Detecting Communities in Networks Using Competitive Hopfield Neural Network

Jin Ding
School of Automation and Electrical Engineering
Zhejiang University of Science and Technology
Hangzhou, China 310023
Email: jding@zust.edu.cn

Yong-zhi Sun
School of Automation and Electrical Engineering
Zhejiang University of Science and Technology
Hangzhou, China 310023
Email: sunyongzhi@hotmail.com

Ping Tan
School of Automation and Electrical Engineering
Zhejiang University of Science and Technology
Hangzhou, China 310023
Email: tanp@supcon.com

Yong Ning
School of Automation and Electrical Engineering
Zhejiang University of Science and Technology
Hangzhou, China 310023
Email: Ningyong0816@126.com

***Abstract*—Community detection finds its applications in the biological networks and social networks, like predicting functional modules of proteins, recommending items to the users based on their interests, and exploring potential relationships among persons. Modularity is a widely-used criterion for evaluating the quality of the detected community structures. Due to modularity maximization is an NP-hard problem, developing the approximate algorithms with good accuracy and computational complexity is challenging and of great significance. In this paper, a novel algorithm based on competitive Hopfield neural network (CHNN for short) for maximizing modularity is proposed, where a new energy function and a two-dimensional topology is designed, and the winner-takes-all strategy for updating the outputs of neurons in each row of CHNN is adopted. Moreover, the convergence of the proposed algorithm is proved. The algorithm is capable of converging fast and achieving good modularity. Experimental results on multiple empirical and synthetic networks show the proposed algorithm can effectively and efficiently identify the community structures of the networks, and has the competitive performance compared to several other baseline algorithms for community detection.**

***Index Terms*—competitive Hopfield neural network, winner-takes-all, modularity, community detection**

I. Introduction

Community structures have been widely found in social networks and biological networks [1]–[4]. Generally speaking, community structures of a network are the subsets of nodes within which the node-node connections are dense, and between which the connections are sparse. Detecting communities of the networks has a broad spectrum of the applications [5]–[9], e.g., it can help us identify groups of people who have the same interest in the social networks, or predict the functional modules of proteins in the protein-protein interaction networks, especially when these networks are large-scale and difficult to examine by the naked eyes.

Detecting communities of the networks is a hot research topic in the domain of the network science [10], and a variety of effective and efficient methods has been proposed in the recent years [11]–[20]. One widely used method employs a criterion function, called modularity, to evaluate the quality of the detected community structures of a network [21]. The modularity is computed as a summation of the difference between the actual number of links and the expected number of links of a node-node pair over all pairs. The larger the modularity, the better the quality of the detected community structures. It is known that modularity maximization is essentially an NP-hard problem [22], so we are forced to rely on the approximate optimization approaches, which try to achieve the tradeoff between the computational complexity and the accuracy. A large number of this kind of approaches has been devised, including greedy algorithms [12], [21], [23], spectral algorithms [24]–[26], deep learning based spectral algorithm [27], Louvain algorithms [28], [29], simulated annealing [30], extremal optimization [31], genetic algorithms [32], and mathematical programming based approaches [33], [34].

Here we focus on a neural network based approach. Hopfield neural network is a form of the recurrent neural network, and it is commonly designed to represent an energy function through the proper setting of its weights. The dynamics of Hopfield neural network can lead the binary outputs of all neurons to a stable state, which corresponds to a local minimum of the energy function [35], [36]. Hence, Hopfield neural network is a promising alternative to solve the optimization problems. It has been successfully applied to polygonal approximation [37], [38], maximum clique problem [39], vector quantization in image compression [40] and image segmentation [41]. In this paper, we develop a novel modularity maximization algorithm based on competitive Hopfield neural network (CHNN for short), which adopts winner-takes-all strategy to update the outputs of the neurons in each row of CHNN rather than the traditional updating strategy that is based on the threshold of each neuron [38]. Firstly, a new energy function is defined in terms of the modularity function, and a two-dimensional topology of CHNN is designed. Secondly, the winner-takes-all strategy is adopted for updating the outputs of the neurons in each row of CHNN, which updates the output of the neuron with the maximum input to 1, and 0 for other neurons in the row. The updating order of the rows is random. Furthermore, the convergence of the proposed algorithm is proved. The proposed algorithm takes a running time of O(kn^2) for an arbitrary network, where k is the number of iterations and n is the number of nodes of the network. Experimental

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

results on multiple empirical and synthetic networks show the proposed algorithm can effectively and efficiently identify the community structures of the networks, and has the competitive performance compared to several other baseline algorithms for community detection.

The rest of the paper is organized as follows. Section II explains the mechanism of CHNN based algorithm for maximizing modularity, which includes the definition of a new energy function, the winner-takes-all updating strategy and the work flow of the proposed algorithm. In section III, the experiments are conducted on multiple empirical and synthetic networks, and the detailed discussions are made. Finally, concluding remarks are given.

II. Competitive Hopfield Neural Network for Modularity Maximization

In this section, a novel modularity maximization algorithm based on competitive Hopfield neural network (CHNN for short) is introduced. Firstly, a new energy function is defined in terms of the modularity function, and a two-dimensional topology is designed. Then, the winner-takes-all updating strategy for the neurons in each row of CHNN is described, and the work flow of the proposed algorithm is depicted. Moreover, the convergence of the algorithm is proved, and the computational complexity of the algorithm is analyzed.

A. Energy function

The energy function of a one-dimensional Hopfield neural network is defined as follows,

$$E = -\frac{1}{2}\sum_{i=1}^{t}\sum_{j=1}^{t} w_{ij}s_i s_j + \sum_{i=1}^{t}\theta_i s_i \tag{1}$$

where w_{ij} is the weight of connection between neuron *i* and neuron *j*, and s_i and s_j are the outputs of neuron *i* and neuron *j*, respectively. The weights of Hopfield neural network are symmetric, i.e., $w_{ij} = w_{ji}$, and $w_{ij} = 0$ if $i = j$. θ_i is the threshold of the neuron *i*. If the input of neuron *i* exceeds θ_i, then $s_i = 1$, otherwise, $s_i = 0$. *t* is the number of the neurons.

It is straightforward to extend the energy function of one-dimensional Hopfield neural network in equation (1) to two-dimensional, shown below,

$$E = -\frac{1}{2}\sum_{p=1}^{c}\sum_{q=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{(i,p)(j,q)}s_{i,p}s_{j,q} + \sum_{p=1}^{c}\sum_{i=1}^{n}\theta_{i,p}s_{i,p} \tag{2}$$

where *n* and *c* are the number of rows and columns of a two-dimensional Hopfield neural network, respectively.

The modularity function of a network can be expressed as follows [21],

$$Q = \sum_{i=1}^{n}\sum_{j=1}^{n}[\frac{A(i,j)}{2m} - \frac{k_i k_j}{2m \cdot 2m}]\delta(y_i, y_j) \tag{3}$$

where *A* is the adjacent matrix of the network, *A(i, j)* = 1 if there exists a connection between node *i* and node *j*, otherwise *A(i, j)* = 0. k_i and k_j are the degrees of node *i* and node *j* respectively, and y_i and y_j are the community number of node *i* and node *j* respectively. If $y_i = y_j$, $\delta(y_i, y_j) = 1$, otherwise, $\delta(y_i, y_j) = 0$. *n* and *m* are the number of nodes and edges of the network respectively.

Note that y_i and y_j in the modularity function of equation (3) is not as the binary variables s_i and s_j in the energy function of equation (1). In order to express the modularity function of a network in the form of the energy function of Hopfield neural network, the binary variables $x_{i,j}$ are introduced to express the modularity function as follows,

$$Q = \sum_{p=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{n}[\frac{A(i,j)}{2m} - \frac{k_i k_j}{2m \cdot 2m}]x_{i,p}x_{j,p} \tag{4}$$

where *A(i, j)*, k_i, k_j, *n* and *m* are the same as in the equation (3). *c* is the number of the communities of the network. $x_{i,p} = 1$ means node *i* belongs to community *p*, otherwise not.

It is clear to see equation (4) is in the similar form as equation (2). The binary output of neuron (*i, p*), $s_{i,p}$ in equation (2), can be used to indicate whether node *i* belongs to community *p* or not. In order to define a new energy function in terms of equation (4), we set $w_{(i,p)(j,q)} = w_{(j,q)(i,p)} = \frac{A(i,j)}{2m} - \frac{k_i k_j}{2m \cdot 2m}$ if $i \neq j$ and $p = q$, $w_{(i,p)(j,q)} = 0$ if $i = j$ or $p \neq q$, and the threshold of each neuron $\theta_{i,p} = 0$ in equation (2), where *i* = *j* = 1, 2, ..., *n*, and *p* = *q* = 1, 2, ..., *c*. Therefore, the energy function of a two-dimensional Hopfield neural network for modularity maximization can be expressed as follows,

$$E_Q = -\frac{1}{2}\sum_{p=1}^{c}\sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n}[\frac{A(i,j)}{2m} - \frac{k_i k_j}{2m \cdot 2m}]s_{i,p}s_{j,p} \tag{5}$$

Modularity function of equation (4) can be expressed in terms of equation (5) as follows,

$$Q = -2E_Q + \sum_{i=1}^{n}(\frac{A(i,i)}{2m} - \frac{k_i k_i}{2m \cdot 2m}) \tag{6}$$

It is worthwhile to note that the term $\sum_{i=1}^{n}(\frac{A(i,i)}{2m} - \frac{k_i k_i}{2m \cdot 2m})$ is constant, and minimizing E_Q by the dynamics of Hopfield neural network is equivalent to maximizing the modularity function *Q*.

The topology of a two-dimensional Hopfield neural network for maximizing modularity is depicted in Fig. 1. The number of rows *n* is equal to the number of nodes of the network, and the number of columns *c* is equal to the preset number of communities of the network. If a neuron of the row *i* and column *p* has the output of 1, it means the node *i* of the network belongs to community *p*, otherwise not.

B. Winner-takes-all updating strategy

The input of a neuron of a two-dimensional Hopfield neural network is defined as a summation of the weighted outputs from other neurons in the same column, denoted as $z_{i,p}$, *i* = 1, 2, ..., *n*, *p* = 1, 2, ..., *c*.

$$z_{i,p} = \sum_{j=1, j\neq i}^{n} w_{(i,p)(j,p)}s_{j,p} \tag{7}$$

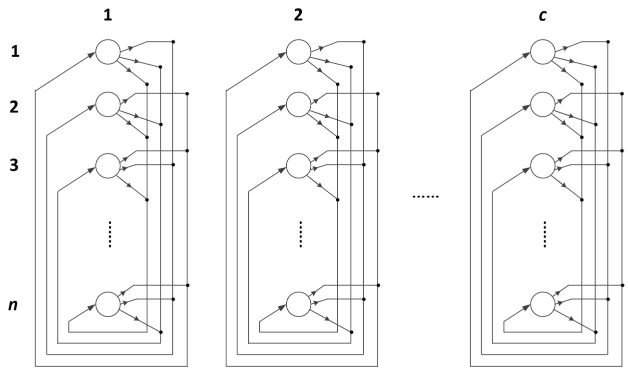

Fig. 1. **The topology of a two-dimensional Hopfield neural network for maximizing modularity.**

According to the traditional updating strategy of the neurons, if the input of a neuron exceeds its predefined threshold, the binary output of the neuron is updated to 1, otherwise 0. The binary output of a neuron $s_{i,p}$ can be expressed as follows, i = 1, 2, ..., n, p = 1, 2, ..., c.

$$s_{i,p} = \begin{cases} 1, & \text{if } z_{i,p} > \theta_{i,p} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

From the equation (8), it can be seen that, it's possible two of the neurons in the same row have outputs of 1, which indicates one node belongs to two communities.

To make one node belongs to only one community, i.e., only one neuron in each row can be updated to 1, we adopt the winner-takes-all strategy to update the outputs of the neurons in each row. The main idea of the winner-takes-all updating strategy is that, for each row of a two-dimensional Hopfield neural networks, the neuron with the maximum input is updated to 1, otherwise 0. A two-dimensional Hopfield neural network which employs the winner-takes-all updating strategy is called competitive Hopfield neural network (CHNN for short) [38]. The winner-takes-all updating strategy can be formulated as follows,

$$s_{i,p} = \begin{cases} 1, & \text{if } z_{i,p} \geq z_{i,q}, q = 1, 2, ..., p-1, p+1, ..., c \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where i = 1, 2, ..., n and p = 1, 2, ..., c.

C. Work flow of the algorithm

Given a network of n nodes and m edges, and the preset number of communities, c, the topology of CHNN is of n rows and c columns, and its weights are calculated according to the equation (5). The proposed algorithm for modularity maximization based on CHNN performs the following steps:

step 1 Initialize the binary outputs of neurons for each row, where only one neuron has the output of 1, and others have the outputs of 0. Besides, make sure in each column there exists at least one neuron which has the output of 1. The initialization process guarantees one node belongs to only one community and one community includes at least one node.

step 2 Run an iterative process to maximize the modularity. In each iteration, predefine an updating order of the rows. For each row, update the outputs of its neurons based on the winner-takes-all updating strategy.

step 3 Check out the states of all the neurons at the end of each iteration. If the outputs of all the neurons have not changed, the iterative process ends. Otherwise, go to step 2.

step 4 Obtain the community structures of the network based on the outputs of the neurons and compute the modularity of the detected community structures according to the equation (5) and (6).

It is worthwhile to note that the predefined updating order of the rows for each iteration has an impact on the modularity obtained when the iterative optimization process ends. The detailed discussion with respect to this is made in section III-A.

D. Proof of convergence

According to the work flow of the proposed algorithm, in each iteration, the rows are updated by a predefined order. And for the selected row g, the neuron which has the maximum input value is updated to 1, and other neurons are updated to 0. The energy of CHNN before updating row g can be expressed as follows,

$$E_{Q,before} = -\frac{1}{2}[\sum_{p=1}^{c} \sum_{i=1, i\neq g}^{n} \sum_{j=1, j\neq i, j\neq g}^{n} w_{(i,p)(j,p)} s_{i,p} s_{j,p} + 2\sum_{p=1}^{c} \sum_{i=1, i\neq g}^{n} w_{(i,p)(g,p)} s_{i,p} s_{g,p}] \tag{10}$$

Suppose in $E_{Q,before}$, $s_{g,h}$ = 1 and $s_{g,j}$ = 0, where j = 1, 2, ..., h-1, h+1, ..., c, which means in row g, only neuron in column h has the output of 1, and other neurons have the outputs of 0. Therefore, $E_{Q,before}$ can be rewritten as follows,

$$E_{Q,before} = -\frac{1}{2}[\sum_{p=1}^{c} \sum_{i=1, i\neq g}^{n} \sum_{j=1, j\neq i, j\neq g}^{n} w_{(i,p)(j,p)} s_{i,p} s_{j,p} + 2\sum_{i=1, i\neq g}^{n} w_{(i,h)(g,h)} s_{i,h}] \tag{11}$$

Suppose after updating row g, the neuron in row g and column h' has the output of 1, i.e., $s_{g,h'}$ = 1, and other neurons in row g have the outputs of 0. The energy of CHNN after updating row g can be expressed as follows,

$$E_{Q,after} = -\frac{1}{2}[\sum_{p=1}^{c} \sum_{i=1, i\neq g}^{n} \sum_{j=1, j\neq i, j\neq g}^{n} w_{(i,p)(j,p)} s_{i,p} s_{j,p} + 2\sum_{i=1, i\neq g}^{n} w_{(i,h')(g,h')} s_{i,h'}] \tag{12}$$

Note that the term $\sum_{i=1,i\neq g}^{n} w_{(i,h)(g,h)} s_{i,h}$ in $E_{Q,before}$ is the input of the neuron (g, h), $z_{g,h}$, and the term $\sum_{i=1,i\neq g}^{n} w_{(i,h')(g,h')} s_{i,h'}$ in $E_{Q,after}$ is the input of the neuron (g, h'), $z_{g,h'}$. According to the equation (9), $z_{g,h'}$ is equal or greater than $z_{g,h}$. So it is clear to see that, $E_{Q,after}$ is equal or less than $E_{Q,before}$. Hence, the energy of CHNN will decrease monotonically at the first several iterations, and then stay unchanged, i.e., CHNN is in a stable state and a local minimum of its energy function is found. The proposed algorithm converges.

It is worthwhile to mention that, updating one row of CHNN is equivalent to update the membership of one node of the network, and after performing the winner-takes-all updating strategy for that row, the node either switches to another community which results in higher modularity value, or keeps its membership unchanged. In each iteration, the membership of all nodes is updated. When there is no change in membership for all nodes, the community structures are identified which corresponds to a local maximum of modularity function. The optimization process for modularity maximization of the proposed algorithm is similar with the first phase of Louvain algorithms [28], [29]. In Louvain algorithms, the preset number of the communities are equal to the number of nodes, and each node is assigned with a distinct community number initially, whereas in the proposed algorithm, nodes are randomly assigned to c communities initially, where c is the manually preset number of the communities.

E. Analysis of computational complexity

Roughly speaking, updating the neurons in each row based on winner-takes-all strategy requires O(n+$\log c$) time, O(n) time for calculating the inputs of c neurons in the row and O($\log c$) time for figuring out the maximum input from the c inputs. And in each iteration, n rows are updated based on a predefined order. Therefore, the running time of the proposed algorithm for an arbitrary network is O($kn(n+\log c)$), equivalent to O(kn^2), where k is the number of the iterations, n is the number of nodes of the network, and c is the preset number of communities. Experiments on multiple empirical and synthetic network datasets find the proposed algorithm converges fast.

Notice that, for updating row i, it is unnecessary to compute the inputs of all c neurons in this row. Instead, it is safe to calculate the inputs of only c' neurons in row i, where c' is the number of the communities the neighbors of node i belong to. The average running time of the proposed algorithm becomes O($kn(\frac{2m}{n} + \frac{c'n}{c} + \log c')$). For the dense networks with $m \propto n^2$, it is equivalent to O(kn^2). And for the sparse networks with $m \propto n$, it can be judged in O($\frac{2m}{n}$) time whether the membership of the neighbors of node i is the same with that of node i itself. If so, we can skip updating row i. Hence, with respect to the sparse networks, the average running time of the proposed algorithm becomes O($k(n-k') + kk'n$), equivalent to O($kk'n$), where k' is the number of the rows which are updated using the winner-takes-all updating strategy.

III. Experimental Evaluation and Discussion

The performance of the proposed algorithm for modularity maximization based on CHNN is evaluated on three real network datasets and two synthetic network datasets with the planted communities, which are generated by the stochastic block model [42]. The node-node linking probability within the community is set to 0.1 and that between the two communities is set to 0.001. The properties of these network datasets are listed in the table I, where $\boldsymbol{n}$ is the number of nodes of the network, $\boldsymbol{m}$ is the number of edges of the network, and $\langle \boldsymbol{k} \rangle$ is the average degree of the network.

Generally speaking, the preset number of the communities for a given network (i.e., the number of columns of CHNN), c, is chosen from [2, $\sqrt{n}$]. And initially, the nodes are assigned to the communities randomly, while guaranteeing one node belongs to only one community and one community includes at least one node (i.e., only one neuron in each row has the output of 1 and at least one neuron in each column has the output of 1).

TABLE I
PROPERTIES OF THE FIVE NETWORK DATASETS

name	$\boldsymbol{n}$	$\boldsymbol{m}$	$\langle \boldsymbol{k} \rangle$	communities
Karate [43]	34	78	4.59	2
Football [4]	115	613	10.85	12
Metabolic [44]	453	2040	9	N/A
SBM-500	500	1337	5.35	10
SBM-1000	1000	4286	8.57	15

A. Effect of the updating order of rows

The work flow of the proposed algorithm in section II-C shows the predefined updating order of the rows has an impact on the dynamics of CHNN for reaching a stable state, which corresponds to a local minimum of the energy function. Here we compare two updating strategies of the rows of CHNN. One is rows are updated sequentially and the other is rows are updated randomly. The preset number of communities c is identical for both updating strategies and the simulation on each network dataset runs 100 times. In the table II, the four statistical quantities (*minimum*, *maximum*, *mean*, and *standard deviation*) on E_Q are calculated to measure the performance of the two updating strategies. It can be seen that in the network datasets of Karate, Football, SBM-500 and SBM-1000, the four statistical quantities of the random updating strategy are less or equal than that of the sequential updating strategy, and in Metabolic network dataset, the four quantities of the sequential updating strategy are less than that of the random updating strategy. The results indicate that, in general, the random updating order of the rows is slightly better than the sequential updating order of the rows of CHNN.

B. Characteristics of convergence

The characteristics of convergence of the proposed algorithm for modularity maximization based on CHNN on the

TABLE II
STATISTICAL QUANTITIES OF E_Q OBTAINED BY THE TWO UPDATING STRATEGIES

		Karate	Football	Metabolic	SBM-500	SBM-1000
random	minimum	**-0.235**	**-0.307**	-0.206	**-0.414**	**-0.396**
	maximum	**-0.188**	**-0.271**	-0.182	**-0.302**	**-0.353**
	mean	**-0.218**	**-0.292**	-0.196	**-0.368**	**-0.378**
	std[1]	**0.012**	**0.009**	0.006	**0.026**	**0.011**
sequential	minimum	**-0.235**	-0.305	**-0.209**	**-0.414**	-0.399
	maximum	-0.177	-0.256	**-0.186**	**-0.302**	-0.338
	mean	-0.216	-0.291	**-0.198**	-0.365	-0.372
	std	0.014	0.011	**0.005**	0.03	0.013

[1] *std* stands for standard deviation.

five network datasets is shown in Fig. 2. It is clear to see the proposed algorithm is capable to converge fast on all five network datasets. For example, in Karate, Football and Metabolic network datasets, the algorithm converges after about 3 iterations. And in SBM-500 and SBM-1000 network datasets, the algorithm converges after about 6 iterations. These results indicate the number of iterations, k, is independent of the number of nodes of the network, n.

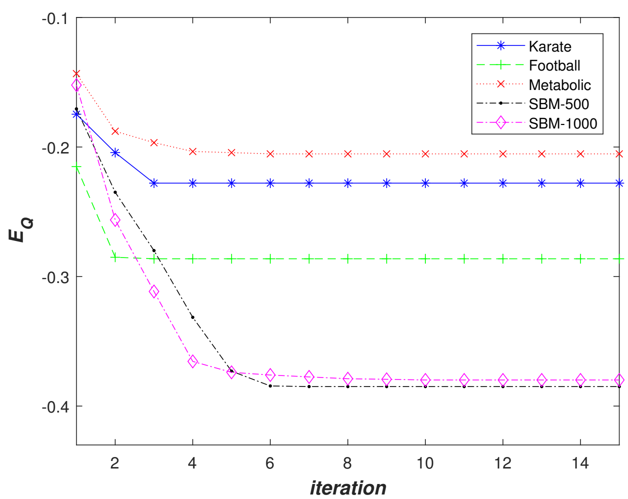

Fig. 2. **The characteristics of convergence of the proposed algorithm on five network datasets.**

C. Comparison with other baseline algorithms

We compare the proposed algorithm to three other popular algorithms for maximizing modularity--Louvain algorithm [28], spectral algorithm [24] and greedy algorithm [12]. The performance is measured by both modularity and normalized mutual information (short for NMI), shown in the table III. The proposed algorithm runs 100 times on each network dataset and the best modularity is recorded. Also, the corresponding community structure is identified and its NMI is calculated when the ground truth exists. Other three algorithms are directly called from igraph package [45].

It can be seen from the table III that the proposed algorithm has better modularity and NMI than the spectral algorithm and the greedy algorithm in all five network datasets. And when compared to the Louvain algorithm, the proposed algorithm can obtain better modularity and NMI in the Karate, SBM-500 and SBN-1000 network dataset, and the same modularity and better NMI in Football network datasets. And in the Metabolic network datasets, the Louvain algorithm has better modularity than the proposed algorithm. The community structures of Karate network with the optimal modularity [34] found by the proposed algorithm is shown in Fig. 3.

TABLE III
MODULARITY AND NMI OF THE FOUR ALGORITHMS

		Karate	Football	Metabolic	SBM-500	SBM-1000
CHNN[1]	modularity	**0.420**	**0.605**	0.433	**0.826**	**0.793**
	NMI[2]	**0.724**	**0.891**	N/A	**0.963**	**0.993**
	communities	4	10	10	10	15
Louvain	modularity	0.419	**0.605**	**0.439**	0.811	0.790
	NMI	0.587	0.890	N/A	0.913	0.976
	communities	4	10	10	15	15
spectral	modularity	0.393	0.493	0.351	0.722	0.592
	NMI	0.677	0.699	N/A	0.834	0.523
	communities	4	8	12	18	21
greedy	modularity	0.381	0.550	0.406	0.820	0.745
	NMI	0.692	0.698	N/A	0.945	0.848
	communities	3	6	11	15	10

[1] CHNN stands for the proposed algorithm;
[2] NMI stands for normalized mutual information.

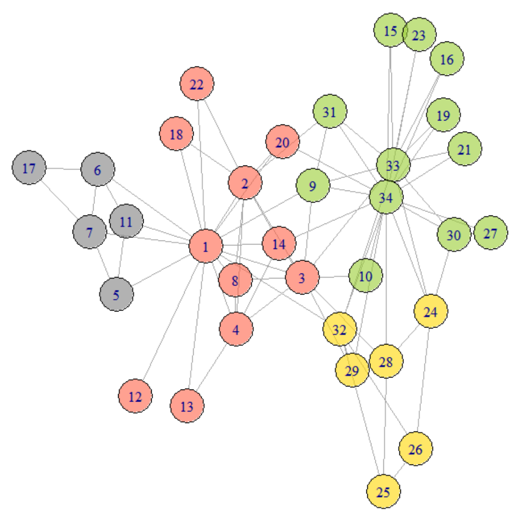

Fig. 3. **Community structures of Karate network with the optimal modularity value.**

In summary, the experiments conducted on three real network datasets and two synthetic network datasets show the effectiveness and efficiency of the proposed CHNN based algorithm for modularity maximization.

IV. Conclusion

Maximizing modularity for detecting communities of the networks is an NP-hard problem. Developing the approximate algorithms of good accuracy and computational complexity is of great significance. In this paper, a novel algorithm for modularity maximization based on competitive Hopfield neural network (CHNN for short) is proposed. Firstly, a new energy function of CHNN is designed in terms of the modularity function and its weighs are calculated based on the topology of networks of interest. Secondly, Neurons in each row are updated by the winner-takes-all updating strategy and the updating order of the rows are random. Moreover, the convergence of the proposed algorithm is proved. When it converges, a local minimum of the energy function of CHNN is found. Experiments conducted on multiple real network datasets and synthetic network datasets show the proposed algorithm is capable of converging fast and achieving good modularity, and has a competitive performance when compared to several other baseline algorithms for modularity maximization. In the future work, we would like to investigate the effect of the stochastic updating mechanism for the neurons in each row of CHNN, which is expected to help CHNN to avoid sticking into a local minimum.

Acknowledgment

This work was supported by the National Natural Science Foundation of China(51677171), Natural Science Foundation of Zhejiang Province(LY17C100001), and Education Ministry of Zhejiang Province(Y201737264).

References

[1] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics Reports*, vol. 659, pp. 1–44, 2016.

[2] M. E. Newman, "Communities, modules and large-scale structure in networks," *Nature Physics*, vol. 8, no. 1, pp. 25–31, 2012.

[3] M. E. Newman, "Mixing patterns in networks," *Physical Review E*, vol. 67, no. 2, p. 026126, 2003.

[4] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[5] A. Godoy-Lorite, R. Guimerà, C. Moore, and M. Sales-Pardo, "Accurate and scalable social recommendation using mixed-membership stochastic block models," *Proceedings of the National Academy of Sciences*, vol. 113, no. 50, pp. 14 207–14 212, 2016.

[6] J. Mcauley and J. Leskovec, "Discovering social circles in ego networks," *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 1, p. 4, 2014.

[7] J. Chen and B. Yuan, "Detecting functional modules in the yeast protein–protein interaction network," *Bioinformatics*, vol. 22, no. 18, pp. 2283–2290, 2006.

[8] S. D. Ghiassian, J. Menche, and A.-L. Barabási, "A disease module detection (diamond) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome," *PLoS computational biology*, vol. 11, no. 4, p. e1004120, 2015.

[9] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.

[10] M. E. Newman, "The structure and function of complex networks," *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.

[11] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[12] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[13] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical review E*, vol. 76, no. 3, p. 036106, 2007.

[14] A. Clauset, C. Moore, and M. E. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

[15] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[16] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[17] B. Karrer and M. E. Newman, "Stochastic blockmodels and community structure in networks," *Physical Review E*, vol. 83, no. 1, p. 016107, 2011.

[18] P. Zhang and C. Moore, "Scalable detection of statistically significant communities and hierarchies, using message passing for modularity," *Proceedings of the National Academy of Sciences*, vol. 111, no. 51, pp. 18 144–18 149, 2014.

[19] M. E. Newman, "Community detection in networks: Modularity optimization and maximum likelihood are equivalent," *arXiv preprint arXiv:1606.02319*, 2016.

[20] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 631–640.

[21] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no. 6, p. 066133, 2004.

[22] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner, "On finding graph clusterings with maximum modularity," in *Proceedings of the 33rd International Workshop on Graph-Theoretic Concepts in Computer Science*. Springer, 2007, pp. 121–132.

[23] P. Schuetz and A. Caflisch, "Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement," *Physical Review E*, vol. 77, no. 4, p. 046112, 2008.

[24] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical review E*, vol. 74, no. 3, p. 036104, 2006.

[25] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[26] X. Zhang and M. E. Newman, "Multiway spectral community detection in networks," *Physical Review E*, vol. 92, no. 5, p. 052808, 2015.

[27] L. Yang, X. Cao, D. He, C. Wang, X. Wang, and W. Zhang, "Modularity based community detection with deep learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. ACM, 2016, pp. 2252–2258.

[28] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[29] V. A. Traag, "Faster unfolding of communities: Speeding up the louvain algorithm," *Physical Review E*, vol. 92, no. 3, p. 032801, 2015.

[30] A. Medus, G. Acuña, and C. Dorso, "Detection of community structures in networks via global optimization," *Physica A: Statistical Mechanics and its Applications*, vol. 358, no. 2, pp. 593–604, 2005.

[31] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical review E*, vol. 72, no. 2, p. 027104, 2005.

[32] R. Shang, J. Bai, L. Jiao, and C. Jin, "Community detection based on modularity and an improved genetic algorithm," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 5, pp. 1215–1231, 2013.

[33] G. Xu, S. Tsoka, and L. G. Papageorgiou, "Finding community structures in complex networks using mixed integer optimisation," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 60, no. 2, pp. 231–239, 2007.

[34] G. Agarwal and D. Kempe, "Modularity-maximizing graph communities via mathematical programming," *The European Physical Journal B*, vol. 66, no. 3, pp. 409–418, 2008.

[35] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[36] J. J. Hopfield and D. W. Tank, "neural computation of decisions in optimization problems," *Biological cybernetics*, vol. 52, no. 3, pp. 141–152, 1985.

[37] J. Wang, Z. Kuang, Y. Zhou, and R.-L. Wang, "Improved stochastic competitive hopfield network for polygonal approximation," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4109–4125, 2011.

[38] P.-C. Chung, C.-T. Tsai, E.-L. Chen, and Y.-N. Sun, "Polygonal approximation using a competitive hopfield neural network," *Pattern recognition*, vol. 27, no. 11, pp. 1505–1512, 1994.

[39] G. Galán-Marın, E. Mérida-Casermeiro, and J. Muñoz-Pérez, "Modelling competitive hopfield networks for the maximum clique problem," *Computers & Operations Research*, vol. 30, no. 4, pp. 603–624, 2003.

[40] J.-S. Lin and S.-H. Liu, "A competitive continuous hopfield neural network for vector quantization in image compression," *Engineering Applications of Artificial Intelligence*, vol. 12, no. 2, pp. 111–118, 1999.

[41] K.-S. Cheng, J.-S. Lin, and C.-W. Mao, "The application of competitive hopfield neural network to medical image segmentation," *IEEE transactions on medical imaging*, vol. 15, no. 4, pp. 560–567, 1996.

[42] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *Journal of Machine Learning Research*, vol. 9, no. Sep, pp. 1981–2014, 2008.

[43] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, vol. 33, no. 4, pp. 452–473, 1977.

[44] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, pp. 651–654, 2000.

[45] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal, Complex Systems*, vol. 1695, no. 5, pp. 1–9, 2006.