

Nguyễn Quang Hoan<sup>1</sup>, Vũ Thị Thềm<sup>2</sup>, Bùi Đình Quân<sup>3</sup><sup>1</sup> Trường Đại học Sư phạm Kỹ thuật Hưng Yên<sup>2</sup> Trung tâm GDTX-HN Gia Lộc<sup>3</sup> UBND Huyện Đông Hưng

Ngày nhận: 05/01/2017

Ngày sửa chữa: 10/02/2017

Ngày xét duyệt: 02/03/2017

**Tóm tắt:**

Mạng nơon nhân tạo được mô phỏng dựa theo não người. Mạng nơon nhân tạo có khả năng nhớ giống não người không? Bài báo này cho tóm tắt cấu trúc, luật học, điều kiện ổn định và khả năng nhớ mẫu của các mạng nơon Hopfield, BAM (Bidirectional Associative Memory), hai trong các mạng nơon hồi quy điển hình. Tiếp đó, bài báo tiến hành thử nghiệm khả năng nhớ mẫu, khả năng chịu lỗi ở các mức độ sai số khác nhau. Ví dụ ứng dụng của các mạng hồi quy cho bài toán nhận mẫu và gán nhãn cũng được trình bày.

**Từ khóa:** Mạng nơon hồi quy, mạng nơon Hopfield, mạng nơon BAM, nhớ mẫu, luật học.

**1. Giới thiệu**

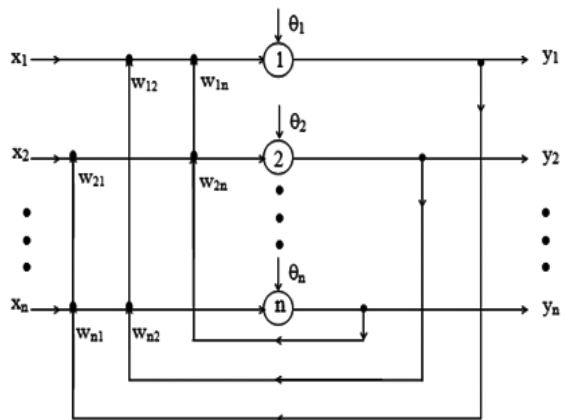
Mạng nơon nhân tạo được phân làm hai nhóm chính: mạng truyền thẳng và phản hồi (hay mạng hồi quy). Nhiều công trình nghiên cứu tính ổn định của mạng hồi quy [1, 2, 3, 5] nhằm xác định miền các tham số tránh phản hồi dương làm mất tính ổn định. Một số nghiên cứu cho biết mạng nơon Hopfield, có khả năng thực hiện nhớ tự-liên-kết (Auto-Associative Memory) - khả năng phục hồi một phần dữ liệu đã được học [1]. Hopfield (năm 1982) bằng thực nghiệm cho biết dung lượng nhớ  $C = 0.15n$  ( $n$  là số nơon. Theo McEliece, 1987):  $C = (1-2\mu)^2/4lnn$ ;  $0 \leq \mu < 1/2$ . Theo Amit, DJ (1989 *Modelling Brain Functions. The World Attractor Neural Networks*)  $C = 1/2lnn$ . Đây chính là hạn chế khả năng nhớ của mạng Hopfield: khả năng nhớ của nó nhỏ hơn nhiều so với bộ nhớ gốc (Fundamental Memories) tức là bộ nhớ thu được từ việc học.

Mạng liên kết hai chiều (BAM) được Bart Kosko [6] đề xuất năm 1988, gồm hai mạng Hopfield đầu ngược chiều nhau, không ngưỡng ( $\theta_i = 0$ ) là bộ nhớ liên-kết khác-mẫu (Hetero-Associative); nghĩa là, khi đưa mẫu  $X$  ( $n$  phần tử) vào BAM nó nhận (ở lớp ra) một mẫu  $Y$  khác kích thước ( $m$  phần tử) so với mẫu vào. BAM có khả năng nhớ  $C = \min(n, m)$  phụ thuộc vào cặp mẫu  $\{X, Y\}$ .

Tuy nhiên, trong các công trình chưa cho khả năng nhớ khi có sai số 1,2,...,  $n$  đặc trưng (hay 1,2,...,  $n$  bit). Bài báo này ngoài việc phân tích tính ổn định của 2 mạng còn nêu khả năng nhớ của các mẫu đã được học, và khả năng nhớ của các mẫu sai số chưa được học.

**2. Cấu trúc và luật học của mạng hồi quy****2.1. Mạng Hopfield****2.1.1. Cấu trúc của mạng Hopfield**

Hopfield đề xuất cấu trúc mạng rời rạc một lớp (Hình 1) (hay mạng hồi quy).



Hình 1. Cấu trúc mạng Hopfield rời rạc

$$x_i(t) = \sum_{j=1}^n w_{ij}y_j(t) - \theta_i; \quad i, j = 1, \dots, n; \quad (1)$$

trong đó,  $x_i(t)$ ,  $y_i(t)$ : tương ứng là đầu vào đầu ra của nơon thứ  $i$ ;  $w_{ij}$ : trọng liên kết phản hồi từ nơon  $i$  tới nơon  $j$ ;  $\theta_i$ : ngưỡng hay độ lệch (là hằng số) của nơon  $i$ ;  $n$ : là số nơon.

**2.1.2. Luật học của mạng Hopfield**

Phương trình mô tả luật cập nhật đầu ra mạng Hopfield [1]:

$$y_i(t+1) = \begin{cases} g(x_i(t)) & \text{khi } x_i(t) \neq 0, i = p \\ y_i(t) & \text{khi } x_i(t) = 0, i \neq p \end{cases} \quad (2)$$

Quan hệ vào/ra là hàm bước nhảy đơn vị:

$$g(x_i(t)) = \begin{cases} 1 & \text{khi } x_i(t) > 0 \\ 0 & \text{khi } x_i(t) < 0 \end{cases} \quad (3)$$

Nhiều trường hợp thay (3) có thể dùng hàm dấu:  $g(x_i(t)) = \text{sgn}(x_i(t))$ . Khi dùng hàm (3), luật học trạng số của mạng Hopfield là:

$$w_{ij} = \begin{cases} \sum_{p=1}^h (2y_{p,i} - 1)(2y_{p,j} - 1) & \text{ khi } i \neq j \\ 0, & i = j \end{cases} \quad (4)$$

$$\theta_i = \frac{1}{2} \sum_{j=1}^n w_{ij} \quad (5)$$

trong đó,  $h$  là tổng số mẫu được học;  $p$  là mẫu học thứ  $p$  đang tác động.

### 2.1.3. Tính ổn định của mạng Hopfield

Tính ổn định của mạng được Hopfield đề xuất thông qua hàm năng lượng (luôn dương)

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1, j \neq i}^n w_{ij} y_i y_j + \sum_{i=1}^n \theta_i y_i \quad (6)$$

Nếu  $w_{jj} = w_{ij} = 0$ ;  $w_{ij} = w_{ji}$ ,  $E$  giảm dần về giá trị cực tiểu (hoặc bằng 0) với mỗi bước thay đổi của đầu ra  $y_i(t)$ , khi đó:

$$\Delta E = -[y_p(t+1) - y_p(t)] \left[ \sum_{j=1}^n w_{pj} y_j - w_p \right] \quad (7)$$

Khi hàm năng lượng đạt cực tiểu (có thể là cực tiểu địa phương) mạng sẽ ổn định [9]. Vì thế, tại trạng thái khởi tạo bất kì, mạng Hopfield luôn hội tụ về một trạng thái, sau một số bước cập nhật hữu hạn, trong đó, tất cả các trạng thái ổn định về một giá trị cực tiểu của hàm  $E$ . Nguyên tắc này vận dụng lý thuyết ổn định Lyapunov, một công cụ hữu dụng cho lý thuyết về mạng nơron.

### 2.1.4. Khả năng nhớ của mạng Hopfield

Mạng Hopfield có khả năng nhớ, hồi phục mẫu [7] theo nguyên tắc không đồng bộ. Giả sử cho vector mẫu  $x^k$  với giá trị lưỡng cực cần được lưu trữ ( $k = 1, 2, \dots, p$ : mẫu). Luật cập nhật trọng số ở dạng véc to-ma trận (chính là luật học (4) khi thay  $x$  bởi  $y$ ) tức là:

$$W = \sum_{k=1}^p x^k (x^k)^T - pI \quad (8)$$

trong đó,  $x^k$ : vector  $n$  chiều;  $I$ : ma trận đơn vị  $n \times n$ ;  $(x^k)^T$  là chuyển vị của  $x^k$ . Bài toán đặt ra: giả sử cần lưu trữ 3 vector mẫu ( $p = 3$ ) hay đặc trưng  $x^k = x^3 = \{x^1, x^2, x^3\} = \{D, E, G\}$  với  $D = [1 \ 1 \ 1 \ 1 \ 1]$ ,  $E = [0 \ 1 \ 1 \ 0 \ 1]$ ,  $G = [1 \ 0 \ 0 \ 1 \ 0]$  Xác định khả năng nhớ của mạng Hopfield.

**Bước 1:** Do (8) mất thông tin khi nhân với 0, nên đổi 0 thành -1 theo (4) mà không làm mất ý nghĩa của luật học, tức  $D' = [1 \ 1 \ 1 \ 1 \ 1]$ ,  $E' = [-1 \ 1 \ 1 \ -1 \ 1]$ ;  $G' = [1 \ -1 \ -1 \ 1 \ -1]$ .

**Bước 2:** Học (ma trận trọng số)  $W$  theo (8)

$$W = D'^T D' + E'^T E' + G'^T G'$$

$$W = \begin{bmatrix} 3 & -1 & -1 & 3 & -1 \\ -1 & 3 & 3 & -1 & 3 \\ -1 & 3 & 3 & -1 & 3 \\ 3 & -1 & -1 & 3 & -1 \\ -1 & 3 & 3 & -1 & 3 \end{bmatrix}$$

### Bước 3: Xác định khả năng nhớ của mạng.

Lập Bảng 1, trong đó ba mẫu học gốc ký hiệu lại là  $D_0$ ;  $E_0$ ;  $G_0$ . Các mẫu có chỉ số từ 1 đến 5 ám chỉ các mẫu đó lần lượt sai số với mẫu gốc 1 đến 5 bits. Cột 2: số mẫu cần tính, trong đó: dòng 1 chỉ có một mẫu gốc; cột 2, dòng 2: sai số 1 bit so với mẫu gốc, lần lượt hoán vị bit sai, trở thành 5 mẫu; cột 2, dòng 3, 4, 5 cũng lập luận tương tự. Các cột còn lại là số liệu thống kê. Ví dụ, tính:

Bảng 1. Nhận mẫu sử dụng mạng Hopfield

Vector đặc trưng	Số lượng mẫu	Số mẫu nhận đúng	Số mẫu không nhận	Số mẫu nhận nhầm	(%) mẫu nhận đúng	(%) mẫu không nhận	(%) mẫu nhận nhầm
$D_0$	1	1	0	0	100	0	0
$D_1$	5	3	2	0	60	40	0
$D_2$	10	0	9	1	0	90	10
$D_3$	10	0	6	4	0	60	40
$D_4$	5	0	0	5	0	0	100
$D_5$	1	1	0	0	100	0	0
$E_0$	1	1	0	0	100	0	0
$E_1$	5	3	2	0	60	40	0
$E_2$	10	3	6	1	30	60	10
$E_3$	10	0	6	4	0	60	40
$E_4$	5	0	3	2	0	60	40
$E_5$	1	0	0	1	0	0	100
$G_0$	1	1	0	0	100	0	0
$G_1$	5	2	3	0	40	60	0
$G_2$	10	0	6	4	0	60	40
$G_3$	10	0	6	4	0	60	40
$G_4$	5	0	2	3	0	40	60
$G_5$	1	0	0	1	0	0	100

- Khi mẫu vào:  $D_0 = D = [1 \ 1 \ 1 \ 1 \ 1]$ ; đầu ra  $Y^0 = DW^T = [1 \ 1 \ 1 \ 1 \ 1]W^T = [0 \ 4 \ 4 \ 0 \ 4] \Rightarrow [1 \ 1 \ 1 \ 1 \ 1] = D$ . Đây đúng là mẫu đã học (bảng 1);

- Khi mẫu vào sai số 1 bit:  $D_1 = [0 \ 1 \ 1 \ 1 \ 1]$  (sai với gốc 1 bit đầu tiên); khi đó đầu ra  $Y^1 = D_1 W^T = [0 \ 1 \ 1 \ 1 \ 1]W^T = [0 \ 5 \ 5 \ -3 \ 5] \Rightarrow [1 \ 1 \ 0 \ 1] \neq D$ . Tính toán tương tự cho tất cả 96 mẫu, thu được Bảng 1 về khả năng nhận mẫu.

**Bước 4. Đánh giá.** Từ Bảng 1, khi mẫu vào:

- Trùng với mẫu đã được học, tỷ lệ nhận mẫu đúng 100%, cho độ chính xác cao.
- Khác với các mẫu gốc 1 đặc trưng (1 bit), tỷ lệ nhận mẫu đúng đạt từ (40%-60%)
- Khác với các mẫu gốc 2 đặc trưng (2 bit), tỷ lệ nhận mẫu đúng đạt từ (0%-30%)
- Khác với các mẫu gốc 3 đặc trưng (3 bit) trở lên, tỷ lệ nhận mẫu đúng 0%. Một vài trường hợp các biệt, có khả năng nhận đúng.

**2.2. Mạng nơron BAM**

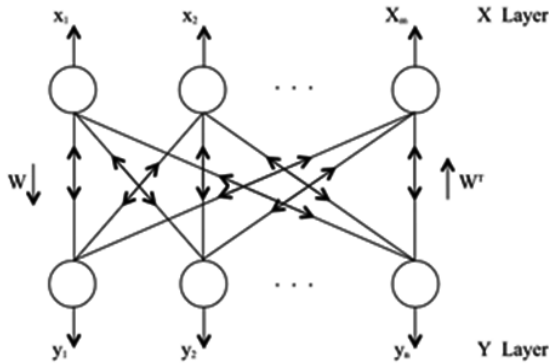
**2.2.1. Cấu trúc của mạng nơron BAM**

Cấu trúc mạng BAM gồm hai tầng tương tác  $X$  (đầu vào),  $Y$  (đầu ra) và có thể mô tả như:

$$y'_i = g(Wx); y'_i = g\left(\sum_{j=1}^m w_{ij}x_j\right); i = 1, 2 \dots n$$

$$x'_j = g(W^T y'); x'_j = g\left(\sum_{i=1}^n w_{ji}y'_i\right); j = 1, 2 \dots m$$

(9)



Hình 2. Cấu trúc mạng BAM

trong đó,  $g(\cdot)$  là hàm tương tác đầu ra, bị chặn theo (2) và (3). Vector ra  $y'$  làm đầu vào cho lớp  $X$  và vector  $x'$  dùng làm đầu vào cho lớp  $Y$ . Quá trình sẽ tiếp tục cho tới khi việc cập nhật  $x$  và  $y$  dừng (ổn định). Trạng thái cập nhật có thể đồng bộ hoặc không đồng bộ.

**2.2.2. Luật học của mạng BAM**

Với  $p$  cặp vector liên kết lưu trữ trong BAM:  $\{(x^1, y^1), (x^2, y^2), \dots, (x^p, y^p)\}$ , luật học trọng số được xác định theo [8, 9]:

$$W_{ij} = \sum_{k=1}^p y_i^k x_j^k$$

(10)

So sánh luật học của mạng BAM (10) với luật học của mạng Hopfield (4) cho thấy dữ liệu học của BAM là cặp dữ liệu vào/ra. Cả hai luật học đều dựa trên tiên đề học Hebb và thuộc loại học không có tín hiệu chỉ đạo.

**2.2.3. Ổn định của mạng BAM**

Tương tự mạng Hopfield, ổn định của BAM đảm bảo, thông qua hàm năng lượng [4]:

$$E(x, y) = -\frac{1}{2}x^T W^T y - \frac{1}{2}y^T W x = -y^T W x$$

(11)

Hoàn toàn có thể chứng minh BAM ổn định:

a)  $\Delta E$  gây ra bởi thay đổi  $\Delta y$ . Theo (11):

$$\Delta E y_i = \frac{\partial E}{\partial y_i} \Delta y_i = W x \Delta y_i = -\left(\sum_{j=1}^m w_{ij} x_j\right) \Delta y_i$$

(12)

Xét ba trường hợp sau:

- $y_i^{(k)} = -1$  và  $y_i^{(k+1)} = +1 \rightarrow \sum_{j=1}^m w_{ij} x_j > 0, \Delta y_i = 2$ ; do đó  $\Delta E_{y_i} < 0$ .
- $y_i^{(k)} = +1$  và  $y_i^{(k+1)} = -1 \rightarrow \sum_{j=1}^m w_{ij} x_j < 0, \Delta y_i = -2$ . Do đó,  $\Delta E_{y_i} < 0$

•  $y_i^{(k)} = y_i^{(k+1)} \rightarrow \Delta y_i = 0$ . Do đó,  $\Delta E_{y_i} = 0$ .

b)  $\Delta E$  gây ra bởi thay đổi  $\Delta x$ . Theo (11):

$$\Delta E x_i = \frac{\partial E}{\partial x_i} \Delta x_i = W y \Delta x_i = -\left(\sum_{j=1}^n w_{ji} y_j\right) \Delta x_i$$

(13)

Lập luận tương tự như  $\Delta y$ , dẫn đến  $\Delta E_{x_i} < 0$ . Điều này chứng tỏ: sau một số bước, hàm  $E$  (đương) giảm dần (do  $\Delta E_{x_i} < 0$  và  $\Delta E_{y_i} < 0$ ) về một giá trị cực tiểu, dẫn tới mạng ổn định.

**2.2.4. Khả năng nhớ mẫu của mạng BAM**

Để xác định khả năng nhớ mẫu của BAM, bằng thực nghiệm, ta tiến hành như sau [8]:

**Bước 1:** cho cặp dữ liệu học  $\{X, Y\}$  với ba mẫu đầu vào (Vector đặc trưng): A, B, C tương ứng với nó là ba mẫu đầu ra (Vector nhãn): D, E, G như Bảng 2:

Bảng 2. Dữ liệu vào/ra cho mạng BAM

Vector đặc trưng	Vector nhãn
A=[1 0 1 0 1]	D=[1 1 1 1 1]
A'=[1 -1 1 -1 1]	D'=[1 1 1 1 1]
B=[1 0 1 0 0]	E=[0 1 1 0 1]
B'=[1 -1 1 -1 -1]	E'=[-1 1 1 -1 1]
C=[0 1 0 1 1]	G=[1 0 0 1 0]
C'=[-1 1 -1 1 1]	G'=[1 -1 -1 1 -1]

**Bước 2:** Học (ma trận trọng số)  $W$  theo (10)

$$W = D^T A' + E^T B' + G^T C'$$

$$W = \begin{bmatrix} -1 & 1 & -1 & 1 & 3 \\ 3 & -3 & 3 & -3 & -1 \\ 3 & -3 & 3 & -3 & -1 \\ -1 & 1 & -1 & 1 & 3 \\ 3 & -3 & 3 & -3 & -1 \end{bmatrix}$$

Bảng 3. Nhận mẫu sử dụng mạng BAM

Vector đặc trưng	Số lượng mẫu	Số mẫu nhận đúng	Số mẫu không nhận	Số mẫu nhận nhầm	(%) mẫu nhận đúng	(%) mẫu không nhận	(%) mẫu nhận nhầm
A <sub>0</sub>	1	1	0	0	100	0	0
A <sub>1</sub>	5	4	0	1	80	0	20
A <sub>2</sub>	10	0	0	10	0	0	100
A <sub>3</sub>	10	6	0	4	60	0	40
A <sub>4</sub>	5	0	0	5	0	0	100
A <sub>5</sub>	1	0	0	1	0	0	100
B <sub>0</sub>	1	1	0	0	100	0	0
B <sub>1</sub>	5	4	0	1	80	0	20
B <sub>2</sub>	10	0	0	10	0	0	100
B <sub>3</sub>	10	0	0	10	0	0	100

$B_4$	5	0	0	5	0	0	100
$B_5$	1	0	0	1	0	0	100
$C_0$	1	1	0	0	100	0	0
$C_1$	5	5	0	0	100	0	0
$C_2$	10	10	0	0	100	0	0
$C_3$	10	0	0	10	0	0	100
$C_4$	5	0	0	5	0	0	100
$C_5$	1	0	0	1	0	0	100

### Bước 3: Khả năng nhớ của mạng BAM.

Tính tương tự như mạng Hopfield cho 96 cặp mẫu, ta có khả năng nhớ của mạng BAM (Bảng 3) với các sai số. Ví dụ:

• Vào:  $A_1=[0\ 0\ 1\ 0\ 1]$ ;  $A_1.W^T=[0\ 0\ 10\ 1].W^T=[2\ 2\ 2\ 2\ 2] \Rightarrow [1\ 1\ 1\ 1\ 1]=D$ ;

• Vào:  $A_2=[0\ 1\ 1\ 0\ 1]$ ;  $A_2.W^T=[0\ 1\ 1\ 0\ 1].W^T=[3\ -1\ -1\ 3\ -1] \Rightarrow [1\ 0\ 0\ 1\ 0] \neq D$ .

**Bước 4. Đánh giá.** Từ Bảng 3, khi mẫu vào:

• Trùng với mẫu đã học, tỷ lệ nhận mẫu đúng 100%, cho độ chính xác tuyệt đối;

• Khác với các mẫu gốc 1 đặc trưng (1 bit), tỷ lệ nhận mẫu đúng đạt (80%-100%);

• Khác với các mẫu gốc 2 đặc trưng (2 bits), tỷ lệ nhận mẫu đúng đạt (0%-100%);

• Khác với các mẫu gốc 3 đặc trưng (3 bits) trở lên, tỷ lệ nhận mẫu đúng đạt (0%-60%);

• Khác với các mẫu gốc 4 đặc trưng (4 bits) trở lên, tỷ lệ nhận mẫu đúng 0%.

### 2.3. So sánh khả năng nhớ của hai mạng

• Hai mạng Hopfield, BAM đều thuộc nhóm nơron hồi quy, cần điều kiện ổn định;

• Khả năng nhớ và chịu lỗi của mạng nơron BAM cao hơn mạng nơron Hopfield;

• Ma trận học mạng Hopfield vuông, chỉ nhận mẫu của chính nó (tự liên kết). Ma trận học của mạng BAM (hai lớp) có dạng chữ nhật; mẫu vào (đặc trưng) và mẫu ra (nhãn) phân biệt, tiện cho nhận dạng và gán nhãn.

### 3. Ứng dụng BAM nhận dạng khuôn mặt

Bảng 4. Dữ liệu cho nhận dạng, gán nhãn

Vector đặc trưng	Vector nhãn
$A=[1\ 0\ 0\ 1\ 1]$	$H=[0\ 1\ 1\ 1]$
$A'=[1\ -1\ -1\ 1\ 1]$	$H'=[-1\ 1\ 1\ 1]$
$B=[0\ 1\ 0\ 1\ 0]$	$I=[1\ 0\ 1\ 1]$
$B'=[-1\ 1\ -1\ 1\ -1]$	$I'=[1\ -1\ 1\ 1]$
$C=[1\ 0\ 1\ 0\ 1]$	$J=[0\ 1\ 0\ 1]$
$C'=[1\ -1\ 1\ -1\ 1]$	$J'=[-1\ 1\ -1\ 1]$
$D=[0\ 1\ 1\ 0\ 1]$	$K=[0\ 0\ 0\ 1]$

$D'=[-1\ 1\ 1\ -1\ 1]$	$K'=[-1\ -1\ -1\ 1]$
$E=[1\ 0\ 0\ 1\ 0]$	$L=[1\ 1\ 1\ 0]$
$E'=[1\ -1\ -1\ 1\ -1]$	$L'=[1\ 1\ 1\ -1]$
$F=[0\ 1\ 0\ 1\ 1]$	$M=[0\ 0\ 1\ 1]$
$F'=[-1\ 1\ -1\ 1\ 1]$	$M'=[-1\ -1\ 1\ 1]$
$G=[1\ 1\ 0\ 0\ 0]$	$N=[1\ 1\ 1\ 1]$
$G'=[1\ 1\ -1\ -1\ -1]$	$N'=[1\ 1\ 1\ 1]$

### Bài toán nhận dạng khuôn mặt, gán nhãn:

Cho 7 (dạng) mẫu (vào) khuôn mặt để học:  $A$ ="Mặt thuận";  $B$ ="Mặt trái tim";  $C$ ="Mặt chữ điền";  $D$ ="Mặt trái xoan";  $E$ ="Mặt tròn";  $F$ ="Mặt tam giác";  $G$ ="Mặt kim cương"; mỗi mẫu vào gồm 5 đặc trưng: (1: Chiều cao; 2: Chiều rộng khuôn mặt; 3: Kích thước trán; 4: Hai góc phía trên khuôn mặt tròn (nhận giá trị 1; nếu không, nhận giá trị 0); 5: Hai góc phía dưới khuôn mặt tròn (nhận giá trị 1 (nếu không tròn, nhận giá trị 0)); đã được mã hóa nhị phân. Bảy mẫu vào  $A, B, C, D, E, F, G$  tương ứng với 7 mẫu ra  $H, I, J, K, L, M, N$ , tức tên người (hay nhãn) được nhị phân hóa như Bảng 4. Cần sử dụng mạng nơron BAM để nhận dạng và gán nhãn.

Bảng 5. Nhận dạng và gán nhãn dùng BAM

Vector đặc trưng	Số lượng mẫu	Số mẫu nhận đúng	Số mẫu không nhận	Số mẫu nhận nhầm	(%) mẫu nhận đúng	(%) mẫu không nhận	(%) mẫu nhận nhầm
$A_0$	1	1	0	0	100	0	0
$A_1$	5	1	0	4	20	0	80
$A_2$	10	0	2	8	0	20	80
$A_3$	10	0	2	8	0	20	80
$A_4$	5	0	1	4	0	20	80
$A_5$	1	0	0	1	0	0	100
$B_0$	1	1	0	0	100	0	0
$B_1$	5	1	1	3	20	20	60
$B_2$	10	0	1	9	0	10	90
$B_3$	10	0	2	8	0	20	80
$B_4$	5	0	1	4	0	20	80
$B_5$	1	0	0	1	0	0	100
$C_0$	1	1	0	0	100	0	0
$C_1$	5	4	1	0	80	20	0
$C_2$	10	2	2	6	20	20	60
$C_3$	10	0	1	9	0	10	90
$C_4$	5	0	1	4	0	20	80
$C_5$	1	0	0	1	0	0	100
$D_0$	1	1	0	0	100	0	0

$D_1$	5	4	0	1	80	0	20
$D_2$	10	4	1	5	40	10	50
$D_3$	10	0	2	8	0	20	80
$D_4$	5	0	2	3	0	40	60
$D_5$	1	0	0	1	0	0	100
$E_0$	1	1	0	0	100	0	0
$E_1$	5	1	2	2	20	40	40
$E_2$	10	0	2	8	0	20	80
$E_3$	10	0	1	9	0	10	90
$E_4$	5	0	0	5	0	0	100
$E_5$	1	0	0	1	0	0	100
$F_0$	1	1	0	0	100	0	0
$F_1$	5	1	0	4	20	0	80
$F_2$	10	0	1	9	0	10	90
$F_3$	10	0	1	9	0	10	90
$F_4$	5	0	2	3	0	40	60
$F_5$	1	0	1	0	0	100	0
$G_0$	1	1	0	0	100	0	0
$G_1$	5	1	0	4	20	0	80
$G_2$	10	2	1	7	20	10	70
$G_3$	10	0	3	7	0	30	70
$G_4$	5	0	1	4	0	20	80
$G_5$	1	0	0	1	0	0	100

**Bước 1:** Đổi (Bảng 4) các giá trị 0 thành -1:  $A', B', C', D', E', F', G'$  và  $H', I', J', K', L', M', N'$ ;

**Bước 2:** Học (ma trận trọng số)  $W$  theo (10)  
 $W = H^T A' + I^T B' + J^T C' + K^T D' + L^T E' + M^T F' + N^T G'$

$$W = \begin{bmatrix} 1 & 1 & -3 & 1 & -7 \\ 7 & -5 & -1 & -1 & -1 \\ 1 & 1 & -7 & 5 & -3 \\ -1 & 3 & -1 & -1 & 3 \end{bmatrix}$$

Khác với 2.2.3, chiều ma trận  $W$  trong trường hợp này là 4x5.

**Tài liệu tham khảo**

[1]. Nguyễn Quang Hoan (2007), *Mạng nơron nhân tạo*, Học viện Công nghệ Bưu chính Viễn thông.  
 [2]. Chin Teng Lin, C.S. George Lee (2002), *Neural Fuzzy Systems, A Neural-Fuzzy Synergism to Intelligent Systems*, Prentice Hall International, Inc.  
 [3]. Michael Negnevitsky (2002), *Artificial Intelligence. A Guide to Intelligent Systems*. Addison Wesley. Pearson Education.  
 [4]. Xiaomei Wang, Shouming Zhong (2010), *Globally Exponential Stability of Periodic Solution of BAM Neural Network with Distributed Time Delays and Impulses*, 2010 International Conference on Computer and Computational Intelligence (ICCCI 2010).

**Bước 3:** Khả năng nhớ của mạng BAM. Tính 224 cặp mẫu như bước 3, mục 2.2.4; kết quả được đưa vào Bảng 5.

**Bước 4:** Đánh giá. Từ Bảng 5, khi mẫu vào:

- Trùng với mẫu đã học, tỷ lệ nhận mẫu đúng 100%, cho độ chính xác tuyệt đối;
- Khác với các mẫu gốc 1 đặc trưng (1 bit), tỷ lệ nhận mẫu đúng đạt (20%-100%);
- Khác với các mẫu gốc 2 đặc trưng (2 bits), tỷ lệ nhận mẫu đúng đạt (0%-40%);
- Khác với các mẫu gốc 3 đặc trưng (3 bits) trở lên, tỷ lệ nhận mẫu đúng đạt 0%;

Điều đặc biệt với tính hai chiều của BAM là khi cho tên, hoàn toàn xác định chính xác các đặc trưng của người đó. Ví dụ người có tên " $L$ "= $[1 \ 1 \ 1 \ 0]$ ; đặc trưng người đó  $C=L.W=[1 \ 1 \ 1 \ 0].W=[9 \ -3 \ -11 \ 5 \ -11] \Rightarrow [1 \ 0 \ 0 \ 1] = C$ .

**4. Kết luận và hướng phát triển tiếp theo**

Đóng góp của bài báo là bằng thực nghiệm, chứng minh và xác định khả năng nhớ mẫu của mạng Hopfield và mạng BAM; tính toán khả năng chịu lỗi của các mạng đó khi các mẫu vào có sai số 1, 2, 3, 4, 5 đặc trưng. Kết quả chỉ ra rằng các mạng có khả năng nhớ mẫu đã được học là 100%; các mẫu đầu vào có sai số từ 1+5 đặc trưng, khả năng nhận được giảm dần (hợp lý). Đóng góp khác là đề xuất ứng dụng cho lớp bài toán dùng mạng BAM nhận dạng và gán nhãn. Bài báo đã chứng tỏ: mạng BAM có thể xử lý hai chiều: chiều thuận: cho đặc trưng đầu vào, xác định được nhãn và ngược lại: cho nhãn (tên đối tượng) hoàn toàn có thể xác định được các đặc trưng của đối tượng với độ chính xác 100% nếu các đặc trưng (hoặc các nhãn) đã được học..

Hướng phát triển tiếp theo của bài báo là nghiên cứu khả năng nhớ mẫu của mạng BAM 3 lớp, của mạng nơron tế bào, cũng như lớp các mạng nơron truy hồi bậc cao. Mặt khác, chúng tôi đang nghiên cứu mức sai lệch mẫu vào có thể xác định bằng các chuẩn khoảng cách để có thể rút ra quy luật chịu lỗi của lớp mạng nơron hồi quy; kết quả sẽ báo cáo ở các công trình tiếp theo.



- [5]. Lejie Yu (2010), *The New Research of Associative Memory Neural Networks*, 2010 International Conference on Computer and Computational Intelligence (ICCCI 2010).
- [6]. Bart Kosko (1988). *Bidirectional Associative Memories*. IEEE Trans. On System, Man, and Cybernetics, vol. 18, No.1, January/February.
- [7]. Nathan John Burles (2014). *Pattern Recognition using Associative Memories*. PhD. Dissertation, University of York.
- [8]. Hui Wang, Yue Wu, Biaobiao Zhang and K. -L. Du (2011). *Recurrent Neural Networks: Associative Memory and Optimization*, J. Inform Tech Soft Eng. 1:104: 10.4172/2165-7866.1000104.
- [9]. Nguyen Quang Hoan. *The Stability of the High-Order Hopfield Neural Networks and Their Application Abilities for Robot*. PhD. Dissertation, 1996.

## PATTERN RECOGNITION ABILITIES OF RECURRENT NEURAL NETWORKS

### Abstract:

*The artificial neural networks are simulating the human brain. Could artificial neural networks memorize as the human brain? The paper presents the structures, the learning rules and the stability of the Hopfield, Bidirectional Associative Memory (BAM), two main recurrent neural networks. We also perform the experiments on their memory abilities, ability of fault isolation for several of failure bits. An example of pattern recognition of image faces and corresponding their labels are also represented.*

**Keywords:** *Recurrent neural networks, Hopfield, BAM neural networks, Liapunov stability, pattern recognition, learning rule.*