International Journal of Innovative Computing, Information and Control
Volume **15**, Number **4**, August **2019**
 ISSN 1349-4198
pp. 1455–1467

MODULARITY MAXIMIZATION FOR COMMUNITY DETECTION IN NETWORKS USING COMPETITIVE HOPFIELD NEURAL NETWORK

Jin Ding, Yongzhi Sun, Ping Tan and Yong Ning

School of Automation and Electrical Engineering
Zhejiang University of Science and Technology
No. 318, Liuhe Road, Xihu District, Hangzhou 310023, P. R. China
jding@zust.edu.cn; sunyongzhi@hotmail.com; tanp@supcon.com; Ningyong0816@126.com

Received October 2018; revised February 2019

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 the NP-hard property of modularity maximization, 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 are 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.*
Keywords: Competitive Hopfield neural network, Winner-takes-all, Modularity, Community detection

1. **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 have been proposed in 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

DOI: 10.24507/ijicic.15.04.1455

the computational complexity and the accuracy. A large number of this kind of approaches have been devised, including greedy algorithms [12, 21, 23], spectral algorithms [24-26], Louvain algorithms [27, 28], simulated annealing [29], extremal optimization [30], genetic algorithms [31], and mathematical programming based approaches [32, 33].

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 [34, 35]. Hence, Hopfield neural network is a promising alternative to solve the optimization problems. It has been successfully applied to semiconductor wafer defect detection [36], polygonal approximation [37, 38], maximum clique problem [39], vector quantization in image compression [40] and image segmentation [41]. For example, in [36], the authors proposed a two-layer three-dimensional Hopfield neural network to classify the pixels of a wafer image into defective and non-defective. Under the two-layer structure, the pixels' spatial information can be effectively integrated into the energy function which describes the variance of gray level and sharp irregularity. And by updating the neurons synchronously using the winner-takes-all strategy, the energy function reaches its minimal value, and the defective regions are found. In this paper, for community detection problem, 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. It is worthwhile to note that, the main difference between the manuscript and [36] has two folds. One is the topology of the Hopfield neural network. In [36], a two-layer three-dimensional Hopfield neural network was proposed, aiming to integrate the pixels' spatial information to the energy function describing the variance of gray level and sharp irregularity. While in the manuscript, a one-layer two-dimensional Hopfield neural network is used to describe the modularity function. The other is the updating mechanism of neurons. In [36], the states of all pixels are updated synchronously in each iteration. While in the manuscript, the membership of all nodes is updated asynchronously in each iteration. Furthermore, the convergence of the proposed algorithm is proved. The proposed algorithm takes a running time of $\mathrm{O}(kn^2)$ for an arbitrary network, where k is the number of iterations and n is the number of nodes of the network. Experimental 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 2 briefly reviews the modularity maximization for community detection and Hopfield neural network. Section 3 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 4, the experiments are conducted on multiple empirical and synthetic networks, and the detailed discussions are made. Finally, concluding remarks are given.

2. Brief Review of Modularity and Hopfield Neural Network.

2.1. Modularity. Modularity is a criterion function that is widely used to evaluate the quality of the detected community structures of a network. The main idea behind this criterion is that, for each community, the larger difference of edge probability between a real network and its degree-sequence preserved random counterpart, the more significant the community is. Therefore, the modularity function of a network can be defined as follows [21],

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

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. The larger modularity indicates the better quality of the detected community structures of the network.

2.2. Hopfield neural network. The Hopfield neural network is a form of the recurrent neural network. In Hopfield neural network, each neuron is connected by all other neurons, each connection has a weight, and all the weights are symmetric, i.e., the weight of the connection from neuron i to neuron j is equal to the weight of the connection from neuron j to neuron i. The input of each neuron is defined as the summation of the weighted outputs from all other neurons, and if the input exceeds a certain threshold, the output of the neuron is 1, otherwise, the output of the neuron is 0.

With the preset weights of a Hopfield neural network, the energy function of the Hopfiled neural network can be defined as follows [34, 35],

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

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. θ_i is the threshold of the neuron i. t is the number of the neurons. Given the initial values of outputs of neurons, the dynamics of the Hopfield neural network can lead the outputs of all neurons to a stable state, which corresponds to a local minimum of the energy function in Equation (2).

3. 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.

3.1. Energy function. Comparing Equation (1) with Equation (2), it is found that y_i and y_j in the modularity function of Equation (1) are not as the binary variables s_i and s_j in the energy function of Equation (2). 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,

$$Q = \sum_{p=1}^{c} \sum_{i=1}^{n} \sum_{j=1}^{n} \left[\frac{A(i,j)}{2m} - \frac{k_i k_j}{2m \cdot 2m} \right] x_{i,p} x_{j,p} \tag{3}$$

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

Moreover, it is straightforward to extend the energy function of one-dimensional Hopfield neural network in Equation (2) to two-dimensional, shown below,

$$E = -\frac{1}{2} \sum_{p=1}^{c} \sum_{q=1}^{c} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{(i,p)(j,q)} s_{i,p} s_{j,q} + \sum_{p=1}^{c} \sum_{i=1}^{n} \theta_{i,p} s_{i,p} \tag{4}$$

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

It is clear to see Equation (3) is in the similar form as Equation (4). The binary output of neuron (i,p), $s_{i,p}$ in Equation (4), 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 (3), 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 (4), where $i = j = 1, 2, \ldots, n$, and $p = q = 1, 2, \ldots, c$. Therefore, the energy function of a two-dimensional Hopfield neural network for modularity maximization can be expressed as follows,

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

Modularity function of Equation (3) can be expressed in terms of Equation (5) as follows,

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

It is worthwhile to note that the term $\sum_{i=1}^{n} \left(\frac{A(i,i)}{2m} - \frac{k_i k_i}{2m \cdot 2m} \right)$ 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 Figure 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.

3.2. **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 $z_{i,p}$, $i = 1, 2, \ldots, n$, $p = 1, 2, \ldots, c$.

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

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

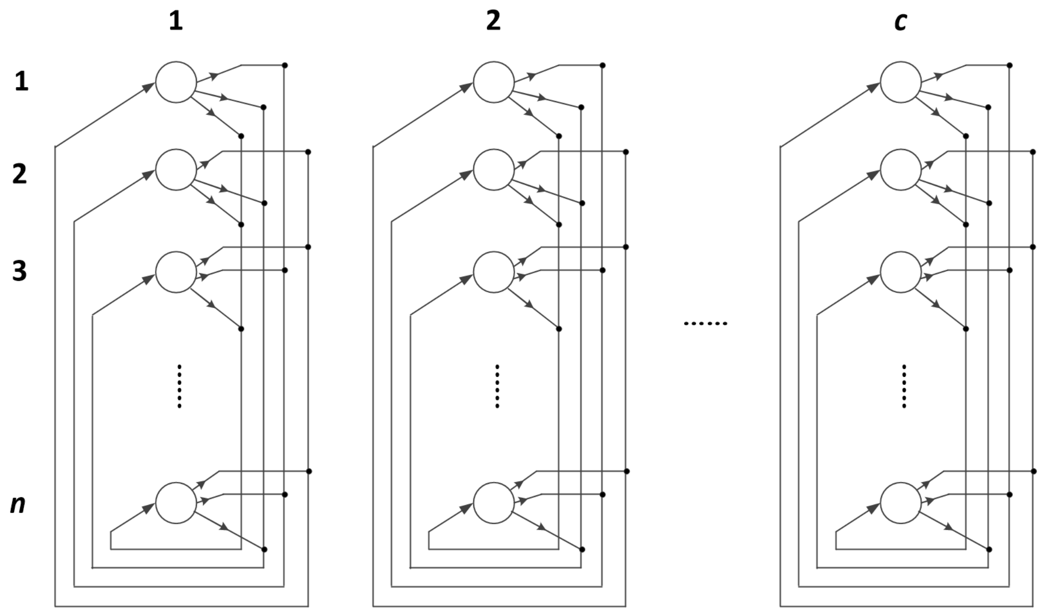

FIGURE 1. The topology of a two-dimensional Hopfield neural network for maximizing modularity

0. The binary output of a neuron $s_{i,p}$ can be expressed as follows, $i = 1, 2, \ldots, n$, $p = 1, 2, \ldots, c$.

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

From Equation (8), it can be seen that, it is 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,

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

where $i = 1, 2, \ldots, n$ and $p = 1, 2, \ldots, c$.

3.3. **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 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 Equations (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 4.1.

3.4. **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,

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

Suppose in $E_{Q,before}$, $s_{g,h} = 1$ and $s_{g,j} = 0$, where $j = 1, 2, \ldots, h-1, h+1, \ldots, c$, 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,

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

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,

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

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 Equation (9), $z_{g,h'}$ is equal to or greater than $z_{g,h}$. So it is clear to see that, $E_{Q,after}$ is equal to 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 to the first phase of Louvain algorithms [27, 28]. In Louvain algorithms, the preset number of the communities is 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.

3.5. **Analysis of computational complexity.** Roughly speaking, updating the neurons in each row based on winner-takes-all strategy requires $\mathrm{O}(n+\log c)$ time, $\mathrm{O}(n)$ time for calculating the inputs of c neurons in the row and $\mathrm{O}(\log c)$ 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 $\mathrm{O}(kn(n+\log c))$, equivalent to $\mathrm{O}(kn^2)$, 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 $\mathrm{O}\left(kn\left(\frac{2m}{n}+\frac{c'n}{c}+\log c'\right)\right)$. For the dense networks with $m \propto n^2$, it is equivalent to $\mathrm{O}(kn^2)$. And for the sparse networks with $m \propto n$, it can be judged in $\mathrm{O}\left(\frac{2m}{n}\right)$ time whether the membership of the neighbors of node i is the same as 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 $\mathrm{O}(k(n-k')+kk'n)$, equivalent to $\mathrm{O}(kk'n)$, where k' is the number of the rows which need to be updated using the winner-takes-all updating strategy.

Moreover, the parallelism can be exploited in updating each row, which could further improve the efficiency of the proposed algorithm.

4. **Experimental Evaluation and Discussion.** The performance of the proposed algorithm for modularity maximization based on CHNN is evaluated on five 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 Table 1.

Generally speaking, the preset number of the communities for a given network (i.e., the number of columns of CHNN), c, can be chosen from a certain range of values which is defined based on the size of the network datasets. 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 1. Properties of the seven 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
Collaboration [45]	12008	118489	19.74	N/A
Email [46]	36692	183831	10.02	N/A
SBM-500	500	1337	5.35	10
SBM-1000	1000	4286	8.57	15

4.1. **Effect of the updating order of rows.** The work flow of the proposed algorithm in Section 3.3 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 orders of the rows of CHNN. One is that rows are updated sequentially and the other is that rows are updated randomly. The preset number of communities c is identical for both updating orders and the simulation on each network dataset runs 100 times. In Table 2, the four statistical quantities (*minimum*, *maximum*, *mean*, and *standard deviation*) on E_Q are calculated to measure the performance of the two updating orders. It can be seen that on the network datasets of Karate, Football, Collaboration and SBM-500, all the four statistical quantities of the random updating order are less than or equal to that of the sequential updating order. On SBM-1000 and Email network datasets, there are three and two statistical quantities of the random updating order less than that of the sequential updating order, respectively. And on Metabolic network dataset, all the four quantities of the sequential updating order are less than that of the random updating order. 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.

Table 2. Statistical quantities of E_Q obtained by the two updating orders of rows

		Karate	Football	Metabolic	Collaboration	Email	SBM-500	SBM-1000
Random	minimum	**−0.235**	**−0.307**	−0.206	**−0.305**	−0.294	**−0.414**	−0.396
	maximum	**−0.188**	**−0.271**	−0.182	**−0.290**	**−0.288**	**−0.302**	**−0.353**
	mean	**−0.218**	**−0.292**	−0.196	**−0.296**	−0.292	**−0.368**	**−0.378**
	std[1]	**0.012**	**0.009**	0.006	**0.004**	**0.003**	**0.026**	**0.011**
Sequential	minimum	**−0.235**	−0.305	**−0.209**	−0.302	**−0.302**	**−0.414**	**−0.399**
	maximum	−0.177	−0.256	**−0.186**	−0.284	−0.287	**−0.302**	−0.338
	mean	−0.216	−0.291	**−0.198**	−0.295	**−0.297**	−0.365	−0.372
	std	0.014	0.011	**0.005**	0.005	0.007	0.03	0.013

[1] *std* stands for standard deviation.

4.2. **Characteristics of convergence.** The characteristics of convergence of the proposed algorithm for modularity maximization based on CHNN on the seven network datasets is shown in Figure 2. It is clear to see the proposed algorithm is capable of converging fast on all seven network datasets. For example, on Karate, Football, Metabolic, and Collaboration network datasets, the algorithm converges after about 3 iterations. And on Email, 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.

Figure 3 illustrates the percentage of nodes changing the membership in each iteration on the seven network datasets. From this figure, it can be seen that on all network datasets, at the first iteration, the majority of the nodes change their membership. And after the first iteration, the number of nodes changing their membership drops drastically, which also indicates the proposed algorithm converges fast.

4.3. **Comparison with other baseline algorithms.** We compare the proposed algorithm to three other popular algorithms for maximizing modularity – Louvain algorithm [27], spectral algorithm [24] and greedy algorithm [12]. The performance is measured by both modularity and normalized mutual information (short for NMI), shown in Table 3.

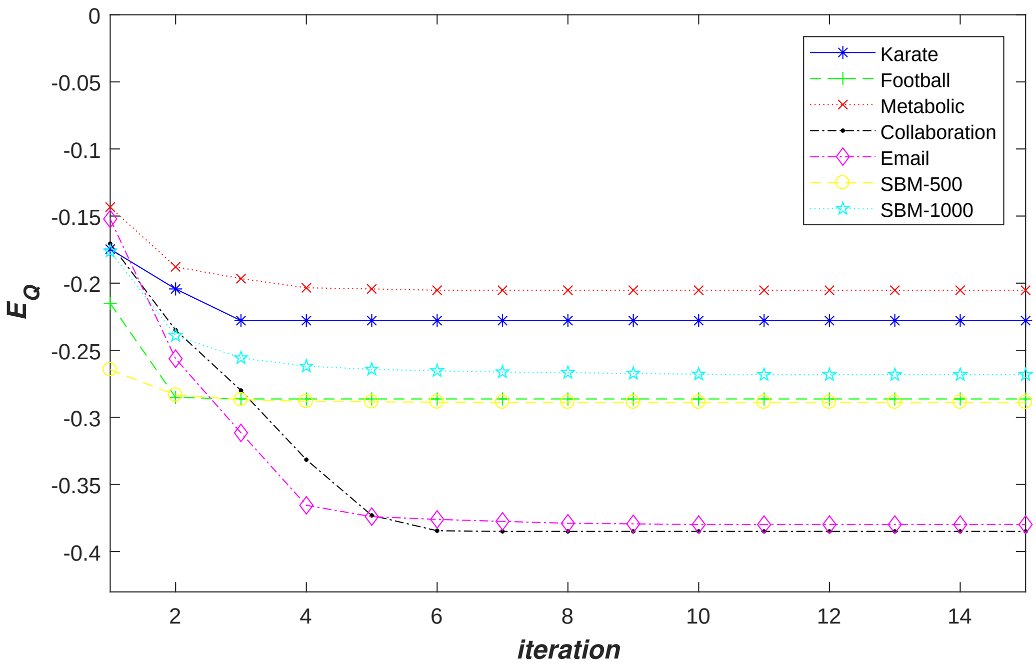

FIGURE 2. The characteristics of convergence of the proposed algorithm on the seven network datasets

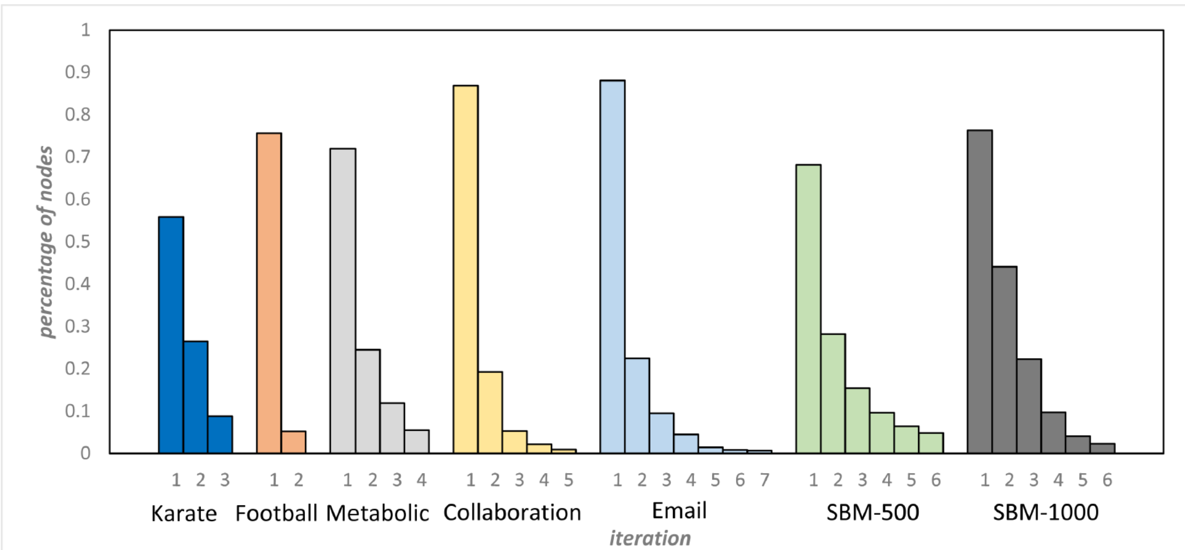

FIGURE 3. Percentage of nodes changing the membership in each iteration on the seven network datasets

The proposed algorithm runs 100 times on each network dataset and the best modularity is recorded. Also, the corresponding community structures are identified and the NMI is calculated when the ground truth exists. Other three algorithms are directly called from igraph package [47].

It can be seen from Table 3 that the proposed algorithm has better modularity and NMI than the spectral algorithm and the greedy algorithm on all seven network datasets. And when compared to the Louvain algorithm, the proposed algorithm can obtain better modularity and NMI on the Karate, Email, SBM-500 and SBN-1000 network datasets, and the same modularity and better NMI on Football network dataset. And on the Metabolic and

Collaboration network datasets, the Louvain algorithm obtains better modularity than the proposed algorithm. These results show the proposed algorithm has the competitive performance compared to other baseline algorithms for modularity maximization. The community structures of Karate network with the optimal modularity [33] found by the proposed algorithm is shown in Figure 4.

TABLE 3. Modularity and NMI of the four algorithms

		Karate	Football	Metabolic	Collaboration	Email	SBM-500	SBM-1000
CHNN[1]	modularity	**0.420**	**0.605**	0.433	0.609	**0.613**	**0.826**	**0.793**
	NMI[2]	**0.724**	**0.891**	N/A	N/A	N/A	**0.963**	**0.993**
	communities	4	10	10	617	1294	10	15
Louvain	modularity	0.419	**0.605**	**0.439**	**0.659**	0.605	0.811	0.790
	NMI	0.587	0.890	N/A	N/A	N/A	0.913	0.976
	communities	4	10	10	315	1320	15	15
Spectral	modularity	0.393	0.493	0.351	0.583	0.439	0.722	0.592
	NMI	0.677	0.699	N/A	N/A	N/A	0.834	0.523
	communities	4	8	12	289	1069	18	21
Greedy	modularity	0.381	0.550	0.406	0.583	0.518	0.820	0.745
	NMI	0.692	0.698	N/A	N/A	N/A	0.945	0.848
	communities	3	6	11	422	1062	15	10

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

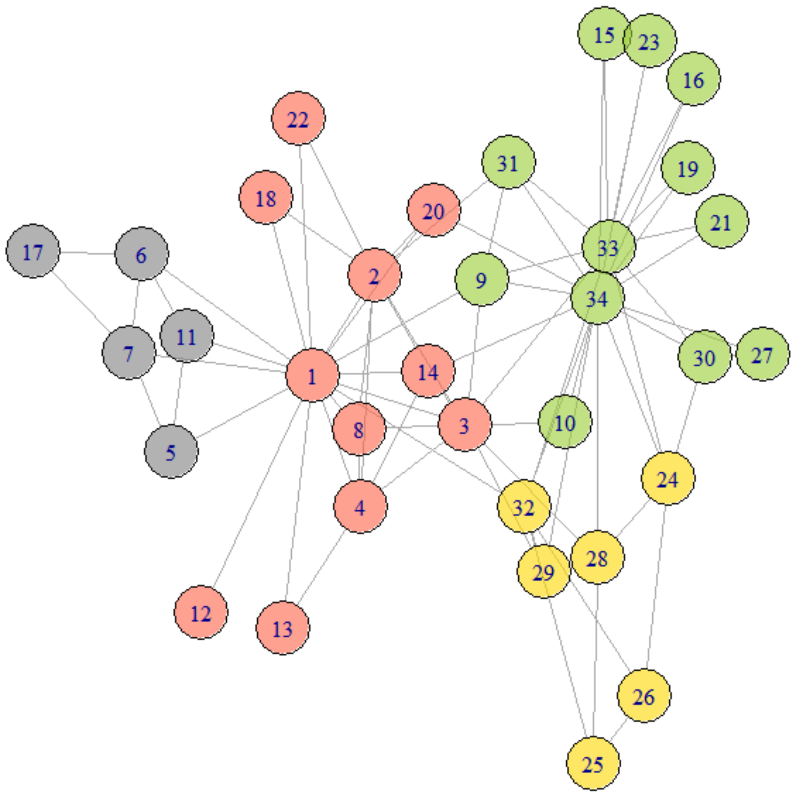

FIGURE 4. Community structures of Karate network with the optimal modularity value

4.4. **Stochastic VS winner-takes-all.** The winner-takes-all updating strategy of neurons in CHNN can lead the proposed algorithm to converge fast as shown in Figure 2, but be prone to stick into a local minimum. With respect to the winner-takes-all updating strategy of neurons, the neuron with the maximum input in a row will have the output of 1, and other neurons in the row output 0. This hints us that figuring out the neuron in each row with output of 1 in a stochastic way rather than winner-takes-all may help avoid local minima, obtaining better modularity. Here, we make a preliminary trial of designing a stochastic updating strategy of neurons in a two-dimensional Hopfield neural network to investigate its performance. In this strategy, each neuron in a row will be assigned with a probability of having output of 1, and the probability is computed according to a power law function [48] which takes the input of the neuron as its input parameter. We compare the performance of the stochastic updating strategy with the winner-takes-all updating strategy in the seven network datasets. For each strategy, the algorithm runs 100 times, and the best modularity is recorded. The results in Table 4 show, the algorithm with the stochastic updating strategy can achieve better modularity than that with winner-takes-all updating strategy on almost all the seven network datasets. Nevertheless, it is found that using the stochastic updating strategy the number of the iterations needed to achieve better modularity is much larger than that using the winner-takes-all updating strategy, which indicates the convergence speed in the stochastic updating strategy is much slower than that in the winner-takes-all strategy. In the future work, we would like to investigate how to design a stochastic updating strategy, which on the one hand can escape from a local minimum and on the other hand can keep a satisfactory convergence speed.

Table 4. Modularity and iteration obtained by stochastic and the winner-takes-all updating strategies

		Karate	Football	Metabolic	Collaboration	Email	SBM-500	SBM-1000
Stochastic	modularity	**0.4198**	**0.6046**	**0.4391**	**0.6362**	**0.6281**	**0.8258**	**0.7933**
	iteration[1]	260	163	444	210	315	484	765
Winner -takes-all	modularity	**0.4198**	0.6045	0.4329	0.6091	0.6133	0.8257	0.7932
	iteration	**3**	**3**	**6**	**7**	**11**	**6**	**9**

[1] *iteration* stands for the number of iterations needed to achieve best modularity.

5. **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 weights 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 orders 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.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (51677171), Natural Science Foundation of Zhejiang Province (LY17C1000 01), 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, 2003.

[4] M. Girvan and M. E. Newman, Community structure in social and biological networks, *Proc. of the National Academy of Sciences of the United States of America*, 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, *Proc. of the National Academy of Sciences of the United States of America*, vol.113, no.50, pp.14207-14212, 2016.

[6] J. Mcauley and J. Leskovec, Discovering social circles in ego networks, *ACM Trans. Knowledge Discovery from Data*, vol.8, no.1, 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, 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, 2004.

[12] A. Clauset, M. E. Newman and C. Moore, Finding community structure in very large networks, *Physical Review E*, vol.70, no.6, 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, 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, *Proc. of the National Academy of Sciences of the United States of America*, 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, 2011.

[18] P. Zhang and C. Moore, Scalable detection of statistically significant communities and hierarchies, using message passing for modularity, *Proc. of the National Academy of Sciences of the United States of America*, vol.111, no.51, pp.18144-18149, 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, *Proc. of the 19th International Conference on World Wide Web*, pp.631-640, 2010.

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

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

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

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

[25] M. E. Newman, Modularity and community structure in networks, *Proc. of the National Academy of Sciences of the United States of America*, 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, 2015.

[27] 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, 2008.
[28] V. A. Traag, Faster unfolding of communities: Speeding up the Louvain algorithm, *Physical Review E*, vol.92, no.3, 2015.
[29] 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.
[30] J. Duch and A. Arenas, Community detection in complex networks using extremal optimization, *Physical Review E*, vol.72, no.2, 2005.
[31] 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.
[32] 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.
[33] G. Agarwal and D. Kempe, Modularity-maximizing graph communities via mathematical programming, *The European Physical Journal B – Condensed Matter and Complex Systems*, vol.66, no.3, pp.409-418, 2008.
[34] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. of the National Academy of Sciences of the United States of America*, vol.79, no.8, pp.2554-2558, 1982.
[35] J. J. Hopfield and D. W. Tank, "Neural" computation of decisions in optimization problems, *Biological Cybernetics*, vol.52, no.3, pp.141-152, 1985.
[36] C.-Y. Chang, S.-Y. Lin and M. Jeng, Using a two-layer competitive Hopfield neural network for semiconductor wafer defect detection, *Proc. of IEEE International Conference on Automation Science and Engineering*, pp.301-306, 2005.
[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 Trans. 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, 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] J. Leskovec, J. Kleinberg and C. Faloutsos, Graph evolution: Densification and shrinking diameters, *ACM Trans. Knowledge Discovery from Data (TKDD)*, vol.1, no.1, 2007.
[46] J. Leskovec, K. J. Lang, A. Dasgupta and M. W. Mahoney, Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters, *Internet Mathematics*, vol.6, no.1, pp.29-123, 2009.
[47] G. Csárdi and T. Nepusz, The igraph software package for complex network research, *InterJournal, Complex Systems*, vol.1695, no.5, pp.1-9, 2006.
[48] S. Boettcher and A. Percus, Nature's way of optimizing, *Artificial Intelligence*, no.1, pp.275-286, 2000.