

# TỔNG HỢP HỆ THỐNG THỐNG SUY DIỄN NEURO-FUZZY THÍCH NGHI (ANFIS) TỪ TẬP DỮ LIỆU SỐ

NGUYỄN SỸ DŨNG<sup>1</sup>, NGÔ KIỀU NHI<sup>2</sup>

<sup>1</sup>Trường Đại học Công nghiệp thành phố Hồ Chí Minh

<sup>2</sup>Trường Đại học Bách khoa thành phố Hồ Chí Minh

**Abstract.** Accuracy of approximation an unknown function from a numerical data set based on neuro-fuzzy inference systems depends on efficiency of data space partition to build fuzzy sets. In this paper we propose a solution to data space partition to build optimal fuzzy sets. The focus of interest in proposed approach is to increase accuracy of approximation by adjusting degree of fit between the given numerical data set and the established fuzzy sets, which are used to approximate it. The solution is based on two new algorithms. The first, named CBMM, is used for building Min-Max clusters, and the another one, named CSHL, is used for direction for the partition of Min-Max clusters into a set of pure Min-Max clusters in input space. Result of the use of these algorithms is a optimal fuzzy set in input space, which is used to build a new algorithm, named HLM, establishing adaptive neuro-fuzzy inference systems. A series of numerical experiments are performed to assess the efficiency of the proposed approach.

**Tóm tắt.** Độ chính xác của hệ thống suy diễn neuro-fuzzy xấp xỉ hàm chưa biết diễn tả mối quan hệ giữa tín hiệu vào và tín hiệu ra của đối tượng thông qua một tập dữ liệu số phụ thuộc vào mức độ phù hợp giữa các tập mờ được thiết lập và đặc thù của không gian dữ liệu. Trong bài báo này chúng tôi trình bày một giải pháp về phân chia không gian dữ liệu để xây dựng các tập mờ tối ưu với tiêu chí đặt ra là cải thiện độ chính xác của phép xấp xỉ. Giải pháp được thực hiện dựa trên hai thuật toán mới: thuật toán chia bó Min-Max (CBMM) và thuật toán cắt siêu hộp lai (CSHL). CBMM thực hiện hai quá trình: phân lớp dữ liệu ở không gian ra, chia bó siêu hộp Min-Max và gán nhãn cho các mẫu dữ liệu ở không gian vào trong mối quan hệ ràng buộc qua lại giữa không gian vào và không gian ra. Tiếp theo, thuật toán CSHL tiếp tục phân chia các bó siêu hộp Min-Max thành một tập các bó siêu hộp thuần chủng đóng vai trò là khung sườn để xây dựng các tập mờ tối ưu ở không gian dữ liệu vào. Trên cơ sở tập mờ tối ưu đã được xác lập, một thuật toán mới về tổng hợp mạng neuro-fuzzy, thuật toán huấn luyện mạng (HLM), được thiết lập để xây dựng mạng neuro-fuzzy từ một tập dữ liệu số cho trước. Bài báo trình bày nhiều thí nghiệm kiểm chứng trên những tập dữ liệu khác nhau để so sánh hiệu quả của các thuật toán mới so với một số thuật toán đã công bố.

## 1. ĐẶT VẤN ĐỀ

Hệ thống suy diễn neuro-fuzzy đã được nghiên cứu và ứng dụng có hiệu quả trong nhiều lĩnh vực khác nhau như đo lường, điều khiển, nhận dạng và dự báo... Thông thường, các ứng dụng trên liên quan tới việc xấp xỉ hàm chưa biết  $y = f(\bar{x})$  phản ánh quan hệ vào-ra

của một tập  $T_\Sigma$  gồm  $P$  cặp dữ liệu số  $(\bar{x}_i, y_i)$ ,  $\bar{x}_i = [x_{i1}x_{i2}\dots x_{in}]$ ,  $i = 1\dots P$ . Việc xác định hàm  $f$  thông qua  $T_\Sigma$  có thể được thực hiện theo nhiều phương pháp khác nhau. Một trong những phương pháp thông dụng là sử dụng mô hình suy diễn mờ MI-SO [15]. Theo phương pháp này, không gian dữ liệu được phân chia thành các bó dữ liệu làm cơ sở để xây dựng các tập mờ. Có nhiều phương pháp chia bó dữ liệu. Nếu dựa vào miền không gian dữ liệu được phân chia ta có thể phân ra: chia bó ở không gian dữ liệu vào [1, 6, 10, 11], chia bó ở không gian dữ liệu ra [9], chia bó trong không gian liên kết vào ra [14]. Nếu dựa vào phương pháp phân chia dữ liệu ta có thể phân làm hai phương pháp chính, đó là phương pháp chia bó cứng (Hard C-Mean, HCM) [4] và phương pháp chia bó mờ (Fuzzy C-Mean, FCM) [2, 11]. Gần đây, một nghiên cứu phát triển phương pháp chia bó của [11] được trình bày trong [7] và [8]. Về cơ bản, đây là phương pháp FCM. Quá trình chia bó nhằm xây dựng các bó dạng siêu hộp min-max trong không gian dữ liệu vào. Tuy nhiên quá trình phân chia của [7, 8] được thực hiện trong mối liên hệ ràng buộc qua lại giữa không gian vào và không gian ra. Theo phương pháp này,  $T_\Sigma$  được chia thành nhiều lớp nhãn. Mỗi lớp nhãn được định biên theo một siêu phẳng ở không gian dữ liệu ra. Tất cả các mẫu dữ liệu trong một lớp nhãn đều mang cùng một nhãn là nhãn của lớp. Tập mẫu được gán nhãn  $T_\Sigma$ , gọi tắt là tập mẫu nhãn, là cơ sở để xây dựng một tập các bó thuần chủng  $pHB$  ở không gian dữ liệu vào, trong đó mỗi  $pHB$  là một siêu hộp chiếm một miền trong không gian dữ liệu  $\mathcal{R}^n$  được giới hạn bởi hai điểm cực trị - điểm min và điểm max. Hàm liên thuộc của từng bó mờ được xây dựng dựa vào các điểm cực trị này. Mỗi tập mờ được xác lập dựa vào các giá trị min, max và hàm liên thuộc của siêu hộp tương ứng.

Độ chính xác của ANFIS phụ thuộc vào mức độ phù hợp giữa các tập mờ được thiết lập và đặc thù của không gian dữ liệu. Việc thực hiện chia bó trong mối liên hệ ràng buộc qua lại giữa không gian vào và không gian ra của [7, 8] đã gia tăng độ chính xác của phép xấp xỉ hàm  $f$  vì các tập mờ được tạo ra phản ánh tốt hơn quan hệ hai chiều trong không gian dữ liệu. Trong bài báo này chúng tôi trình bày một phát triển tiếp theo của [7, 8] về phương pháp tổng hợp ANFIS dựa trên tập dữ liệu số cho trước, được trình bày trong thuật toán mới mang tên thuật toán huấn luyện mạng (HLM), trong đó chúng tôi bổ sung thêm điều kiện dừng theo không gian vào (input space) trong quá trình phân chia không gian dữ liệu để xác lập các bó siêu hộp min-max, được trình bày trong thuật toán CBMM; đồng thời sử dụng giải pháp định hướng tối ưu cho quá trình phân chia không gian dữ liệu để xây dựng các tập mờ dựa trên tập các siêu hộp thuần chủng ở không gian dữ liệu vào, được trình bày trong thuật toán CSHL.

## 2. MỘT SỐ KHÁI NIỆM

Tập mẫu huấn luyện  $T_\Sigma$  gồm  $P$  cặp dữ liệu số  $(\bar{x}_i, y_i)$ ,  $\bar{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]$ ,  $i = 1\dots P$ , tạo ra một trường không gian dữ liệu  $n$  chiều  $\mathcal{R}^n$  ở không gian dữ liệu vào.

**Bó siêu phẳng** (hyperplane). Trong không gian dữ liệu  $\mathcal{R}^n$ , phương trình của một siêu phẳng có dạng  $y = \sum_{j=1}^n a_j(\cdot) + a_0$ , trong đó  $a_j$ ,  $j = 0\dots n$  là các hệ số.

Ta nói rằng, điểm mẫu  $(\bar{x}_i, y_i)$  nằm trong vùng lân cận  $[\varepsilon]$  của siêu phẳng thứ  $k$  nếu:

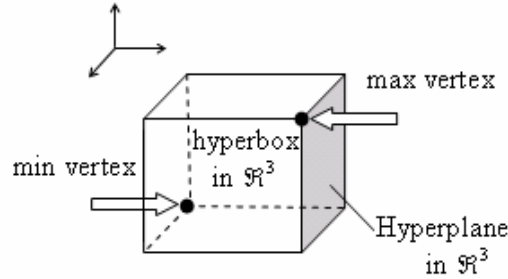
$$\left| y_i - \sum_{j=1}^n (a_j^{(k)} x_{ij} + a_0^{(k)}) \right| \leq [\varepsilon]. \quad (1)$$

Tập các điểm mẫu  $(\bar{x}_i, y_i) \in T_\Sigma$  thỏa bất đẳng thức (1) tạo ra một bó dữ liệu nằm trong vùng lân cận  $[\varepsilon]$  của siêu phẳng thứ  $k$  được gọi là bó siêu phẳng thứ  $k$ , và  $k$  được gọi là nhãn của bó.

**Nhãn của mẫu dữ liệu.** Nếu mẫu  $\bar{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]$  thuộc về bó siêu phẳng nhãn  $k$  thì ta nói rằng nhãn của  $\bar{x}_i$  là  $k$ , nghĩa là nhãn của một mẫu dữ liệu chính là nhãn của bó siêu phẳng chứa mẫu đó.

**Siêu hộp (hyperbox HB).** Trong trường không gian dữ liệu  $\mathfrak{R}^n$ , siêu hộp  $HB$  là vùng không gian được giới hạn bởi các mặt là các siêu phẳng, trong đó mỗi siêu phẳng song song với một mặt phẳng tọa độ và đi qua một trong hai đỉnh cực trị min, max.

Hình 1 thể hiện một siêu hộp hyperbox trong không gian 3 chiều ( $\mathfrak{R}^3$ ).



Hình 1. Siêu hộp trong  $\mathfrak{R}^3$

Siêu hộp thứ  $t$ , ký hiệu  $HB_t$ , có  $T_t$  là tập hợp của các mẫu dữ liệu thuộc nó.

**Đỉnh cực trị min, max (min-max vertexes).** Mỗi siêu hộp  $HB_t$  được đặc trưng bởi hai đỉnh cực trị - đỉnh max,  $\bar{\omega}_t$ , và đỉnh min,  $\bar{v}_t$ , như sau:

$$\bar{\omega}_t = [\omega_{t1}\omega_{t2}\dots\omega_{tn}], \quad \bar{v}_t = [v_{t1}v_{t2}\dots v_{tn}], \quad (2)$$

trong đó,  $\omega_{tj} = \max(x_{ij} | \bar{x}_i \in T_t, j = 1 \dots n)$  và  $v_{tj} = \min(x_{ij} | \bar{x}_i \in T_t, j = 1 \dots n)$ .

**Siêu hộp thuần chủng và siêu hộp lai (pure hyperbox, pHB, và hybrid hyperbox, hHB).** Siêu hộp  $HB_t$  được gọi là siêu hộp thuần chủng nhãn  $m$  (ký hiệu  $pHB_f^{(m)}$ ) nếu tập hợp  $T_t$  chứa toàn bộ các mẫu cùng nhãn  $m$ . Nếu  $T_t \neq \phi$  và không phải tập các phần tử cùng nhãn thì  $HB_t$  được gọi là siêu hộp lai (ký hiệu  $hHB_t$ ).

**Siêu hộp không phủ lên một siêu hộp khác và siêu hộp thỏa tính phủ (\*) (overlap condition).** Cho trước siêu hộp  $HB_h$ . Xét một siêu hộp  $HB_k$  bất kỳ,  $k \neq h$ . Ta nói rằng  $HB_h$  không phủ lên  $HB_k$  khi giữa chúng không có vùng không gian chung.

Để thấy rằng nếu  $\bar{\omega}_h < \bar{v}_k$  hoặc  $\bar{v}_h > \bar{\omega}_k$  thì  $HB_h$  không phủ lên  $HB_k$ . Do đó, trong nghiên cứu này chúng tôi sử dụng các bất đẳng thức này để kiểm tra tính phủ của các siêu hộp thuần chủng.

Gọi  $L_p$  và  $L_h$  theo thứ tự là tập chứa tất cả các siêu hộp thuần chủng và siêu hộp lai được tạo thành từ tập dữ liệu huấn luyện ban đầu  $T_\Sigma$ , nghĩa là  $L_p \cup L_h = T_\Sigma$ . Nếu  $HB_h$  không phủ lên bất kỳ một siêu hộp nào thuộc  $L_p$  và  $L_h$  thì ta nói rằng  $HB_h$  thỏa tính phủ.

**Siêu hộp liên kết (\*\*) (fusion hyperbox).** Cho trước hai siêu hộp cùng nhãn  $m$   $pHB_k^{(m)}$  và  $pHB_h^{(m)}$ . Một siêu hộp  $pHB_j^{(m)}$  cùng nhãn  $m$  được gọi là siêu hộp liên kết của hai siêu hộp trên nếu thỏa mãn đồng thời ba mệnh đề sau:

- (i)  $T_f = T_k \cup T_h$ ,
- (ii)  $\bar{\omega}_f = \max(\bar{\omega}_k, \bar{\omega}_h)$ ,  $\bar{v}_f = \min(\bar{v}_k, \bar{v}_h)$ ,
- (iii)  $pHB_j^{(m)}$  thỏa mãn tính phủ (\*),

trong đó,  $T_h, T_k, T_f$  theo thứ tự là các tập mẫu của  $pHB_h^{(m)}$ ,  $pHB_k^{(m)}$ ,  $pHB_f^{(m)}$ .

### 3. XÁC LẬP SIÊU PHẪNG BẰNG PHƯƠNG PHÁP BÌNH PHƯƠNG CỰC TIỂU (LMS) [12]

Cho tập mẫu huấn luyện  $T_\Sigma$  trong không gian dữ liệu  $\mathcal{R}^n$  gồm  $P$  cặp dữ liệu số. Chia  $T_\Sigma$  thành các bó dữ liệu. Bó thứ  $k$  có  $t_k$  mẫu dữ liệu:

$$(\bar{x}_i, y_i), \bar{x}_i = [x_{i1}x_{i2}\dots x_{in}], i = 1\dots t_k. \quad (3)$$

Phương pháp bình phương cực tiểu LMS thường được sử dụng để nội suy phương trình siêu phẳng cho tập dữ liệu (3). Các hệ số của siêu phẳng phải đảm bảo cho tổng bình phương giá trị sai lệch giữa dữ liệu ra thực tế  $y_i$  và dữ liệu ra mô hình  $\hat{y}_i = \sum_{j=1}^n (a_j^{(k)})x_{ij} + a_0^{(k)}$  là nhỏ nhất:

$$S = \sum_{i=1}^{t_k} [y_i - \sum_{j=1}^n (a_j^{(k)})x_{ij} + a_0^{(k)}]^2 = \min, \quad (4)$$

nghĩa là, điều kiện ràng buộc các hệ số cần tìm là  $\partial S / \partial a_j^{(k)} = 0$ ,  $j = 0\dots n$ . Từ đó ta có hệ phương trình sau:

$$\left\{ \begin{array}{l} a_n^{(k)} \sum_{i=1}^{t_k} x_{in} + a_{n-1}^{(k)} \sum_{i=1}^{t_k} x_{i(n-1)} + \dots + a_1^{(k)} \sum_{i=1}^{t_k} x_{i1} + na_0^{(k)} = \sum_{i=1}^{t_k} y_i \\ a_n^{(k)} \sum_{i=1}^{t_k} x_{in}x_{i1} + \dots + a_1^{(k)} \sum_{i=1}^{t_k} x_{i1}^2 + a_0^{(k)} \sum_{i=1}^{t_k} x_{i1} = \sum_{i=1}^{t_k} y_i x_{i1} \\ a_n^{(k)} \sum_{i=1}^{t_k} x_{in}x_{i2} + \dots + a_2^{(k)} \sum_{i=1}^{t_k} x_{i2}^2 + a_1^{(k)} \sum_{i=1}^{t_k} x_{i1}x_{i2} + a_0^{(k)} \sum_{i=1}^{t_k} x_{i2} = \sum_{i=1}^{t_k} y_i x_{i2} \\ a_n^{(k)} \sum_{i=1}^{t_k} x_{in}^2 + \dots + a_1^{(k)} \sum_{i=1}^{t_k} x_{i1}x_{in} + a_0^{(k)} \sum_{i=1}^{t_k} x_{in} = \sum_{i=1}^{t_k} y_i x_{in} \end{array} \right. \quad (1)$$

Giải ra ta có các hệ số  $a_j^{(k)}$ ,  $j = 0\dots n$  của siêu phẳng cần tìm.

## 4. XÂY DỰNG CÁC TẬP MỜ TỐI ƯU Ở KHÔNG GIAN DỮ LIỆU VÀO

### 4.1. Thuật toán chia bó min-max, CBMM

Thuật toán CBMM dùng để xây dựng các bó dữ liệu dạng siêu hộp min-max ở không gian dữ liệu vào, siêu phẳng ở không gian ra và gán nhãn cho tất cả các mẫu huấn luyện.

Thuật toán CBMM được xây dựng trên cơ sở phát triển thuật toán Hyperplane Clustering của [7] và ứng dụng thuật toán chia bố Hard C-Mean (HCM) của [4]. Trong đó, để gia tăng mức độ ràng buộc về dữ liệu giữa không gian vào và không gian ra, chúng tôi đề xuất việc bổ sung thêm điều kiện dừng của thuật toán: không chỉ thỏa mãn ràng buộc ở không gian ra như thuật toán Hyperplane Clustering của [7] (điều kiện (8)) mà còn phải thỏa mãn điều kiện ràng buộc ở không gian vào (điều kiện (6)). Sự kết hợp đồng thời cả hai điều kiện (6) và (8) nhằm đảm bảo rằng nếu một mẫu dữ liệu đã thuộc một lớp dữ liệu ở không gian ra thì cũng sẽ nằm gần tâm phân bố của bố dữ liệu tương ứng ở không gian vào. Đây là điều kiện cần thiết vì chỉ khi đó mới đảm bảo rằng tập các bố siêu hộp min-max được tạo thành sẽ phủ lên tất cả các mẫu trong tập dữ liệu  $T_\Sigma$  đã cho.

### Hai nguyên tắc khi xây dựng thuật toán CBMM

*Nguyên tắc 1:* Nếu số luật mờ  $M$  xác định thì số bố dữ liệu ở không gian vào cũng như số siêu phẳng ở không gian ra đều bằng  $M$ .

*Nguyên tắc 2:* Nếu một mẫu  $\bar{x}_i$  ở không gian vào thuộc bố thứ  $k$ ,  $\Gamma^{(k)}$ ,  $k = 1 \dots M$  thì đáp ứng vào ra của  $\bar{x}_i$  cũng được gán cho siêu phẳng cùng nhãn  $A_k$  ở không gian ra và ngược lại.

### Nội dung của thuật toán CBMM

Chọn số luật mờ  $M$ . Gọi ma trận phân bố dữ liệu [4] ở vòng lặp thứ  $r$  là  $U^{(r)}$ . Chọn  $U^{(0)}$ .

*Bước 1.* Chia bố ở không gian vào

- Nếu ở chu kỳ khởi tạo của  $CBMM(r = 1)$ : Áp dụng nguyên tắc 1 và thuật toán HCM của [4] để tìm các bố  $\Gamma^{(k)}$ ,  $k = 1 \dots M$  ở không gian vào.

- Nếu ở chu kỳ  $r > 1$ : Sử dụng nguyên tắc 2 để định biên mới cho từng bố ở không gian vào dựa vào biên chế các siêu phẳng cùng nhãn ở không gian ra. Sử dụng thuật toán HCM để sắp xếp, điều chỉnh lại biên chế từng bố. Tính và kiểm tra theo độ sai lệch cho phép  $[\theta_{in}]$  ở không gian vào:

$$kt.in_{(r)} = \sum_{i=1}^M \sum_{j=1}^P \frac{|U_{ij}^{(r)} - U_{ij}^{(r-1)}|}{P} \leq [\theta_{in}]. \quad (6)$$

*Bước 2.* Định biên mới cho các siêu phẳng ở không gian ra dựa trên kết quả của Bước 1 và Nguyên tắc 2: Cập nhật các hệ số  $a_j^{(k)}$ ,  $j = 0 \dots n$ , của siêu phẳng  $A_k$ ,  $k = 1 \dots M$ , ở không gian ra dựa vào LMS (Mục 3).

*Bước 3.* Sắp xếp lại các bố và các siêu phẳng

- Tính khoảng cách  ${}_{(r)}d_i^{(k)}$  từ điểm mẫu  $i$ ,  $i = 1 \dots P$ , tới siêu phẳng thứ  $k$ ,  $k = 1 \dots M$ , tại vòng lặp thứ  $r$ :

$${}_{(r)}d_i^{(k)} = \left| \frac{y_i - \left( \sum_{j=1}^n (a_j^{(k)} x_{ij} + a_0^{(k)}) \right)}{\sqrt{1 + \sum_{j=1}^n (a_j^{(k)})^2}} \right|. \quad (7)$$

- Từng cặp  $(\bar{x}_i, y_i)$ ,  $i = 1 \dots P$  trong tập mẫu huấn luyện được gán về các siêu phẳng thứ  $q$ ,  $A_q$ , và bố thứ  $q$ ,  $\Gamma^{(q)}$  có:

$${}_{(r)}d_i^{(q)} = \min_{(k=1 \dots M)} [{}_{(r)}d_i^{(k)}]$$

và do đó mang nhãn  $q$ .

- Tính và kiểm tra theo giá trị cho phép  $[\theta_{out}]$  ở không gian ra:

$$kt\_out_{(r)} = \frac{1}{P} \sqrt{\sum_{i=1}^P \left( \frac{{}^{(r)}d_i^{(\cdot)} - {}^{(r-1)}d_i^{(\cdot)}}{{}^{(r-1)}d_i^{(\cdot)}} \right)^2} \leq [\theta_{out}]. \quad (8)$$

*Bước 4.* Kiểm tra điều kiện dừng

Giá trị sai lệch trung bình ở không gian vào và ra tại vòng lặp thứ  $r$  và  $r - 1$  được xác định dựa vào (6) và (8).

- Nếu  $kt\_in \leq [\theta_{in}]$  và  $kt\_out \leq [\theta_{out}]$ : Dừng

- Nếu ít nhất một trong hai bất đẳng thức (6) và (8) không thỏa mãn:

+ Nếu số vòng lặp  $r < [r]$  thì quay lại bước 1.

+ Nếu số vòng lặp  $r = [r]$  thì quá trình không hội tụ. Tăng số luật mờ  $M = M + 1$ , quay lại bước 1.

$[r]$  là số vòng lặp cho phép).

## 4.2. Thuật toán cắt siêu hộp lai, CSHL

Thuật toán cắt siêu hộp lai, CSHL, được sử dụng để cắt các siêu hộp lai  $hHB$ , thiết lập một tập các siêu hộp thuần chủng phủ lên toàn bộ các mẫu dữ liệu trong tập mẫu huấn luyện  $T_\Sigma$ , làm cơ sở để xây dựng các tập mờ ở không gian dữ liệu vào.

### 4.2.1 Hàm phạt

Xét việc cắt một siêu hộp lai  $hHB$  trong không gian  $\mathfrak{R}^n$  chứa  $P_l$  mẫu  $(\bar{x}_i, y_i)$ ,  $\bar{x}_i = [x_{i1}x_{i2}...x_{in}]$ , để thiết lập các siêu hộp thuần chủng  $pHB$  chứa tất cả các mẫu này.

Gọi  $n_1$  là số lượng các mẫu cùng nhãn  $nh_1$  có số lượng lớn nhất trong  $hHB$  - gọi tắt là loại 1;  $n_2$  là số lượng các mẫu cùng nhãn  $nh_2$  có số lượng lớn thứ hai trong  $hHB$  - gọi tắt là loại 2, ( $n_1 \geq n_2$ ). Gọi  $C_1$  và  $C_2$  theo thứ tự là tâm phân bố của hai loại mẫu này. Gọi  $d_j$  là khoảng cách giữa  $C_1$  và  $C_2$  đo trên trục tọa độ thứ  $j$ ,  $C_j$  là trung điểm khoảng cách tâm phân bố  $C_1$  và  $C_2$  đo trên trục tọa độ thứ  $j$ ,  $j = 1...n$ .

Sử dụng siêu phẳng cắt  $MC_j$  đi qua  $C_j$  và vuông góc với trục  $j$  để cắt  $hHB$ . Như vậy sẽ có  $n$  siêu phẳng cắt và tương ứng sẽ có  $n$  cách cắt khác nhau trong mỗi lần cắt  $hHB$ .  $MC_j$  sẽ phân chia  $hHB$  thành hai siêu hộp nhỏ  $HB_1$  và  $HB_2$ .

Gọi  $n_i^{1j}$  và  $n_i^{2j}$  là số mẫu loại  $i$ ,  $i = 1, 2$  nằm trong  $HB_1$  và  $HB_2$  khi cắt trên trục  $j$ ,  $j = 1...n$ .

Để thấy rằng:

$$0 \leq \left| \frac{n_1^{1j}}{n_1} - \frac{n_2^{1j}}{n_2} \right| = \left| \frac{n_1^{2j}}{n_1} - \frac{n_2^{2j}}{n_2} \right| \leq 1. \quad (9)$$

*Hàm thuần chủng:* Hàm thuần chủng  $\psi^j$ ,  $j = 1...n$  được định nghĩa như sau:

$$\psi^j = \left| \frac{n_1^{1j}}{n_1} - \frac{n_2^{1j}}{n_2} \right|. \quad (10)$$

Hàm  $\psi^j$  phản ánh tình trạng phân bố các mẫu loại 1 và loại 2 trong  $HB_1$  và  $HB_2$ . Ví dụ:

- Nếu  $\psi^j = 0$ , suy ra nếu cắt trên trục  $j$ , tỷ lệ các mẫu loại 1 và loại 2 trên  $HB_1$  và  $HB_2$  là bằng nhau và bằng 50%.

- Nếu  $\psi^j = 1$ , suy ra nếu cắt trên trục  $j$ , tỷ lệ các mẫu loại 1 và loại 2 trong  $HB_1$  và  $HB_2$  là 0% và 100% hoặc 100% và 0%.

- Tổng quát, nếu  $\psi^j = a$  thì tỷ lệ các mẫu loại 1 và loại 2 trên  $HB_1$  và  $HB_2$  sẽ hoàn toàn tính được theo  $a$ .

*Ý nghĩa của giá trị hàm thuần chủng:* giá trị của hàm thuần chủng  $\psi^j$ , được định nghĩa như trên, phản ánh mức độ thuần chủng của trạng thái phân bố các mẫu loại 1 và loại 2 trong  $HB_1$  và  $HB_2$  khi cắt trên trục thứ  $j$ . Giá trị của  $\psi^j$  càng cao thì mức độ thuần chủng càng cao.

*Hàm phạt:* Hàm phạt  $\tau_j$ ,  $j = 1 \dots n$  được định nghĩa như sau:

$$\tau_j = \begin{cases} 0 & \text{if } \psi^j \leq \varepsilon_1 \\ (\psi^j + \Delta) & \text{if } \psi^j \geq \varepsilon_2 \\ 1 & \text{if } \varepsilon_1 < \psi^j < \varepsilon_2 \end{cases}$$

trong đó,  $[\varepsilon_1, \varepsilon_2, \Delta]$  (11b)

là vector các tham số định hướng.

Trong các thí nghiệm kiểm chứng ở bài báo này, chúng tôi chọn các giá trị mặc định như sau:

$$\varepsilon_1 = 0,05, \quad \varepsilon_2 = 0,95 \quad \text{và} \quad \Delta \in [0,35, 0,5]. \quad (12)$$

Như vậy, sử dụng các  $MC_j$  để cắt  $hHB$  trên các trục  $j$  khác nhau sẽ nhận được những giá trị khác nhau của  $\psi^j$  do đó giá trị hàm phạt cũng sẽ khác nhau.

#### 4.2.2. Thuật toán cắt siêu hộp lai, CSHL

Thuật toán CSHL thực hiện cắt trên trục thứ  $k$  sao cho:

$$\tau_k d_k = \max(\tau_j d_j), \quad j = 1 \dots n. \quad (13)$$

Ưu điểm của thủ tục lựa chọn trục để cắt (trong mỗi vòng lặp) của thuật toán CSHL được thể hiện ở tính ưu tiên, mức độ ưu tiên hoặc bị mất quyền tham gia vào quá trình lựa chọn trục cắt của mỗi giải pháp cắt - thông qua giá trị hàm phạt  $\tau_j$ . Cụ thể như sau:

- Nếu giải pháp cắt trên trục thứ  $j$  có giá trị hàm thuần chủng  $\psi^j$  lớn ( $\psi^j \geq \varepsilon_2$ ) thì hàm  $\tau_j$  được “thưởng” một lượng  $\Delta > 0$ . Khi đó,  $\tau^j = \psi^j + \Delta > 1$ , và do đó  $\tau_j d_j > d_j$ . Điều này làm gia tăng khả năng được chọn của giải pháp cắt trên trục thứ  $j$  vì thuật toán CSHL dựa vào mệnh đề (13) để lựa chọn.

- Ngược lại, nếu giá trị hàm thuần chủng  $\psi^j$  nhỏ ( $\psi^j \leq \varepsilon_1$ ) thì  $\tau_j = 0$ , do đó  $\tau_j d_j = 0$ . Nghĩa là giải pháp cắt trên trục thứ  $j$  bị loại khỏi các giải pháp cắt được tham gia vào quá trình chọn lựa giải pháp tốt nhất.

- Nếu giá trị hàm thuần chủng  $\psi^j$  không nằm ở hai phân cực nêu trên ( $\varepsilon_1 < \psi^j < \varepsilon_2$ ) thì  $\tau_j = 1$ , và do đó  $\tau_j d_j = d_j$ . Nghĩa là trong miền này giải pháp cắt trên trục  $j$  vẫn được xem xét, tuy nhiên không được ưu tiên.

Kết hợp với ý nghĩa của giá trị hàm thuần chủng  $\psi^j$  ta có thể thấy rằng trong mỗi vòng lặp, thủ tục cắt của thuật toán CSHL thực hiện chọn lựa các giải pháp cắt tạo ra độ thuần chủng cao trong hai siêu hộp  $HB_1$  và  $HB_2$  được tạo thành. Điều này thật sự cần thiết để tăng hiệu quả của quá trình phân chia không gian dữ liệu vì mục tiêu của quá trình này là xây dựng một tập các bó dữ liệu siêu hộp thuần chủng  $pHB$  phủ toàn bộ các mẫu của tập dữ liệu đã cho  $T_\Sigma$ .

### Thuật toán CSHL

Gọi  $box\_number$  là số siêu hộp lai trong tập hợp tất cả các siêu hộp lai đã có. Quá trình cắt bắt đầu với  $box\_number = 1$ , nghĩa là toàn bộ các mẫu nhĩn trong tập mẫu  $T_\Sigma$  đều thuộc  $hHB$  xuất phát.

*Bước 1.*

- Nếu  $box\_number = 0$  : qua bước 4;
- Nếu  $box\_number > 0$  : xác định siêu hộp lai  $hHB$  có số thứ tự là  $box\_number$  trong tất cả các  $hHB$ . Ký hiệu siêu hộp lai này là  $hHB_{box\_number}$ .

*Bước 2.* Cắt  $hHB_{box\_number}$  thành  $HB_1, HB_2$ .

- Chọn trục  $k$  thỏa mãn (13). Xác định điểm cắt  $C_k$ .
- Cắt trên trục  $k$  tại  $C_k$  theo nguyên tắc: đối với tất cả các mẫu  $\bar{x}_i = [x_{i1}x_{i2}\dots x_{in}]$  thuộc  $hHB_{box\_number}$  :

Nếu  $x_{ik} \leq C_k$  thì  $\bar{x}_i \in HB_1$ ;

Nếu  $x_{ik} > C_k$  thì  $\bar{x}_i \in HB_2$ .

*Bước 3.* Kiểm tra và phân loại  $HB_1, HB_2$ .

- Nếu trong  $HB_1$  và  $HB_2$  có một siêu hộp thuần chủng:
  - + Lưu siêu hộp thuần chủng qua tập các  $pHB$ , lưu siêu hộp lai qua tập các  $hHB$ . Xóa  $hHB_{box\_number}, HB_1, HB_2$ ;
  - + Giữ nguyên  $box\_number$ .

Quay lại Bước 1.

- Nếu  $HB_1$  và  $HB_2$  là hai siêu hộp thuần chủng:
  - + Lưu cả hai qua tập các  $pHB$ . Xóa  $hHB_{box\_number}, HB_1, HB_2$ ;
  - +  $box\_number := box\_number - 1$ .

Quay lại Bước 1.

- Nếu  $HB_1$  và  $HB_2$  là các siêu hộp lai:
  - + Lưu cả hai qua tập các  $hHB$ . Xóa  $hHB_{box\_number}, HB_1, HB_2$ ;
  - +  $box\_number := box\_number + 1$ .

Quay lại Bước 1.

*Bước 4.* Kiểm tra tính phủ (\*) để liên kết các  $pHB$ , xác lập các  $pHB_{fusion}$  lớn hơn.

Để đơn giản, từ phần này về sau các  $pHB_{fusion}$  cũng được ký hiệu  $pHB^{(j)}$ . Ký hiệu này có nghĩa là siêu hộp thuần chủng mang nhĩn  $j$ .

### 4.3. Xây dựng các tập mờ ở không gian dữ liệu vào

Kết quả của thuật toán CSHL là tạo ra một tập các siêu hộp thuần chủng  $pHB$  ở không gian dữ liệu vào. Mỗi  $pHB$  đóng vai trò là một “khung sườn” để xây dựng một tập mờ ở không gian vào.

- Giá trị liên thuộc của mẫu vào  $\bar{x}_i$  vào tập mờ nhĩn  $k$  (được xây dựng trên cơ sở  $pHB_r^{(k)}$ ,  $r = 1 \dots R_k$ ) được tính theo phương pháp Simpson [11]:

$$\mu_{pHB_r^{(k)}}(\bar{x}_i) = \frac{1}{n} \sum_{j=1}^n [1 - f(x_{ij} - \omega_{rj}, \gamma) - f(v_{rj} - x_{ij}, \gamma)], \quad (14a)$$



$$f(x, \gamma) = \begin{cases} 1, & x\gamma > 1 \\ x\gamma, & 0 \leq x\gamma \leq 1 \\ 0, & x < 0\gamma \end{cases} \quad (14b)$$

trong đó,  $pHB_r^{(k)}$ ,  $r = 1 \dots R_k$  là siêu hộp thuần chủng thứ  $r$  trong  $R_k$  siêu hộp thuần chủng cùng mang nhãn  $k$ ; và  $\bar{\omega}_r = [\omega_{r1}\omega_{r2}\dots\omega_{rn}]$ ,  $\bar{v}_r = [v_{r1}v_{r2}\dots v_{rn}]$  là các đỉnh cực trị max-min của  $pHB_r^{(k)}$ ,  $\gamma$  là hệ số, ở đây lấy  $\gamma = 0, 5$ .

- Giá trị liên thuộc của mẫu  $\bar{x}_i$  vào các tập mờ cùng nhãn  $k$  được tính theo max:

$$\mu_{\bar{B}_i^{(k)}}(\bar{x}_i) = \max \left\{ \mu_{pHB_1^{(k)}}(\bar{x}_i), \dots, \mu_{pHB_r^{(k)}}(\bar{x}_i), \dots, \mu_{pHB_{R_k}^{(k)}}(\bar{x}_i) \right\} \quad (15)$$

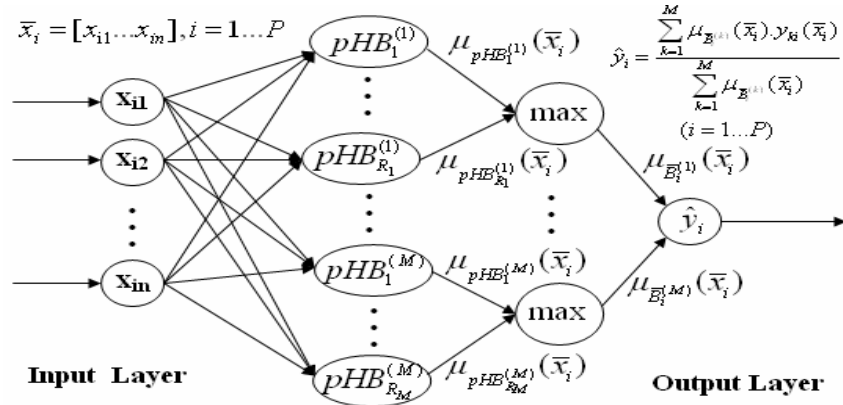
## 5. THUẬT TOÁN HUẤN LUYỆN MẠNG NEURO-FUZZY, HLM

Trong phần này chúng tôi trình bày một thuật toán mới, thuật toán HLM, ứng dụng trong tổng hợp mạng Neuro-Fuzzy xấp xỉ hàm  $f$  từ một tập dữ liệu số cho trước. Thuật toán HLM được chúng tôi xây dựng trên cơ sở ứng dụng các thuật toán CBMM và CSHL (Mục 4). Ưu điểm cơ bản của HLM là cải thiện độ chính xác và tốc độ hội tụ so với [7] và một số thuật toán được khảo sát và kiểm chứng.

Gọi  $M_{\min}$  và  $M_{\max}$  là số luật mờ cực tiểu và cực đại được sử dụng cho khảo sát;  $M$  là số luật mờ hiện tại đang sử dụng để huấn luyện mạng Neuro-Fuzzy.

Giá trị khởi tạo: gán  $j = M_{\min} - 1$ ;

Xác lập giá trị sai số cho phép  $[E]$ .



Hình 2. Cấu trúc mạng Neuron-fuzzy của thuật toán HLM

*Bước 1.* Phân lớp và gán nhãn, xác lập tập mẫu nhãn  $T_\Sigma$

$j := j + 1$ ;  $M = j$ ;

Gọi thuật toán CBMM;

Qua bước 2.

*Bước 2.* Xây dựng mạng Neuro-Fuzzy ứng với số luật mờ được sử dụng là  $j$ ,  $N - F^{(j)}$

- Xây dựng tập các siêu hộp thuần chủng  $pHB$ : gọi thuật toán CSHL;

- Xây dựng mạng Neuro-Fuzzy theo cấu trúc ở Hình 2;

Qua bước 3.

*Bước 3.* Xác định sai số  $E^{(j)}$  theo chuẩn  $L_2$  của mạng  $N - F^{(j)}$

- Tính giá trị liên thuộc theo (14) và (15);
- Tính  $\hat{y}_i$  theo (17);
- Tính sai số bình phương trung bình:

$$E^{(j)} = \frac{1}{N} \sum_{i=1}^P (y_i - \hat{y}_i)^2, \quad (16)$$

trong đó, dữ liệu ra của mạng ứng với mẫu thứ  $i$  được tính (Hình 2):

$$\hat{y}_i = \frac{\sum_{k=1}^M \mu_{B_i}^{(k)}(\bar{x}_i) y_{ki}(\bar{x}_i)}{\sum_{k=1}^M \mu_{B_i}^{(k)}(\bar{x}_i)}, \quad (i = 1 \dots P), \quad (17)$$

$$y_{ki} = \sum_{j=1}^n a_j^{(k)} x_{ij} + a_0^{(k)}.$$

- Kiểm tra số luật mờ:

Nếu  $j < M_{\max}$ : quay lại bước 1.

Nếu  $j = M_{\max}$ : qua bước 4.

*Bước 4.* Chọn mạng tốt nhất,  $N - F^{(g)}$ , sao cho:

$$E^{(g)} = \min(E^{(j)}), \quad j = M_{\min} \dots M_{\max}. \quad (18)$$

Qua bước 5.

*Bước 5.* Kiểm tra điều kiện dừng

- Nếu  $E^{(g)} \leq [E]$ : kết thúc. (Kết quả tổng hợp ANFIS là mạng  $N - F^{(g)}$ );

- Nếu  $E^{(g)} > [E]$ : qua bước 6.

*Bước 6.* Kiểm tra khả năng hội tụ tới  $[E]$  của lời giải

Xét chỉ số  $g$  trong (18):

- Nếu  $g = M_{\max}$ : xác lập lại  $M_{\min} := M_{\max}$ ,  $M_{\max} := M_{\max} + \Delta M_{\max} | \Delta M_{\max} \in N$ ,  $j = M_{\min} - 1$ , quay lại bước 1.

- Nếu  $g = M_{\min}$ : xác lập lại  $M_{\max} := M_{\min}$ ,  $M_{\min} := M_{\min} - \Delta M_{\min} | \Delta M_{\min} \in N$ ,  $j = M_{\min} - 1$ , quay lại bước 1.

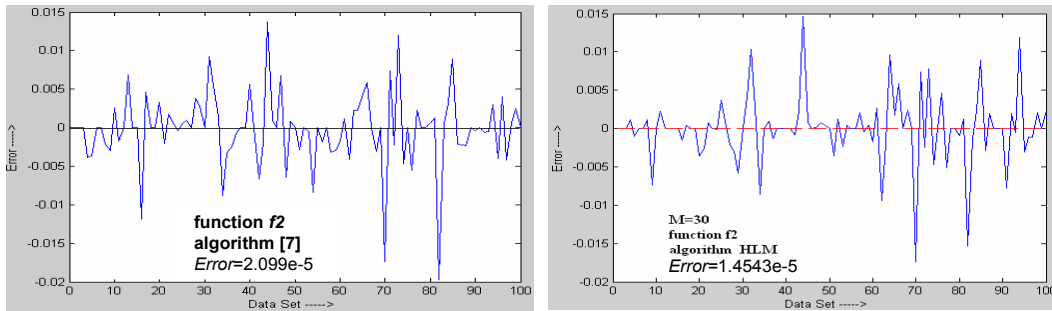
- Nếu  $M_{\min} < g < M_{\max}$ : lời giải không hội tụ tới  $[E]$ . Kết thúc.

## 6. THÍ NGHIỆM KIỂM CHỨNG

### 6.1. Thí nghiệm 1: xấp xỉ hàm $y_2$ [7]

Hàm  $y_2 = (5 - x_2)^2 / [3(5 - x_1)^2 + (5 - x_2)^2]$  của [7] được sử dụng để xây dựng tập mẫu  $tr\_set$  gồm 100 mẫu. Các giá trị  $\bar{x} = [x_1, x_2]$  được lấy ngẫu nhiên trong khoảng  $[0, 10]$  nhờ hàm random của Matlab. Dữ liệu ra được tính theo  $y_2 = (5 - x_2)^2 / [3(5 - x_1)^2 + (5 - x_2)^2]$ .

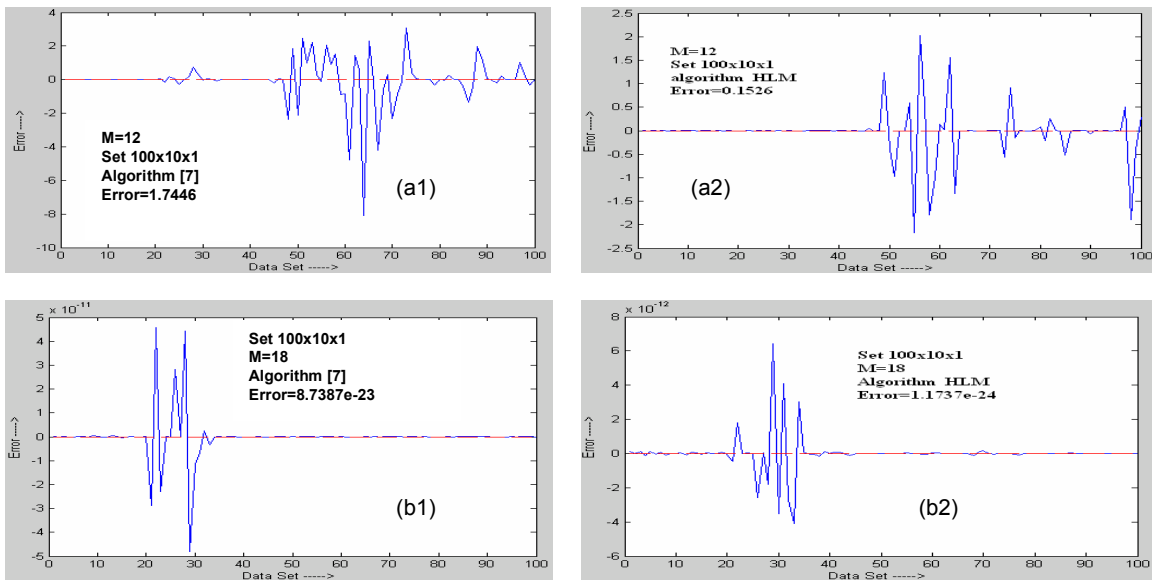
Số luật mờ	Sai số $E(18)$ ứng với từng thuật toán	
	[7]	HLM
<b>M=10</b>	$2,000.10^{-3}$	$1,650.10^{-3}$
<b>M=15</b>	$2,200.10^{-3}$	$2,100.10^{-3}$
<b>M=20</b>	$25,000.10^{-4}$	$1,338.10^{-4}$
<b>M=25</b>	$3,545.10^{-5}$	$2,326.10^{-5}$
<b>M=30</b>	$2,099.10^{-5}$	$1,4543.10^{-5}$



Hình 3. So sánh sai số đáp ứng  $Error_i = y_i - \hat{y}_i, i = 1...100$  và giá trị sai số bình phương trung bình  $E(= Error)$  của thuật toán [7] và HLM ứng với tập mẫu  $tr\_set$  với số luật mờ  $M = 30$ .

Ở Hình 3 thể hiện sai số đáp ứng  $Error_i = y_i - \hat{y}_i, i = 1...100$  và giá trị sai số bình phương trung bình  $E$  của thuật toán [7] và thuật toán HLM (có  $\varepsilon_1 = 0,05, \varepsilon_2 = 0,95, \Delta = 0,35$ ) ứng với số luật mờ  $M = 30$ . Hình 3 và số liệu cụ thể ứng với các giá trị  $M$  được thể hiện trên Bảng 1 cho thấy hiệu quả của thuật toán mới, thuật toán HLM.

**6.2. Thí nghiệm 2. Xấp xỉ hàm từ tập dữ liệu [9]**



Hình 4. Giá trị sai lệch và giá trị sai số bình phương trung bình  $E(= Error)$  của các thuật toán [7] và HLM khi số luật mờ  $M = 12$  (a1, a2) và  $M = 18$  (b1,b2), tập mẫu “Daily Data of Stock A” của [9]

Sử dụng tập dữ liệu gồm 100 mẫu, 10 biến vào một biến ra  $([x_1, x_2, \dots, x_{10}], y)$  trong phụ lục IV “Daily Data of Stock A” của [9] làm tập huấn luyện mạng cho [7] và HLM ( $\varepsilon_1 = 0,05$ ,  $\varepsilon_2 = 0,95$ ,  $\Delta = 0,5$ ). Kết quả thể hiện trên Hình 4 và Bảng 2 cho thấy độ chính xác của HLM cao hơn độ chính xác của [7].

### 6.3. Thí nghiệm 3

Sử dụng hàm  $y$  của [13],  $y = (1 + x_1^{-2} + x_2^{-1,5})^2$ ,  $x_1, x_2 \in [1, 5]$  để xây dựng tập mẫu gồm 50 mẫu tương tự tập mẫu đã được sử dụng trong [13]. Thực hiện huấn luyện mạng xấp xỉ hàm  $y$  với thuật toán HLM, [7] và các thuật toán được trình bày trong [13, 16, 17]. Các kết quả nhận được cho trong Bảng 3 cho thấy độ chính xác của HLM cao hơn rất nhiều so với độ chính xác của [7, 13, 16, 17].

Bảng 2

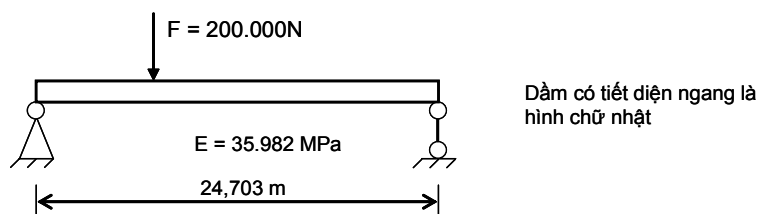
Số luật mờ	Sai số $E$ ứng với từng thuật toán	
	[7]	HLM
<b>M=10</b>	13,4148	0,5222
<b>M=12</b>	1,74460	0,1526
<b>M=14</b>	$3,5028 \cdot 10^{-5}$	$1,2971 \cdot 10^{-6}$
<b>M=18</b>	$8,7387 \cdot 10^{-23}$	$1,1737 \cdot 10^{-24}$
<b>M=20</b>	0,0028	$1,1006 \cdot 10^{-13}$

Bảng 3

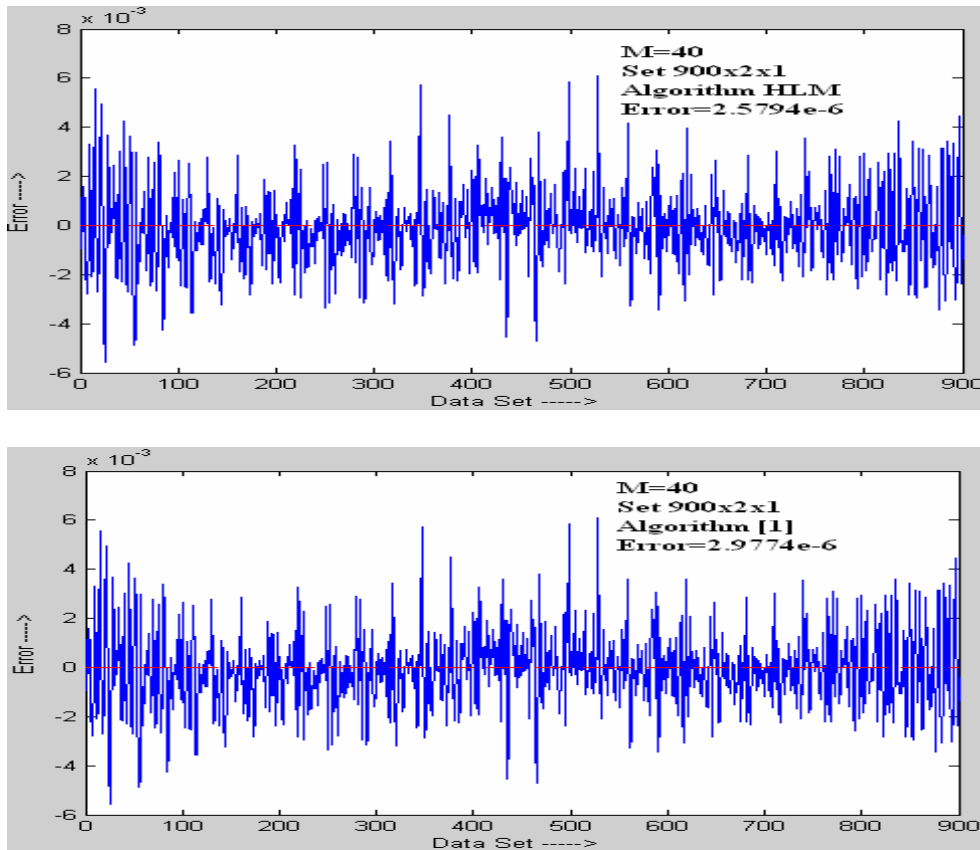
Số luật mờ	ĐL	Sai số $E$ ứng với từng thuật toán				
		[13]	[16]	[17]	[7]	HLM
<b>M=6</b>	E	0,0589	0,0572	0,0599	0,0221	<b>0,0167</b>
<b>M=8</b>	E	0,0500	0,0499	0,0499	0,0220	<b>0,0198</b>
<b>M=10</b>	E	0,0148	0,0149	0,0149	0,0188	<b><math>0,0244 \cdot 10^{-1}</math></b>

### 6.4. Thí nghiệm 4. Nhận dạng đầm cầu trong quan hệ tải - độ võng

Khảo sát một đầm cầu như trên Hình 5. Sử dụng mô hình phần tử hữu hạn (FEM) [3] cho đầm với số nút  $N = 30$ . Lực tập trung  $F = 200.000\text{N}$  được thay đổi điểm đặt tại nhiều vị trí khác nhau trên đầm. Tiến hành đo độ võng của đầm tại các nút để xây dựng hai tập mẫu: tập TrainingSet có số mẫu  $P = 900$  và tập TestSet có số mẫu  $P = 100$  sao cho một mẫu bất kỳ thuộc tập này sẽ không thuộc tập kia.



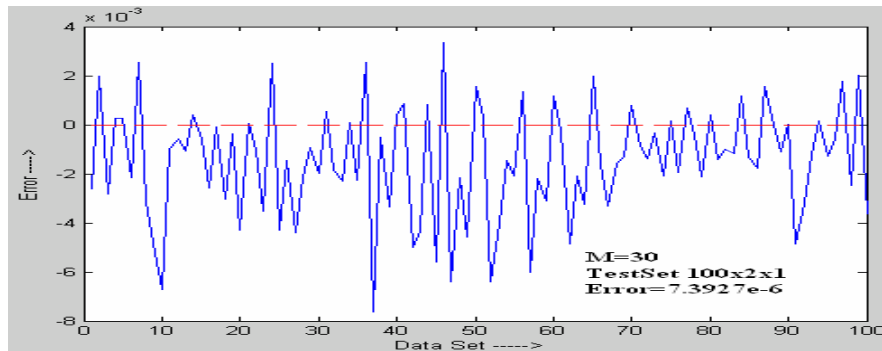
Hình 5. Mô hình thí nghiệm đo độ võng của đầm cầu trên hai gối khi tải thay đổi



Hình 6. Sai lệch  $Error_i = y_i - \hat{y}_i$  và sai số bình phương trung bình  $E$  (18) ứng với từng thuật toán

- Sử dụng hai thuật toán, thuật toán [7] và thuật toán HLM, để tổng hợp mạng Neuro-Fuzzy nhận dạng dầm trong quan hệ lực - độ võng thông qua tập dữ liệu TrainingSet. Sai số  $Error_i = y_i - \hat{y}_i$  và sai số bình phương trung bình  $E$  ứng với từng thuật toán được thể hiện trên Hình 6. Kết quả cho thấy độ chính xác của thuật toán HLM cao hơn độ chính xác của [7].

- Để kiểm tra khả năng nhận dạng và dự báo của mạng Neuro-Fuzzy được tổng hợp bởi thuật toán HLM chúng tôi sử dụng tập TestSet làm dữ liệu vào cho mạng đã được tổng hợp bởi thuật toán HLM với tập dữ liệu khác: tập TrainingSet (gọi tắt là mạng N-F/TrainingSet). Cụ thể, sử dụng tập TestSet và N-F/TrainingSet chúng tôi đã tính và kiểm tra hai nội dung sau. Thứ nhất, tính sai lệch giữa dữ liệu ra cho trong tập mẫu TestSet và dữ liệu ra tương ứng của mạng N-F/TrainingSet (khi sử dụng tín hiệu vào cho mạng là tín hiệu vào của tập TestSet),  $Error_i = y_i - \hat{y}_i$ . Thứ hai, tính sai số bình phương trung bình  $E$  giữa tín hiệu ra của mạng N-F/TrainingSet (khi sử dụng tín hiệu vào cho mạng là tín hiệu vào của tập TestSet) và tín hiệu ra của tập TestSet. Kết quả thể hiện trên Hình 7 cho thấy khả năng nhận dạng đối tượng và dự báo của mạng Neuro-Fuzzy được tổng hợp bởi thuật toán HLM khá tốt.



Hình 7. Khả năng đáp ứng của mạng Neuro-Fuzzy được huấn luyện bởi thuật toán HLM

## 7. KẾT LUẬN

Trong các bài toán nhận dạng, dự báo, đo lường và điều khiển, thông thường để có thể tác động vào đối tượng ta cần phải xây dựng mô hình toán của đối tượng. Đối với các đối tượng phức tạp, khi mà các quá trình tự nhiên diễn ra bên trong chưa thể xác định được thì không thể xây dựng được mô hình toán theo cách mô hình hóa hệ thống (hay còn được gọi là mô hình hộp trắng) mà phải sử dụng phương pháp nhận dạng hệ thống dựa vào dữ liệu vào-ra thực nghiệm và các thông tin biết trước về đối tượng (nếu có). Thông tin biết trước có thể là những hiểu biết không đầy đủ về các quy luật vật lý, hoặc các phát biểu ngôn ngữ mô tả đặc tính của đối tượng... Tùy theo mức độ thông tin biết trước được sử dụng mà ta có mô hình hộp xám hoặc mô hình hộp đen.

Ở bài báo này chúng tôi trình bày một giải pháp tổng hợp hệ thống suy diễn neuro-fuzzy thích nghi ANFIS, thuật toán HLM, là một công cụ cho các bài toán nhận dạng, dự báo, đo lường và điều khiển theo mô hình hộp xám và hộp đen. Đặc điểm của HLM là: với một tập dữ liệu đã cho, HLM có thể tự động xác lập cấu hình mạng neuro-fuzzy, bao gồm tự động xác định số lượng các tập mờ, tự động xác lập các hàm liên thuộc, tự động xác định số lượng neuron trên lớp ẩn phù hợp với độ chính xác yêu cầu; đồng thời HLM có khả năng nhận ra và cho biết mức độ chính xác mà ANFIS không thể đạt tới ứng với tập dữ liệu huấn luyện đã cho.

Việc gia tăng độ chính xác trong giai đoạn chia bố dữ liệu min-max của thuật toán CBMM và trong giai đoạn xây dựng các bó siêu hộp thuần chủng của thuật toán CSHL có tác dụng gia tăng mức độ ràng buộc giữa các tập mờ ở không gian vào và các lớp dữ liệu dạng siêu phẳng ở không gian ra, nghĩa là làm gia tăng mức độ phù hợp trong mối liên hệ giữa không gian nền của các tập mờ (là không gian của các đại lượng vật lý cho trong tập mẫu) với chính các tập mờ được xây dựng trên nó, và do đó làm gia tăng độ chính xác của thuật toán HLM. Ngoài ra, việc định hướng hợp lý trong quá trình chia cắt các siêu hộp có tác dụng làm gia tăng tốc độ hội tụ, giảm số lượng tập mờ (giảm số lượng bó được tạo thành) và do đó giảm mức độ phức tạp của mạng.

## TÀI LIỆU THAM KHẢO

- [1] Chunshien Li and Chun-Yi Lee, Self-organizing neuro-fuzzy system for control of unknown plants, *IEEE Transactions on Fuzzy Systems* **11** (1) (February 2003).
- [2] Frank Hppner, Member, IEEE, and Frank Klawonn, Member, IEEE, A contribution to convergence theory of fuzzy c-means and derivatives, *IEEE Transactions on Fuzzy Systems* **11** (5) (October 2003).
- [3] Gary R. Consolazio, Iterative equation solver for bridge analysis using neural networks, *Computer-Aided Civil and Infrastructure Engineering* **15** (2000) 107–119.
- [4] Ho-Sung Park and Sung-Kwun Oh, Multi-FNN identification based on HCM clustering and evolutionary fuzzy granulation, *International Journal of Control, Automation, and Systems* **1** (2) (June 2003).
- [5] Janusz A. Starzyk, Zhen Zhu, Tsun-Ho Liu, Self-organizing learning array, *IEEE Transactions Networks* **16** ( 2) (March 2005).
- [6] Jianqin Mao, Jiangang Zhang, Yufang Yue, Adaptive-tree-structure-based fuzzy inference system, *IEEE Trans. on Fuzzy Syst.* **13** (1) (February 2005).
- [7] Massimo Panella, Antonio Stanislao Gallo, An input output clustering approach to the synthesis of ANFIS networks, *IEEE Transactions on Fuzzy Systems* **13** (1) (February 2005).
- [8] Massimo Panella, Antonello Rizzi, and Fabio Massimo Frattale Mascioli, Adaptive resolution min-max classifiers, *IEEE Transactions on Neural Networks* **13** (2) (March 2002).
- [9] M. Sugeno and T. Yasukawa, A fuzzy logic based approach to qualitative modeling, *IEEE Trans. on Fuzzy Systems* **1** (1) (Feb. 1993) 7–31.
- [10] Nicola K. Kasabov, Senior member, IEEE, and Qun Song, DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction, *IEEE transactions on Fuzzy Systems* **10** (2) (April 2002).
- [11] P.K. Simpson, Fuzzy min-max neural networks Part 2: Clustering, *IEEE Trans. Neural Netw.* **1** (1) (1993) 32–45.
- [12] Phạm Thượng Hàn, Nguyễn Trọng Quế, và Nguyễn Văn Hòa, *Kỹ thuật đo lường các đại lượng vật lý*, tập 1, Nhà xuất bản Giáo dục, 1996.
- [13] Shie-Jue Lee, Member, IEEE, and Chen-Sen Ouyang, A neuro-fuzzy system modeling with self-constructing rule generation and hybrid SVD-based learning, *IEEE Transactions on Fuzzy Systems* **11** (3) (June 2003).
- [14] S.L. Chiu, Fuzzy model identification based on cluster estimation, *J. Intell. Fuzzy Sys.* **2** (1994).
- [15] T. Takagi and M. Sugeno, Fuzzy identification of systems and applications to modeling and control, *IEEE Trans. Syst. Man, Cybern.* **SMC-15** (1) (Jan. 1985) 116–132.
- [16] Wong and C. C. Chen, A hybrid clustering and gradient decent approach for fuzzy modeling, *IEEE Trans. Syst. Man, Cybern. B* **29** (Dec. 1999) 686–693.
- [17] Y. Lin, G. A. Cunningham III, and S. V. Coggeshall, Using fuzzy partitions to create fuzzy system from input-output data and set the initial weights in fuzzy neural network, *IEEE Trans. Fuzzy systems* **5** (Aug. 1997) 614–621.

Nhận bài ngày 26 - 7 - 2007

Nhận lại sau sửa ngày 14 - 3 - 2008