

Temporal pattern recognition using spiking neural networks for cortical neuronal spike train decoding

Huijuan Fang*, Yongji Wang*, Jiping He**, Shan Liu*

*Key Laboratory for Image Processing and Intelligent Control of Education Ministry of China,
Department of Control Science and Engineering, Huazhong University of Science and Technology,
Wuhan, 430074 China (Tel: 86-27-87557284; e-mail: Huijuan.Fang@gmail.com)

**The Harrington Department of Bioengineering and the Center for Neural Interface Design of The Biodesign Institute,
Arizona State University, Tempe, AZ 85287 USA (e-mail: Jiping.He@asu.edu)

Abstract: Most experimental and decoding algorithm studies of brain neural signals assume that neurons transmit information as a rate coding, but recent studies on the fast cortical computations indicate that temporal coding is probably a more biologically plausible scheme used by neurons. We introduce spiking neural networks (SNN) which consist of spiking neurons propagate information by the timing of spikes to analyze the cortical neural spike trains directly without temporal information lost. The SNN based temporal pattern classification is compared with the conventional artificial neural networks (ANN) based firing rate analysis. The results show that the SNN algorithm can achieve higher accuracy, which demonstrates that temporal coding is a viable code for fast neural information processing and the SNN approach is suitable for recognizing the temporal pattern in the cortical neural signals.

1. INTRODUCTION

To help the people who suffered neurological diseases and injuries to enhance or replace their impaired or lost motor functions, researches have proposed to establish an alternative non-muscular communication and control channel between motor cortical neurons and artificial actuators (Taylor, Helms Tillery, and Schwartz, 2002; Wolpaw, Birbaumer, McFarland, Pfurtscheller, and Vaughan, 2002). For this purpose, it is important to decode the information contained in neural signals and convert it to operate motor prosthetic devices.

Biological neurons send information in the form of a sequence of spikes. Traditionally the firing rate of the spike train is viewed as the representation of neural activity. Many algorithms have been used to extract the useful information from the neural rate coding data (Fang, Wang, Huang, and He, 2006; Georgopoulos, Schwartz and Kettner, 1986; Schwartz, Taylor, and Helms Tillery, 2001; Wessberg *et al.*, 2000), for example, population vector algorithm (PVA), artificial neural networks (ANN), and support vector machines (SVM).

However, in the last few years, an increasing number of researches have shown that at least with regard to fast cortical computations, the rate coding itself has become questionable. For instance, visual pattern analysis and classification can be carried out by the primate in just 100 ms (Perrett, Rolls and Caan, 1982; Thorpe and Imbert, 1989), in spite of the fact that it involves a minimum of 10 synaptic stages from the retina to the temporal lobe. This means that each individual processing stage would need to operate in not much more than 10 ms. Whereas the firing rates of neurons involved in these computations are usually below 100 Hz, this period is so short that few neurons will fire more than one spike in this

time. Thus the rate coding scheme seems quite dubious in the context of fast cortical computations.

On the other hand, experimental evidence has accumulated during the last few years which indicates that many biological neural systems use the timing of spikes to encode information (Kempner, Gerstner, Van Hemmen and Wagner, 1996; Lestienne, 1996; Thorpe and Imbert, 1989). These mean that the temporal coding is more biologically plausible than the rate coding. Furthermore, the experimental results from neurobiology have led to the investigation of a new neural network model, spiking neural networks (SNN), which employ spiking neurons as computational units (Maass, 1997a). Spiking neural networks differ from traditional artificial neural networks in that spiking neurons propagate information by the timing of individual spikes, rather than by the rate of spikes. By the inherent property closer to biological neurons than sigmoidal ones, the SNN with temporal coding have more computational power than ANN with sigmoidal activation functions in theory (Maass, 1997b).

We would attempt to extract arm movement directional intent from the timing of spike trains by the SNN algorithm. The motor cortical neural signals were recorded simultaneously with kinematics of arm movement while the monkey performed two directions (left and right) reaching tasks. The spike trains can be inputted into spiking neural networks directly, no need to be transformed into firing rates that largely excludes temporal information in the neural signals. In this study, an error-backpropagation (BP) learning rule which is analogous to the BP rule in ANN is applied to train the SNN algorithm. For comparison, an artificial neural network trained with BP rule is used to analyze the firing rates of the spike trains.

The rest of this paper is organized as follows: in Section 2, we introduce the SNN algorithm briefly and present the one-layer spiking neural network we use. Then, we decode the cortical neural signals by using the SNN approach in Section 3. The results of analyzing timing of spike trains by the SNN algorithm is compared with that of analyzing firing rate of the same spike trains by the ANN method in Section 4. The results are discussed in Section 5. And conclusions and future work follow in Section 6.

2. SPIKING NEURAL NETWORKS

2.1 Spiking Neuron Model

The spiking neuron is closer to the biological neuron than the sigmoidal one, see Fig. 1. The input and output of a spiking neuron is described by the timing of spike trains. We model the behavior of the spiking neuron according to the Spike Response Model (SRM) (Maass and Bishop, 1999). The SRM formally describes how the incoming spike trains are processed to produce a new spike train leaving the neuron. The state of spiking neuron j in the SRM is described by its potential $u_j(t)$. When this membrane potential reaches a threshold θ from below the neuron fires a spike, that is describes by its spike time t_j^f . We will set the threshold θ to 1 for all our simulations.

The output of neuron j is thus fully characterized by the array of spike-times:

$$\mathcal{F}_j = \{t_j^f, 1 \leq f \leq n\} \quad (1)$$

where n denotes the number of spikes. The spike train is chronologically ordered; so if $1 \leq f < g \leq n$, then $t_j^f < t_j^g$.

The potential of neuron j is influenced not only by the spikes of its presynaptic neurons Γ_j , but also by the spikes produced itself:

$$u_j(t) = \sum_{t_j^f \in \mathcal{F}_j} \eta(t - t_j^f) + \sum_{i \in \Gamma_j} \sum_{t_i^g \in \mathcal{F}_i} w_{ji} \varepsilon(t - t_i^g - d_{ji}) \quad (2)$$

where w_{ji} is the weight from neuron i to neuron j and d_{ji} denotes the axonal delay. ε is a standard post synaptic potential (PSP) caused by a presynaptic spike, and can be described by the following spike response function:

$$\varepsilon(t) = \frac{t}{\tau} \exp\left(1 - \frac{t}{\tau}\right) \mathcal{H}(t) \quad (3)$$

where $\mathcal{H}(t)$ denotes the Heavyside step function: $\mathcal{H}(t) = 0$ for $t \leq 0$ and $\mathcal{H}(t) = 1$ for $t > 0$. The time constant τ which determines the rise and decay of the function is set to 7 ms.

After a neuron emitted a spike, the potential decreases instantly to the resting potential, which is defined as zero. And it is more difficult for the neuron to generate a second spike shortly afterwards, the so-called refractoriness. This is

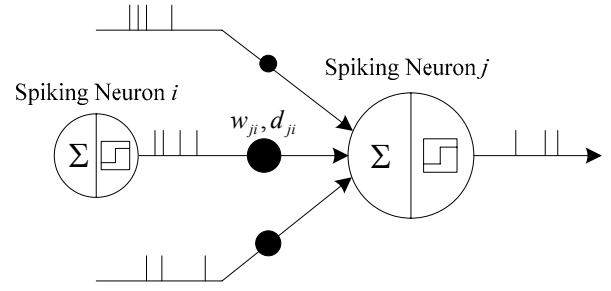


Fig. 1. The input and output of a spiking neuron are series of firing-times called spike trains. The firing-times are represented by vertical bars.

modelled by the η function, for which we use a simple exponential decay:

$$\eta(t) = -\theta \exp\left(-\frac{t}{\tau_r}\right) \mathcal{H}(t) \quad (4)$$

where the time constant τ_r is set to 10 ms and θ is again the threshold of the neuron.

2.2 Network Architecture

The network architecture we use is a simple one-layer feedforward network with multiple delays per connection, as described in Fig. 2(A). The first layer, called input-layer, acts as the input of the network. In our case of spiking neurons the input-neurons are not involved in real processing, but fire predefined spike trains. The second layer is the output-layer. The spike trains of these neurons form the output of the network. In between the input- and output-layer there could be any number of hidden layers, which are not used in this study. The set of connections between two layers of neurons can also be seen as a layer. It is the convention to denote layered network by the number of these layers and not by the number of layers of neurons. So, the network we use is a one-layer network without hidden-layers.

We use multiple synapses per connection. Every synapse has an adjustable weight and a different delay, see Fig. 2(B). These different delays provide a way for the presynaptic neuron to influence the postsynaptic neuron on a longer time-

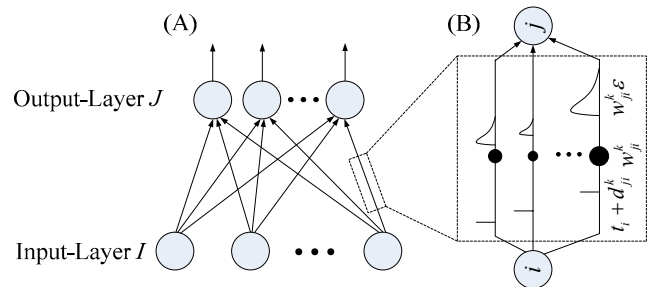


Fig. 2. (A) One-layer feedforward SNN, (B) connection consisting of multiple delayed synapses which all have an adjustable weight.

scale than the time-interval of the spike-response and on a more detailed level. All connection consist of a fixed set of l delays: $D = \{d^k, 1 \leq k \leq l\}$. The corresponding weight is denoted with w_{ji}^k . Equation (2) can thus be rewritten as:

$$u_j(t) = \sum_{t_j^f \in \mathcal{F}_j} \eta(t - t_j^f) + \sum_{i \in \mathcal{I}_j} \sum_{t_i^g \in \mathcal{F}_i} \sum_{k=1}^l w_{ji}^k \varepsilon(t - t_i^g - d^k). \quad (5)$$

2.3 Learning Algorithm

The learning algorithm has to change the weights of the synapses in such a manner to minimize the difference between the actual output and the desired output-pattern. The network could also be learned by changing other parameters, like the synaptic delays (Schrauwen and Van Campenhout, 2004). Here, we choose to keep them fixed. We use the same weight update method with that used by Booij and Nguyen (2005), which is based on the error-backpropagation rule for conventional neural networks and can cope with spiking neurons that emit more than one spike. The weight-change for synapse k from neuron i to neuron j is denoted by

$$\Delta w_{ji}^k = -\kappa \frac{\partial E}{\partial w_{ji}^k}, \quad (6)$$

here κ is the learning rate. The network error is defined to be the mean squared error (MSE) of the first spike of the output neurons J , so later spikes of these neurons are ignored:

$$E = \frac{1}{2} \sum_{j \in J} (t_j^1 - \hat{t}_j^1)^2 \quad (7)$$

where \hat{t}_j^1 denotes the desired spike time. Because the weight w_{ji}^k only influences the spike-times of output-neuron j , the chain rule can be used to expand the second factor of (6) to

$$\frac{\partial E}{\partial w_{ji}^k} = \frac{\partial E}{\partial t_j^1} \frac{\partial t_j^1}{\partial w_{ji}^k}. \quad (8)$$

The first factor on the right-hand side is easy to compute by (7):

$$\frac{\partial E}{\partial t_j^1} = t_j^1 - \hat{t}_j^1. \quad (9)$$

The second factor of (8) is calculated using the SpikeProp algorithm (see Bohte, Kok, and La Poutré, 2002).

$$\frac{\partial t_j^1}{\partial w_{ji}^k} = \frac{\partial u_j(t_j^1)}{\partial w_{ji}^k} \frac{-1}{\partial u_j(t_j^1) / \partial t_j^1}. \quad (10)$$

Both the derivatives on the right-hand side can be derived using (5). Because t_j^1 is the first spike-time, we do not have to worry about the refractoriness-term η . Thus the partial derivatives of the potential with respect to the weight and with respect to the first spike-time are respectively given by

$$\frac{\partial u_j(t_j^1)}{\partial w_{ji}^k} = \sum_{t_i^g \in \mathcal{F}_i} \varepsilon(t_j^1 - t_i^g - d^k), \quad (11)$$

$$\frac{\partial u_j(t_j^1)}{\partial t_j^1} = \sum_{i,k} \sum_{t_i^g \in \mathcal{F}_i} w_{ji}^k \varepsilon'(t_j^1 - t_i^g - d^k). \quad (12)$$

Combing the results of (8) to (12) we can express the formula for the weight-change (6) in a concrete way:

$$\Delta w_{ji}^k = -\kappa \frac{-\sum_{t_i^g \in \mathcal{F}_i} \varepsilon(t_j^1 - t_i^g - d^k)}{\sum_{i,k} \sum_{t_i^g \in \mathcal{F}_i} w_{ji}^k \varepsilon'(t_j^1 - t_i^g - d^k)} (t_j^1 - \hat{t}_j^1). \quad (13)$$

2.4 Solving of No Spiking

If the membrane potential of a neuron is unable to reach the threshold, the neuron will not spike. All the learning rules fail once a neuron stops firing. Some methods have been used to handle the problem, for example, increasing the weight with a small amount and lowering the threshold of the postsynaptic neuron (Moore, 2002; Schrauwen and Van Campenhout, 2004). In our study, we use the following approaches. First the weights should be initialized so high that every neuron initially fires, and that the membrane potential does not surpass the threshold too much. Then we use a small enough learning rate to lower the weights generally to values that minimize the network error, but which do not cause a neuron to stop firing.

3. THE SNN BASED TEMPORAL PATTERN CLASSIFICATION OF CORTICAL NEURAL SPIKE TRAINS

In this section we use the SNN algorithm to classify the temporal pattern in cortical neural spike trains. First the biological experimental design is introduced. Then we brief the ways to acquire neural data which are used to analyze by the SNN method. Finally the simulation setup of the SNN algorithm is described in detail.

3.1 Biological Experimental Design

Rhesus monkeys were trained to perform the visually guided three-dimensional (3-D) reach-to-grasp task on an apparatus (see Fig. 3). The apparatus consists of a central holding pad that serves as the initial position, and two rectangular targets positioned at approximately shoulder height in the frontal plane. The two targets (left and right) are fitted with touch sensors on both sides, and can be rotated in three orientations ($45^\circ, 90^\circ, 135^\circ$). Each movement trial started with monkeys' fingers holding on the central holding pad when the central light was on for a random duration (200-700 ms). Then the central light went off and one of the two targets turned on. The monkeys needed to release the central holding pad, reach for the indicated target, and make a whole-hand grasp so that both sides of the target were contacted. After the target light went off, the animals got a few drops of water as a reward. The target orientation changed randomly after every 3 successful trials to each target. Eighteen trials for one combination of target location and orientation were obtained for most neurons studied. The average duration of cue reaction time (CRT), which is from the illumination of the target light to the central pad release, was 230 ± 90 (SD) ms.

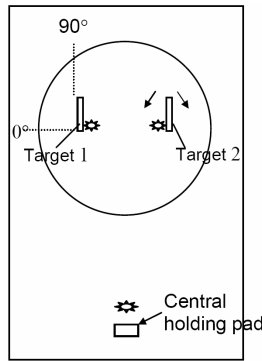


Fig. 3. The front view of the experimental apparatus. It shows two targets and the target orientation definition.

And the average duration of movement time (MT), which is from central pad release to target hit, was 263 ± 62 (SD) ms.

3.2 Neural Data Acquisition and preprocessing

The activity of single neurons in the motor cortex of monkeys was recorded extracellularly by multi-channel micro-drive electrodes when they performed the 3-D reach-to-grasp tasks. In this paper we use the words “units” and “cells” interchangeably to refer to these neurons. The neurons related with tasks are selected to study. A two-way analysis of variance (ANOVA) is used to evaluate whether changes in the average cell discharge in CRT are significantly modulated by target orientation ($a = 3$), or movement direction ($b = 2$), or their interaction effect ($P < 0.05$). Of the 613 neurons studied, 100 (16.3%) are direction-related-only neurons. Their discharge frequency during CRT to one of the targets is significantly higher (ANOVA, effect of movement direction, $P < 0.05$) than that to the other target. And the difference in the neuronal discharge patterns caused by movement direction is consistently observed among movements to the three different target orientations. Thus, for the purpose of extracting directional intent from neural signals, we only analyze the data of the 100 direction-related-only neurons that recorded in the trials to the target orientation 45° in the following.

We use a bin size 50 ms and systematically vary window position (from 200 ms before movement onset to 100 ms after) that was used to extract the neural spike trains. A single-input SNN analysis is performed to find the optimal window position for each direction-related-only neuron. We randomly split the data-set of single neurons into a training-set (50%) and a test-set (50%), and repeat the procedures four times. The window position that maximized the average test-set classification accuracy is the optimal parameter for each unit. Figure 4 shows the distribution of the optimal window position for 100 direction-related-only neurons. Most neurons reached their maximum accuracy with -50 ms (time zero is set at movement onset) window position, so we used that as the optimal window position in the following analysis.

3.3 Simulation Setup

According to the analysis above, we get the spike train data from 75 ms before center release to 25 ms before center release for each direction-related-only neuron and each trial. The spike trains in the 50 ms bin are inputted into the one-layer SNN. And all spike-times are expressed in milliseconds. In some case there is no spike in the bin, so we add a bias input spike (fired at $t = 0$) in each spike train to designate the reference start time. The network output-layer consists of only one spiking neuron, that is trained to fire an early spike at $\hat{t} = 51$ ms if the movement direction is “left” and a late spike at $\hat{t} = 56$ ms if the movement direction is “right”. Each input-neuron is connected to the output-neuron with 28 synapses with delays $\{1, 3, \dots, 55\}$ ms. The weights are randomly initialized between 0 and 0.1, and only positive weights are allowed. The learning rate κ is set to 10^{-5} . The training phase will be stopped if the MSE drops below 3 or it reaches its maximum number of epochs, which is set to 150. The simulation step is 0.1 ms.

After training, the evaluation of the output is done as follows. If the output-spike is closer to the early spike-time ($t < 53.5$) then the classification of the network is “left”, and if the output-spike is closer to the late spike-time ($t > 53.5$) or there is no output-spike the classification of the network is “right”. If the output is exactly in the middle (53.5), a first order approximation of the potential around $t = 53.5$ is performed to get a real valued approximation of the spike-time (Bohte, Kok, and La Poutré, 2002).

4. RESULTS

To investigate the case under normal recording conditions, multi-input-neurons were picked up randomly from the 100 direction-related-only neurons. Instead of running all of the thousands of possible combinations, we randomly pick 20 groups of input-neurons and run the SNN analysis for each group. We calculate the average classification accuracy over the 20 groups with SDs. Every neuron has 36 spike trains (18 for “left” and 18 for “right”) in the data-set. We randomly partition half of the data-set i.e. 18 patterns into a training-set, and the other half into a test-set. Such procedures are repeated for four times, i.e., the accuracy of each group of input-neurons is calculated by averaging over the four times.

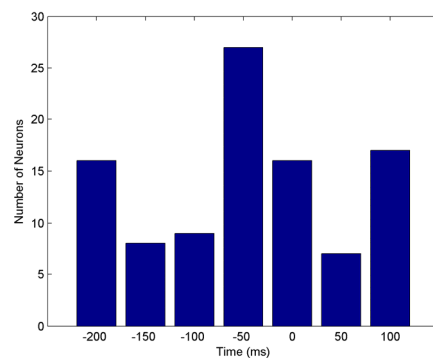


Fig. 4. Distribution of the optimal window positions for 100 direction-related-only neurons. Time zero is set at movement onset (i.e. center release).

To compare with the SNN based temporal pattern classification, we use a one-layer ANN trained with BP rule to analyze the average firing rates of the 50 ms bin spike trains. A tan-sigmoid activation function is used by the output-neuron. The “left” class is represented by an output of +1, and the “right” class is represented by -1. In the test stage, if the output value is greater than 0 the classification is “left”, and if the output value is less than or equal to 0 the classification is “right”.

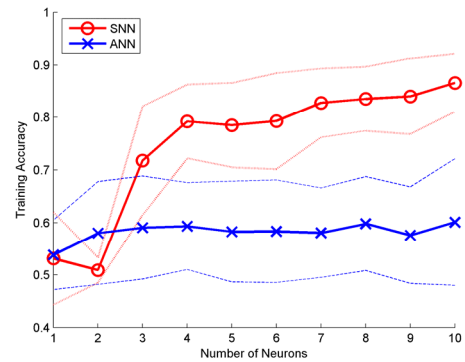
Figure 5 shows the classification accuracy averaged over the 20 groups with SDs as a function of the number of neurons. In Fig. 5(A), it is obvious that the training-set accuracy by using the SNN algorithm to analyze the timing of spike trains is much higher than that by using the ANN approach to analyze the firing rate of spike trains. The training accuracy of the SNN algorithm increases with the number of neurons and can achieve 87% by using 10 neurons, which are randomly picked from the 100 direction-related-only neurons, as input-neurons. However, the change of the training accuracy of the ANN algorithm is small, and 10 neurons can but classify 60% of the training-patterns correctly. Figure 5(B) shows the test-set accuracy of the SNN is also higher than that of ANN. And the maximum accuracy of SNN by using 10 neurons is 65%, while that of ANN is 55%.

Looking more closely at the outcome of each group of neurons, it becomes clear that the accuracy of some groups is very low and even below 50%, while that of others is very high and up to 100%. Partly this has to do with differences of the number of spikes in the 50 ms bin we studied. Some neurons have not a spike in some trials actually, while the maximum number of spikes is 11. And the spike train in the 50 ms bin consists of on average 2.24 spikes. Because of so few spikes, some groups of neurons' patterns are very difficult to classify.

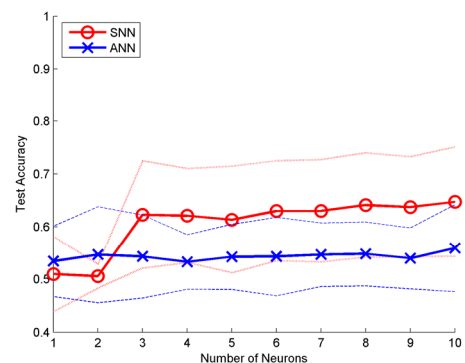
5. DISCUSSIONS

It is clear that the simple one-layer SNN can perform the recognition of the directional information in the cortical neural spike trains, which are extracted from a short 50 ms bin before the movement onset. Thus we can use the SNN algorithm to predict the target location (left or right) before the monkey releasing the central pad to reach for the indicated target. This result is significant for using the cortical neural signals to control motor prosthetic devices to bypass spinal cord injuries.

The results of the comparison between the analysis of the temporal patterns and that of the firing rates of the same spike trains demonstrate that temporal coding is a viable code for fast neural information processing. On the other hand, the average of number of spikes in the 50 ms bin is so few (2.24) that the instantaneous firing rate in the short time window is difficult to estimate correctly on the order of 10 neurons (Gautrais and Thorpe, 1998). These mean that the temporal coding is more biologically plausible than the rate coding.



(A)



(B)

Fig. 5. Comparison of classification accuracy of the SNN and the ANN algorithm versus the number of neurons. The accuracy is calculated in training-set (A) and test-set (B) respectively. The thick solid curves show the average accuracy of 20 random combination groups. The thin dotted and dashed curves show the SDs of the SNN and the ANN respectively.

The spike train data we use are from 75 ms before center release to 25 ms before center release. It is the optimal window position with which most neurons reached their maximum accuracy, see Fig. 4. However, only 27 (27%) neurons distribute in this optimal window. And we randomly pick input-neurons from all 100 direction-related-only neurons. Thus it is possible that the useful information of some neurons is not contained in the spike trains we use actually. And the neural signals sampled at 40 kHz are easy to disturbed by noise. We can record more neurons and choose those with more useful temporal data to improve the accuracy of the SNN recognition. But that is not the goal of this study, which aims to provide a new way of analyzing the temporal pattern in cortical neural spike trains.

Another thing that can be improved is the network architecture. The one-layer network in fact has only one spiking computational unit (i.e. the only output-neuron). Because of the higher computational power of multiple-layer networks (Booij, 2004; Maass, 1997c), we can use one or more hidden-layers between the input- and output-layer, that is left for future work.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced the spiking neural networks to analyze the temporal patterns in cortical neural signals. A one-layer spiking neural network trained by back-propagating the temporal error at the output was used to classify the timing of spike trains, which were recorded from the monkeys' motor cortex when they performed a two targets reaching task. We used the spike trains in a short time window (50 ms) to predict the indicated target before movement onset. The results were acceptable, although the maximum test accuracy by using 10 neurons is only 15% above the 50% expected by chance. There is certainly room for improvement, not only by using more useful temporal data of more neurons but also by improving the network architecture of SNN. That is left for future work.

The temporal pattern recognition by the SNN algorithm was compared with the firing rate analysis by the ANN approach. Our aim is not to compare the computational power between SNN and ANN algorithm, but to find a feasible way to decode the information in neural signals. The comparison results are consistent with the recent development about the neural coding that temporal coding is a viable code for fast neural information and more biologically plausible than the rate coding. On the other hand, the results indicated that the SNN approach could achieve reasonable temporal pattern classification by using a small number of neural signals and with only a single spiking computational unit. Therefore, we can conclude that the SNN algorithm is a promising method to analyze the neural signals and holds hope for designing the neuroprosthetic devices to add quality to the life of paralyzed people.

ACKNOWLEDGMENTS

This work is supported in part by National Nature Science Foundation of China under Grant 60674105, Doctor Foundation of Ministry of Education of China under Grant 20054870013, and Nature Science Foundation of Hubei province of China under Grant 2007ABA027. Authors want to express their appreciation to the Center for Neural Interface Design of the Biodesign Institute for the data provided from their experiment.

REFERENCES

- Bohte, S.M., J.N., Kok, and H. La Poutré (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, **48**, 17-37.
- Booij, O. (2004). Temporal pattern classification using spiking neural networks. Master's thesis, Intelligent Sensory Information Systems, University of Amsterdam, Available from <http://www.xs4all.nl/~obooij/study>.
- Booij, O. and H.T. Nguyen (2005). A gradient descent rule for spiking neurons emitting multiple spikes. *Information Processing Letters*, **95**, 552-558.
- Fang, H., Y. Wang, J. Huang, and J. He (2006). Multi-class support vector machines for brain neural signals recognition. The 6th World Congress on Intelligent Control and Automation, Dalian, **2**, 9940-9944.
- Gautrais, J., S. Thorpe (1998). Rate coding versus temporal order coding: a theoretical approach. *BioSystems*, **48**, 57-65.
- Georgopoulos, A.P., A.B. Schwartz, and R.E. Kettner (1986). Neuronal population coding of movement direction. *Science*, **233**, 1416-1419.
- Kempter, R., W. Gerstner, J. L. van Hemmen, and H. Wagner (1996). Temporal coding in the sub-millisecond range: model of barn owl auditory pathway. In: *Advances in neural information processing systems*, **8**, 124-130. MIT Press, Cambridge, MA.
- Lestienne, R. (1996). Determination of the precision of spike timing in the visual cortex of anaesthetised cats. *Biological Cybernetics*, **74**, 55-61.
- Maass, W. (1997a). Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, **10(9)**, 1659-1671.
- Maass, W. (1997b). Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons. In: *Advances in Neural Information Processing Systems* (M.C. Mozer, M.I. Jordan, T. Petsche (Eds)), **9**, 211-217. MIT Press, Cambridge, MA.
- Maass, W. (1997c). Fast sigmoidal networks via spiking neurons. *Neural Comput.*, **9(2)**, 279-304.
- Maass, W. and C.M. Bishop (Eds.) (1999). *Pulsed Neural Networks*. MIT Press, Cambridge, MA.
- Moore, S.C. (2002). Back-propagation in spiking neural networks. Master's thesis, University of Bath, available online: <http://www.simonchristianmoore.co.uk/back.htm>.
- Perrett, D. 1., E.T. Rolls, and W.C. Caan (1982). Visual neurons responsive to faces in the monkey temporal cortex. *Experimental Brain Research*, **47** 329-342.
- Schwartz, A.B., D.M. Taylor, and S.I. Helms Tillery (2001). Extraction algorithms for cortical control of arm prosthetics. *Current Opinion in Neurobiology*, **11**, Issue 6, 701-707.
- Schrauwen, B. and J. Van Campenhout (2004). Extending SpikeProp. Proceedings of the International Joint Conference on Neural Networks 471-476.
- Taylor, D.M., S.I. Helms Tillery, and A.B. Schwartz (2002). Direct cortical control of 3D neuroprosthetic devices. *Science*, **296**, Issue 5574, 1829-1832.
- Thorpe, S. T. and M. Imbert (1989). Biological constraints on connectionist modelling. In: *Connectionism in perspective* (R. Pfeifer, Z. Schreter, F. Fogelman-Soulié, and L. Steels (Eds.)), 63-92. Elsevier, North Holland, Amsterdam.
- Wessberg, J. et al. (2000). Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, **408**, Issue 6810, 361-365.
- Wolpaw, J.R., N. Birbaumer, D.J. McFarland, G. Pfurtscheller, and T.M. Vaughan (2002). Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, **113**, 767-791.