On Permutation Symmetries of Hopfield Model Neural Network

JIYANG DONG*, SHENCHU XU, ZHENXIANG CHEN and BOXI WU

Department of Physics, Xiamen University, 361005 Xiamen, P. R. China

Discrete Hopfield neural network (DHNN) is studied by performing permutation operations on the synaptic weight matrix. The storable patterns set stored with Hebbian learning algorithm in a network without losing memories is studied, and a condition which makes sure all the patterns of the storable patterns set have a same basin size of attraction is proposed. Then, the permutation symmetries of the network are studied associating with the stored patterns set. A construction of the storable patterns set satisfying that condition is achieved by consideration of their invariance under a point group.

1. INTRODUCTION

Symmetry plays an important role in unification and classification of physical systems [1], *e.g.*, crystals by their point group symmetry or the atomic nucleus by the isospin symmetry. Permutation symmetry may also be important, a prominent example is the replica symmetry breaking in the spin glass theory, where a classification of the mean field theoretical solutions is achieved by consideration of their invariance under subgroups of the symmetric group [2].

Another object for which the study of symmetries proves to be very fruitful is the area of neural networks. Global properties of an individual network can be found from symmetry considerations of the invariance group of the specific pattern set stored by some learning rules [3, 4].

Here, the permutation symmetries on DHNN are associated with that on the stored patterns set. The organization of this paper is as follows. Section 2 gives a simple introduction on the discrete Hopfield neural network. In Section 3, the DHNN is studied by performing permutation operations on the

---

* Corresponding author. e-mail: jydong@jingxian.xmu.edu.cn

synaptic weight matrix, and some interesting properties are found. The permutation symmetries of the network are studied associating with the stored patterns set. A condition that makes sure all patterns of the storable patterns set have a same basin size of attraction is proposed. In Section 4, a construction of the storable patterns set with same basin size of attraction is achieved by consideration of their invariance under a point group. We give some discussions and a conclusion in Section 5.

2. DISCRETE HOPFIELD NEURAL NETWORK

Discrete Hopfield neural network [5] (DHNN) was proposed mainly as an associative memory model. It is a dynamical system uniquely defined by (W, θ). Here W is a symmetric zero-diagonal real weight matrix, where element W_{ij}, denotes the weight of the connection from neuron j to neuron i ($i, j = 1, 2, \ldots, N$ for N-neurons network), θ is a vector of dimension N, where component θ_i denotes the threshold value of neuron i. The state of neuron i is denoted by $X_i(t)$ and it is either "0" or "1". The state vector of the DHNN at time t is denoted by $X(t) = [X_1(t), X_2(t), \ldots, X_N(t)]^t$. The state of neuron i at time $t+1$ is computed by

$$X_i(t + 1) = \text{step}\left(\sum_{j=1}^{N} W_{ij} X_j(t) - \theta_i(t) \right) \quad \forall i \quad (1)$$

here,

$$\text{step}(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0. \end{cases}$$

Without losing of generality, we let $\theta_i = 0$, $i = 1, 2, \ldots, N$ for the sake of simplicity in this paper.

A state $X = (X_1, X_2, \ldots, X_N)^t$ of the network is called stable if

$$X_i = \text{step}\left(\sum_{j=1}^{N} W_{ij} X_j - \theta_i \right) \quad (i = 1, 2, \ldots, N) \quad (2)$$

i.e., if the state of the network never changes as a result of evolution. We usually call the stable state as the attractor of the network.

Let X be a stable state of the network, if $X^m (t) = X$ for any integer t (including $t \to \infty$), we say that $X^m(0)$ is attracted by X, and the state set $\{X^u(0) | X^u(t) = X, \ u = 1, 2, \ldots, \Lambda\}$, which are attracted by X, is the attractor basin of X.

In associative memory, the weight matrix W is generally designed with Hebbian algorithm. That is, to store M patterns $X^k = \{0, 1\}^N$, $k = 1, 2, \ldots$, M, in a N-neurons network, the synaptic weight matrix is

$$W_{ij} = \frac{1}{M} \sum_{k=1}^{M} (2X_i^k - 1)(2X_j^k - 1) - \delta_{ij}$$

$$i, j = 1, 2, \ldots, N \quad (3)$$

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}, \quad \text{which ensures } W_{ij} = 0.$$

Here, we convert firstly the mono-polar pattern into bipolar one, then define the synaptic weight matrix of the network as the summation of the out-products of the corresponding bipolar patterns. So, let $V^k = 2X^k - 1$, there is,

$$W = \frac{1}{M} \sum_{k=1}^{M} V^k (V^k)^t - I, \quad (4)$$

I is identify matrix.

It can be easily verified that either $W_{ii} = 0$ or 1 has the same properties discussed in this paper. But we prefer the model of DHNN that remains $W_{ii} = 0$ for the two reasons: (1) It is proved that DHNN with symmetric zero-diagonal real weight matrix will evolve to a stable state or a limited cycles whose period is 2 for any initial state in synchronous mode [6]. (2) When $W_{ii} = 0$, the network is easy to be implemented for the applications.

3. PERMUTATION SYMMETRY OF DHNN

The permutation can be expressed in terms of matrix called permutation matrix, which is a

double-stochastic square matrix of $(0, 1)$ and have only one 1 in each row or column. For example, consider $q = (ij)$, which may conveniently be written as:

$$
q = \begin{bmatrix}
1 & \cdot & \cdots & \cdot & & & & & 0 \\
 & \ddots & \cdot & & & \cdot & & & \\
 & 1 & \cdot & & \cdots & & \cdot & & \\
 & 0 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\
 & \cdot & 1 & & & & \cdot & & \\
\vdots & & \cdot & \ddots & & & & & \vdots \\
 & \cdot & & & 1 & \cdot & & & \\
 & 1 & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot \\
 & & & & & 1 & & & \\
 & & & & & & & \ddots & \\
0 & & & \cdots & & & & & 1
\end{bmatrix}
\begin{matrix} \\ \\ \\ i \\ \\ \\ \\ j \\ \\ \\ \end{matrix}
$$

$$(5)$$

with column labels i and j at top.

Let $A_{N \times 1}$ be a column vector and $C_{N \times N}$ be a $N \times N$ matrix, then the action of permutation $q = (ij)$ on them would be

$qA_{N \times 1}$, exchange the value of column i with column j.

$qC_{N \times N}q^t$, exchange the value of column i with column j and row i with row j.

Where q^t is the transposed matrix of q.

An important property of the permutation matrix used in this paper is

$$qq^t = I, \quad i.e., \quad q^t = q^{-1}, \quad (6)$$

where q^{-1} is the inverse matrix of q.

When the synaptic weight matrix W of a network H is operated by a random permutation matrix q, the relation between H and the new network $\hat{H}$ whose synaptic matrix $\hat{W} = qWq^t$ can be described by the following theorem.

THEOREM 1   *Let H and $\hat{H}$ be two discrete Hopfield neural networks whose synaptic weight matrices are W and $\hat{W}$ respectively such that $\hat{W} = qWq^t$. Let the*

set of attractors of the network H be denoted as $\Psi_H = \{V^u, u = 1, 2, \ldots, M\}$. Let the attractor basin of H be denoted as $\Omega_H^u = \{X^{uv}, v = 1, \ldots, \Lambda\}$. Then the set of attractors of the network $\hat{H}$ must be $\psi_{\hat{H}} = \{\hat{V}^u = qV^u, u = 1, 2, \ldots, M\}$ and the attractor basin of $\hat{V}^u$ must be $\hat{\Omega}_{\hat{H}}^u = \{\hat{V}^{uv} = qV^{uv}, v = 1, \ldots, \Lambda\}$.

Proof There are two parts to this theorem.

(1) Let an attractor of the network H denoteed as V^u, i.e., $V^u = \text{step}(WV^u)$. It can be rewritten by $V^u = \text{step}(WV^u) = F_W(V^u)$. Then,

$$F_{\hat{W}}(\hat{V}^u) = F_{\hat{W}}(qV^u) = \text{step}(\hat{W}qV^u)$$
$$= \text{step}(qWq^tqV^u)$$

$\because q^t q = 1$ and $\text{step}(qx) = q\,\text{step}(x)$ where x is any $N \times 1$ matrix.

$$\therefore F_{\hat{W}}(\hat{V}^u) = F_{\hat{W}}(qV^u) = q \cdot \text{step}(WV^u)$$
$$= qV^u = \hat{V}^u.$$

So, $\hat{V}^u = qV$ must be an attractor of the network $\hat{H}$.

Assume there is an attractor $\hat{V}^m$ of the network $\hat{H}$, i.e., $F_{\hat{W}}(\hat{V}^m) = \text{step}(\hat{W}\hat{V}^m) = \hat{V}^m$, and $\hat{V}^m \notin \Psi_{\hat{H}} = \{\hat{V}^u = qV^u\}$, thus $q^t\hat{V}^m \notin \Psi_H = \{V^u\}$. Otherwise, there is,

$$F_W(q^t\hat{V}^m) = \text{step}(Wq^t\hat{V}^m) = \text{step}(q^t\hat{W}qq^t\hat{V}^m)$$
$$= q^t\text{step}(\hat{W}\hat{V}^m) = q^t\hat{V}^m$$

i.e., $q^t\hat{V}^m \in \Psi_H = \{V^u\}$. It is opposite.

So, only $\Psi_{\hat{H}} = \{\hat{V}^u = qV^u, u = 1, 2, \ldots, M\}$ is the set of attractors of the network $\hat{H}$.

(2) Let $X(t)$ and $\hat{X}(t)$ respectively denote the trajectories of the dynamical system of H and $\hat{H}$ from the initial state $X(0)$ and $\hat{X}(0)$. Consider $\hat{X}(0) = qX(0)$, thus,

$$\hat{X}(1) = F_{\hat{W}}(\hat{X}(0)) = F_{\hat{W}}(qX(0))$$
$$= \text{step}(\hat{W}qX(0)) = \text{step}(qWq^tqX(0))$$

$\because q^t q = 1$ and $\text{step}(qx) = q \, \text{step}(x)$ where x is any $N \times 1$ matrix.

$$\therefore \hat{X}(1) = \text{step}(qWX(0))$$
$$= q \, \text{step}(WX(0)) = qX(1)$$

Similarly, if $\hat{X}(t) = qX(t)$, then $\hat{X}(t+1) = qX(t+1)$. So, if $\hat{X}(0) = qX(0)$, $\hat{X}(t+l) = qX(t+1)$ exists for any integer t.

Let X^{uv} denote an initial state converged to the attractor of V^u in the network H, *i.e.*, $V(0) = X^{uv}$ and $V(t) = F_W^{(t)}(X^{uv}) =_{t \to \infty} V^u$.
If $\hat{V}(0) = qV(0) = qX^{uv}$, we have

$$F_{\hat{W}}^{(t)}(qX^{uv}) = \hat{V}(t) = qV(t) = F_W^{(t)}(qX^{uv})$$
$$=_{t \to \infty} qV^u$$

Then, the state in $\hat{\Omega}_{\hat{H}}^u = \{\hat{V}^{uv} = qV^{uv}, \ v = 1, \ldots, \Lambda\}$ must converge to the attractor of $\hat{V}^u$ in the network $\hat{H}$.

Assume there is an initial state $\hat{X}$ in the state space of $\hat{H}$ can converge to $\hat{V}^u = qV^u$ (*i.e.*, $\hat{V}(t) = F_{\hat{W}}^{(t)}(\hat{X}) =_{t \to \infty} \hat{V}^u = qV^u$) and $\hat{X} \notin \hat{\Omega}_H^u = \{\hat{V}^{uv} = qV^{uv}\}$, thus $q^t \hat{X} \notin \Omega_H^u = \{X^{uv}\}$.

Otherwise, similarly processed as above, let $V(t)$ and $\hat{V}(t)$ respectively denote the trajectories of the dynamical system H and $\hat{H}$ from the initial state $V(0)$ and $\hat{V}(0)$, and consider $V(0) = q^t \hat{V}(0)$. Thus,

$$V(1) = F_W(V(0)) = F_W(q^t \hat{V}(0))$$
$$= \text{step}(Wq^t \hat{V}(0)) = \text{step}(q^t \hat{W} q q^t \hat{V}(0))$$
$$= q^t \text{step}(\hat{W} \hat{V}(0)) = q^t \hat{V}(0).$$

Similarly, if $V(t) = q^t \hat{V}(t)$ then $V(t+1) = q^t \hat{V}(t+1)$.

So, if $V(0) = q^t \hat{V}(0)$, $V(t) = q^t \hat{V}(t)$ is existence for any integer t.

$\because \hat{V}(0) = \hat{X}$ and $\hat{V}(t) = F_{\hat{W}}^{(t)}(q^t \hat{X}) =_{t \to \infty} \hat{V}^u = qV^u$, and we let $V(0) = q^t \hat{V}(0) = q^t \hat{X}$.

$$\therefore F_W^{(t)}(V(0)) = F_W^{(t)}(q^t \hat{X}) = V(t) = q^t \hat{V}(t)$$
$$= q^t F_{\hat{W}}^{(t)}(\hat{X}) =_{t \to \infty} q^t \hat{V}^u$$
$$= q^t q V^u = V^u$$

i.e., $q^t \hat{V}(0) \in \Omega_H^u = \{X^{uv}\}$ which is opposite to $q^t \hat{X} \notin \Omega_H^u = \{X^{uv}\}$. This means only $\hat{\Omega}_{\hat{H}}^u = \{\hat{V}^{uv} = qV^{uv}, v = 1, \ldots, \Lambda\}$ can be attracted by $\hat{V}^u$ in the network $\hat{H}$.

Thus the theorem is proven. ∎

Theorem 1 describes the variance of the network after performed a random permutation operation. However, there exist some permutations so that the network keeps invariance after permuted by them. Those permutations are the permutation symmetries of the network.

Before studying the permutation symmetry of the network, we introduce firstly some definitions.

DEFINITION 1 The symmetric permutation of the patterns set $\Psi = \{X^u, u = 1, 2, \ldots, M\}$ is defined as the permutation q which maps patterns set Ψ to itself, $\Psi' = \{qX^u, u = 1, 2, \ldots, M\} = \Psi$. The symmetric permutation of a network with the synaptic weight matrix W is defined as the permutation g such that $gWg^t = W$.

DEFINITION 2 A patterns set $\Psi = \{X^u, u = 1, 2, \ldots, M\}$ is storable [8] if all M sample patterns can be the stable states of some network with the synaptic weight matrix W which is a zero-diagonal real symmetric matrix.

DEFINITION 3 The Hamming weight $w(V)$ of the N-dimension binary vector $V = \{1, 0\}^N$ is defined as the number of component 1 in the vector $w(V) = \sum_{i=1}^{N} V_i$.

In Hebbian neural network, the permutation symmetries of the network can be studied by the stored patterns set. We establish the following theorem to show the relationship between them.

THEOREM 2 *Let G be the symmetric permutations set of the sample patterns set $\Psi = \{X^u \mid X^u = (X_1^u, X_2^u, \ldots, X_N^u)^t, \ u = 1, 2, \ldots, M\}$, i.e., $G = \{q/q \Psi = \Psi\}$, where $X^u = \{0, 1\}^N$ and q is a permutation matrix. For $\forall X^i, \forall X^j \in \Psi$, if there exists at least one permutation $q \in G$ such that $qX^i = X^j$, we have*

(1) *All patterns in Ψ have the same Hamming weight.*
(2) *All of the permutations in G are the symmetric permutations of the network with the synaptic*

weight matrix W which is stored the patterns of Ψ by Hebbian rule.

(3) *All patterns in Ψ are stable in the network (W) and have same size of attractor basins if only one of the patterns is stable in the network.*

(4) *G is a group which is a subgroup of S_n consisting of all permutations of the elements from the set $\{1, 2, \ldots, n\}$.*

Proof

(1) Let X^u and X^m belong to patterns set Ψ, and their Hamming weight be K and L respectively. Let the permutation relation of them be q such that $qX^u = X^m$, $q = \begin{pmatrix} 1 & 2 & \ldots & N \\ i_1 & i_2 & \ldots & i_N \end{pmatrix}$, which replace the component order of X^u $(1, 2, \ldots, N)$ with $(i_1, i_2, \ldots, i_N)$, *i.e.*,

$$X^m = qX^u = q(X_1^u, X_2^u, \ldots, X_N^u)^t$$
$$= (X_{i_1}^u, X_{i_2}^u, \ldots, X_{i_N}^u)^t.$$

Assume $K \neq L$, *i.e.*, $w(V^m) \neq w(V^u)$. Otherwise, there is,

$$w(V^m) = w(qV^u) = \sum_{i_k=1}^{N} V_{i_k}^u.$$

Because the summation of the components of a vector immaterial to their orders. So,

$$w(V^m) = \sum_{i_k=1}^{N} V_{i_k}^u = \sum_{i=1}^{N} V_i^u = w(V^u),$$

i.e., $K = L$, which is opposite to the assumption.

Therefore, all patterns in Ψ have the same Hamming weight.

(2) To store the patterns of Ψ in a network, the synaptic weight matrix is

$$W_{ij} = \frac{1}{M} \sum_{u=1}^{M} (2X_i^u - 1)(2X_j^u - 1) - \delta_{ij}$$

For convenience, we let $V^u = 2X^u - 1$, then

$$W = \frac{1}{M} \sum_{k=1}^{M} V^k (V^k)^t - I$$
$$= \frac{1}{M} [V^1(V^1)^t + V^2(V^2)^t + \cdots + V^M(V^M)^t] - I$$

For $\forall q \in G$, *i.e.*, $\Psi' = \{qX^u\} = \Psi$, we let $\hat{W}$ be the weight matrix that stores the patterns of $\Psi' = \{qX^u\}$.

$$\hat{W} = \frac{1}{M} [qV^1(qV^1)^t + qV^2(qV^2)^t + \cdots + qV^m(qV^m)^t] - I$$
$$= \frac{1}{M} [qV^1(V^1)^t q^t + qV^2(V^2)^t q^t + \cdots + qV^m(V^m)^t q^t] - qIq^t$$
$$= q \left\{ \frac{1}{M} [V^1(V^1)^t + V^2(V^2)^t + \cdots + V^m(V^m)^t] - I \right\} q^t$$
$$= qWq^t$$

So, q is the symmetric permutation of the weight matrix W.

Similarly, all the permutations of G are the symmetric permutation of the weight matrix storing the given patterns set Ψ. That is, G is the symmetric permutation of the network.

(3) Let $X^m \in \Psi$ be an attractor of the network whose weight matrix W is storing the patterns set Ψ *i.e.*,

$$W_{ij} = \frac{1}{M} \sum_{k=1}^{M} (2X_i^k - 1)(2X_j^k - 1) - \delta_{ij}$$
$$X^m = \text{step}(WX^m).$$

Consider another vector $X^u \in M$. Let the permutation relationship between X^m and X^u be $X^u = qX^m$, $q \in G$. We have

$$\text{step}(WX^u) = \text{step}(WqX^m)$$

$\because$ q is a symmetric permutation of the network, *i.e.*, $W = qWq^t$.

$$\text{step}(WX^u) = \text{step}(qWq^t qX^m) = \text{step}(qWX^m)$$
$$= q \cdot \text{step}(WX^m) = qX^m = X^u$$

i.e., X^u is an attractor of the network.

Similarly, we can prove that all patterns in Ψ are the attractors of the network, *i.e.*, the patterns set Ψ is storable.

(4) In the permutation set G, the product of two permutations pq is defined as the resultant permutation of first permuting with q and then with p. Now we prove that the following postulates are satisfied.

CLOSURE   *i.e., if q and p belong to G, then $c = qp$ also belong to G.*

$\because$ *Both q and p are the symmetric permutations of Ψ i.e., $p\Psi = \{pX^u\} = \{X^u\}$, $q\Psi = \{qX^u\} = \{X^u\}$.*

$\therefore$ $c\Psi = qp\Psi = \{qpX^u\} = \{qX^u\} = \{X^u\}$, *i.e., c is the symmetric permutation of Ψ. so G satisfies the postulate of closure.*

Associativity   We know from group theory that this postulate of associativity is true for any permutation, *i.e.*, $c(qp) = (cq)p$, for any permutations c, q, p. Obviously, G also satisfies this postulate.

Identity Element e   The permutation matrix of e is an identity matrix, so $eq = qe = q$ holds for any permutation. So do the permutations in G. Therefore, G satisfies this postulate.

Inverse Element   We know that $q^{-1} = q^t$ is true for any permutation.

For any symmetric permutation q, we have,

$$q^{-1}q\Psi = q^{-1}\Psi,$$

$$\because \quad q^{-1}q = e$$

$$\therefore \quad q^{-1}\Psi = e\,\Psi = \Psi$$

So q^{-1} is also the symmetric permutation of Ψ, *i.e.*, $q^{-1} \in G$.

The four postulates are satisfied for G. Therefore, G is a group, and accurately, subgroup of the symmetry group S_n consisting of all permutations of the elements from the set $\{1, 2, \dots, n\}$.

Thus, the theorem is proved.

In Theorem 2, we give a condition of permutation symmetry that makes sure a set of patterns be storable and have a same basin size of attraction, which can be a selecting rule of the storing patterns set in associative memory, especially for special use of it.

4. APPLICATION

In associative memory, the main task is to design a weight matrix to store the given patterns set through a certain learning process, so that the stable states of the network are exactly the given patterns set. If some of the given patterns are not the stable states of the network, we can say that the network loses some memories. If the state V is a stable state of the network but doesn't belong to the given patterns set, we say that V is a spurious stable state of the network. Both losing memories and existence of spurious stable states are the main problems of associative memory. And the former mainly results from the reason that the stored patterns set is not storable, *e.g.*, the similarity of the stored patterns may cause this problem. In this section, we demonstrate how to construct a storable patterns set by the method of group theory.

It is well known that the point group can summerize perfectly the rotation symmetries of the geometric graph in Euclidean space. And any point group is a subgroup of the symmetry group S_n. So we can use the rotation symmetries of the geometric graph to construct the storable patterns set described in Theorem 2.

For example, there are 24 rotation symmetries in the cube. The 24 rotation symmetries consist of a group called hexahedron group which is isomorphic to S_4. We can name randomly the vertexes of the cube with $\{1, 2, \dots, 8\}$ (see Fig. 1). Then a symmetric rotation of the cube will correspond to a
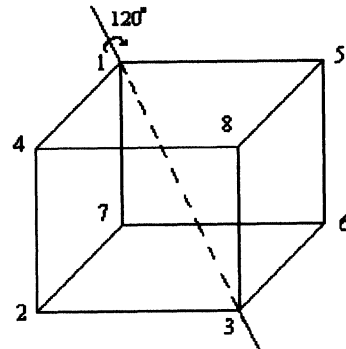

FIGURE 1   The cube.

permutation of the vertex's index. For example, when we rotate the cube 120° around the axes 13, we can see that the cube occupies the same position in the space, but the vertexes' position are changed, which amount to a permutation operation (475) (826).

However, some interesting things would be found. If we put one vector on the cent of each plane of the cube, there are 4 numbers around every vector. Let the component corresponding to the 4 numbers be 1 and the others be 0, then the 6 vectors will be $\{A^1 = (1,0,0,1,1,0,0,1)$, $A^2 = (0,0,1,0,1,1,0,1)$, $A^3 = (0,1,1,1,0,0,0,1)$, $A^4 = (1,0,0,0,1,1,1,0)$, $A^5 = (1,1,0,1,0,0,1,0)$, $A^6 = (0,1,1,0,0,1,1,0)\}$. When connected with some lines, an octahedron with 6 vertexes denoted by the 6 vectors would be seen (see Fig. 2). We can know from group theory that the rotation symmetries of the octahedron are exactly that of the cube. So a symmetric rotation of the octahedron is related to a symmetric permutation of the 6 vectors. In addition, we can see from Figure 2 that any vertex can be replaced by another through rotating the octahedron. That is, in the set of the 6 vectors, any vector can be obtained from another by a symmetric permutation. So the set of vectors is storable according to Theorem 2.

We can examine it by constructing a network with the 6 vectors. Computed by Eq. (3). the synaptic weight matrix is

$$W = \begin{bmatrix} 0 & -1/3 & -1 & 1/3 & 1/3 & -1/3 & 1/3 & -1/3 \\ -1/3 & 0 & 1/3 & 1/3 & -1 & -1/3 & 1/3 & -1/3 \\ -1 & 1/3 & 0 & -1/3 & -1/3 & 1/3 & -1/3 & 1/3 \\ 1/3 & 1/3 & -1/3 & 0 & -1/3 & -1 & -1/3 & 1/3 \\ 1/3 & -1 & -1/3 & -1/3 & 0 & 1/3 & -1/3 & 1/3 \\ -1/3 & -1/3 & 1/3 & -1 & 1/3 & 0 & 1/3 & -1/3 \\ 1/3 & 1/3 & -1/3 & -1/3 & -1/3 & 1/3 & 0 & -1 \\ -1/3 & -1/3 & 1/3 & 1/3 & 1/3 & -1/3 & -1 & 0 \end{bmatrix}$$
(7)

For the writing convenience, we list all the attractors and their domains of attraction by their transposed forms in Table I.

We can see from Table I that the patterns set consisting of the 6 vectors is storable and all the 6 vectors have a same basin size of attraction.
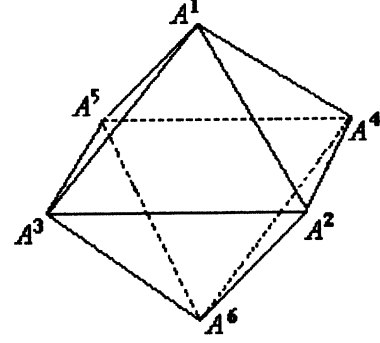

FIGURE 2   The octahedron.

TABLE I The classification table of the network with the synaptic matrix Eq. (7). In each column, vector beyond the line is attractor, vectors below the line form a basin of attraction

A^1	A^2	A^3	A^4	A^5	A^6
10011001	00101101	01110001	10001110	11010010	01100110
00100011	00000111	00010011	00011001	00110001	00001011
00100110	00001101	00010110	00011100	00110100	00001110
01000110	00010101	01010010	01011000	01000011	01001010
01001100	00100101	10010010	10000011	01001001	10000110
01010100	00101001	11000010	10001001	01010001	10001010
01100010	00101100	11001000	10010001	01100001	10001100
01100100	00101101	11010000	10011000	01110000	10001110
01100110	00101111	11010010	10011001	01110001	10001111
01100111	00111101	11010011	10011011	01110011	10010100
01101110	01101000	11010110	10011101	01110101	10011110
01110110	01101101	11011010	10100001	01111001	10100100
10100010	10101000	11100000	10111001	10110000	10101110
11100110	10101101	11110010	11011001	11110001	11001110

In this example, we show that the storable patterns set can be constructed by an octahedral group with the aid of geometric graph in Euclidean space. However, it can easily be generalized to the other group.

5. DISCUSSION AND CONCLUSION

Discrete Hopfield neural network was proposed mainly as an associative memory model. Hopfield demonstrated by computer simulation that the network with N neurons can store about $0.138\,N$ patterns in the form of the stable states, that is, not any patterns set is storable. One may thinks that

the main reasons for the small capacity is the dynamic system of the network itself, and some researchers try to improve the capacity by modifying the dynamic system of the network, such as the dynamical behavior [9], the connecting dynamics [10, 11], the output function [12], and so on. But results are not satisfactory well. All those show that a storable patterns set is necessary for associative memory, especially for special use of it.

In addition, it should also be noticed that all the prototypes built to this date, in optical or very large-scale integrated technology, are still of a fairly small size. So, it is necessary to derive exact properties for the network as opposed to asymptotic or to noncompletely rigorous ones, such as those which are based on mean-field theory in statistical mechanics which may be valid only in the limit of large systems.

In this paper, we study the discrete Hopfield neural network by the method of permutation, and associate the permutation symmetries of the network with its storing patterns set. By this way, we find some interesting properties in DHNN, which can be applied to associative memory in selecting a storable patterns set. It can be easily verified that the well-known Hadamard patterns set [13] omitted the trivial pattern $(0, 0, \ldots, 0)$, whose

permutation symmetries are proved to be isomorphic to the group $GL(n, F_2)$ by Folk [14], satisfies the condition proposed in Theorem 2. So, the Hadamard network, in which the Hadamard patterns set has been stored by Hebbian learning rule, can be regarded as a special case in this paper.

References

[1] Eiliot, J. P. and Dawber, P. G. (1979). *Symmetry in Physics* (London: Macmillan).
[2] Lautrup, B. (1989). *The Hopfield Model* (Singapore: World Scientific).
[3] Baldi, P. (1987). *Phys. Rev. Lett.*, **59**, 1976.
[4] Baldi, P. (1988). *SIAM J. Disc. Math.*, **1**(1).
[5] Hopfield, J. J. (1982). *Proc. Natl. Acad. Sci.*, USA[C], **79**, 2554–2558.
[6] Ma Jinwen (1997). *Neural Networks*, **10**, 1109–1116.
[7] Jin-quan Chen (1989). *Group representation theory for physicists*. (Singapore: World Scientific).
[8] Ma Jinwen (1999). *Neural Computing and Applications*, **8**, 25–32.
[9] Morita, M. (1993). *Neural Networks*, **6**, 115–126.
[10] Venkatesh, S. S. and Psaltis, D. (1989). *IEEE Trans on Information Theory*, **35**, 558–568.
[11] Ott, P. and Lueder, E. (1994). In: *Proc. The 6th Microcomputer Workshop, Neural Networks Theory and Applications*, Sedmihorky, Tschechische Republick, pp. 218–224.
[12] Marcus, C. M. *et al.* (1991). *Physica D*, **51**, 234–247.
[13] Krisenment, O. (1990). *Z Physik B*, **80**, 415.
[14] Folk, R., Kartashov, A., Lisonek, P. and Paule, P. (1993). *J. Phys. A: Math.*, **26**, 3159–3164.

Special Issue on
Intelligent Computational Methods for Financial Engineering

Call for Papers

As a multidisciplinary field, financial engineering is becoming increasingly important in today's economic and financial world, especially in areas such as portfolio management, asset valuation and prediction, fraud detection, and credit risk management. For example, in a credit risk context, the recently approved Basel II guidelines advise financial institutions to build comprehensible credit risk models in order to optimize their capital allocation policy. Computational methods are being intensively studied and applied to improve the quality of the financial decisions that need to be made. Until now, computational methods and models are central to the analysis of economic and financial decisions.

However, more and more researchers have found that the financial environment is not ruled by mathematical distributions or statistical models. In such situations, some attempts have also been made to develop financial engineering models using intelligent computing approaches. For example, an artificial neural network (ANN) is a nonparametric estimation technique which does not make any distributional assumptions regarding the underlying asset. Instead, ANN approach develops a model using sets of unknown parameters and lets the optimization routine seek the best fitting parameters to obtain the desired results. The main aim of this special issue is not to merely illustrate the superior performance of a new intelligent computational method, but also to demonstrate how it can be used effectively in a financial engineering environment to improve and facilitate financial decision making. In this sense, the submissions should especially address how the results of estimated computational models (e.g., ANN, support vector machines, evolutionary algorithm, and fuzzy models) can be used to develop intelligent, easy-to-use, and/or comprehensible computational systems (e.g., decision support systems, agent-based system, and web-based systems)

This special issue will include (but not be limited to) the following topics:

- **Computational methods**: artificial intelligence, neural networks, evolutionary algorithms, fuzzy inference, hybrid learning, ensemble learning, cooperative learning, multiagent learning

- **Application fields**: asset valuation and prediction, asset allocation and portfolio selection, bankruptcy prediction, fraud detection, credit risk management
- **Implementation aspects**: decision support systems, expert systems, information systems, intelligent agents, web service, monitoring, deployment, implementation

Authors should follow the Journal of Applied Mathematics and Decision Sciences manuscript format described at the journal site http://www.hindawi.com/journals/jamds/. Prospective authors should submit an electronic copy of their complete manuscript through the journal Manuscript Tracking System at http://mts.hindawi.com/, according to the following timetable:

Manuscript Due	December 1, 2008
First Round of Reviews	March 1, 2009
Publication Date	June 1, 2009

Guest Editors

Lean Yu, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China; Department of Management Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong; yulean@amss.ac.cn

Shouyang Wang, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China; sywang@amss.ac.cn

K. K. Lai, Department of Management Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong; mskklai@cityu.edu.hk