

Analysis and design of multivalued many-to-one associative memory driven by external inputs

Qiang Fang(方强) and Hao Zhang(张浩)

Citation: Chin. Phys. B, 2025, 34 (**8**): 080701. DOI: 10.1088/1674-1056/add24b
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Deep learning-enabled inverse design of polarization-selective structural color based on coding metasurface

Haolin Yang(杨昊霖), Bo Ni(倪波), Junhong Guo(郭俊宏), Hua Zhou(周华), and Jianhua Chang(常建华)
Chin. Phys. B, 2025, 34 (**5**): 050702. DOI: 10.1088/1674-1056/adb94a

Coexistence and local Mittag-Leffler stability of fractional-order recurrent neural networks with discontinuous activation functions

Yu-Jiao Huang(黄玉娇), Shi-Jun Chen(陈时俊), Xu-Hua Yang(杨旭华), Jie Xiao(肖杰)
Chin. Phys. B, 2019, 28 (**4**): 040701. DOI: 10.1088/1674-1056/28/4/040701

Synthesization of high-capacity auto-associative memories using complex-valued neural networks

Yu-Jiao Huang(黄玉娇), Xiao-Yan Wang(汪晓妍), Hai-Xia Long(龙海霞), Xu-Hua Yang(杨旭华)
Chin. Phys. B, 2016, 25 (**12**): 120701. DOI: 10.1088/1674-1056/25/12/120701

Multistability of delayed complex-valued recurrent neural networks with discontinuous real-imaginary-type activation functions

Huang Yu-Jiao (黄玉娇), Hu Hai-Gen (胡海根)
Chin. Phys. B, 2015, 24 (**12**): 120701. DOI: 10.1088/1674-1056/24/12/120701

Stability analysis of Markovian jumping stochastic Cohen–Grossberg neural networks with discrete and distributed time varying delays

M. Syed Ali
Chin. Phys. B, 2014, 23 (**6**): 060702. DOI: 10.1088/1674-1056/23/6/060702

SPECIAL TOPIC — Computational programs in complex systems

Analysis and design of multivalued many-to-one associative memory driven by external inputs

Qiang Fang(方强) and Hao Zhang(张浩)†

College of Informatics, Huazhong Agricultural University, Wuhan 430000, *China*

(Received 2 April 2025; revised manuscript received 27 April 2025; accepted manuscript online 30 April 2025)

This paper proposes a novel multivalued recurrent neural network model driven by external inputs, along with two innovative learning algorithms. By incorporating a multivalued activation function, the proposed model can achieve multivalued many-to-one associative memory, and the newly developed algorithms enable effective storage of many-to-one patterns in the coefficient matrix while maintaining the indispensability of inputs in many-to-one associative memory. The proposed learning algorithm addresses a critical limitation of existing models which fail to ensure completely erroneous outputs when facing partial input missing in many-to-one associative memory tasks. The methodology is rigorously derived through theoretical analysis, incorporating comprehensive verification of both the existence and global exponential stability of equilibrium points. Demonstrative examples are provided in the paper to show the effectiveness of the proposed theory.

Keywords: many-to-one associative memories, recurrent neural network, global exponential stability, external input

PACS: 07.05.Mh, 02.30.Ks

DOI: 10.1088/1674-1056/add24b **CSTR:** 32038.14.CPB.add24b

1. Introduction

Associative memory is a cognitive ability of the human brain and serves as the foundation for other cognitive functions. Constructing neural computational models to simulate this function is of great significance for understanding the memory processes of the brain and designing brain-inspired computing systems. In the 1970s, single-layer feedforward networks were able to achieve associative memory.[1,2] However, the robustness of these models cannot be guaranteed, as they fail to effectively produce the desired output when noise is introduced into the input. Moreover, in some of the aforementioned models, the stored patterns required mutual orthogonality, which limited their ability to accurately simulate real-world associative memory functions. This limitation persisted until 1982, the Hopfield network[3] established an associative memory framework based on neural dynamics using the concept of an energy function. It can effectively store and retrieve memory patterns, forming the foundation of modern artificial memory models, and it has been widely studied. Since then, numerous models based on recurrent neural networks have been proposed.[4–8]

According to the difference between the input probe and the desired pattern, associative memory can be divided into two types: autoassociative memory and heteroassociative memory. In autoassociative memory, the input probe and desired pattern are both identical in content and form. In contrast, in heteroassociative memory, there are significant differences in the content and form of input probe and desired patterns. Traditional Hopfield networks only achieve autoassociative functionality. Kosko[9] proposed bidirectional associative memory (BAM) and modified the original network structure, changing the single-layer fully connected network to a structure where neurons in different layers are interconnected but neurons in the same layer are not, thereby achieving heteroassociative functionality. The concept of heteroassociative memory can be classified based on the number of input stimuli into one-to-many, many-to-one, and many-to-many associative memories, corresponding to divergent thinking, convergent thinking, and integrative thinking in cognitive processes of the human brain. The multidirectional associative memory model proposed by Hagiwara[10] can achieve all these functionalities. Currently, there are two methods for driving associative memory: the initial-state-driven method (ISDM) and the input-driven method (IDM). As the name suggests, the ISDM sets the initial state of the network's neurons as the network's input. Some models have already been developed based on this method.[11–20] ISDM is based on local stability. Its advantages include a simple learning algorithm and easier implementation when handling large amounts of data. However, its drawback is the presence of spurious equilibrium points, meaning that errors and extra attractors exist in the network after it is designed. The IDM completely avoids this issue because it is based on global stability, where each input pattern corresponds to only one equilibrium point. In the

†Corresponding author. E-mail: hzhang2021@mail.hzau.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

ISDM, the input provides stimulation to the network only during the first iteration, and as the iterations progress, the stimulation from the original input gradually fades. In contrast, in the IDM, the original input continuously stimulates the network throughout the process. When the human brain engages in many-to-one associative memory, it always relies on the original content for recall and retrieval. Grassi[21] was the first to propose an IDM based on global stability. His learning algorithm places parameters in a large matrix and then performs matrix operations to store patterns. However, the robustness of this learning algorithm lacks theoretical support, making it impossible to precisely calculate its robustness range. Zeng *et al.*[22] first stored patterns within the saturation range of the activation function, ensuring the correctness of the output and the robustness of the model. We can precisely calculate the model's tolerance range to input noise through the learning algorithm in Ref. [22]. In the continuous bidirectional associative memory model, Sha *et al.*[23] implemented the IDM to achieve many-to-many functionality, where the inputs of the two layers correspond to different outputs. Additionally, Sha *et al.* extended this bidirectional associative memory model to a multidirectional associative memory model. Many recently proposed models by various scholars are based on the IDM.[24–27]

Current associative memory models are predominantly limited to storing and recalling binary patterns. To handle multi-valued patterns, these models require additional encoding before storage and decoding after retrieval. This process inevitably increases the number of network neurons and model parameters, leading to substantial resource consumption. However, by incorporating a multi-valued activation function, each neuron can directly achieve desired state values, thereby enabling the network to efficiently store and recall multi-valued patterns without such overhead. The model proposed by Zhang *et al.*[28] stores patterns within the saturation range of a multi-valued activation function, achieving high storage capacity associative memory. There are already some research findings on neural networks with multi-valued activation functions.[29–31]

The human brain's associative memory process often relies on multiple stimuli of varying importance to complete an association. Crucially, more significant stimuli exert greater influence on the final memory outcome. In certain scenarios, the accurate retrieval of a specific object requires the simultaneous presence of all relevant stimuli. For instance, recalling the memory of a "white cloud" necessitates three essential stimulus elements: 'whiteness', 'floating in the sky', and 'water composition' — the absence of any single element will lead to inaccurate recollection. However, current many-to-one or many-to-many associative memory models lack theoretical guarantees for reliably accomplishing such cognitive tasks. In Ref. [32], the process of using the same image with different noises as input to obtain the same output is considered a many-to-one associative memory process. Similarly, Hattori *et al.*[33] and Li *et al.*[34] discussed a related idea, suggesting that many-to-one memories can be decomposed into multiple one-to-one associative memories. However, this approach wastes storage resources and fails to simulate the biological associative memory process. We believe that many-to-one associative memory should involve simultaneously processing multiple inputs to achieve a single output. Yano *et al.*[35] achieved the many-to-one and many-to-many functionalities by incorporating a chaotic layer into the bidirectional associative memory model, which alters the equilibrium points. Currently, there are other models capable of achieving many-to-one associative memory,[36–42] but most are based on the ISDM and still have the issue of spurious equilibrium points. Moreover, the aforementioned models capable of achieving many-to-one associative memory cannot theoretically guarantee completely erroneous outputs in the presence of missing inputs.

In this paper, a multivalued many-to-one associative memory model based on IDM is proposed. By employing the IDM architecture, we achieve many-to-one associative memory through applying multiple input patterns at the input layer during a single recall process and mapping the network equilibrium points to a single output pattern. Additionally, we utilize the properties of the saturation region of multivalued activation functions to simulate the essential nature of the initial stimulus in many-to-one associative memory in the human brain. The following are the innovative points of this paper:

1. A multivalued many-to-one associative memory model based on IDM is proposed. Unlike the model in Ref. [34], the proposed model achieves many-to-one mapping through one-shot association. Furthermore, the model overcomes the spurious equilibrium point issue inherent in multidirectional associative memory models,[39] and successfully accomplishes storage and accurate recall of 256-level grayscale images.

2. A new learning algorithm for many-to-one associative memory is proposed. By utilizing the saturation regions of the activation function, we achieve equilibrium point storage and the indispensability of inputs. Compared with existing methods,[23,34,39] the proposed algorithm successfully addresses a critical issue in multi-input scenarios — ensuring completely erroneous network outputs when partial inputs are missing, with rigorous theoretical proofs provided for this capability.

The remaining part of this article is organized as follows. In Section 2, we describe some preliminaries, model, and definitions. In Section 3, the design procedure is provided for the designed model. Two examples are given in Section 4. Finally, the conclusion is presented in Section 5.

2. Preliminaries

In this section, the considered problem and the model are explained in detail.

2.1. Design problem

Denote $\Psi^n = \{x \in \boldsymbol{R}^n,\ x = (x_1,\ldots,x_n)^{\mathrm{T}},\ x_i = -(4q-3)$ or $4q-3, q = 1,\ldots,k, i = 1,\ldots,n\}$ as the set of the n-dimensional multipolar vectors.

In this article, we consider the following design problem: Given ρ ($\rho \le \min\{(2k)^n,(2k)^m,(2k)^p\}$) pairwise vectors $(u^{1(1)},u^{2(1)},s^{(1)}),(u^{1(2)},u^{2(2)},s^{(2)}),\ldots,(u^{1(\rho)},u^{2(\rho)},s^{(\rho)})$, where $(u^{1(l)},u^{2(l)},s^{(l)}) \in \Psi^n \times \Psi^m \times \Psi^p$ ($l \in (1,\ldots,\rho)$). Design a two-to-one associative memory based on recurrent neural networks, where $u^{1(l)}$ and $u^{2(l)}$ are the inputs to the network, $s^{(l)}$ is the corresponding output and the network will not produce the output $s^{(l)}$ if either $u^{1(l)}$ or $u^{2(l)}$ is missing.

2.2. Model

Considering the following recurrent neural networks:

$$\dot{\boldsymbol{X}}(t) = -\boldsymbol{C}\boldsymbol{X}(t) + \boldsymbol{A}\boldsymbol{f}(\boldsymbol{X}(t)) + \boldsymbol{E}^1\boldsymbol{U}^1 + \boldsymbol{E}^2\boldsymbol{U}^2 + \boldsymbol{V}, \quad (1)$$

where $t \ge 0$, $\boldsymbol{X}(t) = x(t) = (x_1(t),\ldots,x_n(t))^{\mathrm{T}}$ is the state vector, $\boldsymbol{C} = \mathrm{diag}(c_1,\ldots,c_n)$ is the self-inhibition matrix, c_i is a positive constant, $\boldsymbol{A} = (a_{ij})_{n\times n}$ is the connection weight matrix, $\boldsymbol{f}(\boldsymbol{X}(t)) = (f(x_1(t)),\ldots,f(x_n(t)))^{\mathrm{T}}$ is the activation function, $\boldsymbol{E}^1 = (e^1_{ij})_{n\times m}$ and $\boldsymbol{E}^2 = (e^2_{ij})_{n\times p}$ are the input matrices, $\boldsymbol{U}^1 = (u^1_1,\ldots,u^1_m)^{\mathrm{T}}$ and $\boldsymbol{U}^2 = (u^2_1,\ldots,u^2_p)^{\mathrm{T}}$ are the external input vectors, $\boldsymbol{V} = (v_1,\ldots,v_n)^{\mathrm{T}}$ is the bias vector.

Remark 1 Unlike the ISDM models,[33,38] the input-driven model maintains continuous stimulation from the original input throughout the entire associative process. In the initial-state-driven model, the original input stimulation is present only at the beginning of the iteration and disappears once the iteration starts.

The activation function is defined as follows:

$$f(x) = \begin{cases} -(4k-3), & x \in (-\infty, 3-4k], \\ 2x+(4k-3), & x \in (3-4k, 5-4k], \\ \ldots & \\ 2x+5, & x \in (-5,-3], \\ -1, & x \in (-3,-1], \\ x, & x \in (-1,1), \\ 1, & x \in [1,3), \\ 2x-5, & x \in [3,5), \\ \ldots & \\ 2x-(4k-3), & x \in [4k-5, 4k-3), \\ 4k-3, & x \in [4k-3, +\infty). \end{cases} \quad (2)$$

Remark 2 The multi-valued piecewise activation function used in this study can adjust the number of saturation intervals by setting parameter k. When $k = 1$ is set, this activation function becomes the conventional sign activation function. When $k = 2$, as shown in Fig. 1, this activation function has four saturation regions, meaning the correct output states of the network can take values of -5, -1, 1, and 5. The lengths of the two central saturation regions are 2, while the saturation regions at the extremes are infinitely long. We designate the central saturation regions as the correct output regions and the extreme saturation regions as the incorrect equilibrium regions. This property will later be utilized to ensure the indispensability of inputs; therefore, $k \ge 2$ is chosen for subsequent analysis.

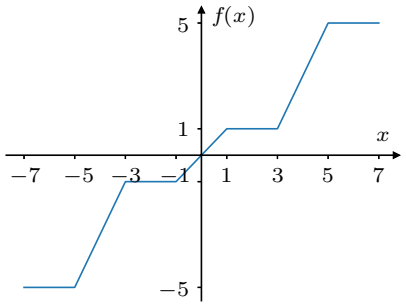


Fig. 1. Activation function graph for $k = 2$.

According to the definition of the activation function, for $i, j \in \{1,\ldots,n\}$, it can be obtained that

$$\begin{aligned} &\max f(x_i) = 4k-3, \\ &\min f(x_i) = -4k+3, \\ &\max\{f(x_i)-f(x_j)\} = 2(4k-3), \\ &\min\{f(x_i)-f(x_j)\} = -2(4k-3). \end{aligned}$$

The definitions of the equilibrium point and desired memory vector are given below:

Definition 1 Let $\boldsymbol{x}^* = (x^*_1,\ldots,x^*_n)^{\mathrm{T}}$, the vector $\boldsymbol{x}^*$ is the equilibrium point of system (1) if

$$0 = -c_i x^*_i + \sum_{j=1}^{n} a_{ij} f(x^*_i) + \sum_{j=1}^{m} e^1_{ij} u^1_j + \sum_{j=1}^{p} e^2_{ij} u^2_j + v_i.$$

Definition 2 If there exist constants $\alpha > 0$ and $\beta > 0$ such that

$$\| \boldsymbol{x}(t) - \boldsymbol{x}^* \| \le \beta \| \boldsymbol{x}(0) - \boldsymbol{x}^* \| \, \mathrm{e}^{-\alpha t}$$

holds for any $t \ge 0$, where $\boldsymbol{x}(t) = (x_1(t),\ldots,x_n(t))^{\mathrm{T}}$ is the solution of system (1) with initial state $\boldsymbol{x}(0)$. The equilibrium point $\boldsymbol{x}^*$ is globally exponentially stable.

Definition3[22] If the vector $\boldsymbol{x}^*$ is the globally exponentially stable equilibrium point of system (1), and $\boldsymbol{s} = f(\boldsymbol{x}^*)$, then the vectors $\boldsymbol{s} = (s_1,\ldots,s_n)^{\mathrm{T}}$ are the stable memory vectors.

To verify the stability of network (1), we have introduced the use of $v^h_i(l)$, $\Gamma_i(l)$, $\Gamma(l)$, v^+_i and v^-_i.

Denote

$$v^h_i(l) = h \times c_i - \sum_{j=1}^{n} a_{ij} s^{(l)}_j - \sum_{j=1}^{m} e^1_{ij} u^{1(l)}_j - \sum_{j=1}^{p} e^2_{ij} u^{2(l)}_j, \quad (3)$$

where $s_i^{(l)}$ is the output of the network for the lth pattern pair, $u_i^{1(l)}$ and $u_i^{2(l)}$ are the inputs of the network for the lth pattern pair.

$$\Gamma_i(l)=\begin{cases}(-\infty, v_i^{-(4k-3)}(l)), & s_i^{(l)}=-(4k-3),\\ (v_i^{-(4k-5)}(l), v_i^{-(4k-7)}(l)), & s_i^{(l)}=-(4k-7),\\ \cdots & \\ (v_i^{-3}(l), v_i^{-1}(l)), & s_i^{(l)}=-1,\\ (v_i^{1}(l), v_i^{3}(l)), & s_i^{(l)}=1,\\ \cdots & \\ (v_i^{(4k-7)}(l), v_i^{(4k-5)}(l)), & s_i^{(l)}=(4k-7),\\ (v_i^{(4k-3)}(l), +\infty), & s_i^{(l)}=(4k-3).\end{cases}$$

$\Gamma(l)=\prod_{i=1}^{n}\Gamma_i(l)$.

Denote

$$v_i^{+}=\min_{\substack{1\le l\le\rho\\ 1\le q\le k}}\{v_i^h(l)|h=-4q+3\},$$

$$v_i^{-}=\max_{\substack{1\le l\le\rho\\ 1\le q\le k}}\{v_i^h(l)|h=4q-3\}.$$

3. Main results

In this section, a learning algorithm for storing and retrieving memory patterns is proposed, based on the existence and convergence of the equilibrium points.

3.1. Stability

In this subsection, we will provide the conditions for the stability of the network and present the proof.

Theorem 1 If there exists $l\in\{1,\dots,\rho\}$ such that the bias vectors $\boldsymbol{v}=(v_1,\dots,v_n)^{\mathrm{T}}\in\Gamma(l)$ and

$$c_i>\sum_{j=1}^{n}|a_{ij}| \tag{4}$$

hold for $i\in\{1,\dots,n\}$, then the output vector $\boldsymbol{s}^{(l)}$ of system (1) will be globally exponentially stable with the input vectors $\boldsymbol{u}^{1(l)}$ and $\boldsymbol{u}^{2(l)}$.

Proof Denote

$$\tilde{x}_i=\frac{1}{c_i}\left(\sum_{j=1}^{n}a_{ij}s_j^{(l)}+\sum_{j=1}^{m}e_{ij}^1u_j^{1(l)}+\sum_{j=1}^{p}e_{ij}^2u_j^{2(l)}+v_i\right). \tag{5}$$

Due to $v_i(l)\in\Gamma(l)$, when $s_i^{(l)}=4k-3$, according to the definition of $\Gamma(l)$, we have

$$v_i>v_i^{(4k-3)}(l).$$

From Eq. (3), it can be obtained that

$$v_i>(4k-3)c_i-\sum_{j=1}^{n}a_{ij}s_j^{(l)}-\sum_{j=1}^{m}e_{ij}^1u_j^{1(l)}-\sum_{j=1}^{p}e_{ij}^2u_j^{2(l)}.$$

Therefore, when $s_i^{(l)}=4k-3$,

$$\tilde{x}_i>4k-3,\quad f(\tilde{x}_i)=s_i^{(l)}=4k-3.$$

Similar to the above, when $s_i^{(l)}=-(4k-3)$,

$$\tilde{x}_i<-(4k-3),\quad f(\tilde{x}_i)=s_i^{(l)}=-(4k-3).$$

For $k\ge2$, if $s_i^{(l)}=4q-3$, $q=1,\dots,k-1$, then

$$v_i^{(4q-1)}(l)>v_i>v_i^{(4q-3)}(l).$$

From Eq. (3), it can be obtained that

$$v_i>(4q-3)c_i-\sum_{j=1}^{n}a_{ij}s_j^{(l)}-\sum_{j=1}^{m}e_{ij}^1u_j^{1(l)}-\sum_{j=1}^{p}e_{ij}^2u_j^{2(l)},$$

$$v_i<(4q-1)c_i-\sum_{j=1}^{n}a_{ij}s_j^{(l)}-\sum_{j=1}^{m}e_{ij}^1u_j^{1(l)}-\sum_{j=1}^{p}e_{ij}^2u_j^{2(l)}.$$

Therefore, when $s_i^{(l)}=4q-3$,

$$\tilde{x}_i\in(4q-3,4q-1),\quad f(\tilde{x}_i)=4q-3.$$

Similarly, when $s_i^{(l)}=-(4q-3)$,

$$\tilde{x}_i\in(-4q+1,-4q+3),\quad f(\tilde{x}_i)=-(4q-3).$$

From the above, it can be obtained that for any $i\in1,\dots,n$, $f(\tilde{x}_i)=s_i^{(l)}$, that is

$$\tilde{x}_i=\frac{1}{c_i}\left(\sum_{j=1}^{n}a_{ij}f(\tilde{x}_j)+\sum_{j=1}^{m}e_{ij}^1u_j^{1(l)}+\sum_{j=1}^{p}e_{ij}^2u_j^{2(l)}+v_i\right).$$

From Definition 1, $\tilde{\boldsymbol{x}}=(\tilde{x}_1,\dots,\tilde{x}_n)^{\mathrm{T}}$ is the equilibrium point of system (1).

Denote $y_i(t)=x_i(t)-x^*$, we have

$$\dot{y}_i(t)=-c_iy_i(t)+\sum_{j=1}^{n}a_{ij}[f(x_j(t))-f(x_j^*)].$$

Let

$$H(\omega)=\omega-c_i+\sum_{j=1}^{n}|a_{ij}|. \tag{6}$$

From Eq. (6), we can obtain $H(0)=-c_i+\sum_{j=1}^{n}|a_{ij}|<0$, and there exists a constant $\lambda>0$ such that $H(\lambda)<0$.

Denote $M=\max_{1\le i\le n}\{|x_i(0)-x_i^*|\}$, $r_i(t)=\mathrm{e}^{\lambda t}|q_i(t)|$. Next we will prove $r_i(t)\le M$. Assume that $r_i(t)\le M$ is not true, then there exists t_1 such that

$$\begin{cases}r_i(t)<M, & t\in(-\infty,t_1),\\ D^+r_i(t_1)\ge0,\\ r_i(t_1)=M.\end{cases}$$

Then, we calculate

$$\begin{aligned}D^+(|y_i(t_1)|)&=\mathrm{sgn}(y_i(t_1))\dot{y}_i(t_1)\\ &=\mathrm{sgn}(y_i(t_1))[-c_iy_i(t_1)\\ &\quad+\sum_{j=1}^{n}a_{ij}(f(x_j(t_1))-f(x_j^*))]\end{aligned}$$

$$\leq -c_i|y_i(t_1)| + \sum_{j=1}^{n}|a_{ij}||x_j(t_1)-x_j^*|$$
$$\leq -c_i|y_i(t_1)| + \sum_{j=1}^{n}|a_{ij}||y_j(t_1)|.$$

From the above, we have

$$D^+r_i(t_1) = \lambda\,\mathrm{e}^{\lambda t_1}|y_i(t_1)| + \mathrm{e}^{\lambda t_1}D^+(|y_i(t_1)|)$$
$$\leq (\lambda - c_i)r_i(t_1) + \sum_{j=1}^{n}|a_{ij}|r_j(t_1)$$
$$\leq \left\{\lambda - c_i + \sum_{j=1}^{n}|a_{ij}|\right\}M < 0,$$

which is a contradiction to $D^+r_i(t_1)\geq 0$, then we have $r_i(t)\leq M$, such that

$$|x_i(t)-x_i^*| \leq \max_{1\leq i\leq n}\{|x_i(0)-x_i^*|\}\,\mathrm{e}^{-\lambda t}$$

holds for any $t\geq 0, i=1,\ldots,n$.

Therefore the equilibrium point $\boldsymbol{x}^*$ of system (1) is globally exponentially stable, and thus the output vector $\boldsymbol{s}^{(l)}$ is globally exponentially stable with the external input vectors $\boldsymbol{u}^{1(l)}$ and $\boldsymbol{u}^{2(l)}$.

In the following subsection, we will propose a many-to-one associative memory learning algorithm and a multi-input indispensability learning algorithm in many-to-one associative memory, along with the corresponding proofs.

3.2. Many-to-one associative memory

The $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1,\ldots,\lambda_n)$, $\boldsymbol{U}^1 = (\boldsymbol{u}^{1(1)},\ldots,\boldsymbol{u}^{1(\rho)}) \in \boldsymbol{R}^{m\times\rho}$, $\boldsymbol{U}^2 = (\boldsymbol{u}^{2(1)},\ldots,\boldsymbol{u}^{2(\rho)}) \in \boldsymbol{R}^{p\times\rho}$ are the external input matrices, $\boldsymbol{S} = (\boldsymbol{s}^{(1)},\ldots,\boldsymbol{s}^{(\rho)}) \in \boldsymbol{R}^{n\times\rho}$ is the output matrix, $\boldsymbol{\alpha}^1 = \mathrm{diag}(\alpha_1^1,\ldots,\alpha_n^1) \in \boldsymbol{R}^{n\times n}$ and $\boldsymbol{\alpha}^2 = \mathrm{diag}(\alpha_1^2,\ldots,\alpha_n^2) \in \boldsymbol{R}^{n\times n}$ are the input control matrices.

Theorem 2 If there exist $\boldsymbol{\Lambda}$ satisfying $1<\lambda_i<\frac{4k-6}{4k-7}$ and $0<\alpha_i^1,\alpha_i^2<1$, such that

$$\alpha_i^1 + \alpha_i^2 = 1, \tag{7}$$
$$\boldsymbol{E}^1\boldsymbol{U}^1 = \boldsymbol{\alpha}^1(\boldsymbol{\Lambda C} - \boldsymbol{A})\boldsymbol{S}, \tag{8}$$
$$\boldsymbol{E}^2\boldsymbol{U}^2 = \boldsymbol{\alpha}^2(\boldsymbol{\Lambda C} - \boldsymbol{A})\boldsymbol{S} \tag{9}$$

and Eq. (4) hold for $i\in\{1,\ldots,n\}$, then $\bigcap_{l=1}^{\rho}\Gamma(l) = \Gamma(1)\bigcap\ldots\bigcap\Gamma(\rho)$ is not empty. For any $v\in\bigcap_{l=1}^{\rho}\Gamma(l)$, the output vector $\boldsymbol{s}^{(l)}$ is globally exponentially stable with input vectors $\boldsymbol{u}^{1(l)}$ and $\boldsymbol{u}^{2(l)}$ for $l\in\{1,\ldots,\rho\}$.

Proof According to Eqs. (7) and (8), for any $i\in\{1,\ldots,n\}$ and $l\in\{1,\ldots,\rho\}$

$$\sum_{j=1}^{m}e_{ij}^1u_j^{1(l)} = \alpha_i^1\left(\lambda_ic_is_i^{(l)} - \sum_{j=1}^{n}a_{ij}s_j^{(l)}\right),$$
$$\sum_{j=1}^{m}e_{ij}^2u_j^{2(l)} = \alpha_i^2\left(\lambda_ic_is_i^{(l)} - \sum_{j=1}^{n}a_{ij}s_j^{(l)}\right).$$

If $s_i^{(r)} = 4k-3, s_i^{(o)} = -(4k-3)$ for $r,o\in\{1,\ldots,\rho\}$, we can obtain

$$\sum_{j=1}^{m}e_{ij}^1u_j^{1(r)} + \sum_{j=1}^{p}e_{ij}^2u_j^{2(r)} + \sum_{j=1}^{n}a_{ij}s_j^{(r)} = (4k-3)\lambda_ic_i,$$
$$\sum_{l=1}^{m}e_{ij}^1u_j^{1(o)} + \sum_{j=1}^{p}e_{ij}^2u_j^{2(o)} + \sum_{j=1}^{n}a_{ij}s_j^{(o)} = -(4k-3)\lambda_ic_i.$$

From $\frac{4k-6}{4k-7} > \lambda_i > 1$,

$$(4k-3)c_i - \sum_{j=1}^{m}e_{ij}^1u_j^{1(r)} - \sum_{j=1}^{p}e_{ij}^2u_j^{2(r)} - \sum_{j=1}^{n}a_{ij}s_j^{(r)}$$
$$< -(4k-3)c_i - \sum_{j=1}^{m}e_{ij}^1u_j^{1(o)} - \sum_{j=1}^{p}e_{ij}^2u_j^{2(o)} - \sum_{j=1}^{n}a_{ij}s_j^{(o)}.$$

Hence, $(v_i^{4k-3}(r),+\infty)\cap(-\infty,v_i^{-4k+3}(o))$ is not empty.

Similarly to the method above, we can prove for any $q\in\{1,\ldots,k-1\}$, $s_i^r = (4q-3)$, $s_i^o = -4q+3$, $\{v_i^{4q-3}(r), v_i^{4q-1}(r)\}\bigcap\{v_i^{-4q+1}(o), v_i^{-4q+3}(o)\}$ is not empty. With the increase of q, for $s_i^r = 4q-3$ and $s_i^r = -4q+3$, $v_i^{4q-3}(r)$ is monotonically decreasing, $v_i^{-4q+3}(o)$ is monotonically increasing. Therefore, $\bigcap_{l=1}^{\rho}\Gamma(l)$ is not empty. According to Theorem 1, the output vector $\boldsymbol{s}^{(l)}$ is globally exponentially stable with input vectors $\boldsymbol{u}^{1(l)}$ and $\boldsymbol{u}^{2(l)}$.

Remark 3 The robustness of this algorithm lies in the saturation region of the activation function. As long as the equilibrium point is within the correct saturation range, the output mapped will be the correct result. Therefore, we can determine the model's robustness to input by calculating the size of the saturation region. For example, if the correct equilibrium point is $x=1.5$ and the desired output is $s=1$, with the saturation range of the activation function being 2, this means that when the equilibrium point with added noise falls within the interval (1, 3), the final output of the neuron remains correct. We can use the length of the saturation range to calculate the specific value of robustness, similar to the approach in Ref. [22].

Remark 4 This model serves as an example of two-to-one associative memory and can be extended to many-to-one associative memory without theoretical obstacles.

The following is the parameter design for many-to-one associative memory when the input quantity is ϖ. Let $\boldsymbol{\alpha}^\varepsilon = \mathrm{diag}(\alpha_1^\varepsilon,\ldots,\alpha_n^\varepsilon)$, $\varepsilon\in\{1,\ldots,\varpi\}$ be the εth input control matrix.

Corollary 1 If there exist $\boldsymbol{\Lambda}$ satisfying $1<\lambda_i<\frac{4k-6}{4k-7}$, $\varepsilon\in\{1,\ldots,\varpi\}$, and $0<\alpha_i^\varepsilon<1$, such that

$$\sum_{\varepsilon=1}^{\varpi}\alpha_i^\varepsilon = 1, \tag{10}$$
$$\boldsymbol{E}^\varepsilon\boldsymbol{U}^\varepsilon = \boldsymbol{\alpha}^\varepsilon(\boldsymbol{\Lambda C} - \boldsymbol{A})\boldsymbol{S} \tag{11}$$

and Eq. (4) hold for $i \in \{1,\ldots,n\}$, then $\bigcap_{l=1}^{\rho} \Gamma(l) = \Gamma(1)\bigcap\ldots\bigcap\Gamma(\rho)$ is not empty. For any $v \in \bigcap_{l=1}^{\rho} \Gamma(l)$, the output vector $\boldsymbol{s}^{(l)}$ is globally exponentially stable with input vectors $\boldsymbol{u}^{1(l)}, \boldsymbol{u}^{2(l)},\ldots,\boldsymbol{u}^{\varepsilon(l)}$ for $l \in \{1,\ldots,\rho\}$.

Remark 5 The above many-to-one associative memory learning algorithm can still produce approximately correct results when one part of the input is missing. Due to the fact that the control weights of the inputs in this algorithm are between 0 and 1, each input represents a portion of the output. For example, if the network has two inputs a and b with control weights of 0.3 and 0.7 respectively, then even if input a is missing, 0.7 of the correct input still enters the network. According to global stability, the output of the network will be 70% of the correct result, meaning that each neuron outputs 70% of the correct value. In a multi-valued image, this manifests as a decrease in the image's brightness. However, in practical scenarios of many-to-one associative memory, all inputs are indispensable. To address this issue, we propose the following learning algorithm.

Denote

$$\aleph = \frac{2(4k-3)(c_i + \sum_{j=1}^{n} |a_{ij}|)}{(\lambda_i c_i - \sum_{j=1}^{n} a_{ij})}.$$

Theorem 3 If there exist $\alpha_i^1,\ \alpha_i^2 \in (-\infty,-\aleph)\cup(\aleph+1,+\infty)$ and $\boldsymbol{\Lambda}$ satisfying $1 < \lambda_i < \frac{4k-6}{4k-7}$, such that

$$\alpha_i^1 + \alpha_i^2 = 1, \tag{12}$$

$$\boldsymbol{E}^1\boldsymbol{U}^1 = \boldsymbol{\alpha}^1(\boldsymbol{\Lambda C} - \boldsymbol{A})\boldsymbol{S}, \tag{13}$$

$$\boldsymbol{E}^2\boldsymbol{U}^2 = \boldsymbol{\alpha}^2(\boldsymbol{\Lambda C} - \boldsymbol{A})\boldsymbol{S} \tag{14}$$

and Eq. (4) hold for $i \in \{1,\ldots,n\}$, $k \geq 2$, then the output vector $\boldsymbol{s}^{(l)}$ is globally exponentially stable with input vectors $\boldsymbol{u}^{1(l)}$ and $\boldsymbol{u}^{2(l)}$, if the network is missing one input, it will produce a completely incorrect output.

Proof Let

$$\Phi_i = \frac{1}{c_i}\left(\sum_{j=1}^{n} a_{ij}(f(x_j^{(l)\triangle}) - f(x_j^{(l)\triangledown})) + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right),$$

$\boldsymbol{x}^{(l)\triangle}$ be the equilibrium point when the network is not missing any input, $\boldsymbol{x}^{(l)\triangledown}$ be the equilibrium point when the network is missing one input, $s_i^{(l)}$ be the lth output pattern.

$|\Phi_i|$ represents the absolute distance between $x_i^{(l)\triangle}$ and $x^{(l)\triangledown}$. The $x_i^{(l)\triangle}$ is the correct equilibrium point, according to the activation function, so $x_i^{(l)\triangle} \in (-4k+3, 4k-3)$. When the absolute distance $|\Phi_i|$ is greater than the distance between the two extreme values of the activation function, specifically when $|\Phi_i| > 2(4k-3)$, $x_i^{(l)\triangledown}$ will necessarily fall within one of the saturation regions at the extremes of the activation function. Consequently, the output of the network will be completely incorrect when an input is missing.

We assume $\boldsymbol{u}^{1(l)}$ is missing, due to $\alpha_i^1 \in (-\infty,-\aleph)\cup(\aleph+1,+\infty)$, it can be obtained that

$$\frac{2(4k-3)(c_i + \sum_{j=1}^{n} |a_{ij}|)}{(\lambda_i c_i - \sum_{j=1}^{n} a_{ij})} < \frac{2(4k-3)(c_i + \sum_{j=1}^{n} |a_{ij}|)}{(\lambda_i c_i - \sum_{j=1}^{n} a_{ij})} + 1 < \alpha_i^1, \tag{15}$$

or

$$-\frac{2(4k-3)(c_i + \sum_{j=1}^{n} |a_{ij}|)}{(\lambda_i c_i - \sum_{j=1}^{n} a_{ij})} > \alpha_i^1. \tag{16}$$

Situation 1: For $s_i^{(l)} \in \{1,\ldots,4k-7\}$, from Eq. (15), we have

$$2(4k-3)(c_i + \sum_{j=1}^{n} |a_{ij}|) < \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right),$$

$$\frac{1}{c_i}\left(-2(4k-3)\sum_{j=1}^{n} |a_{ij}| + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right) > 2(4k-3).$$

According to the definition of the activation function, it can be obtained that

$$\frac{1}{c_i}\left(\sum_{j=1}^{n} a_{ij}(f(x_j^{(l)\triangle}) - f(x_j^{(l)\triangledown})) + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right) > \frac{1}{c_i}\left(-2(4k-3)\sum_{j=1}^{n} |a_{ij}| + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right).$$

Then it can be obtained that

$$\Phi_i > 2(4k-3). \tag{17}$$

From Eq. (16), we have

$$-2(4k-3)(c_i + \sum_{j=1}^{n} |a_{ij}|) > \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right),$$

$$\frac{1}{c_i}\left(-2(4k-3)\sum_{j=1}^{n} |a_{ij}| + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right) < -2(4k-3).$$

According to the definition of the activation function, it can be obtained that

$$\frac{1}{c_i}\left(\sum_{j=1}^{n} a_{ij}(f(x_j^{(l)\triangle}) - f(x_j^{(l)\triangledown})) + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right) < \frac{1}{c_i}\left(2(4k-3)\sum_{j=1}^{n} |a_{ij}| + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right).$$

Then it can be obtained that

$$\Phi_i < -2(4k-3). \tag{18}$$

Situation 2: For $s_i^{(l)} \in \{-(4k-7),\ldots,-1\}$, from Eq. (15), we have

$$-2(4k-3)(c_i+\sum_{j=1}^{n}|a_{ij}|) > \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right),$$
$$\frac{1}{c_i}\left(2(4k-3)\sum_{j=1}^{n}|a_{ij}| + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right) < -2(4k-3).$$

According to the definition of the activation function, it can be obtained that

$$\frac{1}{c_i}\left(\sum_{j=1}^{n} a_{ij}(f(x_j^{(l)\triangle}) - f(x_j^{(l)\triangledown})) + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right) < \frac{1}{c_i}\left(2(4k-3)\sum_{j=1}^{n}|a_{ij}| + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right).$$

Then it can be obtained that

$$\Phi_i < -2(4k-3). \tag{19}$$

From Eq. (16), we have

$$2(4k-3)(c_i+\sum_{j=1}^{n}|a_{ij}|) < \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right),$$
$$\frac{1}{c_i}\left(-2(4k-3)\sum_{j=1}^{n}|a_{ij}| + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right) > 2(4k-3).$$

According to the definition of the activation function, it can be obtained that

$$\frac{1}{c_i}\left(\sum_{j=1}^{n} a_{ij}(f(x_j^{(l)\triangle}) - f(x_j^{(l)\triangledown})) + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right) > \frac{1}{c_i}\left(-2(4k-3)\sum_{j=1}^{n}|a_{ij}| + \alpha_i^1\left(\lambda_i c_i s_i^{(l)} - \sum_{j=1}^{n} a_{ij}s_j^{(l)}\right)\right).$$

Then it can be obtained that

$$\Phi_i > 2(4k-3). \tag{20}$$

From Eqs. (17)–(20), we can obtain that

$$|\Phi_i| > 2(4k-3).$$

According to Eq. (12), when $\alpha_i^1 \in (\aleph+1,+\infty)$, it can be obtained that

$$\alpha_i^2 = 1-\alpha_i^1 \in (-\infty,-\aleph).$$

According to Eq. (12), when $\alpha_i^1 \in (-\infty,-\aleph)$, it can be obtained that

$$\alpha_i^2 = 1-\alpha_i^1 \in (\aleph+1,+\infty).$$

The same conclusion can be proven when $\boldsymbol{u}^{2(l)}$ is missing.

Then when Theorem 3 holds and the network is missing an input, the network cannot produce the correct output.

Remark 6 To achieve the indispensability of many-to-one associative memory input, meaning that the output is completely incorrect when input is missing, we move the equilibrium point's value to the extreme saturation regions of the activation function when input is missing. This way, the resulting mapping is completely incorrect. The value of k determines the number of saturation regions of the activation function. We must ensure that this property holds when the output's absolute value is set to the maximum multi-valued setting. Therefore, the number of saturation regions of the activation function must be at least one more than the number of set multi-values. For example, if we set the neuron to have four values: -5, -1, 1, and 5, the required number of saturation regions is 4. Thus, we need $k=3$ to achieve this functionality.

Corollary 2 If there exist constants $\alpha_i^\varepsilon \in (-\infty,-\aleph)\cup(\aleph+1,+\infty)$ and $\boldsymbol{\Lambda}$ satisfying $1<\lambda_i<\frac{4k-6}{4k-7}$, $\varepsilon\in\{1,\ldots,\varpi\}$, such that

$$\sum_{\varepsilon=1}^{\varepsilon}\alpha_i^\varepsilon = 1, \tag{21}$$
$$\boldsymbol{E}^\varepsilon\boldsymbol{U}^\varepsilon = \boldsymbol{\alpha}^\varepsilon(\boldsymbol{\Lambda C}-\boldsymbol{A})\boldsymbol{S} \tag{22}$$

and Eq. (4) hold for $i\in\{1,\ldots,n\}$, then the output vector $\boldsymbol{s}^{(l)}$ is globally exponentially stable with input vectors $\boldsymbol{u}^{1(l)}$, $\boldsymbol{u}^{2(l)},\ldots,\boldsymbol{u}^{\varpi(l)}$, if the network is missing any input, it will produce a completely incorrect output.

Remark 7 It is always possible to find that $\alpha_i^\varepsilon \in (-\infty,-\aleph)\cup(\aleph+1,+\infty)$ satisfies Eq. (21), then $\boldsymbol{E}^\varepsilon$ can be solved through Eq. (22). When selecting α_i^ε in actual situations, you can reserve a value of $\alpha_i^{\varepsilon*}$ to be taken, randomly pick other α_i^ε to satisfy $\alpha_i^\varepsilon \in (-\infty,-\aleph)\cup(\aleph+1,+\infty)$, and finally take the $\alpha_i^{\varepsilon*}$ to satisfy Eq. (21).

3.3. Design procedure

The following design procedure of two-to-one associative memories is based on the above.

1. Denote ρ ($\rho \le \min\{(2k)^n,(2k)^p,(2k)^m\}$) pairwise desired patterns vectors and external input vectors as $\begin{pmatrix}\boldsymbol{u}^{1(l)}\\ \boldsymbol{u}^{2(l)}\\ \boldsymbol{s}^{(l)}\end{pmatrix}$, $l\in\{1,\ldots,\rho\}$, arrange ρ sets of input and output vectors respectively to determine input matrices $\boldsymbol{U}^1$, $\boldsymbol{U}^2$ and output matrix $\boldsymbol{S}$.

2. Determine $\boldsymbol{C},\boldsymbol{A},\boldsymbol{E}^1,\boldsymbol{E}^2,\boldsymbol{\alpha}^1,\boldsymbol{\alpha}^2$ and $\boldsymbol{\Lambda}$ such that Eqs. (4), (7)–(9) or (4), (12)–(14) hold.

3. Determine bias vector v by calculating v_i^- and v_i^+, such that $v_i\in[v_i^-,v_i^+]$.

4. Synthesize the system (1) with the parameters determined above.

4. Illustrative examples

In this section, we give examples of many-to-one associative memories.

4.1. Two-to-one associative memories

This example will demonstrate the verification of the model's ability to achieve two-to-one associative memory functionality when the parameters satisfy Theorems 1 and 2.

Given $k = 129$, many-to-one associative memories are designed based on neural network (1). $k = 129$ represents an activation function with 258 levels, where the middle 256 levels represent 256 shades of gray. The two extreme levels of the activation function are used to ensure the indispensability of inputs. The input vectors are obtained by serializing 10×46 and 10×34 pixel grayscale images. The output vectors are obtained by serializing 128×128 pixel grayscale images. Cute cat corresponds to an image of a cat, and wild dog corresponds to an image of a wolf.

Calculate each parameter matrix according to the above design procedure. It can be obtained for $i, j = 1, 2, \ldots, 16384$ that

$$c_i = 1, \quad \lambda_i = 1.001, \quad a_{ij} = 0.00001,$$
$$v_i = 0, \quad \alpha_i^1 = 0.5, \quad \alpha_i^2 = 0.5.$$

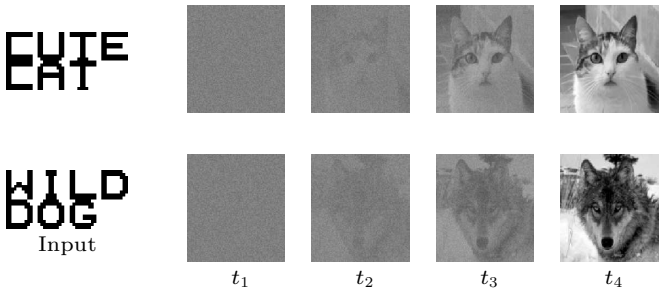


Fig. 2. The evolutions of the output image with noiseless input images under random initial states.

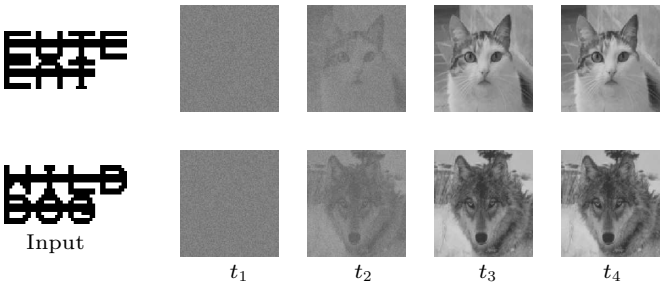


Fig. 3. The evolutions of output image with noisy input images under random initial states.

If two descriptive words are input into the model, the corresponding image can be obtained. In Fig. 2, it depicts the processes of the network recalling clear images from a random initial network neuron's state within the range of $[-50, 50]$. The results demonstrate the advantage of the IDM model over the ISDM model, where the initial neuronal state in the IDM model has no influence on the network's output. Figure 3 illustrates the network's evolution process and final output when all pixels in rows 4 to 6 of each input image are turned black (20% of the pixels are directly blackened). The results show that the final output image remains clearly recognizable, with changes primarily manifested in brightness levels, highlighting the superior performance of our model in image pattern storage and processing.

4.2. Three-to-one associative memories

This example will demonstrate the verification of the model's ability to achieve many-to-one associative memory functionality when the parameters satisfy Theorem 1 and Corollary 1.

Given $k = 129$. The input vectors are obtained by serializing 10×46 and 10×34 pixel grayscale images representing names, ages, and genders. The output vectors are obtained by serializing 112×92-pixel grayscale facial images, which were downloaded from the AT&T Laboratories Cambridge.

Calculate each parameter matrix according to the above design procedure. It can be obtained for $i, j = 1, 2, \ldots, 10304$ that

$$c_i = 1, \quad \lambda_i = 1.001, \quad a_{ij} = 0.00001,$$
$$v_i = 0, \quad \alpha_i^1 = 0.3, \quad \alpha_i^2 = 0.4, \quad \alpha_i^3 = 0.3.$$

Figure 4 presents the evolution of output patterns when three input neural states are initialized randomly.The results demonstrate that the network maintains associative memory functionality despite both triple inputs and random neuronal initialization.

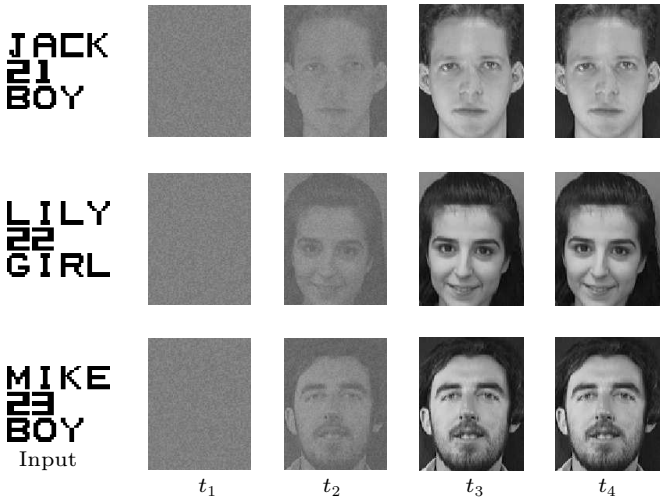


Fig. 4. The evolutions of the output faces image with noiseless input images under random initial states.

4.3. The indispensability of inputs

These examples will demonstrate the network's evolution output when one input is missing.

Calculate each parameter matrix according to the aforementioned design procedure to ensure compliance with Theorem 3. It can be obtained for $i, j = 1, 2, \ldots, 16384$ that

$$c_i = 1, \quad \lambda_i = 1.001, \quad a_{ij} = 0.00001, \quad v_i = 0,$$
$$\alpha_i^1 = -1200, \quad \alpha_i^2 = 1201 \quad \text{or} \quad \alpha_i^1 = 1201, \quad \alpha_i^2 = -1200.$$

Figure 5 illustrates the dynamic evolution of network outputs under identical parameters to those in the previous experiment but with one missing input (completely set to white pixels). Experimental results demonstrate that the network maintains certain recognizability in its outputs despite the absent input, indicating robustness against partial input loss. Figure 5 demonstrates that the learning algorithm satisfying Theorem 2 fails to achieve input indispensability.

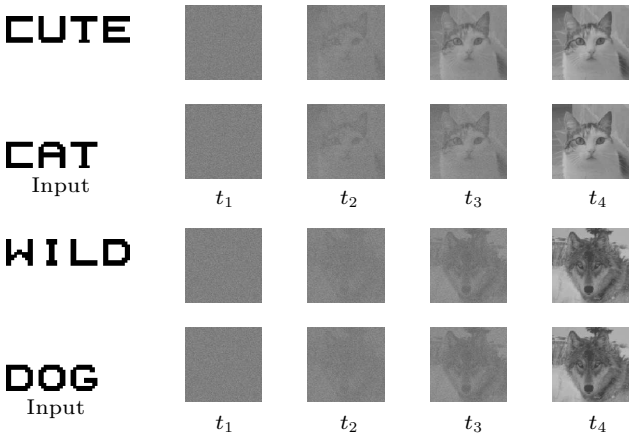


Fig. 5. The evolutions of the output image with one input missing under random initial states when parameters satisfy Theorem 2.

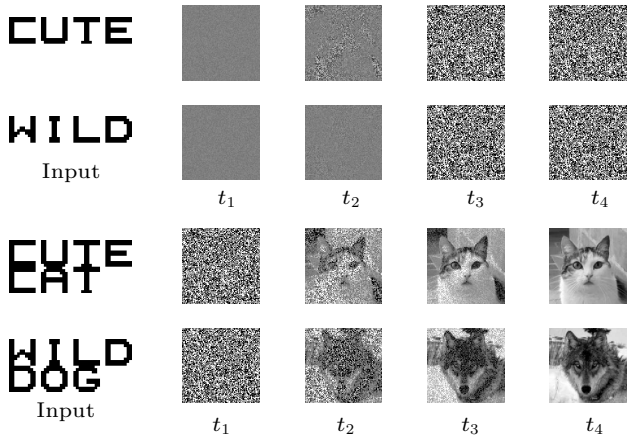


Fig. 6. The evolution of output images under random initialization with missing inputs (when parameters satisfy Theorem 3), and their recovery processes after input restoration.

In Fig. 6, the time points t_1 to t_4 depict the evolution process of the network's output images when the network parameters satisfy Theorem 3 with one missing input image (completely set to white pixels). During evolution, contours of the original image transiently emerge. This phenomenon occurs because neuronal initial states are randomly initialized within $[-50, 50]$. As evolution progresses, absolute values of neuronal states gradually increase, causing neurons with larger initial values to approach their original states, thereby briefly reconstructing image contours. However, without restoring the missing input, output images eventually degrade into unrecognizable noise. Time points t_5 to t_8 demonstrate the recovery process after reintroducing the missing input, showing the network's capability to progressively restore complete output images. Comparative analysis of Figs. 5 and 6 reveals that when network parameters satisfy Theorem 3, the network achieves input necessity in many-to-one associative memory — the missing input plays a pivotal role in output recovery.

Figure 7 further presents temporal variations in output pattern accuracy with one missing input. During t_1 to t_2, accuracy of two output patterns shows transient improvement. This occurs because output neurons' initial states (randomly initialized in $[-50, 50]$) are influenced by control parameters, causing some neuronal states to amplify and temporarily map correct outputs. From t_2 to t_4, accuracy continuously decays to zero, confirming the network's inability to maintain correct outputs without essential inputs. During t_5 to t_8, accuracy gradually recovers to 100% after supplementing the missing input pattern between t_4 and t_5, enabling complete output restoration.

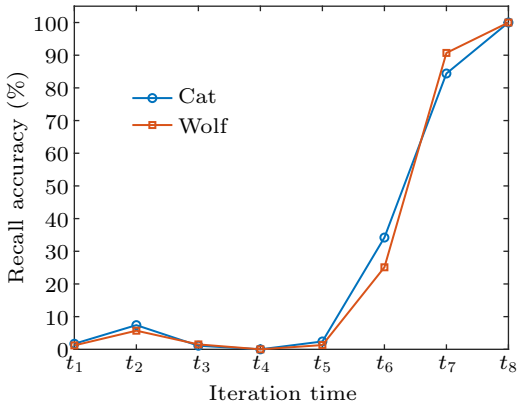


Fig. 7. In the random initial state, the accuracy of the output pattern over time with one missing input and after input restoration.

Calculate each parameter matrix according to the aforementioned design procedure to ensure compliance with Theorem 1 and Corollary 2. It can be obtained for $i, j = 1, 2, \ldots, 10304$ that

$$c_i = 1, \quad \lambda_i = 1.001, \quad a_{ij} = 0.00001, \quad v_i = 0,$$
$$\alpha_i^1 = \alpha_i^2 = -1200, \quad \alpha_i^3 = 2401 \text{ or}$$
$$\alpha_i^1 = \alpha_i^3 = -1200, \quad \alpha_i^2 = 2401 \text{ or}$$
$$\alpha_i^1 = 2401, \quad \alpha_i^2 = \alpha_i^3 = -1200.$$

Figure 8 demonstrates the temporal evolution of facial pattern outputs under conditions of missing input with random initial neuronal states during t_1–t_4. The results show that

absence of any single input invariably leads to completely erroneous outputs. During t_5–t_8, restoration of the missing input pattern enables the network output to progressively evolve from completely incorrect patterns to correct memorized patterns.

Generate 100 pattern pairs by uniform sampling, with each pair consisting of three 100-dimensional random patterns composed of $\{-1, 1\}$. Configure a three-layer MAM with 100 neurons per layer, where two layers serve as input and one layer functions as output. The missing input is set as an all-zero vector. Set $k = 2$ and calculate each parameter matrix according to the aforementioned design procedure to ensure compliance with Theorems 1 and 3. It can be obtained for $i, j = 1, 2, \ldots, 100$ that

$$c_i = 1, \quad \lambda_i = 1.001, \quad a_{ij} = 0.00001, \quad v_i = 0,$$
$$\alpha_i^1 = -3, \quad \alpha_i^2 = 4 \quad \text{or} \quad \alpha_i^1 = 4, \quad \alpha_i^2 = 3.$$

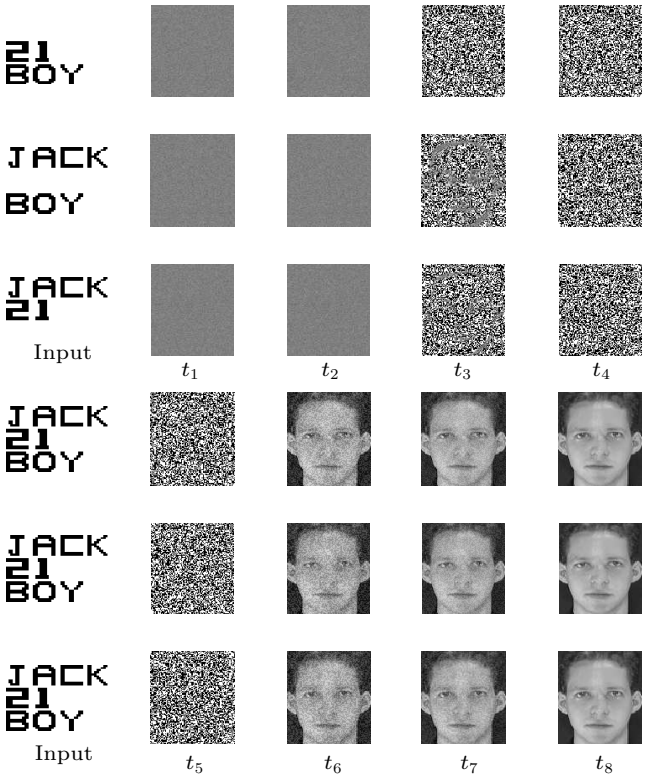


Fig. 8. The evolution of facial output images under random initialization with missing inputs (when parameters satisfy Corollary 2), and their recovery processes after input restoration.

Figure 9 demonstrates the performance comparison of input indispensability among different models under missing input conditions. The results show that the proposed model (IDMMO) effectively maintains input indispensability when processing multiple inputs with partial missing data, producing completely erroneous outputs when any input is absent. In contrast, MAM fails to fully achieve input indispensability under missing input scenarios.

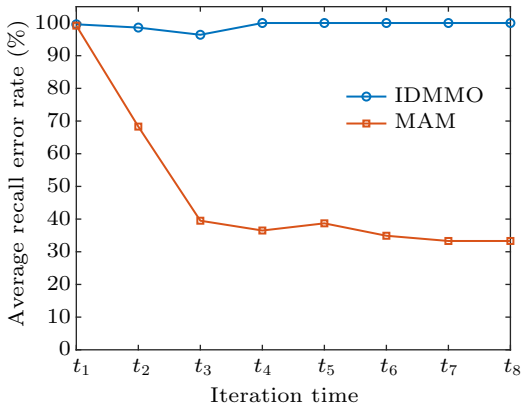


Fig. 9. Average error rate evolution of output pattern recall in many-to-one associative memory models under missing inputs, for 100 stored memory pattern pairs, plotted against iteration time.

5. Conclusion

This paper analyzes and designs a many-to-one associative memory model based on externally input-driven multivalued recurrent neural networks. By introducing an input-driven mechanism (IDM), the model effectively avoids the generation of spurious equilibrium points, thereby ensuring the accuracy and stability of output patterns. Based on stability theory and matrix analysis, we establish a series of sufficient conditions that prove the global exponential stability of equilibrium points, providing theoretical guarantees for model reliability and convergence. Furthermore, we propose a novel learning algorithm that not only enables efficient storage and retrieval of multivalued patterns but also ensures input indispensability — the model reliably produces error patterns when facing partially missing input patterns. Experimental results demonstrate the model's excellent performance in handling multivalued many-to-one associative memory tasks, exhibiting strong robustness and noise immunity. In grayscale image storage and retrieval tasks, for instance, the model accurately maps multiple input patterns to a single output pattern while maintaining high recall accuracy even with disturbed inputs. Compared with existing models, the proposed approach shows significant advantages in terms of neuron count, computational efficiency, and functional implementation. Moreover, the incorporation of the input-driven mechanism enhances the model's flexibility, enabling adaptation to more complex application scenarios.

Program availability

The source code is openly available in Science Data Bank at https://doi.org/10.57760/sciencedb.24317.

Acknowledgements

Project supported by the National Natural Science Foundation of China (Grant Nos. 62376105, 12101208, and

61906072) and the Fundamental Research Funds for the Central Universities (Grant No. 2662022XXQD001).

References

[1] Kohonen T 1972 *IEEE Trans. Comput.* **100** 353
[2] Amari S 1972 *IEEE Trans. Comput.* **100** 1197
[3] Hopfield J J 1982 *Proc. Natl. Acad. Sci. USA* **79** 2554
[4] Simas R, Sa-Couto L and Wichert A 2023 *Neurocomputing* **551** 126514
[5] Sun J, Han G, Zeng Z and Wang Y 2019 *IEEE Trans. Cybern.* **50** 2935
[6] Hu S G, Liu Y, Liu Z, Chen T P, Wang J J, Yu Q and Hosaka S 2015 *Nat. Commun.* **6** 7522
[7] Liu X, Zeng Z and Wen S 2016 *IEEE Trans. Circuits. Syst.* **63** 1454
[8] Chen X, Song Q and Li Z 2017 *IEEE Trans. Syst. Man Cybern.* **48** 2305
[9] Kosko B 1988 *IEEE Trans. Syst. Man Cybern.* **18** 49
[10] Hagiwara M 1990 *Proceedings of the International Joint Conference on Neural Networks* **1** 3
[11] Park J, Kim H Y, Park Y and Lee S W 2001 *Neural Netw.* **14** 107
[12] Liu D and Lu Z 1997 *IEEE Trans. Neural Netw.* **8** 1468
[13] Li F, Li A Q, Gan Q D and Ma H Y 2024 *Chin. Phys. B* **33** 040307
[14] Shao X G, Zhang J and Lu Y J 2024 *Chin. Phys. B* **33** 070203
[15] Lu J L, *et al.* 2025 *Chin. Phys. B* **34** 018703
[16] Xi Y G, Yu Y G, Zhang S and Hai X D 2018 *Chin. Phys. B* **27** 010202
[17] Qi A X, Zhu B D and Wang G Y 2022 *Chin. Phys. B* **31** 020502
[18] Zeng Z G and Wang J 2007 *Neural Comput.* **19** 2149
[19] Michel A N, Farrell J A and Sun H F 1990 *IEEE Trans. Circuits. Syst.* **37** 1356
[20] Lu Z and Liu D 1998 *IEEE Trans Circuits Syst* **45** 1601
[21] Grassi G 1997 *IEEE Trans. Circuits. Syst.* **44** 835
[22] Zeng Z G and Wang J 2008 *IEEE Trans. Syst. Man Cybern.* **38** 1525
[23] Sha C and Zhao H 2017 *Neurocomputing* **266** 433
[24] Zhang H, Huang Y, Wang B and Wang Z 2014 *Neurocomputing* **136** 337
[25] Shang Y 2015 *Math. Probl. Eng.* **2015** 1
[26] Zhou C, Zeng X, Yu J and Jiang H 2016 *Neurocomputing* **186** 44
[27] Huang Y J, Wang X Y, Long H X and Yang X H 2016 *Chin. Phys. B* **25** 120701
[28] Zhang J, Zhu S, Bao G, Liu X and Wen S 2021 *IEEE Trans. Cybern.* **52** 12989
[29] Nie X B and Zheng W X 2016 *IEEE Trans. Neural Netw. Learn. Syst.* **46** 679
[30] Guo Z Y, Liu L L and Wang J 2019 *IEEE Trans. Neural Netw. Learn. Syst.* **30** 2052
[31] Guo Z Y, Liu L L and Wang J 2020 *IEEE Trans. Syst. Man Cybern.* **50** 4458
[32] Johnson M and Chartier S 2014 *Proceedings of the International Conference on Artificial General Intelligence* **1** 53
[33] Hattori M and Hagiwara M 1998 *Neurocomputing* **19** 99
[34] Li Y, Li J, Duan S, Wang L and Guo M 2021 *Neurocomputing* **454** 382
[35] Yano Y and Osana Y 2009 *Proceedings of the International Conference on Artificial General Intelligence* **43** 3444
[36] Sudo A, Sato A and Hasegawa O 2007 *Proceedings of the International Conference on Artificial General Intelligence* **11** 619
[37] Shen F, Ouyang Q, Ksai W and Hasegawa O 2013 *Neurocomputing* **104** 57
[38] Masuyama N, Loo C K, Seera M and Kubota N 2018 *IEEE Trans. Neural Netw. Learn. Syst.* **29** 1058
[39] Du S, Zhang Z, Li J, Sun C, Sun J and Hong Q 2023 *IEEE Trans. Biomed Circuits. Syst.* **17** 433
[40] Zhang Y, Lv J and Zeng Z 2022 *IEEE Trans. Cybern.* **53** 7844
[41] Wang W, Sun Y, Yuan M, Wang Z, Cheng J, Fan D and Wang C 2021 *Chaos Soliton. Fract.* **150** 111110
[42] Wang S, Gao M, Wu H, Luo F, Jiang F and Tao L 2024 *Appl. Soft Comput.* **167** 112381