

## XÂY DỰNG HỆ THỐNG SUY DIỄN NEURO-FUZZY TRÊN CƠ SỞ XÁC LẬP CÁC TẬP MỜ TỐI ƯU Ở KHÔNG GIAN VÀO

Nguyễn Sỹ Dũng<sup>(1)</sup>, Ngô Kiều Nhi<sup>(2)</sup>

(1) Trường Đại học Công nghiệp Tp.HCM

(2) Trường Đại học Bách khoa, ĐHQG-HCM

(Bài nhận ngày 30 tháng 07 năm 2007, hoàn chỉnh sửa chữa ngày 06 tháng 03 năm 2008)

**TÓM TẮT:** Bài báo trình bày ba thuật toán mới: thuật toán cắt siêu hộp lai (CSHL) và hai thuật toán huấn luyện mạng neuron – fuzzy (HLM1 và HLM2), được dùng để tổng hợp hệ thống suy diễn neuro-fuzzy xấp xỉ hàm chưa biết  $y = f(\bar{x})$  diễn tả mối quan hệ giữa tín hiệu vào và tín hiệu ra của đối tượng thông qua một tập dữ liệu số. Mục tiêu đặt ra là gia tăng độ chính xác của phép xấp xỉ. Thuật toán CSHL, được xây dựng trên cơ sở ứng dụng một công cụ mới mang tên hàm phạt  $\tau$ , dùng để phân chia không gian dữ liệu thành các bó dữ liệu, làm cơ sở xác lập các tập mờ ở không gian vào. HLM1, được phát triển trên cơ sở kết hợp thuật toán CSHL và thuật toán Hyperplane Clustering của [1], là thuật toán huấn luyện mạng bao hàm việc định lượng mờ ở input, thiết lập cấu trúc mạng neuron-fuzzy và giải mờ ở output. Thuật toán HLM2 là sự phát triển tiếp theo của HLM1 trong đó đề cập tới việc xác định các tập mờ tối ưu ở không gian vào bằng phương pháp huấn luyện mạng neuron. 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 thuật toán [1] và một số thuật toán đã công bố.

### 1. ĐẶT VẤN ĐỀ

Cho trước một tập  $T_{\Sigma}$  gồm  $P$  cặp dữ liệu số  $(\bar{x}_i, y_i)$   $\bar{x}_i = [x_{i1} x_{i2} \dots x_{in}]$  thể hiện giá trị của một hàm chưa biết  $f$  tại các điểm  $\bar{x}_i$ , ( $y_i = f(\bar{x}_i)$ ),  $i = 1 \dots P$ . 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 của Takagi và Sugeno [7], còn được gọi là mô hình T-S. Theo mô hình này hàm  $f$  được xấp xỉ qua một hệ thống suy diễn mờ gồm  $M$  luật mờ T-S. Luật thứ  $k$  có dạng:

$$R^{(k)} : \text{nếu } x_{i1} \text{ là } B_1^{(k)} \text{ và } \dots \text{ và } x_{in} \text{ là } B_n^{(k)} \text{ thì } y_{ki} = \sum_{j=1}^n a_j^{(k)} x_{ij} + a_0^{(k)} \quad (1)$$

trong đó:  $\bar{x}_i = [x_{i1} x_{i2} \dots x_{in}]$  là vector dữ liệu vào thứ  $i$ ,  $i = 1 \dots P$ .  $B^{(k)}$  là tập mờ ở input;  $a_j^{(k)}$   $j = 1 \dots n$ , là các trọng số thực ở output;  $y_{ki}$  là dữ liệu ra ứng với luật mờ thứ  $k$ ,  $k = 1 \dots M$ .

Theo mô hình T-S, phải thực hiện chia bó dữ liệu để xây dựng các tập mờ  $B^{(k)}$  ở không gian vào. Một trong những phương pháp chia bó thường được sử dụng là phương pháp chia bó mờ của [5]. Gần đây, một nghiên cứu phát triển phương pháp này được trình bày trong [1] và [2], trong đó sử dụng giải pháp chia lớp dữ liệu ở không gian dữ liệu vào nhưng quá trình phân chia được tiến hành trong mối liên hệ ràng buộc qua lại giữa không gian dữ liệu vào và không gian dữ liệu ra. Theo phương pháp này, tập dữ liệu huấn luyện  $T_{\Sigma}$  được chia thành nhiều lớp nhẵn. Tập mẫu được gán nhãn  $T_{\Sigma}$ , gọi tắt là tập mẫu nhẵn, là cơ sở để xây dựng một tập các bó thuần chủng  $pHB$ , 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 được xây dựng dựa vào các điểm cực trị này. Tập mờ  $B^{(k)}$  được xác lập dựa vào các giá trị min, max và hàm liên thuộc của siêu hộp tương ứng.

Phương pháp chia bó của [1][2] phản ánh quan hệ ràng buộc về dữ liệu giữa không gian vào và không gian ra của tập dữ liệu huấn luyện mạng thông qua các tập mờ được tạo thành, do đó đã gia tăng độ chính xác của phép xấp xỉ hàm  $f$  so với các thuật toán chia bó chỉ dựa vào thuần túy các đặc trưng dữ liệu của từng miền: chỉ dựa vào không gian dữ liệu vào [5]; chỉ dựa vào không gian dữ liệu ra [3]. Tuy nhiên, hạn chế của thuật toán chia bó ARC của [2], được ứng dụng để tổng hợp mạng ANFIS của [1], bộc lộ khi lựa chọn giải pháp phân chia không gian dữ liệu thành các bó dữ liệu (sẽ được trình bày chi tiết ở mục III) đã làm giảm hiệu quả của [1]. Trong bài báo này chúng tôi trình bày một phát triển tiếp theo của [1][2], trong đó giải pháp định hướng tối ưu cho quá trình phân chia không gian dữ liệu để xây dựng các tập mờ  $B^{(k)}$  được đề xuất làm cơ sở để phát triển ba thuật toán mới: thuật toán chia bó CSHL và hai thuật toán tổng hợp mạng neuro-fuzzy: thuật toán HLM1 và HLM2.

## 2. MỘT SỐ KHÁI NIỆM VÀ THUẬT TOÁN LIÊN QUAN

### 2.1. Một số khái niệm

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

- Bó siêu phẳng, nhãn của bó siêu phẳng và nhãn của mẫu dữ liệu. Nếu sử dụng thuật toán Hyperplane Clustering của [1] cho tập mẫu  $T_x$  với M luật mờ chúng ta sẽ nhận được M bó dạng siêu phẳng ở không gian dữ liệu vào, gọi tắt là bó siêu phẳng, được gán nhãn. Nếu mẫu  $\bar{x}_i = [x_{i1} x_{i2} \dots x_{im}]$  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 n chiều, siêu hộp HB có các mặt là các siêu phẳng, mỗi siêu phẳng song song với một mặt phẳng tọa độ và đi qua một trong hai đỉnh cực trị min, max.

Siêu hộp thứ t, ký hiệu HBt, có Tt là tập hợp của các mẫu thuộc nó.

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

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

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

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

- Siêu hộp không phủ lên một siêu hộp khác - thỏa tính phủ (\*) (overlap condition): Cho trước siêu hộp HBh. Xét một siêu hộp HBk bất kỳ. Ta nói rằng HBh không phủ lên HBk khi và chỉ khi:

$$\bar{\omega}_h < \bar{v}_k \quad \text{hoặc} \quad \bar{v}_h > \bar{\omega}_k$$

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 HBh không phủ lên bất kỳ một siêu hộp nào thuộc  $L_p$  và  $L_h$  thì ta nói rằng HBh 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  $pHB_f^{(m)}$  cùng nhãn  $m$  được gọi là siêu hộp liên kết của hai siêu hộp trên nếu thỏa mãn đồng thời ba mệnh đề sau:

- $T_f = T_k \cup T_h$
- $\bar{\omega}_f = \max(\bar{\omega}_k, \bar{\omega}_h)$ ;  $\bar{v}_f = \min(\bar{v}_k, \bar{v}_h)$
- $pHB_f^{(m)}$  thỏa mãn tính phủ.

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

## 2.2. Thuật toán Hyperplane Clustering [1]

Sử dụng thuật toán Hyperplane Clustering của [1], không gian dữ liệu của tập mẫu sẽ được phân chia để xác lập các bó siêu phẳng ở không gian dữ liệu vào, thiết lập các siêu phẳng ở không gian dữ liệu ra, và gán nhãn cho tập mẫu huấn luyện  $T_\Sigma$  nhằm xác lập một siêu hộp lai (ký hiệu là hHB) chứa toàn bộ các mẫu đã được gán nhãn trong  $T_\Sigma$ . Thuật toán dựa trên hai nguyên tắc:

- Số lớp ở input bằng số siêu phẳng ở output và bằng số luật mờ M.
- Nếu một mẫu  $\bar{x}_i$  ở input thuộc lớp thứ  $k$ ,  $\Gamma^{(k)}$ ,  $k=1...M$  thì  $(\bar{x}_i, y_i)$  sẽ được gán cho siêu phẳng cùng nhãn  $A_k$  ở output và ngược lại.

## 3. HÀM PHẠT VÀ THUẬT TOÁN CẮT SIÊU HỘP LAI (CSHL)

Trong phần này chúng tôi đề xuất một thuật toán mới, 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.

### 3.1. Hàm phạt

Xét việc cắt một hHB trong không gian  $\mathfrak{R}^n$  chứa  $Pl$  mẫu  $(\bar{x}_i, y_i)$ ,  $\bar{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]$  để thiết lập các 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 mặt phẳng cắt MCj đi qua Cj và vuông góc với trục j để cắt hHB. Như vậy sẽ có n mặt phẳng cắt và tương ứng sẽ có n cách cắt khác nhau trong mỗi lần cắt hHB. Mặt phẳng MCj sẽ phân chia hHB thành hai siêu hộp nhỏ HB1 và HB2.

Gọi  $n_i^{1j}$  và  $n_i^{2j}$  là số mẫu loại i,  $i=1,2$  nằm trong HB1 và HB2 khi cắt trên trục j,  $j=1\dots n$ .

Gọi  $\psi^{1j}$  và  $\psi^{2j}$  là các hàm được định nghĩa:

$$\psi^{1j} = \left| \frac{n_1^{1j}}{n_1} - \frac{n_2^{1j}}{n_2} \right|; \quad \psi^{2j} = \left| \frac{n_1^{2j}}{n_1} - \frac{n_2^{2j}}{n_2} \right| \quad (3)$$

Dễ thấy rằng:  $0 \leq \psi^{1j} = \psi^{2j} \leq 1$

$$\text{Đặt} \quad \psi^j = \psi^{1j} = \psi^{2j} \quad (4)$$

Hàm  $\psi^j$ , được gọi là hàm thuần chủng, phản ánh tình trạng phân bố các mẫu loại 1 và loại 2 trong HB1 và HB2. Ví dụ:

- Nếu  $\psi^j = 0$ , suy ra nếu cắt trên trục j, tỷ lệ các mẫu loại 1 và loại 2 trên HB1 và HB2 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 HB1 và HB2 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 HB1 và HB2 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 HB1 và HB2 khi cắt trên trục thứ j. Giá trị của  $\psi^j$  càng cao thì mức độ thuần chủng càng cao. Mức độ thuần chủng cao là cơ khi lựa chọn giải pháp cắt vì khi đó thời gian phân chia tập dữ liệu để xây dựng các siêu hộp thuần chủng sẽ rút ngắn lại.

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

$$\tau_j = \begin{cases} 0 & \text{if } \psi^j \leq \varepsilon_1 \\ (\psi^j + \Delta) & \text{if } \psi^j \geq \varepsilon_2 \\ 1 & \text{if } \varepsilon_1 < \psi^j < \varepsilon_2 \end{cases} \quad (5a)$$

trong đó:

$$[\varepsilon_1, \varepsilon_2, \Delta] \quad (5b)$$

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:

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

Như vậy, sử dụng các MCj 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.

### 3.2. Thuật toán CSHL

Sự khác nhau giữa thủ tục cắt siêu hộp lai (CSHL) để xây dựng một tập các siêu hộp thuần chủng được đề xuất trong bài báo này với thủ tục ARC cutting của [2] thể hiện ở chỗ nếu như ARC cutting thực hiện cắt trên trục thứ  $k$  có khoảng cách  $d_k$  giữa  $C1$  và  $C2$  lớn nhất:

$$d_k = \max(d_j), j = 1 \dots n. \quad (6)$$

thì đối với thuật toán CSHL việc chọn trục  $k$  để cắt trong mỗi lần cắt phải dựa vào hai tiêu chí ưu tiên: một là giá trị hàm thuần chủng  $\psi^k$  lớn, hai là khoảng cách tâm  $d_k$  lớn. Kết quả là CSHL thực hiện cắt trên trục thứ  $k$  sao cho:

$$\tau_k d_k = \max(\tau_j d_j), j = 1 \dots n \quad (7)$$

Ưu điểm của thủ tục lựa chọn trục để cắt (trong mỗi vòng lặp) của thuật toán CSHL so với thủ tục ARC cutting của [2] 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  $\Delta$ . 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  $j$  (so với thủ tục cắt ARC cutting của [2]) vì thuật toán CSHL dựa vào mệnh đề (7) 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 thủ tục cắt của thuật toán CSHL và ARC cutting của [2] là như nhau vì các mệnh đề (6) và (7) là đồng nhất.

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 HB1 và HB2 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  $T_x$ . Khác với ARC cutting của [2], thủ tục cắt của thuật toán CSHL khai thác triệt để hai miền phân cực của hàm thuần chủng ( $\psi^j \leq \varepsilon_1$  và  $\psi^j \geq \varepsilon_2$ ): ưu tiên các trường hợp thuộc miền có  $\psi^j \geq \varepsilon_2$  và loại, không xét các trường hợp thuộc miền có  $\psi^j \leq \varepsilon_1$ . Định hướng này nhằm rút ngắn quá trình phân chia không gian dữ liệu.

Ta có thể định lượng rõ hơn kết luận mang tính định tính nêu trên qua ví dụ sau:

Cắt hHB trong không gian  $\mathcal{R}^2$  chứa 3 loại mẫu với số lượng:  $\circ$   $n_1 = 20$ ;  $\bullet$   $n_2 = 20$ ;  $*$   $n_3 = 12$ . Các mẫu  $\circ$  và  $\bullet$  có số lượng lớn nên được chọn để thực hiện quy trình cắt. Xét



hai trường hợp ở hình 2 với giả thiết khoảng cách tâm  $d_1, d_2$  trong hai trường hợp đã được định trước.

Trường hợp ở hình 2a  $d_1 = 3 < d_2 = 11$

Để dàng tính được:  $\tau_1 d_1 = 0,21 > \tau_2 d_2 = 0$

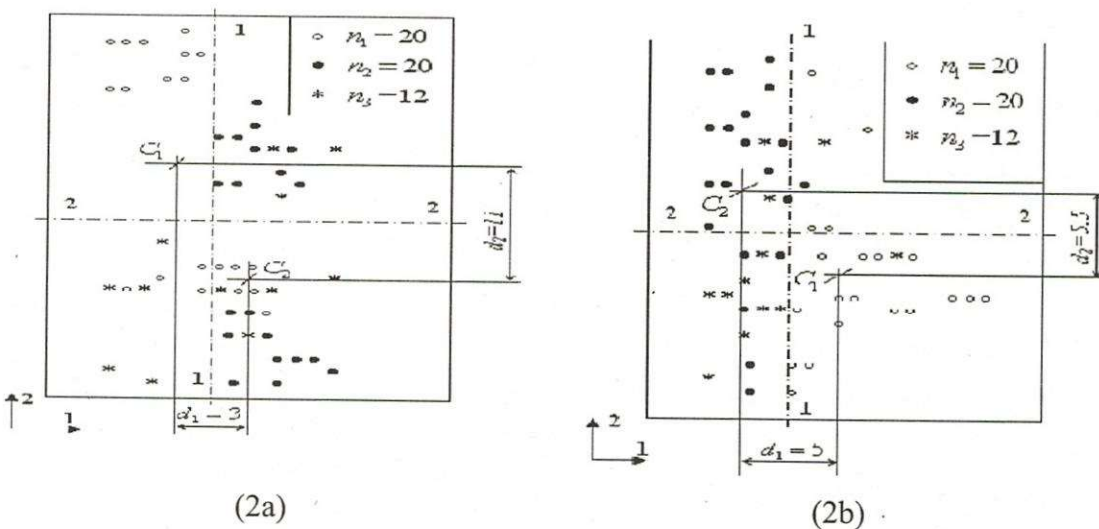
Do đó ARC của [1] cắt trên trục 2; CSHL cắt trên trục 1.

Trường hợp ở hình 2b  $d_1 = 5 < d_2 = 5,5$

Tương tự, ta tính được:  $\tau_1 d_1 = 6,425 > \tau_2 d_2 = 5,5$

Do đó ARC của [1] cắt trên trục 2; CSHL cắt trên trục 1.

Như vậy, cả hai trường hợp CSHL chọn trục cắt là trục 1 (cắt theo 1-1) mặc dù có  $d_1 < d_2$ ; ARC cắt trên trục 2 (cắt theo 2-2). Xét phân bố các mẫu trên hai hình ta thấy rằng việc cắt trên trục 1 hợp lý hơn vì sẽ tạo ra HB1 và HB2 có độ thuần chủng cao hơn và do đó làm gia tăng tốc độ hội tụ của quá trình chia bố.



Hình 1. Chọn giải pháp cắt theo ARC [1] và CSHL

**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 nhãn trong tập mẫu  $T_x$  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 (7). 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 pHBfusion lớn hơn.

Để đơn giản, từ phần này về sau các pHBfusion cũng được ký hiệu  $pHB_i^{(j)}$ . Ký hiệu này có nghĩa là siêu hộp thuần chủng thứ i, mang nhãn j.

#### 4. THUẬT TOÁN HUẤN LUYỆN MẠNG NEURO-FUZZY THỨ NHẤT, HLM1

##### 4.1. Cấu trúc mạng neuro-fuzzy của HLM1

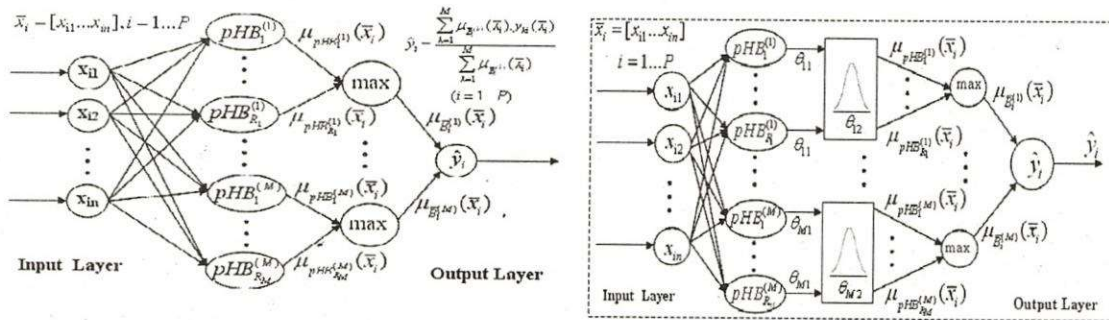
Cấu trúc mạng neuro-fuzzy của HLM1 tương tự như cấu trúc ANFIS của [1], tuy nhiên  $\hat{y}_i, i = 1 \dots P$  được tính theo phương pháp điểm trọng tâm (hình 3a).

- Giá trị liên thuộc của mẫu vào  $\bar{x}_i, i = 1 \dots P$  vào tập mờ nhân k,  $k = 1 \dots M$  (được xây dựng trên cơ sở  $pHB_r^{(k)}, r = 1 \dots R_k$ ) được tính theo phương pháp Simpson [5]:

$$\mu_{pHB_r^{(k)}}(\bar{x}_i) = \frac{1}{n} \sum_{j=1}^n [1 - f(x_{ij} - \omega_{rj}, \gamma) - f(v_{rj} - x_{ij}, \gamma)] \quad (8a)$$

$$f(x, \gamma) = \begin{cases} 1, & x\gamma > 1; \\ x\gamma, & 0 \leq x\gamma \leq 1; \\ 0, & x\gamma < 0. \end{cases} \quad (8b)$$

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  $\bar{\omega}_r = [\omega_{r1} \omega_{r2} \dots \omega_{rm}]$ ,  $\bar{v}_r = [v_{r1} v_{r2} \dots v_{rm}]$  là các đỉnh cực trị max-min của  $pHB_r^{(k)}$ .  $\gamma$  là hệ số dốc, ở đây lấy  $\gamma = 0.5$ .



Hình 3. Cấu trúc mạng Neuron-fuzzy

- a/ Cấu trúc mạng Neuron-fuzzy của thuật toán HLM1;
- b/ Cấu trúc mạng Neuron-fuzzy của thuật toán HLM2

- Giá trị liên thuộc của mẫu  $\bar{x}_i$  vào các tập mờ cùng nhãn k,  $k=1 \dots M$  được tính theo Max:

$$\mu_{\bar{B}_i^{(k)}}(\bar{x}_i) = \max \left\{ \mu_{pHB_1^{(k)}}(\bar{x}_i), \dots, \mu_{pHB_r^{(k)}}(\bar{x}_i), \dots, \mu_{pHB_{R_k}^{(k)}}(\bar{x}_i) \right\} \quad (9)$$

$k = 1 \dots M, i = 1 \dots P, r = 1 \dots R_k$

- Dữ liệu ra của mạng ứng với mẫu thứ i:

$$\hat{y}_i = \frac{\sum_{k=1}^M \mu_{\bar{B}_i^{(k)}}(\bar{x}_i) \cdot y_{ki}(\bar{x}_i)}{\sum_{k=1}^M \mu_{\bar{B}_i^{(k)}}(\bar{x}_i)}, \quad (i = 1 \dots P) \quad (10)$$

$$y_{ki} = \sum_{j=1}^n a_j^{(k)} x_{ij} + a_0^{(k)} \quad (11)$$

#### 4.2. Thuật toán huấn luyện mạng thứ nhất, HLM1

HLM1 là thuật toán dùng xác định mạng tối ưu cho một tập mẫu  $T_x$  cho trước trên cơ sở sử dụng các thuật toán Hyperplane Clustering của [1] và thuật toán CSHL được chúng tôi đề xuất trong nghiên cứu này. Do đó, ưu điểm của thuật toán HLM1 là sự kết hợp và phát triển từ các ưu điểm của hai thuật toán này.

Gọi Mmin và Mmax là số luật mờ cực tiểu và cực đại được sử dụng cho khảo sát.

Giá trị khởi tạo: gán  $M=Mmin - 1$ ;



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

$M:=M+1$ . Gọi thuật toán Hyperplane Clustering

Bước 2. Xây dựng tập các siêu hộp thuần chủng pHB: gọi thuật toán CSHL

Bước 3. Xác định sai số theo chuẩn L2

- Tính giá trị liên thuộc theo (8) và (9);
- Tính  $\hat{y}_i$  theo (10) và (11);
- Tính sai số bình phương trung bình

$$E = \frac{1}{P} \sum_{i=1}^P (y_i - \hat{y}_i)^2 \quad (12)$$

Bước 4. Kiểm tra điều kiện dừng

- Nếu  $M < M_{\max}$ , quay lại bước 1.
- Nếu  $M = M_{\max}$ , qua bước 5.

Bước 5. Chọn mạng tối ưu có sai số  $E \leq [E]$  và có M nhỏ.

## 5. THUẬT TOÁN HUẤN LUYỆN MẠNG NEURO-FUZZY THỨ HAI, HLM2

### 5.1. Cấu trúc mạng neuro-fuzzy của HLM2

Cấu trúc mạng neuro-fuzzy của thuật toán HLM2 thể hiện trên hình 3b. Các lớp input và output của mạng này hoàn toàn giống các lớp tương ứng của mạng ở hình 3a của thuật toán HLM1. Sự khác nhau giữa hai mạng thể hiện ở lớp ẩn, trong đó, mạng của thuật toán HLM2 sử dụng hàm Gauss với đường tâm và độ rộng của mỗi đặc tính Gauss được quyết định bởi hai tham số  $\theta_{i1}, \theta_{i2}, i = 1 \dots M$ . Như vậy, nếu sử dụng M luật mờ (1) ta sẽ có 2M tham số  $\theta_{ij}$  đóng vai trò là bộ trọng số W của mạng. Bộ trọng số tối ưu của mạng, ký hiệu Wop, tính theo chuẩn L2 là tập hợp các  $\theta_{ij}$  sao cho hàm tổng bình phương sai số (12) đạt cực tiểu:

$$E = \frac{1}{P} \sum_{i=1}^P (y_i - \hat{y}_i)^2 \rightarrow \min \quad (13)$$

Wop được xác định bằng phương pháp huấn luyện mạng neuron theo những thuật toán quen thuộc. Trong các thí nghiệm kiểm chứng trình bày trong bài báo này chúng tôi sử dụng thuật toán Conjugate Gradient [4] để tìm Wop.

Bộ trọng số Wop có tác dụng đảm bảo việc xác lập một tập các tập mờ tối ưu ở input khi đã có một tập các pHB là kết quả của thuật toán CSHL. Giá trị liên thuộc của mẫu vào  $\bar{x}_i$ ,  $i = 1 \dots P$  vào tập mờ nhân k,  $k = 1 \dots M$  được tính:

$$\mu_{pHB_r^{(k)}}(\bar{x}_i) = e^{-\frac{\sum_{j=1}^n \left[ x_{ij} - \frac{1}{2} \theta_{k1} (\omega_{rj} + v_{rj}) \right]^2}{n(\theta_{k1})^2}} \quad (14)$$

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  $\bar{\omega}_r = [\omega_{r1} \omega_{r2} \dots \omega_{rm}]$ ,  $\bar{v}_r = [v_{r1} v_{r2} \dots v_{rm}]$  là các đỉnh cực trị max-min của  $pHB_r^{(k)}$ .

- Giá trị liên thuộc của mẫu  $\bar{x}_i$  vào các tập mờ cùng nhãn k,  $k=1 \dots M$  được tính theo (9).

- Dữ liệu ra của mạng ứng với mẫu thứ i được tính theo (10) và (11).

## 5.2. Thuật toán huấn luyện mạng neuro-fuzzy, HLM2

HLM2 là thuật toán dùng xác định mạng neuro-fuzzy tối ưu cho một tập mẫu  $T_\Sigma$  cho trước trên cơ sở sử dụng các thuật toán Hyperplane Clustering của [1], thuật toán CSHL, và kỹ thuật giải bài toán cực trị bằng mạng neuron. Cũng như HLM1, ưu điểm của thuật toán HLM1 là sự kết hợp và phát triển từ các ưu điểm của hai thuật toán này. Ngoài ra, bộ trọng số tối ưu Wop có tác dụng đảm bảo việc xác lập một tập các tập mờ tối ưu ở không gian dữ liệu vào khi đã xây dựng được một tập các siêu hộp thuần chủng pHB (là kết quả của thuật toán CSHL). Điều này đã làm làm gia tăng mức độ chính xác của thuật toán HLM2.

Gọi Mmin và Mmax là số luật mờ cực tiểu và cực đại được sử dụng cho khảo sát.

Khởi tạo: gán  $M=Mmin - 1$ ;

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

$M:=M+1$ ; Gọi thuật toán Hyperplanr Clustering.

Bước 2. Xây dựng tập các siêu hộp thuần chủng pHB: gọi thuật toán CSHL;

Bước 3. Xác định các tập mờ tối ưu ở input thông qua bộ trọng số tối ưu Wop bằng cách huấn luyện mạng 3b để cực tiểu hàm sai số (13). Trong đó:

- Tính giá trị liên thuộc theo (14) và (9);

- Tính  $\hat{y}_i$  theo (10) và (11);

Bước 4. Kiểm tra điều kiện dừng

- Nếu  $M < M_{max}$ , quay lại bước 1.

- Nếu  $M = M_{max}$ , qua bước 5.

Bước 5. Chọn mạng tối ưu với bộ trọng số tối ưu Wop có sai số  $E \leq [E]$  và có M nhỏ.

## 6. THÍ NGHIỆM KIỂM CHỨNG

### 6.1. Thí nghiệm 1: sử dụng tập mẫu ngẫu nhiên

Sử dụng tập mẫu tr\_set1 15 mẫu, 3 biến vào một biến ra là những giá trị ngẫu nhiên xác định theo Matlab. Sử dụng thuật toán [1] và hai thuật toán mới, HLM1 (có các hệ số định hướng (5.b) là  $\varepsilon_1 = 0.05$ ;  $\varepsilon_2 = 0.95$ ;  $\Delta = 0.35$ ) và HLM2 để huấn luyện mạng xấp xỉ hàm  $y_1 = f_1(x_1, x_2, x_3)$ .

Kết quả được thể hiện trên bảng 1 cho thấy tốc độ hội tụ của HLM1 và HLM2 cao hơn [1].

Bảng 1



Số luật mờ	Các thuật toán		
	[1]	HLM1	HLM2
M=5	0,02760	0,0213	0,0246
M=6	0,02561	0,0107	$7,1148 \cdot 10^{-4}$
M=7	$2,8576 \cdot 10^{-6}$	$1,0199 \cdot 10^{-7}$	$7,3906 \cdot 10^{-8}$
M=8	$7,0300 \cdot 10^{-6}$	$1,7844 \cdot 10^{-7}$	$8,6461 \cdot 10^{-9}$
M=9	$5,6341 \cdot 10^{-6}$	$4,2997 \cdot 10^{-7}$	$1,1859 \cdot 10^{-7}$

**Bảng 2**

Số luật mờ	Các thuật toán		
	[1]	HLM1	HLM2
M=10	$2,000 \cdot 10^{-3}$	$1,700 \cdot 10^{-3}$	$2,769 \cdot 10^{-4}$
M=20	$25,000 \cdot 10^{-4}$	$1,477 \cdot 10^{-4}$	$1,233 \cdot 10^{-4}$
M=30	$2,099 \cdot 10^{-5}$	$1,704 \cdot 10^{-5}$	$1,669 \cdot 10^{-5}$

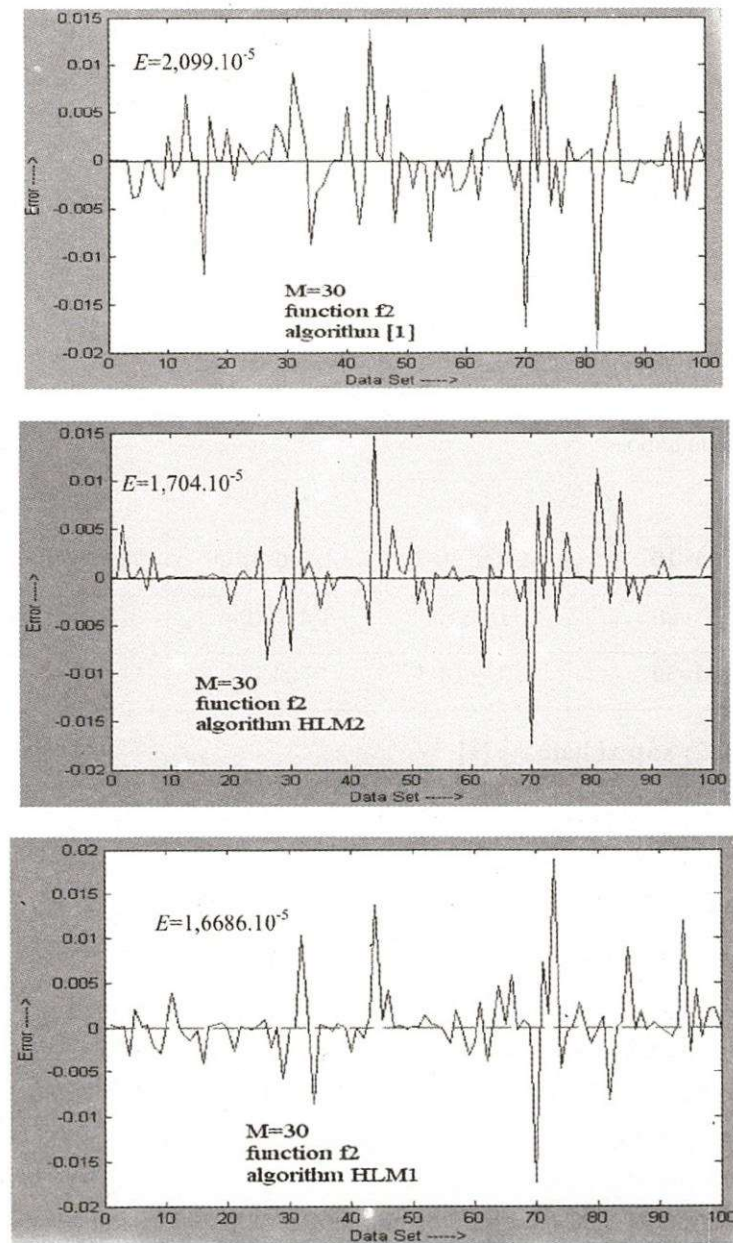
**6.2. Thí nghiệm 2: xấp xỉ hàm  $y_2$  [1]**

Hàm  $y_2 = f_2(x_1, x_2)$  của [1] được sử dụng để xây dựng tập mẫu tr\_set2 gồm 100 mẫu.

$$y_2 = (5 - x_2)^2 / [3(5 - x_1)^2 + (5 - x_2)^2]$$

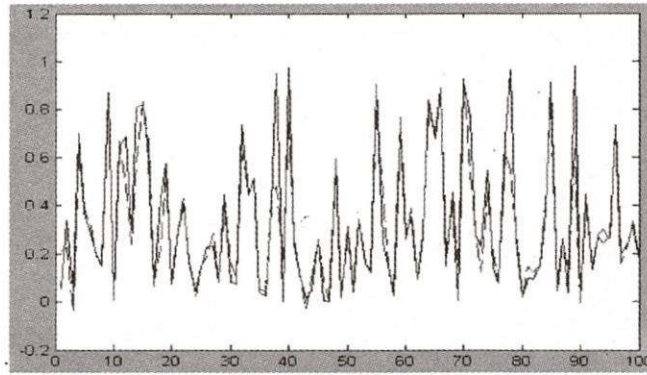
Các giá trị  $\bar{x} = [x_1, x_2]$  được lấy ngẫu nhiên trong khoảng [0,10] nhờ hàm random của Matlab. Dữ liệu ra được tính theo  $y_2 = (5 - x_2)^2 / [3(5 - x_1)^2 + (5 - x_2)^2]$ .

Kết quả khảo sát được thể hiện trên Hình 4, hình 5 và bảng 2. Ở hình 4 thể hiện sai số đáp ứng  $Error_i = y_i - \hat{y}_i$ ,  $i = 1 \dots 100$  và giá trị sai số bình phương trung bình E (12) của thuật toán [1], HLM1 (có các hệ số định hướng  $\epsilon_1 = 0.05$ ;  $\epsilon_2 = 0.95$ ;  $\Delta = 0.35$ ) và HLM2 ứng với số luật mờ M=30. Ở Hình 5, biểu diễn chung trên một hệ trục dữ liệu ra của tập mẫu huấn luyện tr\_set2,  $y_i$   $i = 1 \dots 100$  (nét liền) và tín hiệu ra của mạng  $\hat{y}_i$  (nét đứt) ứng với hai thuật toán [1] và HLM2 với số luật mờ M=20. Trên hình 5a cho thấy sự khác biệt giữa hai đường  $y_i$  và  $\hat{y}_i$ ; ngược lại ở hình 5b, hai đường này gần như trùng nhau, chứng tỏ ở hình 5b giá trị ra của mạng tiệm cận tới giá trị mong muốn. Bảng 2 và hình 5 cho thấy độ chính xác của các thuật toán mới cao hơn độ chính xác của thuật toán [1].

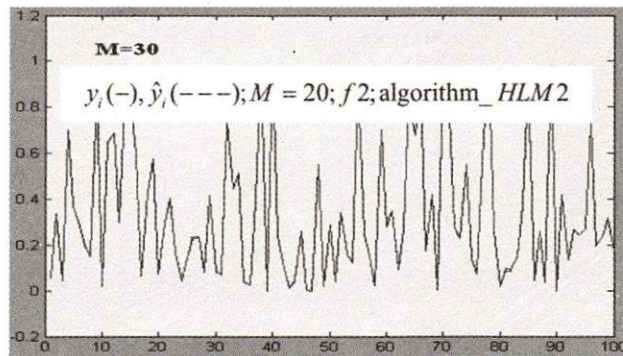


Hình 4. So sánh sai số đáp ứng  $Error_i = y_i - \hat{y}_i, i = 1...100$  và giá trị sai số bình phương trung bình E (12) của thuật toán [1], HLM1 và HLM2 ứng với tập mẫu tr\_set2 với số luật mờ M=30





(a)



(b)

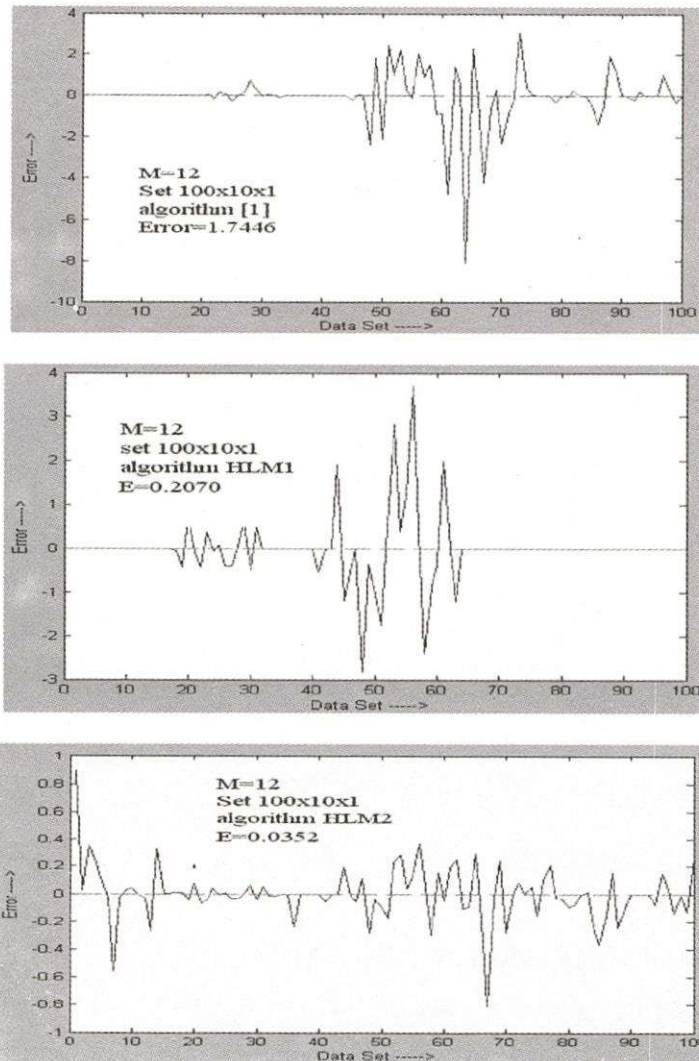
**Hình 5.** Tín hiệu ra của tập mẫu tập  $tr\_set2$   $y_i$  và của mạng  $\hat{y}_i, i = 1 \dots 100$  ứng với hai thuật toán [1] và HLM2

**6.3. Thí nghiệm 3: xấp xỉ hàm từ tập dữ liệu [3]**

Sử dụng tập dữ liệu gồm 100 mẫu, 10 biến vào một biến ra ( $[x_1, x_1, \dots, x_{10}], y$ ) trong phụ lục IV “Daily Data of Stock A” của [3] làm tập huấn luyện mạng cho [1], HLM1 ( $\epsilon_1 = 0.05; \epsilon_2 = 0.95; \Delta = 0.5$ ) và HLM2. Kết quả thể hiện trên hình 6 và bảng 3 cho thấy độ chính xác của HLM2 và HLM1 cao hơn độ chính xác của [1].

**Bảng 3**

Số luật mờ	Các thuật toán		
	[1]	HLM1	HLM2
<b>M=10</b>	13,4148	0,5618	0,0675
<b>M=12</b>	1,74460	0,2070	0,0352
<b>M=14</b>	$3,5028 \cdot 10^{-5}$	$2,1013 \cdot 10^{-6}$	$1,2101 \cdot 10^{-6}$



Hình 6. Giá trị sai lệch  $Error_i = y_i - \hat{y}_i$  và giá trị sai số bình phương trung bình E (12) của các thuật toán [1], HLM1 và HLM2 khi số luật mờ M=12, tập mẫu “Daily Data of Stock A” của [3]

#### 6.4. Thí nghiệm 4

Sử dụng hàm y của [6]:  $y = (1 + x_1^{-2} + x_1^{-1.5})^2$ ,  $x_1, x_2 \in [1, 5]$  để xây dựng tập mẫu gồm 50 mẫu tương tự tập mẫu đã được sử dụng trong [6]. Thực hiện huấn luyện mạng xấp xỉ hàm y với thuật toán HLM1, HLM2, [1] và các thuật toán được trình bày trong [6][8][9] (để đơn giản, các thuật toán này được gọi tắt là [6][8][9]). Các kết quả nhận được cho trong bảng 4 cho thấy độ chính xác trung bình của HLM1 và HLM2 cao hơn rất nhiều so với độ chính xác trung bình của [1][6][8][9].



Bảng 4

Số luật mờ	ĐL	Các thuật toán					
		[6]	[8]	[9]	[1]	HLM1	HLM2
M=6	E	0,0589	0,0572	0,0599	0,0221	0,0182	$0,0196.10^{-2}$
M=8	E	0,0500	0,0499	0,0499	0,0220	0,0218	$0,0185.10^{-2}$
M=10	E	0,0148	0,0149	0,0149	0,0188	$0,0260.10^{-1}$	$0,0198.10^{-2}$

## 7. KẾT LUẬN

Kết quả thí nghiệm cho thấy hiệu quả tác động của hàm phạt  $\tau_j$ . Hàm  $\tau_j$  thông qua bộ tham số định hướng  $[\varepsilon_1; \varepsilon_2; \Delta]$  đóng vai trò định hướng quá trình phân chia không gian dữ liệu để xác lập các tập mờ, 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. Hàm  $\tau_j$  còn 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 HLM1 và HLM2.

Các tập dữ liệu khác nhau sẽ có những đặc điểm phân bố dữ liệu khác nhau. Do đó, khi thay đổi tập dữ liệu, nếu cần tác động vào độ chính xác của phép xấp xỉ ta thay đổi đại lượng  $\varepsilon_1$ ,  $\varepsilon_2$  và  $\Delta$  trong vector  $[\varepsilon_1; \varepsilon_2; \Delta]$  của (5b). Hiện nay chúng tôi đang nghiên cứu quy luật tác động của vector tham số  $[\varepsilon_1; \varepsilon_2; \Delta]$  tới cấu trúc mạng neuro-fuzzy và độ chính xác của phép xấp xỉ, trên cơ sở đó tìm ra phương pháp chung để xác định vector tham số  $[\varepsilon_1; \varepsilon_2; \Delta]$ .

Sai số của HLM2 nhỏ hơn HLM1 tuy nhiên hạn chế cơ bản của HLM2 là thời gian huấn luyện mạng và yêu cầu dung lượng nhớ của máy tính cao hơn nhiều so với sử dụng HLM1.

Phương pháp tổng hợp mạng neuro-fuzzy được đề xuất trên có thể được sử dụng rất hiệu quả trong nhiều lĩnh vực khác nhau: các bài toán về đo lường, nhận dạng, dự báo và điều khiển theo mô hình black-box. Hiện nay chúng tôi đang nghiên cứu ứng dụng phương pháp này cho bài toán nhận dạng động lực học cơ hệ; bài toán xác định vị trí hư hỏng và dự báo mức độ hư hỏng của cầu đường bộ.

## BUILDING NEURO-FUZZY INFERENCE SYSTEMS BASED ON INPUT- OPTIMAL-FUZZY-SET ESTABLISHMENT

Nguyen Sy Dung<sup>(1)</sup>, Ngo Kieu Nhi<sup>(2)</sup>

(1) University of Industry of HoChiMinh City

(2) University of Technology, VNU-HCM

**ABSTRACT:** This study presents an approach for approximation an unknown function  $y = f(\bar{x})$  from a numerical data set based on a neuro-fuzzy inference system modeling. The focus of interest in proposed approach is to increase degree of accuracy of the degree of this approximation. New algorithms named CSHL, HLM1 and HLM2, which are used for this target, are presented. The first new algorithm, CSHL, which uses functions named pure function  $\psi$  and penalty function  $\tau$  effecting as direction for input data space partition, is used to build data clusters. The second and the third algorithm based on the Hyperplane Clustering algorithm of [1] and the CSHL algorithm are used to establish adaptive neuro-fuzzy inference systems. A series of numerical experiments are performed to assess the efficiency of the proposed approach.

### TÀI LIỆU THAM KHẢO

- [1]. Massimo Panella, Antonio Stanislao Gallo. *An Input – Output Clustering Approach to the Synthesis of ANFIS Networks*, IEEE, Transactions on fuzzy systems, Vol. 13, No. 1, February (2005).
- [2]. Massimo Panella, Antonello Rizzi, and Fabio Massimo Frattale Mascioli. *Adaptive Resolution Min-Max Classifier*, IEEE Transactions on Neural Networks, Vol. 13, No. 2, March (2002).
- [3]. M. Sugeno and T. Yasukawa, *A fuzzy logic based approach to qualitative modeling*, IEEE Trans, On Fuzzy Systems, Vol. 1, No. 1, pp. 7-31, Feb (1993).
- [4]. Nguyễn Sỹ Dũng, Lê Hoài Quốc, *Thuật toán thích nghi huấn luyện mạng neuron trên cơ sở phương pháp Conjugate Gradient*, Tạp chí Khoa học và Công nghệ các trường Đại học kỹ thuật, trang 68-73, Số 58/(2006).
- [5]. P. K. Simpson, *Fuzzy min-max neural networks – Part 2: Clustering*, IEEE Trans. Neural Netw , Vol. 1, No. 1, pp. 32-45, (1993).
- [6]. 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, Vol.11, No. 3, June (2003).
- [7]. T. Takagi and M. Sugeno, *Fuzzy identification of systems and applications to modeling and control*, IEEE Trans. Syst. Man, Cybern. , Vol. SMC-15, No. 1, pp. 116-132, Jan. (1985).
- [8]. Wong and C. C. Chen, *A hybrid clustering and gradient decent approach for fuzzy modeling*, IEEE Trans. Syst. Man, Cybern. B, Vol. 29, pp. 686-693, December (1999).
- [9]. 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, Vol. 5, pp. 614-621, Aug (1997).