to form the four patterns of the XOR problem: (0, 0), (0, 1), (1, 0), and (1, 1) as demonstrated in Fig. 5(a). We also employ the concept of reservoir computing with a network of liquid state machine (LSM) like in [13], [47], and [48]. The LSM uses spiking neurons connected by dynamic synapses to project the inputs to a higher dimensional feature space, which can facilitate the learning task. The network used in this experiment consists of two input neurons, a noise-free reservoir with 500 LIF neurons and one readout neuron. The setup details about the LSM are attached in the Appendix.

The target spike train could be randomly generated over the time window. For simplicity, we specify the target spike train for each category. For inputs of (0, 0) and (1, 1), the output neuron is trained to spike at [110, 190] ms, while for (0, 1) and (1, 0), it is trained to fire another target train of [70, 150] ms. Other choices of the target spikes could also be acceptable. The initial synaptic weights of the output neuron are randomly drawn from a normal distribution with a mean value of 0.5 nA and a standard deviation of 0.2 nA. This initial condition of synaptic weights is also used for other experiments in this paper. These synaptic weights are adjusted by the PSD rule with a set of learning parameters $\eta = 0.01$ and $\tau_s = 10$ ms. The results are averaged over 100 runs.

Fig. 5(b) shows the results of a typical run, with the actual output spikes for each of the four input patterns during the learning. At the beginning, both the firing rates and the precise timings of the output spike trains are different from those of the target spike trains. After tens of learning epochs, the readout neuron can gradually learn to fire the target spike trains according to different input patterns. After hundreds of learning epochs, the readout neuron stabilizes at the target spike trains. This phenomenon can be also seen from the spike distance between the actual and the target
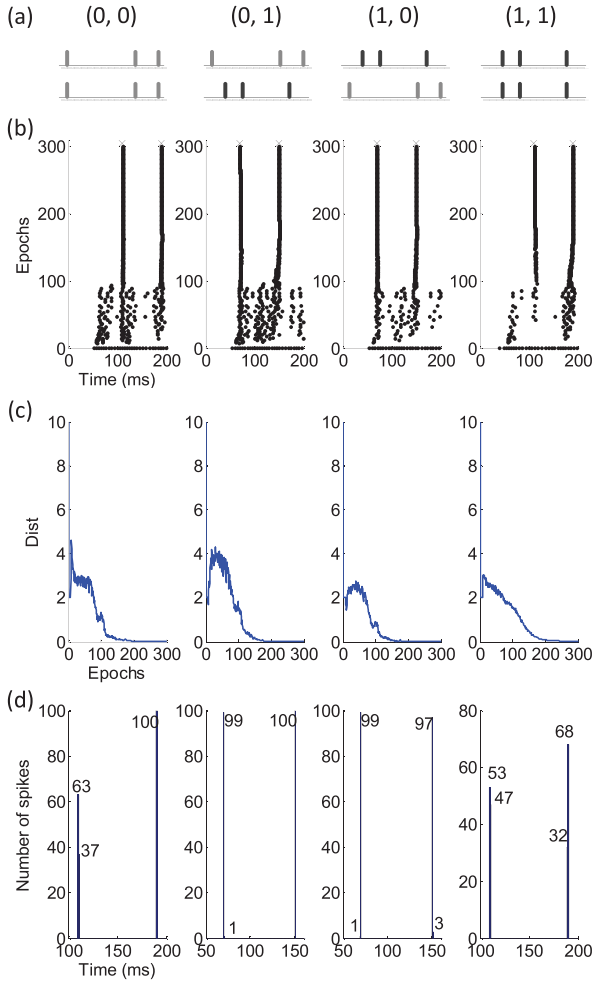
Fig. 5. Performance of the PSD rule on the XOR task with the LSM. (a) General illustration of the four inputs of the XOR task. The values of 0 and 1 are represented by different spike trains. (b) Output spike signals for each of the four input patterns during learning in a typical run. × denotes the desired spike time. (c) and (d) Results of the output neuron after 100 runs. (c) Average spike distance between the desired and the actual output spike trains. The average spike distance for each input pattern is presented. (d) Spike histogram showing the distribution of the actual output spikes.
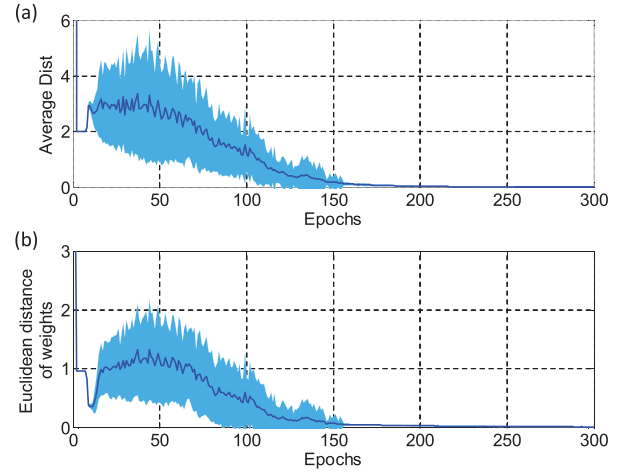

Fig. 6. Convergent performance. (a) Average spike distance over all the four input patterns. (b) Euclidean distance between the weights before and after each learning epoch. All the results are averaged over 100 runs.

spike trains [Fig. 5(c)]. A larger spike distance occurs at the beginning due to the initial conditions, followed by a gradually decreasing spike distance along the learning, and it finally converges to zero. Fig. 5(d) shows the distribution of the actual output spikes corresponding to the four input patterns over 100 runs. From these histograms, we can see our approach with the PSD rule obtains better performance than that in [47]. First, there are no undesired extra spikes or missing desired spikes in our approach. In the 100 runs of experiments, the trained neuron fires exactly 100 spikes around each desired time. Second, the actual output spikes are precisely and reliably close to the desired time. The maximum error of spike time is ∼1 ms. Thus, the learning success rate of our approach is higher than that in [47].

Fig. 6 shows the convergent performance during the learning process. The average spike distance over all four input patterns is presented as well as the Euclidean distance between the weights before and after each learning epoch. As can be seen from Fig. 6, irregular distances occur at the first several learning epochs because of the random initial conditions. After that, the distances gradually decrease and converge to zero. The zero spike distance corresponds to the readout neuron firing exactly the target spike train, and the zero weight distance implies that there are no more changes occurring on the weights. These two distance graphs also show the ability of the PSD rule to modify the weights in order to produce the desired output spikes. Either of these two types of distances can be used as a stopping criterion for the learning process.

This experiment with the XOR problem demonstrates the ability of the PSD rule for classifying spatiotemporal patterns. The PSD rule can perform the task as desired, as long as enough dimensionality is provided by the input patterns. Considering the real-world stimuli such as images, normally, the dimensionality of the input space is not an issue. In the following, we apply the PSD rule in our system, and investigate the performance of our system for item recognition and sequence order recognition.

B. Item Recognition

In this section, the functionality of our system for the item recognition is considered. The encoding neurons and learning neurons in Fig. 1 are involved for the item recognition. A set of optical characters of 0–9 are used. Each image has a size of 20×20 black/white pixels, and each would be corrupted by a reversal noise where each pixel is randomly reversed with a probability denoted as the noise level. Some clean and noisy samples are demonstrated in Fig. 7(a) and (b).

The phase encoding method shown in Fig. 2 is used to convert the images into spatiotemporal spike patterns. Each pixel acts as an input x to each encoding unit. x is normalized into the range of $[-0.5\theta_E, 0.5\theta_E]$ where $\theta_E = 1$ is the threshold of the encoding neurons. We set $M = 0.5\theta_E$, thus the encoded spikes will only occur at peaks of the SMOs. The number of encoding units N_{en} is equal to the number of pixels which is 400 here. We set the oscillation period T of the SMOs to be 200 ms. The chosen scale of hundreds of milliseconds

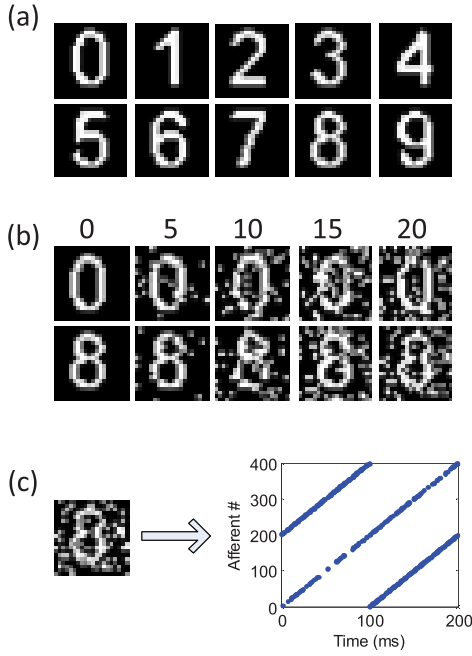(a)

(b)   0   5   10   15   20

(c)

Fig. 7.   OCR samples. (a) Template images. (b) Some image samples with different levels of reversal noise. (c) Phase encoding result of a given image sample. Each dot denotes a spike.

Fig. 8.   Performance of the PSD rule with different numbers of target spikes under the case of jitter noise.

matches with the biological experimental results [12], [49]. Fig. 7(c) demonstrates an encoding result with our phase coding method. The output spikes are sparsely distributed over the time window. This sparseness is compatible with the biology [50]. In addition, the sparseness could also benefit the PSD rule for constructing causal connections. However, the sparse spikes will not be obtained if we use the rank-order coding [51] for the given task. All the white or black pixels will result in highly synchronized spikes. With the advantage of an additional phase coding dimension, the encoded spikes are sparsely distributed by our coding method.

We select ten learning neurons trained by the PSD rule, with each learning neuron corresponding to one category. The learning parameters in the PSD rule are set to be $\eta = 0.06$ and $\tau_s = 10$ ms. All the learning neurons are trained to fire a target spike train with the corresponding category. The target spike train is set to be evenly distributed over the time window T (200 ms here) with a specified number of spikes n. The firing time of the ith target spike is expressed as: $t_i = i/(n+1) \cdot T$, $i = 1, 2, \ldots, n$. We choose $n = 4$ by default, otherwise will be stated. In the item recognition, the relative confidence criterion [16] is used for the PSD rule, where the incoming pattern is represented by the neuron that fires the most closest spike train to its target spike train.

In this section, several noisy scenarios are considered to evaluate the robustness of our system for item recognition: 1) spike jitter noise where a Gaussian jitter with a standard deviation (denoted as the jitter strength) is added into each encoded spike; 2) reversal noise as shown in Fig. 7(b), where each pixel is randomly reversed with a probability denoted as the noise level; and 3) combined noise, where both the jitter and the reversal noises are involved.
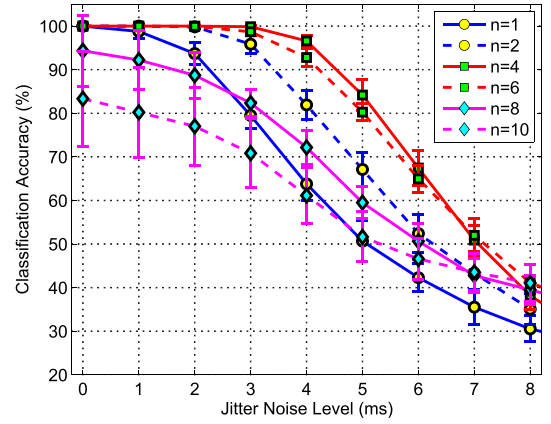
1) Spike Jitter Noise: In this scenario, the image templates are first encoded into spatiotemporal spike patterns. After that, jitter noises are added to generate noisy patterns. The learning neurons are trained for 100 epochs with a jitter strength of 2 ms. In each learning epoch, a training set of 100 patterns, with 10 for each category, is generated. After training, a jitter range of 0–8 ms is used to investigate the generalization ability. The number of the testing patterns for each jitter strength is set to 200. The PSD rule is applied with different numbers of target spikes ($n = 1, 2, 4, 6, 8, 10$). All the results are averaged over 100 runs.

Fig. 8 shows the effects of the number of the target spikes on the learning performance of the PSD rule. As can be seen from Fig. 8, when n is low (e.g., 1, 2), the recognition performance is also relatively low. An increasing number of the target spikes can improve the recognition performance significantly (see $n = 1, 2 \rightarrow n = 4, 6$). However, a further increase in the number of target spikes ($n = 6 \rightarrow n = 8, 10$) would reduce the recognition performance. The reasons for this phenomenon are due to the local temporal features associated with each target spike. For small number of target spikes, the neurons make decision based on a relatively less number of temporal features. This small number of features only covers a part range of the whole time window, which inevitably leads to a lower performance compared with a more number of spikes. However, when the number of spikes continues increasing, an interference of local learning processes [13] occurs and increases the difficulty of the learning. Thus, a higher number of spikes normally cannot lead to a better performance due to the interference. Noteworthily, compared with the idealized case of random patterns that normally considered, the effect of the local interference is more obvious for the learning of the practical patterns since these patterns share more common features than random patterns.

Fig. 9 shows the performance of different learning rules for the same classification task. We use a similar approach for the perceptron rule as in [47] and [48], where the spatiotemporal spike patterns are transformed into continuous states by a low-pass filter. The target spike trains are separated into bins of size t_{smp}, with $t_{smp} = 2$ ms being the sampling time. The target vectors for the perceptron contain values of 0 and 1,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

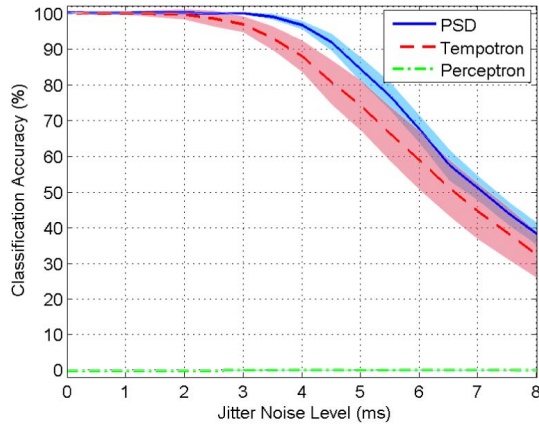YU *et al.*: SNN SYSTEM FOR ROBUST SEQUENCE RECOGNITION

9

Fig. 9. Robustness of different rules against the jitter noise. The PSD rule uses $n = 4$ target spikes. The PSD rule outperforms the other two rules in the considered task.
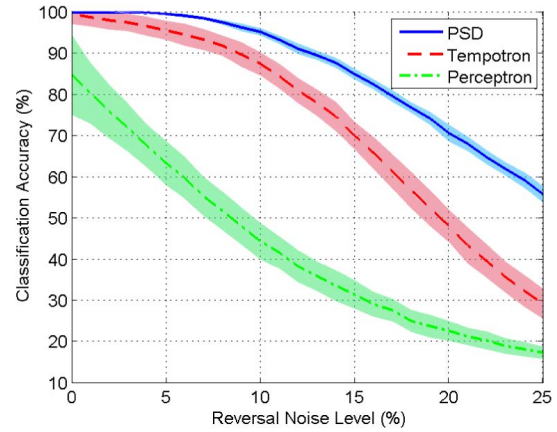

Fig. 10. Robustness of different rules against the reversal noise. The PSD rule uses $n = 4$ target spikes. The PSD rule outperforms the other two rules even when the noise level is high.

with 1 (or 0) corresponding to those bins that contain (or not contain) a target spike in the bin. The input vectors for the perceptron are sampled from the continuous states with t_{smp}. The input pattern will be classified by the winning perceptron that has the closest output vector to the target vector.

As can be seen from Fig. 9, the PSD rule outperforms both the tempotron rule and the perceptron rule. The inferior performance of the perceptron rule can be explained. The complexity of the classification for the perceptron rule depends on the dimensionality of the feature space and the number of input vectors for decisions. A value of $t_{smp} = 2$ ms will generate 100 input vectors for each input pattern. These 100 points in 400-D space are to be classified into 1 or 0. Learning a single spike pattern can increase the difficulty for the perceptron rule, let alone considering a large number of input patterns from different categories. Without separating the time window into bins, the spiking neurons by their nature are more powerful than the traditional neurons such as the perceptron. Both the PSD rule and the tempotron rule are better than the perceptron rule. The PSD rule is better than the tempotron rule since the PSD rule makes a decision based on a combination of several local temporal features over the entire time window, but the tempotron rule only makes a decision by firing one spike or not based on one local temporal feature.

2) Reversal Noise: In this scenario, the reversal noise is used for generating noisy patterns as shown in Fig. 7(b). The learning neurons are trained for 100 epochs with a reversal noise level randomly drawn from the range of 0%–10% in each learning epoch. Meanwhile, a training set of 100 noisy patterns, with 10 for each category, is generated for each learning epoch. After training, another number of 100 noisy patterns are generated and used for each reversal noise level to test the generalization ability. The noise range for testing covers 0%–25% as shown in Fig. 10.

As can be seen from Fig. 10, the performances of all the three rules decrease with the increasing noise level. The performance of the PSD rule again outperforms the other two rules as in the previous scenario. Spiking neurons trained by the PSD rule can obtain a high classification accuracy (∼85%) even when the reversal noise reaches a high level (15%). The performance of the perceptron rule in this scenario is
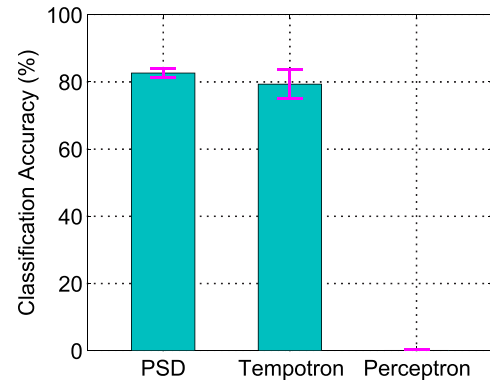

Fig. 11. Robustness of different rules against the combination of the jitter and reversal noises. A 10% reversal noise and a 4-ms jitter noise are used for testing.

much better than that in the previous scenario. This is because of the type of the noise. The performance of the perceptron rule is quite susceptible to the changes in state vectors. Every spike of the input spatiotemporal spike patterns in the case of spike jitter noise suffers a change, while in the case of reversal noise, a change only occurs with a probability of the reversal noise level. That is to say, the elements in a filtered state vector have a less chance to change under the reversal noise than that under the jitter noise. Thus, the performance of the perceptron rule [47] under the reversal noise is better than that under the jitter noise.

3) Combined Noise: In this scenario, the jitter noise and the reversal noise are combined together to evaluate the robustness of our item recognition. Again, the learning neurons are trained for 100 epochs. In each epoch, a random reversal noise level chosen from 0%–10% is used, as well as a jitter noise level of 2 ms. After training, a reversal noise level of 10% and a jitter noise level of 4 ms is used to investigate the generalization ability.

Fig. 11 shows that the combined noise has a stronger impact on the performance than each single noise alone on the performance. This is expected since the effects of the two noises are combined. The perceptron rule still has a poor performance due to the jitter noise. The PSD rule still performs the best with a high average accuracy and a low deviation.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

10   IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS
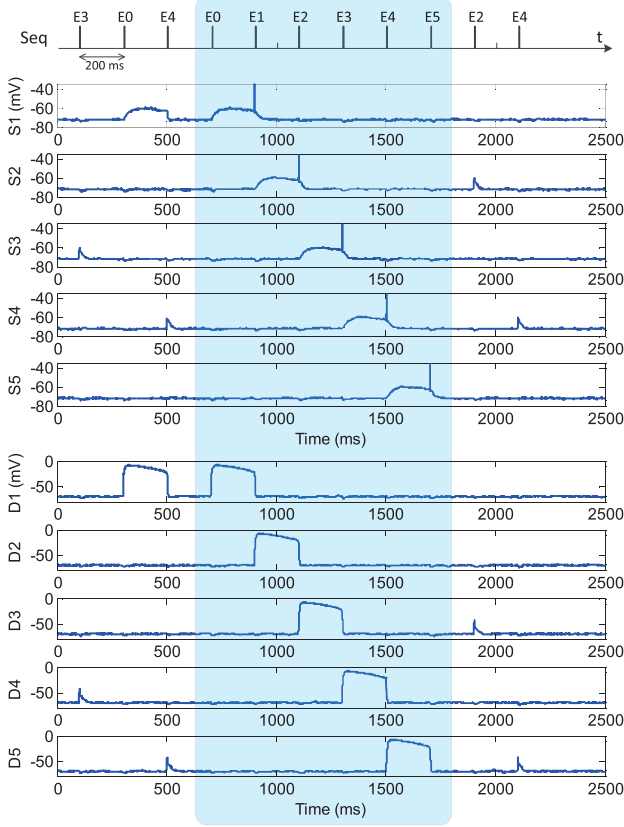

Fig. 12. Reliable response of the spike sequence decoding system. A synthetic spike sequence is used as the input (denoted as Seq). The target sequence pattern of $E0$, $E1, \ldots, E5$ is highlighted by the shaded area. The potentials of the somas ($S1$–5) and the dendrites ($D1$–5) are shown. The interval spike time in the input sequence is 200 ms. The neurons can be successfully activated to fire when the target sequence presents.
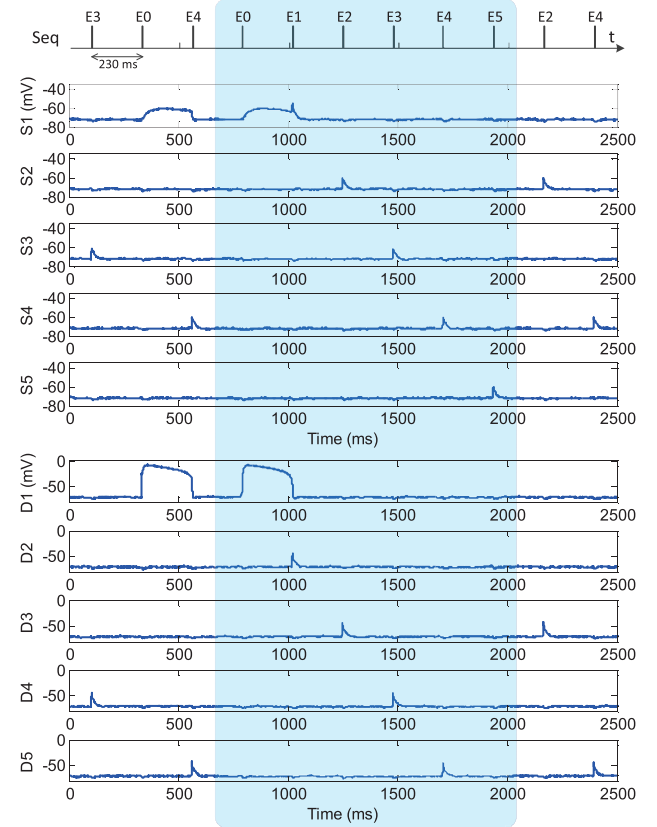

Fig. 13. Unreliable response of the spike sequence decoding system. The interval spike time in the input sequence is 230 ms. When the interval time is over a certain range (225 ms for this experimental setup), the neurons cannot be activated to fire even when the target sequence presents. This is because the potential of the soma cannot sustain in the UP state for such a long interval.

The results in this section demonstrate our item recognition with the PSD rule is robust to different noisy sensory inputs. A reliable recognition on the incoming items is essential for further sequence recognition.

C. Spike Sequence Decoding

In this section, we investigate the performance of our decoding system for spike sequence recognition. The structure of this decoding system is presented in Fig. 3. This decoding structure can recognize a specific sequence of $E0$, $E1, \ldots, E5$. The size of the sequence can be easily extended to different scales through modification of the decoding network structure. We denote the synaptic connections as: 1) $E0 \rightarrow D1$ (G_0); 2) $E1$–$5 \rightarrow S1$–5 (G_1); 3) $E1$–$5 \rightarrow D2$–5 (G_2); 4) $S1$–$5 \rightarrow D2$–5 (G_3); 5) Inh $\rightarrow D1$–5 (G_4); and 6) Inh $\rightarrow S1$–5 (G_5). We set $G_0 = 5$, $G_1 = 2.5$, $G_2 = G_3 = 3$, $G_4 = 5$, and $G_5 = 6$. We generate a spike input feeding into our decoding system, with Fig. 12 showing a 200-ms interval between nearby spikes and 230 ms for Fig. 13.

As can be seen from Fig. 12, the decoding system successfully recognizes the sequence through a firing from $S5$. A strong, excitatory input to the dendrite can make its potential go to a plateau potential that is transiently stable for a time. The plateau potential of the dendrite then drives the potential of the soma to a high-depolarized state. Without the plateau

potential of the dendrite, the potential of the soma stays near the resting potential. We refer the high-depolarized state of the soma as the UP state, and the state near the resting potential as the DOWN state. Two conditions are required to make a soma fire: 1) the potential of the soma sustains in the UP state and 2) when an excitatory spike input comes to this soma.

Under the experimental setup of our decoding system, the UP state of the soma can sustain for a period $\sim$225 ms, during which the soma can reliably fire a desired spike when corresponding excitatory neuron fires. We refer this period as the reliable period. When the time interval between spikes is shorter than the reliable period, the decoding system can perform the recognition well (Fig. 12). When the time interval between input spikes is longer than the reliable period, the UP state of a soma no longer sustains at a reliably high state. This leads to that a corresponding excitatory input spike no longer reliably drives a spike on the soma (Fig. 13).

The experimental results indicate that our spike sequence decoding system is invariant to changes in the intervals between input spikes within a certain range (0–225 ms here).

D. Sequence Recognition System

In this section, the performance of the proposed whole system is investigated. The sensory encoding, temporal learning and spike sequence decoding are consistently combined together for sequence recognition. We perform the experiment with the previous digits used in Section III-B.
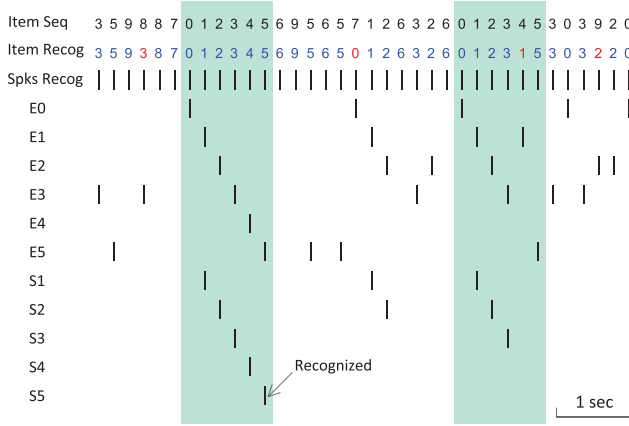
Fig. 14. Performance of the combined sequence recognition system. An image sequence input is fed into the sequence recognition system. Each image suffers a reversal noise of 15%. The target of this system is to detect and recognize a specified target sequence of 012345 (the shaded areas). Item Seq denotes the input sequence of the images. Item Recog is the output results of the learning neurons, with the blue/red color representing correct/incorrect recognition. Each output of the learning neurons results in a spike in the corresponding excitatory input neurons of the spike decoding network (Spks E). $S1–5$ denotes the spike output of neurons in the sequence decoding network.
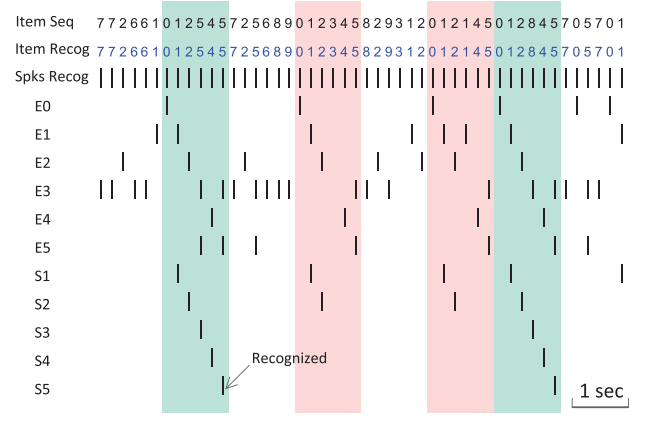

Fig. 15. Performance on a target sequence with one semiblind item. The input sequence is considered in a noise-free condition. The target of this system is to detect and recognize a specified target sequence of 012?45 where "?" is from a specific range of 5–9 (shown in the shaded light-cyan areas). Shaded light-pink areas: some interference sequence patterns where "?" is not chosen from the allowed range.

These optical digits are used to form a sequence pattern, with each digit image in the sequence being corrupted by a reversal noise level of 15%. We can specify a target sequence through building connections between the output neurons of the item recognition network and the excitatory input neurons of the spike sequence decoding network. For simplicity, we specify a target sequence order of digits as: 012345. Thus, the learning neurons corresponding to the categories in this target sequence are connected to the excitatory input neurons in the sequence decoding network one by one. Each digit image is presented for 200 ms. In addition, the interval between two successive images is not allowed to exceed 25 ms, guaranteeing a reliable performance of the spike decoding system.

We construct a sequence pattern of six segments, with six images for each segment. Every image in this sequence is randomly chosen from the ten categories. Then, the target sequence of 012345 is embedded into this sequence, with a probability of 1/3 replacing each initial segment in the sequence. After this, we feed the whole sequence to our system. The target of our system is to detect and recognize the target sequence embedded in the whole sequence.

Fig. 14 shows the performance of our system for sequence recognition. An important step before recognizing the sequence order is to correctly recognize each input item. Only after knowing what is what, a recognition on the sequence order can be applied. The detected target sequence is represented by the firing of $S5$. As can be seen from Fig. 14, the first target sequence is successfully recognize through the sequential firing of $S1–5$, while the second target sequence is not correctly recognized due to a failure recognition on image 4.

In addition, we conduct another experiment, where one item in the target sequence is semi-blind. This semi-blind item is conditioned to a specific range. We specify a target sequence of 012?45, where "?" is restricted to the range of 5–9. Other

digits of "?" being out of this specific range lead to nontarget sequences. For the sake of simplicity, this experiment is conducted in a noise-free condition. We randomly construct an input sequence with 48 items, and then embed the target sequences, as well as some interference sequences, into the input sequence. In order to detect and recognize the semiblind target sequences, we reconstruct the connections between the output neurons of the item recognition network and the excitatory input neurons of the spike sequence decoding network, with all learning neurons for digits 5–9 connecting to $E3$ in Fig. 3. Other connections are not changed. As can be seen from Fig. 15, the semiblind target sequences are successfully recognized, and those interference sequences are also successfully declined. Our system successfully recognizes the target sequence of 012?45 with "?" only belonging to 5–9.

These experiments show that our system with spiking neurons can perform the sequence recognition well, even under some noisy conditions. Item recognition is an essential step for a successful recognition of the target sequence. The step before recognizing the sequence order is to recognize what are the items in the input sequence. A failure recognition of the item in the target sequence would directly affect the further recognition on the sequence order.

IV. DISCUSSION

In this paper, a biologically plausible system with spiking neurons is presented for sequence recognition. Our SNN system contains three functional parts, including sensory encoding, temporal learning, and spike sequence decoding. The whole system is consistently combined in a temporal framework, where the precise timing of spikes is considered for information processing and cognitive computing. The fact of utilizing precise-timing spikes in biological neural systems is widely observed in recent years [20], [22]–[24], [52]. Thus, the whole system is more biologically plausible than the conventional approaches [10], [11] for sequence recognition. Through the encoding mechanism, the external stimuli are con-

verted into spike representations with a spatiotemporal form. With a temporal learning rule, the learning neurons adjust their synaptic weights to learn the encoded spatiotemporal patterns. The learning neurons emit spikes, that containing recognition results of the input items, to the sequence decoding network, where the sequence order of the input items is finally detected and recognized.

A. Temporal Learning Rules and Spiking Neurons

The PSD rule [16], proposed in the concept of processing and memorizing spatiotemporal spike patterns, is applied in our system for item recognition. In the PSD rule, the synaptic adaptation is driven by the precise timing of the actual and the target spikes. Without a complex error calculation, the PSD rule is simple and beneficial for computation [16]. According to the classification tasks considered in this paper, the PSD rule outperforms both the tempotron rule [3], [33] and the perceptron rule [47], [48].

The computational power of the spiking neurons over the traditional neurons (perceptrons) is reflected by the better performance of both the PSD rule and the tempotron rule than the perceptron rule (Figs. 9 and 10). This is because that the spiking neurons, by their nature, are designed for processing in a time domain with a complex evolving dynamics on the membrane potential. A major difference between the perceptrons and the spiking neurons is this dynamic membrane potential. The perceptrons calculate current states in a static manner that only based on the current inputs, while the spiking neurons evolve current states in a dynamic manner that not only based on the current inputs but also the past states. In addition, due to the ability of the spiking neurons to operate online, it can benefit the computation of a sequential procession with time elapsing.

Between the two temporal learning rules for spiking neurons, the performance of the PSD rule is better than the tempotron rule. The decisions made by the neurons under the PSD rule are based on a combination of several local temporal features over the whole time window. By contrast, the tempotron rule trains a neuron to make a decision only based on one local temporal feature if the neuron is supposed to fire a spike. A decision based on several local temporal features would result in a better performance than that based on only one local temporal feature. In addition, the PSD rule is not limited to a classification task, but it can also train a neuron to associate spatiotemporal patterns with the specified desired spike trains.

B. Spike Sequence Decoding Network

Our spike sequence decoding network is biologically realistic that can behave like FSM to recognize spike sequences [7], [17]. The functionality of this network is achieved through transitions between the UP and DOWN states of neurons. Transitions between bistable membrane potentials are widely observed through various experiments in cortical pyramidal neurons *in vivo* [53], [54]. The transitions between the states are controlled by feedforward excitation, lateral excitation, and feedforward inhibition. The neurons
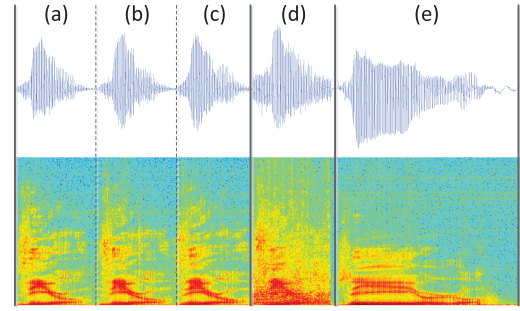

Fig. 16. Voice samples of zero. (a)–(c) Samples spoken by a person in clean conditions with a similar manner for each recording. (d) Sample under a 5-dB noise. (e) Warped sample spoken in a different manner. Top panel: sound waves. Bottom panel: corresponding spectrograms.

enter the UP state if their dendrites have a plateau potential. The neurons will return to the DOWN state from the UP state when enough long time elapses without excitatory input spikes. In addition, the recognition is robust to time warping of the sequence. The recognition is intact as long as the interval between input spikes lies in a reliable range, 0–225 ms, which is quite broad (Figs. 12 and 13). Invariance to time warping is beneficial for tasks like speech recognition [6], [8].

C. Potential Applications in Authentication

Our system provides a general structure for sequence recognition. With proper encoding mechanisms, this system could also be applied to acoustic, tactile, and olfactory signals in addition to visual signals. The encoding mechanisms are used to get the external stimuli into spiking neurons properly, where spikes should be sparsely distributed. The processes of the item recognition and the sequence order recognition in our system could be used for user authentication to access approval. It can provide a double-phase checking scheme for gaining access. Only if both the items and also their orders are correct, the person would be allowed to access.

We preliminarily applied these concepts to the speech task with our previously proposed encoding scheme [9] for sounds. The spoken digits from 0 to 9 were considered. It is still a very challenging task for spiking neurons to process audio signals due to variations of speed, pitch, tone, and volume. Our system could be successful in the case where words are spoken in a similar manner such as samples shown in Fig. 16(a)–(c), but it would be failed if the voice is changed significantly as shown in Fig. 16(d) and (e). Further study is required for speech recognition with spiking neurons, and further results would be presented elsewhere.

V. CONCLUSION

In this paper, a biologically plausible network is proposed for sequence recognition. This is the first attempt to solve the sequence recognition with the network of spiking neurons by considering both the upstream and the downstream neurons together. The system is consistently integrated with functionalities of sensory encoding, learning, and decoding. The system operates in a temporal framework, where the precise timing of spikes is considered for information processing and cognitive

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

YU *et al.*: SNN SYSTEM FOR ROBUST SEQUENCE RECOGNITION
13

computing. The recognition performance of the system is robust to different noisy sensory inputs, and it is also invariant to changes in the intervals between input stimuli within a certain range. The proposed system structure would also be beneficial for developments in both hardware and software.

APPENDIX

A. Simulation Information

All experiments are conducted in the platform of MATLAB, with a simulation step size of 0.1 ms. In order to make our model biologically plausible, most parameters for the neuron models are chosen based on biological inspirations to some extent [3], [12], [17], [20], [21], [49].

B. Details to the Reservoir Network

The details of the LSM used in the XOR benchmark were listed below. The size of the reservoir was $5 \times 10 \times 10$ LIF neurons, with 80% being excitatory and 20% being inhibitory. Parameters of connections from the inputs to the reservoir were as follows: connection probability from the input neuron to any reservoir neuron $p = 0.5$, all connections were excitatory, strength of synaptic connections uniformly drawn from $(0, 10 \cdot 10^{-9})$, synaptic delays uniformly drawn from $(0, 6)$ ms. Parameters within the reservoir were as follows: excitatory synaptic weights and inhibitory synaptic weights were uniformly drawn from $(0, 5 \cdot 10^{-9})$ and $(0, 15 \cdot 10^{-9})$ respectively, and synaptic delays were uniformly drawn from $(0, 6)$ ms. The probability of creating a connection between two neurons is given by $p = c \cdot \exp(-D(a, b)/\lambda)$, where c and λ are positive constants and $D(a, b)$ is the Euclidean distance between neurons a and b. Here, $\lambda = 2.5$, and depending on whether neurons are inhibitory (I) or excitatory (E), c was set at 0.3 (EE), 0.2 (EI), 0.4 (IE), or 0.1 (II).

All reservoir neurons are connected to the output neuron, and only the synaptic weights between the reservoir and the output neuron are adjusted using the PSD rule.

ACKNOWLEDGMENT

REFERENCES

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

[2] S. M. Bohte, J. N. Kok, and H. La Poutré, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, nos. 1–4, pp. 17–37, 2002.

[3] R. Gütig and H. Sompolinsky, "The tempotron: A neuron that learns spike timing–based decisions," *Nature Neurosci.*, vol. 9, no. 3, pp. 420–428, 2006.

[4] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of Neural Science*, vol. 4. New York, NY, USA: McGraw-Hill, 2000.

[5] R. L. Buckner and D. C. Carroll, "Self-projection and the brain," *Trends Cognit. Sci.*, vol. 11, no. 2, pp. 49–57, 2007.

[6] J. J. Hopfield and C. D. Brody, "What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 3, pp. 1282–1287, 2001.

[7] D. Z. Jin, "Spiking neural network for recognizing spatiotemporal sequences of spikes," *Phys. Rev. E*, vol. 69, no. 2, p. 021905, 2004.

[8] R. Gütig and H. Sompolinsky, "Time-warp–invariant neuronal processing," *PLoS Biol.*, vol. 7, no. 7, p. e1000141, 2009.

[9] J. Dennis, Q. Yu, H. Tang, H. D. Tran, and H. Li, "Temporal coding of local spectrogram features for robust sound recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2013, pp. 803–807.

[10] J. A. Starzyk and H. He, "Spatio–temporal memories for machine learning: A long-term memory organization," *IEEE Trans. Neural Netw.*, vol. 20, no. 5, pp. 768–780, May 2009.

[11] V. A. Nguyen, J. A. Starzyk, W.-B. Goh, and D. Jachyra, "Neural network structure for spatio-temporal long-term memory," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 6, pp. 971–983, Jun. 2012.

[12] S. Panzeri, N. Brunel, N. K. Logothetis, and C. Kayser, "Sensory neural codes using multiplexed temporal scales," *Trends Neurosci.*, vol. 33, no. 3, pp. 111–120, 2010.

[13] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting," *Neural Comput.*, vol. 22, no. 2, pp. 467–510, 2010.

[14] R. V. Florian, "The chronotron: A neuron that learns to fire temporally precise spike patterns," *PLoS One*, vol. 7, no. 8, p. e40233, 2012.

[15] A. Mohemmed, S. Schliebs, S. Matsuda, and N. Kasabov, "SPAN: Spike pattern association neuron for learning spatio-temporal spike patterns," *Int. J. Neural Syst.*, vol. 22, no. 4, pp. 1250012-1–1250012-17, 2012.

[16] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns," *PLoS One*, vol. 8, no. 11, p. e78318, 2013.

[17] D. Z. Jin, "Decoding spatiotemporal spike sequences via the finite state automata dynamics of spiking neural networks," *New J. Phys.*, vol. 10, no. 1, p. 015010, 2008.

[18] S. Byrnes, A. N. Burkitt, D. B. Grayden, and H. Meffin, "Learning a sparse code for temporal sequences using STDP and sequence compression," *Neural Comput.*, vol. 23, no. 10, pp. 2567–2598, 2011.

[19] R. Kempter, W. Gerstner, and J. L. van Hemmen, "Spike-based compared to rate-based Hebbian learning," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 1998, pp. 125–131.

[20] R. VanRullen, R. Guyonneau, and S. J. Thorpe, "Spike times make sense," *Trends Neurosci.*, vol. 28, no. 1, pp. 1–4, 2005.

[21] R. Gütig, "To spike, or when to spike?" *Current Opinion Neurobiol.*, vol. 25, pp. 134–139, Apr. 2014.

[22] V. J. Uzzell and E. J. Chichilnisky, "Precision of spike trains in primate retinal ganglion cells," *J. Neurophysiol.*, vol. 92, no. 2, pp. 780–789, 2004.

[23] P. Reinagel and R. C. Reid, "Temporal coding of visual information in the thalamus," *J. Neurosci.*, vol. 20, no. 14, pp. 5392–5400, 2000.

[24] Z. F. Mainen and T. J. Sejnowski, "Reliability of spike timing in neocortical neurons," *Science*, vol. 268, no. 5216, pp. 1503–1506, 1995.

[25] J. J. Hopfield, "Pattern recognition computation using action potential timing for stimulus representation," *Nature*, vol. 376, no. 6535, pp. 33–36, 1995.

[26] N. Kasabov *et al.*, "Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke," *Neurocomputing*, vol. 134, pp. 269–279, Jun. 2014.

[27] T. J. Strain, L. McDaid, T. M. McGinnity, L. P. Maguire, and H. M. Sayers, "An STDP training algorithm for a spiking neural network with dynamic threshold neurons," *Int. J. Neural Syst.*, vol. 20, no. 6, pp. 463–480, 2010.

[28] J. J. Wade, L. J. McDaid, J. A. Santos, and H. M. Sayers, "SWAT: A spiking neural network training algorithm for classification problems," *IEEE Trans. Neural Netw.*, vol. 21, no. 11, pp. 1817–1830, Nov. 2010.

[29] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neurosci.*, vol. 3, no. 9, pp. 919–926, 2000.

[30] D. E. Feldman, "The spike-timing dependence of plasticity," *Neuron*, vol. 75, no. 4, pp. 556–571, 2012.

[31] R. Legenstein, C. Naeger, and W. Maass, "What can a neuron learn with spike-timing-dependent plasticity?" *Neural Comput.*, vol. 17, no. 11, pp. 2337–2382, 2005.

[32] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2014.

[33] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Rapid feedforward computation by temporal encoding and learning with spiking neurons," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1539–1552, Oct. 2013.

[34] Q. Yu, H. Tang, K. C. Tan, and H. Yu, "A brain-inspired spiking neural network model with temporal encoding and learning," *Neurocomputing*, vol. 138, pp. 3–13, Aug. 2014.
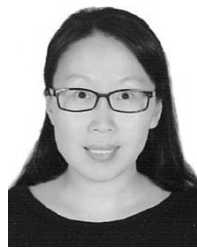
[35] S. M. Bohte, H. La Poutré, and J. N. Kok, "Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 426–435, Mar. 2002.

[36] Z. Nadasdy, "Information encoding and reconstruction from the phase of action potentials," *Frontiers Syst. Neurosci.*, vol. 3, p. 6, Jul. 2009.

[37] R. R. Llinás, A. A. Grace, and Y. Yarom, "*In vitro* neurons in mammalian cortical layer 4 exhibit intrinsic oscillatory activity in the 10- to 50-Hz frequency range," *Proc. Nat. Acad. Sci. USA*, vol. 88, no. 3, pp. 897–901, 1991.

[38] J. Jacobs, M. J. Kahana, A. D. Ekstrom, and I. Fried, "Brain oscillations control timing of single-neuron activity in humans," *J. Neurosci.*, vol. 27, no. 14, pp. 3839–3844, 2007.

[39] K. Koepsell *et al.*, "Retinal oscillations carry visual information to cortex," *Frontiers Syst. Neurosci.*, vol. 3, p. 4, Apr. 2009.

[40] C. Kayser, M. A. Montemurro, N. K. Logothetis, and S. Panzeri, "Spike-phase coding boosts and stabilizes information carried by spatial and temporal spike patterns," *Neuron*, vol. 61, no. 4, pp. 597–608, 2009.

[41] J. Hu, H. Tang, K. C. Tan, H. Li, and L. Shi, "A spike-timing-based integrated model for pattern recognition," *Neural Comput.*, vol. 25, no. 2, pp. 450–472, 2013.

[42] M. C. W. van Rossum, "A novel spike distance," *Neural Comput.*, vol. 13, no. 4, pp. 751–763, 2001.

[43] N. E. Schoppa and G. L. Westbrook, "Regulation of synaptic timing in the olfactory bulb by an A-type potassium current," *Nature Neurosci.*, vol. 2, no. 12, pp. 1106–1113, 1999.

[44] O. Shriki, D. Hansel, and H. Sompolinsky, "Rate models for conductance-based cortical neuronal networks," *Neural Comput.*, vol. 15, no. 8, pp. 1809–1841, 2003.

[45] S. Ghosh-Dastidar and H. Adeli, "A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection," *Neural Netw.*, vol. 22, no. 10, pp. 1419–1431, 2009.

[46] I. Sporea and A. Grüning, "Supervised learning in multilayer spiking neural networks," *Neural Comput.*, vol. 25, no. 2, pp. 473–509, 2013.

[47] Y. Xu, X. Zeng, and S. Zhong, "A new supervised learning algorithm for spiking neurons," *Neural Comput.*, vol. 25, no. 6, pp. 1472–1511, 2013.

[48] 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, no. 11, pp. 2531–2560, 2002.

[49] D. A. Butts *et al.*, "Temporal precision in the neural code and the timescales of natural vision," *Nature*, vol. 449, no. 7158, pp. 92–95, 2007.

[50] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.

[51] S. Thorpe and J. Gautrais, "Rank order coding," in *Computational Neuroscience*. New York, NY, USA: Springer-Verlag, 1998, pp. 113–118.

[52] W. Bair and C. Koch, "Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey," *Neural Comput.*, vol. 8, no. 6, pp. 1185–1202, 1996.

[53] B. L. Lewis and P. O'Donnell, "Ventral tegmental area afferents to the prefrontal cortex maintain membrane potential 'up' states in pyramidal neurons via D_1 dopamine receptors," *Cerebral Cortex*, vol. 10, no. 12, pp. 1168–1175, 2000.

[54] J. Anderson, I. Lampl, I. Reichova, M. Carandini, and D. Ferster, "Stimulus dependence of two-state fluctuations of membrane potential in cat visual cortex," *Nature Neurosci.*, vol. 3, no. 6, pp. 617–621, 2000.

Rui Yan (M'11) received the bachelor's and master's degrees from the Department of Mathematics, Sichuan University, Chengdu, China, in 1998 and 2001, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, in 2006.

She is currently a Research Scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. Her current research interests include intelligent robots, nonlinear control, neural computation, power systems analysis, and control.

Huajin Tang (M'01) received the B.Eng. degree from Zhejiang University, Hangzhou, China, in 1998, the M.Eng. degree from Shanghai Jiao Tong University, Shanghai, China, in 2001, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2005.

He was a System Engineer with STMicroelectronics, Singapore, from 2004 to 2006, and then a Post-Doctoral Fellow with the Queensland Brain Institute, University of Queensland, Brisbane, QLD, Australia, from 2006 to 2008. He is currently a Research Scientist leading the Cognitive Computing Group with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. He has authored one monograph (Springer-Verlag, 2007) and over 30 international journal papers. His current research interests include neural computation, neuromorphic cognitive systems, neurocognitive robots, and machine learning.

Dr. Tang serves as an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and an Editorial Board Member of *Frontiers in Robotics and AI*.

Kay Chen Tan (SM'08–F'14) received the B.Eng. (Hons.) degree in electronics and electrical engineering and the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He is involved in research on computational and artificial intelligence, with applications to multiobjective optimization, scheduling, automation, data mining, and games. He has authored over 100 journal papers, over 100 papers in conference proceedings, and co-authored five books.

Dr. Tan was a recipient of the IEEE Computational Intelligence Society Outstanding Early Career Award for his contributions to evolutionary computation in multiobjective optimization in 2012. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. He was the Editor-in-Chief of the *IEEE Computational Intelligence Magazine* from 2010 to 2013. He serves as an Associate Editor/Editorial Board Member of over 15 international journals, such as the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, *Evolutionary Computation* (MIT Press), the *European Journal of Operational Research*, the *Journal of Scheduling*, and the *International Journal of Systems Science*.

Qiang Yu (S'12) received the B.Eng. degree in electrical engineering and automation from the Harbin Institute of Technology, Harbin, China, in 2010, and the Ph.D. degree from the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, in 2014.

He is currently a Post-Doctoral Research Fellow with the Max-Planck-Institute for Experimental Medicine, Göttingen, Germany. He is involved in research on neural circuit theories for cognitive computing. His current research interests include learning algorithms in spiking neural networks, neural coding, and cognitive computations.

Haizhou Li (M'91–SM'01–F'14) received the B.Sc., M.Sc., and Ph.D. degrees in electrical and electronic engineering from the South China University of Technology, Guangzhou, China, in 1984, 1987, and 1990, respectively.

He taught with the University of Hong Kong, Hong Kong, from 1988 to 1990, and the South China University of Technology from 1990 to 1994. He was a Visiting Professor with CRIN from 1994 to 1995, a Research Manager with the Apple-ISS Research Centre, Singapore, from 1996 to 1998, the Research Director of Lernout & Hauspie Asia Pacific, Singapore, from 1999 to 2001, and the Vice President with InfoTalk Corporation Ltd., Hong Kong, from 2001 to 2003. He is currently the Principal Scientist and Department Head of Human Language Technology with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. He is also an Adjunct Professor with the National University of Singapore, Singapore, and a Conjoint Professor with the University of New South Wales, Sydney, NSW, Australia. His current research interests include automatic speech recognition, speaker and language recognition, and natural language processing.

Dr. Li was a recipient of the National Infocomm Award in 2002, and the President's Technology Award in Singapore in 2013. He was named one of the two Nokia Visiting Professors by the Nokia Foundation in 2009. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING and the *ACM Transactions on Audio, Speech and Language Processing* from 2015 to 2017. He served on the Editorial Board of *Computer Speech and Language* from 2012 to 2014. He is an Elected Member of the IEEE Speech and Language Processing Technical Committee from 2013 to 2015, and the President of the Asia Pacific Signal and Information Processing Association from 2015 to 2016. He was the Vice President of the International Speech Communication Association from 2013 to 2014. He was the General Chair of ACL 2012 and INTERSPEECH 2014.