# NHẬN DẠNG ẢNH MẶT NGƯỜI SỬ DỤNG MẠNG NƠRON LIÊN KẾT HAI CHIỀU

## PATTERN RECOGNITION OF FACE IMAGES USING BIDIRECTIONAL ASSOCITIVE MEMORY

**Bùi Tiến Chiến[1*], Nguyễn Kim Quế[1], Dương Đức Anh[2], Vũ Thị Thêm[2], Nguyễn Quang Hoan[3]**

[1]Trường Đai học Điện lực, [2]Viện Nghiên cứu Điện tử, Tin học, Tự động hóa
[3]Học viện Công nghệ Bưu chính Viễn thông

Ngày nhận bài: 04/10/2023, Ngày chấp nhận đăng: 17/10/2023, Phản biện: PGS. TS Trần Đức Thuận

**Tóm tắt:**

Chủ đề của bài báo là khai thác, sử dụng bộ nhớ liên kết hai hướng (BAM: Bidirectional Associative Memory) một loại mạng nơron truy hồi để nhận dạng ảnh mặt người. Bài báo xây dựng cấu trúc BAM và thuật toán nhận dạng ảnh mặt người được xây dựng dựa trên luật học Hebb và mạng BAM. Hệ thống nhận dạng có kích thước nhỏ, gọn nhẹ, tín hiệu ra được thử nghiệm ghép với thiết bị điện tử và loa thông báo để cảnh báo nhận dạng đúng – sai và có thể dùng cho điều khiển ON/OFF thích hợp với những hệ thống vừa và nhỏ như ngôi nhà thông minh hoặc tương đương.

**Từ khóa:**

Mạng nơron nhân tạo, nhận mẫu, bộ nhớ liên kết, luật học.

**Abstract:**

The focus of the article is the exploration and application of Bidirectional Associative Memory (BAM), a form of recurrent neural network, for the purpose of recognizing human facial images. The article presents a novel algorithmic framework rooted in Hebb's rule, tailored specifically for the recognition of human faces. This pattern recognition system is characterized by its compact size and efficiency. The output signal undergoes testing in conjunction with electronic devices and notification speakers, serving as a reliable indicator for correct and incorrect pattern recognitions. It is adaptable for ON/OFF control, making it well-suited for various applications, particularly in the realm of small and medium-sized systems such as smart homes and their equivalents.

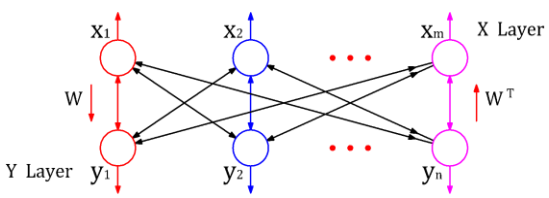
**Keywords:**

Artificial Neural Networks, Pattern Recognition, Associative Memory, Learning Rule.

## 1. MỞ ĐẦU

Ngày nay, mạng nơron học sâu với kích thước lớn (hàng nghìn lớp, mỗi lớp có hàng trăm nơron) nhằm giải quyết những bài toán nhận mẫu ảnh, tiếng nói cũng như dự báo có dữ liệu lớn. Tuy nhiên, nhiều hệ thống vừa và nhỏ, có dữ liệu không lớn thích hợp cho ứng dụng những mạng nơron có kích thước nhỏ phù hợp với công nghiệp và dân dụng của nước ta với chi phí thấp. Mạng nơron liên kết hai chiều, một trong các dạng mạng nơron hồi

quy thích hợp cho các ứng dụng loại này [1]. Nhóm mạng hồi quy có ba loại phổ biến: mạng Hopfield, mạng BAM [6] và mạng nơron tế bào (Cellular Neural Networks: CeNN) [1], [2], [5]. Về mặt lý thuyết, họ mạng nơron hồi quy có cấu trúc phản hồi, có thể gây mất ổn định nên tiêu chuẩn ổn định là điều kiện cần đầu tiên. Như vậy, khác với mạng nơron truyền thẳng, các trọng số của mạng nơron truy hồi vừa phải đảm bảo ổn định, vừa phải đảm bảo đầu ra hội tụ về các giá trị mong muốn được xác định. Về mặt ứng dụng, mạng nơron tế bào [3], một trong những mạng thuộc nhóm mạng Hopfield đã thành công trong xây dựng máy tính nơron dạng mảng dựa theo mạng nơron tế bào [4]. Các mạng hồi quy, nhất là mạng BAM được sử dụng điển hình làm bộ nhớ liên kết [2]. Khác với các ứng dụng điển hình đã nêu, trong bài báo này, nhóm tác giả xây dựng kiến trúc mạng BAM với 11 nơron để nhận dạng mặt người khả dĩ cho phép mở cửa cho khách vào ngôi nhà thông minh khi xác định đó là người quen biết.




**Hình 1. Mô hình cấu trúc mạng BAM**

## 2. CẤU TRÚC VÀ LUẬT HỌC MẠNG BAM

### 2.1. Mô hình cấu trúc mạng BAM

Tư tưởng tạo mạng hồi quy đầu tiên được Kohonen, Anderson và Nakano đề ra năm 1972 [7]. Năm 1982, Hopfield đã hiệu chỉnh mạng đó thành mạng Hopfield rời rạc [8] và hai năm sau (1984) cũng chính Hopfield xây dựng thành mạng liên tục, chứng minh và đưa ra các điều kiện ổn định cho hai loại mạng này [9]. Năm 1988, Kosko đề xuất mạng BAM [10] dựa trên hai lớp mạng Hopfield: lớp thứ nhất, lớp *X*: lớp truyền thẳng; lớp thứ hai, lớp *Y*: lớp phản hồi (Hình 1).

a) Phương trình trạng thái của lớp mạng thứ nhất (với $n=6$ nơron ở Hình 2):

$$x_j^* = \sum_{i=1}^{n} w_{ij} y_i \,; i,j \in R, j = 1..m \quad (1)$$

trong đó:

$y_i = f(y_j^*)$ là đầu vào thứ i của lớp Y (xem (3)); $i = 1..n$;

$x_j^*$ là trạng thái thứ $j$ lớp $X$, $j = 1..m$;

$w_{ij}$ là trọng số liên kết đầu vào thứ i đến trạng thái thứ j của lớp mạng thứ nhất X.

b) Phương trình trạng thái của lớp mạng thứ hai (chi tiết với 5 nơron ở Hình 2):

$$y_j^* = \sum_{i=1}^{n} w_{ij} x_i \,; i,j \in R, \; i = 1..n; \quad (2)$$

trong đó:

$x_i = f(x_j^*)$ ; $y_i$ là đầu vào thứ $i$ của lớp $X$;

$y_j^*$ là trạng thái thứ $j$, $j = 1..m$;

$w_{ij}$ là trọng số liên kết đầu vào thứ $i$ đến trạng thái thứ $j$ của lớp mạng thứ hai.

c) Phương trình đầu ra $y_j(k+1) = f(y_j^*(k+1))$ cập nhật ở bước thứ $k+1$, (Hình 2):

$$y_j(k+1) = \begin{cases} 1 & \text{Nếu } y_j^*(k+1) > 0 \\ y_j(k) & \text{Nếu } y_j^*(k+1) = 0 \\ 0 & \text{Nếu } y_j^*(k+1) < 0 \end{cases} \quad (3)$$

d) Phương trình đầu vào: $x_i(k+1)=f(x_j^*(k+1))$ cập nhật ở bước thứ $k+1$ , (Hình 2):

$$x_i(k+1)=\begin{cases}1 & \text{Nếu } x_i^*(k+1)>0\\ x_i(k) & \text{Nếu } x_i^*(k+1)=0\\ 0 & \text{Nếu } x_i^*(k+1)<0\end{cases} \quad (4)$$

### 2.2. Luật học cho mạng BAM

Học trong mạng nơron gồm học cấu trúc và học tham số. Giả sử cấu trúc đã được chọn như ở mục trước. Học tham số là phương pháp xác định các trọng số của mạng dựa trên bộ các cặp dữ liệu mẫu vào/ra, gọi là dữ liệu học. Luật Hebb giải thích việc chỉnh trọng của mạng nơron mà không cần tín hiệu chỉ đạo từ ngoài như $d_i$ ở luật học Perceptron Delta [8]. Hopfield cũng cải tiến luật Hebb cho các mạng tự liên kết dùng để nhớ mẫu. Luật Hebb có thể cải tiến thành nhiều dạng khác nhau như luật học Hopfield theo (5)

$$w_{ij}=\sum_{p=1}^{P} y_i^p\, x_j^p; i=1..n, j=1..m \quad (5)$$

ở đây, *p* là mẫu thứ *p; p=1..P.* Có thể viết (5) ở dạng vectơ - ma trận:

$$\boldsymbol{W}=\sum_{p=1}^{P}(\boldsymbol{Y^p})^T\boldsymbol{X^p} \quad (6)$$

ở đây, *T* là chuyển vị của vectơ.

***Ví dụ 1.*** Ví dụ 1 minh họa việc tính ma trận trọng số $\boldsymbol{W}$ cho lớp l và $\boldsymbol{W^T}$ cho lớp 2 của BAM với 4 mẫu ở Bảng 1. Ký hiệu:

$$\boldsymbol{A}=\boldsymbol{X_1}=[x_i^p]=[x_1^2\ \ x_2^2\ \ x_3^2\ \ x_4^2\ \ x_5^2\ \ x_6^2]=$$
$$=[0\ \ 0\ \ 1\ \ 1\ \ 0\ \ 1]$$

$\boldsymbol{B}=\boldsymbol{X_2}=[x_i^7]$; $\boldsymbol{C}=\boldsymbol{X_3}=[x_i^{16}]$; $\boldsymbol{D}=\boldsymbol{X_4}=[x_i^{23}]$

Tương tự với các vectơ nhãn, giá trị (Bảng 1) và các biến vào/ra tương tự ở Hình 2.

$$\boldsymbol{K}=\boldsymbol{Y_1}=[y_j^p]=[y_1^2\ \ y_2^2\ \ y_3^2\ \ y_4^2\ \ y_5^2]=$$
$$=[1\ \ 0\ \ 0\ \ 0\ \ 1]$$

$\boldsymbol{L}=\boldsymbol{Y_2}=[y_j^7]$; $\boldsymbol{M}=\boldsymbol{Y_3}=[x_i^{16}]$; $\boldsymbol{N}=\boldsymbol{Y_4}=[y_j^{23}]$

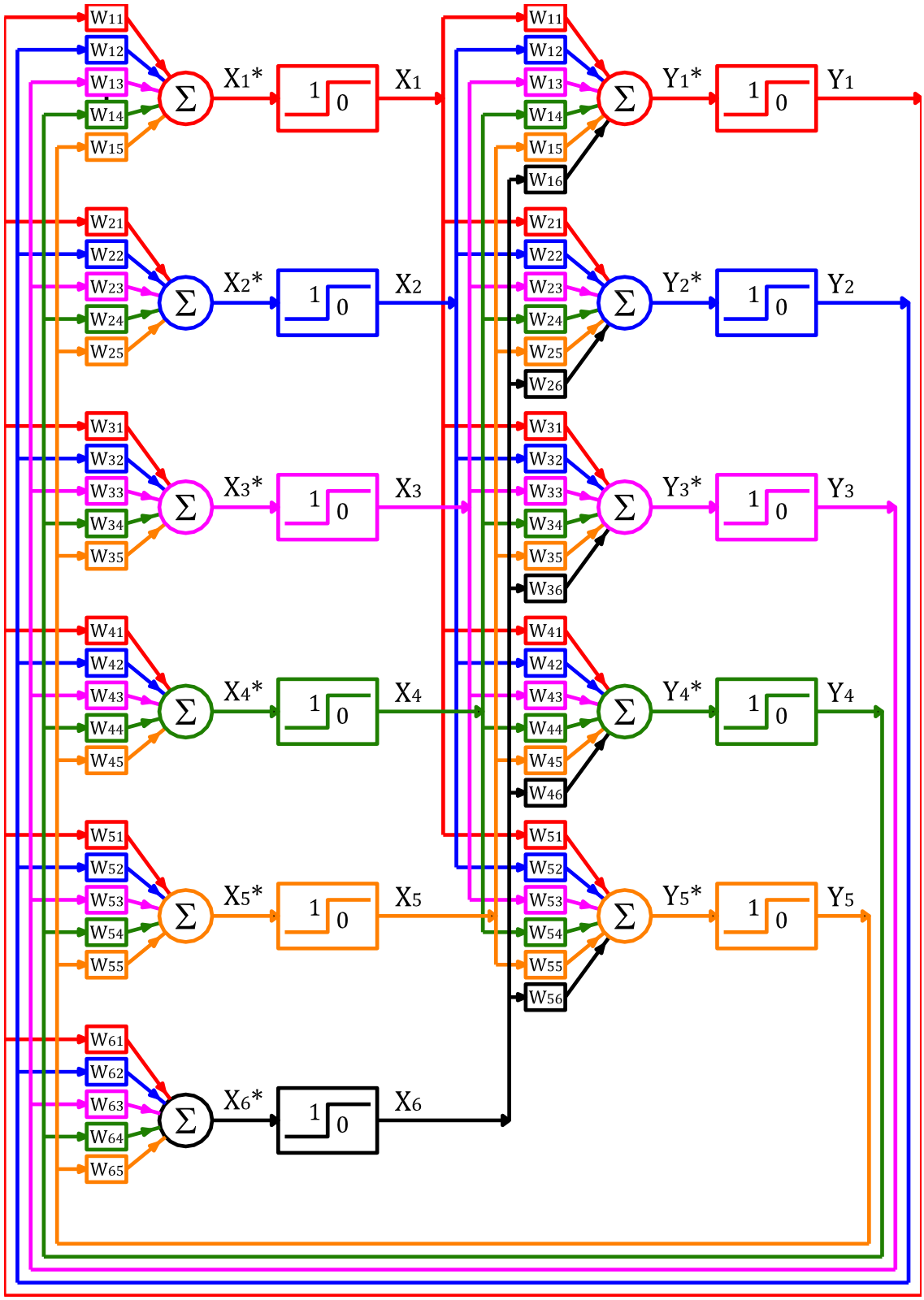
Khi đó, từ (6), đổi các vectơ có giá trị "0" thành "-1" để không mất giá trị của trọng số (không thay đổi ý nghĩa của phép tính trọng).

$$\boldsymbol{W}=\sum_{p=1}^{P}(\boldsymbol{Y^p})^T\boldsymbol{X^p}=$$
$$=K'^TA'+L'^TB'+M'^TC'+N'^TD' \quad (7)$$

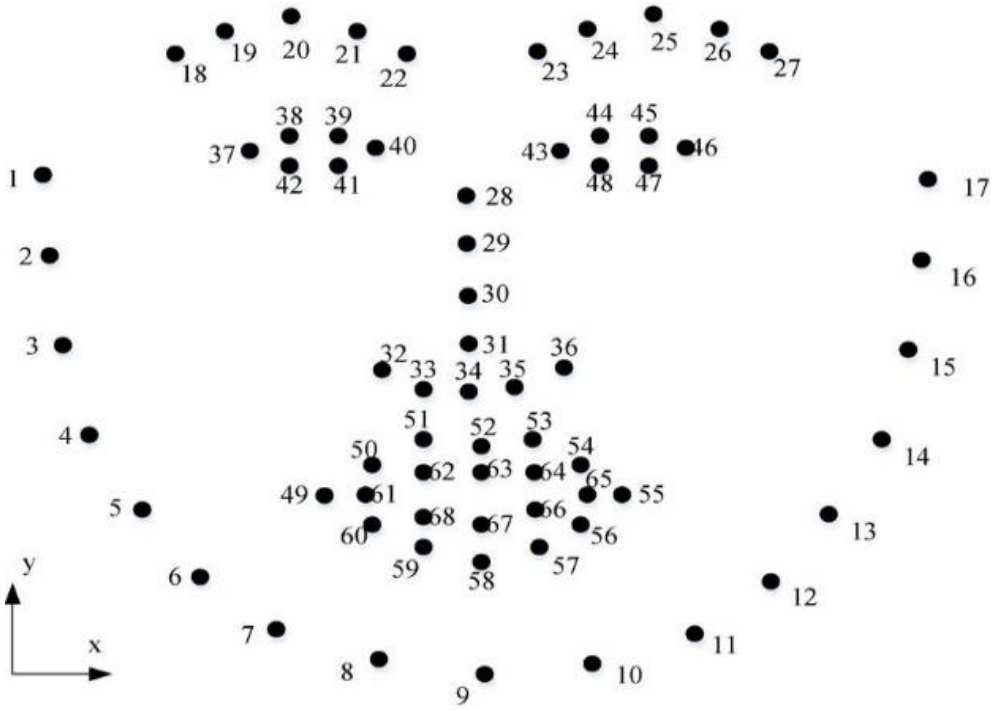
$$W=\begin{bmatrix}-2 & -4 & 0 & 2 & 0 & 4\\ 2 & 4 & 0 & -2 & 0 & -4\\ 0 & 2 & 2 & 0 & 2 & -2\\ 0 & -2 & -2 & 0 & 2 & 2\\ -4 & -2 & 2 & 4 & -2 & 2\end{bmatrix} \quad (8)$$

và $$W^T=\begin{bmatrix}-2 & 2 & 0 & 0 & -4\\ -4 & 4 & 2 & -2 & -2\\ 0 & 0 & 2 & -2 & 2\\ 2 & -2 & 0 & 0 & 4\\ 0 & 0 & 2 & 2 & -2\\ 4 & -4 & -2 & 2 & 2\end{bmatrix} \quad (9)$$

Từ ví dụ, có thể quy nạp để tính *n*m=6*5=30* giá trị cho hai bộ trọng số $W^T$ không phải cho 4 mẫu của lớp mạng thứ nhất (9) và *W* cho lớp mạng thứ hai (8) như Bảng 1 mà cho 32 mẫu như ở Phụ lục (xem thêm Hình 2) (không khó để thực hiện trên bất kỳ phần mềm tùy ý). Với 32 mẫu, công thức (7) khi đó gồm tổng của 32 số hạng. Như vậy, pha học là bước tính các giá trị để gán cho từng trọng số ở cấu trúc Hình 1.




**Hình 2. Thiết kế cấu trúc mạng BAM theo bài toán ở mục 3.1 với số nơron: *n*m=6*5***




**Hình 3. Các điểm trên khuôn mặt người**

**Bảng 1. Bốn mẫu (Samples) đại diện: 2, 7, 16, 23 làm ví dụ cho pha học**

| TT | ĐẦU VÀO (Đặc trưng) | | | ĐẦU RA (Nhãn) | | |
|---|---|---|---|---|---|---|
| | Tên đặc trưng | Mã hóa đặc trưng $[x_1x_2x_3x_4x_5x_6]$ | $X^* = (K, L, M, N).W$ | Tên nhãn | Mã hóa nhãn $[y_1y_2y_3y_4y_5]$ | $Y^* = (A, B, C, D).W^T$ |
| 2 | $A$<br>$A'$ | [ 0 0 1 1 0 1]<br>[-1 -1 1 1 -1 1] | [-6 -6 2 6 -2 6] | $K$<br>$K'$ | [1 0 0 0 1]<br>[1 -1 -1 -1 1] | [ 6 -6 0 0 8] |
| 7 | $B$<br>$B'$ | [0 1 1 1 0 0]<br>[-1 1 1 1 -1 -1] | [-2 4 4 2 0 -4] | $L$<br>$L'$ | [0.. 1 1 .0 1]<br>[-1 1 1 -1 1] | [-2 2 4 -4 4] |
| 16 | $C$<br>$C'$ | [0 ..0 ..0 1 .0 1]<br>[-1 -1 -1 1 -1 1] | [-6 -8 0 6 0 8] | $M$<br>$M'$ | [1 0 0 1 1]<br>[1 -1 -1 1 1] | [ 6 -6 -2 2 6] |
| 23 | $D$<br>$D'$ | [0.. 0 1 1 1 1]<br>[-1 -1 1 1 1 1] | [-6 -6 2 6 2 6] | $N$<br>$N'$ | [1 0 0 1 1]<br>[1 -1 -1 1 1] | [ 6 -6 2 2 6] |

## 3. NHẬN DẠNG ẢNH MẶT NGƯỜI VỚI BAM

### 3.1. Phát biểu bài toán

Cho tập 40 mẫu vào/ra thử nghiệm được tách thành hai phần: phần dữ liệu học 32 mẫu (chiếm 80%); phần dữ liệu thử 6 (chiếm 20%) trong đó, mỗi phần chứa hai tập: tập mẫu vào (là đặc trưng ảnh mặt người); tập mẫu ra (là tên người được mã

hóa thành nhãn) với các giá trị nhị phân. (Bảng 1 là bốn mẫu minh họa về dữ liệu học). Nhiệm vụ: i) thiết kế mạng BAM kích thước *n*m=6*5,* ii) thu thập và xử lý dữ liệu; xây dựng thuật toán học để tìm bộ trọng số và iii) nhận dạng ảnh mặt người.

### 3.2. Thiết kế cấu trúc mạng BAM

Trên cơ sở dữ liệu Bảng 1 và bài toán, chúng tôi đã thiết kế mạng nơron BAM (Hình 2) kích thước *n*m=6*5* có cấu trúc hai lớp theo (1), (2); các hàm tương tác dạng bước nhảy theo (3), (4), thích ứng hoàn toàn với bảng dữ liệu học.

### 3.3. Xử lý dữ liệu và gán nhãn cho BAM

Bộ dữ liệu 40 mẫu do chúng tôi thu thập, có ảnh tùy ý: gồm ảnh có một hoặc nhiều người; kích thước ảnh có thể khác nhau. Với dữ liệu tùy tiện (gần với thực tế); việc đầu tiên, từ kho dữ liệu ảnh, chúng ta tiến hành tiền xử lý gồm: khâu tách ảnh có nhiều người để chọn một mặt người, sau đó đưa vào khung ảnh chuẩn… sau đó trích chọn đặc trưng khuôn mặt. Ở đây, chúng tôi chọn vectơ đầu vào đặc trưng $[x_1x_2x_3x_4x_5x_6]$ mã hóa 6 BIT, vectơ ra $[y_1y_2y_3y_4y_5]$ được mã hóa 5 BIT tạo thành một cặp mẫu vào/ra theo đầu bài.

Để xác định các đặc trưng đầu vào, ta tính các đặc điểm trên khuôn mặt người được xác định nhờ phần mềm Python dựa trên sơ đồ Hình 4 gồm 68 đặc điểm trên khuôn mặt người, lấy ra sáu đặc trưng gồm: kích thước mũi ($x_1$); mũi ngắn $x_1$=0, mũi dài $x_1$=1 được tính dựa vào các điểm từ 28 đến 36. Kích thước miệng ($x_2$): Được tính từ điểm 49 đến 68. Kích thước khuôn mặt ($x_3$): từ điểm 1 đến 17. Khoảng cách hai mắt ($x_4$): từ điểm 37 đến 48.

Khoảng cách lông mày tới mũi ($x_5$): từ điểm 18 đến điểm 36. Khoảng cách mắt tới miệng ($x_6$): từ điểm 37 đến 68. Sử dụng phần mềm Python, ta có được các chỉ số đo các đặc trưng đầu vào. So sánh và quy đổi qua cột tiêu chuẩn (Bảng 3), nếu lớn hơn hoặc bằng giá trị tiêu chuẩn thì quy ước là 1, ngược lại quy ước là 0.

**Bảng 2. Mã hóa đặc trưng khuôn mặt**

| Đặc trưng | Kích thước miệng | | Khoảng cách hai mắt | | Khoảng cách lông mày - mũi | | Kích thước mũi | | Kích thước mặt | | Khoảng cách mắt - miệng | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Giá trị | To | Nhỏ | Xa | Gần | Xa | Gần | To | Nhỏ | To | Nhỏ | Xa | Gần |
| Mã hóa | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |

**Bảng 3. Tính toán đặc trưng đầu vào**

| Tên | A | B | C | D | >=1 <0? | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 1581 | 1533 | 1432 | 1575 | 1600 | 0 | 0 | 0 | 0 |
| $x_2$ | 645 | 691 | 444 | 671 | 680 | 0 | 1 | 0 | 0 |

| Tên | A | B | C | D | >=1 <0? | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|
| $x_3$ | 2548 | 2489 | 2087 | 2535 | 2480 | 1 | 1 | 0 | 1 |
| $x_4$ | 650 | 495 | 495 | 566 | 490 | 1 | 1 | 1 | 1 |
| $x_5$ | 1304 | 1140 | 1028 | 1335 | 1330 | 0 | 0 | 0 | 1 |
| $x_6$ | 743 | 564 | 600 | 688 | 600 | 1 | 0 | 1 | 1 |

Hoàn toàn có thể tăng, giảm số đặc trưng trên.

Với nhãn đầu ra, chúng tôi mã hóa cho từng đối tượng, mỗi đối tượng là một vectơ $[\boldsymbol{y_1 y_2 y_3 y_4 y_5}]$. Số lượng vectơ tối đa có thể $2^m=2^5=32$ (bằng số lượng đầu ra đã chọn).

### 3.4. Thuật toán nhận dạng ảnh dùng BAM

Nhận dạng ảnh gồm hai giai đoạn: tiền xử lý ảnh và nhận mẫu.

*Input: - Ảnh người (nhận được qua Camera)*

*- Mạng BAM (Hình 2 đã thiết kế)*

*Output: Nhận dạng ảnh và thông báo.*

***Thuật toán***

*B1.* Tách ảnh từ nhiều người. Chuẩn khung ảnh.

*B2.* Trích chọn đặc trưng; số hóa theo Bảng 2, 3.

*B3.* Tính các ma trận trọng số $\boldsymbol{W}$, $\boldsymbol{W^T}$ theo (8); (9)

*B4.* $\boldsymbol{Y^{*P}= A\ W^T}$ theo (1);

*B5.* $\boldsymbol{Y^P= f(Y^{*P})=K?}$ theo (4)

*B6. Nếu* $\boldsymbol{Y^P=K}$ *(***A** *và* **K** *cùng cặp mẫu): Nhận đúng*

*Nếu* $\boldsymbol{Y^P\neq K}$ *(***A** *và* **K** *khác cặp mẫu**):** Nhận sai*

*B7.* Thông báo

*B8.* Chuyển tín hiệu ra thiết bị

Ở thuật toán trên, tại các bước B1 và B2, chúng tôi sử dụng phần mềm xử lý ảnh của thư viện Python. Các bước B3-B8, chúng tôi lập trình và xây dựng. Để minh họa các bước B3-B8 (phần trọng tâm của bài báo), chúng tôi tính cho 4 mẫu theo Bảng 1 (trích trong 32 mẫu).

***Ví dụ 2***. Ví dụ 2 thể hiện khả năng nhận dạng của BAM cả hai chiều.

a) Cho mẫu vào, nhận nhãn (mẫu ra)

Ý nghĩa của việc này là nếu có một người qua camera để nhận ảnh, qua xử lý ảnh, nhận mẫu, mạng nơron cho mã người đó (tức là nhãn). Ký hiệu $\{\boldsymbol{A_0, B_0, C_0, D_0}\}$ lần lượt là các mẫu ảnh vào dùng để thử nghiệm có giá trị đúng với các mẫu đã được học $\{\boldsymbol{A, B, C, D}\}$ đã được nhớ trong $\boldsymbol{W^T, W}$. Khi đó, BAM nhận dạng và sẽ cho các kết quả tương ứng sau:

**- Khi mẫu vào:** $\boldsymbol{A_0} = [\mathbf{0\ 0\ 1\ 1\ 0\ 1}]$

Đầu ra được tính theo công thức: $\boldsymbol{Y^*= A_0 W^T} = [6\ \ -6\ \ 0\ \ 0\ \ 8]$. Áp dụng hàm tương tác đầu ra, hàm chặn theo (3) và Hình 2 ta có:

$\Rightarrow \boldsymbol{Y} = [1\ 0\ 0\ 0\ 1] = \boldsymbol{K}$, đúng mẫu đã học.

b) Cho mẫu ra (nhãn), nhận đặc trưng vào

Ở hướng ngược lại, ký hiệu $\{\boldsymbol{K_0, L_0, M_0, N_0}\}$ là các mẫu trùng với các

nhãn $\{\boldsymbol{K}, \boldsymbol{L}, \boldsymbol{M}, \boldsymbol{N}\}$ thì:

Khi mẫu vào: $\boldsymbol{K_0} = [\mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{1}]$, đầu ra được tính theo công thức: $\boldsymbol{X}^* = \boldsymbol{L_0 W} = [-6 \; -6 \; 2 \; 6 \; -2 \; 6]$. Áp dụng hàm tương tác đầu ra (4) và Hình 2: ta được:

$\Rightarrow \boldsymbol{X} = [0\ 0\ 1\ 1\ 0\ 1] = \boldsymbol{A}$: đúng mẫu đã học.

Tương tự, có thể thử với các mẫu còn lại với kết quả đúng 100%. Như vậy, đặc trưng của một ảnh người đưa vào BAM, đầu ra đúng tên tương ứng với đặc điểm của người đó và ngược lại.

Bài toán nhận dạng hoàn toàn có thể sử dụng các phương pháp truyền thống. Ưu điểm của phương pháp sử dụng BAM ở chỗ: mẫu học được ghi nhớ trong bộ ma trận $\boldsymbol{W}, \boldsymbol{W^T}$, thay vì phải dùng bộ nhớ theo các phương pháp khác. Ngoài ra, các bộ nhớ $\boldsymbol{W}, \boldsymbol{W^T}$ còn có khả năng chịu lỗi sẽ trình bày như mục dưới đây.

### 3.5. Khả năng chịu lỗi của mạng BAM

#### *3.5.1. Mẫu vào/ra đúng như đã học*

Để đánh giá độ chính xác nhận dạng của mạng BAM, chúng ta sử dụng ma trận nhầm lẫn (Confusion Matrix). Để tính độ chính xác, chỉ cần tính tổng phần tử trên đường chéo của ma trận Confusion chia cho tổng phần tử. Tổng số phần tử không nằm trên đường chéo là lỗi dự đoán của giải thuật. Trên cơ sở ma trận nhầm lẫn ta xây dựng Bảng 4 theo ví dụ 2.

#### 3.5.2. Khả năng chịu lỗi của mạng BAM

Đối với các mẫu chưa được học, có nghĩa là mã của các vectơ đặc trưng vào và mã của vectơ nhãn (đầu ra) có thể khác nhau từ 1 đến 5 BIT cho nhãn, 6 BIT cho đặc trưng vào. Sự sai khác một số BIT có thể xảy ra ở các vị trí cột khác nhau của các vectơ. Chúng tôi đã tiến hành thử các trường hợp sai số khác nhau và rút ra kết luận sau:

- Một số trường hợp sai, BAM vẫn nhận dạng đúng mẫu. Lý do có thể giải thích: BAM có khả năng chịu lỗi do hàm tương tác đầu ra là hàm bị chặn trên và dưới: hàm (3) và hàm (4), trong phạm vi nhất định BAM có khả năng chịu lỗi. Đặc điểm này, BAM giống não người: nghĩa là, con người biến dạng một chút vẫn có thể nhận dạng. Đây là khả năng "suy diễn" của BAM.
- Nếu mẫu vào/ra sai nhiều, hiển nhiên, BAM sẽ nhận biết mẫu sai đó gần với một đối tượng khác và xảy ra nhầm lẫn.

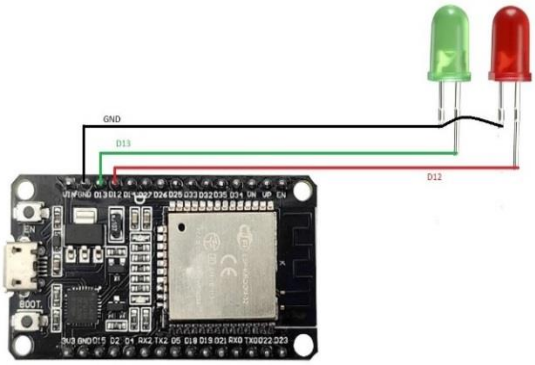
Để khắc phục khả năng nhầm lẫn, cần có một khoảng cách (hay sự khác biệt) nhất định giữa các mẫu. Để thực hiện điều đó, chúng ta cần nhiều đặc trưng (ứng với nhiều phần tử nơron) hoặc giảm số mẫu. Điều này đòi hỏi phải tăng số nơron (thay đổi cấu trúc) và tăng số lượng tính toán, ngoài phạm vi bài báo.

## 4. ỨNG DỤNG BAM CHO NGÔI NHÀ THÔNG MINH

Từ các kết quả trên, chúng tôi có ý tưởng dùng khả năng nhận biết đúng/sai của BAM làm nhiệm vụ mở của tự động khi khách quen (ảnh đã được học) đến ngôi nhà thông minh. Trong khuôn khổ bài báo, chúng tôi thiết kế phần cứng để

chuyển tín hiệu nhận dạng sang thiết bị báo hiệu đèn xanh (đúng người quen), đèn đỏ (người lạ) mang tính tượng trưng.

Để hiển thị kết quả nhận dạng cho tín hiệu đèn thông báo, chúng tôi đã thiết kế thiết bị ngoại vi kết nối với phần mềm Python như sau:




**Hình 4. Thiết bị tạo tín hiệu đèn thông báo**

Lắp sơ đồ đấu nối, thông số kỹ thuật của cổng ngoại vi ESP32 ESP-WROOM-32DEVBOARD (Hình 4). Một vài thông số thiết bị cơ bản như:

- Điện áp logic là 2,7 đến 3,3 V.
- 1 Enable Button (Chân reset) và 1 User Button (GPIO 0). Clock: 240 MHz.
- Led báo nguồn và User Led (GPIO 2).
- 3 UART: Serial Debug mặc định: UART 0.

Khi đã có tín hiệu, hoàn toàn có khả năng thiết kế với các mạch logic, điều khiển động cơ đóng mở chốt cửa ra/vào được trang bị cho các ổ khóa trong các ngôi nhà thông minh.

## 5. THẢO LUẬN VÀ KẾT LUẬN

Bài báo có các đóng góp: i) xây dựng cấu trúc BAM; ii) Mô tả luật học bằng ví dụ minh họa và tiến tới xây dựng thuật toán nhận dạng, trong đó phần xử ảnh sử dụng thư viện Python, phần học và nhận dạng chúng tôi lập trình tính đưa ra kết quả; iii) Kết quả xuất tín hiệu-ra thiết bị điện tử (tự thiết kế) để mô phỏng.

Với các kết quả ban đầu, chúng tôi hy vọng đây sẽ là sở cứ cho các cấu trúc có số nơron tùy ý và mở ra triển vọng có thể sử dụng mạng nơron BAM điều khiển ON/OFF cho các hệ thống vừa và nhỏ như ngôi nhà thông minh. Điều này chứng tỏ, các công cụ nơron tiên tiến trong lĩnh vực trí tuệ nhân tạo có thể áp dụng không những cho các hệ thống lớn phức tạp mà có thể cho các đối tượng nhỏ, đơn giản trong việc phối ghép phần cứng và phần mềm.

## PHỤ LỤC: BẢNG DỮ LIỆU HỌC

| TT | Đầu vào | | Đầu ra | |
|---|---|---|---|---|
| | Đặc trưng | Mã hóa đặc trưng | Nhãn | Mã hóa nhãn |
| | Pi | [x1x2x3x4x5x6] | Si | [y1y2y3y4y5] |
| 1 | P1 | [1 0 1 0 0 1] | S1 | [1 1 0 1 0] |
| 2 | P2 | [0 0 1 1 0 1] | S2 | [1 0 0 0 1] |

| TT | Đầu vào | | Đầu ra | |
|---|---|---|---|---|
| | Đặc trưng | Mã hóa đặc trưng | Nhãn | Mã hóa nhãn |
| | Pi | [x1x2x3x4x5x6] | Si | [y1y2y3y4y5] |
| 3 | P3 | [1 1 0 1 0 1] | S3 | [0 0 1 1 1] |
| 4 | P4 | [0 0 0 1 1 1] | S4 | [0 0 0 1 0] |
| 5 | P5 | [1 1 1 1 1 1] | S5 | [1 1 0 0 1] |
| 6 | P6 | [1 1 0 0 0 1] | S6 | [1 0 0 1 0] |
| 7 | P7 | [0 1 1 1 0 0] | S7 | [0 1 1 0 1] |
| 8 | P8 | [1 0 1 0 1 0] | S8 | [0 1 0 1 1] |
| 9 | P9 | [0 0 0 1 0 0] | S9 | [1 0 0 0 0] |
| 10 | P10 | [1 1 1 1 0 1] | S10 | [1 0 1 0 1] |
| 11 | P11 | [0 1 0 0 1 0] | S11 | [1 1 0 0 0] |
| 12 | P12 | [0 1 0 1 1 1] | S12 | [1 0 1 0 0] |
| 13 | P13 | [1 1 1 0 1 1] | S13 | [1 0 1 1 0] |
| 14 | P14 | [1 1 1 1 1 0] | S14 | [1 1 0 1 1] |
| 15 | P15 | [0 0 1 0 0 0] | S15 | [0 1 0 0 0] |
| 16 | P16 | [0 0 0 1 0 1] | S16 | [1 0 0 1 1] |
| 17 | P17 | [1 0 0 0 0 0] | S17 | [1 1 1 1 0] |
| 18 | P18 | [1 0 1 0 1 1] | S18 | [0 0 0 1 1] |
| 19 | P19 | [1 0 1 0 0 0] | S19 | [0 0 0 0 0] |
| 20 | P20 | [1 1 1 0 0 0] | S20 | [0 1 1 1 0] |
| 21 | P21 | [1 0 0 0 0 1] | S21 | [0 0 1 0 1] |
| 22 | P22 | [1 0 1 1 0 0] | S22 | [0 0 0 0 1] |
| 23 | P23 | [0 0 1 1 1 1] | S23 | [1 0 1 1 1] |
| 24 | P24 | [0 1 0 1 1 0] | S24 | [1 1 1 1 1] |
| 25 | P25 | [0 0 1 1 1 0] | S25 | [1 1 1 0 0] |

| TT | Đầu vào | | Đầu ra | |
|---|---|---|---|---|
| | Đặc trưng | Mã hóa đặc trưng | Nhãn | Mã hóa nhãn |
| | Pi | [x1x2x3x4x5x6] | Si | [y1y2y3y4y5] |
| 26 | P26 | [0 0 1 1 0 0] | S26 | [0 1 1 1 1] |
| 27 | P27 | [1 1 0 0 1 0] | S27 | [0 0 1 1 0] |
| 28 | P28 | [0 1 1 1 1 1] | S28 | [0 1 0 0 1] |
| 29 | P29 | [1 0 0 1 0 1] | S29 | [0 0 1 0 0] |
| 30 | P30 | [1 0 0 1 1 0] | S30 | [0 1 0 1 0] |
| 31 | P31 | [1 1 1 0 0 1] | S31 | [1 1 1 0 1] |
| 32 | P32 | [0 0 0 1 1 0] | S32 | [0 1 1 0 0] |

## TÀI LIỆU THAM KHẢO

[1] Jichen Shi, Zhigang Zeng, "Design of In-Situ Learning Bidirectional Associative Memory Neural Network Circuit With Memristor Synapse," *IEEE Transactions on Emerging Topics in Computational Intelligence,* vol. 5, no. 5, pp. 743 - 754, 2021.

[2] Nguyễn Quang Hoan, Vũ Thị Thềm, Bùi Đình Quân, "Khả năng nhớ mẫu của mạng nơron hồi quy," *Tạp chí Khoa học và Công nghệ Đại học Sư phạm Hưng Yên,* vol. 13, no. 3, pp. 44-49, 2017.

[3] Boumediène Allaoqua, Abdellah Laoufi, Brahim Gasbaoui, "Multi-Drive Paper System Control Based on Multi-Input Multi-Output PID Controller," *Leonardo Journal of Sciences,* vol. 2010, no. 16, pp. 59-70, 2010.

[4] Zhongyang Liu, Shaoheng Luo, Xiaowei Xu, Cheng Zhuo, "Cellular Neural Network (CENN) FPGA Implementation Using Multi-level Optimization," in *China Semiconductor Technology International*, Shanghai, China, 2018.

[5] T. Roska, L.O. Chua, "The CNN Universal Machine: an Analogic Array Computer," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 40, no. 3, pp. 163-173, 1993.

[6] B. Kosko, "Bidirectional Associative Memories: Unsupervised Hebbian Learning to Bidirectional Backpropagation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 51, no. 1, pp. 103 - 115, 2021.

[7] T. Kohonen, Self-Organizing Maps, Berlin, Germany: Springer , 1995.

[8] J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," in *National Academy of Sciences*, United States of America, 1982.

[9] J. Hopfield, "Neurons with Graded Response have Collective Computational Properties like Those of Two-state Neuron," in *The National Academy of Sciences*, USA, 1984.

[10] Kosko, "Feedback Stability and Unsupervised Learning," in *IEEE 1988 International Conference on Neural Networks*, San Diego, CA, USA, 1988.

**Giới thiệu tác giả:**

Tác giả Bùi Tiến Chiến tốt nghiệp đại học ngành tự động hóa xí nghiệp công nghiệp tại Đại học Bách khoa Hà Nội năm 2015. Hiện nay tác giả đang theo học lớp cao học chuyên ngành tự động hóa của Trường Đại học Điện lực Hà Nội.

Lĩnh vực nghiên cứu: học máy, mạng nơron, điều khiển và tự động hóa.

Tác giả Nguyễn Kim Quế tốt nghiệp đại học ngành tự động hóa xí nghiệp công nghiệp tại Trường Đại học Điện lực Hà Nội năm 2019. Hiện nay tác giả đang theo học lớp cao học chuyên ngành tự động hóa của Trường Đại học Điện lực Hà Nội.

Lĩnh vực nghiên cứu: học máy, mạng nơron, điều khiển và tự động hóa.

Tác giả Nguyễn Quang Hoan tốt nghiệp đại học ở Moskơva (Liên Xô) năm 1967, nhận học hàm Phó giáo sư năm 2002. Tác giả nguyên là Trưởng Khoa Công nghệ thông tin, Học viện Công nghệ Bưu chính Viễn thông.

Lĩnh vực nghiên cứu: học máy, điều khiển tối ưu, điều khiển thông minh.

Tác giả Dương Đức Anh tốt nghiệp đại học ngành tự động hóa xí nghiệp công nghiệp tại Đại học Bách khoa Hà Nội năm 2007, nhận bằng Thạc sĩ ngành tự động hóa và điều khiển năm 2009. Hiện nay tác giả là Giám đốc Trung tâm Công nghệ cao - Viện Nghiên cứu  Điện tử, Tin học, Tự động hóa - Bộ Công Thương.

Lĩnh vực nghiên cứu: học máy, mạng nơron, điều khiển và tự động hóa.

Tác giả Vũ Thị Thềm tốt nghiệp đại học năm 2012 và nhận bằng Thạc sĩ năm 2017 tại Trường Đại học Sư phạm kỹ thuật Hưng Yên. Hiện nay tác giả là nghiên cứu sinh về chủ đề mạng nơron tế bào.

Lĩnh vực nghiên cứu: học máy, mạng nơron.