

# Phát triển thuật toán Perceptron hồi quy cho mạng nơ ron tế bào bậc hai

## Development of Recurrent Perceptron Learning Algorithm for Second-Order Cellular Neural Networks

Duong Duc Anh<sup>1\*</sup>, Nguyen Quang Hoan<sup>2</sup>, Nguyen Tai Tuyen<sup>2</sup>, Lai Thi Van Quyen<sup>1</sup>, Hoang Tuan Dat<sup>1</sup>

<sup>1</sup>Viện Nghiên cứu Điện tử, Tin học, Tự động hóa

<sup>2</sup>Học viện Bưu chính Viễn thông

\*Corresponding author E-mail: [ddatdh1@gmail.com](mailto:ddatdh1@gmail.com)

### Abstract

This paper develops a new learning rule to be applied to the calculation of all weights of Second-Order Cellular Neural Networks (SOCNNs) based on the Recurrent Perceptron Learning Algorithms (RPLA). By integrating not only the First-order input and output signals but also the Second-order input and output signals into a general input signal, the research team transformed the networks of SOCNNs into an equivalent structure with the traditional Perceptron Networks. From there, the parameters of SOCNNs can be determined by the supervised learning method. The paper also simulates SOCNNs on MATLAB to check the correctness and efficiency of the proposed algorithm.

**Keywords:** MATLAB, Recurrent Perceptron Learning Algorithm, Second Order Cellular Neural Networks, Templates, Weights

### Ký hiệu

Ký hiệu	Mô tả
<b>A1, B1, A2, B2, I</b>	Ma trận trọng số đầu vào, phân hồi bậc nhất; đầu vào, đầu ra phân hồi bậc 2 tương ứng
$u_{ij}, u_{kl}, u_{mn}$	Tín hiệu đầu vào của tế bào thứ (k, l); (i, j); (m, n) tương ứng
$x_{ij}(t), x_{kl}(t), x_{mn}(t)$	Tín hiệu trạng thái của tế bào thứ (k, l); (i, j); (m, n) tương ứng
$y_{ij}(t), y_{kl}(t), y_{mn}(t)$	Tín hiệu đầu ra của tế bào thứ (k, l); (i, j); (m, n) tương ứng
$i, j, k, l, m, n$	Biến xác định vị trí của từng tế bào trong SOCNNs
$a_{11}, a_{12}, a_{13}, a_{14}, a_{15}$	Toán hạng ma trận phân hồi bậc nhất
$b_{11}, b_{12}, b_{13}, b_{14}, b_{15}$	Toán hạng ma trận đầu vào bậc nhất
$a_{211}, a_{212}, \dots, a_{252}, \dots, a_{294}, a_{295}$	Toán hạng ma trận phân hồi bậc hai
$b_{211}, b_{212}, \dots, b_{252}, \dots, b_{294}, b_{295}$	Toán hạng ma trận đầu vào bậc hai

### Từ viết tắt

CNNs	Cellular Neural Networks
HOCNNs	High – Order Cellular Neural Networks

NNs	Neural Networks
RPLA	Recurrent Perceptron Learning Algorithm
SOCNNs	Second Order Cellular Neural Networks
SORPLA	Second Order Recurrent Perceptron Learning Algorithm
CNNs	Cellular Neural Networks/ Cellular Non-linear Networks
RPLA	Recurrent Perceptron Learning Algorithm
SOCNNs	Second Order Cellular Neural Networks
SORPLA	Second Order Recurrent Perceptron Learning Algorithm

### Tóm tắt

Bài báo này phát triển một luật học mới được áp dụng cho việc tính toán toàn bộ các trọng số của mạng nơ ron tế bào bậc hai (SOCNNs) dựa trên phương pháp học Perceptron hồi quy (RPLA). Bằng việc tích hợp các tín hiệu đầu vào, đầu ra bậc nhất và bậc hai thành một tín hiệu đầu vào tổng quát, nhóm nghiên cứu đã biến đổi SOCNNs thành cấu trúc tương đương với mạng Perceptron truyền thống. Từ đó các tham số của SOCNNs hoàn toàn có thể được xác định theo phương pháp học có giám sát. Bài báo cũng tiến mô phỏng SOCNNs trên MATLAB để kiểm tra tính đúng đắn và hiệu quả của thuật toán đề xuất.

## 1. Đặt vấn đề

Ngày nay, mạng nơ ron là một lĩnh vực của học máy nói riêng và của trí tuệ nhân tạo nói chung. Mạng nơ ron có tính thời sự và phát triển mạnh bởi khả năng học, xử lý song song cũng như giải các bài toán có tính phi tuyến. Phân loại mạng nơ ron làm hai nhóm chính: mạng truyền thẳng và mạng hồi quy (truy hồi). Để NNs hoạt động, đầu tiên cần xác định cấu trúc và số nơ ron trong từng lớp (học cấu trúc), sau đó xác định các trọng số (học tham số). Trong nhóm mạng truyền thẳng, việc tính trọng số thực hiện bằng phương pháp học có giám sát với một số luật học điển hình như : luật học Perceptron [1], luật học Gradient Descent [2], luật học lan truyền ngược [3], ... và các biến thể của các luật học đó. Nhóm mạng hồi quy hầu hết sử dụng luật học Hebb [4].

Nội dung chủ yếu của bài báo này là nghiên cứu luật học cho mạng nơ ron tế bào (CNNs) – thuộc nhóm mạng nơ ron hồi quy. Trong CNNs, luật học Hebb thường được sử dụng để xác định bộ trọng số phản hồi  $A$  làm bộ nhớ liên kết [5]. Tuy nhiên, trong cấu trúc CNNs ngoài trọng số phản hồi  $A$  còn có tập trọng số đầu vào  $B$ , trọng số ngưỡng  $I$  cho nên luật Hebb chưa đáp ứng yêu cầu để tính đầy đủ bộ trọng số của mạng. Làm thế nào để tính được bộ trọng số đầy đủ  $[A B I]$  của CNNs nói chung hay SOCNNs nói riêng? Hiện nay, có nhiều công trình về các luật học nhằm tính bộ trọng số này cho mạng nơ ron tế bào bậc nhất [4], [6], [7], [8], [9], [10], [11]. Đặc biệt, C.GuKzelis [12], [13] đã đưa ra RPLA để tính bộ trọng số đầy đủ  $[A B I]$  cho mạng CNNs bậc nhất. Thực chất RPLA là một thuật toán học có tính toàn cục. Điều này có nghĩa là mục đích của thuật toán này không chỉ dùng để học cho một điểm đầu ra đạt trạng thái cân bằng ổn định mà còn dùng để học cho cả một miền ổn định của đầu ra [14].

Đóng góp của bài báo này là phát triển một luật học SORPLA cải tiến được xây dựng trên dựa [13] để áp dụng cho mạng nơ ron tế bào bậc hai. Mạng nơ ron tế bào bậc cao đã được một số tác giả công bố, chứng minh tính ổn định và xác định bộ trọng số  $A$  theo luật Hebb [15]. Để phát triển thuật toán học SORPLA, chúng tôi tích hợp các tín hiệu đầu vào, đầu ra SOCNNs về một ma trận tổng, bao gồm ma trận tín hiệu đầu vào, đầu ra. Tương tự, bộ ma trận trọng số cũng được ghép thành tổng các ma trận gồm: ma trận trọng số của đầu vào bậc nhất, bậc hai; ma trận trọng số phản hồi đầu ra bậc nhất, bậc hai và trọng số ngưỡng (điều này được làm rõ tại mục 3). Ưu điểm của thuật toán SORPLA này là xác định được toàn bộ các trọng số của mạng SOCNNs bao gồm  $[A1 B1 I A2 B2]$  mà [15] chưa thực hiện. Nhược điểm của SORPLA là yêu cầu thực hiện nhiều phép toán phức tạp, tuy nhiên máy tính hiện nay có khả năng hoàn toàn đáp ứng.

Nội dung của bài báo được chia làm 5 phần, trong đó phần hai nêu khái quát về cấu trúc mạng nơ ron bậc hai, phần ba nêu phương pháp học có giám sát sử dụng cho SOCNNs, phần bốn trình bày nội dung thuật toán SORPLA và cuối cùng phần năm là kiểm nghiệm tính đúng đắn của thuật toán qua chương trình tính toán và mô phỏng trên MATLAB.

## 2. Cấu trúc mạng nơ ron tế bào bậc hai

Đầu tiên, chúng ta sẽ đề cập lại khái niệm nơ ron tế bào của Chua [5], [16] [17], [18]. Một tế bào trong CNNs được định nghĩa là tổng của phép nhân chập giữa các tín hiệu đầu ra và các tín hiệu điều khiển bất kỳ đối với các tế bào lân cận

$C(k, l)$  và  $C(m, n)$  của tế bào  $C(i, j)$ , trong đó lân cận của tế bào là tập các tế bào quanh tế bào trung tâm  $C(i, j)$  với các bán kính  $r$  tương ứng. Bán kính ở đây được hiểu là số lớp kề cận với tế bào trung tâm  $C(i, j)$ . Khi  $r=1$  tức là lớp kề cận gần nhất bao gồm 08 tế bào tương tác cộng với chính nó tạo thành một bộ 09 tế bào gọi là láng giềng (neighbour) hay lân cận của  $C(i, j)$ , (bảng 1).

$C(i-1, j-1)$	$C(i-1, j)$	$C(i-1, j+1)$
$C(i, j-1)$	$C(i, j)$	$C(i, j+1)$
$C(i+1, j-1)$	$C(i, j+1)$	$C(i+1, j+1)$

**Bảng 1:** Bảng các tế bào lân cận của tế bào  $C(i, j)$  tương ứng  $r=1$

Theo công bố tại [15], cấu trúc mạng SOCNNs được xây dựng dựa trên CNNs [5] gồm bộ trọng số liên kết đầu vào bậc nhất, trọng số liên kết phản hồi đầu ra bậc nhất, trọng số liên kết đầu vào bậc hai và trọng số liên kết phản hồi đầu ra bậc hai và trọng số ngưỡng. Từ quan điểm xây dựng đó, phương trình động học cho mạng nơ ron tế bào bậc hai [15] như sau:

$$C \frac{dx_{ij}(t)}{dt} = -\frac{1}{R_x} x_{ij}(t) + \sum_{C(k,l) \in N_r(i,j)} A1_{i,j;k,l} y_{kl}(t) + \sum_{C(k,l) \in N_r(i,j)} B1_{i,j;k,l} u_{kl} + I + \sum_{C(k,l), C(m,n) \in N_r(i,j)} A2_{i,j;k,l;m,n} y_{kl}(t) y_{mn}(t) + \sum_{C(k,l), C(m,n) \in N_r(i,j)} B2_{i,j;k,l;m,n} u_{kl} u_{mn} \quad (1)$$

Hàm tương tác đầu ra [6]:

$$y_{ij}(t) = \frac{1}{2} |x_{ij}(t) + 1| - |x_{ij}(t) - 1| \quad (2)$$

trong đó:

$C$ : Tụ điện tuyến tính của SOCNNs, thường được chọn  $1\mu F$ ;

$R_x$ : Điện trở tuyến tính của SOCNNs, thường được chọn  $1k\Omega$  hoặc  $1M\Omega$ ;

$i, j$ : Thẻ hiện vị trí của tế bào  $C(i, j)$  trong SOCNNs;  $i, j \in N^*$

$r$ : Bán kính lân cận của tế bào  $C(i, j)$ , chọn  $r=1$ ;

$k, l, m, n$ : Thẻ hiện vị trí các tế bào lân cận của  $C(i, j)$  tương ứng với bán kính lân cận;  $k, l, m, n \in N^*$

$x_{ij}(t)$ : Tín hiệu trạng thái của tế bào  $C(i, j)$ ;

$y_{ij}(t)$ : Tín hiệu đầu ra của tế bào  $C(i, j)$ ;

$u_{ij}$ : Tín hiệu đầu vào của tế bào  $C(i, j)$ ;

$y_{kl}(t)$ : Tín hiệu đầu ra của SOCNNs;

$u_{kl}$ : Tín hiệu đầu vào của SOCNNs;

$A1_{i,j;k,l}$ : Ma trận trọng số phản hồi thành phần bậc nhất, kích thước  $(3 \times 3)$ ;

$A2_{i,j;k,l;m,n}$  : Ma trận trọng số phản hồi thành phần bậc hai, kích thước  $(9 \times 9)$ ;

$B1_{i,j;k,l}$  : Ma trận trọng số đầu vào của thành phần bậc nhất, kích thước  $(3 \times 3)$ ;

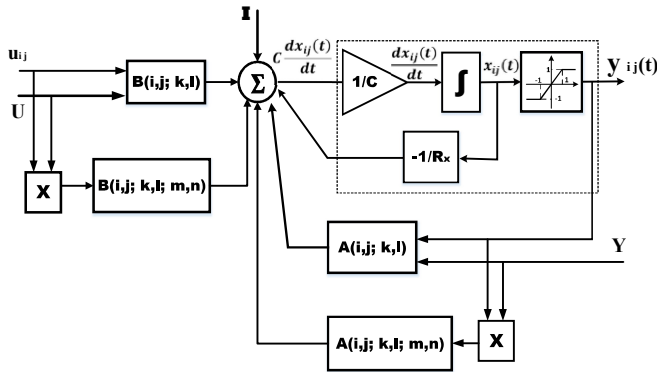
$B2_{i,j;k,l;m,n}$  : Ma trận trọng số đầu vào của thành phần bậc hai, kích thước  $(9 \times 9)$ ;

$M, N$ : Kích thước của mạng SOCNNs,  $M, N \in \mathbb{N}^*$

$I$ : Ma trận trọng số ngưỡng SOCNNs, kích thước  $(1 \times 1)$ ;

$U$ : Ma trận tín hiệu đầu vào lân cận của tế bào  $C(i, j)$ ;

$Y$ : Ma trận tín hiệu đầu ra lân cận của tế bào  $C(i, j)$ ;



Hình 1. Cấu trúc tổng quát SOCNNs

Các điều kiện ràng buộc ban đầu:

$$\begin{cases} |x_{ij}| \leq 1 \\ |u_{ij}| \leq 1 \end{cases} \quad (3)$$

Các điều kiện đảm bảo SOCNNs ổn định là các ma trận ma trận phản hồi  $A1(\cdot), A2(\cdot)$  phải đối xứng.

$$A1_{i,j;k,l} = A1_{k,l;i,j}$$

$$\begin{aligned} A2_{i,j;k,l;m,n} &= A2_{i,j;m,n;k,l} = A2_{k,l;i,j;m,n} = \\ &= A2_{k,l;m,n;i,j} = A2_{m,n;k,l;i,j} = A2_{m,n;i,j;k,l} \end{aligned} \quad (4)$$

$$1 \leq i, k, m \leq M; 1 \leq j, l, n \leq N$$

$$R_x > 0, C > 0$$

*Ghi chú:*  $R_x$  và  $C$  tại (1) được xác định trong mạch điện tử dùng nhằm thử nghiệm mô hình động học CNNs [5]. Về mặt toán học, trong nhiều công bố [5], [14], [19], nếu coi  $R_x = 1M\Omega$  và  $C = 1\mu F$  thì  $R_x$  và  $C$  khi đó có thể được bỏ qua.

Để tiện tính toán bộ trọng số cho SOCNNs (hình 1), với  $r=1$ , ta có thể chuyển các ma trận trọng số phản hồi bậc hai  $A2$  thành 09 bộ trọng số  $A21, A22, A23, A24, A25, A26, A27, A28, A29$ ; ma trận  $B2$  tương ứng thành 09 bộ  $B21, B22, B23, B24, B25, B26, B27, B28, B29$ . Mỗi một bộ trọng số này đi kèm với tích của các tín hiệu đầu vào hoặc đầu ra bậc hai, và được mô tả trong phương trình (6). Khi đó, về mặt hình thức các ma trận trọng số bậc hai tương đồng với các ma trận trọng số bậc nhất và có kích thước  $3 \times 3$ . Các điều kiện ràng buộc bậc hai đối với (4) được viết lại như sau:

$$\begin{aligned} A21_{i,j;r,s} &= A21_{r,s;i,j}; & A22_{i,j;r,s} &= A22_{r,s;i,j}; \\ A23_{i,j;r,s} &= A23_{r,s;i,j}; & A24_{i,j;r,s} &= A24_{r,s;i,j}; \\ A25_{i,j;r,s} &= A25_{r,s;i,j}; & A26_{i,j;r,s} &= A26_{r,s;i,j}; \\ A27_{i,j;r,s} &= A27_{r,s;i,j}; & A28_{i,j;r,s} &= A28_{r,s;i,j}; \\ A29_{i,j;r,s} &= A29_{r,s;i,j}; \end{aligned} \quad (5)$$

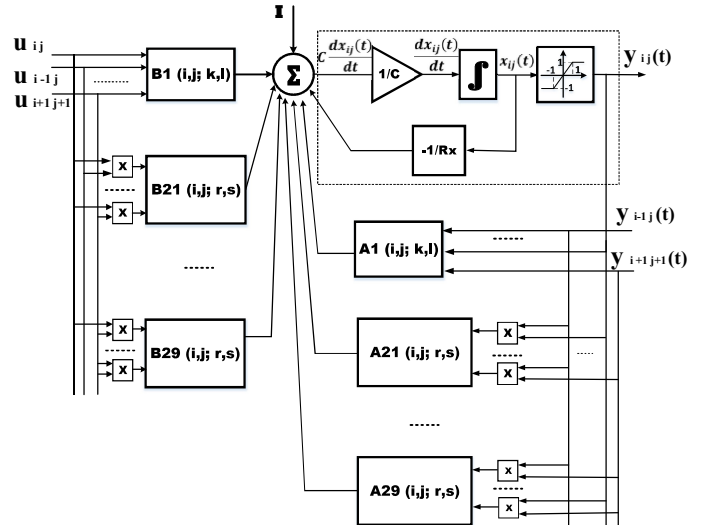
$$R_x > 0, C > 0$$

$$1 \leq i, r \leq M; 1 \leq j, s \leq N$$

Với cách đặt đó, phương trình (1) có thể viết lại như sau:

$$\begin{aligned} C \frac{dx_{ij}(t)}{dt} &= -\frac{1}{R_x} x_{ij}(t) + \sum_{C(k,l) \in N_r(i,j)} A1_{i,j;k,l} y_{kl}(t) + \\ &+ \sum_{C(k,l) \in N_r(i,j)} B1_{i,j;k,l} u_{kl} + I + \\ &+ \sum_{C(r,s) \in N_r(i,j)} A21_{i,j;r,s} y_{i-1,j-1}(t) y_{rs}(t) + \\ &+ \dots + \sum_{C(r,s) \in N_r(i,j)} A29_{i,j;r,s} y_{i+1,j+1}(t) y_{rs}(t) + \\ &+ \sum_{C(r,s) \in N_r(i,j)} B21_{i,j;r,s} u_{i-1,j-1} u_{rs} + \\ &+ \dots + \sum_{C(r,s) \in N_r(i,j)} B29_{i,j;r,s} u_{i+1,j+1} u_{rs} \end{aligned} \quad (6)$$

Từ (6), SOCNNs có thể được mô tả bằng mô hình hình học tương ứng như sau:



Hình 2. Sơ đồ cấu trúc mạng CNNs bậc hai tương đương trong đó:

$r, s$ : Thứ tự của tế bào tương ứng với các bộ trọng số đầu vào, đầu ra bậc hai

$A21(i,j;r,s), \dots, A29(i,j;r,s)$ : Ma trận trọng số phản hồi đầu ra bậc hai SOCNNs, kích thước  $(3 \times 3)$

$B21(i,j;r,s), \dots, B29(i,j;r,s)$ : Ma trận trọng số đầu vào bậc hai SOCNNs, kích thước  $(3 \times 3)$

### 3. Học có giám sát cho SOCNNs

Năm 1999, C. GuKzelis đã hoàn thiện thuật toán Perceptron hồi quy (RPLA) cho CNNs bậc nhất [13], trong đó, trọng số  $W$  bao gồm các trọng số phản hồi bậc nhất  $AI$ , trọng số đầu vào bậc nhất  $BI$ , và trọng số ngưỡng  $I$ , tức là:

$$W = \begin{bmatrix} AI^T & BI^T & I \end{bmatrix}^T \quad (7)$$

Khi đó, CNNs hồi quy chuẩn được chuyển sang dạng Perceptron truyền thẳng và việc tính toán trọng số mới được quy đổi  $W$  của CNNs có thể thực hiện theo thuật toán Perceptron hồi quy (RPLA).

Với cách tiệm cận tương tự, chúng ta hoàn toàn có thể chuyển mạng SOCNNs sang mạng truyền thẳng với cách sử dụng một thuật toán Perceptron hồi quy có cải biên. Trong trường hợp này, bộ trọng số mới được xác định như sau:

$$W = \begin{bmatrix} AI^T & A21^T & \dots & A29^T & BI^T & B21^T & \dots & B29^T & I \end{bmatrix}^T \quad (8)$$

trong đó:

$$\begin{aligned} AI^T &= a_{11} \ a_{12} \ a_{13} \ a_{14} \ a_{15} & BI^T &= b_{11} \ b_{12} \ b_{13} \ b_{14} \ b_{15} \\ A21^T &= a_{211} \ a_{212} \ a_{213} \ a_{214} \ a_{215} & B21^T &= b_{211} \ b_{212} \ b_{213} \ b_{214} \ b_{215} \\ A22^T &= a_{221} \ a_{222} \ a_{223} \ a_{224} \ a_{225} & B22^T &= b_{221} \ b_{222} \ b_{223} \ b_{224} \ b_{225} \\ A23^T &= a_{231} \ a_{232} \ a_{233} \ a_{234} \ a_{235} & B23^T &= b_{231} \ b_{232} \ b_{233} \ b_{234} \ b_{235} \\ A24^T &= a_{241} \ a_{242} \ a_{243} \ a_{244} \ a_{245} & B24^T &= b_{241} \ b_{242} \ b_{243} \ b_{244} \ b_{245} \\ A25^T &= a_{251} \ a_{252} \ a_{253} \ a_{254} \ a_{255} & B25^T &= b_{251} \ b_{252} \ b_{253} \ b_{254} \ b_{255} \\ \\ A26^T &= a_{261} \ a_{262} \ a_{263} \ a_{264} \ a_{265} & B26^T &= b_{261} \ b_{262} \ b_{263} \ b_{264} \ b_{265} \\ A27^T &= a_{271} \ a_{272} \ a_{273} \ a_{274} \ a_{275} & B27^T &= b_{271} \ b_{272} \ b_{273} \ b_{274} \ b_{275} \\ A28^T &= a_{281} \ a_{282} \ a_{283} \ a_{284} \ a_{285} & B28^T &= b_{281} \ b_{282} \ b_{283} \ b_{284} \ b_{285} \\ A29^T &= a_{291} \ a_{292} \ a_{293} \ a_{294} \ a_{295} & B29^T &= b_{291} \ b_{292} \ b_{293} \ b_{294} \ b_{295} \end{aligned}$$

Với :

$a_{11}, a_{12}, a_{13}, a_{14}, a_{15}$ : Toán hạng của ma trận phản hồi đầu ra bậc nhất;

$b_{11}, b_{12}, b_{13}, b_{14}, b_{15}$ : Toán hạng của ma trận đầu vào bậc nhất;

$a_{211}, a_{212}, a_{213}, \dots, a_{294}, a_{295}$ : Toán hạng của ma trận đầu ra phản hồi bậc hai quy đổi;

$b_{211}, b_{212}, b_{213}, \dots, b_{294}, b_{295}$ : Toán hạng của ma trận đầu vào bậc hai quy đổi;

Rõ ràng (8) có độ phức tạp hơn (7). Đó là nội dung bài toán cần giải quyết.

Mô hình (6) là một hệ động học phản hồi có thể gây tự kích dao động làm mất ổn định của hệ thống. Chúng tôi đã chứng minh tính ổn định [15] dựa vào các điều kiện ràng buộc (4) hoặc (5). Khi SOCNNs đạt trạng thái ổn định thì  $x_{ij} \ t$  là hằng số, dẫn tới  $\frac{dx_{ij} \ t}{dt} = 0$ , phương trình (5) có thể được viết lại như sau (chọn  $R_x = 1 \text{ M}\Omega$ ):

$$\begin{aligned} x_{ij} \ t &= \sum_{C(k,l) \in N_r(i,j)} AI \ i,j;k,l \ y_{kl} \ t + \sum_{C(k,l) \in N_r(i,j)} BI \ i,j;k,l \ u_{kl} + \\ &+ \sum_{p=1}^9 \sum_{C(k,l,m,n) \in N_r(i,j)} A2p \ i,j;k,l;m,n \ y_{kl} \ t \ y_{mn} \ t + \\ &+ \sum_{p=1}^9 \sum_{C(k,l,m,n) \in N_r(i,j)} B2p \ i,j;k,l;m,n \ u_{kl} u_{mn} + I \end{aligned} \quad (9)$$

Đặt ma trận đầu vào tổng như sau:

$$Y_{ij}^s = \begin{bmatrix} y_{ij}^{s*} \ \infty \end{bmatrix}^T \begin{bmatrix} y_{kl}^{s*} \ \infty \end{bmatrix} \begin{bmatrix} y_{ij}^{s*} \ \infty \end{bmatrix}^T \begin{bmatrix} u_{ij}^{s*} \end{bmatrix}^T \begin{bmatrix} u_{kl}^{s*} \end{bmatrix}^T \begin{bmatrix} u_{ij}^{s*} \end{bmatrix}^T \begin{bmatrix} 1 \end{bmatrix}^T \quad (10)$$

trong đó bộ ma trận  $y_{ij}^{s*} \ \infty$ ,  $u_{ij}^{s*}$  được định nghĩa như sau:

$$y_{ij}^{s*} \ \infty = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{bmatrix} = \begin{bmatrix} y_{i-1,j-1}^s \ \infty + y_{i+1,j+1}^s \ \infty \\ y_{i-1,j}^s \ \infty + y_{i+1,j}^s \ \infty \\ y_{i-1,j+1}^s \ \infty + y_{i+1,j-1}^s \ \infty \\ y_{ij-1}^s \ \infty + y_{ij+1}^s \ \infty \\ y_{ij}^s \ \infty \end{bmatrix} \quad (11)$$

$$u_{ij}^{s*} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{bmatrix} = \begin{bmatrix} u_{i-1,j-1}^s + u_{i+1,j+1}^s \\ u_{i-1,j}^s + u_{i+1,j}^s \\ u_{i-1,j+1}^s + u_{i+1,j-1}^s \\ u_{ij-1}^s + u_{ij+1}^s \\ u_{ij}^s \end{bmatrix} \quad (12)$$

$$y_{kl}^{s*} \ \infty = \begin{bmatrix} y_{i-1,j-1}^{s*} \ \infty & y_{i-1,j}^{s*} \ \infty & \dots & y_{i+1,j+1}^{s*} \ \infty \end{bmatrix} \quad (13)$$

$$u_{kl}^{s*} = \begin{bmatrix} u_{i-1,j-1}^{s*} & u_{i-1,j}^{s*} & \dots & u_{i+1,j+1}^{s*} \end{bmatrix}$$

Phương trình trạng thái CNNs bậc 2 ổn định (10) được xác định như sau:

$$x_{ij}^s \ \infty = \begin{bmatrix} Y_{ij}^s \end{bmatrix}^T * W \quad (14)$$

Từ phương trình (11)  $Y_{ij}^s$  đóng vai trò đầu vào tổng của tín hiệu đầu vào bậc nhất, tín hiệu đầu vào bậc hai và tín hiệu đầu ra bậc nhất, tín hiệu đầu ra bậc hai. Khi đó, việc tính toán trọng số của SOCNNs được thực hiện theo phương pháp học Thử-Sai-Chỉnh.

Giả thuyết tín hiệu đầu vào, đầu ra phân cực (giá trị  $+1$  hoặc  $-1$ ) nên hàm tương tác đầu ra của SOCNNs là hàm dấu  $\text{sgn}(\cdot)$ , tức là:

$$y_{ij}^s(\infty) = \text{sgn} \left[ x_{ij}^s \ \infty \right] = \text{sgn} \left[ \begin{bmatrix} Y_{ij}^s \end{bmatrix}^T * W \right] \quad (15)$$

Tại khoảng giá trị đầu vào  $x \in (-1; +1)$ , theo Chua và Thiran [20], giá trị đầu ra ở trạng thái ổn định được tính như sau:

$$y_{ij}^s \infty = \text{sgn} \left( \beta + \varphi_1 + \sum_{q=1}^9 \varphi_{2q} + I \right) \quad (16)$$

trong đó  $\beta, \varphi_1, \varphi_{21}, \dots, \varphi_{29}$  là các hệ số được tính toán như các công thức sau:

$$\beta = [a_{15} \ 0 \ -1/A] * y_{ij}^s \ 0 + [a_{215} \ 0 \ -1/A] * y_{i-1,j-1}^s \ 0 + \dots + [a_{295} \ 0 \ -1/A] * y_{i+1,j+1}^s \ 0$$

$$\varphi_1 = s_1 * b_{11} + s_2 * b_{12} + s_3 * b_{13} + s_4 * b_{14} + s_5 * b_{15}$$

$$\varphi_{21} = s_1 * b_{211} + s_2 * b_{212} + s_3 * b_{213} + s_4 * b_{214} + s_5 * b_{215} * u_{i-1,j-1}$$

$$\varphi_{22} = s_1 * b_{221} + s_2 * b_{222} + s_3 * b_{223} + s_4 * b_{224} + s_5 * b_{225} * u_{i-1,j}$$

$$\varphi_{23} = s_1 * b_{231} + s_2 * b_{232} + s_3 * b_{233} + s_4 * b_{234} + s_5 * b_{235} * u_{i-1,j+1}$$

$$\varphi_{24} = s_1 * b_{241} + s_2 * b_{242} + s_3 * b_{243} + s_4 * b_{244} + s_5 * b_{245} * u_{ij-1}$$

$$\varphi_{25} = s_1 * b_{251} + s_2 * b_{252} + s_3 * b_{253} + s_4 * b_{254} + s_5 * b_{255} * u_{ij}$$

$$\varphi_{26} = s_1 * b_{261} + s_2 * b_{262} + s_3 * b_{263} + s_4 * b_{264} + s_5 * b_{265} * u_{ij+1}$$

$$\varphi_{27} = s_1 * b_{271} + s_2 * b_{272} + s_3 * b_{273} + s_4 * b_{274} + s_5 * b_{275} * u_{i+1,j-1}$$

$$\varphi_{28} = s_1 * b_{281} + s_2 * b_{282} + s_3 * b_{283} + s_4 * b_{284} + s_5 * b_{285} * u_{i+1,j}$$

$$\varphi_{29} = s_1 * b_{291} + s_2 * b_{292} + s_3 * b_{293} + s_4 * b_{294} + s_5 * b_{295} * u_{i+1,j+1}$$

#### 4. Thuật toán Perceptron hồi quy cho mạng CNNs bậc hai (SORPLA)

Thuật toán Perceptron hồi quy cho SOCNNs được xây dựng dựa trên phương trình (10) khi mạng ổn định với đầu ra phân cực. Giá trị đầu ra ổn định được xác định theo (16).

Để tính toán bộ trọng số cho mạng SOCNNs tại trạng thái ổn định, đề xuất phương pháp SORPLA có lưu đồ thuật toán như hình 3.

Sai lệch giữa đầu ra thực tế và đầu ra mong muốn được tính như sau [13]:

$$\begin{aligned} \varepsilon w[n] &= \frac{1}{2} \sum_{i,j,s} y_{ij}^s(\infty) \times (y_{ij}^s(\infty) - d_{ij}^s) \\ &= \sum_{i,j \in D^+} y_{ij}^s(\infty) - \sum_{i,j \in D^-} y_{ij}^s(\infty) \end{aligned} \quad (17)$$

trong đó:

$s$ : số tập mẫu huấn luyện của mạng

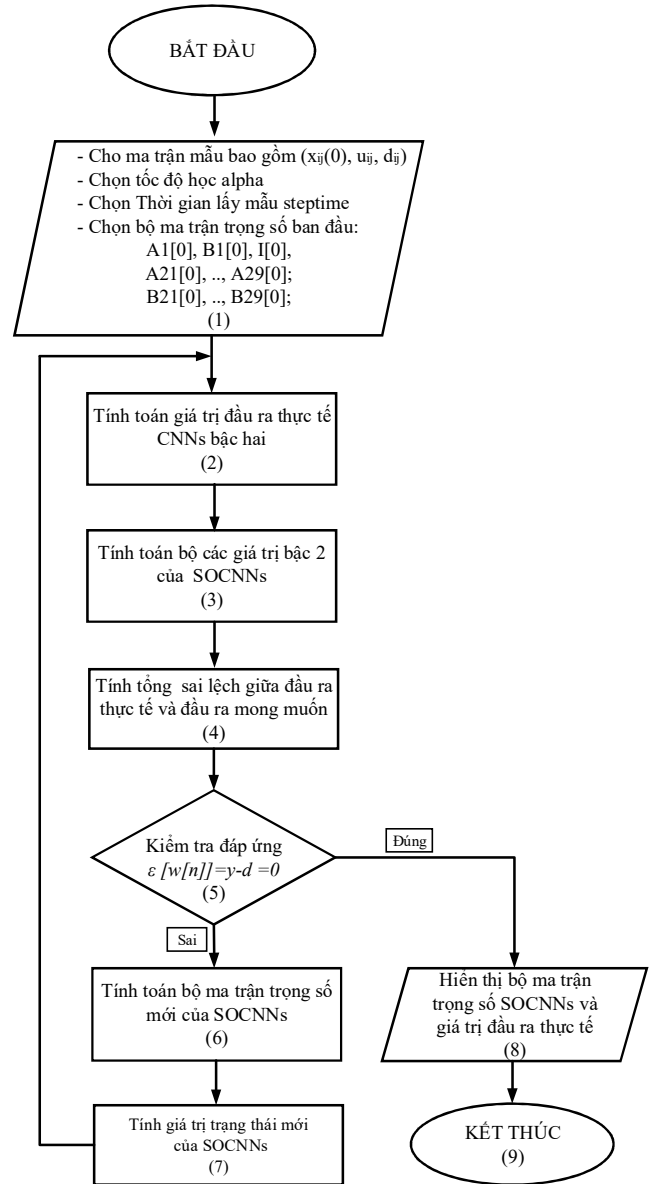
$\varepsilon[w \ n]$ : tổng vec tơ sai lệch đầu ra mạng SOCNNs;

$d_{ij}^s$ : giá trị đầu ra mong muốn mạng SOCNNs;

$y_{ij}^s(\infty)$ : giá trị đầu ra thực tế ổn định mạng SOCNNs;

$D^+, D^-$ : miền giá trị đầu ra sai lệch tính toán dựa theo nguyên tắc sau:

$$D^+ = \begin{cases} y_{ij}^s(\infty) = 1 \\ d_{ij}^s = -1 \end{cases} \quad D^- = \begin{cases} y_{ij}^s(\infty) = -1 \\ d_{ij}^s = +1 \end{cases}$$



Hình 3. Lưu đồ thuật toán học cho mạng CNNs bậc hai

Việc tính toán giá trị trọng số của mạng CNNs bậc hai khi đó được thực hiện như sau:

$$W \ n+1 = [W \ n - \Delta W \ n]$$

$$\Delta W \ n = \alpha * \left( \sum_{i,j,s \in D^+} Y_{ij}^s \ n - \sum_{i,j,s \in D^-} Y_{ij}^s \ n \right) \quad (18)$$

trong đó:

$\alpha$ : tốc độ học của mạng, thường được chọn là một giá trị hằng số dương;

Trong quá trình thực hiện thuật toán SORPLA, bộ vec tơ trọng số được điều chỉnh khi xét ảnh hưởng của từng tế bào cho tất cả các mẫu huấn luyện. Khi sai số giữa đầu ra thực tế và đầu ra mong muốn tại mỗi tế bào bằng không, thì bỏ qua cập nhập trọng số tại tế bào đó và chuyển sang tế bào tiếp theo nguyên tắc từ trái qua phải và từ trên xuống dưới. Thuật toán chỉ kết thúc khi tổng các sai lệch của tất cả các điểm ảnh trong tập mẫu đạt giá trị bằng 0.

### 5. Thử nghiệm SOCNNs dựa trên SORPLA

Khi đề xuất cấu trúc SOCNNs hồi quy chúng ta cần tiến hành ba pha: i) pha một: kiểm tra tính ổn định của cấu trúc truy hồi này do tín hiệu phản hồi đầu ra có thể gây tự kích dao động (đã chứng minh [15]); ii) pha hai: pha học để xác định các giá trị bộ trọng số  $\{A B I\}$ ; sẽ tiến hành ở mục 5.1; iii) pha ba: pha chạy thử bộ trọng số đã xác định trên cấu trúc đó và sẽ thực hiện tại mục 5.2.

#### 5.1. Tính trọng số SORPLA với bộ ảnh mẫu cho trước

Hiện nay, có 03 bài toán xử lý ảnh chủ yếu mà thuật toán SORPLA có khả năng áp dụng như sau: Dò biên, dò đỉnh và phủ đều vùng ảnh [13]. Trong bài báo này, chúng áp dụng cho bài toán dò biên.

Vai trò của thuật toán SORPLA ở đây dùng để xác định bộ trọng số dựa trên bộ ảnh mẫu cho trước như là mẫu học. Bộ mẫu học cho mạng SOCNNs gồm 03 thành phần: i) một ảnh tín hiệu đầu vào  $u_{ij}$ , ii) một ảnh trạng thái ban đầu  $x_{ij}(0)$ , và iii) một ảnh đầu ra mong muốn  $d_{ij}$ . Cả ba ảnh đó đều có kích thước 16x16. Trong thực tế cần có nhiều bộ mẫu ảnh học, tuy nhiên để đơn giản chúng tôi thử nghiệm cho 01 bộ mẫu (hình 4). Bộ trọng số này dùng như là một cửa sổ để dò biên ảnh.

Thuật toán và trình tự thực hiện SORPLA như sau:

Input: Cho  $\{u_{ij}, x_{ij}(0), d_{ij}\}; \alpha=0.001$ ; thời gian cập nhập trạng thái mới:  $Steptime=0.1s$

Output: Bộ trọng số  $\{A1, B1, I, A21..A29, B21..B29\}$

Bước khởi tạo: Chọn bộ vec tơ ma trận trọng số SOCNNs cho tín hiệu đầu vào, đầu ra làm giá trị khởi tạo. Bộ trọng số khởi tạo đưa ra dựa trên bộ trọng số tại [5], [15].

Đối với tín hiệu bậc nhất:

$$A1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}; B1 = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}; I = 0$$

Đối với tín hiệu bậc hai:

$$A21 = A22 = .. = A29 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

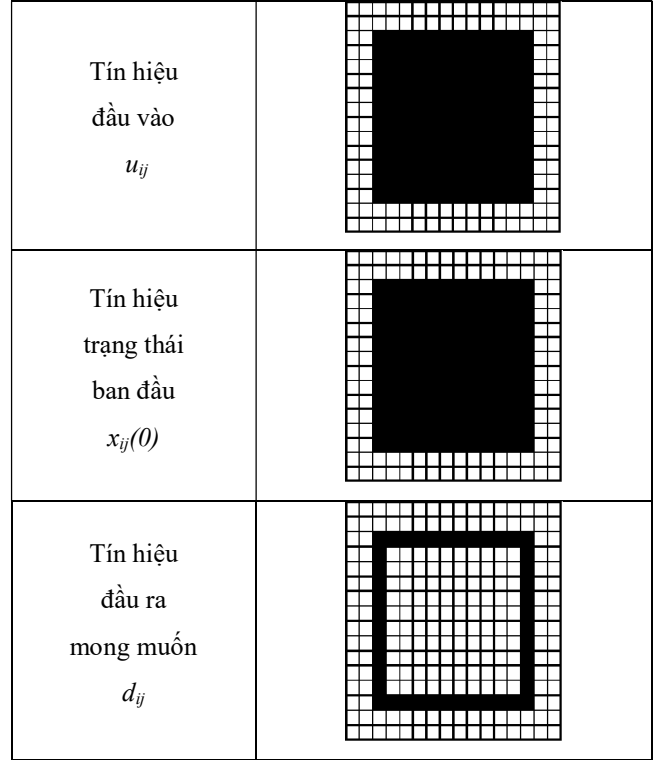
$$B21 = B22 = .. = B29 = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}$$

- Tính giá trị đầu ra thực tế của SOCNNs theo (17). (Tại vòng tính toán đầu tiên do  $x_{ij}(0)$  phân cực nên giá trị đầu ra  $y_{ij}(0)$  bằng trạng thái ban đầu  $x_{ij}(0)$ ).

- Tính tổng sai lệch giữa tín hiệu đầu ra thực tế và tín hiệu đầu ra mong muốn:

$$\varepsilon[w 0] = \sum_{i,j \in D^+} y_{ij}^s 0 - \sum_{i,j \in D^-} y_{ij}^s 0 = 100$$

Do sai số  $\varepsilon[w 0] = 100 \neq 0$  nên phải tính bộ trọng số theo các vòng lặp tiếp theo.



Hình 4. Tập mẫu huấn luyện SOCNNs

Vòng lặp đầu tiên:

Bước 1: Tính bộ trọng số cho vòng lặp đầu tiên theo (19).

Kết quả vòng lặp đầu tiên như sau:

$$A1 1 = \begin{bmatrix} 0.1480 & 0.1480 & 0.1480 \\ 0.1480 & 0.076 & 0.1480 \\ 0.1480 & 0.1480 & 0.1480 \end{bmatrix}$$

$$B1 1 = \begin{bmatrix} 0.1449 & 0.1449 & 0.1449 \\ 0.1449 & 0.1449 & 0.1449 \\ 0.1449 & 0.1449 & 0.1449 \end{bmatrix} \quad I 1 = -0.0043$$

$$A21 1 = A22 1 = .. = A29 1 = \begin{bmatrix} -0.0321 & -0.0218 & -0.0321 \\ -0.0218 & 0.0139 & -0.0218 \\ -0.0321 & -0.0218 & -0.0321 \end{bmatrix}$$

$$B21 1 = B22 1 = .. = B29 1 = \begin{bmatrix} -0.0359 & -0.0359 & -0.0359 \\ -0.0359 & -0.0359 & -0.0359 \\ -0.0359 & -0.0359 & -0.0359 \end{bmatrix}$$

Bước 2: Tính trạng thái mới  $x_{ij}^s(\infty)$  dựa trên bộ vec tơ trọng số mới theo (10).

Bước 3: Tính đầu ra thực tế  $y_{ij}^s(\infty)$  theo (17).

Bước 4: Tính tổng sai lệch  $|\varepsilon[w 1]|$  theo (18).

Bước 5: Khi sai lệch  $|\varepsilon[w 1]| > 0$ , thực hiện tiếp tục vòng lặp tiếp theo. Ngược lại, khi  $|\varepsilon[w 1]| = 0$ , dừng.

Vòng lặp tiếp theo:

Lặp lại các bước 1, 2, 3, 4, 5 như vòng lặp thứ nhất.

Trong chương trình tính toán, sau 55 vòng lặp, tổng sai lệch  $\|\varepsilon[w_{55}]\| = 0$ , thuật toán dừng. Đây chính là đầu ra cuối cùng của bộ trọng số cho SOCNNs:

$$A1_{55} = \begin{bmatrix} -0.0290 & -0.1059 & 0.0142 \\ -0.1606 & -0.1516 & -0.1606 \\ 0.0142 & -0.1059 & -0.0290 \end{bmatrix}$$

$$B1[55] = \begin{bmatrix} 0.1202 & 0.1906 & 0.1237 \\ 0.2475 & 0.1639 & 0.2475 \\ 0.1237 & 0.1906 & 0.1202 \end{bmatrix} \quad I1[55] = -3.8775$$

$$A21_{55} = \begin{bmatrix} 1.4029 & 1.4325 & 1.3147 \\ 1.4479 & 0.6631 & 1.4479 \\ 1.3147 & 1.4325 & 1.4029 \end{bmatrix}$$

$$A22_{55} = \begin{bmatrix} -0.0462 & 0.0282 & -0.0586 \\ 0.0342 & 0.0786 & 0.0342 \\ -0.0586 & 0.0282 & -0.0462 \end{bmatrix}$$

$$A23_{55} = \begin{bmatrix} 0.1149 & 0.1911 & -0.0586 \\ 0.0342 & 0.0789 & 0.0342 \\ -0.0586 & 0.1911 & 0.1149 \end{bmatrix}$$

$$A24_{55} = \begin{bmatrix} 0.0253 & 0.1883 & 0.0127 \\ 0.0461 & 0.1687 & 0.0461 \\ 0.0127 & 0.1883 & 0.0253 \end{bmatrix}$$

$$A25_{55} = \begin{bmatrix} -1.2172 & -1.3127 & -1.2183 \\ -1.1835 & -0.6149 & -1.1835 \\ -1.2183 & -1.3127 & -1.2172 \end{bmatrix}$$

$$A26_{55} = \begin{bmatrix} -0.5511 & -0.6042 & -0.5141 \\ -0.5985 & -0.2410 & -0.5985 \\ -0.5141 & -0.6042 & -0.5511 \end{bmatrix}$$

$$A27_{55} = \begin{bmatrix} -0.5511 & -0.6042 & -0.5141 \\ -0.5985 & -0.2410 & -0.5985 \\ -0.5141 & -0.6042 & -0.5511 \end{bmatrix}$$

$$A28_{55} = \begin{bmatrix} -0.2327 & -0.2804 & -0.2141 \\ -0.1922 & -0.1645 & -0.1922 \\ -0.2141 & -0.2804 & -0.2327 \end{bmatrix}$$

$$A29_{55} = \begin{bmatrix} 1.0956 & 1.0968 & 1.0842 \\ 1.1640 & 0.5636 & 1.1640 \\ 1.0842 & 1.0968 & 1.0956 \end{bmatrix}$$

$$B21_{55} = \begin{bmatrix} 1.4253 & 1.4849 & 1.4256 \\ 1.4654 & 0.7987 & 1.4654 \\ 1.4256 & 1.4849 & 1.4253 \end{bmatrix}$$

$$B22_{55} = \begin{bmatrix} -0.3936 & -0.2356 & -0.3872 \\ -0.5009 & -0.1746 & -0.5009 \\ -0.3872 & -0.2356 & -0.3936 \end{bmatrix}$$

$$B23_{55} = \begin{bmatrix} 0.0570 & -0.0445 & 0.0101 \\ -0.0508 & -0.0235 & -0.0508 \\ 0.0101 & -0.0445 & 0.0570 \end{bmatrix}$$

$$B24_{55} = \begin{bmatrix} -0.4399 & -0.5134 & -0.4185 \\ -0.2747 & -0.1777 & -0.2747 \\ -0.4185 & -0.5134 & -0.4399 \end{bmatrix}$$

$$B25_{55} = \begin{bmatrix} -0.6549 & -0.9667 & -0.6511 \\ -0.8818 & -0.6073 & -0.8818 \\ -0.6511 & -0.9667 & -0.6549 \end{bmatrix}$$

$$B26_{55} = \begin{bmatrix} -0.6525 & -0.7205 & -0.6754 \\ -0.5635 & -0.3127 & -0.5635 \\ -0.6754 & -0.7205 & -0.6525 \end{bmatrix}$$

$$B27_{55} = \begin{bmatrix} -0.6525 & -0.7205 & -0.6754 \\ -0.5635 & -0.3127 & -0.5635 \\ -0.6754 & -0.0445 & -0.6525 \end{bmatrix}$$

$$B28_{55} = \begin{bmatrix} 0.4513 & 0.1292 & 0.4563 \\ 0.1365 & -0.1132 & -0.0508 \\ 0.4563 & 0.1292 & 0.4513 \end{bmatrix}$$

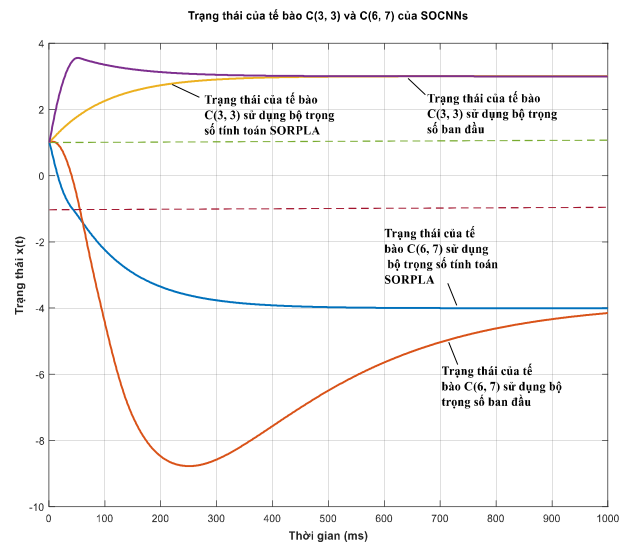
$$B29_{55} = \begin{bmatrix} 0.9910 & 1.0094 & 0.9357 \\ 0.8012 & 0.4265 & 0.8012 \\ 0.9357 & 1.0094 & 0.9910 \end{bmatrix}$$

## 5.2. Thử nghiệm bộ trọng số trên SOCNNs

Sử dụng kết quả tính toán bộ ma trận trọng số trên, chúng tôi tiến hành mô phỏng hoạt động SOCNNs trên MATLAB với bộ dữ liệu mẫu đầu vào như hình 4; Áp dụng chương trình như đề xuất ở [15] để thử nghiệm hoạt động SOCNNs với hai bộ trọng số khác nhau, cụ thể:

*Thử nghiệm 1:* mô phỏng cho trạng thái  $x_{ij}(t)$  tại hai tế bào  $C(3,3)$ ,  $C(6,7)$  cho bộ trọng số khởi tạo ban đầu (đã cho tại mục 4: Bước khởi tạo).

Kết quả thu được trạng thái  $x_{33}(t)$  cho tế bào  $C(3,3)$  minh họa hội tụ về giá trị  $+3$  (hình 5). Trạng thái này qua hàm tương tác đầu ra thu được giá trị  $+1$ . Tương tự, trạng thái  $x_{67}(t)$  cho tế bào  $C(6,7)$  minh họa theo hội tụ về giá trị  $-4$  (hình 5). Trạng thái này qua hàm tương tác đầu ra  $y = \text{sgn}(x_{67}(t))$  thu được giá trị  $-1$ .



Hình 5. Quá trình quá độ của  $x(t)$  tại tế bào  $C(3,3)$ ;  $C(6,7)$  trong SOCNNs

*Thử nghiệm 2:* mô phỏng trạng thái  $x_{33}(t)$ ,  $x_{67}(t)$  tại tế bào  $C(3, 3)$  và  $C(6, 7)$  của SOCNNs cho bộ trọng số tính toán từ SORPLA. Kết quả thu được trạng thái  $x_{33}(t)$ ,  $x_{67}(t)$  hội tụ về giá trị  $x_{33}(t)=+3$ , trạng thái của tế bào  $C(6, 7)$  hội tụ về giá trị  $x_{67}(t)=-4$  (hình 5), tương ứng.

*Đánh giá và so sánh:*

1) Hai bộ trọng số khởi tạo ban đầu và bộ trọng số tính toán từ SORPLA đều cho SOCNNs đạt trạng thái ổn định;

2) Tuy nhiên bộ trọng số sau khi sử dụng SORPLA có những ưu điểm như sau:

- Về độ quá điều chỉnh: Bộ trọng số sử dụng SORPLA không xảy ra tình trạng độ quá điều chỉnh. Cụ thể đối với bộ trọng số khởi tạo, biên độ trạng thái  $x_{33}(t)$  của tế bào  $C(3, 3)$  có độ quá điều chỉnh tăng khoảng 20% rồi giảm dần về giá trị ổn định  $x_{33}(t)=3$ ; trong tự, tại tế bào  $C(6, 7)$ , độ quá điều chỉnh vượt khoảng 120% rồi giảm dần về giá trị ổn định  $x_{33}(t)=-4$ ;

- Về thời gian quá độ: Đối với trạng thái  $x_{33}(t)$  của tế bào  $C(3, 3)$ , sau khoảng 400ms, trạng thái  $x_{33}(t)$  sử dụng hai bộ trọng số nêu trên đều tiến đến giá trị ổn định. Tuy nhiên, tại tế bào  $C(6, 7)$ , thời gian quá độ của SOCNNs sử dụng bộ trọng số tính toán SORPLA (400ms) giảm khoảng 2,5 lần so với thời gian quá độ của SOCNNs sử dụng bộ trọng số khởi tạo ban đầu (1000ms).

3) Thử nghiệm cũng được tiến hành cho các trạng thái còn lại của ảnh có kích thước 16x16 trên SOCNNs, và thu được kết quả tương tự.

Từ quá trình mô phỏng trên có thể khẳng định mạng SOCNNs sử dụng bộ trọng số tính theo thuật toán SORPLA có đặc tính của quá trình quá độ có chất lượng cao hơn so với sử dụng bộ trọng số khởi tạo.

## 6. Kết luận

Đóng góp lớn nhất của bài báo là xây dựng một thuật toán SORPLA cho mạng nơ ron bậc hai. Đây có thể coi là sự phát triển của thuật toán RPLA [13] áp dụng cho một cấu trúc mới là SOCNNs – đại diện cho mạng nơ ron bậc cao.

Bài báo đã trình bày chi tiết thuật toán SORPLA dùng cho SOCNNs, với các bước trình tự thực hiện theo lưu đồ chương trình. Chúng tôi cũng tiến hành thử nghiệm trên phần mềm MATLAB với hai bộ trọng số cho SOCNNs và thu được một số kết quả như sau:

1) Trạng thái của mạng đảm bảo ổn định bền vững. Trạng thái đó không xảy ra trường hợp quá điều chỉnh. Một số trường hợp thì cho thời gian ổn định nhanh hơn (nhanh gấp 2,5 lần so với bộ trọng lựa chọn tùy ý).

2) Việc tính toán trọng số SOCNNs hiển nhiên phức tạp do phải thực hiện nhiều phép tính cho từng tế bào (điểm ảnh) trong một ma trận ảnh lớn.

3) Từ thử nghiệm ảnh 16x16 cho thấy mạng nơ ron tế bào bậc hai có khả năng ứng dụng cho các bài toán xử lý ảnh [21], [22]. Do SOCNNs là mạng hồi quy tương đương với mạng nhiều lớp truyền thẳng và dữ liệu đầu vào hai chiều thích hợp cho các bài toán xử lý ảnh (cũng hai chiều) đòi hỏi tốc độ cao, dữ liệu lớn.

Thuật toán SORPLA tại bài báo này có thể coi là một công cụ tìm kiếm bộ trọng số của SOCNNs đảm bảo ảnh đầu ra thực tế tương đồng với ảnh đầu ra mong muốn.

Hướng nghiên cứu tiếp theo của bài báo là sử dụng thuật toán đã phát triển áp dụng cho các bài toán xử lý ảnh trong thực tế [23] [24].

## 7. Tài liệu tham khảo

- [1] Mutasem khalil Alsmadi, Mutasem khalil Alsmadi, Shahrul Azman Noah, Ibrahim Almarashdah, "Performance Comparison of Multi-layer Perceptron (Back Propagation, Delta Rule and Perceptron) Algorithms in Neural Networks," in *IEEE International Advance Computing Conference*, 2009.
- [2] Liang Gong, Chengliang Liu, Yanming Li, Fuqing Yuan, "Training Feed-forward Neural Networks Using the Gradient Descent Method with the Optimal Step-size," *Computational Information Systems*, pp. 1359-1371, 2012.
- [3] C.-H. Chang, "Deep and Shallow Architecture of Multilayer Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2477-2486, 2015.
- [4] B. Mirzai, Z. Cheng, G. S. Moschytz, "Learning Algorithms for Cellular Neural Networks," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Monterey, CA, USA, 1998.
- [5] L.O. Chua, L. Yang, "Cellular Neural Networks: Theory," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, pp. 1257-1272, 1988.
- [6] F. Zou, S. Schwarz, J.A. Nossek, "Cellular Neural Network Design Using a Learning Algorithm," in *IEEE International Workshop on Cellular Neural Networks and Their Applications*, Budapest, Hungary, 1990.
- [7] H. Mizutani, "A New Learning Method for Multilayered Cellular Neural Networks," in *Third IEEE International Workshop on Cellular Neural Networks and Their Applications*, Rome, Italy, 1994.
- [8] Sergey Pudov, "Learning of Cellular Neural Networks," *Future Generation Computer Systems*, vol. 17, pp. 689-697, 2001.
- [9] S. Taraglio, A. Zanela, "Cellular Neural Networks: A Genetic Algorithm for Parameters Optimization in Artificial Vision Applications," in *Fourth IEEE International Workshop on Cellular Neural Networks and Their Applications Proceedings*, Seville, Spain, 1996.
- [10] Michele Zamparelli, "Genetically Trained Cellular Neural Networks," *Neural Networks*, vol. 10, no. 6, pp. 1143-1151, 1997.
- [11] C. Guzelis, "Supervised Learning of the Steady-state Outputs in Generalized Cellular Networks," in *Proceedings Second International Workshop on Cellular Neural Networks and Their Applications*, Munich, Germany, 1992.
- [12] C. Guzelis, S. Karamahmut, "Recurrent Perceptron Learning Algorithm for Completely Stable Cellular Neural Networks," in *Proceedings of the Third IEEE International Workshop on Cellular Neural Networks and Their Applications*, Rome, Italy, 1994.
- [13] C. GuK zelis, S. Karamamut, I0 . Genc, "A Recurrent Perceptron Learning Algorithm for Cellular Neural Networks," *Springer-Verlag*, pp. 296-309, 1999.
- [14] J. Nossek, "Design and Learning with Cellular Neural Networks +," *International Journal of Circuit Theory and Application*, vol. 24, pp. 15-24, 1996.
- [15] Nguyễn Quang Hoan, Nguyễn Tài Tuyên, Dương Đức Anh, "Architecture and Stability of the Second - Order Cellular Neural," *UTEHY Journal of Science and Technology*, vol. 27, pp. 91-97, 2020.
- [16] L.O. Chua, L. Yang, "Cellular Neural Networks: Applications," *IEEE Transactions on Circuits and Systems*, vol. 35, pp. 1273-1290, 1988.
- [17] Adnène Arbi, Jinde Cao, Mohssine Es-saiydy, Mohammed Zarhouni, Mohamed Zitane, "Dynamics of Delayed Cellular Neural Networks in the Stepanov Pseudo Almost Automorphic Space," *Discrete and Continuous Dynamical Systems - Series S*, vol. 15, no. 11, pp. 3097-3109, 2022.
- [18] Davut Aydogan, "Training of Cellular Neural Networks and Application to Geophysics," *Istanbul Yerbilimleri Dergisi*, vol. 1, no. 26, pp. 53-64, 2013.



- [19] A. Slavova, *Cellular Neural Networks: Dynamics and Modelling*, Springer, 2003.
- [20] L.O. Chua, P. Thiran, "An Analytic Method for Designing Simple Cellular Neural Networks," *IEEE Transactions on Circuits and Systems*, vol. 38, no. 11, pp. 1332-1341, 1991.
- [21] Andras Fülöp, András Horváth, "Application of Cellular Neural Networks in Semantic Segmentation," in *IEEE Explore*, 2021.
- [22] András Horváth, Michael Hillmer, Qiuwen Lou, X. Sharon Hu, Michael Niemier, "Cellular Neural Network Friendly Convolutional Neural Networks — CNNs with CNNs," in *IEEE*, Lausanne, Switzerland, Design, Automation & Test in Europe Conference & Exhibition.
- [23] Luping Ji, Mingzhe Chang, Yulin Shen, Qian Zhang, "Recurrent Convolutions of Binary-Constraint Cellular Neural Network for Texture Recognition," *Neurocomputing*, vol. 387, pp. 161-171, 2020.
- [24] Huaqing Li, Xiaofeng Liao, Chuandong Li, Hongyu Huang, Chaojie Li, "Edge Detection of Noisy Images Based on Cellular Neural Networks," *Commun Nonlinear Sci Numer Simulat*, vol. 16, pp. 3746-3759, 2011.