Find Synaptic Topology from Spike Trains

Tian Ge, Wenlian Lu, *Member, IEEE*, and Jianfeng Feng

***Abstract*— Can you retrieve the underlying neuronal network topology which generates an ensemble of desired spiking trains? This is one of the key questions if one wants to implement learning in spiking neuronal network. We propose an approach to solve the question. Our approach ensures that the retrieved spiking neuronal network not only generates the desired spike timing pattern but also has a sparse topology. We analyze the solvability and robustness of our algorithm in details based on the linear programming theory. Two numerical examples are included to illustrate the approach. One example is artificial and the spike trains are generated by a leaky integrate-and-fire neuronal network. The other is from experimental data of the neuronal spikes recorded in hippocampal CA3 area. Our results demonstrate that the approach can provide us with a framework to deal with the learning problem in spiking neuronal networks.**

I. Introduction

Spiking neuronal networks are under intensive investigations due to its importance both in biology studies and in engineering applications [1]. With the advances in experimental tools, we have been able to record the spike dynamics for hundreds of thousands of neurons simultaneously [2], [3]. However, to experimentally find the network structure which generates the recorded spike data remains impossible. In theoretical studies, we know how to find the weight if the desired output is given in the framework of artificial neural networks: for example, using the BP algorithm [4]. Nevertheless, the corresponding question in the framework of spiking neuronal network is not solved, to the best of our knowledge. The difficulty lies in the fact that, even in the simplest spiking model, the leaky integrate-and-fire model [1], [5], the input-output function is not differentiable. Hence, a direct application of the gradient method, as in the BP algorithm, does not work.

Undoubtedly, the relation between the spike activities and the synaptic couplings has been attracting increasing interests from various research fields of neuroscience [6], [7]. A basic problem is how to detect the system structure from the neural activities. In the classic statistical approaches, such as correlation analysis, cluster analysis, principal component analysis (PCA), independent component analysis (ICA) [8] and causality analysis [9], [10], the system is considered as linear and analyzed without including the biophysical properties of neurons. With the biophysical knowledge, it is more realistic and efficient to utilize a specific model to describe the data and detect the system structure.

The purpose of this paper is to find out the synaptic structure only from the spike timing patterns using the leaky integrate-and-fire (LIF) neuronal network model. Here we face a few tough problems. First, the spiking timing records are insufficient to identify the parameters of a LIF network owing to synchrony. One neuron receives a sufficiently large packet of exciting synchronous spikes and emits one spike at the same time as long as the synaptic coupling is large enough. So, we can not identify the synaptic coupling in the synchronous scenario. In other words, even if the spikes are generated by 'some' LIF neuronal network, we have a class of LIF neuronal networks which can generate precisely the same spike trains. That is, this problem may be *underdetermined*. Second, it is natural that there is more or less errors to describe realistic data by a LIF model. So, there are cases that no LIF model can fit the spike timing data precisely. That is, the problem may be *overdetermined*. Finally, the spike timing is very sensitive with respect to the parameters in the LIF model. A small perturbation of parameters may lead to a totally different spike trains. That is, the problem may be *ill-conditioned*. Therefore, if we only have the spiking timing records of each neuron, any approach mentioned above for reconstructing the network structure is not feasible. Partly due to this, no existing literature is concerned with the problem in this scenario.

In this paper, we present an approach to achieve the goal and solve the problems mentioned above by the following ideas. First, in the case that the problem is underdetermined, we find the class of LIF networks which can generate the same spike trains and 'select' the sparsest one as the solution via minimizing an L_1-norm of the coupling matrix. Second, in the case that the problem is overdetermined, we add minimal number of 'hidden' neurons and assign them with selective spike timing series so that the problem is solvable. To deal with the sensitivity issue, we coarse-grain the time interval by time bins with a common and not-so-small length.

Indeed, we solve the synaptic graph topology from the spike timing records instead of identifying or detecting the graph topology. We argue that this approach can help understand the synaptic structure from the spike timing data. In addition, this approach can be generalized to other spiking neuronal models, such as LIF model with NMPA, AMPA, and GABA currents, and the Hodgkin-Huxley type models [11]. In other words, the approach serves as a learning rule equivalent to the BP in the classical artificial neural networks. One can use our approach to generate spike trains with given

Tian Ge is with the School of Mathematical Sciences and the Centre for Computational Systems Biology, Fudan University, 200433 Shanghai, China (email: tge@fudan.edu.cn).

Wenlian is with the Laboratory for Mathematical Neuroscience, RIKEN Brain Science Institute, Hirosawa 2-1, Wako-shi, 351-0198 Saitama, JAPAN, the Centre for Computational Systems Biology, Fudan University, 200433 Shanghai, China (email: wenlian@fudan.edu.cn).

Jianfeng Feng is with the Centre for Scientific Computing, Warwick University, CV4 7AL Coventry, UK, and the Centre for Computational Systems Biology, Fudan University, 200433 Shanghai, China (email: jianfeng.feng@warwick.ac.uk).

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

spike patterns, which has potential applications in control problem by spiking neuronal network [13], [14].

II. Methods and Results

We consider a network of N neurons and each neuron can be characterized by an integrate-and-fire neuronal network model. The membrane potential of the ith neuron ($1 \leq i \leq N$) is described as the following ordinary differential equations (ODE):

$$C_m \frac{dV^i(t)}{dt} = I^i_{leak}(t) + I^i_s(t) + I^i_{inj}(t), \ i = 1, \cdots, N. \quad (1)$$

Here, C_m is the membrane capacitance, $I^i_{leak}(t)$ is the current due to the passive leak of the membrane, $I^i_s(t)$ is a current describing the effect of synaptic input to the neuron from other neurons, and $I^i_{inj}(t)$ is a current injected into the neuron and supposed to be constant: $I^i_{inj}(t) \equiv I^i_{inj}$. The leak current is

$$I^i_{leak}(t) = -\frac{C_m}{\tau_m}[V^i(t) - V^i_0],$$

where V_0 is the resting potential and τ_m is the passive membrane time constant which satisfies $\tau_m = R_m C_m$, both of which are assumed to be constants. The synaptic current for a current synapse is independent of the membrane potential and is described as

$$I^i_s(t) = C_m \sum_{k=1, k\neq i}^{N} w_{ki} S_k(t),$$

where the amplitudes w_{ki} are the change in potential due to a single synaptic event. The associated charge delivered to the neuron by an excitatory ($w_{ki} > 0$) or inhibitory ($w_{ki} < 0$) synaptic input is $C_m w_{ki}$. The synaptic inputs as a train of input spikes to each synapse can be expressed as

$$S_k(t) = \sum_{t_k} \delta(t - t_k),$$

where t_k, $k = 1, 2, \cdots$, are the timing of the synaptic input spikes for the synapses and $\delta(\cdot)$ is the Dirac delta function, i.e.,

$$\delta(x) = \begin{cases} +\infty, x = 0 \\ 0, x \neq 0 \end{cases} \quad (2)$$

and $\int_{-\infty}^{+\infty} \delta(x)dx = 1$. An action potential is generated when the membrane potential reaches the spiking threshold V^i_{th} and immediately following the spike, the membrane potential is reset to its initial value $V^i(t_0) = V^i_{reset} \triangleq V^i_0$. Note that the inputs are summed linearly and output spikes are represented as δ-functions at the times when the membrane potential crosses threshold. Hence, for $t \geq t_0$, we provide the solution of the ordinary differential equations as follows:

$$\begin{aligned} V^i(t) &= V^i_0 + \frac{\tau_m I^i_{inj}}{C_m}(1 - \mathrm{e}^{-\frac{t-t_0}{\tau_m}}) + \\ &\sum_{k=1,k\neq i}^{N} w_{ki}[\sum_{t_k} \Theta(t - t_k)\mathrm{e}^{-\frac{t-t_k}{\tau_m}}] \end{aligned}$$

with

$$V^i(t) \geq V^i_{th}, V^i(t) \leftarrow V^i_{reset} = V^i_0,$$

where $\Theta(\cdot)$ is the Heaviside step function, i.e.,

$$\Theta(x) = \begin{cases} 1, x \geq 0, \\ 0, x < 0. \end{cases}$$

The problem is transformed as: with the solution expressions of the ODE that describe the membrane potentials of the neurons and the spiking times of all the neurons recorded, we want to recover all amplitudes or connection strength w_{ij} ($1 \leq i \leq N, 1 \leq j \leq N, i \neq j$) as well as the injected currents I^i_{inj} ($1 \leq i \leq N$), which can generate the same spike trains as we record.

In order to convert the continuous-time ODEs into a series of discrete-time linear programming problems, we discretise the whole recording time into t_{max} time bins with a common length and introduce a N by t_{max} $0-1$ matrix S to denote the spiking pattern, where each column records the neural activity in the particular time interval and each row corresponds to the spiking pattern of one neuron. In details, $S(i,j) = 1$ means that the ith neuron generated a spike in the jth time interval; Similarly, $S(i,j) = 0$ means that no spike was generated in the jth interval by the ith neuron.

Furthermore, the spiking times of the kth neuron are denoted as $t^1_k, t^2_k, \cdots, t^{n_k}_k$, $1 \leq k \leq N$, which is assumed to have at least two spikes for each neuron. For the ith neuron, $1 \leq i \leq N$ and for each time point t, $t^1_i < t \leq t_{max}$, if there exists an integer m such that $t = t^m_i$, $2 \leq m \leq n_i$, i.e., $S(i,t) = 1$, then

$$\begin{aligned} V^i(t) &= V^i_0 + \frac{\tau_m I^i_{inj}}{C_m}(1 - \mathrm{e}^{-\frac{t^m_i - t^{m-1}_i}{\tau_m}}) \\ &\quad + \sum_{k=1,k\neq i}^{N} w_{ki}[\sum_{\substack{t^{m-1}_i \leq t^\sigma_k < t^m_i \\ 1 \leq \sigma \leq n_k}} \mathrm{e}^{-\frac{t^m_i - t^\sigma_k}{\tau_m}}] \geq V_{th}. \end{aligned}$$

Otherwise, if $S(i,t) = 0$, then there exists an integer m such that $t^{m-1}_i < t \leq \min\{t^m_i, t_{max}\}$ and

$$\begin{aligned} V^i(t) &= V^i_0 + \frac{\tau_m I^i_{inj}}{C_m}(1 - \mathrm{e}^{-\frac{t - t^{m-1}_i}{\tau_m}}) \\ &\quad + \sum_{k=1,k\neq i}^{N} w_{ki}[\sum_{\substack{t^{m-1}_i \leq t^\sigma_k < t \\ 1 \leq \sigma \leq n_k}} \mathrm{e}^{-\frac{t - t^\sigma_k}{\tau_m}}] < V_{th}. \end{aligned}$$

Hence, for the ith neuron and for each time point t, $t^1_i < t \leq t_{max}$, we obtain an inequality depending on the state of the neuron at that time. Note that both kinds of inequalities are linear with regard to the unknowns w_{ki} and I^i_{inj}. Hence, for each neuron i, combining all the $t_{max} - t^1_i$ inequalities, we can rewrite them as a system of linear inequalities denoted as $A^i w^i \leq b^i$, where A^i and b^i are known coefficient matrix and vector respectively. $w^i = [w_{1i}, w_{2i}, \cdots, w_{Ni}, I^i_{inj}]^T$, $(\cdot)^T$ is the transpose of a vector. In details, if $S(i,u) = 1$

which corresponds to the mth spike of the ith neuron, i.e., m satisfies $u = t_i^m$, then

$$b_u^i = V_0^i - V_{th},$$

$$\begin{aligned} A_{uv}^i &= -\sum_{\substack{t_i^{m-1} \leq t_v^\sigma < t_i^m \\ 1 \leq \sigma \leq n_v}} \mathrm{e}^{-\frac{t_i^m - t_v^\sigma}{\tau_m}}, 1 \leq v \leq N, \\ A_{u,N+1}^i &= -\frac{\tau_m}{C_m}(1 - \mathrm{e}^{-\frac{t_i^m - t_i^{m-1}}{\tau_m}}). \end{aligned}$$

On the other hand, if $S(i,u) = 0$ and $t_i^{m-1} < u \leq \min\{t_i^m, t_{max}\}$, then

$$b_u^i = V_{th} - V_0^i,$$

$$\begin{aligned} A_{uv}^i &= \sum_{\substack{t_i^{m-1} \leq t_v^\sigma < u \\ 1 \leq \sigma \leq n_v}} \mathrm{e}^{-\frac{u - t_v^\sigma}{\tau_m}}, 1 \leq v \leq N, \\ A_{u,N+1}^i &= \frac{\tau_m}{C_m}(1 - \mathrm{e}^{-\frac{u - t_i^{m-1}}{\tau_m}}), \end{aligned}$$

where b_u^i is the uth component of the vector b^i and A_{uv}^i is the element in the position (u,v) of the matrix A^i.

It's widely believed and argued that the structure of a neural network is as sparse as possible to perform its functions in the most efficient way. Based on this idea, we try to find the solution of the above system under the restrictions to minimize the L_1-norm of w^i, i.e.,

$$\begin{cases} \min \|w^i\|_1 \\ s.t.\ A^i w^i \leq b^i, \end{cases} \tag{3}$$

where $\|w^i\|_1 = \sum_{k=1}^N |w_k^i|$ with w_k^i being the kth component of the vector w^i.

w^i can be decomposed as $w^i = u^i - v^i$, where $u_k^i \geq 0$ and $v_k^i \geq 0$ for any $1 \leq k \leq N$. $\|w^i\|_1 = \|u^i + v^i\|_1 \leq \|u^i\|_1 + \|v^i\|_1 = \sum_{k=1}^N (u_k^i + v_k^i) \triangleq \|\tilde{w}^i\|_1$.

Hence, defining

$$f = [1, 1, \cdots, 1]^T,$$

we can rewrite the above linear programming problem as

$$\begin{cases} \min \|\tilde{w}^i\|_1 = f^T \tilde{w}^i \\ s.t.\ [A^i - A^i] \begin{bmatrix} u^i \\ v^i \end{bmatrix} \triangleq \tilde{A}^i \tilde{w}^i \leq b^i \\ \tilde{w}_k^i \geq 0, \forall k. \end{cases}$$

Such problem can be numerically solved by the function *linprog* in Matlab. And, we can obtain the connection strength w^i as well as the injected current I_{inj}^i for each i and exactly reproduce the spiking pattern with the assumption that all the neurons can still be described by the same integrate-and-fire neuronal network model.

A. Robustness

Due to inevitability of both noises and time recording errors, it is necessary to analyze the robustness of our algorithm. We use the theory of linear programming to conduct robust analysis. More details about the theory can be found in [12].

Consider the following general linear programming:

$$L : \min\{(f,w) | Aw \leq b, w \geq 0\}.$$

Its dual problem is:

$$L^* : \max\{-(b,x) | A^T x \geq -f, x \geq 0\}.$$

We also introduce the standard notations: $ArgL$ and $ArgL^*$ are the optimal set of the problem L and L^* respectively. $optL$ and $optL^*$ are the optimal values of the problem L and L^* respectively. The following relations are evident:

$$ArgL = M \cap \{w | (f,w) = optL\}$$
$$ArgL^* = M^* \cap \{x | -(b,x) = optL^*\}$$

where $M = \{w | Aw \leq b, w \geq 0\}$, $M^* = \{x | A^T x \geq -f, x \geq 0\}$.

We denote the optimal value $optL$ of the problem L as $\tilde{f}$ which depends on A, b and f, i.e., $\tilde{f} = \tilde{f}(A,b,f)$. The values A, b and f play the role of parameters when we analyze the stability of the problem L. We suppose that y is the vector consisted of the whole parameter set.

Definition 1: A problem L is called *solvable* if $M \neq \emptyset$ and $\alpha = \inf\{(f,w) | w \in M\} > -\infty$

Definition 2: A solvable problem L is *$\tilde{f}$-stable* with respect to the parameter y in the point $y = y_0$ if its optimum function $\tilde{f}(y)$ is defined in a certain neighborhood of y_0 and is continuous in this point.

In this definition and further we assume that the space of variable y is allotted by the Euclidean norm.

The following two lemmas follow from linear programming theory [12].

Lemma 1: If the problem L is solvable, then the problem L^* is also solvable; moreover, their optimal values coincide.

Lemma 2: Solvable system L is $\tilde{f}$-stable with respect to $[A,b,f]$ if and only if both $ArgL$ and $ArgL^*$ are bounded.

Theorem 1: If the linear programming problem (3) is solvable, then the solution is robust to small perturbations of the spike timing and parameters in the LIF model.

Proof: First, we show that both $ArgL$ and $ArgL^*$ are bounded in our system L. It is trivial that the set $ArgL$ is bounded. According to Lemma 1, we have $optL = optL^*$ and is also bounded. Since b_k will take the form $\pm(V_{th} - V_0)$ and $V_{th} \neq V_0$, b_k will not be equal to zero obviously. Here b_k denotes the kth component of the vector b. Hence, $ArgL^*$ is bounded, because if not so,otherwise, $optL^*$ will be unbounded, which contradicts to the the conclusion that $optL^*$ is bounded.

According to Lemma 2, one can conclude that the linear programming problem L is $\tilde{f}$-stable with respect to $[A,b,f]$, which implies that $optL$ is continuous in a certain neighborhood of A, b and f. If a time scale to generate the spiking matrix from the experimental data is sufficiently small, due to

$$|\mathrm{e}^{-\frac{t-t_k}{\tau_m}} - \mathrm{e}^{-\frac{\hat{t}-t_k}{\tau_m}}| \le |\frac{t-\hat{t}}{\tau_m}|,$$

the perturbation of the elements of matrix A and vector b due to the error in the measurement can be as small as expected. This proves the robustness of our approach. ■

B. Solvability

In some situation, we may miss several spikes of one neuron, which causes that the linear programming problem L got from the spiking pattern to become unsolvable. Without loss of generality, we assume that the linear programming problem for the ith neuron

$$L^i : \min\{(f, \tilde{w}^i) | \tilde{A}^i \tilde{w}^i \le b^i, \tilde{w}^i \ge 0\}, 1 \le i < k$$

is solvable but L^k is unsolvable. In order to obtain a reasonable network of the neurons, we add one 'hidden' neuron to the original N neurons and resolve the system with the $N+1$ neurons and the new spiking matrix $\hat{S}$ with one more row comparing to the original matrix S. The spiking pattern of the 'hidden' neuron, i.e., the additional row in the spiking matrix, is $\hat{S}(N+1,t) = S(k,t+1)$ for all t. The following result from [12] shows the feasibility of this method.

Lemma 3: The problem L is solvable if and only if $M \neq \emptyset$ and $M^* \neq \emptyset$.

Theorem 2: Suppose the linear programming problem $L^i, 1 \le i < k$ is solvable and L^k is unsolvable with the spiking matrix S. Then, the new set of problems

$$\hat{L}^i : \min\{(f, \hat{w}^i) | \hat{A}^i \hat{w}^i \le b^i, \hat{w}^i \ge 0\}, 1 \le i \le N+1$$

deduced from the matrix $\hat{S}$, which satisfies $\hat{S}(j,t) \equiv S(j,t)$ for all $1 \le j \le N$ and t and $\hat{S}(N+1,t) = S(k,t+1)$ for all t, is solvable at least up to $\hat{L}^k$.

Proof: Since L^i, $1 \le i < k$ is solvable, there exist vectors $\bar{u}^i$ and $\bar{v}^i$ such that

$$[A^i, -A^i] \begin{bmatrix} \bar{u}^i \\ \bar{v}^i \end{bmatrix} \le b^i, 1 \le i < k.$$

The new row in the spiking matrix introduces a new column denoted as a^i in the matrix A^i. So, we have

$$[A^i \quad a^i, -A^i, -a^i] \begin{bmatrix} \bar{u}^i \\ 0 \\ \bar{v}^i \\ 0 \end{bmatrix} \triangleq \hat{A}^i \bar{w}^i \le b^i, 1 \le i < k.$$

For the kth neuron, since $\hat{S}(N+1,t) = S(k,t+1)$ holds for all t, according to the approach we get the linear programming problem, we can conclude: if $b^k_m < 0$, then $a^k_m < 0$; if $b^k_m > 0$, then $a^k_m = 0$ where the subscript m denotes the mth component of the vector. Then, there exists a sufficient large number u^k such that

$$[A^k, a^k, -A^k, -a^k] \begin{bmatrix} 0 \\ u^k \\ 0 \\ 0 \end{bmatrix} \triangleq \hat{A}^k \bar{w}^k \le b^k.$$

Hence, the set $\hat{M}^i = \{\hat{w}^i | \hat{A}^i \hat{w}^i \le b^i, \hat{w}^i \ge 0\} \neq \emptyset, 1 \le i \le k$.

On the other hand, since $f = [1, 1, \cdots, 1]^T$, we have

$$\hat{A}^i \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \ge -f.$$

This implies that $\hat{M}^{*i} \neq \emptyset, 1 \le i \le k$ holds, where $\hat{M}^{*i} = \{\hat{x}^i | \hat{A}^{iT} \hat{x}^i \ge -f, \hat{x}^i \ge 0\}$.

According to Lemma 3, we have shown that problem $\hat{L}^i$ will be solvable at least up to $\hat{L}^k$. Using this method to add new 'hidden' neurons recursively, it's easy to find that we need at most N 'hidden' neurons to detect the network of the original neurons. ■

III. Numerical Examples

A. Example 1

We generate a random graph with 300 nodes to represent a neuronal network. The connection probability is 10% where 8% of the weights are random numbers between 0 and 5 while the other 2% of the weights are random numbers between -5 and 0. We use the integrate-and-fire neuronal network model to generate a spiking pattern from this network with 1000 time bins. The initial value of the membrane potential for each neuron is randomly set between -70 and 0. Other parameters are picked as follows: $C_m = 1, I_{inj} = 3.51, \tau_m = 20$. Thus, one can verify this spiking pattern is solvable. Fig. 1 shows the recovered network using our approach. The connection rate of the recovered network is only 3.45%, much sparser than the original random graph, which shows the efficiency of optimization.

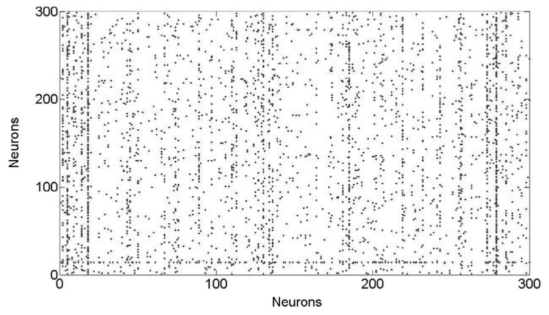

Fig. 1. The synaptic graph topology solved from the spiking pattern generated by an integrate-and-fire neuronal network model.

To test the feasibility of our approach in the overdetermined case, we generate another random graph with 150 nodes and the connection probability is 15%, where 10% of the weights are random numbers between 0 and 10 while the other 5% of the weights are random numbers between -10 and 0. Other parameters are selected as the same above. After obtaining the spiking pattern, we randomly select 100 neurons with their timings of spikes and the problem turns out to be unsolvable. 'Hidden' neurons are added according to the approach introduced in the previous section and a synaptic graph topology is successfully received as shown in Fig. 2.

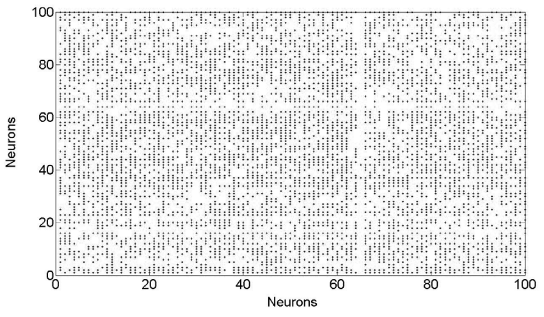

Fig. 2. The synaptic graph topology solved from the overdetermined problem.

The robustness of our proposed method is also investigated. We randomly perturb the timing of spikes by 1 time bin with the probability of 1% in the pattern with 50 neurons and compare the graph topology with and without perturbation. By repeating this experiment 100 times, we conclude that the average change rate of the topology is around 7.5%.

B. Example 2

We apply our method it for some real experimental spiking timing records. All data are downloaded from http://gaya.jp/data/, where the activities of the neurons in *hippocampal CA3* are recorded using Functional Multineuron Calcium Imaging (fMCI). Further information including the methods of experiment, data representation, the region of the neurons recorded and the experimental conditions is available on the website above.

We selecte two data files with 52 and 187 neurons respectively to test our method. Fig. 3 shows the two spiking patterns. It can be found that the first pattern with 52 neurons is quite irregular while the second pattern with 187 neurons shows synchronization.

We downsample the data with the rate 1:10 which led to the length of the time bins as 1 second and then apply our methods to the scaled pattern. Parameters are picked as follows: $C_m = 1, \tau_m = 20$. 'Hidden' neurons are needed in this case to solve a feasible structure. The numbers of hidden neurons added are 40 and 25 respectively. Fig. 4 shows the structure of the two obtained weight matrices. The first pattern shows quite random connection while the second one shows that some neurons drive the other neurons to spike.

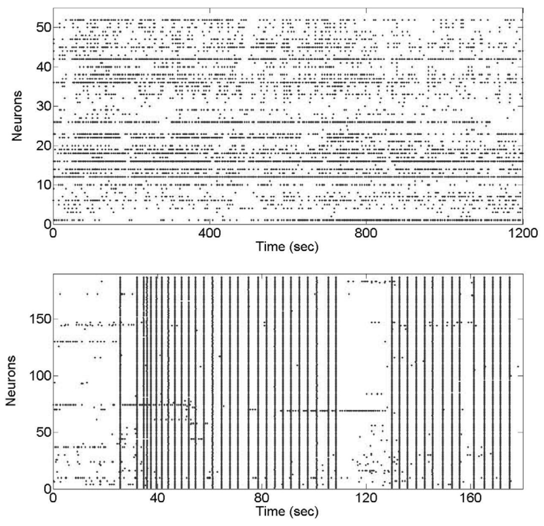

Fig. 3. Two spiking patterns.

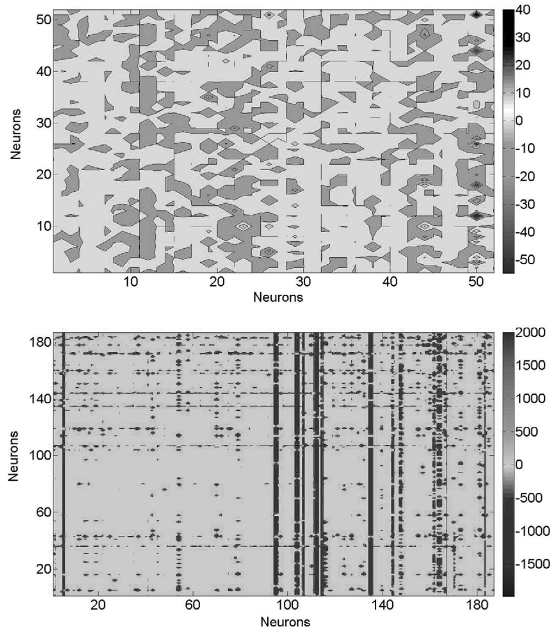

Fig. 4. Two weight matrices.

IV. Conclusions and discussions

In this paper, we propose an approach to solve the synaptic graph topology from spiking timing data within a LIF neuronal network model. We argue that our algorithm can not 'identify' the graph topology since this problem is either underdetermined or overdetermined due to synchronization or the spikes activities being partially observed. Our purpose is to find the class of LIF network topology, which can generate those spikes with precise timing, and select the sparsest one or add a number of 'hidden' neurons as small as possible to make it solvable. We utilize a coarse-graining course to conquer the ill-conditioned problem. In such a setup, our algorithm is robust against the perturbation of spiking timing. Linear programming theory guarantees the feasibility of our approach. Several numerical examples are illustrated to verify this algorithm.

Learning and memory are the most interesting and challenging issues in Neuroscience and play a critical role in engineering applications of spiking neuronal network. In the current paper, we have found a way to recover the network topology using spiking data. However, the topology is static and does not reflect the interesting dynamics of learning [6], [7]. It is certainly one of our future research topics: using sliding windows to reconstruct a dynamical network topology.

In engineering applications, how to generate a desired spiking pattern to control, for example, an arm movement, has been an active research area [13], [14]. Our approach here tells us how we can control a network topology to generate the desired spiking pattern so that we can accomplish a control task. We will test the idea as well.

Acknowledgments

We thank Dr. Yuji Ikegaya from the Graduate School of Pharmaceutical Sciences at the University of Tokyo, Japan, for providing the data in example 2. W. L. Lu is supported by the Foundation for the Author of National Excellent Doctoral Dissertation of PR China No. 200921, the National Natural Sciences Foundation of China under Grant No. 60804044, and the Shanghai Pujiang Program No. 08PJ14019, and J. F. Feng is supported by an EPSRC (UK) grant CARMEN. We also acknowledge the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the FET-OPEN grant agreement BION, number 213219.

References

[1] J. F. Feng (Ed.), *Computational Neuroscience: A Comprehensive Approach*, Boca Raton: Chapman and Hall / CRC Press, 2003.

[2] N. Takahashi, T. Sasaki, A. Usami, N. Matsuki, Y. Ikegaya, "Watching neuronal circuit dynamics through functional multineuron calcium imaging (fMCI)", *Neurosci. Res.*, vol. 58, pp. 219–225, 2007.

[3] K. M. Kendrick, Y. Zhan, H. Fischer, A. U. Nicol, X. J. Zhang, J. F. Feng, "Learning alters theta-nested gamma oscillations in inferotemporal cortex", *Nature Precedings*, hdl: 10101/ npre. 2009.3151.1., 2009.

[4] X. Yao, "Evolving Artificial Neural Networks", *Proceedings of the IEEE*, vol. 87: 9, pp. 1423–1447, 1999

[5] L. Lapicque, "Recherches quantitatives sur lexcitation electrique des nerfs traitee comme une polarization", *J. Physiol. Pathol. Gen.*, vol. 9, pp. 620-635, 1907.

[6] G. Q. Bi, M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type", *J. Neurosci.*, vol. 18, pp. 10464–10472, 1998.

[7] E. S. Nikitin, D. V. Vavoulis, et. al., "Persistent sodium current is a non-synaptic substrate for long-term memory", *Current Biology*, vol. 18, pp. 1221–1226, 2008.

[8] A. Hyvärinen, J. Karhunen, E. Oja, *Independent Component Analysis*, John Wiley & Sons, 2001.

[9] S. X. Guo, J. H. Wu, M. Z. Ding, J. F. Feng, "Uncovering interactions in the frequence domain", *PLoS Comput. Biol.*, vol. 4;5, e1000087. doi:10.1371/journal.pcbi.1000087, 2008.

[10] T. Ge, K. Kendrick, J. F. Feng, "A Unified Dynamic and Granger Causal Model Approach Demonstrates Brain Hemispheric Differences During Face Recognition Learning", *PLoS Comp. Biol.*, vol. 5(11), e1000570, 2009.

[11] A. Hodgkin, A. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve", *J. Physiol.*, vol. 117, pp. 500-544, 1952.

[12] I. I. Eremin, Theory of Linear Optimization, *V.S.P. Intl Science*, 2002.

[13] C. M. Harris, D. M. Wolpert, "Signal-dependent noise determines motor planning", *Natute*, vol. 394, pp. 780–784, 1998.

[14] J. F. Feng, H. C. Tuckwell, "Optimal control of neuronal activity", *Phys. Rev. Letts.*, vol. 91, 018101, 2003.