WCCI 2010 IEEE World Congress on Computational Intelligence
July, 18-23, 2010 - CCIB, Barcelona, Spain IJCNN

Encoding Real Values into Polychronous Spiking Networks

Cameron Johnson, *Student Member, IEEE,* and G. K. Venayagamoorthy, *Senior Member, IEEE*

Abstract—**Spiking neural networks show promising capability in handling the same kind of scaling up of problems as living brains, due to their more faithful similarity to biological neural networks. The big challenge of dealing with spiking neural networks is getting data into and out of them, which requires proper encoding and decoding methods. Presented in this paper is an adaptation of Izhikevich's model of a polychronous spiking network and an encoding scheme for real valued data. Data is chosen arbitrarily to cover the range of the encoding scheme in order to best demonstrate the network's response to different inputs. Preliminary results show that the network is able to recognize distinct input values and respond to them with unique spiking patterns.**

I. Introduction

PAST work with an abstracted model of a spiking neuron in [1] has shown potential for use in neuroidentification, improving on a traditional MLP in both precision and accuracy. However, the abstracted model amounts to little more than a more complicated activation function. In order to truly harness the power of a spiking neuron, it is probable that a more biologically faithful, literal model must be tried.

Many different spiking neural network algorithms have been designed with an eye towards real-number function approximation. Iannella and Back propose a model for a spiking neural network in [2] which relies on training the individual neurons to classify inputs and return an output value in order to model a function. This is a very linear network, in that its precision is directly correlated to the number of spiking neurons present. In [3], Hopfield proposes using time-encoding for calculating what is referred to as an "analog match." That is, a stimulus in the form of a natural signal which must be categorized into recognized patterns with labels. The encoding neurons in Hopfield's model use an oscillatory background to provide a baseline pulse against which the timing of input spikes may be measured. The encoding is thus accomplished in the timing of the spikes. At the time this paper [3] was written, this was a distinct change from the standard practice of using firing rate as the encoding mechanism.

The oscillatory requirement mentioned above is also present in Freeman's KIII model, based on topologies of neurons categorized by Katchalsky [4]. The K0 sets consist of neurons that share common inputs and a common sign for outputs. KI sets are interconnected units of K0 sets that share a feedback with a common sign, thus acting as either excitatory or inhibitory structures. KI sets are the first level with "weights." KII sets are built from densely interconnected KI sets; and [4] indicates the "most interesting" behavior occurs when the KI sets making up a KII set are of opposite polarity (i.e. some excitatory and others inhibitory). KIII sets are the full network, used primarily to model olfactory systems. Like Hopfield's work, a great deal of concern is shown in [4] for the translation of real-world signals to manipulable symbols. The Freeman model is analyzed further and applied to other problems in [5], [6], and [7], but the model explored in this paper is the Izhikevich model, chosen primarily because it is a recent advance in simulation of brain-like behaviors. It demonstrates many activities and behaviors common to real brains, and the encoding method introduced in this study relies on spatial as well as temporal mechanisms in the neural pool [8].

The Izhikevich neuron model introduced in [8] reproduces a number of spiking output patterns resembling activity in living brains, depending on the values given to four variables (*a*, *b*, *c*, and *d*, shown in (1) and (2)) in the update equations. It is hoped that using this spiking neuron model will allow for more flexible function approximation with greater scaling capacity and more efficient computation.

In [9], Izhikevich updates the equations given in Fig. 1 by introducing the idea of "polychrony." That is, by introducing synaptic delays to the network, it becomes possible to have coincidental spike arrivals from temporally-separated neural spiking. This polychronous spiking network (PSN) utilizes these delayed synaptic connections to cause the timing of multiple neurons' firings to be important in determining whether a given secondary neuron fires. Such behavior could allow for rapid recognition of patterns, with a single neuron's spiking output being a flagged result of a pattern of other neurons reacting to stimulus, resulting in very fast, brain-like identification of and reaction to particular complex stimuli [10]. In this paper, the PSN's ability to respond to encoded, meaningful information is explored, with an eye towards achieving function approximation with a PSN.

In order to test the encoding scheme, it is necessary to choose some test inputs. These numbers may be selected arbitrarily, but should be reasonably well-spaced from each other across the interval of numbers which can be encoded. This allows the output response of the PSN to be examined across the range the encoding scheme can handle. It is desirous that the output responses of the PSN be distinct for different inputs; this behavior would indicate that the encoding method is successfully inputting meaningful information into the network.

In the next section, the Izhikevich equations are discussed, as is the mechanism whereby synaptic delays are introduced.

Manuscript revised April 28, 2010. This work was supported by NSF/EFRI COPN #0836017.

C. Johnson is with the Real-Time Power and Intelligent Systems Laboratory, Rolla, MO 65409-0220 USA (e-mail: cameron.e.johnson@gmail.com).

G. K. Venayagamoorthy is the Director of the Real-Time Power and Intelligent Systems Laboratory, Rolla, MO 65409-0220 USA (e-mail: gkumar@ieee.org).

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

A sample of how the code functions with random thalamic excitations is demonstrated, so that it may be contrasted with the third section of this paper, which unveils a possible method for encoding data into the PSN and demonstrates the performance of the Izhikevich model of a PSN when given real values so encoded. The paper then concludes with a discussion of decoding mechanisms and challenges of using a PSN for function approximation.

II. Izhikevich Model of a Polychronous Spiking Network

A. *Mathematical Model*

This neuron model, developed by Izhikevich in [8], uses equations (1) and (2) to govern the charge on a single neuron, as well as to monitor its firing output.

$$v(k)=\begin{cases} v(k-1)+0.04v^2(k-1)+5v(k-1)\ldots & \text{if } v<30\text{mV} \\ \ldots+140+u(k-1)+I & \\ c & \text{if } v\geq 30\text{mV} \end{cases} \quad (1)$$

$$u(k)=\begin{cases} u(k-1)+a\left[bv(k-1)-u(k-1)\right] & \text{if } v<30\text{mV} \\ u(k-1)+d & \text{if } v\geq 30\text{mV} \end{cases} \quad (2)$$

The variables *a, b, c,* and *d* control the spiking behavior of the Izhikevich neuron. Variables *a* and *b* control the sensitivity of the neuron to increasing voltage and how quickly it can be ready to spike again after having spiked once, while *c* and *d* are reset-control variables. The thalamic input *I* is where the affects of other neurons' spiking, as well as ambient electrical noise, are injected into the neuron. In other words, this is what causes a neuron to gain charge on its soma, making it possible for it to spike at all. The value of *I* is the total influx of spiking information (after all synaptic delays have been accounted for) into the neuron at time step *k*. Eq. (1) gives the update model for the voltage *v* on the soma at any discrete time *k*, and (2) gives the update model for the decay value *u* at the same instant *k* in order to keep the neuron model behaving like a real spiking neuron.

B. *Performance of the Izhikevich Model*

In Izhikevich's Spiking Neural Network (SNN) model, a sparsely-connected pool of these neurons are initialized with random synaptic weights, connecting them with roughly 10% of the other neurons in the pool. In preparation to compare to the PSN and the PSN with encoded real-valued inputs, this SNN was run for one simulated second, with a time-step of 1ms, as shown in Fig. 1. The thalamic input *I* is stimulated with random noise on top of the spikes incoming from connected neurons. See [8] for the actual Matlab code used to generate Fig. 1. Neurons 1 through 800 are excitatory, and neurons 801 through 1000 are inhibitory.

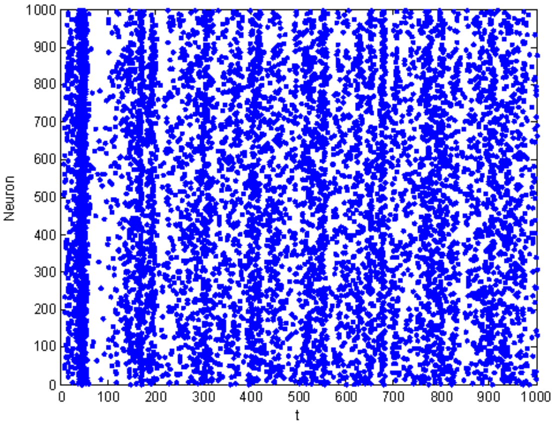

Fig. 1. Izhikevich Matlab model output: 1000 neurons firing over 1000ms. Each dot represents that neuron firing during that millisecond.

The PSN code introduced in [9] adds synaptic delays to the neural communications. The delays are randomly determined, but fixed once set. The synaptic weights, on the other hand, are trained via spike-timing-dependent plasticity (STDP), which is a method that encourages connections that seem to trigger spikes to strengthen, and causes connections which carry spikes that accomplish nothing to decay. The code needs to run for roughly six hours of simulated time before the weights are trained enough for use. Fortunately, there is no need to initialize the PSN's synaptic weights randomly each time the code is used; by storing the trained weights to use as the starting point for the PSN code, only one such six-hour (simulated time) session is required.

As potentially exciting as Izhikevich's PSN model is, however, as presented it is a model that produces pretty spiking patterns, mainly of interest only to neurobiologists. This paper attempts to make it of interest to the Computational Intelligence (CI) community by exploring means of inputting data into the spiking neural network. The eventual goal, perhaps obviously, is extracting data from the network, in order to use the PSN as a function approximator (much as better-established neural networks are utilized today). It is hoped that the PSN's more powerful modeling of living brain activity, once harnessed, will prove to be a much more powerful neural network than those currently considered "standard."

The most obvious place to inject real data into the network neurons is in the thalamic input. The variable *I* is where the neuron receives external input, whether from other neurons connected via synapses or from the random thalamic noise. So, if the real data is to be injected via *I*, the question arises as to how essential the thalamic input – the random noise – is to the function of the PSN. As Fig. 2 shows, without any sort of external stimulus to the neurons, the network does not respond at all.

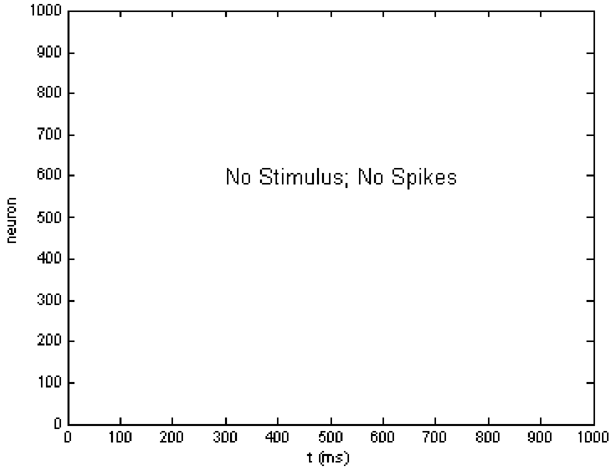

Fig. 2. One second of the PSN with no thalamic stimulation and no input.

Thus, some form of external stimulus is required to get the PSN to do anything. As will soon be seen, providing external stimulus in the form of the desired input data has definite impact on the spiking behavior of the PSN. Before real values can be injected into the network, however, some method of encoding that data into a form useable by the spiking neurons must be developed.

III. Encoding Real Values for a PSN

The PSN used in this experiment consists of 1000 spiking neurons. Of these, 800 are excitatory and the remaining 200 are inhibitory. Twenty-four neurons are chosen to be input channels through which external excitation can be applied to the network.

These 24 input neurons are chosen from the reservoir of 800 excitatory neurons, and these encoded spike patterns are injected directly into the thalamic input variable *I* in (1). The output of all 800 excitatory neurons and all 200 inhibitory neurons is then monitored. Fig. 3 shows the conceptualization of the 24-neuron, 12-bit encoding with the spiking neuron pool as a dynamic reservoir, as well as eight randomly selected neurons whose outputs are monitored.

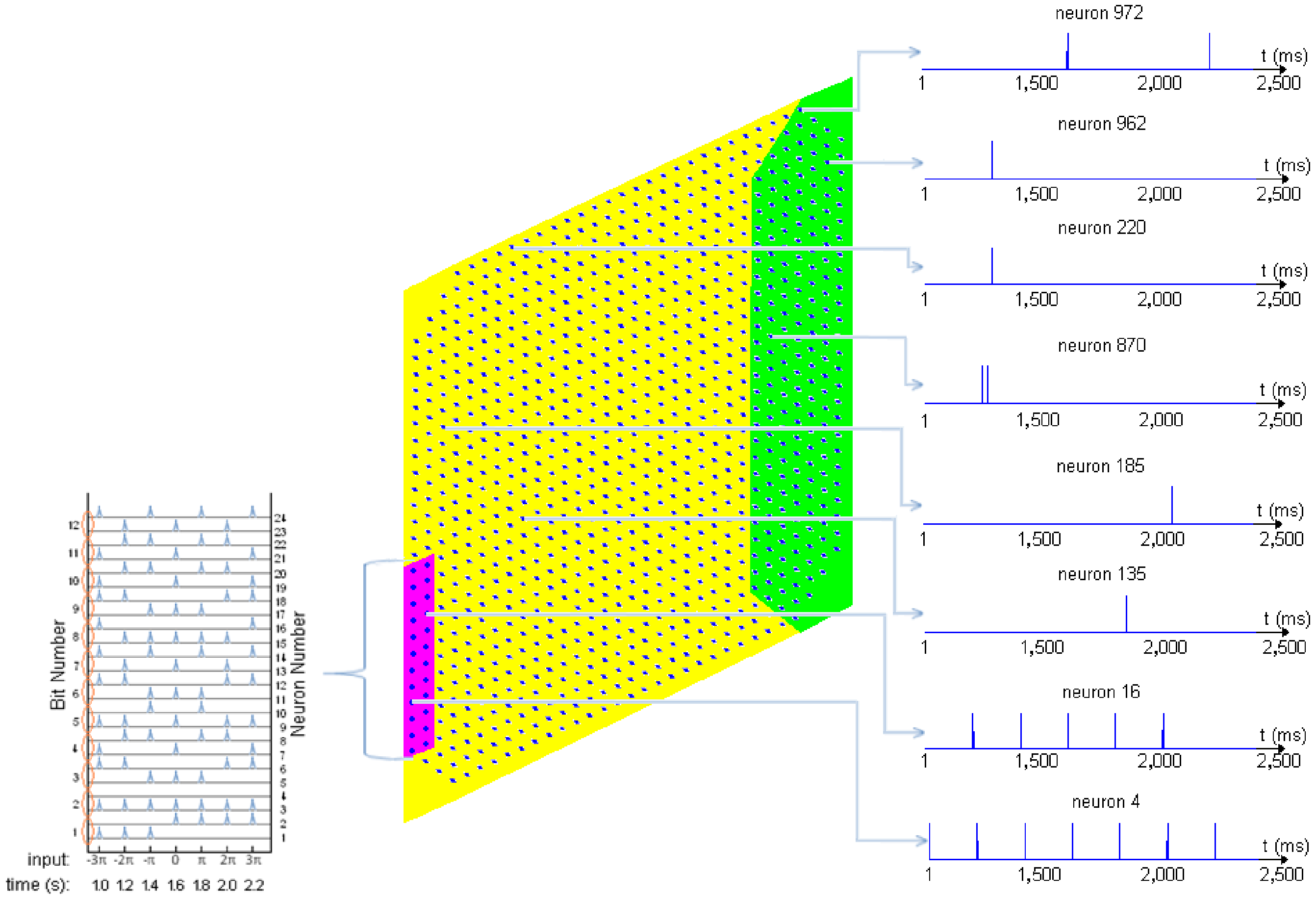

Fig. 3. Input neuron concept. 24 neurons used to encode 12 bits. The number of bits of resolution is arbitrarily chosen; the requirement is two neurons per bit. e.g., for 32 bit resolution, 64 input neurons must be chosen.

A. *Encoding Scheme*

The experimental input method is to utilize grey code to convert the numerical input we wish to give the PSN into patterns of spiking inputs. The precise method chosen is to utilize 11 bits to encode numbers from 0 to 10, and a 12th bit to serve as a sign, thus allowing for a range of -3π to 3π for the experiments presented here. A 12-bit encoding scheme is chosen for compatibility with the real-time digital simulator (RTDS) available in the Real-Time Power and Intelligent Systems (RTPIS) Lab for future real-time simulations. Gray Coding is used to ensure that changes from one value to the next have minimal difference in the input pattern, and thus any continuous topology in the network can be taken advantage of. Twenty-four neurons out of a pool of 1000 are used as the inputs.

Fig. 4 illustrates how each bit has two neurons to which its input goes: one neuron receives a spike when the bit is "1," the other receives a spike when the bit is "0." If no input is being given, none of the input neurons receive external spikes. There is no "random noise" introduced into these thalamic inputs; all spiking responses are strictly to the input patterns.

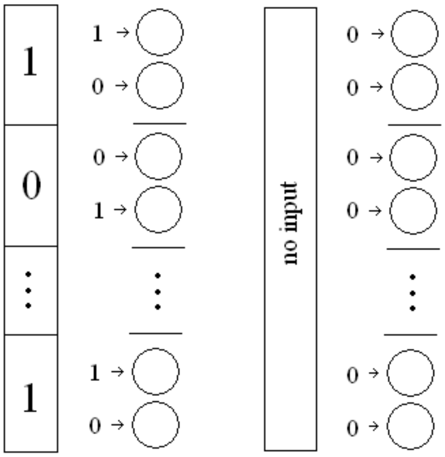

Fig. 4. Encoding scheme for the 24 neurons, with example inputs and their meanings. These 24 neurons represent 12 bits, which are used to input real values in a gray code.

B. *Response of PSN to Sample Encoded Values*

With the means of encoding data established, the experiment can be performed. In this paper, input patterns representing -3π, -2π, -π, 0, π, 2π, and 3π have been chosen. In Fig. 5 through Fig. 11, the PSN is initialized with the synaptic weights that had been trained for six simulated hours. In the first millisecond of the simulation, the input is presented; the remaining simulation time simply shows the spiking response pattern of the network.

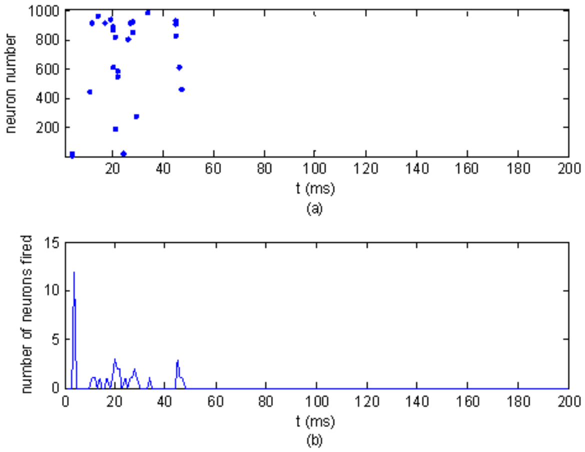

Fig. 5. -3π input for 1 ms. (a) spiking response of each neuron. (b) total number of neurons firing each millisecond.

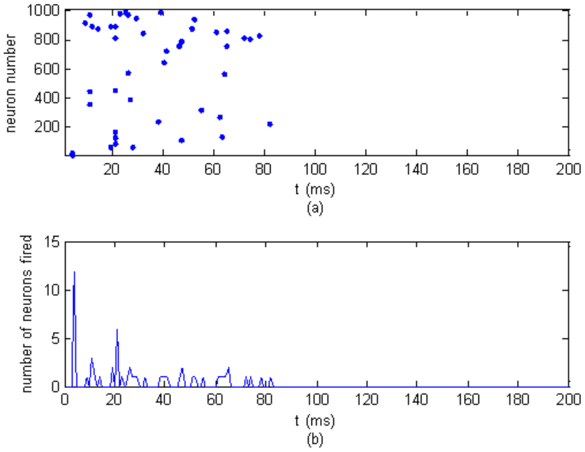

Fig. 6. -2π input for 1 ms. (a) spiking response of each neuron. (b) total number of neurons firing each millisecond.

Every neuron in the PSN is monitored for its spiking response, whether excitatory or inhibitory, and whether it is an input neuron or not. As before, neurons 1 through 800 are excitatory, and neurons 801 through 1000 are inhibitory. Neurons 1 through 24 receive the encoded stimulus in the first millisecond. The bottom graph in each of the figures simply shows how many neurons fired in any given millisecond.

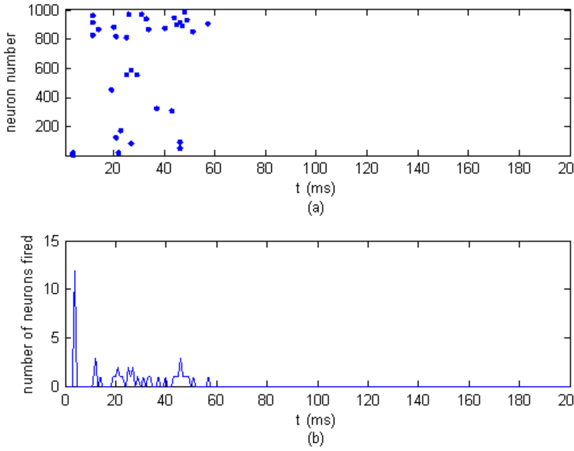

Fig. 7. -π input for 1 ms. (a) spiking response of each neuron. (b) total number of neurons firing each millisecond.

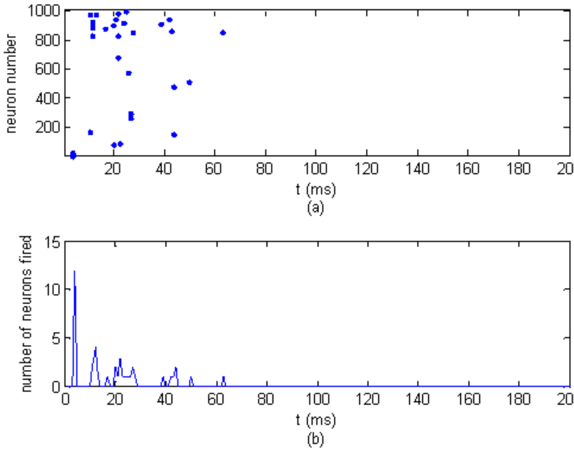

Fig. 8. 0 input for 1 ms. (a) spiking response of each neuron. (b) total number of neurons firing each millisecond.

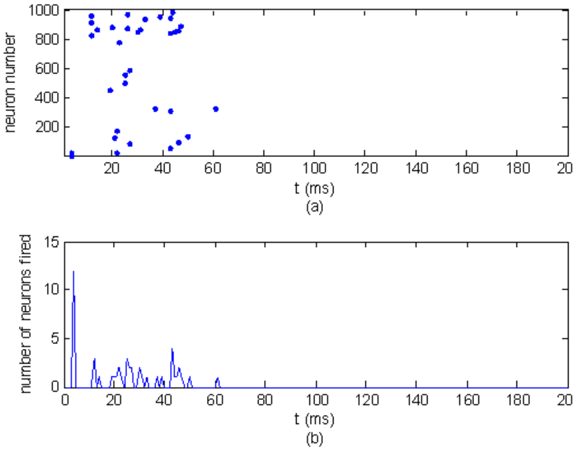

Fig. 9. π input for 1 ms. (a) spiking response of each neuron. (b) total number of neurons firing each millisecond.

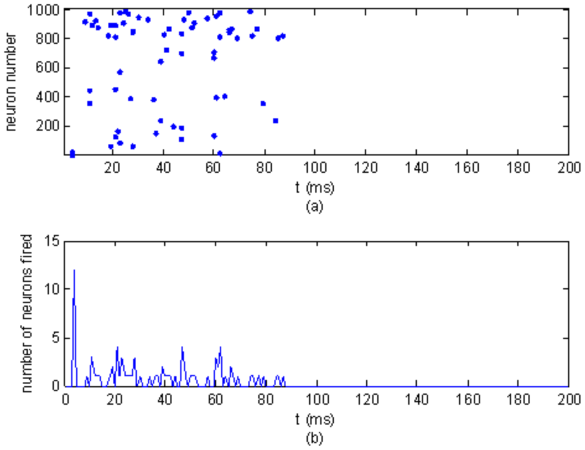

Fig. 10. 2π input for 1 ms. (a) spiking response of each neuron. (b) total number of neurons firing each millisecond.

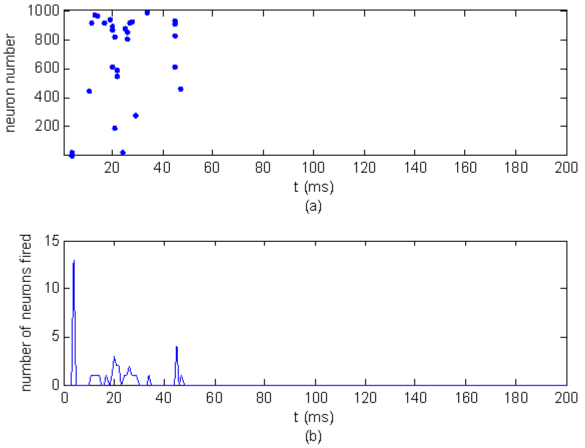

Fig. 11. 3π input for 1 ms. (a) spiking response of each neuron. (b) total number of neurons firing each millisecond.

As can be seen, there is still significant spiking response even with a single ms of input, though it does die out on the order of 100 ms. Of particular note is the fact that each of these response patterns are unique. Thus, there is definite recognition in the network of different inputs.

However, the question arises: are these outputs stable? The answer is "yes," at least to a certain degree. Further experiments applying the same input repeatedly, without resetting the synaptic weights (and thus allowing them to adapt according to STDP), yields identical output patterns over the first six repetitions (with one input presented every 200 ms). In the seventh, there is a mild alteration to the trailing end of the spiking response, as can be seen clearly in Fig. 12 when compared one-by-one to the corresponding Fig. 11. While experiments were done with the other input values (-3π through 2π), Fig. 12 in comparison to Fig. 11 is typical. Perhaps even more interesting, when presentations were continued out to as many as 80 iterations, at 200 ms between presentations, the output pattern remained the same as the seventh, implying that, at least to that point, the outputs are stable even while STDP training is occurring.

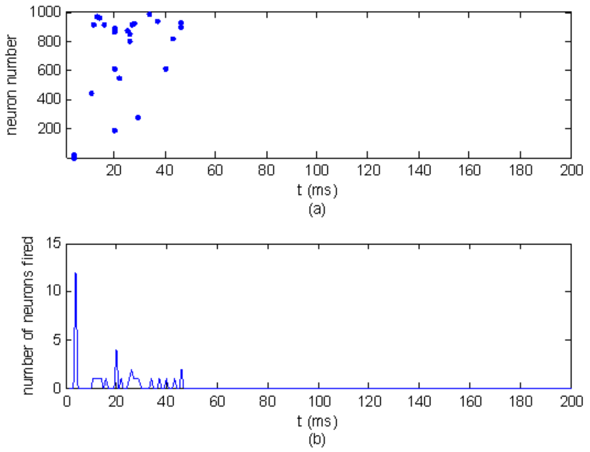

Fig. 12. 3π input for 1 ms on the 7th iteration of the same input without resetting the synaptic weights. This is the first iteration that shows a different response from the first presentation, and no iterations after this 7th one are different from this 7th one. (a) spiking response of each neuron. (b) total number of neurons firing each millisecond.

As illustrative as the pictures of the spiking response are, they may be deceptive. Further analysis of the PSN's response to the chosen input set to determine just how unique the responses are has been done by examining exactly which neurons fired in response to each stimulus. To simplify analysis, only whether a given neuron fired at all is considered; if it fired more than once, it still only counts once. Table I shows a count of the number of unique neurons which fired for both the stimulus represented in the row and in the column of each cell. Thus, the diagonal gives the total number of neurons that fired at least once for a given input stimulus (since, for instance, every neuron that fired in response to 3π also fired in response to 3π, so there is 100% overlap). While Table I does not provide sufficient information to decode outputs, the dissimilarity in which neurons fired in response to a given input indicates that the different inputs are encoded in a way that the PSN recognizes as distinct and different.

TABLE I
NUMBER OF SHARED NEURAL FIRINGS BETWEEN SAMPLE INPUTS

	-3π	-2π	-π	0	π	2π	3π
-3π	36	11	8	9	7	10	36
-2π	11	54	15	11	13	39	11
-π	8	15	44	8	36	17	9
0	9	11	8	40	10	14	11
π	7	13	36	10	46	16	9
2π	10	39	17	14	16	74	11
3π	36	11	9	11	9	11	39

To further analyze the data presented in Table I, Fig. 13 is a bar graph representing the number of unique neural firings each stimulus caused. This corresponds to the diagonal values in Table I. Fig. 14 represents Table I as a three-dimensional bar graph. Looking at these, it becomes clear that there is sufficient uniqueness between the firing patterns of the PSN in response to the different inputs that one can declare that the PSN has, in fact, recognized them as different.

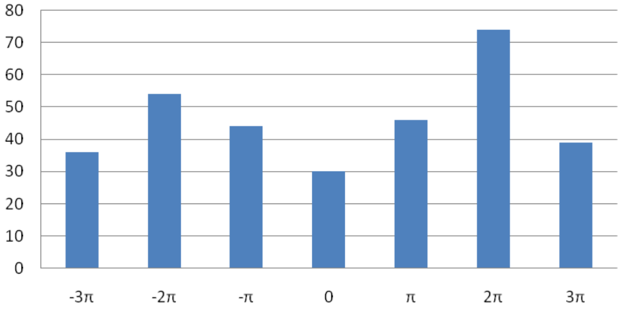

Fig. 13. A count of unique neural firings for different encoded inputs.

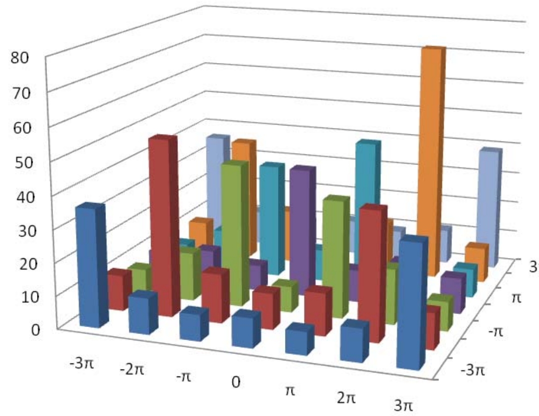

Fig. 14. The number of neurons which fired for two given encoded inputs. The sharp drop-off in numbers of shared neural firings between two non-identical inputs is a good sign that the firing patterns are unique to each input pattern.

Fewer shared neurons firing than neurons fired for any single input indicates that there are a number of neurons that *only* fire for one input, and not others. While increasing the number of inputs in the testing set will likely lead to finding more input patterns that a given neuron will react to, the unique combinations of neural firing patterns will be sufficiently rich for the PSN to demonstrate its understanding of the encoded input.

IV. DECODING: THE KEY TO FUNCTION APPROXIMATION

This simple encoding produces patterns that can be identified with specific inputs, which can lead to powerful classifiers. Work has been done with spiking networks as classifiers [11], accomplishing such things as using an SNN to enact the XOR function, and the results shown above indicate that, with proper identification tools on the decoding end, the PSN could mimic this level of categorization.

However, the goal of this encoding method is not simple classification. To the knowledge of the authors, there has been no successful work thus far in performing continuous function approximation using spiking neurons. It is the belief of the authors that, with appropriate decoding, such function approximation is within the capability of the PSN.

While there is sufficient demonstration of the uniqueness of the firing patterns of the PSN shown in Table I and Figs. 13 and 14, identifying the unique response patterns is still a challenge without doing mere pattern-matching. Information lost in the analysis in Section III includes how many times any given neuron fired (other than “not at all” or “at least once”) and the timing of the spikes, both of which are further unique identifiers which may aid in determining how the PSN processes the encoded inputs.

The exact method of decoding is the subject of ongoing research. Hopfield’s coincidence-detection method is worth investigating, but the sheer volume of PSN spikes makes this prohibitive as a first step [3]. Currently, work is being done to explore similarities between Radial Basis Function (RBF) networks and the patterns of output spikes demonstrated by the PSN, in hopes that the similarity between the SNN used in [1] and RBF functions might lead to a breakthrough on decoding the PSN’s spike patterns. One possible way to look at the PSN is as a form of echo-state network (ESN), which also has similarities to an RBF.

The trick, in this form of decoding, would seem to be to determine how the spike patterns respond to a large enough set of training inputs. Then, using a clustering mechanism such as *k*-means, the inputs and outputs would be classified, and the only thing that needs online training would be the output weights used to ultimately translate the spike patterns into meaningful output data. In this fashion, it is hoped to translate the discrete classification space into a continuous function space.

V. Conclusion

Real data has been successfully encoded and propagated through a polychronous spiking network. The PSN has been shown to have unique, repeatable spiking patterns in response to any given input. Such patterns can be used to identify categorically what input the PSN received, and thus show that the PSN is as capable of categorization problems as other SNN works. Because the encoding method chosen has a 12 bit granularity, there are 2^{12} possible inputs, each of which has a unique output, leading to very powerful classification capabilities.

The challenge the authors have taken on, however, is to move beyond this preliminary progress with the PSN, and demonstrate that, with proper decoding mechanisms, the PSN can be a function approximator in continuous state space. While this work is still underway, there are several promising avenues of research under investigation, with exciting possible ties to both echo state network, liquid state machine and radial basis function network models. With the PSN demonstrated to be able to take real-valued data and operate on it in repeatable fashions, it is perhaps the closest spiking neural network model to realizing the promise of SNNs in general: biological faithfulness yielding increasing scalability and power.

References

[1] C. Johnson, G. K. Venayagamoorthy, and P. Mitra, “Comparison of a spiking neural network and an MLP for robust identificaiton of generator dynamics in a multimachine power system,” *Neural Networks*, Vol. 22, Issues 5-6, July/August 2009, pp. 833 - 841.

[2] N. Iannella, A. D. Back, “A spiking neural network architecture for nonlinear function approximation.” *Neural Networks*, Vol. 14, Special Issue, April 2001, pp. 933-939.

[3] J. J. Hopfield, “Pattern recognition computation using action potential timing for stimulus representation.” *Nature*, Vol 376, July 1995, pp. 33-36.

[4] J. C. Principe, V. G. Taveres, J. G. Harris, and W. J. Freeman, “Design and implementation of a biologically realistic olfactory cortex in analog VLSI.” *Proceedings of the IEEE*, Vol. 89, Issue 7. 2001. pp. 1030-1051.

[5] R. Kozma, W. J. Freeman. “A possible mechanism for intermittent oscillations in the KIII model of dynamic memories.” *International Joint Conference on Neural Networks*, Vol. 1. 1999. pp. 52-57.

[6] R. Kozma, W. J. Freeman. “Encoding and recall of noisy data as chaotic spatio-temporal memory patterns in the style of the brains.” *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks,* Vol. 5. 2000. pp. 33-38.

[7] J. Zhang, G. Li, and W. J. Freeman. “Analysis of bionic olfactory neural networks based on small-world networks view.” *7th World Congress on Intelligent Controll and Automation*, 2008. pp. 1537-1539.

[8] E. Izhikevich, “Simple Model of Spiking Neurons,” *IEEE Transactions on Neural Networks*, Vol. 14, Nov. 2003, pp. 1569-1572.

[9] E. Izhikevich, “Polychronization: computation with spikes,” Neural *Computation*, Vol. 18, pp. 245-282.

[10] W. Gerstner, W. Kistler, Spiking Neuron Models. Cambridge University Press, 2002.

[11] S. Ghosh-Dastidar, H. Adeli, “A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seisure detection,” *Neural Networks*, Vol. 22, 2009, pp. 1419-1431.