

Mạng nơron min-max mờ tăng trưởng

A Growing Fuzzy Min-max Neural Network

Lê Anh Tú

Trường Đại học Công nghệ Thông tin và Truyền thông Thái Nguyên

e-Mail: anh.tuc@ngtt.tnu.edu.vn

Vũ Đình Minh

Trường Cao đẳng Công nghiệp Thái Nguyên

e-Mail: vmc802@gmail.com

Tóm tắt

Bài báo này trình bày một mạng nơron được cải tiến từ mô hình mạng nơron phân cụm mờ min-max (FMNN). Mô hình cải tiến được gọi là GFMNN nhằm giải quyết hai hạn chế căn bản của FMNN. Thứ nhất, GFMNN có cấu trúc tự động tăng trưởng số lượng nơron trong quá trình huấn luyện tùy theo đặc tính riêng của mỗi tập dữ liệu. Thứ hai, GFMNN cho phép xử lý trực tiếp các dữ liệu thực mà không cần chuẩn hóa trước. Các thực nghiệm đã được chúng tôi tiến hành trên hai tập dữ liệu PID và Iris để so sánh GFMNN với các phương pháp khác đã được công bố.

Từ khóa: min-max mờ, mạng nơron, phân cụm, phân lớp, tăng trưởng.

Abstract: This paper presents an improved fuzzy min-max neural network (FMNN). The improved model is called Growing FMNN (GFMNN) which aims to overcome the two drawbacks of FMNN. Firstly, GFMNN has growing structure which automatically increases the number of neurons during training depending on specific characteristics of each dataset. Secondly, GFMNN allows directly processing real data without standardizing. The experiments were conducted on two data sets IRIS and PID to compare GFMNN with other methods which have been published previously.

Keywords: Fuzzy min-max, neural network, clustering, classification, growing.

Chữ viết tắt

FMNN	Fuzzy Min-max Neural Network
PARC	Pruning Adaptive Resolution Classifier
GFMNN	Growing Fuzzy Min-max Neural Network
PID	Pima Indians Diabetes
BMU	Best Matching Unit

1. Phần mở đầu

Lý thuyết tập mờ được Zadeh đưa ra vào năm 1965 [1]. Các nghiên cứu về nhận biết và phân loại tập mờ đã được nhiều nhà nghiên cứu thực hiện, điển hình là các nghiên cứu nhằm kết hợp giữa logic mờ và mạng nơron để phát triển các hệ thống thông minh [2,3,4]. Điều này nhằm kết hợp sức mạnh của lý luận mờ

trong xử lý thông tin không chắc chắn với khả năng học của mạng nơron [5,6,7,8]. Lin [9] dựa trên các ý tưởng của điều khiển logic mờ, cấu trúc mạng nơron và khả năng học của mạng nơron để đưa ra mô hình điều khiển logic mờ và hệ thống quyết định. Archer [10], Mukaidono [11] sử dụng tập mờ trong mạng nơron phân lớp.

Dựa trên những ưu điểm của việc kết hợp logic mờ và mạng nơron, Simpson đã đề xuất một mô hình mạng nơron phân cụm mờ min-max, gọi là FMNN [14,15], cho phép kết hợp mạng nơron và lý thuyết min-max mờ để giải quyết bài toán phân lớp và phân cụm. FMNN dựa trên sự tổng hợp của các hyperbox mờ [16] để xác định và giới hạn các không gian con trong không gian mẫu n-chiều. Mỗi không gian con được xác định bằng hai điểm min và max được gọi là một hyperbox. Nói cách khác, FMNN phân cụm bằng cách tạo ra các hyperbox, mỗi hyperbox đại diện cho một cụm sẽ tương ứng là một nơron trong lớp đầu ra. Số lượng nơron của lớp đầu ra được ước lượng cố định trước sao cho lớn hơn số cụm có thể hình thành. Thuật toán học của FMNN nhằm mục tiêu mở rộng và co lại các hyperbox thông qua việc điều chỉnh trọng số của mạng, nếu quá trình mở rộng tạo ra sự chồng lấn giữa các hyperbox thì thực hiện quá trình co lại để khử chồng lấn.

Đến nay, có một số nghiên cứu đề xuất cải tiến hiệu suất thuật toán học của FMNN. Rizzi đã cải tiến FMNN của Simpson bằng cách áp dụng các kỹ thuật phân loại thích ứng [17], cắt tỉa [18], khái quát PARC (Pruning Adaptive Resolution Classifier) [19] sử dụng kỹ thuật cài đặt đệ quy. Tuy nhiên, các giải pháp này có chi phí tính toán lớn, do hạn chế của việc sử dụng đệ quy [18]. Meneganti và nhóm nghiên cứu [20,21] đề xuất thuật toán học bằng cách bắt đầu với số các hyperbox bằng đúng với số hyperbox đầu ra. Thuật toán này phụ thuộc vào thứ tự trình bày các dữ liệu và các kích thước của hyperbox. Các giải pháp học của cả Rizzi và Meneganti là “offline” do phải duyệt lại mẫu huấn luyện nhiều lần để xác định các cụm.

Trong các thuật toán này, đều cần dự báo trước số lượng các hyperbox cần được tạo ra. Việc dự báo trước chính xác các hyperbox là đặc biệt khó đối với loại dữ liệu có độ nhiễu cao, số cụm lớn. Trong bài báo này, chúng tôi đề xuất một mô hình mạng nơron mờ truyền thẳng có khả năng tự tăng trưởng trong quá

trình huấn luyện. Mô hình này được cải tiến từ mô hình của Simpson. Khả năng tự tăng trưởng trong quá trình huấn luyện cho phép tạo ra số lượng hyperbox tương xứng với số lượng cụm dữ liệu thực sự, do đó thuật toán học không cần xác định trước số lượng hyperbox.

Các phần còn lại của bài báo bao gồm: Phần 2 và 3 tổng hợp lại nghiên cứu nền tảng về thuật toán min-max mờ và mô hình mạng nơron FMNN; Phần 4 trình bày mô hình mạng nơron đề xuất của chúng tôi; Phần 5 trình bày các kết quả thực nghiệm và phần cuối cùng là kết luận.

2. Phân cụm mờ min-max

2.1 Khái niệm hyperbox mờ

Cho tập dữ liệu huấn luyện $D = \{A_h | h=1, 2, \dots, m\}$, với $A_h = (a_{h1}, a_{h2}, \dots, a_{hn}) \in R^n$ là mẫu dữ liệu thứ h .

Một hyperbox mờ B_j được định nghĩa [10] như sau:

$$B_j = \{A_h, V_j, W_j, b_j(A_h, V_j, W_j)\}$$

Trong đó:

- $V_j = (v_{j1}, v_{j2}, \dots, v_{jn})$ là điểm min của B_j
- $W_j = (w_{j1}, w_{j2}, \dots, w_{jn})$ là điểm max của B_j
- $b_j(A_h, V_j, W_j)$ là hàm thuộc của mẫu vào A_h với hyperbox B_j

Giá trị hàm thuộc b_j mô tả độ thuộc của mẫu A_h vào hyperbox B_j được nghĩa thông qua V_j và W_j như sau:

$$b_j(A_h, V_j, W_j) = \frac{1}{n} \sum_{i=1}^n [1 - f(a_{hi} - w_{ji}, \gamma) - f(v_{ji} - a_{hi}, \gamma)] \quad (1)$$

trong đó: γ là tham số độ nhạy dùng để điều chỉnh giảm nhanh giá trị hàm thuộc khi một mẫu vào bị tách ra khỏi hyperbox; Hàm ngưỡng hai tham số f được xác định:

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



(a) Hyperbox 2-chiều (b) Hyperbox 3-chiều

H. 1 Minh họa hyperbox mờ

2.2 Thuật toán phân cụm mờ min-max

Với mỗi $A_h \in D$, thuật toán thực hiện 3 bước:

Bước 1. tìm một hyperbox B_j có độ thuộc cao nhất theo (1) trong số các hyperbox thỏa mãn biểu thức điều kiện giới hạn kích thước (2).

$$\sum_{i=1}^n (\max(w_{ji}, a_{hi}) - \min(v_{ji}, a_{hi})) \leq \theta \quad (2)$$

Điều chỉnh các điểm min, max của hyperbox B_j theo (3) và (4).

$$v_{ji}^{new} = \min(v_{ji}^{old}, a_{hi}) \quad \forall i = 1, 2, \dots, n \quad (3)$$

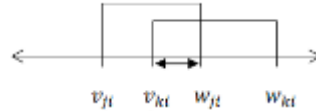
$$w_{ji}^{new} = \max(w_{ji}^{old}, a_{hi}) \quad \forall i = 1, 2, \dots, n \quad (4)$$

Nếu không có hyperbox nào thỏa mãn (2) thì tạo một hyperbox mới và khởi tạo các điểm min, max của nó theo (5):

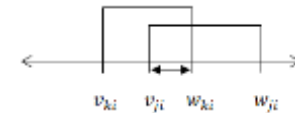
$$V_j = W_j = A_h \quad (5)$$

Bước 2. Kiểm tra chồng lấn giữa hyperbox vừa điều chỉnh B_j với các hyperbox khác. Giả sử hyperbox B_k cần được kiểm tra chồng lấn với B_j . Thực hiện so sánh trên các chiều tương ứng giữa B_j và B_k . Nếu B_j và B_k có chồng lấn, một trong bốn trường hợp sau xảy ra (trên mỗi chiều):

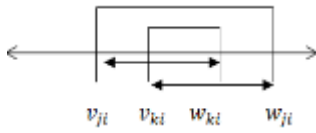
(a) Trường hợp 1: max của B_j chồng lấn với min của B_k : $v_{ji} < v_{ki} < w_{ji} < w_{ki}$



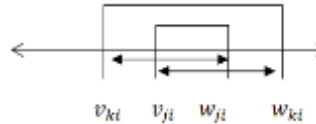
(b) Trường hợp 2: Min của B_j chồng lấn với max của B_k : $v_{ki} < v_{ji} < w_{ki} < w_{ji}$



(c) Trường hợp 3: B_k bị co lại trong B_j : $v_{ji} < v_{ki} < w_{ki} < w_{ji}$



(d) Trường hợp 4: B_j bị co lại trong B_k : $v_{ki} < v_{ji} < w_{ji} < w_{ki}$



Bước 3. Loại bỏ chồng lấn giữa các hyperbox: nếu hyperbox B_j và B_k có chồng lấn, thực hiện điều chỉnh để loại bỏ chồng lấn theo 4 trường hợp tương ứng:

(a) Trường hợp 1:

$$v_{ki}^{new} = w_{ji}^{new} = \frac{v_{ki}^{old} + w_{ji}^{old}}{2} \quad (6)$$

(b) Trường hợp 2:

$$v_{ji}^{new} = w_{ki}^{new} = \frac{v_{ji}^{old} + w_{ki}^{old}}{2} \quad (7)$$

(c) Trường hợp 3:

- Nếu $w_{ki} - v_{ji} < w_{ji} - v_{ki}$:

$$v_{ji}^{new} = w_{ki}^{old} \quad (8)$$

- Nếu $w_{ki} - v_{ji} \geq w_{ji} - v_{ki}$:

$$w_{ji}^{new} = v_{ki}^{old} \quad (9)$$

(d) Trường hợp 4:

- Nếu $w_{ji} - v_{ki} < w_{ki} - v_{ji}$:

$$v_{ki}^{new} = w_{ji}^{old} \quad (10)$$

- Nếu $w_{ji} - v_{ki} \geq w_{ki} - v_{ji}$:

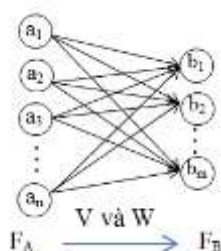
$$w_{ki}^{new} = v_{ji}^{old} \quad (11)$$

Các bước từ 1 đến 3 được lặp lại đối với tất cả các mẫu trong tập dữ liệu. Điều kiện dừng của thuật toán là khi các hyperbox ổn định, tức là tất cả các điểm min và max của các hyperbox không thay đổi hoặc thay đổi dưới một ngưỡng cho phép trong khi duyệt mẫu kế tiếp của tập dữ liệu.

3. Mạng nơ-ron phân cụm mờ min-max

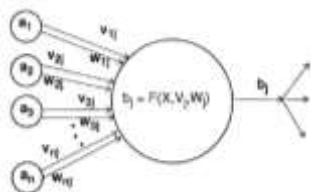
3.1. Cấu trúc mạng

Cấu trúc mạng nơ-ron phân cụm mờ min-max [14] gồm 2 lớp (H. 2), lớp vào F_A gồm n nút (mỗi nút là một chiều của mẫu dữ liệu), lớp ra F_B gồm m nút (mỗi nút là một cụm).



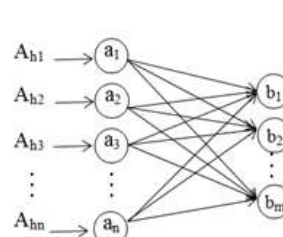
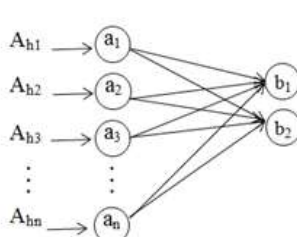
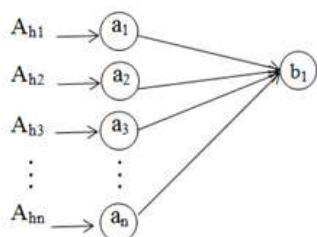
H. 2 Mô hình FMNN

Mỗi một nơ-ron thứ j được kết nối với các đầu vào thông qua một bộ trọng số kép, tương ứng là hai vector V_j và W_j (H. 3).



H. 3 Minh họa cấu tạo của một nơ-ron b_j .

Mỗi đầu vào gắn với một thành phần của A_h . Kết nối giữa đầu vào thứ i và nút thứ j trong F_B thông qua hai trọng số min v_{ij} và max w_{ij} . Tức là các điểm min và max của nút thứ j là các vector $V_j = (v_{1j}, v_{2j}, \dots, v_{nj})$ và $W_j = (w_{1j}, w_{2j}, \dots, w_{nj})$.



(a) Khởi tạo gồm 1 nút ; (b) Tăng trưởng thêm nút mới ; (c) Mạng sau khi đã huấn luyện xong

H.4 Minh họa cấu trúc mạng GFMNN

GFMNN được thiết kế gồm 2 lớp, truyền thẳng tương tự như FMNN. Lớp đầu vào thực chất chỉ bao gồm các tín hiệu vào được nối với lớp nơ-ron đầu ra. Mỗi tín hiệu đầu vào thứ i (với $i=1..n$) được kết nối với nơ-ron b_j thông qua một cặp trọng số kép gồm hai

Với mẫu vào A_h , giá trị ra của $b_j(A_h, V_j, W_j)$ được tính theo (1).

3.2. Thuật toán huấn luyện

Thuật toán huấn luyện của FMNN [14] yêu cầu ước lượng trước số hyperbox có khả năng sẽ được tạo ra. Ký hiệu tập các hyperbox ước lượng ban đầu là U (Uncommitted), là các hyperbox chưa có mẫu vào. Tập các hyperbox có các điểm min, max đã được điều chỉnh là C (Committed), là các hyperbox đã có mẫu vào.

Bước 1: Khởi tạo các trọng số của hyperbox $B_j \in U$ theo (12) và (13):

$$V_j = \underline{1} \quad (12)$$

$$W_j = \underline{0} \quad (13)$$

trong đó, $\underline{1}$ là vector n-chiều có các giá trị bằng 1; $\underline{0}$ là vector n-chiều có các giá trị = 0.

Thuật toán học thực hiện chuyển hyperbox từ tập U đưa vào tập C . Khi một hyperbox được chuyển thì cặp trọng số kép được khởi tạo theo (5).

Bước 2: Mở rộng hyperbox.

Với mỗi mẫu vào A_h , tìm một hyperbox B_j có giá trị hàm thuộc (1) cao nhất. Nếu biểu thức điều kiện (2) thỏa mãn thì điều chỉnh hyperbox theo (3) và (4), ngược lại, chọn một hyperbox $B_k \in U$ và khởi tạo lại giá trị theo (5).

Bước 3: Điều chỉnh chồng lấn nếu có

Kiểm tra chồng lấn giữa hyperbox hiện thời với các hyperbox thuộc tập C . Nếu phát hiện chồng lấn thì loại bỏ chồng lấn theo (6),(7),(8),(9),(10),(11).

4. Mạng nơ-ron phân cụm mờ min-max với cấu trúc tăng trưởng

Để khắc phục hạn chế phải cố định trước số cụm của mô hình FMNN, chúng tôi đề xuất một mô hình mạng nơ-ron tự động tăng trưởng để đạt được số cụm phù hợp với tính chất của mỗi tập dữ liệu. Điều này đúng với bản chất của thuật toán phân cụm mờ min-max. Mô hình của chúng tôi được gọi là Growing FMNN (GFMNN).

4.1. Cấu trúc mạng

thành phần v_{ij} và w_{ij} . Tức là mỗi nơ-ron b_j sẽ có 2 vector trọng số V_j và W_j .

4.2. Thuật toán huấn luyện

Ban đầu GFMMN chỉ có một nút duy nhất b_1 (Hình 4a). Các vector trọng số của b_1 là $V_1(v_{1,1}, v_{2,1}, \dots, v_{n,1})$ và $W_1(w_{1,1}, w_{2,1}, \dots, w_{n,1})$ được khởi tạo bằng một mẫu đầu vào $A_k(a_1, a_2, \dots, a_n)$ bất kỳ.

Trong quá trình huấn luyện, với mỗi mẫu đầu vào A_h , thực hiện:

Bước 1. Tìm noron j khớp nhất (BMU) với A_h bằng cách:

- Xác định một tập N_h gồm các noron thỏa mãn (2).
- Tìm noron $j \in N_h$ có giá trị độ thuộc f_j thỏa mãn (14):

$$f_j = \max_i \{b_i(A_h, V_i, W_i)\} \tag{14}$$

Bước 2. Nếu xác định được noron j thỏa mãn (2) và (14) thì cập nhật các trọng số cho noron j :

- Cập nhật V_j theo (3) và W_j theo (4).
- Điều chỉnh lại V_j và W_j nếu có chồng lấn theo (6), (7), (8), (9), (10), (11).

Ngược lại, nếu không xác định được noron j thỏa mãn (2) và (14) thì tăng trưởng thêm một noron mới cho mạng (Hình 4b, 4c) và khởi tạo các trọng số cho noron mới này bằng mẫu vào A_h .

4.3. Cải tiến hàm thuộc cho phép mạng xử lý các dữ liệu thực

Để GFMMN có thể xử lý được các giá trị thực của tập dữ liệu mà không phải chuẩn hóa về [0,1] như thuật toán gốc của phân cụm mờ min-max và mô hình mạng noron FMNN được đề xuất trước đây chúng tôi đề xuất điều chỉnh lại hàm xác định độ thuộc (1) như sau:

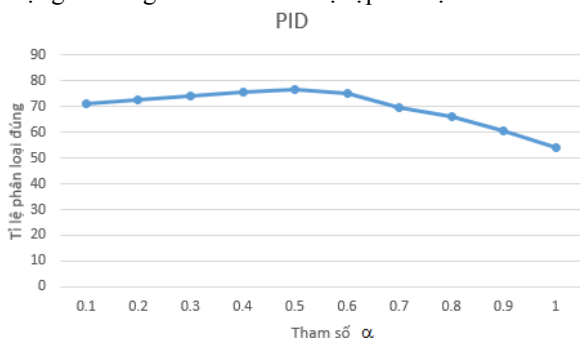
$$b_j(A_h, V_j, W_j) = g \frac{d(V_j, W_j)}{d(A_h, V_j) + d(A_h, W_j)} \tag{15}$$

trong đó, $d(a,b)$ là khoảng cách Euclidean giữa hai vector a và b .

Ngưỡng θ trong biểu thức điều kiện (2) được xác định như sau:

$$\theta = \alpha \times \overline{QE_A} \tag{16}$$

trong đó, $0 < \alpha \leq 1$ là tham số điều chỉnh, $\overline{QE_A}$ là lỗi lượng tử trung bình của toàn bộ tập dữ liệu.



H.5 Kết quả biến động khi thay đổi tham số α

Kết quả phân loại (H.5) tốt nhất đối với tập dữ liệu PID là 76.8% khi $\alpha=0.5$, tập Iris là 94.4% khi $\alpha=0.6$. Bảng 2 dưới đây so sánh kết quả đạt được tốt nhất của GFMMN với các phương pháp khác đã được công bố trước đây căn cứ theo các tài liệu [22, 23].

Công thức (16) cho phép giới hạn kích thước của hyperbox không vượt quá lỗi lượng tử trung bình của tập dữ liệu.

5. Thục nghiệm

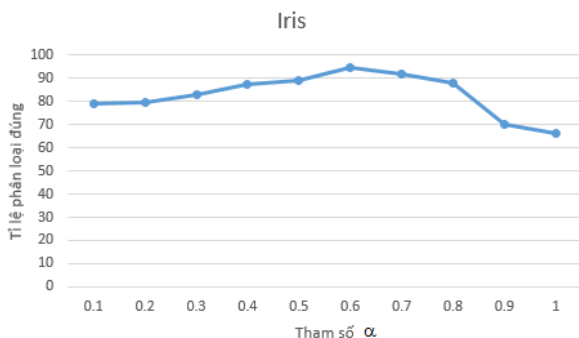
Chúng tôi tiến hành thực nghiệm GFMMN trên hai tập dữ liệu PID và Iris để đánh giá kết quả đạt được so với các giải pháp khác đã công bố trong các tài liệu [22, 23]. PID là tập dữ liệu chứa các thông tin về bệnh tiểu đường [25] gồm 768 mẫu, chia làm 2 cụm thuộc hai trường hợp: 268 mẫu được xác định mắc bệnh tiểu đường (chiếm 35%), các mẫu còn lại là khỏe mạnh. Iris là tập dữ liệu chứa thông tin về 3 loại hoa *Iris setosa*, *Iris virginica* và *Iris versicolor* [25], bao gồm 150 mẫu, mỗi loại hoa có 50 mẫu.

Phương pháp thực nghiệm được tiến hành tương tự với công bố của Anas [24], trong đó: tập PID sử dụng 75% số mẫu để huấn luyện, 25% còn lại để kiểm tra. Tập Iris sử dụng 80% số mẫu để huấn luyện, 20% còn lại để kiểm tra.

Chúng tôi thực hiện thay đổi giá trị của tham số α từ 0.1 đến 1, tham số $\gamma=1$. Kết quả nhận dạng có sự biến động khi tham số α thay đổi. Bảng dưới đây thống kê kết quả hình thành số cụm và tỉ lệ phân loại đúng.

Bảng 1. Thống kê kết quả hình thành số cụm và tỉ lệ phân loại đúng

Tham số α	PID		IRIS	
	Kết quả (%)	Số cụm	Kết quả (%)	Số cụm
0.1	71.1	3	78.9	4
0.2	72.7	3	79.5	4
0.3	74.3	2	83.1	3
0.4	75.9	2	87.7	3
0.5	76.8	2	89.2	3
0.6	75.3	2	94.4	3
0.7	69.8	2	91.8	3
0.8	65.9	2	88.0	3
0.9	60.4	2	70.4	2
1	54.2	2	66.1	2



Nhận thấy rằng, GFMMN có kết quả tương đương so với FMNN đối với cả hai tập dữ liệu PID và Iris. Lý do là GFMMN có thuật toán học hoàn toàn giống FMNN, chỉ khác ở cách thức tăng trưởng tự động

noron. Ngoài ra, GFMNN điều chỉnh lại cách xác định hàm thuộc cho phép xử lý trực tiếp các dữ liệu thực.

Bảng 2. So sánh kết quả thực nghiệm GFMNN với một số phương thức khác.

PID		IRIS	
Phương thức	Kết quả (%)	Phương thức	Kết quả (%)
LDA	77.5	C4.5	91.6
C4.5	73.0	OC1	93.9
CART	72.8	LMDT	95.5
K-NN	71.9	LVQ	92.6
FMNN	76.6	FMNN	94.0
GFMNN	76.8	GFMNN	94.4

6. Kết luận

Bài báo đã trình bày mô hình mạng noron GFMNN với cấu trúc mạng tự động tăng trưởng, cho phép giải quyết bài toán phân cụm mờ dựa trên cách thức thực hiện của thuật toán phân cụm mờ min-max. GFMNN được cải tiến từ mô hình FMNN, ngoài khả năng tăng trưởng số lượng noron động, GFMNN còn được điều chỉnh để có thể xử lý trực tiếp dữ liệu thực. Các kết quả thực nghiệm cho thấy GFMNN có kết quả tốt khi xử lý các dữ liệu thực. So với FMNN thì GFMNN có ưu thế khi xử lý các tập dữ liệu không biết trước số cụm có thể được hình thành, tức là số cụm thực tế sẽ do mạng tự xác định sau quá trình huấn luyện. Đây là tình huống cần thiết khi phân tích một tập dữ liệu mới, cần phải phát hiện ra những đặc trưng tự nhiên của cụm.

Tuy nhiên, để đạt được kết quả phân cụm tốt thì cả hai mô hình GFMNN và FMNN đòi hỏi thời gian và kinh nghiệm bằng việc “thử sai” nhiều lần để xác định các tham số điều chỉnh. Đây cũng là hạn chế của thuật toán phân cụm mờ min-max nói riêng và của hầu hết các mô hình mạng noron nói chung. Ngoài ra, việc xác định kích thước giới hạn chung cho tất cả các cụm (hyperbox) bằng ngưỡng θ là không hợp lý, do thực tế kích thước và mật độ dữ liệu của mỗi cụm dữ liệu trong không gian đầu vào là hoàn toàn khác nhau. Đây cũng là một hướng nghiên cứu tiếp theo cần được xem xét.

Tài liệu tham khảo

[1] L. A. Zadeh, “Fuzzy sets,” *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.
 [2] F. J. Lin and P. H. Shen, “Robust fuzzy neural network sliding-mode control for two-axis motion control system,” *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1209–1225, Aug. 2006.
 [3] R. J. Wai and J. D. Lee, “Adaptive fuzzy-neural-network control for Maglev transportation system,” *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 54–70, Jan. 2008.
 [4] S. Yilmaz and Y. Oysal, “Fuzzy wavelet neural network models for prediction and identification of dynamical systems,” *IEEE Trans. Neural Netw.*, vol. 21, no. 10, pp. 1599–1609, Oct. 2010.

[5] B. Kosko, *Neural Networks and Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
 [6] Z. S. Wang, H. G. Zhang, and W. Yu, “Robust stability of Cohen Grossberg neural networks via state transmission matrix,” *IEEE Trans. Neural Netw.*, vol. 19, no. 5, pp. 169–174, May 2009.
 [7] H. G. Zhang, Z. S. Wang, and D. Liu, “Global asymptotic stability of neural networks with multiple time-varying delays,” *IEEE Trans. Neural Netw.*, vol. 19, no. 5, pp. 855–873, May 2008.
 [8] H. G. Zhang, Y. H. Luo, and D. Liu, “Neural-network-based nearoptimal control for a class of discrete-time affine nonlinear systems with control constraints,”
 [9] C. T. Lin and C. S. G. Lee, “Neural network based fuzzy logic control and decision system,” *IEEE Trans. Comput.*, vol. 40, no. 12, pp. 1320–1336, Dec. 1991.
 [10] N. P. Archer and S. Wang, “Fuzzy set representation of neural network classification boundaries,” *IEEE Trans. Syst., Man Cybern.*, vol. 21, no. 4, pp. 735–742, Jul. 1991.
 [11] R. P. Li, M. Mukaidono, and I. B. Turksen, “A fuzzy neural network for pattern classification and feature selection,” *Fuzzy Sets Syst.*, vol. 130, no. 1, pp. 101–108, Aug. 2002.
 [12] C. T. Lin, C. M. Yeh, S. F. Liang, J. F. Chung, and N. Kumar, “Support-vector-based fuzzy neural network for pattern classification,” *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 1, pp. 31–41, Feb. 2006.
 [13] C. T. Lin and C. S. G. Lee, “Neural network based fuzzy logic control and decision system,” *IEEE Trans. Comput.*, vol. 40, no. 12, pp. 1320–1336, Dec. 1991.
 [14] P. K. Simpson, “Fuzzy min-max neural networks-part I: Classification,” *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 776–786, Sep. 1992.
 [15] P. K. Simpson, “Fuzzy min-max neural network-part II: Clustering,” *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 32–45, Feb. 1993.
 [16] B. Alpern and L. Carter, “The hyperbox,” in *Proc. IEEE Conf. Visual.*, Oct. 1991, pp. 133–139.
 [17] A. Rizzi, M. Panella, and F. M. F. Mascioli, “A recursive algorithm for fuzzy min-max networks,” in *Proc. IEEE/INNS/ENNS Int. Joint Conf. Neural Netw.*, vol. 6, Como, Italy, Jul. 2000, pp. 541–546.
 [18] A. Rizzi, F. M. F. Mascioli, and G. Martinelli, “Generalized min-max classifier,” in *Proc. 9th IEEE Int. Conf. Fuzzy Syst.*, vol. 1, San Antonio, TX, May 2000, pp. 36–41.
 [19] A. Rizzi, M. Panella, and F. M. F. Mascioli, “Adaptive resolution min-max classifiers,” *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 402–414, Mar. 2002.
 [20] M. Meneganti, F. S. Saviello, and R. Tagliaferri, “Fuzzy neural networks for classification and detection of anomalies,” *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 846–861, Sep. 1998.
 [21] R. Tagliaferri, A. Eleuteri, M. Meneganti, and F. Barone, “Fuzzy minmax neural networks: From classification to regression,” *Soft Comput.*, vol. 5, no. 6, pp. 69–76, Feb. 2001.
 [22] Hoang, A., “Supervised Classifier Performance on the UCI Data Set”. M.Sc Thesis, Department of Computer Science, University of Adelaide, Australia. 1997.
 [23] Ster, B. and Dobnikar, A. “Neural Network in medical diagnosis: Comparison with other methods”. In: A. Bulsari et al., eds, *Proc. Int. Conf. EANN’96*. pp. 427430. 1996.

- [24] Anas M. Quteishat and Chee Peng Lim, “A Modified Fuzzy Min-Max Neural Network and Its Application to Fault Classification”, Applied Soft Computing archive Volume 8 Issue 2, Pages 985-995, March, 2008
- [25] David Aha and fellow graduate students at UC Irvine, UCI Center for Machine Learning and Intelligent Systems, <https://archive.ics.uci.edu/ml/datasets.html>.



Le Anh Tu was born in 1980. He received the M.Sc degrees in Computer Science from Thai Nguyen University of Viet Nam, in 2007. At the moment, he is studying for Ph.D degrees in Information Technology Institute - Vietnam Academy of Science and Technology.

He has been working at Thai Nguyen University of Information and Communication Technology (ICTU) since 2004. The main fields of his research are artificial neuron network and multimedia data processing.



Vu Dinh Minh was born in 1976. He received the M.Sc degrees in Computer Science from Thai Nguyen University of Viet Nam, in 2005. At the moment, he is studying for Ph.D degrees in Information Technology Institute - Academy of Military Science and Technology.

He has been working at Thai Nguyen Industrial College (TNIC) since 2001. The main fields of his research are data mining.