

# MỘT THUẬT TOÁN VỀ HUẤN LUYỆN MẠNG NEURAL NETWORKS TRÊN CƠ SỞ PHƯƠNG PHÁP CONJUGATE GRADIENT

## AN ALGORITHM FOR TRAINING OF NEURAL NETWORKS BASED ON CONJUGATE GRADIENT

Nguyễn Sĩ Dũng\*, Lê Hoài Quốc\*\*

(\*) Khoa Cơ khí, trường Đại học Công nghiệp thành phố Hồ Chí Minh. [nsidung@yahoo.com](mailto:nsidung@yahoo.com)

(\*\*) Khoa Cơ khí, trường Đại học Bách khoa thành phố Hồ Chí Minh.

---

### BẢN TÓM TẮT

Mạng Neuron nhân tạo có cấu trúc gồm nhiều neuron nằm trong các lớp khác nhau. Tín hiệu ở lớp vào và lớp ra liên hệ với nhau qua các neuron trung gian nằm trên một hay một số lớp ẩn thông qua ma trận trọng số mạng. Quá trình huấn luyện mạng, hay còn gọi là quá trình học của mạng trong học giám sát bao hàm việc điều chỉnh, cập nhật ma trận trọng số sao cho ứng với tập tín hiệu vào xác định, tín hiệu ra của mạng tiệm cận tới giá trị mong muốn. Một trong những vấn đề cần quan tâm trong huấn luyện mạng, đặc biệt trong huấn luyện trực tuyến là tốc độ hội tụ của quá trình huấn luyện. Trong nghiên cứu này, chúng tôi xây dựng một thuật toán mới (TT\*) được xây dựng trên cơ sở phương pháp Conjugate Gradient, theo đó, bước dịch chuyển tại mỗi vòng lặp trong [1],[2],[3] được điều chỉnh bởi một hệ số hiệu chỉnh nhằm đưa bước dịch chuyển trọng số tiến gần hơn điểm cực tiểu theo hướng dịch chuyển cận tối ưu đã xác định. Kết quả thí nghiệm kiểm chứng cho thấy nếu ma trận trọng số của mạng không lớn, sử dụng thuật toán TT\* có tốc độ hội tụ cao hơn thuật toán [1],[2],[3].

### ABSTRACT

Feed forward neural networks are composed in which the input layer of neurons is connected to the output layer through one or more hidden layers of intermediate neurons. The training process of neural networks involves adjusting the weights till a desired input, output relationship is obtained. The paper presents our research on application of the Conjugate Gradient method so that, we found out the algorithm to be as fit as possible to reach the minimized point at each iteration of neural network training. A coefficient to adjust the step of training to be used. Simulation results showed that new algorithm could reach the global optimal point faster than the algorithm of [1],[2],[3] if the H matrix is not too large.

### I. ĐẶT VẤN ĐỀ

Mạng Neuron nhân tạo (ANN) cho phép chúng ta thiết kế các hệ thống phi tuyến với một số lượng lớn các tín hiệu vào-ra. ANN hoàn toàn thích hợp với các ứng dụng trong môi trường động bằng cách tự thay đổi thích nghi cấu trúc hoặc thông số của mạng [1] thích ứng với sự thay đổi của môi trường. Vì vậy, ANN đã được ứng dụng

rộng rãi trong rất nhiều lĩnh vực khác nhau. Trong nhận dạng và điều khiển, đặc biệt trong lĩnh vực kỹ thuật robot, robot tự hành, ANN đã được ứng dụng để phát triển một hình thức điều khiển mới được gọi là điều khiển ứng xử (behavioural control) hoặc ứng xử phản ứng (reaction behaviour) [11], trong đó, một tác vụ phức tạp sẽ được giải quyết bằng cách kết hợp một cách thỏa đáng các ứng xử cơ bản cùng

một lúc hoặc tuần tự theo thời gian. Cơ sở để kết hợp là tri thức mạng, được hình thành, tích lũy và duy trì trong quá trình học, huấn luyện và ứng dụng mạng.

Huấn luyện ANN để thực hiện một ứng dụng cụ thể là điều chỉnh, xác lập các giá trị trọng số liên kết - còn được gọi là bộ trọng số kết nối của mạng (trong bài báo này ký hiệu là  $W$ ) - giữa các neuron trong mạng và của các bias. Trong học giám sát, các cặp tín hiệu vào ra được dùng để huấn luyện mạng sao cho tín hiệu ra của mạng tiệm cận tới tín hiệu ra mong muốn của hệ thống (Hình 1).

Sai số dự báo là sai lệch giữa tín hiệu ra mong muốn và tín hiệu ra của mạng:

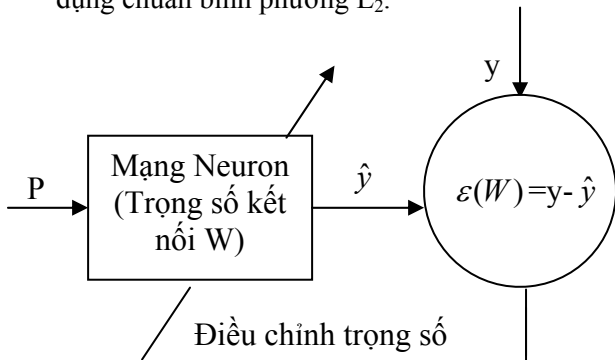
$$e(i, W) = y(t) - \hat{y}(t, W) \quad (1)$$

Bộ trọng số của mạng nhận được sau huấn luyện chính là ma trận  $W$  làm tối thiểu tiêu chuẩn ước lượng:

$$E(W) = \frac{1}{P} \sum_{i=1}^P l(e(i, W)) \rightarrow \min \quad (2)$$

trong đó,  $P$  là số mẫu dữ liệu huấn luyện.

$l(.)$  là chuẩn xác định dương, ở nghiên cứu này sử dụng chuẩn bình phương  $L_2$ .



Hình 1. Sơ đồ huấn luyện ANN trong học giám sát. Tín hiệu ra của mạng  $\hat{y}$ , tín hiệu ra mong muốn của đối tượng  $y$ , tín hiệu vào của mạng  $P$ .

Trong lĩnh vực điều khiển robot tự hành, các thông số môi trường hoặc chúng ta không thể xác định hết vì có rất nhiều các tình huống có thể xảy ra; hoặc được xác định trước nhưng giá trị của

chúng luôn thay đổi theo thời gian. Trong những trường hợp này, huấn luyện mạng trực tuyến (Online) được đặt ra, và do đó vấn đề tốc độ huấn luyện mạng phải được đặt lên hàng đầu.

Hiện có nhiều thuật toán về huấn luyện ANN phát triển trên cơ sở cực tiểu hoá khai triển Taylor hàm sai lệch tín hiệu ra (2), có thể phân làm ba nhóm chính: nhóm các phương pháp Gradient Descent, nhóm các phương pháp Conjugate Gradient và nhóm các phương pháp Newton.

Trong bài báo này, chúng tôi xây dựng thuật toán mới về huấn luyện mạng dựa vào phương pháp Conjugate Gradient, trong đó mục tiêu đặt ra là cải thiện tốc độ hội tụ của quá trình huấn luyện ANN. Bố cục của bài viết gồm năm phần. Phần I, đặt vấn đề, nêu tóm tắt về hướng nghiên cứu. Phần II nêu nội dung phương pháp Conjugate Gradient và một số nghiên cứu điển hình. Phần III là phần chính của bài báo, trình bày cơ sở toán học của vấn đề được đề cập và một thuật toán mới về huấn luyện ANN được xây dựng trên cơ sở toán học nêu trên. Phần IV, thí nghiệm kiểm chứng. Trong phần này chúng tôi xây dựng mạng neuron 5-5-1, sử dụng hàm truyền Logsig(n)-Purelin(n). Viết các chương trình huấn luyện mạng neuron trên cơ sở thuật toán mới (TT\*) và thuật toán trong [1], [2], [3] (gọi tắt là thuật toán [1]) bằng Matlab 7.1 để chạy mô phỏng, so sánh và kiểm chứng. Phần V, kết luận.

## II. PHƯƠNG PHÁP CONJUGATE GRADIENT

Trong mục này, trình bày về nội dung của phương pháp Conjugate Gradient đã được nêu lên trong [1], [2], đồng thời để cụ thể hoá nội dung, chúng tôi nêu lên một thuật toán Conjugate Gradient [3] được ứng dụng trong tính toán của Matlab 7.1.

Hàm sai số (2) theo chuẩn  $L_2$ :

$$E_r(W) = \frac{1}{P} \sum (e_1^2 + e_2^2 + \dots + e_p^2) \quad (3)$$

Khai triển Taylor hàm (3)

$$E(W) \approx E(W_n) + \nabla E^T|_{W=W_n} (W - W_n) + \frac{1}{2!} (W - W_n)^T \nabla^2 E|_{W=W_n} (W - W_n) \quad (4)$$

- **Bước dịch chuyển tối ưu:** Giả sử hướng dịch chuyển được chọn tại bước lặp thứ  $n$  là  $p_n$  với bước dịch chuyển là tối ưu  $\alpha_n$ , như vậy điểm trọng số thứ  $n+1$  sẽ là:

$$W_{n+1} = W_n + \alpha_n \cdot p_n \quad (5)$$

[1] đã chứng minh:

$$\frac{\partial}{\partial \alpha_n} E(W)|_{W=W_{n+1}} = 0, \text{ và} \quad (6)$$

$$g_{n+1}^T \cdot p_n = 0, \text{ với } g_{n+1} = \nabla E|_{W=W_{n+1}} \quad (7)$$

và bước tối ưu được xác định [2], [3]:

$$\alpha_n = -\frac{g_n^T p_n}{p_n^T A_n p_n} \quad (8)$$

trong đó,  $g_n = \nabla E|_{W=W_n}$  và  $A = \nabla^2 E|_{W=W_n}$

- **Hướng dịch chuyển  $p_n$ :** Hướng dịch chuyển tại bước  $n$  [1], [2], [3]:

$$p_{n+1} = -g_{n+1} + \beta_{n+1} \cdot p_n \quad (9)$$

Có một số hàm liên kết khác nhau:

Fletcher-Reeves (10), Polak-Ribiere (11) và Powell-BealeResatarts (12).

$$\beta_{n+1} = \frac{g_{n+1}^T \cdot g_{n+1}}{g_n^T \cdot g_n} \quad (10)$$

$$\beta_{n+1} = \frac{g_{n+1}^T \cdot (g_{n+1} - g_n)}{g_n^T \cdot g_n} \quad (11)$$

$$\beta_{n+1} = \frac{g_{n+1}^T \cdot (g_{n+1} - g_n)}{p_n^T \cdot (g_{n+1} - g_n)} \quad (12)$$

Cơ sở lý thuyết trình được bày trên có thể thuật toán hoá như sau [3]:

- Bước 1: Cho điểm xuất phát  $W_0$ , hướng dịch chuyển đầu tiên là hướng âm (-) của gradient tại  $W_0$ ,  $p_0 = -g_0$

- Bước 2: Chọn hệ số học tối ưu:

$$\alpha_n = -\frac{\nabla E(W)^T|_{W=W_n} \cdot p_n}{p_n^T \cdot \nabla^2 E(W)|_{W=W_n} \cdot p_n} \quad (13)$$

$$\alpha_n = -\frac{g_n^T p_n}{p_n^T A_n p_n}$$

- Bước 3: Chọn hướng dịch chuyển tiếp theo

$$p_{n+1} = -g_{n+1} + \beta_{n+1} \cdot p_n$$

- Bước 4: Điểm trọng số mới

$$W_{n+1} = W_n + \alpha_n \cdot p_n$$

Nếu thuật toán chưa hội tụ, quay lại bước 2.

### III. THUẬT TOÁN MỚI

#### 1. Cơ sở toán học của thuật toán

##### 1.1 Bổ đề

Để trình bày rõ các phần sau, tác giả bài báo xin phát biểu và chứng minh bổ đề sau:

**Phát biểu:** Nếu hàm nhiều biến  $F(X)$  bất kỳ có hàm khai triển Taylor dạng đúng và gần đúng tới đạo hàm bậc hai tại các điểm lân cận của  $X_0$  như sau:

- Dạng đúng:

$$F_r(X) = F(X_0) + \nabla E^T|_{X=X_0} (X - X_0) + \frac{1}{2!} (X - X_0)^T \nabla^2 F|_{X=X_0} (X - X_0) + R_2 \quad (14)$$

- Dạng gần đúng:

$$F(X) = F(X_0) + \nabla E^T|_{X=X_0} (X - X_0) + \frac{1}{2!} (X - X_0)^T \nabla^2 F|_{X=X_0} (X - X_0) \quad (15)$$

trong đó,

$$\mathbf{R}_2 = \frac{1}{3!} (\mathbf{X} - \mathbf{X}_0)^T \nabla [(\mathbf{X} - \mathbf{X}_0)^T \cdot \nabla^2 \mathbf{E}(\xi)] \cdot (\mathbf{X} - \mathbf{X}_0)$$

và  $\xi$  là một điểm nằm trong khoảng  $\mathbf{X}$  và  $\mathbf{X}_0$  thì các điểm cực trị của hai hàm  $F(\mathbf{X})$  và  $F_r(\mathbf{X})$  không trùng nhau.

**Chứng minh:** Từ (15) ta có:

$$\begin{aligned} \nabla F &= \nabla[\nabla F^T(\mathbf{X}_0)(\mathbf{X} - \mathbf{X}_0)] + \\ &\quad \frac{1}{2} \cdot \nabla[(\mathbf{X} - \mathbf{X}_0)^T \nabla^2 F(\mathbf{X}_0)(\mathbf{X} - \mathbf{X}_0)] \\ &= \nabla F|_{\mathbf{X}=\mathbf{X}_0} + \nabla^2 F|_{\mathbf{X}=\mathbf{X}_0} \cdot (\mathbf{X} - \mathbf{X}_0) \end{aligned}$$

ở đây:

$$\begin{aligned} \mathbf{X}^T &= [x_1 \ x_2 \ \dots \ x_k]; \quad \mathbf{X}_0^T = [x_{01} \ x_{02} \ \dots \ x_{0k}] \\ \nabla F^T(\mathbf{X}_0) &= [\partial F / \partial x_1, \partial F / \partial x_2, \dots, \partial F / \partial x_k] \Big|_{\mathbf{X}=\mathbf{X}_0} \\ \nabla^2 F|_{\mathbf{X}=\mathbf{X}_0} &= \begin{bmatrix} \frac{\partial^2 F}{\partial x_1^2} & \frac{\partial^2 F}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 F}{\partial x_1 \partial x_k} \\ \frac{\partial^2 F}{\partial x_2 \partial x_1} & \frac{\partial^2 F}{\partial x_2^2} & \dots & \frac{\partial^2 F}{\partial x_2 \partial x_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial x_k \partial x_1} & \frac{\partial^2 F}{\partial x_k \partial x_2} & \dots & \frac{\partial^2 F}{\partial x_k^2} \end{bmatrix} \Big|_{\mathbf{X}=\mathbf{X}_0} \end{aligned}$$

Để  $F(\mathbf{X})$  đạt cực trị tại  $\mathbf{X}$  ta phải có  $\nabla F = 0$

$$\begin{aligned} \Leftrightarrow \nabla F|_{\mathbf{X}=\mathbf{X}_0} + \nabla^2 F|_{\mathbf{X}=\mathbf{X}_0} \cdot (\mathbf{X} - \mathbf{X}_0) &= 0 \\ \Leftrightarrow \mathbf{X} = \mathbf{X}_0 - (\nabla^2 F|_{\mathbf{X}=\mathbf{X}_0})^{-1} \nabla F|_{\mathbf{X}=\mathbf{X}_0} \end{aligned} \quad (16)$$

$\mathbf{X}$  là điểm cực trị của  $F(\mathbf{X})$  trong (15)

Tương tự:

$$\begin{aligned} \nabla F_r &= \nabla[\nabla F_r^T(\mathbf{X}_0)(\mathbf{X} - \mathbf{X}_0)] + \\ &\quad \frac{1}{2} \cdot \nabla[(\mathbf{X} - \mathbf{X}_0)^T \nabla^2 F_r(\mathbf{X}_0)(\mathbf{X} - \mathbf{X}_0)] + \nabla R_2 \\ &= \nabla F|_{\mathbf{X}=\mathbf{X}_0} + \nabla^2 F|_{\mathbf{X}=\mathbf{X}_0} \cdot (\mathbf{X} - \mathbf{X}_0) + \nabla R_2 \end{aligned} \quad (17)$$

Thay  $\mathbf{X}$  từ (16) vào (17) ta có:

$$\nabla F_r = \nabla F|_{\mathbf{X}=\mathbf{X}_0} +$$

$$\begin{aligned} &\nabla^2 F|_{\mathbf{X}=\mathbf{X}_0} \cdot (-\nabla^2 F|_{\mathbf{X}=\mathbf{X}_0})^{-1} \nabla F|_{\mathbf{X}=\mathbf{X}_0} + \nabla R_2 \\ &= \nabla R_2|_{\mathbf{X}=\mathbf{X}_0 - (\nabla^2 F|_{\mathbf{X}=\mathbf{X}_0})^{-1} \nabla F|_{\mathbf{X}=\mathbf{X}_0}} \neq 0 \end{aligned}$$

nghĩa là điểm cực trị  $\mathbf{X}$  ở (16) của phương trình (15) không phải là điểm cực trị của hàm  $F_r(\mathbf{X})$  ở (14): điều phải chứng minh.

## 1.2 Nhận xét

Trong thuật toán Conjugate Gradient của [1],[2],[3] đã trình bày ở mục II chúng ta thấy rằng: Bước dịch chuyển  $\alpha_n$  ở (13) là nghiệm của phương trình (6), trong đó  $E(W)$  là hàm gần đúng tới đạo hàm bậc 2 của khai triển Taylor hàm sai số gốc  $E_r(W)$  ở (3). Vì vậy, nếu theo bước  $\alpha_n$  trên hướng  $p_n$  chúng ta chỉ mới nhận được giá trị cực tiểu của hàm gần đúng  $E(W)$ . Mặt khác, dựa vào *bổ đề 1.1* ta thấy điểm cực trị hàm khai triển  $E(W)$  không phải là điểm cực trị của hàm gốc  $E_r(W)$ , nghĩa là  $\alpha_n$  tính theo [1],[2],[3] chưa phải là bước tối ưu, theo đó chúng ta nhận được cực tiểu của hàm sai số gốc  $E_r(W)$  ở (3). Thực tế, bước dịch chuyển tối ưu:

$$\alpha_{nop} = \alpha_n + \Delta \alpha_n \quad (18)$$

trong đó  $\Delta \alpha_n$  là số gia và  $\alpha_{nop} = \alpha_n + \Delta \alpha_n$  phải được xác định từ hàm gốc  $E_r(W)$  hoặc từ hàm khai triển dạng (14) của giá trị sai lệch.

Sự sai lệch của bước dịch chuyển là yếu tố làm chậm tốc độ hội tụ của thuật toán [1], [2], [3]. Trong bài báo này chúng tôi trình bày phương pháp gần đúng xác định  $\Delta \alpha_n$  làm cơ sở để xây dựng thuật toán mới có tốc độ hội tụ cao hơn.

## 1.3 Xác định $\Delta \alpha_n$

Giả sử hàm sai số đã xác định ở  $W_n$  có  $g_n, p_n$ , chúng ta cần tìm điểm cực trị  $W_{n+1}$  theo hướng  $p_n$ . Quá trình phải thực hiện theo hai bước:

- Từ  $W_n$  xác định điểm  $W_n^*$  bằng cách dịch chuyển theo hướng  $p_n$ , bước dịch chuyển  $\alpha_n$

$$\alpha_n = -\frac{\mathbf{g}_n^T \mathbf{p}_n}{\mathbf{p}_n^T \mathbf{A}_n \mathbf{p}_n}$$

- Tiếp tục dịch chuyển theo hướng  $\mathbf{p}_n$ , bước dịch chuyển  $\Delta\alpha_n$  được xác định từ (19).

$$\frac{\partial}{\partial \Delta\alpha_n} E(W)|_{W=W_n^*+\Delta\alpha_n \mathbf{p}_n} = 0 \quad (19)$$

$$\Delta\alpha_n = -\frac{\nabla E(W)^T|_{W=W_n^*} \cdot \mathbf{p}_n}{\mathbf{p}_n^T \cdot \nabla^2 E(W)|_{W=W_n^*} \cdot \mathbf{p}_n} = -\frac{\mathbf{g}_n^{*T} \cdot \mathbf{p}_n}{\mathbf{p}_n^T \cdot \mathbf{A}_n^* \cdot \mathbf{p}_n} \quad (20)$$

là số gia của bước dịch chuyển theo hướng  $\mathbf{p}_n$ , theo đó chúng ta tiệm cận gần hơn tới điểm cực trị của hàm gốc  $E_r(W)$  theo hướng  $\mathbf{p}_n$ .

Như vậy, bước dịch chuyển tối ưu tại  $W_n$  là:

$$\begin{aligned} \alpha_{nop} &= \alpha_n + \Delta\alpha_n \\ \alpha_{nop} &= -\frac{\mathbf{g}_n^T \mathbf{p}_n}{\mathbf{g}_n^T \mathbf{A}_n \mathbf{p}_n} - \frac{\mathbf{g}_n^{*T} \cdot \mathbf{p}_n}{\mathbf{p}_n^T \cdot \mathbf{A}_n^* \cdot \mathbf{p}_n} \end{aligned} \quad (21)$$

Từ đó, chúng ta có thuật toán xác định trọng số như sau:

## 2. Thuật toán TT\*

### - Bước 1:

Cho điểm xuất phát  $W_0$ , hướng dịch chuyển đầu tiên là hướng âm (-) của gradient tại  $W_0$ ,

$$\mathbf{p}_0 = -\mathbf{g}_0$$

### - Bước 2:

Tại điểm  $W_n$ , tính  $\mathbf{g}_n$ ,  $\beta_n$ ,  $\mathbf{A}_n$ . Xác định hướng dịch chuyển  $\mathbf{p}_n$ :

$$\mathbf{p}_n = -\mathbf{g}_n + \beta_n \cdot \mathbf{p}_{n-1}$$

### - Bước 3:

Xác định bước dịch chuyển:

$$\alpha_n = -\frac{\nabla E(W)^T|_{W=W_n} \cdot \mathbf{p}_n}{\mathbf{p}_n^T \cdot \nabla^2 E(W)|_{W=W_n} \cdot \mathbf{p}_n} = -\frac{\mathbf{g}_n^T \cdot \mathbf{p}_n}{\mathbf{p}_n^T \cdot \mathbf{A}_n \cdot \mathbf{p}_n}$$

Cực tiểu hóa theo hướng  $\mathbf{p}_n$  nhằm xác định điểm trung gian  $W_n^*$

$$W_n^* = W_n + \alpha_n \mathbf{p}_n$$

### - Bước 4:

Tại điểm  $W_n^*$ , tính  $\mathbf{g}_n^*$ ,  $\mathbf{A}_n^*$ . Xác định số gia  $\Delta\alpha_n$  của bước dịch chuyển:

$$\Delta\alpha_n = -\frac{\nabla E(W)^T|_{W=W_n^*} \cdot \mathbf{p}_n}{\mathbf{p}_n^T \cdot \nabla^2 E(W)|_{W=W_n^*} \cdot \mathbf{p}_n} = -\frac{\mathbf{g}_n^{*T} \cdot \mathbf{p}_n}{\mathbf{p}_n^T \cdot \mathbf{A}_n^* \cdot \mathbf{p}_n}$$

Cực tiểu hóa theo hướng  $\mathbf{p}_n$  nhằm xác định điểm  $W_{n+1}$

$$W_{n+1} = W_n^* + \Delta\alpha_n \mathbf{p}_n$$

Nếu thuật toán chưa hội tụ, quay lại bước 2.

## IV. THÍ NGHIỆM KIỂM CHỨNG

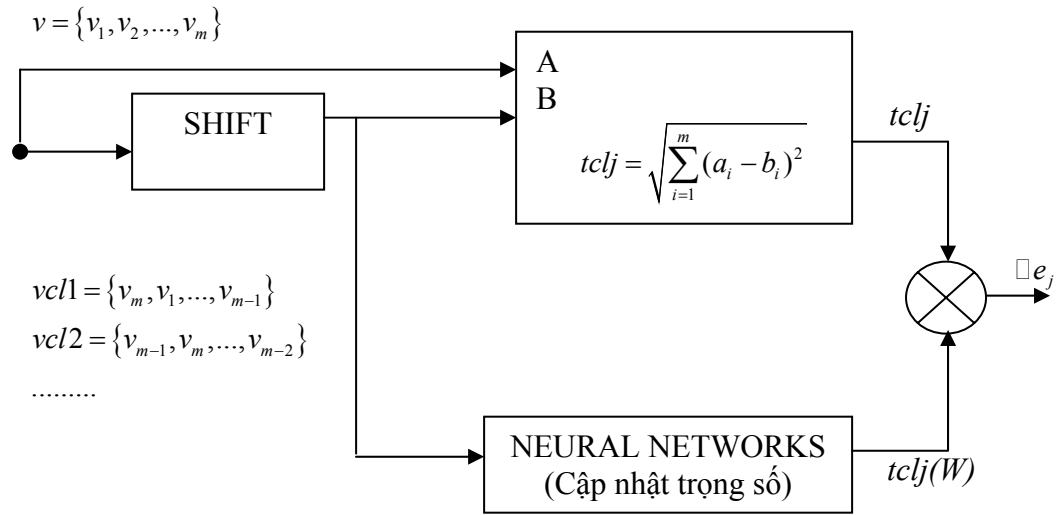
Trong phần này, chúng tôi sử dụng hai thuật toán TT\* và [1] để huấn luyện mạng mạng 5-5-1 ANN (hình 3) nhận dạng vector đặc trưng của ảnh theo hàm sai số tự tương quan (hình 2): một vấn đề đã được nghiên cứu trong nhận dạng ảnh [12], liên quan mật thiết với lĩnh vực kỹ thuật robot.

Trong thí nghiệm, vector đặc trưng của ảnh và bốn vector vclj như sau:

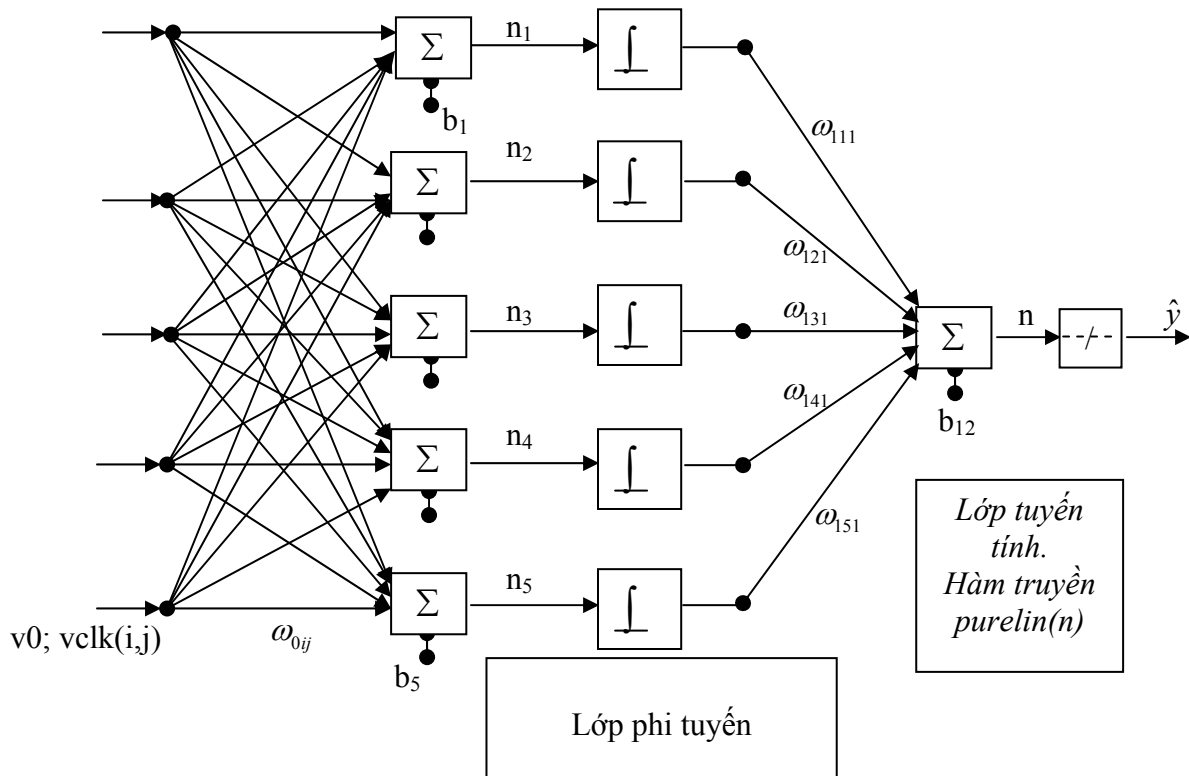
$$\begin{aligned} \mathbf{v} &= \{1, 2, 1, 0, 1\}; \\ \mathbf{vcl1} &= \{1, 1, 2, 1, 0\}; \\ \mathbf{vcl2} &= \{0, 1, 1, 2, 1\}; \\ \mathbf{vcl3} &= \{1, 0, 1, 1, 2\}; \\ \mathbf{vcl4} &= \{2, 1, 0, 1, 1\} \end{aligned}$$

Khảo sát hàm sai số của mạng  $E(w)$  ứng với tất cả các mẫu huấn luyện và tín hiệu ra của mạng ứng với từng mẫu huấn luyện ( $\mathbf{v}, \mathbf{vcl1}, \mathbf{vcl2}, \mathbf{vcl3}, \mathbf{vcl4}$ ),  $j=0, 1, \dots, 4$ .

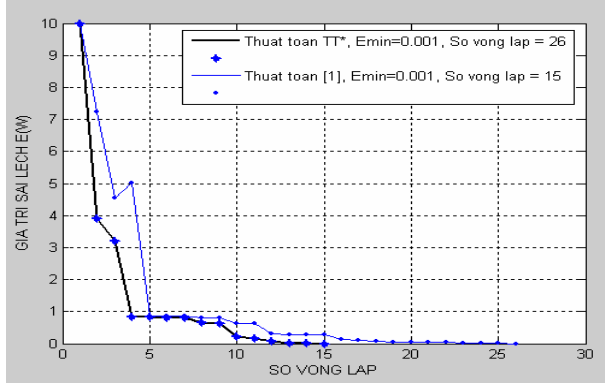
Kết quả cho thấy tốc độ hội tụ (giá trị trung bình của số vòng lặp) theo TT\* nhanh hơn [1], đồng thời tín hiệu ra của mạng theo TT\* ổn định hơn theo [1]. (từ hình 4 đến hình 9)



Hình 2. Huấn luyện mạng neuron- dùng hàm sai số tự tương quan- để nhận dạng vector đặc trưng của ảnh.

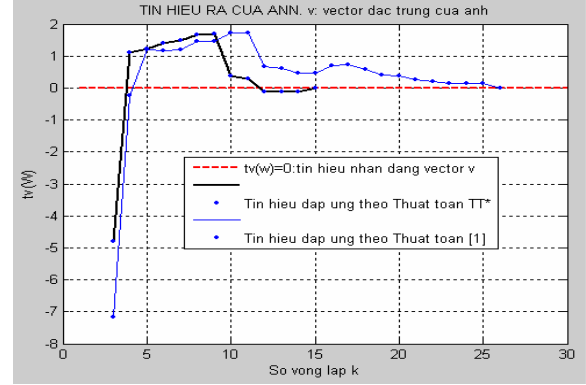


Hình 3. Mạng 5-5-1 sử dụng trong thí nghiệm kiểm chứng

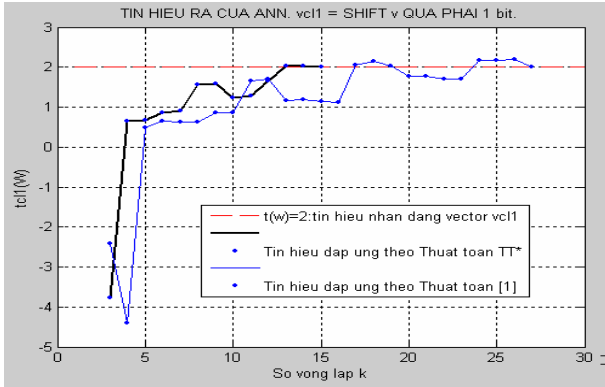


Hình 4. Tín hiệu ra của mạng, hàm sai lệch

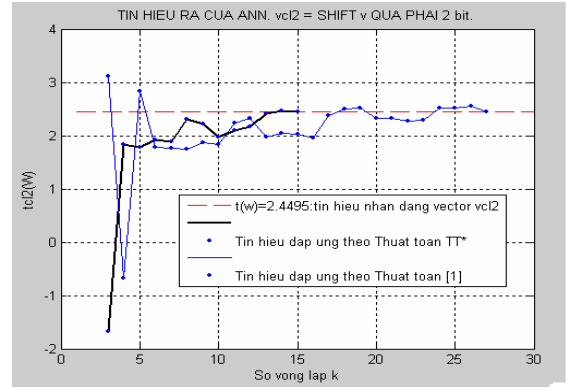
$$E(W) = \frac{1}{5} \sum_{j=1}^5 \Delta e_j^2 \text{ ứng với 2 thuật toán}$$



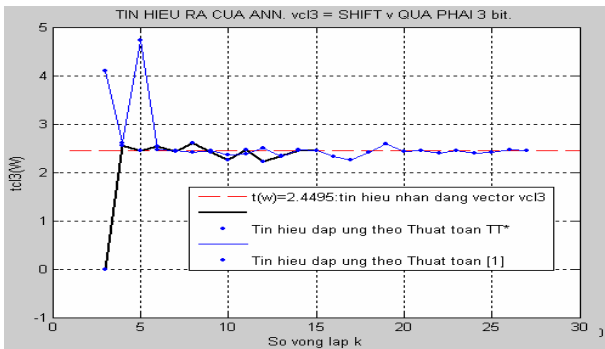
Hình 5. Tín hiệu ra của mạng, hàm sai lệch  $tv(W)$  theo 2 thuật toán. Mẫu huấn luyện vào-ra  $P_0 = \{v, vcl1, 2, 3, 4 - tcl0 = 0\}$



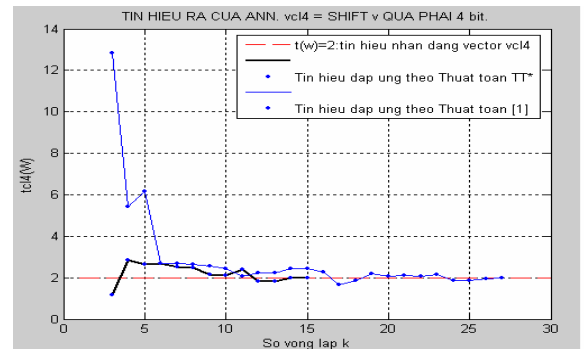
Hình 6. Tín hiệu ra của mạng, hàm sai lệch  $tcl1(W)$  theo 2 thuật toán. Mẫu huấn luyện vào-ra  $P_1 = \{v, vcl1, 2, 3, 4 - tcl1 = 2\}$



Hình 7. Tín hiệu ra của mạng, hàm sai lệch  $tcl2(W)$  theo 2 thuật toán. Mẫu huấn luyện vào-ra  $P_2 = \{v, vcl1, 2, 3, 4 - tcl2 = 2.4495\}$



Hình 8. Tín hiệu ra của mạng, hàm sai lệch  $tv(W)$  theo 2 thuật toán. Mẫu huấn luyện vào-ra  $P_3 = \{v, vcl1, 2, 3, 4 - tcl3 = 2.4495\}$



Hình 9. Tín hiệu ra của mạng, hàm sai lệch  $tv(W)$  theo 2 thuật toán. Mẫu huấn luyện vào-ra  $P_4 = \{v, vcl1, 2, 3, 4 - tcl4 = 2\}$

## V. KẾT LUẬN

Trong nghiên cứu này, chúng tôi đã xây dựng một thuật toán mới (TT\*) về huấn luyện ANN có số vòng lặp trung bình nhỏ hơn thuật toán Conjugate Gradient trong [1],[2],[3]. Điều này có ý nghĩa trong việc huấn luyện mạng neuron trực tuyến Online trong nhiều ứng dụng về nhận dạng và điều khiển trong môi trường động.

Về mặt toán học chúng ta thấy tính chặt chẽ của thuật toán TT\* thể hiện ở thành phần, ý nghĩa và vai trò của  $\Delta\alpha_n$  trong thuật toán:

$$\Delta\alpha_n = -\frac{\nabla E(W)^T|_{W=W_n^*} \cdot P_n}{P_n^T \cdot \nabla^2 E(W)|_{W=W_n^*} \cdot P_n} = -\frac{g_n^{*T} \cdot P_n}{P_n^T \cdot A_n^* \cdot P_n}$$

- Nếu  $\alpha_n$  là bước tối ưu: khi đó  $W_n^*$  là điểm cực trị của  $Er(W)$  (hình 2) và do đó từ (6), (7) (đã được chứng minh trong [1],[2]) suy ra  $g_n^{*T} \cdot P_n = 0$ , nghĩa là  $\Delta\alpha_n = 0$  và  $\alpha_{nop} = \alpha_n$ .

- Nếu  $\alpha_n$  không phải là bước tối ưu: Lý luận tương tự như trên, suy ra  $\Delta\alpha_n \neq 0$  và do đó  $\alpha_{nop} = \alpha_n + \Delta\alpha_n$  là bước tối ưu hướng quá trình khảo sát tới tiệm cận với điểm cực trị theo hướng  $p_n$  ở vòng lặp thứ n.

Thực tế, trong huấn luyện mạng 1-5-1 (A=16x16) và 5-5-1 (A=36x36) chúng tôi nhận thấy những vòng lặp đầu tiên thường  $\Delta\alpha_n \neq 0$ . Sự tham gia của  $\Delta\alpha_n$  đã làm tăng tốc độ hội tụ của thuật toán.

Xét về số vòng lặp trung bình trong huấn luyện mạng, TT\* có giá trị nhỏ hơn đáng kể so với [1] tuy nhiên xét về thời gian huấn luyện, sử dụng thuật toán TT\* chỉ nhanh hơn [1] khi mạng có ma trận trọng số không lớn. Nếu mạng có ma trận trọng số lớn, TT\* không nhanh hơn [1] mặc dù số vòng lặp ít hơn. Đây là vấn đề đặt ra cho nghiên cứu tiếp theo của chúng tôi nhằm tăng phạm vi ứng dụng của TT\*.

## TÀI LIỆU THAM KHẢO

1. Syed Muhammad Aqil Burney, Tahseen Ahmed Jilani, Cemal Ardil. 2004. *A comparison of First and Second Order Training Algorithms for Artificial Neural Networks*. International Journal of Computational Intelligence. Volume 1 number 3, 2004 ISSN: 1304-4508.
2. Ajith Abraham. 2003. *Meta Learning Evolutionary Artificial Neural Networks*. [www.ElsevierComputerScience.com](http://www.ElsevierComputerScience.com) Powered by Science @ Director. Neurocomputing 56 (2004) 1-38. [www.Elsevier.com/locate/neucom](http://www.Elsevier.com/locate/neucom).
3. Prof. Matin Hagan of Oklahoma State University, Demuth, Beale. *Neural Networks Design*. ISBN 0-9717321-0-8. <http://www.ee.Okstate.edu/mhagan/nnd.html>; <http://www.cubooks.colorado.edu>.
4. Bogdan M. Wilamowski, M. Önder Efe. *An Algorithm for Fast Convergence in Training Neural Networks*. 2001. IEEE.
5. Deniz Erdogmus, Jose C.Principe. *Generalized Information Potential Criterion for Adaptive System Training*. 2002. IEEE.
6. Nikolaos Ampazis, Stavros J. Perantonis. *Two Highly Efficient Second-Order Algorithms for Training Feedforward Networks*. 2002.IEEE.
9. Segio Lima Netto and Panajotis Agatholis. *Efficient Lattice Realizations of Adaptive IIR Algorithms*. IEEE 1998.
10. Ryan Mukai, Victor A. Vilnrotter. *Adaptive Acquisition and Tracking for Deep Space Array Feed Antennas*. 2002 IEEE. VOL 13.NO.5. September 2002.
11. *Neural Networks for Robotic Control. Theory and Application*. A.M.S. Zalzal and A.S. Morris. Britain 1996.
12. Nguyễn Đức Minh. *Điều khiển Robot Scorbot dùng thị giác máy tính*. Luận văn Thạc sĩ. Mã số ĐKKT-K13-009. Đại học Bách khoa TP. HCM. 2004.