#### Check for updates

#### OPEN ACCESS

EDITED BY Qihe Shan, Dalian Maritime University, China

REVIEWED BY Zhao Zhang, University of Science and Technology Liaoning, China Sanbo Ding, Hebei University of Technology, China

\*CORRESPONDENCE Hui Tian ⊠ tianhui@cqupt.edu.cn

RECEIVED 11 February 2024 ACCEPTED 04 March 2024 PUBLISHED 19 March 2024

CITATION

Hou Y, Tian H and Wang C (2024) A novel associative memory model based on semi-tensor product (STP) *Front. Comput. Neurosci.* 18:1384924. doi: 10.3389/fncom.2024.1384924

#### COPYRIGHT

© 2024 Hou, Tian and Wang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# A novel associative memory model based on semi-tensor product (STP)

#### Yanfang Hou<sup>1</sup>, Hui Tian<sup>2\*</sup> and Chengmao Wang<sup>2</sup>

<sup>1</sup>School of Electrical and Electronic Engineering, Chongqing University of Technology, Chongqing, China, <sup>2</sup>School of Automation, Chongqing University of Posts and Telecommunications, Chongqiong, China

A good intelligent learning model is the key to complete recognition of scene information and accurate recognition of specific targets in intelligent unmanned system. This study proposes a new associative memory model based on the semi-tensor product (STP) of matrices, to address the problems of information storage capacity and association. First, some preliminaries are introduced to facilitate modeling, and the problem of information storage capacity in the application of discrete Hopfield neural network (DHNN) to associative memory is pointed out. Second, learning modes are equivalently converted into their algebraic forms by using STP. A memory matrix is constructed to accurately remember these learning modes. Furthermore, an algorithm for updating the memory matrix is developed to improve the association ability of the model. And another algorithm is provided to show how our model learns and associates. Finally, some examples are given to demonstrate the effectiveness and advantages of our results. Compared with mainstream DHNNs, our model can remember learning modes more accurately with fewer nodes.

#### KEYWORDS

associative memory model, discrete Hopfield neural network (DHNN), information storage capacity, semi-tensor product (STP) of matrices, associative memory matrix

## 1 Introduction

Under complex circumstances, autonomous perception and scene understanding of the environment are a prerequisite for intelligent unmanned systems to operate autonomously (Gonzalez-Jorge et al., 2017; Muniraj and Farhood, 2019). It directly affects whether the task can be successfully completed. By utilizing multiple sensors to collaborate with each other, the environmental feature information is extracted, and a sufficient number of learning datasets with precise calibration are established (Balestrieri et al., 2021). Then, building an intelligent learning model (Jafari and Xu, 2018; Eski and Kus, 2019; Guo et al., 2022; Kang et al., 2022) can achieve complete recognition of scene information and accurate recognition of designated targets by intelligent unmanned systems.

There are many intelligent learning models, such as neural networks (Ding et al., 2019; Pasa et al., 2022; García-Treviño et al., 2024) and fuzzy control (Zhang et al., 2021; Su et al., 2023), etc. They are effective in modeling and dealing with many nonlinear systems. The discrete Hopfield neural network (DHNN) is a classical neural network (Tank and Hopfield, 1986; Kobayashi, 2019a). It adopts a fully interconnected and completely feedback structure. Therefore, the output of any neuron is fed back to all neurons as inputs through connection weights. Its purpose is to enable the output of any neuron to be controlled by the output of all neurons so that the outputs of all neurons can be mutually constrained. Associative memory is the primary function of DHNN. The working process is divided into two stages: memory and association. In the memory stage, through design or learning of network weights, the network has several equilibrium states, which are also known as attractors. These attractors are the extreme points of the energy function of the DHNN. The memory process is to store those modes that need to be remembered as network attractors. The associative stage is such a process that the associative memory network reaches a stable state from a given state through the evolution of dynamics, i.e., converging to the attractors and recalling stored modes.

The number of attractors represents the memory capacity or storage capacity of the network. It is actually the maximum number of non-interfering modes stored in the network under a certain association error tolerance. The more attractors a network has, the larger the information storage capacity is and the stronger the associative ability is. In fact, when using DHNN for associative memory, it is constrained by memory capacity and sample differences (Kobayashi, 2019b). If there are too many samples to memorize the network may converge to a pseudo mode that is different from any learning mode. Moreover, when the network scale is fixed, the number of modes that can be remembered is very limited. In general, the maximum number of modes that a network can store is called the network capacity. It is related to size, algorithm, and vector distribution of memory modes of the network. As the number of memory modes increases, some memory modes intersect with each other. When the number of modes exceeds the network capacity, the network not only gradually forgets the modes previously memorized but also cannot remember the new modes. For a DHNN with n neurons, the maximum number of modes that can be remembered is 0.15n by appropriately selecting the connection weight matrix. It indicates that when the network size is fixed, the more the modes to be remembered are, the greater the likelihood of making mistakes during the association is. On the contrary, the lower the allowable error is, the smaller the information storage capacity is.

To solve the problem of information storage capacity, this study proposes a new associative memory model. The main mathematical tool is the semi-tensor product (STP) of matrices, which is proposed and developed by Cheng'group (Cheng and Qi, 2009). STP extends ordinary matrix multiplication to a general case, where the number of columns in the front matrix is not required to equal to that of rows in the back matrix. It not only retains the basic properties of the ordinary matrix multiplication but also has some special properties, such as pseudo commutativity. After development in the past two decades, STP has made significant progress in both theory and application, including logical control systems (Zhang et al., 2016; Tian et al., 2017; Tian and Hou, 2019), finite game (Zheng et al., 2022; Hou and Tian, 2023), and finite automata (Zhang et al., 2020; Yan et al., 2023), and so on.

The main contributions are summarized as follows:

- 1. A new associative memory model is proposed. Compared with the mainstream DHNNs, the new model can accurately remember more learning modes with fewer nodes.
- 2. Unlike most existing models that perform mode recognition after learning, our model automatically learns when needed.

3. The accuracy of mode recognition can be controlled effectively using our recognition model.

The rest of the study is organized as follows. In Section 2, some preliminaries are introduced, including DHNN, outer product method, and STP. In addition, the problem of information storage capacity is pointed out. Section 3 proposes a new memory model and its update algorithm. The learning and association process of the new model is provided in another algorithm. Section 4 gives an illustrative example, which is followed by a brief conclusion mentioned in Section 5.

## 2 Materials and methods

#### 2.1 Preliminaries

#### 2.1.1 Notations

For ease of expression, some notations are first introduced.

- $\mathbb{R}^n$ : the set of all *n*-dimensional real vectors;
- $\mathbb{M}_{m \times n}$ : the set of  $m \times n$  dimensional real matrices;
- *Col<sub>i</sub>*(*M*) (*Row<sub>i</sub>*(*M*)): the *i*th column (row) of matrix *M*;
- 0: a zero column vector with a appropriate dimension;
- δ<sup>i</sup><sub>n</sub> := Col<sub>i</sub>(I<sub>n</sub>), the *i*th column of n × n dimensional identity matrix I<sub>n</sub>;
- $\Delta_n := Col(I_n) = \{\delta_n^i \mid i = 1, 2, \dots, n\}$ , the set of all columns of  $I_n$ ;
- $r = [r_1, \cdots, r_n]^T \in \mathbb{R}^n$  is a probabilistic vector, if  $r_i \ge 0, i = 1 \cdots, n$ , and  $\sum_{i=1}^n r_i = 1$ .

#### 2.1.2 Discrete Hopfield neural network

Discrete Hopfield Neural Network (DHNN) is a type of binary neural network. Each neuron's output only takes two states 1 and -1, representing activation and inhibition, respectively. It mainly has the following characteristics:

- (1) Each unit has no self feedback (no connection to itself). That is,  $w_{ii} = 0$ .
- (2) The connection weights between units are symmetrical, i.e  $w_{ij} = w_{ji}, \forall i \neq j.$

For a DHNN with *n* binary units, it is easy to know that the cardinality of its state space is  $2^n$ .

The associative memory is an important function of DHNN. The core of implementing associative memory is to design a set of appropriate network connection weights and thresholds, based on energy extreme points (also known as learning modes). There are many design methods and learning algorithms for connection weight of DHNN, such as outer product method, projection learning rule, pseudo inverse method, and eigen structure method, etc. However, the most commonly used method is the outer product method that is based on Hebb learning rules. We recall it in the following.

Assume that the network has *n* nodes and the connection weight matrix of the network is *W*. If the network processes *m* pairwise orthogonal pattern samples, i.e., the learning task set of the network is  $X^k = (x_1^k, x_2^k, \cdots, x_n^k)^T, k = 1, 2, \cdots, m, W$  is calculated as follows:

$$W = \alpha \sum_{k=1}^{m} [X^{k} (X^{k})^{T} - I],$$
(1)

where  $\alpha$  is a positive constant.

Algorithm 1. Outer product method based on Hebb learning rules.

#### 2.1.3 Semi-tensor product of matrices

Since the main mathematical tool used in this study is STP, it is necessary to recall it here.

Definition 1. Cheng and Qi (2010) Let  $A \in \mathbb{M}_{m \times n}$ ,  $B \in \mathbb{M}_{p \times q}$ , and denote the least common multiplier lcm(n, p) of n and p by r. Then, the STP of A and B is

$$A \ltimes B := (A \otimes I_{\frac{r}{n}})(B \otimes I_{\frac{r}{p}}) \in \mathbb{M}_{\frac{mr}{n} \times \frac{qr}{p}},$$

where  $\otimes$  is the Kronecker product of matrices.

It is noted that STP has properties similar to those of the ordinary matrix products. Especially, when n = p, STP happens to be the ordinary matrix product. Therefore, STP is generalization of the ordinary matrix product. Throughout this study, the matrix products are STP and the symbol  $\ltimes$  is usually omitted.

Here, it is necessary to recall an important property. Readers refer to Cheng et al. (2011) for more details.

Let  $x = (x_1, x_2, \dots, x_m)^T \in \mathbb{R}^m, y = (y_1, y_2, \dots, y_n)^T \in \mathbb{R}^n$ . Then

$$x \ltimes y = (x_1y_1, \cdots, x_1y_n, \cdots, x_my_1, \cdots, x_my_n)^T \in \mathbb{R}^{mn}.$$

### 2.2 Problem analysis

Although DHNN has strong associative memory function, it also has some shortcomings. They are listed as follows:

- The limitation of memory capacity.
- When the memory modes are relatively close, the mode associated or recovered by the network may not be the closest to the input one.
- In some cases, the mode recalled by the network is not any of the memory modes. It leads to falling into a "pseudo state."

Next, an example will be used to illustrate these problems faced when applying DHNN to associative memory.

Example 1. Consider a DHNN with five nodes, whose learning modes are  $X^1 = (1, 1, -1, 1, 1)^T$ ,  $X^2 = (1, -1, 1, -1, -1)^T$ ,  $X^3 = (-1, 1, 1, -1, 1)^T$ , and  $X^4 = (-1, -1, -1, 1, 1)^T$ . Calculate its connection weight matrix *W* and verify its memory ability.

First, according to Equation 1 in Algorithm 1, the connection weight matrix *W* can be computed as follows:

Next, we use W that is obtained above to verify the memory ability of the network.

$$sgn(WX^{1}) = sgn(\begin{pmatrix} 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & -4 & -2 \\ 0 & 0 & -4 & 0 & 2 \\ -2 & 2 & -2 & 2 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ 1 \end{pmatrix})$$
$$= sgn((-2 \ 2 \ -6 \ 6 \ 4)^{T}) = (-1 \ 1 \ -1 \ 1 \ 1)^{T} \neq X^{1}.$$

$$sgn(WX^{2}) = sgn(\begin{pmatrix} 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & -4 & -2 \\ 0 & 0 & -4 & 0 & 2 \\ -2 & 2 & -2 & 2 & 0 \end{pmatrix}^{T}) = \begin{pmatrix} 1 & -1 \\ 1 \\ -1 \\ -1 \end{pmatrix},$$
$$= sgn(\begin{pmatrix} 2 & -2 & 6 & -6 & 8 \end{pmatrix}^{T}) = \begin{pmatrix} 1 & -1 & 1 & -1 & -1 \end{pmatrix}^{T} = X^{2}.$$
$$sgn(WX^{3}) = sgn(\begin{pmatrix} 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & -4 & -2 \\ 0 & 0 & -4 & 0 & 2 \\ -2 & 2 & -2 & 2 & 0 \end{pmatrix}^{T}) = \begin{pmatrix} -1 & 1 & 1 & -1 & -1 \end{pmatrix}^{T} \neq X^{3}.$$
$$sgn(WX^{4}) = sgn(\begin{pmatrix} 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 2 \\ -2 & 2 & -2 & 2 & 0 \end{pmatrix}^{T}) = \begin{pmatrix} -1 & 1 & 1 & -1 & -1 \end{pmatrix}^{T} \neq X^{3}.$$
$$sgn(WX^{4}) = sgn(\begin{pmatrix} 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & -4 & -2 \\ 0 & 0 & -4 & 0 & 2 \\ -2 & 2 & -2 & 2 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \\ 1 \end{pmatrix},$$
$$= sgn(\begin{pmatrix} -2 & 2 & -6 & 6 & 4 \end{pmatrix}^{T}) = \begin{pmatrix} -1 & 1 & -1 & 1 & 1 \end{pmatrix}^{T} \neq X^{4}.$$

From the above results, it can be observed that the network cannot perform normal associative memory on learning modes  $X^1$ ,  $X^3$ , and  $X^4$  because the input modes  $X^1$ ,  $X^2$ ,  $X^3$ , and  $X^4$  are non-orthogonal. When the modes to be memorized are similar, it is easy for them to intersect with each other. The network cannot fully distinguish these modes. Moreover, errors may still occur, even if a memorized mode is used as the object to be recognized. In fact, for non-orthogonal modes of a DHNN with *n* neurons, the information storage capacity is generally 0.13n-0.15n. When the number of storage modes exceeds 0.15n, the associative memory of the network may go wrong.

## **3** Results

#### 3.1 A new memory model

For any learning mode  $X^k = (l_1, l_2, \dots, l_n)^T$  of a DHNN with n nodes, we first transform it into an equivalent vector  $Y^k = (k_1, k_2, \dots, k_n)^T$ , where

$$k_i = \begin{cases} l_i, & \text{if } l_i = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Next, we continue to give the algebraic form of  $Y^k$ .  $\delta_2^{2-i}$  is identified with i, i = 0, 1. Then,  $(k_1, k_2, \dots, k_n)^T$  can be replaced by  $(\delta_2^{2-k_1}, \delta_2^{2-k_2}, \dots, \delta_2^{2-k_n})^T$ . From the study by Cheng et al. (2011),  $(\delta_2^{2-k_1}, \delta_2^{2-k_2}, \dots, \delta_2^{2-k_n})^T$  is equivalent to  $\delta_{2n}^{r_k}$ , where  $r_k = (1 - k_1)2^{n-1} + (1 - k_2)2^{n-2} + \dots + (1 - k_{n-1})2 + (2 - k_n)$ :



$$Y^{k} = (k_{1}, k_{2}, \cdots, k_{n})^{T} \sim \delta_{2}^{2-k_{1}} \delta_{2}^{2-k_{2}} \cdots \delta_{2}^{2-k_{n}} = \delta_{2^{n}}^{r_{k}},$$

where  $\delta_{2^n}^{r_k}$  is called the algebraic form of the learning mode  $Y^k$ .

In the following, we give a new memory model based on algebraic forms of learning modes.

Theorem 1. Assume that *n*-dimensional modes  $Y^k$ ,  $k = 1, 2, \dots, m$ , are required to be memorized, whose algebraic forms are  $\delta_{2^n}^{r_k}$ , respectively. A memory matrix is designed as follows:

$$L = \sum_{k=1}^{m} \delta_{2^{n}}^{r_{k}} (\delta_{2^{n}}^{r_{k}})^{T}.$$
 (2)

Then, all modes  $Y^k$  are memorized by *L* and fully recognized.

PROOF. For any mode  $Y^i$ , its algebraic form is  $\delta_{2^n}^{r_i}$ . Then,

$$L\delta_{2^{n}}^{r_{i}} = \sum_{k=1}^{m} \delta_{2^{n}}^{r_{k}} (\delta_{2^{n}}^{r_{k}})^{T} \delta_{2^{n}}^{r_{i}}.$$

Since  $(\delta_{2^n}^{r_i})^T \delta_{2^n}^{r_i} = 1$  and  $(\delta_{2^n}^{r_k})^T \delta_{2^n}^{r_i} = 0, k \neq i$ , we have

$$L\delta_{2^{n}}^{r_{i}} = \delta_{2^{n}}^{r_{i}} (\delta_{2^{n}}^{r_{i}})^{T} \delta_{2^{n}}^{r_{i}} = \delta_{2^{n}}^{r_{i}}.$$
(3)

Because of the arbitrariness of selecting  $Y^i$ , Equation 3 implies that all modes  $Y^k$  are memorized and recognized by *L*. The proof is completed.

For convenience, we denote  $Row_i(Col_j(L))$  by  $l_{ij}$ . It is easy to observe that

$$l_{ij} = \begin{cases} 1, & \text{if } i = j = r_k, \\ 0, & \text{otherwise.} \end{cases}$$
(4)

From Equation 4, we find: (1) the memory matrix *L* is a Boolean matrix, (2) only on the main diagonal, there may be non-zero elements. For any mode  $\delta_{2n}^{r_i}$  to be recognized, if  $L\delta_{2n}^{r_i} = 0$ , this mode is not a stored standard learning mode. Otherwise,  $L\delta_{2n}^{r_i}$  must be  $\delta_{2n}^{r_i}$ , which means the mode  $\delta_{2n}^{r_i}$  can be recognized accurately. The recognition process is shown in Figure 1.

Example 2. The four learning modes in Example 1 are used to verify the memory ability of the new memory model (Equation 2).

First, it is easy to calculate algebraic forms of the four learning modes as follows:

$$\begin{split} Y^{1} &= (1, 1, 0, 1, 1)^{T} \sim \delta_{2}^{1} \delta_{2}^{1} \delta_{2}^{2} \delta_{2}^{1} \delta_{1}^{1} = \delta_{32}^{5}, \\ Y^{2} &= (1, 0, 1, 0, 0)^{T} \sim \delta_{2}^{1} \delta_{2}^{2} \delta_{2}^{1} \delta_{2}^{2} \delta_{2}^{2} = \delta_{32}^{12}, \\ Y^{3} &= (0, 1, 1, 0, 1)^{T} \sim \delta_{2}^{2} \delta_{2}^{1} \delta_{2}^{1} \delta_{2}^{1} \delta_{2}^{1} = \delta_{32}^{19}, \\ Y^{4} &= (0, 0, 0, 1, 1)^{T} \sim \delta_{2}^{2} \delta_{2}^{2} \delta_{2}^{2} \delta_{2}^{1} \delta_{2}^{1} = \delta_{32}^{29}. \end{split}$$
(5)

According to Equation 2, the memory matrix *L* is as follows:

 $L = \delta_{32}^{5} (\delta_{32}^{5})^{T} + \delta_{32}^{12} (\delta_{32}^{12})^{T} + \delta_{32}^{19} (\delta_{32}^{19})^{T} + \delta_{32}^{29} (\delta_{32}^{29})^{T}$ 

0 
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0 1 0 
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0 0 0 0 0 0 0 0 0 0  $0 \ 0 \ 0 \ 0$ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 

 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0 (6)

Next, we verify the memory ability of *L*.

$$\begin{split} LY^1 &= [\delta_{32}^5 (\delta_{32}^5)^T + \delta_{32}^{12} (\delta_{32}^{12})^T + \delta_{32}^{19} (\delta_{32}^{19})^T + \delta_{32}^{29} (\delta_{32}^{29})^T] \delta_{32}^5 = \delta_{32}^5 \\ &= Y^1, \\ LY^2 &= [\delta_{32}^5 (\delta_{32}^5)^T + \delta_{32}^{12} (\delta_{32}^{12})^T + \delta_{32}^{19} (\delta_{32}^{19})^T + \delta_{32}^{29} (\delta_{32}^{29})^T] \delta_{32}^{12} = \delta_{32}^{12} \\ &= Y^2, \\ LY^3 &= [\delta_{32}^5 (\delta_{32}^5)^T + \delta_{32}^{12} (\delta_{32}^{12})^T + \delta_{32}^{19} (\delta_{32}^{19})^T + \delta_{32}^{29} (\delta_{32}^{29})^T] \delta_{32}^{19} = \delta_{32}^{19} \\ &= Y^3, \\ LY^4 &= [\delta_{32}^5 (\delta_{32}^5)^T + \delta_{32}^{12} (\delta_{32}^{12})^T + \delta_{32}^{19} (\delta_{32}^{19})^T + \delta_{32}^{29} (\delta_{32}^{29})^T] \delta_{32}^{29} = \delta_{32}^{29} \\ &= Y^4. \end{split}$$

Remark 1. From the verification results in Equation 7, it is found that the four learning modes in Example 1 can be completely remembered by our memory model (Equation 6) without any errors. Furthermore, it can be directly verified that all modes from  $\Delta_{2^n}$  can be accurately remembered if learned by our model.

Although this new memory model (Equation 2) can remember all modes accurately, its associative ability (robustness) is not good. Especially, in the real environment, there always be interference signals that may cause bit errors and then lead to recognition error. Therefore, it is necessary to improve the associative ability of the memory model.

# 3.2 Update the associative memory matrix *L*

As mentioned above, there are always interference signals that cause some bit errors in a realistic environment. When the error is within the allowable range, the interfered mode should be correctly identified. However, the memory model (Equation 2) cannot achieve this goal because it lacks associative ability.

In practical engineering, the accuracy of mode recognition is often guaranteed through an error control parameter. There are many such parameters. The bit error control parameter (BEC) is the simplest and most effective one among them. Specifically, the number of different bits between the input mode i and the kth stored learning mode, denoted by  $s_{ki}$ , is used to define the degree of difference between the two modes. If  $s_{ki}$  is smaller than or equal to BEC, mode i may be recognized and classified as the kth learning mode. Otherwise, it does not belong to the kth learning modes with difference degrees less than or equal to the BEC, mode i should be classified into the above learning modes according to a probability distribution.

Without loss of generality, we let mode i be different from all learning modes that have been stored in the associative memory matrix L. In the following, we provide an update algorithm for L in Algorithm 2, according to a given BEC.

1:	Determine a BEC $E_c$ according to engineering
	requirements. Calculate the number of
	different bits between a mode $i$ to be
	recognized and the $k$ th stored learning mode,
	and denote the number by $s_{ki}(k=1,2,\cdots,m)$ .
2:	for $k=1$ to $m$ do
3:	(Calculate the transition probability
	$l_{r_k r_i}$ . )
4:	if $0 < s_{ki} \leq E_c$ , then
	1
5:	$l_{r_k r_i} = \frac{s_{ki}}{m}$ ;
	$\sum \frac{1}{s_{ij}}$
	$\substack{j=1\\0 < s_{ii} < E_c}$
6:	else
7:	$l_{r_{L}r_{i}}=0$ .
8:	end if
9:	end for
10:	Return the updated associative memory matrix
	$L = (l_{r_k r_i})_{2^n \times 2^n}$ .

Algorithm 2. The associative memory matrix *L* in Equation 2 can be updated according to a given BEC to improve its associative ability.

It is worth noting that the parameter  $s_{ki}$  indicates the difference degree between mode *i* and mode *k*. The larger  $s_{ki}$  is, the greater the difference between the two modes is and the lower the probability of mode *i* being classified as the *k*th learning mode is.

# 3.3 Learning and association of the new associative memory model

We put the results of Sections 3.1, 3.2 together and then present a new algorithm for the learning and association of model (Equation 2).

(7)



For ease of understanding, the use and learning process of our model are illustrated in the form of a flowchart, see Figure 2.

Algorithm 3 includes two core parts: recognition and learning. In fact, after an object to be recognized enters our model for recognition, if it is recognized as a standard mode, the model outputs the result. Otherwise, the model learns immediately, reidentifies the aforementioned object, and finally outputs the result, which is the final result. The above algorithm can be summarized as a diagram, see Figure 3.

Compared with DHNN, this new associative memory model has the following advantages:

• It can accurately remember more learning modes with fewer nodes. All modes in their algebraic forms are orthogonal to each other. Therefore, theoretically speaking, our model can remember all modes.

- The accuracy of mode recognition can be effectively controlled. As analyzed in Section 3.2, the parameter BEC can be used to control the mode recognition errors.
- This model works more efficiently because it learns automatically and only when needed. That is, when the input mode cannot be recognized for the first time, our associative learning model automatically uses the above input mode to learn (update the memory matrix) and then proceeds re-recognition. This is different from the most existing models, which perform mode recognition after learning.

# 4 Discussion

In the following, an example is given to show the effectiveness of the method developed in this study.

-	
	Equation 2.
Step 2.	For a mode <i>i</i> to be recognized, obtain its
	algebraic form $\delta_{2^n}^{r_i}$ .
Step 3.	Calculate $L\delta_{2^n}^{r_i}=Col_{r_i}(L)$ . If $L\delta_{2^n}^{r_i}\neq {\bf 0}$ , classify
	mode <i>i</i> according to the probability
	distribution $Col_{r_i}(L)$ . Else, go to Step 4.
Step 4.	Use Algorithm 2 to update the associative
	memory matrix L.
Ctop F	Check whether $I_{i}^{r_{i}} \neq 0$ holds on not. If was

Step 1. Construct the memory matrix L according to

Step 5. Check whether  $L\delta_{2n}^{\prime i} \neq \mathbf{0}$  holds or not. If yes, classify mode i according to the probability distribution  $Col_{r_i}(L)$ . Else, Mode i does not belong to any learning mode.

Algorithm 3. Assume that there are *m n*-dimensional learning modes. The mode recognition and model learning can be carried out by the following steps.



Example 3. Use Algorithm 3 to verify the associative memory ability of the four learning modes in Example 1. Assume that the BEC is  $E_c = 2$ .

When the mode  $X^i$  to be recognized is one of the learning modes, it has been verified in Example 2 that  $X^i$  can be accurately identified. In the following, we mainly consider the case that the object to be recognized is not a stored learning mode.

Step 1. From Example 2, we know that the algebraic forms of four learning modes and the memory matrix L have been given in Equations 5, 6, respectively.

Step 2. Let the object to be recognized be

$$Y^{i} = (0, 1, 0, 1, 1)^{T} \sim \delta_{2}^{2} \delta_{2}^{1} \delta_{2}^{2} \delta_{2}^{1} \delta_{2}^{1} = \delta_{32}^{21}.$$

Step 3. We calculate

Since  $L\delta_{32}^{21} = 0$ , we should use Algorithm 2 to update the associative memory matrix *L*.

Step 4. Throughout calculation, we get the numbers of different bits between  $Y^i$  and  $Y^1$ ,  $Y^2$ ,  $Y^3$ , and  $Y^4$  which are  $s_{1i} = 1$ ,  $s_{2i} = 5$ ,



 $s_{3i} = 2$ , and  $s_{4i} = 1$ , respectively. It is noted that  $E_c = 2$ ,  $s_{1i} \le E_c$ ,  $s_{2i} > E_c$ ,  $s_{3i} \le E_c$ ,  $s_{4i} \le E_c$ . According to

$$l_{r_k r_i} = \begin{cases} \frac{\frac{1}{s_{ki}}}{\sum_{j=1}^{4} \frac{1}{s_{ji}}}, & 0 < s_{ki} \le 2, \\ \sum_{j=1}^{4} \frac{1}{s_{ji}} & \\ 0 < s_{ji} \le 2 \\ 0, & s_{ki} > 2, \end{cases}$$

we have  $l_{5,21} = \frac{2}{5}$ ,  $l_{12,21} = 0$ ,  $l_{19,21} = \frac{1}{5}$ , and  $l_{29,21} = \frac{2}{5}$ . Then, the 21th column of the associative memory matrix *L* is

updated to

Moreover, mode  $\delta_{32}^{21}$  (equivalently,  $(0, 1, 0, 1, 1)^T$ ) is classified as  $\delta_{32}^5$  (equivalently,  $(1, 1, 0, 1, 1)^T$ ),  $\delta_{32}^{19}$  (equivalently,  $(0, 1, 1, 0, 1)^T$ ), and  $\delta_{32}^{29}$  (equivalently,  $(0, 0, 0, 1, 1)^T$ ) stored in the network with

probabilities of 0.4, 0.2, and 0.4, respectively. The recognition result of our model is shown in Figure 4.

From the example, we can observe that the associative memory can also be performed through Algorithm 3, even though the mode  $X^i$  is not one of the learning modes.

If all modes in  $\Delta_{2^5}$  are identified, the leaning result of *L* is as follows:

main mathematical tool used in this study was STP. It can equivalently transform ordinary vector modes into the algebraic forms that are mutually orthogonal. Therefore, the model proposed in this study can accurately remember all learning modes theoretically. However, due to the complexity of the environment, there are inevitably some interfering factors. Hence, in order to improve the association ability or robustness of the model, an on-demand learning update algorithm for the

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	$\frac{2}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	0	1	1	$\frac{2}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{4}$	0	0	0	2	$\frac{1}{2}$	$\frac{1}{4}$	0	0	0	0	0	0	0	0	0
	3 0		$\frac{4}{0}$	0	0	0	3 0	20	3 0	0	0	0		3 0	3 0	0	4 0	0	0	0	5 0	20	$^{4}_{0}$	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	1	1	2	0	0	0	1	1	1	2	1	0	1	1	1	0	0	0	1	0	0	0	0	0	1	1	2	0	0	0	1
	0	2	4	3	0	0	0	2	3	0	3	0	0	3	3	0	0	0	0	3	0	0	0	0	0	2	4	3	0	0	0	2
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
=	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	1	1	1	2	1	0	1	1	1	0	1	1	0	0	1	0
	3	0	2	3	0	0	3	0	0	0	3	0	0	0	0	0	2	•	1	3	5	0	2	0	3	0	2	3	0	0	3	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	1	0	0	0	2	1	1	0	2	1	1	0	1	1	2	1
	0	0	0	0	0	0	0	0	3	0	0	0	2	3	3	0	4	0	0	0	5	2	4	0	3	2	4	0	•	•	3	2
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	_ U	U	U	U	U	0	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	υ	U	U	U	U	U	U	U	0	0

whose state transition diagram is shown in Figure 5.

L

Of course, in practical application, it is not necessary to calculate all elements of L. In fact, when mode i requires to be identified, we only need to determine the value of the  $r_i$ th column of matrix L.

## 5 Conclusion

A new associative memory model has been proposed to solve the problem of information storage capacity. The

model was developed. It utilizes parameter BEC to effectively control the recognition errors. Compared with the classical DHNN, the new model can store more information with fewer nodes.

However, when the parameter BEC is too large, the new model may encounter such a situation that the input mode belongs to multiple different learning modes, according to a probability distribution. In this case, the input mode may not be accurately recognized. Therefore, how to choose the best BEC based on the actual situation and how to apply the associative memory model to practical scenarios as soon as possible are our future research topics.



# Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

### Author contributions

YH: Conceptualization, Formal analysis, Investigation, Methodology, Validation, Writing—original draft, Writing review & editing. HT: Conceptualization, Formal analysis, Funding acquisition, Methodology, Project administration, Supervision, Writing—review & editing. CW: Formal analysis, Investigation, Writing—review & editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This research was funded by the National Key R&D Program of China (2021YFB3203202) and Chongqing Nature Science Foundation (cstc2020jcyj-msxmX0708).

# References

Balestrieri, E., Daponte, P., De Vito, L., and Lamonaca, F. (2021). Sensors and measurements for unmanned systems: an overview. *Sensors* 21:1518. doi: 10.3390/s21041518

Cheng, D., and Qi, H. (2009). Controllability and observability of boolean control networks. *Automatica* 45, 1659–1667. doi: 10.1016/j.automatica.2009.03.006

Cheng, D., and Qi, H. (2010). A linear representation of dynamics of boolean networks. *IEEE Trans. Autom. Control* 55, 2251–2258. doi: 10.1109/TAC.2010.2043294

Cheng, D., Qi, H., and Li, Z. (2011). Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach. London: Springer-Verlag. doi: 10.1007/978-0-85729-097-7

## Acknowledgments

The authors would like to thank the reviewers for their constructive and valuable suggestions on the earlier drafts of this manuscript.

# **Conflict of interest**

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

# Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Ding, S., Wang, Z., and Zhang, H. (2019). Quasi-synchronization of delayed memristive neural networks via region-partitioning-dependent intermittent control. *IEEE Trans. Cybern.* 49, 4066–4077. doi: 10.1109/TCYB.2018.2856907

Eski, I., and Kus, Z. Α. (2019). Control of unmanned using neural ppl. 31, 58 control agricultural vehicles network-based 10 1007/s00521-017-3 583-595 Neural Comput. Appl. doi 026-4

García-Treviño, E. S., Yang, P., and Barria, J. A. (2024). Wavelet probabilistic neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 35, 376–389. doi: 10.1109/TNNLS.2022.3174705

Gonzalez-Jorge, H., Martinez-Sanchez, J., Bueno, M., and Arias, P. (2017). Unmanned aerial systems for civil applications: a review. *Drones* 1:2. doi: 10.3390/drones1010002

Guo, Y., Zhao, Y., Rothfus, T. A., and Avalos, A. S. (2022). A novel invasive plant detection approach using time series images from unmanned aerial systems based on convolutional and recurrent neural networks. *Neural Comput. Appl.* 34, 20135–20147. doi: 10.1007/s00521-022-07560-3

Hou, Y., and Tian, H. (2023). Research on the dynamic characteristics of photovoltaic power production and sales based on game theory. *Sustainability* 15, 1–19. doi: 10.3390/su151914645

Jafari, M., and Xu, H. (2018). Intelligent control for unmanned aerial systems with system uncertainties and disturbances using artificial neural network. *Drones* 2:30. doi: 10.3390/drones2030030

Kang, H. H., Lee, D. K., and Ahn, C. K. (2022). Finite-memory-structured online training algorithm for system identification of unmanned aerial vehicles with neural networks. *IEEE ASME Trans. Mechatron.* 27, 5846–5856. doi: 10.1109/TMECH.2022.3190053

Kobayashi, M. (2019a). O(2)-valued hopfield neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 30, 3833–3838. doi: 10.1109/TNNLS.2019.2897994

Kobayashi, M. (2019b). Storage capacity of hyperbolic hopfield neural networks. *Neuron* 369, 185-190. doi: 10.1016/j.neucom.2019.08.064

Muniraj, D., and Farhood, M. (2019). Detection and mitigation of actuator attacks on small unmanned aircraft systems. *Control Eng. Pract.* 83, 188–202. doi: 10.1016/j.conengprac.2018.10.022

Pasa, L., Navarin, N., and Sperduti, A. (2022). Multiresolution reservoir graph neural network. *IEEE Trans. Neural Netw. Learn. Syst.* 33, 2642–2653. doi: 10.1109/TNNLS.2021.3090503

Su, Y., Shan, Q., Li, T., and Chen, C. L. P. (2023). Variable separation-based fuzzy optimal control for multiagent systems in nonstrict-feedback form. *IEEE Trans. Fuzzy Syst.* 34, 547–561. doi: 10.1109/TFUZZ.2023.3302293

Tank, D. W., and Hopfield, J. J. (1986). Simple 'neural' optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit. *IEEE Trans. Circuits Syst.* 33, 533–541. doi: 10.1109/TCS.1986.1085953

Tian, H., and Hou, Y. (2019). State feedback design for set stabilization of probabilistic boolean control networks. *J. Franklin Inst.* 356, 4358-4377. doi: 10.1016/j.jfranklin.2018.12.027

Tian, H., Zhang, H., Wang, Z., and Hou, Y. (2017). Stabilization of k-valued logical control networks by open-loop control via the reverse-transfer method. *Automatica* 83, 387–390. doi: 10.1016/j.automatica.2016.12.040

Yan, Y., Cheng, D., Feng, J. E., Li, H., and Yue, J. (2023). Survey on applications of algebraic state space theory of logical systems to finite state machines. *Sci. China Inf. Sci.* 66, 1–20. doi: 10.1007/s11432-022-3538-4

Zhang, H., Liu, Y., Dai, J., and Wang, Y. (2021). Command filter based adaptive fuzzy finite-time control for a class of uncertain nonlinear systems with hysteresis. *IEEE Trans. Fuzzy Syst.* 29, 2553–2564. doi: 10.1109/TFUZZ.2020.3003499

Zhang, H., Tian, H., Wang, Z., and Hou, Y. (2016). Synchronization analysis and design of coupled boolean networks based on periodic switching sequences. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 2754–2759. doi: 10.1109/TNNLS.2015.2499446

Zhang, Z., Chen, Z., Han, X., and Liu, Z. (2020). Stabilization of probabilistic finite automata based on semi-tensor product of matrices. *J. Franklin Inst.* 357, 5173–5186. doi: 10.1016/j.jfranklin.2020.02.028

Zheng, Z., Tian, H., Zhu, P., Chi, Y., Liu, Y., and Jia, X. (2022). Research on strategy of green electricity acquisition transaction of park-level energy internet by using stp. *Front. Energy Res.* 10, 1–7. doi: 10.3389/fenrg.2022.953039