

VÀI CÔNG NGHỆ MỚI: LOGIC MỜ VÀ NEURON MỜ TRONG CÁC HỆ THỐNG ĐIỀU KHIỂN

Nguyễn Thiện Thành

Bộ môn điều Khiển Tự Động

Khoa Điện – Điện Tử Trường Đại Học Kỹ Thuật

(Nhận được ngày 23/05/1998)

Tóm tắt:

Bài báo này mô tả hệ thống điều khiển thông minh được trang bị bằng các công nghệ mới về xử lý thông minh như logic mờ và neuron mờ. Bài báo được xây dựng trên nguyên lý cơ bản như thuyết logic mờ, lý thuyết mạng neuron nhân tạo, và sau đó chúng được kết hợp lại để cùng tham gia khả năng cải tiến thông minh của các hệ thống điều khiển trong các môi trường không chắc chắn, không chính xác và nhiễu. Muốn có các hệ thống điều khiển làm việc trong các môi trường như vậy, các hệ thống phải được xây dựng bằng bốn thành phần cơ bản: chiến lược hóa mờ, cơ sở trí thức mờ, suy diễn mờ và chiến lược giải mờ. Những thành phần cơ bản này được kết hợp với nguyên lý cơ bản của các mạng neuron nhân tạo để tạo ra các hệ thống neuron mờ được huấn luyện sử dụng các chế độ học khác nhau.

I) Giới Thiệu Tổng Quát:

Những năm đầu của thập kỷ 90, một ngành điều khiển tự động được phát triển rất mạnh và đã đem lại thành tựu bất ngờ trong lĩnh vực điều khiển, đó là điều khiển mờ và điều khiển neuron mờ. Logic mờ và neuron mờ là những vấn đề thời sự thu hút nhiều sự quan tâm của các nhà chuyên môn. Đó là những phương pháp mới về xử lý thông tin trong lĩnh vực khoa học trí tuệ nhân tạo. Tập mờ và logic mờ của nó là một loại phương pháp xử lý thông tin không chính xác được dựa trên cơ sở phương pháp xử lý thông tin của bộ não con người đã làm được, trong khi đó các mạng neuron được mô hình hóa sau kiến trúc vật lý của bộ não người. Logic mờ và mạng neuron nhân tạo là hai lĩnh vực khoa học có những điểm giống nhau và cũng có những điểm khác nhau đang thu hút rất nhiều nhà nghiên cứu chuyên môn thăm dò những ứng dụng riêng biệt của chúng và sự tổ hợp của chúng trong các hệ thống thông minh thực sự cho nhiều ứng dụng khác nhau. Các sản phẩm tiêu dùng công nghiệp ngày hôm nay mà ta có thể thường gặp như máy hút bụi, máy giặt, máy ảnh thông minh v.v. là những sản phẩm mà đã được trang bị bằng các thứ công nghệ mới về xử lý thông tin này. Phải nói rằng logic mờ, mạng neuron, và sự tổ hợp của chúng đã giải quyết được nhiều vấn đề mà trước đây các thứ công nghệ thông tin cổ điển không thể giải quyết được và vì vậy chúng đã vươn lên đứng vị trí hàng đầu của các thứ công nghệ về xử lý thông tin ngày hôm nay và sau năm 2000.

Thật vậy, lý thuyết logic mờ đã được khám phá năm 1965, do giáo sư Lofti A.Zadeh tại trường đại học California, Berkeley, là phương pháp toán để biểu diễn mờ trong các ngôn ngữ, nó bắt nguồn từ việc khảo sát tổng quát của lý thuyết tập hợp crisp.

Trong tập Crisp, mỗi phần tử của không gian có thể phụ thuộc hoặc không phụ thuộc vào một tập hợp nào đó. Điều đó có nghĩa là mỗi thành viên trong không gian có thể thuộc về một lớp (crisp) hoặc không thuộc về một lớp, nếu thành viên thuộc về lớp thì nó lấy giá trị đúng tương ứng với 1, mặt khác nếu thành viên không thuộc về lớp thì nó lấy giá trị sai tương ứng với 0. Như vậy tập hợp crisp được định nghĩa bằng hàm đặc tính của nó bằng cách ánh xạ mỗi phần tử từ một không gian phát biểu đến một không gian phạm vi 0 và 1.

Tập mờ là một sự tổng quát hóa của tập hợp crisp trong đó nó cho phép mức thành viên của mỗi phần tử có thể lấy bất kỳ giá trị nào giữa 0 và 1. Vậy thì, hàm thành viên của một tập mờ ánh xạ mỗi phần tử của một không gian phát biểu đến một không gian phạm vi của nó trong khoảng [0,1].

Từ năm 1985, tập mờ và sự kết hợp logic mờ của nó đã được sử dụng rộng rãi trong nhiều ứng dụng: quá trình điều khiển, nhận dạng và phân lớp, quản trị và làm ra các quyết định v.v.

II) Nguyên Lý Cơ Bản Về Lý Thuyết Logic Mờ :

Để làm sáng tỏ nguyên lý cơ bản về logic mờ, chúng ta nhìn lại nguyên lý cơ bản về lý thuyết tập hợp crisp và logic cổ điển của nó. Lý thuyết tập hợp crisp là một sự phân loại theo hai tổ hợp: thành viên và không thành viên được định nghĩa bằng hàm đặc tính của nó. Cho phép A là một tập hợp trong không gian phát biểu U, x là các phần tử của U. A được gọi là một tập hợp crisp trong U nếu A được định nghĩa bằng hàm đặc tính $X_A(x)$ của nó sao cho x thuộc về A, hàm $X_A(x)$ lấy giá trị 1 và x không thuộc về A, hàm $X_A(x)$ lấy giá trị 0. Điều đó có nghĩa là

$$X_A(x) = \begin{cases} 1 & \text{nếu và chỉ nếu } x \in A \\ 0 & \text{nếu và chỉ nếu } x \notin A \end{cases}$$

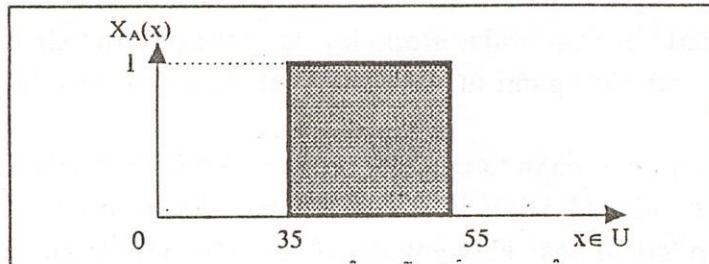
Có nhiều tính chất, chúng ta đã gán đến người và vật không thích hợp cho lắm đối với khung làm việc của lý thuyết tập hợp crisp, cụ thể như xét một vài ví dụ đơn giản, đó là sự phân loại theo ba tổ hợp tuổi: 'Trẻ', 'Trung niên' và 'Già' được định nghĩa như sau:

Trẻ	< 35 tuổi
Trung niên	35 → 55 tuổi
Già	≥ 56 tuổi

Cho phép A là một tập hợp crisp trong không gian phát biểu tuổi U, biểu diễn tổ hợp tuổi trung niên mà được định nghĩa bằng hàm đặc tính của nó:

$$X_A(x) = \begin{cases} 1 & 35 \leq x \leq 55 \text{ cho } x \in A \\ 0 & \text{mặt khác cho } x \notin A \end{cases}$$

và được biểu diễn ở hình 1



Hình 1: tập hợp Crisp A biểu diễn tổ hợp tuổi trung niên

Sự phân loại này dường như không thích hợp đối với khung làm việc của lý thuyết tập hợp crisp cụ thể như một người 55 tuổi thuộc thành viên của tập trung niên trong khi đó một người khác 56 tuổi thì không thuộc thành viên của tập trung niên mà thuộc thành viên của tập già. Một sự phân loại như thế là hơi bất hợp lý. Tại sao một thành viên 55 tuổi thì được coi như là trung niên trong khi đó một thành viên 56 tuổi không được coi là trung niên mà lại coi như là già. Tuy nhiên, lý thuyết tập hợp crisp và logic cổ điển của nó đã được sử dụng trong một thời gian khá dài để mô hình hóa một thế giới hiện thực như “đen” và “trắng”, “đúng” và “sai”, và “0” và “1”. Hai giá trị logic cổ điển này đã được chứng minh rất thành công và có nhiều hiệu quả trong cách giải quyết được nhiều vấn đề. Tuy nhiên, hai giá trị logic cổ điển này chỉ có thể xác nhận thành viên chắc chắn thuộc về một tập hợp hoặc không chắc chắn thuộc về một tập hợp. Ưu điểm của tập crisp và logic cổ điển của nó là sở hữu được những suy diễn chính xác, tuy nhiên nhược điểm của nó là mô tả một thế giới hiện thực không chỉ có “đen và trắng”, “đúng và sai”, và “0 và 1”. Hay nói một cách khác, nó chỉ mô tả một thế giới hiện thực chỉ gói trọn trong hai chữ số 0 và 1 mà đã được ứng dụng trong các loại máy tính cổ điển đang được dùng ngày hôm nay có chức năng tính toán chính xác. Trong thực tế, sự mô tả này không hoàn toàn giống như sự mô tả của bộ não con người về một thế giới hiện thực cụ thể như mô tả về màu đen của một đối tượng thì chỉ có đen chứ không xác định được mức đen của nó trong một tập hợp của các đối tượng có màu đen, trong khi đó, sự mô tả của bộ não con người về màu đen của một đối tượng trong một tập hợp của các đối tượng có màu đen là đen nhiều hay ít; hay không quá đen so với các đối tượng có màu đen khác.

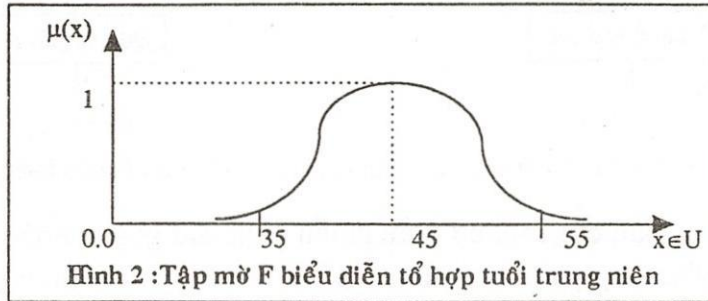
Để có một sự mô tả về thế giới hiện thực như sự mô tả của bộ óc con người nói trên, có một loại tập hợp được phát triển từ tập hợp crisp, nó được định nghĩa bằng hàm thành viên của nó sao cho hàm có thể lấy bất kỳ giá trị nào giữa 0 và 1, loại tập hợp đó được gọi là tập hợp mờ. Cho phép F là một tập hợp trong một không gian phát biểu U , và x là các phần tử của U . F được gọi là một tập hợp mờ trong U nếu F được định nghĩa bằng hàm thành viên $\mu_F(x)$; lấy bất kỳ giá trị nào giữa 0 và 1 mà không bị giới hạn chỉ có hai giá trị 0 và 1. Vậy tập hợp mờ F có thể được định nghĩa bởi:

$$F = \{(x, \mu_F(x)) \mid x \in U\}$$

trong đó, x là biến thành viên thuộc về tập hợp mờ F và $\mu_F(x)$ là giá trị mức của thành viên x trong F , để đánh giá mức độ phụ thuộc của thành viên x trong tập mờ F .

Khảo sát ví dụ đơn giản về sự phân loại người theo ba tổ hợp tuổi như đã được định nghĩa nói trên. Cho F là một tập mờ biểu diễn tổ hợp trung niên trong không gian

tuổi U được định nghĩa bằng hàm thành viên $\mu_F(x)$ của nó sao cho tất cả những thành viên tròn tuổi 45, hàm $\mu_F(x)$ lấy giá trị 1 để đánh giá mức phụ thuộc của chúng trong tập mờ F , từ đó hàm sẽ giảm dần giá trị mức phụ thuộc của các thành viên khác trong tập mờ F xuống đến mức 0 cho những thành viên có mức tuổi dưới 35 và trên 55, được biểu diễn ở hình 2:



- Biến ngôn ngữ và các giá trị của biến ngôn ngữ:

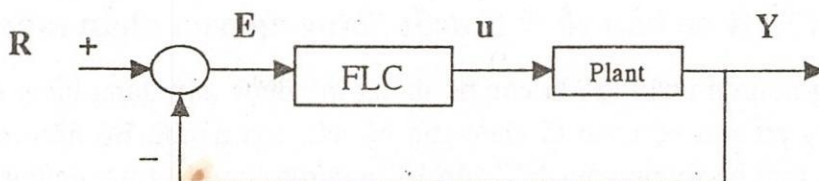
Biến ngôn ngữ là một khái niệm rất quan trọng trong logic mờ và lý giải xấp xỉ trong nhiều ứng dụng khác nhau của nó cụ thể như trong các hệ chuyên gia mờ và các hệ thống điều khiển logic mờ. Biến ngôn ngữ là biến mà các giá trị của nó là những lời nói hoặc những câu dưới dạng ngôn ngữ tự nhiên hoặc ngôn ngữ nhân tạo. Khảo sát một vài ví dụ đơn giản về tốc độ của xe: Tốc độ của xe thường được mô tả bằng các lời nói hoặc các câu dưới dạng ngôn ngữ như: “rất chậm”, “chậm”, “nhanh” và “rất nhanh”, và các giá trị ngôn ngữ như: “rất chậm”, “chậm”, “nhanh” và “rất nhanh” được xem như là những biến mờ trong các tập hợp mờ biểu diễn “rất chậm”, “chậm”, “nhanh” và “rất nhanh”. Nói tóm lại, biến ngôn ngữ là biến mà các giá trị của nó là các biến mờ.

Khi các tập mờ đã được định nghĩa, chúng có thể được sử dụng để xây dựng các qui tắc cho các hệ chuyên gia mờ và thực hiện suy diễn mờ.

Thông thường tri thức trong lý giải mờ được biểu diễn dưới dạng các qui tắc: “If x is A , then y is B ”, trong đó x và y là các biến mờ, và A và V là các giá trị mờ. Như vậy, qui tắc gồm có hai vế: vế trái là tiền điều kiện và vế phải là kết quả. Những phát biểu tiền điều kiện trong vế trái của các qui tắc được kết nối nhau thông qua các toán tử logic mờ như AND và OR.

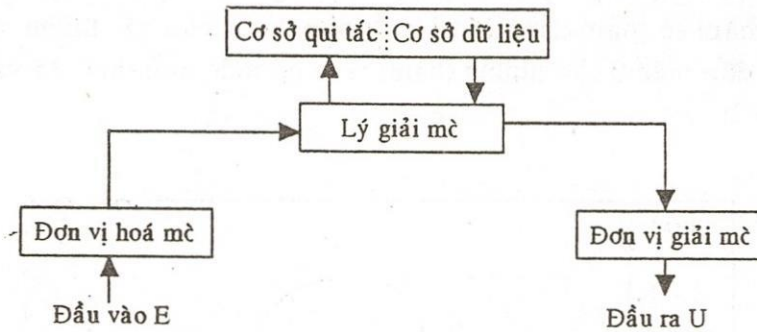
III) Các Hệ Thống Điều Khiển Logic Mờ:

Một hệ thống điều khiển logic mờ được mô tả như ở hình 3:



Hình 3: Chi Hệ Thống Điều Khiển logic Mờ

trong đó, FLC là bộ điều khiển mờ được mô tả ở hình 4:



Hình 4 Chỉ Cấu Trúc Cơ Bản Của Một Bộ Điều Khiển logic Mờ

Cấu trúc cơ bản của một bộ điều khiển logic mờ gồm có bốn thành phần chính: đơn vị hóa mờ, đơn vị lý giải logic mờ, cơ sở tri thức mờ, và đơn vị giải mờ. Tùy thuộc các mục tiêu thiết kế, các loại bộ điều khiển logic mờ khác nhau có thể được xây dựng cụ thể như: Bộ điều khiển logic mờ tĩnh FLC được xây dựng bằng cơ sở tri thức mờ không có khả năng học, hoặc bộ điều khiển logic mờ động FLC được xây dựng bằng một cơ sở tri thức mờ có khả năng học.

Cơ sở tri thức mờ chứa hai loại thông tin chính: cơ sở dữ liệu để định nghĩa các hàm thành viên của các tập mờ được sử dụng như các giá trị cho mỗi biến hệ thống và cơ sở qui tắc chủ yếu để ánh xạ các giá trị mờ của các đầu vào đến các giá trị của các đầu ra. Biến hệ thống gồm có hai loại: biến đầu vào E được đo từ quá trình điều khiển được và biến đầu ra U được sử dụng bằng bộ điều khiển FLC để điều khiển quá trình. Các giá trị thực sự được thu thập từ hoặc được gửi đến quá trình điều khiển được thường ở dạng crisp, và các đơn vị hóa mờ và giải mờ các chức chuyển đổi từ dạng crisp sang dạng mờ và từ mờ sang crisp.

Cơ sở qui tắc mờ gồm có tập hợp của các qui tắc điều khiển mờ dưới dạng If - Then cụ thể như khảo sát một bộ điều khiển mờ FLC với một đầu vào và một đầu ra, qui tắc mờ thứ i thường được biểu hiện bằng If x is PB Then y is NB, trong đó x là biến đầu vào, y là biến đầu ra, PB là một của các tập mờ được định nghĩa cho x trong không gian phát biểu X, và NB là một của các tập mờ được định nghĩa cho biến y trong không gian phát biểu Y.

Cơ sở tri thức mờ có thể ở dạng tĩnh hoặc dạng động. Cơ sở tri thức mờ động có khả năng học trong khi đó cơ sở tri thức mờ tĩnh không có khả năng học hoặc tự tổ chức bằng bộ điều khiển mờ FLC.

Lý giải logic mờ chính nó sử dụng logic mờ để làm ra quyết định giống như quyết định làm ra của con người. Nó thực hiện suy diễn mờ để đi đến tác động điều khiển mờ bằng cách tìm kiếm cơ sở tri thức cho các đầu vào đã được hóa mờ.

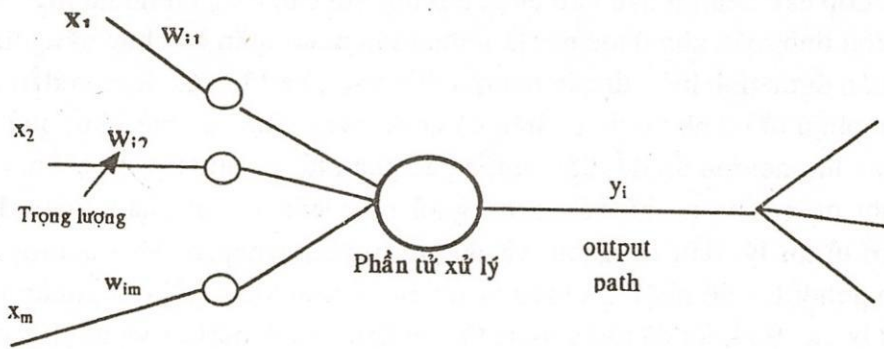
IV) Nguyên lý cơ bản về lý thuyết mạng neuron nhân tạo:

Mạng neuron nhân tạo là các hệ thống mà được xây dựng bằng cách sao chép lại nguyên lý cơ bản về cách tổ chức của bộ não con người. Bộ não con người được thừa nhận là một hệ thống gồm 10^{10} đến 10^{12} neurons có cấu trúc, những neuron có thể được tổ chức theo ba lớp: lớp các neuron đầu vào, lớp các neuron tính toán và lớp các

neuron đầu ra. Lớp các neuron đầu vào được kết nối với các cảm biến như mắt, mũi, tai và da, lớp neuron tính toán còn được gọi là những lớp neuron ẩn có chức năng tính toán để xử lý khi nhận được tín hiệu từ các neuron đầu vào, và lớp các neuron đầu ra được kết nối với các phần tử cơ như tay và chân có chức năng nhận tín hiệu kết quả đã tính toán được từ các lớp neuron ẩn để điều khiển các phần tử cơ như tay và chân. Cấu trúc cơ bản của một neuron trong bộ não con người gồm bốn thành phần: các đầu vào dendrites, phần tử xử lý, đầu ra Axon, và đường truyền synapse. Mỗi neuron này có nhiều đầu vào dendrites để nhận tín hiệu từ nhiều neuron khác gửi đến, phần tử xử lý neuron này xử lý các tín hiệu đã nhận được từ các đầu vào dendrites và gửi kết quả đến đầu ra axon. Mỗi neuron có rất nhiều đầu vào dendrites nhưng chỉ có một đầu ra axon, đường truyền synapse có chức năng kết nối một đầu vào của neuron này với đầu ra neuron khác và đầu ra của mỗi neuron khác và đầu ra của mỗi neuron có thể được kết nối với nhiều neuron đầu vào khác. Mỗi phần tử xử lý neuron gồm có hai thành phần: tính toán và giá trị ngưỡng, tính toán có chức năng tính lấy giá trị tổng của các tín hiệu đã nhận được từ các đầu vào dendrites, so sánh với giá trị ngưỡng để đưa ra quyết định gửi tín hiệu hoặc không gửi tín hiệu đến những neuron khác nếu giá trị tổng lớn hơn hoặc bằng giá trị ngưỡng và ngược lại. Tín hiệu được truyền từ neuron này đến neuron khác dưới dạng điện áp, nếu điện áp dương đó là loại tín hiệu kích động neuron gửi tín hiệu đến những neuron khác và nếu điện áp âm đó là loại tín hiệu ức chế gửi tín hiệu đi. Chất lượng kết nối giữa các neuron làm gia tăng tín hiệu hoặc suy giảm tín hiệu trên đường truyền synapses được gọi là cường độ kết nối. Giá trị của cường độ kết nối này được gọi là hệ số trọng lượng trên đường truyền synapse. Khi hệ thống neuron nhận thấy một đối tượng từ các cảm biến như mắt, nó sẽ huấn luyện, cường độ bắn tín hiệu đi của các neuron ẩn được gia tăng theo trọng thời gian hệ thống tiếp xúc đối tượng để phản ánh toàn bộ thông tin về đối tượng đó. Nếu nó nhận bắt thấy một đối tượng khác xuất hiện, thì hệ thống sẽ bắt đầu huấn luyện lại để cường độ bắn tín hiệu của các neuron được gia tăng theo trọng thời gian tiếp xúc đối tượng đó để phản ánh toàn bộ thông tin về đối tượng đó. Nếu hệ thống có sự chẩn đoán sai, nó sẽ tự điều chỉnh bằng cách huấn luyện lại sử dụng các chế độ học khác nhau để có một chẩn đoán đúng.

Để có một mạng neuron nhân tạo bằng cách sao chép tất cả các nguyên lý cơ bản về cách tổ chức của bộ não con người nói trên sao cho hệ thống làm việc thông suốt như một hệ thống neuron con người. Như vậy, một mạng neuron nhân tạo cũng gồm có: các neuron đầu vào để kết nối với các cảm biến như television cameras và earphones, các neuron ẩn để tính toán cụ thể như các máy tính nhỏ, và các neuron đầu ra để kết nối với các thiết bị phần cơ như tay và chân robo.

Khảo sát mô hình đơn giản của một mạng neuron nhân tạo được đề xuất bởi McCulloch và Pitts vào năm 1943, thường được gọi là mạng neuron M-P được chỉ ở hình 5.



Hình 5: Chỉ mô hình đơn giản của một mạng neuron nhân tạo

Trong mô hình này, phần tử xử lý thứ i tính tổng hệ số trọng lượng lấy được từ các đầu vào x_i , và các đầu ra $y_i = 1$ (bắt tín hiệu) hoặc $y_i = 0$ (không bắt tín hiệu) tuân theo tổng hệ số trọng lượng lấy được từ các đầu vào trên hoặc dưới giá trị ngưỡng θ_i .

$$y_i(t+1) = a \left(\sum_{j=1}^m w_{ij} x_j(t) - \theta_i \right)$$

Trong đó, hàm tác động $a(f)$ là hàm unit step: $a(f) = \begin{cases} 1 & \text{nếu } f \geq 0 \\ 0 & \text{nếu } f < 0 \end{cases}$

Trọng lượng w_{ij} biểu diễn cường độ của synapse để kết nối neuron thứ j với neuron thứ i . Hệ số trọng lượng dương tương ứng với một sự kích động synapse và hệ số trọng lượng âm tương ứng với một sự ức chế synapse, và nếu $w_{ij} = 0$ thì không có sự kết nối giữa hai neuron. Thân neuron là phần tử xử lý kết nối với đầu vào và đầu ra. Kết hợp với đầu vào của phần tử xử lý là một hàm tổng hợp f có chức năng tổ hợp thông tin, tương tác, hoặc bằng chứng từ nguồn bên ngoài hoặc các phần tử xử lý khác đến với phần tử xử lý. Hàm tổ hợp này thường là hàm tuyến tính với các đầu vào x_j như mô tả trên:

$$f_i = \sum_{j=1}^m w_{ij} x_j - \theta_i$$

Trong đó θ_i là ngưỡng của phần tử xử lý thứ i .

Thành phần quan trọng thứ ba của mạng neuron là phương pháp huấn luyện sử dụng các qui tắc học khác nhau. Có hai loại học trong các mạng neuron nhân tạo: Học thông số và học cấu trúc. Học thông số là phương pháp học bằng cách cập nhật các hệ số trọng lượng kết nối và học cấu trúc là phương pháp học bằng cách thay đổi bên trong cấu trúc mạng bao gồm số các phần tử xử lý và các kiểu kết nối mạng. Học cấu trúc sẽ được thăm dò về sau này và ở đây chúng ta chỉ trình bày các chế độ học thông số.

Giả sử có n phần tử xử lý trong một mạng neuron nhân tạo và mỗi phần tử xử lý có chính xác m hệ số trọng lượng thích nghi. Ma trận hệ số trọng lượng W được định nghĩa bởi:

$$W = \begin{bmatrix} W_1^T \\ W_2^T \\ \vdots \\ W_n^T \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & \dots & W_{1m} \\ W_{21} & W_{22} & \dots & W_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ W_{n1} & W_{n2} & \dots & W_{nm} \end{bmatrix}$$

Trong đó $W_i = (w_{i1}, w_{i2}, \dots, w_{im})^T$, $i = 1, 2, \dots, n$ là vector hệ số trọng lượng của phần tử xử lý thứ j đến phần tử xử lý thứ i .

Trong quá trình xử lý thông tin, học thông số được sử dụng để huấn luyện mạng nhằm tìm ra một ma trận hệ số trọng lượng W thích hợp cho sự kết nối các phần tử xử lý của mạng. Trong sự kết nối này, các hệ số trọng lượng được coi như vùng nhớ dài hạn LTM (Long – term – memory) và các trạng thái tác động của các neuron được coi như vùng nhớ ngắn hạn STM (Short – Term – Memory). Học thông số được chia làm ba chế độ học: học giám sát (supervised learning), học củng cố (reinforcement learning) và học không giám sát (unsupervised learning).

Học giám sát là chế độ học có thầy giáo giám sát để bảo mạng chính xác cái gì nó nên phát ra tại đầu ra. trong chế độ học giám sát, mạng được cung cấp bằng các cặp đầu vào – đầu ra mong muốn được : $(x^{(1)}, d^{(1)}, (x^{(2)}, d^{(2)}), \dots, (x^{(k)}, d^{(k)})$. Khi mỗi đầu vào $x^{(k)}$ được đặt vào mạng, đầu ra mong muốn được tương ứng $d^{(k)}$ được đo trong bộ phát sinh tín hiệu sai số (error signal generator) cái mà sau đó sản xuất ra các tín hiệu sai số cho mạng để điều chỉnh các hệ số trọng lượng của nó sao cho đầu ra thật sự sẽ tiến gần đến đầu ra mong muốn được.

Học củng cố là chế độ học giám sát, các giá trị đầu ra được biết cho mọi đầu vào. Tuy nhiên trong một vài tình huống thiếu thông tin chi tiết có sẵn, cụ thể như mạng neuron chỉ được bảo rằng đầu ra thật sự hiện có của nó là quá cao hoặc chỉ đúng có 50 phần trăm. Trong trường hợp khắc nghiệt này. chỉ có một Bit thông tin hồi tiếp để cho biết đầu ra có giá trị đúng hay sai. học dựa trên cơ sở loại thông tin đánh giá này được gọi là học củng cố và thông tin hồi tiếp được gọi là tín hiệu củng cố (reinforcement signal).

Học không giám sát là chế độ học mà mạng không có thầy giáo cung cấp cho bất kỳ thông tin hồi tiếp nào. Không có hồi tiếp từ môi trường để bảo cái gì các đầu ra nên hoặc không nên chúng được điều chỉnh đúng. Mạng phải khám phá cho các mẫu chính nó, các đặc trưng, các quan hệ, hoặc các loại trong dữ liệu nhập và mã hóa chúng ở đầu ra. Trong thời gian đang khám phá các đặc trưng này, mạng đang trải qua nhiều sự thay đổi trong các thông số của nó; Quá trình này được gọi là tự tổ chức.

Ba chế độ học thông số đã được tham khảo , kế theo đó chúng ta khảo sát một cấu trúc huấn luyện tổng quát cho một phần tử xử lý trong một mạng như được chỉ ở hình trên.

Trong hình trên, đầu vào x_j , $j = 1, 2, \dots, m$, có thể là một đầu ra từ một phần tử xử lý khác hoặc có thể là một đầu vào bên ngoài. Tín hiệu mong muốn được d_j chỉ có sẵn trong chế độ học giám sát hoặc trong chế độ học củng cố, trong đó d_j là tín hiệu củng cố (signal inforcement). Vì vậy, hai chế độ học này, các hệ số trọng lượng của phần tử thứ i được cải tiến tuân theo tín hiệu đầu vào nó nhận được, giá trị đầu ra của nó, là sự

đáp ứng dạy đã kết hợp được. Tuy nhiên chế độ học không giám sát, phần xử lý cải tiến các hệ số trọng lượng của nó chỉ được dựa trên cơ sở các giá trị đầu vào và / hoặc các giá trị đầu ra.

Dạng tổng quát của qui tắc học hệ số trọng lượng trong các mạng neuron cho biết rằng sự gia tăng của vector hệ số trọng lượng w_i , vector hệ số trọng lượng w_i được sản xuất bằng bước học tại thời điểm t thì tỉ lệ với tích của tín hiệu học r và đầu vào $X(t)$:

$$\Delta w_i(t) = \eta r X(t)$$

Trong đó η là một số dương được gọi hằng số học để xác định tốc độ học và r là tín hiệu học, nếu có sẵn một tín hiệu của thầy giáo d_i ; thì tín hiệu học là một hàm công nghệ tổng quát:

$$r = f_r(w_i, X, d_i)$$

Vì vậy, học hệ số trọng lượng tổng quát trong phương trình $\Delta w_i = \eta r x(t)$ cho biết rằng vector hàm số trọng lượng w_i gia tăng theo tỉ lệ với tích của đầu vào $x(t)$ và tín hiệu học r . Với các phương trình cho trên, vector hệ số trọng lượng tại thời điểm bước học $(t+1)$ là:

$$w_i(t+1) = w_i(t) + \eta f_r(w_i, x(t), d_i) X(t)$$

mà cũng có thể được viết như:

$$w_i^{(t+1)} = w_i^t + \eta w_i^t + \eta f_r(w_i^t, x^t, d_i^t) X^t$$

Phương trình này chỉ được sử dụng để cải tiến một dãy của các hệ số trọng lượng thời gian rời rạc. Nếu các sự cải tiến thời gian hệ số trọng lượng thời gian liên tục, thì sử dụng phương trình được biểu hiện như sau:

$$\frac{dw_i(t)}{dt} = \eta r X(t)$$

Trong đề xuất của Hebbian, các hệ số trọng lượng được điều chỉnh tuân theo những tương quan trước và sau, tín hiệu học r trong qui tắc học tổng quát có thể được thiết lập như:

$$r = a(w_i^t X) = y_i$$

trong đó $a(.)$ là hàm tác động của phần tử xử lý.

Như vậy sự gia tăng Δw_i của vector hệ số trọng lượng sẽ trở nên

$$\Delta w_i = \eta a(w_i^t X) X = \eta y_i X$$

Điều đó có nghĩa là, các thành phần của vector hệ số trọng lượng được cập nhật với một lượng của

$$\Delta w_{ij} = \eta a(w_i^t X) X_j = \eta y_i X_j, \text{ cho } i = 1, 2, \dots, \text{neuron}; j = 1, 2, \dots, m$$

Điều này cho thấy rằng qui tắc học bebbian là một qui tắc học không giám sát cho một mạng neuron nuôi tiến vì nó chỉ sử dụng tích của các đầu vào và các đầu ra thật sự và cải tiến các hệ số trọng lượng. Không có các đầu ra mong muốn được cho phép để phát sinh ra tín hiệu học để cập nhật các hệ số trọng lượng. Qui tắc học này yêu cầu đến hệ số trọng lượng ban đầu ở các giá trị ngẫu nhiên nhỏ xung quanh 0 trước khi học. Phương trình trên cho biết rằng nhóm tương quan đầu vào – đầu ra $y_i x_j$ là số dương thì hệ số trọng lượng w_{ij} tăng lên; mặt khác hệ trọng lượng w_{ij} giảm xuống.

Nếu đề xuất tín hiệu học $a(w_i^l X) = y_i = d_i$ trong đó d_i là đầu ra mong muốn được để điều chỉnh các hệ số trọng lượng, vậy thì các thành phần của vector trọng lượng được cập nhật bởi:

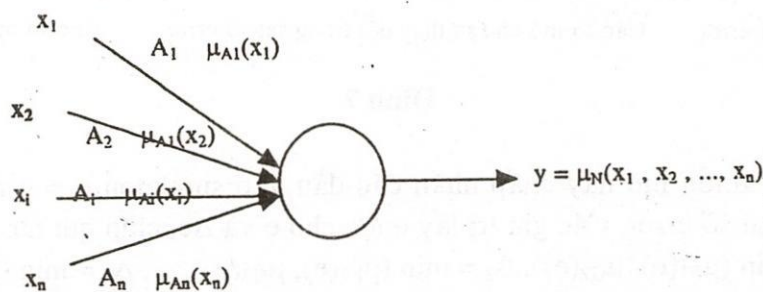
$$\Delta w_{ij} = \eta d_i x_j, \text{ cho } i = 1, 2, \dots, n; j = 1, 2, \dots, m$$

Điều này cho thấy rằng qui tắc học tổng quát giám sát cho một mạng neuron.

V) Nguyên lý cơ bản về sự tổng hợp của logic mờ và các mạng neuron:

Sự kết hợp logic mờ với mô hình mạng neuron được gọi là mô hình neuron mờ. Các hệ mờ và các mạng neuron cùng tham gia để cải tiến sự thông minh của các hệ thống làm việc trong các môi trường không chắc chắn, không chính xác có nhiều. Có ba loại mô hình neuron mờ thông dụng: Neuron mờ với các tín hiệu crisp được sử dụng để đánh giá các hệ số trọng lượng mờ, neuron mờ với các tín hiệu mờ mà được tổ hợp với các hệ số trọng lượng mờ và neuron mờ được mô tả bằng các phương trình logic mờ.

Kiểu mạng neuron mờ với các tín hiệu crisp được sử dụng để đánh giá các hệ số trọng lượng mờ được chỉ ở hình 6 dưới đây



Hình 6: Kiểu mạng Neuron với các tín hiệu crisp được sử dụng để đánh giá các hệ số trọng lượng mờ

Kiểu mạng neuron mờ này có neuron đầu vào dưới dạng crisp x_1, x_2, \dots, x_n và các hệ số trọng lượng là các tập mờ $A_i, 1 \leq i \leq n$; do đó, các phép toán hệ số trọng lượng phải được thay thế bằng các hàm thành viên. Kết quả của mỗi phép toán hệ số trọng lượng là một giá trị thành viên $\mu_{A_i}(x_i)$ của đầu vào tương ứng x_i trong tập mờ A_i . Tất cả các giá trị thành viên này được tổ hợp lại để cho một đầu ra trong khoảng $[0, 1]$, cái mà có thể được khảo sát mức tin cậy. Các phép toán tổ hợp được biểu diễn bằng \otimes có thể sử dụng cho bất kỳ toán tử tổ hợp nào cụ thể như min hoặc max và các toán tử khác t-norm và t-conorm.

Một sự biểu diễn toán học của mạng neuron mờ kiểu này được viết như:

$$\mu_N(x_1, x_2, \dots, x_n) = \mu_{A_1}(x_1) \otimes \mu_{A_2}(x_2) \otimes \dots \otimes \mu_{A_i}(x_i) \otimes \dots \otimes \mu_{A_n}(x_n)$$

trong đó x_i là đầu vào dưới dạng crisp thứ i và $\mu_{A_i}(x_i)$ là hàm thành viên của tập mờ A_i , $\mu_N(\cdot)$ là hàm thành viên đầu ra của mạng neuron và \otimes là toán tử tổ hợp.

Ví dụ : ví dụ sau đây dẫn chứng cách sử dụng kiểu mạng neuron mờ với các tín hiệu được sử dụng để đánh giá các hệ số trọng lượng mờ. Xây dựng một mạng neuron

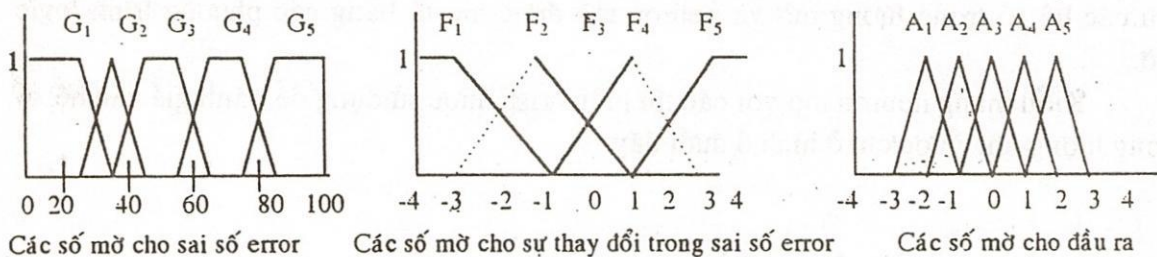
mờ như một bộ điều khiển mờ được chỉ định bằng các qui tắc được cho phép trong bảng dưới đây

$\Delta e =$ Sự thay đổi trong sai số error

$e =$ error

	G_1	G_2	G_3	G_4	G_5
F_1		A_1^1			
F_2		A_1^2		A_2^3	
F_3	A_2^4		A_3^5		A_4^6
F_4		A_4^7		A_5^8	
F_5				A_5^9	

Và các định nghĩa của các số mờ được biểu diễn trong hình 7 theo sau:

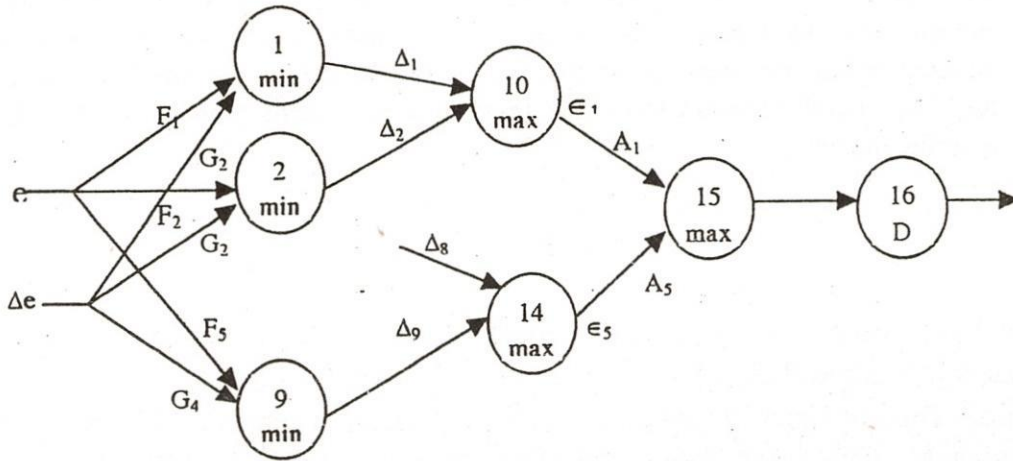


Hình 7

Bộ điều khiển mờ này chấp nhận các đầu vào singleton $e =$ error và $\Delta e =$ sự thay đổi trong sai số error. Các giá trị lấy được cho e và Δe , chín qui tắc được đánh giá như sau: $\Delta_1 = \min(\mu_{F_1}(e), \mu_{G_2}(e))$, $\Delta_2 = \min(\mu_{F_2}(e), \mu_{G_2}(e))$, ... , $\Delta_9 = \min(\mu_{F_5}(e), \mu_{G_4}(e))$. Vì một vài qui tắc có cùng những tác động điều khiển, ta tối đa Δ_i tương thích với cùng tác động điều khiển A_k như: $\epsilon_1 = \max(\Delta_1, \Delta_2)$, $\epsilon_2 = \max(\Delta_3, \Delta_4)$, $\epsilon_3 = \Delta_5$, $\epsilon_4 = \max(\Delta_6, \Delta_7)$, $\epsilon_5 = \max(\Delta_8, \Delta_9)$. Sau đó mỗi ϵ_k được gán với giá trị A_k của nó, $1 \leq k \leq 5$. Để giải mờ kết quả, đầu tiên ta tính $A = \cup(\epsilon_k, A_k)$ (trong đó toán tử hợp được lấy giá trị tối đa) và sau đó tìm y mà bằng trung tâm hấp dẫn của A ; y là đầu ra giải mờ được từ bộ điều khiển.

Bộ điều khiển mờ này được mô hình như một mạng neuron mờ được trình bày ở hình 8 dưới đây. Trong mạng này, các nút 1 đến 9 biểu hiện các quy tắc, các nút 10 đến 14 không có các hệ số trọng lượng đầu vào, và nút 16 không có hệ số trọng lượng, là một đơn vị giải mờ. Rõ ràng, các nút 1 đến 9 là những neuron mờ với các tín hiệu crisp để đánh giá hệ số trọng lượng mờ, nhưng nút 15 gần giống như một neuron mờ của kiểu mạng neuron mờ với các tín hiệu mờ mà được tổ hợp với các hệ số trọng lượng mờ. Lý do là các đầu vào của nút 15 là $\epsilon_i A_i$, $1 \leq i \leq 5$, chúng là các tập mờ, trong khi đó các đầu vào của nút 1 là $\mu_{F_1}(e)$ và $\mu_{G_2}(e)$, chúng là tích vô hướng. mạng neuron mờ xây dựng được có thể được mong muốn để học các số mờ thông suốt F_i , G_j và A_p được cho phép một vài dữ liệu huấn luyện (ϵ_k, A_k) cho các đầu vào và y_k^d cho các đầu ra giải mờ được, $k = 1, 2, \dots$. Để làm được điều này, ta cần có một giải thuật học sao cho

mạng sẽ học các hệ số trọng lượng của nó (các tập mờ trong các quy tắc) từ dữ liệu huấn luyện. những chi tiết học này sẽ được trình bày về sau.



Hình 8

Kiểu neuron mờ với các tín hiệu mờ mà được tổ hợp với các hệ số trọng lượng mờ giống như mạng neuron mờ được mô tả trên ngoại trừ tất cả các đầu vào và các đầu ra là các tập mờ đúng hơn là các giá trị crisp

Kiểu neuron mờ được mô tả bằng các phương trình logic mờ với neuron đầu vào mờ và một đầu ra mờ. Mối quan hệ đầu vào-đầu ra được biểu diễn bằng một quy tắc IF - THEN:

$$\text{IF } X_1 \text{ AND } X_2 \text{ AND } \dots \text{ AND } X_n \text{ THEN } Y.$$

trong đó X_1, X_2, \dots, X_n là các đầu vào hiện có và Y là đầu ra hiện có. Kiểu mạng neuron mờ này có thể được mô tả bằng một quan hệ mờ R , cụ thể như :

$$R = f(X_1, X_2, \dots, X_n, Y)$$

trong đó $f(\cdot)$ biểu diễn một hàm suy luận. Vì vậy, kiểu mạng neuron mờ này được mô tả bằng một quan hệ truyền mờ R . Cho phép các luồng đầu vào (mờ hoặc không mờ) x_1, x_2, \dots, x_n , tuân theo quy tắc gán phức hợp của suy diễn, neuron mờ có đầu ra như sau:

$$Y = x_1 \circ (x_2 \circ (\dots \circ (x_n \circ R_i) \dots))$$

trong đó \circ biểu diễn bất kì các quy tắc gán phức hợp của toán tử suy diễn cụ thể như max, t - norm. Kiểu mạng neuron mờ này thường hay xuất hiện trong các hệ chuyên gia mờ, sẽ được thảo luận chi tiết về sau.

**SOME NEW TECHNOLOGIES:
FUZZY LOGIC AND FUZZY NEURON
IN THE CONTROL SYSTEMS**

Nguyễn Thiện Thành

Astract:

This paper describes the intelligent control systems implemented by new technologies about processing information such as fuzzy logic and fuzzy neuron. This paper is

constructed by the basic concept of fuzzy logic and artificial neural networks, and they are then combined to share the ability for modifying the intelligence of control systems working in uncertain, imprecise. And noisy environments. Systems desired working in such these environments, systems must be build up by the four main elements: fuzzification strategies, knowledge base, fuzzy reasoning mechanism, and defuzzification strategies. These basic elements associated with the basic concepts of artificial neural networks forms the fuzzy neuron systems trained by using different learning modes.

Tài liệu tham khảo

1. Jun Yan, Micheal ryan and James Power, Using Fuzzy Logic Towards intelligent systems, Prentice Hall (1994)
2. Chin – Teng lin and C.S Geogre Lee, Neural Fuzzy Systems, Prentice Hall (1996)
3. Madan M. Gupta and dandina H. Rao, Neural – Control Systems Theory and Applications, IEEE Press (1994)
4. Timothy J. Ross, Fuzzy Logic with engineering Applications, McGran–Hill (1995)
5. Constan Von Altrock, Fuzzy Logic and Neuron Fuzzy Applications Explained, Pretice Hall (1995)
6. Robert Harvey, Neural network Principles, Prentice Hall (1994)
7. Duc Truong Pham and Ercan Oztemel, Intelligent Quality Systems Springer – Verlag London (1996)
8. Duc Truong Pham and Liu Xing, Neural network for Identification, Prediction and Control Springer –Verlag London (1997)