Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

DIRECTIONAL CLASSIFICATION OF CORTICAL SIGNALS USING A LIQUID STATE MACHINE

Jiangshuai Huang*, Huijuan Fang*, Yongji Wang*

**Key Lab. for Image Processing & Intelligent Control of Education Ministry of China*
Department of Control Science and Engineering
Huazhong University of Science and Technology
Wuhan，430074，China,(e-mail hbhldl@163.com)

Abstract: Liquid State Machine (LSM) is a newly developed computational model with many interesting properties. It has great advantages of dealing with biologic computing when compared to the traditional computational model. In this paper, the LSM was used to deal with the direction classification problem of the spike series which were distilled from the neurons in motor cortex of a monkey. In the output layer, a linear regression and back-propagation are employed as the training algorithms. Compare to outcomes of the two algorithms, it is showed that ideal classification results were derived when using BP as the training algorithm.

1. INTRODUCTION

We may have some ideas that the paralyzed people can move as they will by artificial limbs which are controlled directly by their minds. It's a gospel for all the people in need around the world. As we know, the neurons deal with the biologic information processing in terms of spikes (pulse) (Gerstner, 2002), so we may think, can we find the relationship between the man's intention and the spikes of the neurons in their brains? Truly, there is a successful trial with an owl monkey: several micro-electrodes were inserted into the monkey's motor cortex and about one hundred neurons were noted. We can note the spikes while the monkey does some arm movements. So, we can distill the spikes when monkey do the experiment we order. The subsequent work is to analyze the time series of spikes and find the relationship of them and the movement direction or trajectory. The biologic information is inherently temporal. That is to say, we don't concern the spikes or their numbers, but the specific sequence and precise occurrence in time. A proper computing tool must be adopted if we want to do a more accurate analyzing.

However, most computational models don't take the temporal aspect into account. The time-dependent temporal input must be transform to static numerical input, for example, five spikes of a time series is turned to a numeral five. Undoubtedly, there is a lot of information lost. However, the Liquid State Machine (Maass, 2002a) avoids this problem by its computational theory and particular structure. What the Liquid State Machine really is? Before introducing the LSM, we suppose that one people throw stones randomly into a pond and another one must judge how many stones are dropped and when. If he is experienced enough, he can do it because the ripples will tell him. And perhaps that's where the name of LSM comes from. When touch the liquid, some information will be left.

Maass developed the concept of Liquid State Machine and claim that it allows for universal real-time computation by feeding the input streams into a complex enough recurrent neural network. The core of the LSM is the recurrent neural network, and the parameters in the network are never being trained. Due to its dynamics and short-term memory property, it contains enough information of the input streams currently and formerly. That is to say, the influence of the input fed into the microcircuits last for a while before dying out (Maass, 2002b), and this property is called 'temporal integration'. What the randomly-made and never-be-trained networks do is to transform the irregular low-dimensional temporal input streams into high-dimensional 'liquid' state, which are regular and contain all the information of the low-dimensional (Häusler *et al*.,2003) temporal input streams. Then we apply some algorithms to transform the 'liquid' state to what we need in the output layer. This idea is similar to the support vector machine, which uses a kernel to project the input data to very high-dimensional space, in which data can be separated much more easily than in original input data space. The microcircuit is just the kernel of the LSM, but is not exactly the same due to the high-dynamics and short-term memory properties of the microcircuit. It can be used in real-time computing and most important of all, temporal series

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.1060

computing. So, it is the proper computational tool we adopt to do the neural spikes analyzing.

The LSM is not just a computational powerful model but also a biological most plausible model so far. Thus, it provides a hypothesis for computing in neural system. Some anatomical discovery find that in the cortex of many different creatures, from low-grade to high-grade, some recurrent neural structure exists which resembles the structure of microcircuit in LSM. From the evolutional point of view, this may imply that this kind of microcircuit has powerful computational ability, because after millions year of evolution, this kind of structure remains.

The whole paper is arranged as follow. After the introduction, we give the definition and structure of the LSM. Section 4 details the experiment setup and Section 5 contains the results. After this, we draw some conclusions.

2. DEFINITION OF THE LSM

The reservoir or the microcircuit, which we call to the core of the LSM, can be any type of network that has enough internal dynamics. If we use a recurrent analog neural network, it is called echo state machine (Matthias *et al.*, 2005) and it can deal with the continuous input streams. It also can be a bucket filling with water, which has been applied to a XOR (Fernando, 2003) problem because the ripples can also note the dynamics. The most commonly is, which will be using in this article, the Spiking Neural Network (Maass, 1997) whose neurons are set in a spacial array as a recurrent neural network. It can deal with the spikes of time series and is the most biologically alike. A schematic structure of LSM is shown in Fig.1 .We assume that, in order for the microcircuit to map the input stream $u(\bullet)$ to the output stream $y(\bullet)$, the microcircuit generates the internal liquid state $x^M(t)$ which is generated by u(s) for $s \le t$, and it consists of analogue values and may change continuously over time. The liquid state is not designed for a specific task. The internal state $x^M(t)$ is the output of the liquid circuit L^M fed by the input stream $u(\bullet)$. In mathematical term as:

$$x^M(t) = (L^M u)(t) \tag{1}$$

In this experiment, the circuit composes of spiking neurons. And we need another memory-less readout f^M map the $x^M(t)$ to the $y(t)$ we need.

$$y(t) = f^M(x^M(t)) \tag{2}$$

Different from the liquid circuit L^M, the readout filter f^M is design for a specific task we order. Formally, there are many kind of readout units or algorithms can be applied. Perceptrons with threshold gates, linear regression, back-propagation, P-delta rules and many others. Another obvious virtue of the LSM is that it allows multi-task in the same time (Harald *et al.*, 2007), that is to say, for the same microcircuit, different readout units with different algorithms will map the internal state $x^M(t)$ to different output $y(t)$ we need.

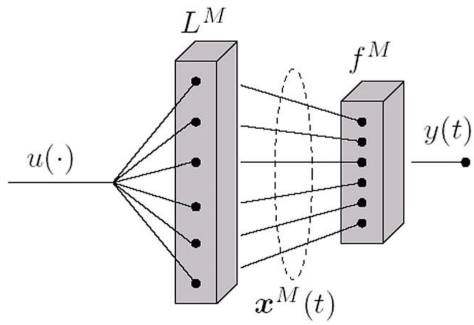


Fig. 1 Schematic structure of the LSM

Figure 1 gives a schematic structure of the LSM. Where, the neural input $u(\bullet)$ is fed into the L^M, and the $x^M(t)$ defines the internal state and the output layer f^M generates the output $y(t)$ we need (Harald *et al.*, 2007).

3. STRUCTURE OF THE LSM

In this paper, spiking neurons are used to construct the microcircuit. The internal states $x^M(t)$ which is driven by the spike trains from the motor cortex of the monkey is defined as the membrane potentials of the neurons within the microcircuit, and it is determined by all the current and former input series. The basic guideline of the LSM for solving problems is that we don't try to train the internal weights of the microcircuit but only train the weights of readout neurons. This makes the process of training much easier. In most cases, just simple algorithms can be used to adjust the weights of readout neurons.

The structure of a LSM can be organized as input layer, microcircuits and output layer. Formally, the liquid circuit is set in a 3D space. The number of neurons in x, y and z axis is n_x, n_y n_z, and they are chosen as we will, decided by the scale of the task we deal with. With the biologic resemblance some neurons of the liquid circuit are chosen as inhibitory. The probability of a connection between two neurons i and j within the microcircuit is decided by (Schwartz *et al.*, 1997):

$$p(i,j) = c \cdot \mathrm{e}^{D(i,j)/\lambda^2} \tag{3}$$

Where, $D(i,j)$ is the Euclidean distance between the two neurons in the microcircuit, λ is an important parameter that determines the number and range of the connections in the microcircuit. c is a parameter depending on the type of the connecting of the two neurons (excitatory or inhibitory), so there are four possible values for c in a liquid circuit (II,IE,EI,EE where E denotes excitatory neuron and I denotes inhibitory neuron).

In the liquid circuit the neuron commonly is chosen as standard Leaky Integrate-and-Fire model which connect each other with dynamic synapses. The spikes are changed to currents by a filter when pass a synapse to another neuron approximated by the equation as follow:

$$i(t) = \omega \cdot \mathrm{e}^{-t/\tau_{syn}} \quad (4)$$

Where, ω is a synaptic weight and τ_{syn} is synaptic time constant. The neuron accepts currents from all synapses (Abbot and Nelson, 2000) connected to it and the membrane potential alters after this equation:

$$\tau_m \frac{\partial u}{\partial t} = -u(t) + RI(t) \quad (5)$$

The τ_m is the membrane time constant which defines the speed that the voltage 'leaky' away. The neuron fires once the voltage cross the determinate threshold θ .The R is resistance constant typically set to be 1 $M\Omega$ .The membrane potential as internal state is what we need.

4. EXPERIMENTAL SETUP

The spiking data for this experiment are noted from about 324 trials in one day with a monkey. The monkey was trained to move its right arm to the eight targets labeled as 1 to 8 in a 3D imaginary cube after some signals, see Fig.3. When it finished the trail and reached the target successfully, it would get some award. We got 43 spike trains of neurons in 324 trails all together (with some trails failed). What we need to do is to predict the target the monkey reaches by analyzing the spike trains we noted from the 43 motor cortex neurons. 43 neurons' spike trains are treated as inputs and the back-propagation and linear regression algorithm are employed to train the readout layer. For the simulation of training and testing we use the neural circuit simulator CSIM .The liquid circuit is composed of Leaky Integrate-and-Fire neurons, which is shown in Fig. 2, and the parameters are shown in Table 1 and Table 2.

Table 1. parameters of the dynamic spiking synapse

W_{mean}	U_{mean}	F_{mean}	D_{mean}	U_0	R_0	Delay
0.5	0.2	0.3	1.0	0.2	1.0	0.001

Table 2. parameters of the Leaky integrate-and-Fire neuron

C_m/mF	R_m/ $M\Omega$	V_{thresh} /V	V_{reset} /V	$V_{resting}$ /V
0.03	1	0.015	0.005	0
V_{init} /V	I_{inject} /mA	I_{noise} /mA	$T_{refract}$ /s	
0.001	0.002	0	0.003	

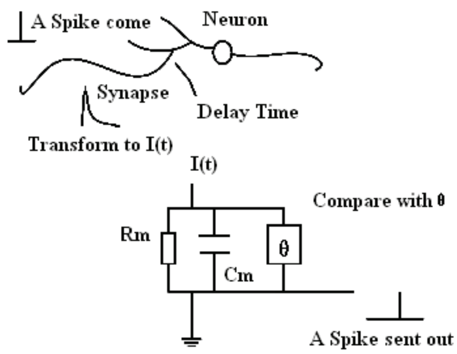


Fig. 2. Schematic diagram of leaky-integrate-and fire model

In Figure 2, a spike is transformed to I (t) first then is sent to the neuron. The model is the RC circuit. The current I (t) charges the C_m with a delay time, whose initial voltage is V_{init} , if the voltage exceeds the V_{thresh} then a spike sent out. After this the voltage comes back to V_{reset} .

The monkey moves a cursor from the center to one of eight targets in a 3D imaginary cube. It is shown as figure 3. The 3D position of the cursor is determined by the position sensor taped to the monkeys' wrist. The cursor and targets are shown with dotted outlines in the monkey's workspace, but do not physically exist.

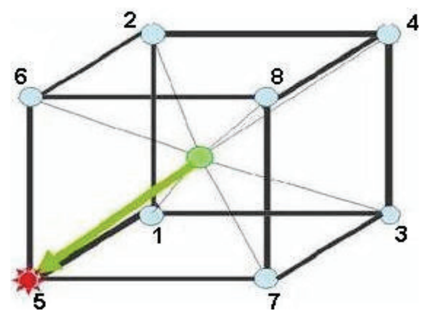


Fig.3. the eight directions adopted in this experiment.

The result is shown as figure 4. They are nine randomly chosen neurons of the liquid circuit in one training process. Formally, the membrane potential of one neuron change when a spike comes, the neurons interact within themselves. When the membrane potential crosses the threshold, it fires a spike. The membrane potentials can be chosen as the internal liquid state $x^M(t)$.

5. RESULT

Three neurons exist in readout layer because we just need to know the one of eight directions numbered as 1 to 8 (defined as 000-111). And the readout neurons just need to export an analog number to define the direction. If the

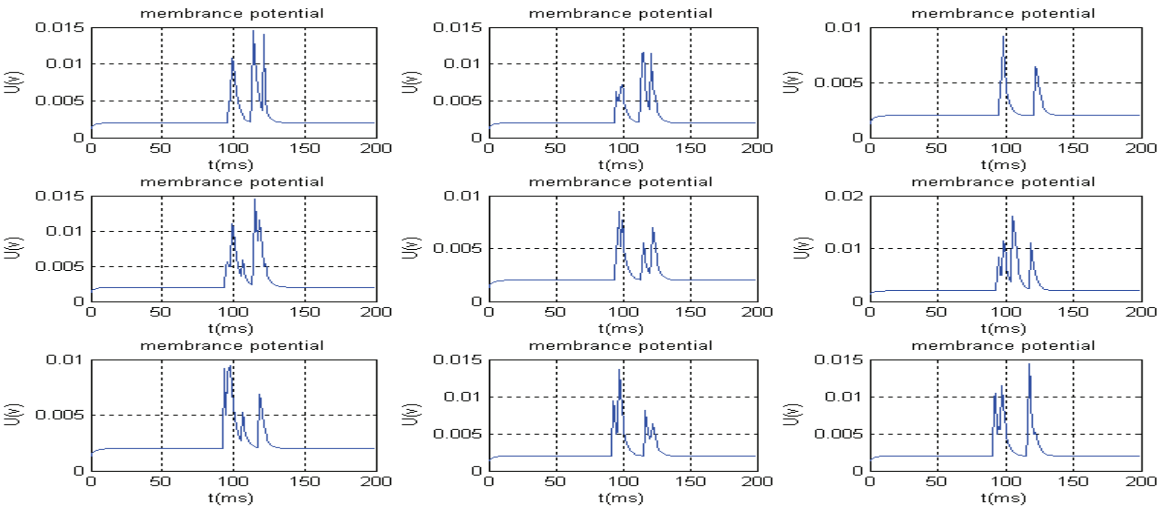
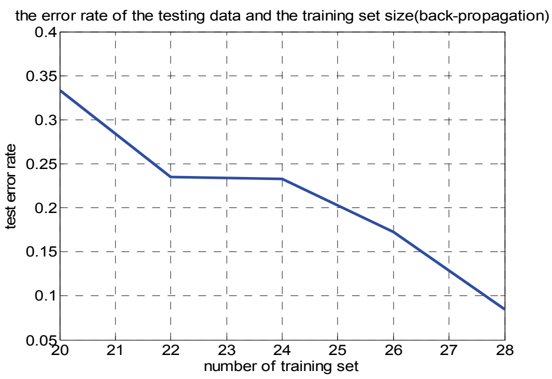


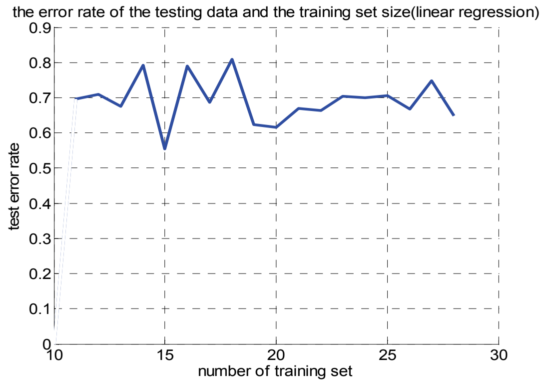
Fig.4: schematic membrane potentials changing as experiment going on

analog number is just in the threshold (we set it as 0.1) of the right direction the test data come from, we say it is a successful direction. The training algorithms are back-propagation with a hidden layer and linear regression. Due to the convergence rate and other consideration, we just distill membrane potentials of eight neurons in the liquid pool and eight potential data of each neuron as the liquid state in the processing of simulation. So one spike trains set of the 43 neurons is projected in the microcircuit to a high-dimension of 64 numerical data.

When all of the parameters are set already, simulations were carried out with the CSIM tool. Apart from the unsuccessful trials, there are 295 groups of spike trains available distilled from the monkey. Some of them are used to train the LSM and the left to test. In the microcircuit, the membrane potentials were chosen as the liquid state (see as Fig.4). As the number of training set gets larger, the error rate gets smaller. At last error rate can be less than 10% using back-propagation, (see as Fig.5), that is to say, more than 90% of testing data are successful. Before apply the LSM as the computation model, we have been trying all kinds of algorithms and computation model to this classification



(a) error of back-propagation



(b) error of linear regression

Fig.5: the testing error rate of the two algorithms as the training set get larger.

problem, except for the support vector machine, the effect of other algorithms are not so ideal. The accuracies of the classification are all around 40% to 50% (Fang, 2006). It also can be seen that the linear regression is not suitable for the biological problem for the error rate is around 70%. That is because the biologic prototype is a nonlinear model and when the spike train are projected to the high-dimensional space, they are still nonlinear, so it can't be described by the linear model.

6. CONCLUSION

In this paper we introduced a new computational model: the Liquid State Machine, and apply it to a classification problem of spike trains that recorded from the motor cortex of the monkey. The advantage of the LSM is that it can project the input data into a high-dimensional space of data so just very simple algorithms can absorb the temporal information of the input data, so it can easily classify the input information. There is a randomly-made and never-trained circuit called liquid circuit. It is the core of the LSM. It can store the nonlinear and temporal information of the input data. If there are enough neurons in the liquid circuit, no nonlinearity can hide in the input sequence.

Another virtue is that the LSM allow parallel computing with the same circuit, since the microcircuit is not task-specific. With the same input data, we can distill different information with different readout in the same time. It is also

powerful for its real-time computing ability. The liquid circuit is a recurrent neural network and is a memory-fading circuit. The liquid state $x^M(t)$ reflects the input data currently and historically. In all, the Liquid State Machine is a promising real-time and nonlinear computational model, and it will be applied to all kinds of field comprehensively in the near future.

The next work will be the trajectory simulation of the monkey's right wrist. Finally, our goal is to distill the spiking data from our motor cortex with safe equipment and find the relation between the spiking data and the trajectory of our arms or legs. So, thousands of people in the world who are in need will get help by artificial limb because they can control the artificial limb by their mind, just like the artificial limb is part of their body. We strongly think that this day will come in near future.

ACKNOWLEGMENTS

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 20050487013 and Nature Science Foundation of Hubei Province of China under Grant 2007ABA027. Authors also want to express their appreciation to Prof. Jiping He and the Center for Neural Interface Design of the Biodesign Institute at Arizona State University of USA for the data provided from their experiments.

REFERENCE

Abbot L.F.,Nelson S.B. (2000), Synaptic plasticity*:* taming the beast*, Nature Neuroscience*, **Vol. 3**:1178-1183

C.Fernando (2003). Pattern recognition in a bucket: a real liquid brain. *Lecture Notes on Computeing Science,* **2801,** pp. 588-593

D.Verstraeten,B.schrauven, Stroobandt, J. Van. Campenhout (2005), Isolated word recognition with the Liquid State Machine: A case study, *Information Processing*, **Vol. 95, No. 6,** pp.521-528

Fang huijuan, Wang Yongji, Huang J. and He J.P.(2006) Multi-class support vector machines for brain neural signals recognition, *Proc. of 6th World Congress on Intelligent Control and Automation,* **Vol. 2,** pp. 9940-9944

Gerstner W (2002). *Spiking Neuron Models,* pp. 23-25, Cambridge University Press, Cambridge

Harald B., Mark K., Alexander Leopold (2007), Movement prediction from real-world images using a liquid state machine Applied, *Intelligence*, **Vol. 26, No. 2,** pp.99-109

Häusler S.,Markram H.,Maass W(2003). Perspectives of the High Dimensional Dynamics of Neural Microcircuits from the Point of View of Low Dimensional Readout*s*, *complexity*, **Vol. 8, No. 4**, pp. 9-50

Maass W (1997), Networks of Spiking Neurons: The Third Generation of Neural Network Models*, Neural Networks* , **No. 10,** pp. 1659-1671

Maass W., Natschlager, H.Markram (2002a), Real-time computing without stable states: A new framework for neural computation based on perturbation, *Neural Computing*, **Vol.14, No.11,** pp. 2531-2560

Maass (2002b), *Liquid Computing*, *Neural Computation,* Graz University Press, Graz, Austria

Matthias S, Paul G. Ploger. (2005) Echo State Network used for motor control, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation Barcelona*, **Vol.1,** pp. 1953-1958

Miguel A.L. Nicolelis and John K. Chapin (2002). Controlling Robots with the Mind in: *Scientific American Magazine*, **Vol. 287,** pp. 46-53

Schwartz Doron Tal, Eric L (1997).Computing with the leaky integrate and fire neuron: logarithmic computation and multiplication, *Neural Computation*, **Vol. 9,** pp. 305 – 318

Vreeken J. (2004) *On real-world temporal pattern recognition using Liquid State Machines*, Utrecht University Press, Utrecht