GT.0000026665

Ộ CÔNG THƯƠNG
NG ĐẠI HỌC SAO ĐỎ

TS. ĐINH VĂN NHƯỢNG

GIÁO TRÌNH

MỘT SỐ ỨNG DỤNG MẠNG NƠRON XÂY DỰNG MÔ HÌNH NHẬN DẠNG VÀ DỰ BÁO

I NGUYÊN
HỌC LIỆU
3
J

NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

96792

BỘ CÔNG THƯƠNG – TRƯỜNG ĐẠI HỌC SAO ĐỎ

TS. ĐINH VĂN NHƯỢNG

GIÁO TRÌNH

MỘT SỐ ỨNG DỤNG MẠNG NƠRON XÂY DỰNG MÔ HÌNH NHẬN DẠNG VÀ DỰ BÁO

ĐẠI HỌC THÁI NGUYÊN
TRUNG TÂM HỌC LIỆU

NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
HÀ NỘI – 2013

UNIVERSITY OF MICHIGAN

Lời nói đầu

Các ứng dụng mạng nơron trong mô hình nhận dạng và dự báo đã và đang trong giai đoạn phát triển mạnh mẽ cả về phương diện lý thuyết cũng như thực tế, thu hút sự quan tâm của đông đảo các nhà khoa học trong và ngoài nước. Nhiều mô hình điều khiển mờ đã được nghiên cứu và xây dựng dựa trên các quy tắc suy luận của trí tuệ nhân tạo, giúp con người có khả năng chế ngự được những đối tượng có những thông số biến động mạnh và phức tạp. Điều khiển mờ, hay điều khiển mờ thích nghi đã nâng cao chất lượng điều khiển, đặc biệt đối với các bài toán nhận dạng, dự báo mà tín hiệu đầu vào có nhiều thông số biến thiên phức tạp. Một trong những bài toán đó phải kể đến là các bài toán nhận dạng phân loại sản phẩm công nghiệp và các bài toán dự báo nói chung, đã và đang được ứng dụng mạnh và cho kết quả tốt.

Cuốn ***"Giáo trình một số ứng dụng mạng nơron xây dựng mô hình nhận dạng và dự báo"*** *gồm có 3 phần, 4 chương. Nội dung đề cập đến một số vấn đề lý thuyết cơ bản của mạng nơron, đồng thời cũng đưa ra một số mô hình ứng dụng mạng nơron trong nhận dạng đối tượng là hàm phi tuyến, một số sản phẩm công nghiệp cũng như dự báo sự cố xảy ra trong máy điện...*

Trong quá trình biên soạn cuốn giáo trình này, tác giả đã tham khảo nhiều sách, nhiều công trình nghiên cứu cùng với những kết quả ứng dụng mô hình mạng nơron của chính các tác giả, của các bài báo khoa học đã được báo cáo trong các hội thảo khoa học, được đăng trên các tạp chí nghiên cứu khoa học có uy tín trong và ngoài nước. Đồng thời, trong quá trình biên soạn tác giả cũng nhận được sự đóng góp ý kiến của các đồng nghiệp khoa Điện, khoa Điện tử – Tin học (Trường Đại học Sao Đỏ) và Ban Biên tập (Nhà xuất bản Khoa học và Kỹ thuật). Hy vọng rằng, cuốn giáo trình sẽ là tài liệu học tập, tham khảo hữu ích cho sinh viên, cán bộ kỹ thuật nghiên cứu thuộc các lĩnh vực Điện, Điện tử, Đo lường điều khiển và Tin học công nghiệp.

Mặc dù đã cố gắng, song cuốn giáo trình xuất bản lần đầu nên khó tránh khỏi những thiếu sót, tác giả mong nhận được sự góp ý của bạn đọc để ngày càng hoàn thiện hơn trong lần xuất bản sau.

Thư góp ý xin được gửi về theo địa chỉ Email: nhuongdv2000@gmail.com .

Trân trọng cảm ơn.

TÁC GIẢ

MỤC LỤC

Trang

LỜI NÓI ĐẦU 3

Phần I. NHẬP MÔN 5

CHƯƠNG 1. TỔNG QUAN VỀ MẠNG NƠRON 5

1.1. Lịch sử phát triển và ứng dụng mạng nơron 5

1.2. Mạng nơron nhân tạo 9

1.3. Mô hình mạng nơron 12

1.4. Huấn luyện mạng nơron 17

1.5. Thuật toán lan truyền ngược 21

Phần II. MỘT SỐ MẠNG NƠRON THƯỜNG SỬ DỤNG TRONG NHẬN DẠNG PHÂN LOẠI 29

CHƯƠNG 2. MẠNG NƠRON 29

2.1. Mạng nơron một lớp 29

2.2. Mạng MLP (Multilayer Perceptrons Network) 44

CHƯƠNG 3. MẠNG NƠRON LÔGIC MỜ TSK 50

3.1. Lôgic mờ 50

3.2. Mạng TSK (Takaga – Sugeno – Kang) 56

Phần III. MỘT SỐ ỨNG DỤNG MẠNG NƠRON TRONG NHẬN DẠNG VÀ DỰ BÁO 88

CHƯƠNG 4. ỨNG DỤNG MẠNG NƠRON TRONG NHẬN DẠNG VÀ DỰ BÁO 88

4.1. Nhận dạng và các phương pháp tiếp cận 88

4.2. Ứng dụng trong nhận dạng hàm phi tuyến 90

4.3. Ứng dụng trong nhận dạng, phân loại 102

4.4. Ứng dụng trong dự báo sự cố động cơ 109

TÀI LIỆU THAM KHẢO 125

Phần I
NHẬP MÔN

Chương 1
TỔNG QUAN VỀ MẠNG NƠRON

1.1. LỊCH SỬ PHÁT TRIỂN VÀ ỨNG DỤNG MẠNG NƠRON

1.1.1. Lịch sử phát triển mạng nơron

Trong những thập niên vừa qua một ngành khoa học mới đã và đang được đầu tư nghiên cứu và phát triển rất mạnh trên thế giới cũng như ở Việt Nam, đó là những nghiên cứu về ứng dụng trí tuệ nhân tạo trong việc giải các bài toán: Xử lý tín hiệu; Đo lường; Điều khiển; Dự báo...

Thế kỷ XX đã đánh dấu việc xuất hiện của các phần tử bán dẫn, đặc biệt là transistor và sau đó là các mạch tích hợp mà ứng dụng quan trọng nhất là tạo ra được máy tính điện tử. Kể từ ngày ra đời, máy tính điện tử đã không ngừng được hoàn thiện, tốc độ tính toán đạt tới kỷ lục khó tưởng tượng, kích thước các mạch tính toán ngày càng nhỏ gọn. Tuy nhiên, nếu so sánh với khả năng con người thì các máy móc còn thua kém rất xa. Trước tiên có thể kể tới các giác quan và khả năng phân tích, xử lý thông tin của con người. Ví dụ như con người có khả năng phân biệt mùi qua khứu giác, trong khi những ma trận cảm biến khí hiện đại nhất hiện nay trên thế giới cũng không thể đạt được mức độ chính xác như khứu giác con người và thường cũng chỉ phản ứng với một vài loại khí và mùi khác nhau. Cũng có thể lấy ví dụ về mắt người có một khả năng ghi nhớ các hình ảnh và phân tích, nhận dạng các ảnh mới ngay cả trong trường hợp đối tượng được nhận dạng đã bị thay đổi rất nhiều. Khả năng này hiện nay cũng đang được các nhà khoa học trên thế giới tìm cách mô phỏng, tuy nhiên kết quả còn hết sức hạn chế. Khả năng lớn nhất của con người mà hiện nay chưa có hệ thống

nào mô phỏng được với kết quả khả quan đó là: khả năng tư duy, tự suy nghĩ và tự tìm giải pháp từ kết quả tư duy. Chính khả năng này đã đưa lại được tính "thông minh" cho con người.

Do đó trong lĩnh vực trí tuệ nhân tạo ta nghiên cứu phương pháp mô phỏng các hoạt động, các phương pháp tư duy, phân tích của con người để xây dựng những thiết bị, hệ thống có chức năng tương tự. Để giải quyết vấn đề này có một ngành đã và đang được phát triển mạnh với nhiều kết quả được áp dụng trong thực tế đó là các nghiên cứu về mạng nơron và nơron lôgic mờ. Các mạng nơron và nơron lôgic mờ được xây dựng nhằm mục đích mô phỏng quá trình học và suy luận tương tự như học và suy luận của con người.

Hàng loạt các ứng dụng thực tế đã áp dụng các mạng nơron và nơron lôgic mờ với kết quả tốt hơn hẳn so với những giải pháp kinh điển. Một trong những ứng dụng quan trọng đầu tiên của các giải pháp nơron và lôgic mờ là hệ thống điều khiển tự động các đường tàu điện ngầm tại Nhật Bản vào cuối những năm 70, đầu những năm 80 của thế kỷ trước. Ngày nay ta thấy những giải pháp nơron và nơron lôgic mờ có mặt ở khắp nơi, ngay cả trong những thiết bị điện tử dân dụng như bộ điều khiển máy giặt, máy điều hoà nhiệt độ, tủ lạnh và đang được ứng dụng mạnh mẽ vào trong công nghiệp và được nhiều nhà khoa học quan tâm nghiên cứu. Theo R.Schalkoff thì có thể chia sự phát triển của mạng nơron nhân tạo thành 3 giai đoạn:

Giai đoạn 1: Tiền Perceptron (những năm 1940 – 1960): Trong giai đoạn này mạng chưa đủ phức tạp cho nên chưa có khả năng giải quyết các bài toán khó có sức thuyết phục. Các sự kiện trong giai đoạn này cần kể đến:

– Năm 1943 McCulloch lần đầu tiên giới thiệu mô hình toán học của mạng nơron.

– Năm 1957 Rosenblatt định nghĩa Perceptron mong muốn khẳng định các nơron liên kết, phi tuyến tạo nên mạng thích nghi có thể góp phần giải quyết các bài toán nhận dạng.

– Năm 1960 Windrow đóng góp chính là thuật toán trung bình bình phương cực tiểu LMS cho mô hình Adaline/Madaline.

– Kết quả của Minsky và Papert năm 1969.

Giai đoạn 2: Hậu Perceptron: Giai đoạn này Perceptron được phát huy với những thuật toán truyền thẳng và liên kết suy rộng. Tìm thêm nhiều cấu trúc mới, trong đó cần kể tới:

– Mạng truyền thẳng với thuật toán lan truyền ngược (luật Delta suy rộng-GDR) năm 1985.

– Mạng dùng các hàm cơ sở xuyên tâm (mạng RBF).

– Các mạng Hopfield hồi quy năm 1982.

– Bộ nhớ liên hợp hai chiều (ABAM) năm 1987.

– Công trình các mạng thích nghi của Grossberg và Kohonen.

Giai đoạn 3: Giai đoạn gần đây và hiện nay tiếp tục nghiên cứu và đưa vào thực tiễn nhiều mô hình và thuật toán đã hoàn chỉnh hơn. Những vấn đề chính hiện nay đang cần nghiên cứu là:

– Đánh giá xác thực những hạn chế của mạng nơron.

– Các khả năng suy rộng khác.

– Phối hợp công nghệ mạng nơron và các công nghệ lôgic mờ và các thuật toán di truyền.

– Cài đặt các mạng nơron nhân tạo bằng các phần cứng chuyên dụng.

Những thuật toán điều khiển mờ đang được quan tâm và đạt được nhiều kết quả khả quan, ứng dụng nhiều trong công nghiệp đó là:

Điều khiển Mamdani (MC- Mamdani Control).

Điều khiển mờ trượt (SMFC-Sliding Mode Fuzzy Control).

Điều khiển tra bảng (CM-Cell Maping Control).

Điều khiển Takaga – Sugeno – Kang (TSK).

1.1.2. Phạm vi ứng dụng

Lĩnh vực ứng dụng của mạng nơron nhân tạo rất rộng, chủ yếu tập trung trong các lĩnh vực sau:

Lĩnh vực 1: Phân lớp (classification), tách cụm (clustering), dự đoán (diagnoisis) và liên kết. Đây là lĩnh vực tìm thấy nhiều ứng dụng nhất và cũng được nghiên cứu nhiều nhất. Nhóm mô hình này nhận dạng những tín hiệu vào tĩnh hoặc tín hiệu theo thời gian và cần nhận dạng hoặc phân lớp chúng. Thuật toán phân lớp cần huấn luyện mạng sao cho khi tín hiệu vào bị

biến dạng ít nhiều thì mạng vẫn nhận đúng dạng thực tế của chúng. Trong lĩnh vực này, yêu cầu mạng có khả năng miễn nhiễu tốt, đây cũng là mong muốn của nhiều ứng dụng.

Lĩnh vực 2: Các bài toán tối ưu. Vấn đề chính ở đây là tìm những thuật toán huấn luyện mạng sao cho góp phần tìm nghiệm cho nhiều lớp bài toán tối ưu toàn cục. Trong nhóm các thuật toán ứng dụng mạng nơron, người ta đã quan tâm đến sự kết hợp mạng nơron với các thuật toán di truyền.

Lĩnh vực 3: Hồi quy và tổng quát hoá (Regression and Generalization). Trước đây các bài toán hồi quy đã được tích cực nghiên cứu. Qua hồi quy tuyến tính và phi tuyến người ta cố gắng tìm các đường thẳng hoặc các đường hồi quy phi tuyến trơn sao cho khớp với mẫu. Trong bài toán hồi quy người ta thường dùng các thuật toán có giám sát nên bài toán suy rộng khó hơn, vì dữ liệu được học mới chỉ có một phần.

Lĩnh vực 4: Hoàn chỉnh dạng (Pattern completion). Bài toán là hoàn chỉnh "Đủ" dữ liệu ban đầu sau khi đã bị mất đi một phần hay ta chỉ thu được một phần. Người ta đã quan tâm tới hai mô hình: Mô hình Markov và các mạng có độ trễ với các mạng nơron nhiều lớp, mạng Bolzmann và mạng Hopfield tĩnh.

Trong những năm của thập kỷ này được xem là thời kỳ nở rộ của các công trình khoa học nghiên cứu về mạng Mờ-nơron cũng như Nơron-mờ với những ứng dụng trong nhận dạng hình ảnh, trong hệ thống hỗ trợ quyết định, trong cơ chế suy diễn Nơron-mờ. Nguyên nhân của sự phát triển đó là sự ra đời của mạng Hopfield, Tank, tiếp nối là sự hoàn thiện các thuật toán lan truyền ngược của Rumenlhart, Hinton, Willams, Nauck và Kruse cho mạng MLP (Multilayer perceptrons Network). Nguyên nhân nữa thúc đẩy sự phát triển này chính là các sản phẩm lôgic mờ ở Nhật Bản phát triển mạnh mẽ và các "Chíp mờ" đã được ứng dụng trong điều khiển: Máy giặt, nồi cơm điện, máy điều hoà. Hiện nay hệ thống điều khiển mờ đang được ứng dụng ở một số nhà máy xi măng có hệ thống tự động hoá hiện đại ở nước ta, trong đó có Tổng công ty xi măng Hoàng Thạch.

Các công trình nghiên cứu ứng dụng mạng nơron lôgic mờ trong các bài toán nhận dạng như: Nhận dạng chữ viết; Nhận dạng tiếng nói; nhận dạng dấu vân tay; nhận dạng sự cố tiềm ẩn trong thiết bị điện; nhận dạng

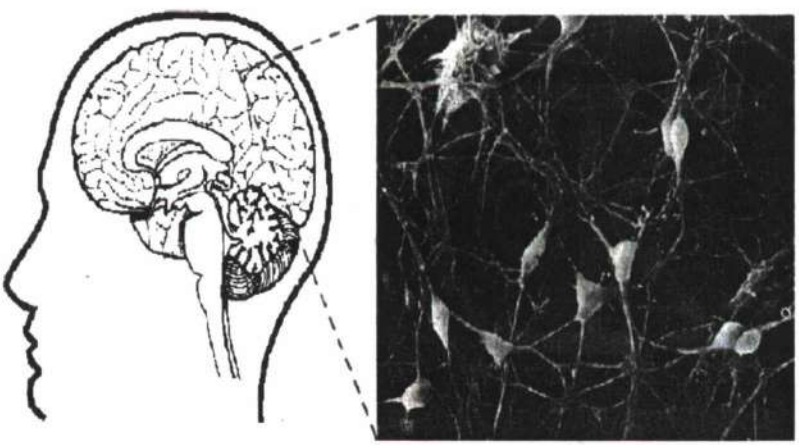
phân loại khí thải công nghiệp,...; xử lý ảnh và nhiều ứng dụng khác trong các lĩnh vực Quốc phòng, Y tế, Xây dựng,...

1.2. MẠNG NƠRON NHÂN TẠO

1.2.1. Não và nơron sinh học

Não là tổ chức cao cấp có cấu tạo vô cùng phức tạp, dày đặc các mối liên kết giữa các nơron nhưng xử lý thông tin linh hoạt trong một môi trường bất định.

Trong bộ não có khoảng 10^{11} đến 10^{12} nơron và mỗi nơron có thể liên kết với 10^4 nơron khác qua các khớp nối. Những kích hoạt ức chế này được truyền qua trục nơron (axon) đến các nơron khác. Trên hình 1.1 là hình ảnh của tế bào nơron trong bộ não của con người.



Hình 1.1. **Hình ảnh của tế bào nơron trong bộ não của con người**

Nơron sinh học: Phần tử xử lý cơ bản của một mạng nơron sinh học là một nơron, phần tử này có thể chia làm bốn thành phần cơ bản như sau: dendrites, soma, axon, và synapses.

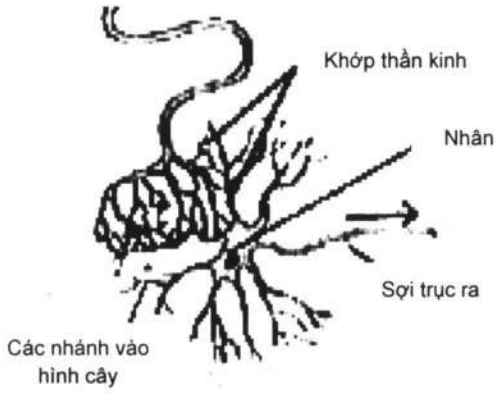
– Dendrites: là phần nhận tín hiệu đầu vào.

– Soma: là hạt nhân.

– Axon: là phần dẫn ra tín hiệu xử lý.

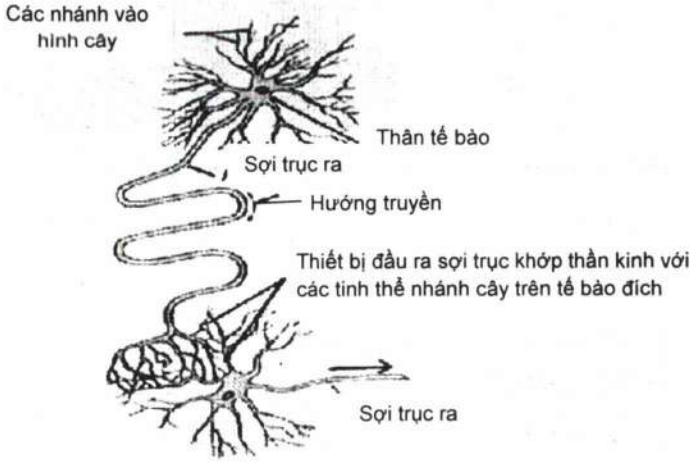
– Synapses: là đường tín hiệu điện hóa giao tiếp giữa các nơron.

Một cách tổng quát, thì một nơron sinh học nhận đầu vào từ các nguồn khác nhau, kết hợp chúng tại với nhau, thực thi tổ hợp phi tuyến chúng để cho ra kết quả cuối cùng ở đầu ra. Hình 1.2 chỉ ra mối quan hệ giữa bốn phần tử của một nơron sinh học.



Hình 1.2. **Một nơron sinh học**

Một nơron sinh học chỉ có một số chức năng cơ bản, như vậy ta nhận thấy khả năng xử lý thông tin của nó là rất yếu. Để có được khả năng xử lý thông tin hoàn hảo như bộ não con người, thì các nơron phải kết hợp và trao đổi thông tin với nhau. Ta hình dung sơ đồ liên kết, và trao đổi thông tin giữa hai nơron như hình 1.3



Hình 1.3. **Sự liên kết các nơron**

Khi ta nhìn não từ góc độ tính toán, chúng ta dễ dàng phát hiện cách thức tính toán của não khác xa với tính toán theo thuật toán và chương trình chúng ta thường làm với sự trợ giúp của máy tính. Sự khác biệt trước tiên ở hai điểm quan trọng là:

– Quá trình tính toán được tiến hành song song và phân tán trên nhiều nơron gần như đồng thời.

– Tính toán thực chất là quá trình học chứ không phải theo sơ đồ định sẵn từ trước.

Thông thường, một mạng nơron bao gồm một hoặc nhiều các nơron được kết nối vật lý với nhau hoặc có liên kết với nhau về mặt chức năng. Một nơron đơn có thể được nối với nhiều nơron khác và tổng số nơron kết nối trong một mạng có thể là giá trị cực kỳ lớn. Các kết nối gọi là các khớp thần kinh (synapeses), thường nối từ các axon tới các tế bào tua gai thần kinh (dendrite), tuy có thể có các vi mạch dendrodentritic và các kết nối khác. Do vậy, cũng như các mạng sinh học khác, mạng nơron vô cùng phức tạp.

1.2.2. Nơron nhân tạo

Mạng nơron nhân tạo là một mô phỏng xử lý thông tin, được nghiên cứu dựa trên hệ thống thần kinh của sinh vật, giống như bộ não để xử lý thông tin. Nó bao gồm số lượng lớn các mối gắn kết cấp cao để xử lý các yếu tố làm việc trong mối liên hệ giải quyết vấn đề rõ ràng. ANN (Antifical Neural Networks) giống như con người, được học bởi kinh nghiệm, lưu giữ những kinh nghiệm hiểu biết và sử dụng trong những tình huống phù hợp.

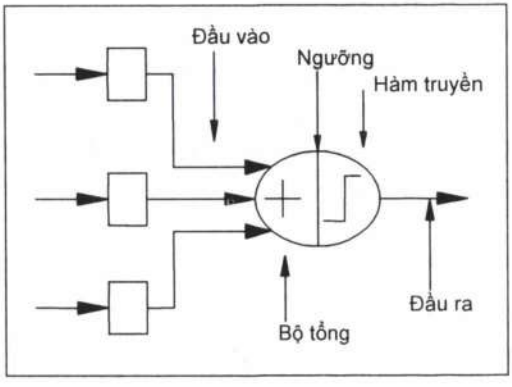
Đầu tiên ANN được giới thiệu năm 1943 bởi nhà thần kinh học Warren McCulloch và nhà lôgic học Walter Pits. Nhưng với những kỹ thuật trong thời gian này chưa cho phép họ nghiên cứu được nhiều. Những năm gần đây mô phỏng ANN xuất hiện và phát triển. Các nghiên cứu ứng dụng đó được thực hiện trong các ngành: điện, điện tử, kỹ thuật chế tạo, y học, quân sự, kinh tế... và mới nhất là các nghiên cứu ứng dụng trong lĩnh vực quản lý dự án xây dựng. Tại Việt Nam việc nghiên cứu ứng dụng ANN vào quản lý xây dựng chỉ mới bắt đầu trong vài năm gần đây và cần được phát triển.

1.3. MÔ HÌNH MẠNG NƠRON

1.3.1. Mô hình về một nơron

Mô hình toán học của nơron sinh học được đề xuất bởi McCulloch và Pitts, thường được gọi là nơron M-P, ngoài ra nó còn được gọi là phần tử xử lý và được ký hiệu là PE (Processing Element).

Mô hình nơron có m đầu vào $x_1, x_2, ..., x_m$, và một đầu ra y_i như sau:



Hình 1.4. **Mô hình một nơron nhân tạo**

Giải thích các thành phần cơ bản:

– Tập các đầu vào: Là các tín hiệu vào của nơron, các tín hiệu này thường được đưa vào dưới dạng một vectơ m chiều.

– Tập các liên kết (các trọng số): Mỗi liên kết được thể hiện bởi một trọng số (thường được gọi là trọng số liên kết). Trọng số liên kết giữa tín hiệu vào thứ j cho nơron i thường được ký hiệu là w_{ij}. Thông thường các trọng số này được khởi tạo ngẫu nhiên ở thời điểm khởi tạo mạng và được cập nhật liên tục trong quá trình học mạng.

– Bộ tổng: Thường dùng để tính tổng của tích các đầu vào với trọng số liên kết của nó.

– Ngưỡng: Ngưỡng này thường được đưa vào như một thành phần của hàm truyền.

– Hàm truyền: Hàm này dùng để giới hạn phạm vi đầu ra của mỗi nơron. Nó nhận đầu vào là kết quả của hàm tổng và ngưỡng đã cho. Thông thường, phạm vi đầu ra của mỗi nơron được giới hạn trong đoạn [0,1] hoặc

[−1,1]. Các hàm truyền rất đa dạng, có thể là các hàm tuyến tính hoặc phi tuyến. Việc lựa chọn hàm truyền tùy thuộc vào từng bài toán và kinh nghiệm của người thiết kế mạng.

– Đầu ra: Là tín hiệu đầu ra của một nơron, với mỗi nơron sẽ có tối đa một đầu ra.

Về mặt toán học, cấu trúc của một nơron i được mô tả bằng cặp biểu thức sau:

$$y_i = f(net_i - \theta_i) \text{ và } net_i = \sum_{j=1}^{n} w_{ij} x_j \tag{1.1}$$

trong đó: $x_1, x_2, \ldots x_m$ – các tín hiệu đầu vào;

$w_{i1}, w_{i2}, \ldots, w_{im}$ – các trọng số kết nối của nơron thứ i;

net_i – hàm tổng;

f – hàm truyền;

θ_i – một ngưỡng;

y_i – tín hiệu đầu ra của nơron.

Như vậy, tương tự như nơron sinh học, nơron nhân tạo cũng nhận các tín hiệu đầu vào, xử lý (nhân các tín hiệu này với trọng số liên kết, tính tổng các tích thu được rồi gửi kết quả đến hàm truyền), và cho một tín hiệu đầu ra (là kết quả của hàm truyền).

Hàm truyền có thể có các dạng sau:

– Hàm bước $y = \begin{cases} 1 & \text{khi} \quad x \geq 0 \\ 0 & \text{khi} \quad x < 0 \end{cases}$ (1.2)

– Hàm giới hạn chặt (hay còn gọi là hàm bước)

$$y = \text{sgn}(x) = \begin{cases} 1 & \text{khi} \quad x \geq 0 \\ -1 & \text{khi} \quad x < 0 \end{cases} \tag{1.3}$$

– Hàm bậc thang

$$y = \text{sgn}(x) = \begin{cases} 1 & \text{khi} \quad x > 1 \\ x & \text{khi} \quad 0 \leq x \leq 1 \\ 0 & \text{khi} \quad x < 0 \end{cases} \tag{1.4}$$

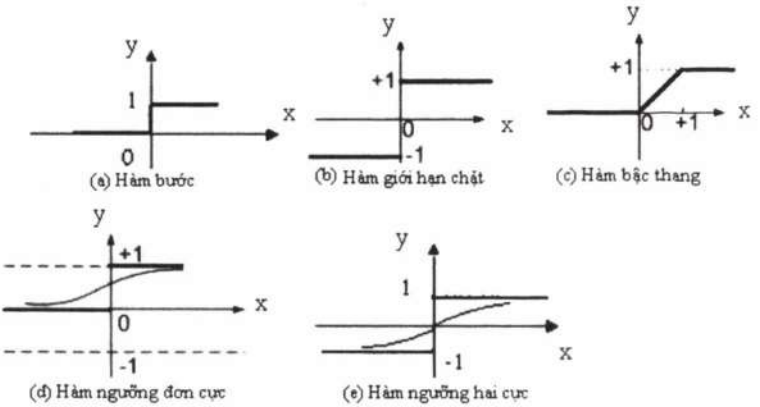
– Hàm ngưỡng đơn cực

$$y = \frac{1}{1 + e^{-\lambda x}} \text{ với } \lambda > 0 \tag{1.5}$$

– Hàm ngưỡng hai cực

$$y = \frac{2}{1+e^{-\lambda x}} - 1 \text{ với } \lambda > 0 \qquad (1.6)$$

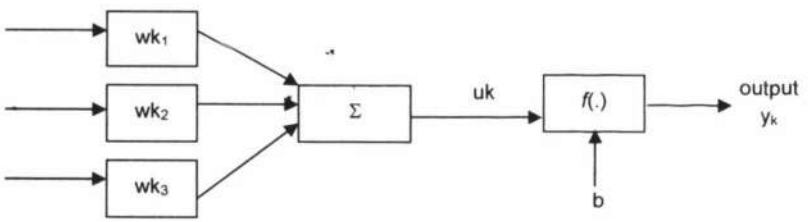
Đồ thị các dạng hàm truyền được biểu diễn như hình 1.5:



Hình 1.5. **Đồ thị các dạng hàm truyền**

1.3.2. Mô hình về mạng nơron

Mỗi nơron (nút) là một đơn vị xử lý thông tin của mạng nơron, là yếu tố cơ bản để cấu tạo nên mạng nơron.



Hình 1.6. **Mô hình mạng nơron**

x_i – các tín hiệu input;

W_{kp} – trọng số của từng input;

f(.) – hàm kích hoạt;

y_k – kết xuất của nơron;

b – thông số ảnh hưởng đến ngưỡng ra của output.

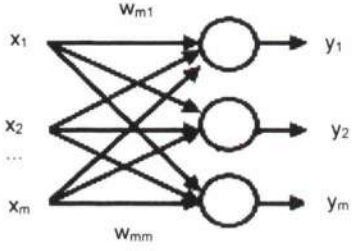
• ***Mạng một lớp***

Mỗi một nơron có thể phối hợp với các nơron khác tạo thành một lớp các trọng số. Mạng một lớp truyền thẳng như hình 1.7a. Một lớp nơron là một nhóm các nơron mà chúng đều có cùng trọng số, nhận cùng một tín hiệu đầu vào đồng thời.

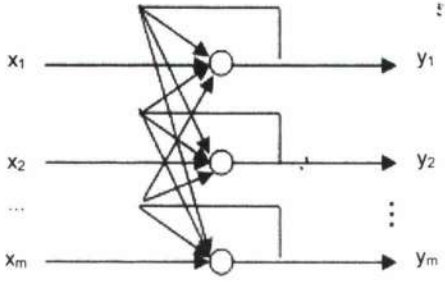
Trong ma trận trọng số, các hàng là thể hiện nơron, hàng thứ j có thể đặt nhãn như một vectơ w_j của nơron thứ j gồm m trọng số w_{ji}. Các trọng số trong cùng một cột thứ j (j=1,2,...,n) đồng thời cùng nhận một tín hiệu đầu vào x_j.

$$w_j = [w_{j1}, w_{j2}, ..., w_{jm}] \quad (1.7)$$

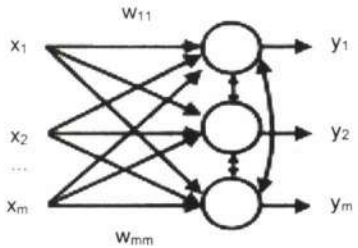
Tại cùng một thời điểm, vectơ đầu vào $x = [x_1, x_2,..., x_n]$ có thể là một nguồn bên ngoài là cảm biến hoặc thiết bị đo lường đưa tới mạng.



Hình 1.7a. Mạng truyền thẳng một lớp



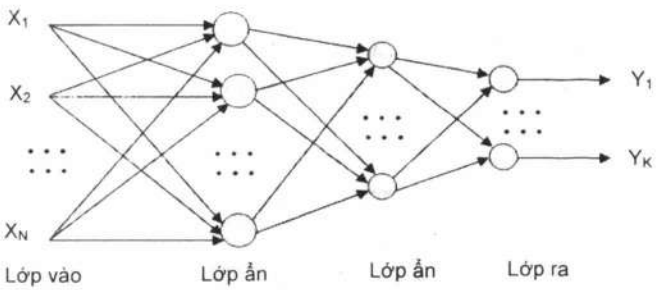
Hình 1.7b. Mạng hồi tiếp một lớp



Hình 1.7c. Mạng nơron hồi quy một lớp

***Hình.1.7.* Một số dạng mạng nơron một lớp**

• ***Mạng nhiều lớp***



***Hình 1.8.* Mạng nơ rôn nhiều lớp**

Mạng nơron nhiều lớp có thể giải quyết các bài toán phi tuyến nhờ vào các lớp ẩn. Các lớp ẩn này xen giữa các input bên ngoài và output của mạng. Càng nhiều lớp ẩn thì khả năng mở rộng thông tin càng cao và xử lý tốt mạng có nhiều input và output. Ngoài ra còn có mạng hồi quy và mạng nơron dạng lưới.

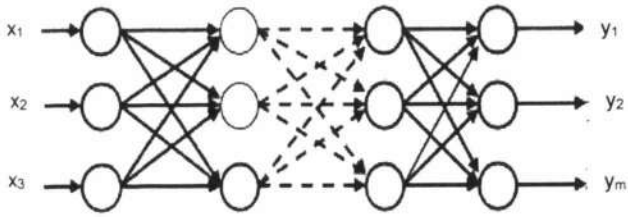
Mạng nơron truyền thẳng một lớp

Hình 1.9 mô hình mạng nơron truyền thẳng một lớp. Cấu trúc gồm có:

– Lớp vào là lớp nơron đầu tiên nhận tín hiệu vào x_i (i = 1, 2, ..., n). Mỗi tín hiệu x_i được đưa đến tất cả các nơron của lớp đầu vào. Thông thường, các nơron đầu vào không làm biến đổi các tín hiệu vào x_i, tức là chúng không có các trọng số hoặc không có các loại hàm chuyển đổi nào, chúng chỉ đóng vai trò phân phối các tín hiệu.

– Lớp ẩn là lớp nơron sau lớp vào, chúng không trực tiếp liên hệ với thế giới bên ngoài như các lớp nơron vào/ra.

– Lớp ra là lớp nơron tạo ra các tín hiệu ra cuối cùng.



Hình 1.9. **Mạng truyền thẳng nhiều lớp**

1.4. HUẤN LUYỆN MẠNG NƠRON

1.4.1. Hàm hoạt động của mạng nơron

Các hàm hoạt động phải có các đặc tính sau:

– Hàm bị chặn trên và chặn dưới.

– Hàm có tính đơn điệu.

– Hàm phải có tính liên tục và trơn.

Trong thực tế, thông thường người ta thường chọn các hàm sau:

– Hàm Threhold:

$$f(u) = \begin{cases} 1 & u \geq 0 \\ 0 & u \leq 0 \end{cases} \tag{1.8}$$

– Hàm piecewwise – linear

$$f(u) = \begin{cases} 1 & u \geq 1/2 \\ u & 1/2 > u > -1/2 \\ 0 & u \leq -1/2 \end{cases} \tag{1.9}$$

– Hàm sigmoid (logistic)

$$f(u) = 1 / (1 + \exp(-au)) \tag{1.10}$$

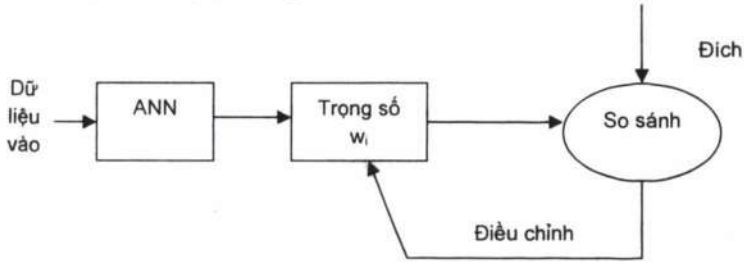
– Hàm tang – hyperbol

$$f(u) = \tanh(u) = \frac{e^{u} - e^{-u}}{e^{u} + e^{-u}} \tag{1.11}$$

ĐẠI HỌC THÁI NGUYÊN
TRUNG TÂM HỌC LIỆU

1.4.2. Các luật học

Thông thường, mạng nơron được điều chỉnh hoặc được huấn luyện để hướng các đầu vào riêng biệt đến đích ở đầu ra. Cấu trúc huấn luyện mạng được chỉ ra ở hình 1.10. Ở đây, hàm trọng số của mạng được điều chỉnh trên cơ sở so sánh đầu ra với đích mong muốn (taget), cho tới khi đầu ra của mạng phù hợp với đích. Những cặp vào/đích (input/taget) được dùng để giám sát cho sự huấn luyện mạng.



Hình 1.10. **Cấu trúc huấn luyện mạng nơron**

Để có được một số cặp vào/ra, ở đó mỗi giá trị vào được gửi đến mạng và giá trị ra tương ứng được thực hiện bằng mạng là sự xem xét và so sánh với giá trị mong muốn. Bình thường, nó sẽ tồn tại một sai số vì giá trị mong muốn không hoàn toàn phù hợp với giá trị thực. Sau mỗi lần chạy, ta có tổng bình phương của tất cả các sai số. Sai số này được sử dụng để xác định các hàm trọng số mới.

Sau mỗi lần chạy, hàm trọng số của mạng được sửa đổi với đặc tính tốt hơn tương ứng với đặc tính mong muốn. Từng cặp giá trị vào/ra phải được kiểm tra và trọng số được điều chỉnh một vài lần. Sự thay đổi các hàm trọng số của mạng sẽ được dừng lại, nếu tổng các bình phương sai số nhỏ hơn một giá trị đặt trước, hoặc đã chạy đủ một số lần chạy xác định (trong trường hợp này, mạng có thể không thoả mãn yêu cầu đặt ra do sai lệch còn cao). Có hai kiểu học:

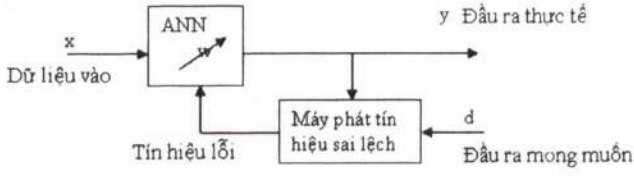
– Học tham số: Là các tham số về trọng số cập nhật kết nối giữa các nơron.

– Học cấu trúc: Trọng tâm là sự biến đổi cấu trúc của các mạng nơron gồm số lượng nút và các loại liên kết.

Giả sử ma trận trọng số bao gồm tất cả các phần tử thích ứng của mạng nơron. Nhiệm vụ của việc học tham số là tìm ra được ma trận chính xác mong muốn từ ma trận giả thiết ban đầu (với cấu trúc của mạng nơron có sẵn). Để làm được điều này thì mạng nơron phải sử dụng các trọng số điều chỉnh, với nhiều phương pháp học khác nhau để có thể tính toán gần đúng ma trận W cần tìm đặc trưng cho mạng. Sau đây là 3 phương pháp học:

1.4.2.1. Học có giám sát

Học có giám sát: Là quá trình học có tín hiệu chỉ đạo bên ngoài d. Trong học có giám sát, thì tại mỗi thời điểm khi đầu vào được cung cấp tới mạng nơron, phản ứng đầu ra mong muốn d tương ứng của hệ thống được đưa ra, khi mỗi đầu vào $x^{(k)}$ được đặt vào mạng, đầu ra mong muốn tương ứng $d^{(k)}$ cũng được cung cấp tới mạng. Hiệu giữa đầu ra thực $y^{(k)}$ và đầu ra mong muốn $d^{(k)}$ được đo trong máy phát tín hiệu lỗi. Hệ thống này sẽ tạo ra tín hiệu lỗi cho mạng để hiệu chỉnh các trọng số của mạng, và với các hiệu chỉnh này thì đầu ra thực sẽ tiến sát với đầu ra mong muốn.

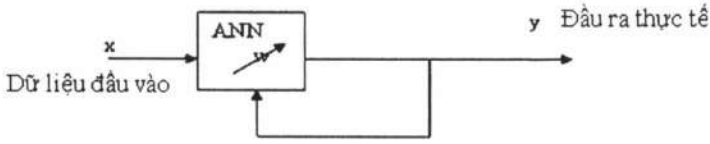


Hình 1.11. **Học có giám sát**

1.4.2.2. Học củng cố

Tín hiệu chủ đạo d có thể lấy từ môi trường bên ngoài, nhưng tín hiệu này không được đầy đủ, mà chỉ có một vài bit đại diện có tính chất kiểm tra quá trình tốt hay xấu. Học củng cố cũng là một dạng của học có giám sát, bởi vì mạng vẫn nhận một số tín hiệu từ bên ngoài. Nhưng tín hiệu phản hồi chỉ mang tính chất đánh giá hơn là mạng tính chất chỉ dẫn. Nó cho biết mức độ tốt hay xấu của một đầu ra đặc biệt. Tín hiệu củng cố bên ngoài thường được xử lý bằng máy phát tín hiệu đánh giá để tạo ra nhiều hơn nữa các thông tin tín hiệu đánh giá, sau đó dùng để điều chỉnh các trọng số với mục đích đạt được tín hiệu đánh giá tốt hơn.

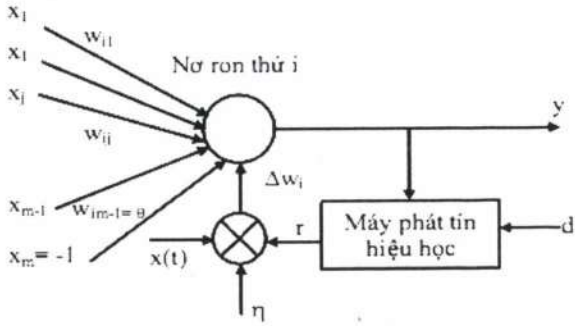
1.4.2.3. Học không có giám sát



Hình 1.12. **Học không có giám sát**

Trong phần học không có giám sát, sẽ không có thầy hướng dẫn, tức là không có tín hiệu d cung cấp tới mạch phản hồi. Điều này cho thấy, ta sẽ không biết đầu ra đạt giá trị gì. Với loại này, thì các nơron tự xoay xở với các dữ liệu mẫu mà nó có được, chứ không có "thầy" gợi ý cần luyện theo hướng nào. Mạng phải tự khám phá mẫu, đặc tính, sự tương quan hay loại đầu vào. Trong khi khám phá những đặc tính này, tham số của mạng sẽ bị thay đổi. Quá trình này được gọi là tự tổ chức. Một ví dụ điển hình là quá trình phân loại đối tượng không có thầy, những lớp thích hợp được hình thành bằng cách khám phá sự tương tự và không tương tự trong số các đối tượng.

Hình 1.13 mô tả cấu trúc chung của quá trình học của ba phương pháp học đã nêu ở trên. Trong tín hiệu vào x_j (j = 1, 2,..., m), có thể được lấy từ đầu ra của các nơron khác hoặc có thể được lấy ra từ bên ngoài. Trọng số của nơron thứ i được thay đổi tùy theo tín hiệu ở đầu vào mà nó thu nhận giá trị đầu ra của nó.



Hình 1.13. **Sơ đồ cấu trúc chung của quá trình học**

Dạng tổng quát của luật học trọng số của mạng nơron cho biết số gia của vectơ w_i là Δw_i tỉ lệ với tín hiệu học r và tín hiệu đầu vào x(t).

$$\Delta w_i(t) = \eta.r.x(t) \tag{1.12}$$

η – một số dương và được gọi là hằng số học dùng để xác định tốc độ học;

r – tín hiệu học và phụ thuộc:

$$r = f_r(w_i, x, d_i). \tag{1.13}$$

Từ hình 1.13 ta thấy, vectơ trọng số $w_i = [w_{i1}, w_{i2},..., w_{im}]^T$ có số gia tỷ lệ với tín hiệu vào x và tín hiệu học r. Vectơ trọng số ở thời điểm (t+1) được tính như sau:

$$w_i(t+1) = w_i(t) + \eta f_r(w_i(t),x(t),d(t)).x(t). \tag{1.14}$$

Phương trình liên quan đến sự biến đổi trọng số trong mạng nơron rời rạc và tương ứng với sự thay đổi trọng số trong mạng nơron liên tục theo biểu thức sau:

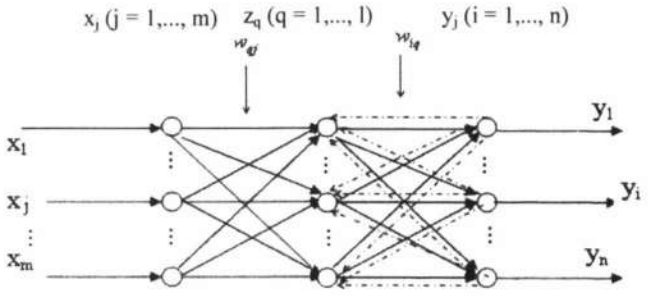
$$\frac{dw_i}{dt} = \eta.r.x(t) \tag{1.15}$$

Vấn đề quan trọng trong việc phân biệt luật học cập nhật trọng số có giám sát hay không có giám sát, hoặc học củng cố là tín hiệu học r. Như vậy, đối với tín hiệu học r thì làm thế nào để thay đổi hoặc cập nhật trọng số trong mạng nơron.

1.5. THUẬT TOÁN LAN TRUYỀN NGƯỢC

Thuật toán lan truyền ngược được ứng dụng để giải các bài toán điều khiển các hệ phi tuyến phức tạp và bất ổn định. Lan truyền ngược là một phương pháp cho phép xác định tập trọng số tốt nhất của mạng để giải một bài toán đã cho. Việc áp dụng phương pháp lan truyền ngược là một quá trình lặp đi lặp lại nhiều lần hai tiến trình chính: lan truyền tiến để thực hiện ánh xạ và lan truyền ngược sai số để cập nhật các trọng số. Các trọng số của mạng là các hệ số của mô hình. Phương pháp giảm gradient được dùng để cập nhật những hệ số này sao cho giảm thiểu được sai số của mô hình.

Xét lớp mạng 3 lớp như hình 1.14



Hình 1.14. **Mạng 3 lớp lan truyền ngược**

Thuật toán:

Đầu tiên ta cho lan truyền thẳng suốt trong mạng, qua các phần tử nơron và được tiếp tục với các hàm kích hoạt của phần tử nơron. Các mạng được nghiên cứu cùng với thuật toán học lan truyền ngược được gọi là mạng lan truyền ngược.

Huấn luyện các cặp vào/ra.

$$\{(x^{(k)}, d^{(k)})\},\ k = 1, 2,..., p$$

Thuật toán cung cấp một thủ tục cho việc thay đổi các vectơ trọng số trong mạng, đầu ra của mạng được lan truyền ngược trở lại lớp đầu vào cho đúng các mẫu. Cơ sở cho việc cập nhật các trọng số là phương pháp độ dốc Gradient.

Với cặp vào ra $(x^{(k)}, d^{(k)})$, thuật toán lan truyền ngược thực hiện các bước như sau:

– Đầu tiên, mẫu $x^{(k)}$ được lan truyền từ lớp đầu vào qua các lớp ẩn đi đến lớp đầu ra có kết quả là $y^{(k)}$. Sau đó, sai số giữa $y^{(k)}$ và $d^{(k)}$ được lan truyền ngược trở lại từ lớp đầu ra tới lớp đầu vào để cập nhật trọng số. Hình 1.14 diễn giải thuật toán lan truyền ngược. Kết quả có thể mở rộng sang mạng nơron nhiều lớp.

– Trên hình 1.14 có m phần tử nơron đầu vào, l phần tử nơron ở lớp ẩn, và n phần tử nơron ở lớp đầu ra. Đường nét liền diễn tả lan truyền thẳng của các tín hiệu, đường nét đứt diễn tả lan truyền ngược của các sai số. Đầu tiên huấn luyện vào cặp vào/ra ký hiệu (x, d) để cho đơn giản ta bỏ chỉ số k.

Khi một mẫu đầu vào x được đưa vào thì các phần tử trong mạng sẽ được tính như sau:

+ Đầu vào phần tử q của lớp ẩn sẽ được tính theo phương trình 1.16

$$net_q = \sum_{j=1}^{m} v_{qj} x_j \tag{1.16}$$

+ Phương trình đầu ra của q sẽ là:

$$z_q = a(net_q) = a\left(\sum_{j=1}^{m} v_{qj} x_j\right) \tag{1.17}$$

+ Đầu vào phần tử thứ i của lớp đầu ra sẽ là:

$$net_i = \sum_{q=1}^{l} w_{iq} z_q = \sum_{q=1}^{l} w_{iq} a\left(\sum_{j=1}^{m} v_{qj} x_j\right) \tag{1.18}$$

Phương trình đầu ra của phần tử nơron thứ i sẽ là:

$$y_i = a(net_i) = a\left(\sum_{q=1}^{l} w_{iq} z_q\right) = a\left(\sum_{q=1}^{l} w_{iq} . a\left(\sum_{j=1}^{m} v_{qj} x_j\right)\right) \tag{1.19}$$

Các chỉ số trên được tính toán cho sự lan truyền tiến của các tín hiệu đầu vào xuyên suốt qua các lớp mạng nơron. Trước khi đề cập đến các tín hiệu sai số của sự lan truyền ngược, ta sẽ định nghĩa một hàm mục tiêu như sau:

$$E(w) = \frac{1}{2}\sum_{i=1}^{n}(d_i - y_i)^2 = \frac{1}{2}\sum_{i=1}^{n}\left[d_i - a(net_i)\right]^2 = \frac{1}{2}\sum_{i=1}^{n}\left[d_i - a\left(\sum_{q=1}^{l} w_{iq} z_q\right)\right]^2 \tag{1.20}$$

Sau đó, theo phương pháp độ dốc Gradient, các trọng số nối giữa lớp ẩn và lớp đầu ra được cập nhật bởi Δw_{iq}, và nó được tính theo công thức sau:

$$\Delta w_{iq} = -\eta \frac{\partial E}{\partial w_{iq}} \tag{1.21}$$

Sử dụng các công thức (1.21-1.23) và thay đổi luật với $\frac{\partial E}{\partial w_{iq}}$, ta có:

$$\Delta w_{iq} = -\eta\left[\frac{\partial E}{\partial y_i}\right]\left[\frac{\partial y_i}{\partial net_i}\right]\left[\frac{\partial net_i}{\partial net_i}\right]\left[\frac{\partial net_i}{\partial w_{iq}}\right] = \left[d_i - y_i\right]\left[a'(net_i)\right]\left[z_q\right] = \eta\delta_{oi}z_q \tag{1.22}$$

Trong đó, δ_{oi} là tín hiệu sai số, chỉ số dưới thứ hai là điểm thứ i trong lớp đầu ra. Sai số tín hiệu được định nghĩa bởi:

$$\delta_{oi} = -\frac{\partial E}{\partial net_i} = -\left[\frac{\partial E}{\partial y_i}\right]\left[\frac{\partial y_i}{\partial net_i}\right] = \left[d_i - y_i\right]\left[a'(net_i)\right] \tag{1.23}$$

Trong đó, net_i là đầu vào của phần tử nơron thứ i trong lớp đầu ra và

$$a'(net_i) = \frac{\partial a(net_i)}{\partial net_i}. \tag{1.24}$$

Bây giờ ta phải tính đầu ra z_q của lớp ẩn:

Với trọng số nối giữa đầu vào và các lớp ẩn, ta sử dụng thay đổi luật cùng phương pháp độ dốc Gradient, ta cập nhật trọng số để kết nối giữa phần tử thứ j của lớp đầu vào với phần tử thứ q của lớp ẩn. Khi đó:

$$\Delta v_{qj} = -\eta\left[\frac{\partial E}{\partial v_{qj}}\right] = -\eta\left[\frac{\partial E}{\partial net_q}\right]\left[\frac{\partial net_q}{\partial v_{qj}}\right] = -\eta\left[\frac{\partial E}{\partial z_q}\right]\left[\frac{\partial z_q}{\partial net_q}\right]\left[\frac{\partial net_q}{\partial v_{qj}}\right] \tag{1.25}$$

Từ công thức (1.20), thì mỗi sai số $[d_i-y_i]$, với i=1,2,...,n là một hàm của z_q.

Đánh giá thay đổi luật ta có:

$$\Delta v_{qj} = -\eta\sum_{i=1}^{n}\left[\left(d_i - y_i\right).a'(net_i).w_{iq}\right].a'(net_q).x_j \tag{1.26}$$

Sử dụng công thức (1.23), ta có thể viết lại công thức (1.26) như sau:

$$\Delta v_{qj} = -\eta\sum_{i=1}^{n}\left[\delta_{oi}.w_{iq}\right].a'(net_q).x_j = \eta\partial_{hq}x_j \tag{1.27}$$

Ở đây, ∂_{hq} là sai số tín hiệu của phần tử thứ q của lớp ẩn và được định nghĩa như sau:

$$\partial_{hq} = -\frac{\partial E}{\partial net_q} = -\left[\frac{\partial E}{\partial z_q}\right]\left[\frac{\partial z_q}{\partial net_q}\right] = a'(net_q)\sum_{i=1}^{n}\delta_{oi}w_{iq} \tag{1.28}$$

Trong đó, net_q là đầu vào phần tử thứ q của lớp ẩn.

Tín hiệu sai số của một phần tử trong lớp ẩn khác so với tín hiệu sai số của một phần tử trong lớp đầu ra. Do có sự khác nhau này, nên các thủ tục cập nhật các trọng số trên được gọi là luật học delta tổng quát. Chúng ta xem xét công thức (1.28), sai số tín hiệu ∂_{hq} của phần tử lớp ẩn q có thể được xác định trong các mẫu của các tín hiệu sai số δ_{oi} của các phần tử ở lớp ra thứ i (y_i) cung ứng. Các hệ số là các trọng số được sử dụng cho lan truyền thẳng, nhưng ở đây chúng truyền các tín hiệu sai số (δ_{oi}) ngược trở lại, đó chính là các đường nét đứt trong hình 1.14. Điều này đã chứng tỏ được đặc điểm quan trọng của thuật toán lan truyền ngược – luật cập nhật cục bộ, đây chính là tính toán trọng số thay đổi dựa vào sự kết nối, và chúng ta chỉ cần giá trị ở hai đầu của kết nối này.

Sự đạo hàm ở trên có thể dễ dàng mở rộng cho mạng có nhiều hơn một lớp ẩn, bằng cách sử dụng chuỗi luật liên tiếp. Trong trường hợp chung, với số lớp tùy ý, thì sự lan truyền ngược được cập nhật luật ở dạng sau:

$$\Delta w_{ij} = \eta\delta_i x_j = \eta\delta_{output-i} x_{input-j} \tag{1.29}$$

Ở đây, (output-i) và (input-j) quy vào hai đầu của sự kết nối từ phần tử thứ j tới phần tử thứ i, x_j là đầu vào cuối cùng kích hoạt từ một phần tử lớp ẩn, hoặc từ một đầu vào bên ngoài. Ngoài ra, δ_i là tín hiệu học được định nghĩa bởi công thức (1.23) với đầu ra hoặc lớp cuối cùng của các trọng số kết nối, và được định nghĩa bởi công thức (1.28) cho tất cả các lớp khác. Khi hàm sigmoid lưỡng cực được sử dụng làm hàm kích hoạt, đồng thời sử dụng (1.23) và (1.28) ta có hàm y được xác định như sau:

$$y = a(net) = \frac{1}{1+e^{-net}} - 1 \tag{1.30}$$

Khi đó ta có: $$a'(net) = \frac{\partial a(net)}{\partial net} = \frac{1}{2}\left[1 - a^2(net)\right] = \frac{1}{2}(1-y^2) \tag{1.31}$$

$$\delta_{oi} = \frac{1}{2}(1-y^2)\left[d_i - y_i\right] \tag{1.32}$$

$$\delta_{hq} = \frac{1}{2}(1-z^2)\sum_{i=1}^{n}\delta_{oi} w_{iq} \tag{1.33}$$

Thuật toán lan truyền ngược

Xem xét một mạng với Q lớp lan truyền ngược, q = 1, 2,..., Q; với ${}^{q}net_i$ và ${}^{q}y_i$ lần lượt là đầu vào và đầu ra của khối trong lớp thứ q. Mạng có m nơron đầu vào, l nơron ở lớp ẩn, và n nơron đầu ra. Với ${}^{q}w_{ij}$ là trọng số nối từ ${}^{q-1}w_j$ đến ${}^{q}y_i$.

Đầu vào: các cặp huấn luyện $\{x^{(k)}, d^{(k)} \mid k=1, 2,..., p\}$, ở đó giá trị đầu vào của phần tử cuối cùng bằng –1, tức là $x_{m+1}^{(k)} = -1$.

Bước 1 (Đặt giá trị ban đầu)

– Lựa chọn bước tính (Hằng số học) $0 < \eta < 1$ và E_{max} (sai số lớn nhất cho phép).

– Chọn các trọng số ban đầu nối từ phần tử thứ j của lớp (q – 1) đến phần tử thứ i của lớp q là ${}^{q}w_{ij}$ có giá trị nhỏ và ngẫu nhiên.

– Cho sai số E = 0 và k = 1.

Bước 2 (Vòng lặp huấn luyện)

Áp dụng mẫu vào thứ k, cho lớp đầu vào q = 1. Khi đó ta có:

$${}^{q}y_i = {}^{1}y_i = x_i^{(k)} \text{ cho tất cả các } i = 1, 2, 3,\ldots, m. \qquad (1.34)$$

Bước 3 (Lan truyền thẳng)

Lan truyền tín hiệu thẳng xuyên suốt mạng sử dụng công thức (1.35) cho mỗi i và q cho tới khi các đầu ra của lớp đầu ra ${}^{Q}y_i$ được thực hiện.

$${}^{q}y_i = a({}^{q}net_i) = a\left(\sum_j {}^{q}w_{ij}^{q-1} y_j\right) \qquad (1.35)$$

Bước 4 (Đo lường sai số đầu ra)

Tính toán giá trị sai lệch và tín hiệu sai lệch ${}^{Q}\delta_i$ cho lớp đầu ra như sau:

$$E = \frac{1}{2}\sum_{i=1}^{n}(d_i^{(k)} - {}^{Q}y_i)^2 + E; \qquad (1.36)$$

$${}^{Q}\delta_i = (d_i^{(k)} - {}^{Q}y_i).a'({}^{Q}net_i) \qquad (1.37)$$

trong đó: ${}^{Q}\delta_i$ – tín hiệu sai lệch của nơron thứ i cho lớp ra Q;

$a'({}^{Q}net_i)$ – đạo hàm của hàm truyền a(.) theo tổng trọng số của phần tử i của lớp đầu ra là ${}^{Q}net_i$.

$$a'({}^{Q}net_i) = \frac{da}{d({}^{Q}net_i)} \quad (1.38)$$

Bước 5 (Lan truyền ngược sai số)

Các sai số lan truyền ngược với mục đích để cập nhật các trọng số và tính toán các tín hiệu sai lệch ${}^{q-1}\delta_i$ cho các lớp xử lý:

$$\Delta^{q}w_{ij} = \eta \,.\, {}^{q}\delta_i \,.\, {}^{q-1}y_j \,;\; {}^{q}w_{ij}^{new} = {}^{q}w_{ij}^{old} + \Delta^{q}w_{ij} \quad (1.39)$$

$${}^{q-1}\delta_i = a'({}^{q-1}net_i)\sum_j {}^{q}w_{ji}\,{}^{q}\delta_j \,;\; \text{với } q = Q, Q-1, \ldots, 2 \quad (1.40)$$

Trong đó:

$\Delta^{q}w_{ij}$ – sai lệch tại thời điểm tính của giá trị trọng số liên kết cập nhật mới và cũ, liên kết từ phần tử thứ j của lớp q – 1 đến phần tử i của lớp q;

${}^{q}w_{ij}^{new}$ – giá trị trọng số liên kết cập nhật mới từ phần tử thứ j của lớp (q – 1) đến phần tử i của lớp q;

${}^{q}w_{ij}^{old}$ – giá trị trọng số liên kết cũ từ phần tử thứ j của lớp (q – 1) đến phần tử i của lớp q.

${}^{q-1}y_j$ – tín hiệu ra của phần tử j của lớp (q – 1).

Bước 6 (Sau mỗi vòng lặp)

Kiểm tra xem đã lặp hết các giá trị mẫu huấn luyện chưa, nếu chưa quay vòng hết (tức là k < p) tăng k = k + 1, và nhảy tới bước 1, ngược lại (tức k = p) thì chuyển sang bước 6.

Bước 7 (Kiểm tra tổng sai số)

Kiểm tra sự khác nhau giữa tổng sai số và sai số cho phép:

– Nếu tổng sai số nhỏ hơn sai số cho phép (tức là $E < E_{max}$) thì kết thúc quá trình huấn luyện và ghi lại các giá trị trọng số cuối cùng.

– Trái lại, gán E = 0, k = 1 và bắt đầu một quá trình huấn luyện mới bằng cách nhảy tới bước 1.

Ưu điểm sử dụng mô hình mạng nơron

– Khả năng của các quá trình xử lý song song và phân tán: Có thể đưa vào mạng một lượng lớn các nơron liên kết với nhau theo các lược đồ với các cấu trúc khác nhau.

– Khả năng thích nghi và tự tổ chức: Về đặc trưng này người ta đề cập tới khả năng xử lý thích nghi và điều chỉnh bền vững dựa vào các thuật toán học thích nghi và các quy tắc tự tổ chức. Chỉ cần đưa vào mạng một tập mẫu dữ liệu trong quá trình học là mạng có khả năng phát hiện những ràng buộc dữ liệu và áp dụng những ràng buộc này trong quá trình sử dụng mà không cần phải thêm các tri thức về miền ứng dụng.

– Khả năng dung thứ lỗi: Có khả năng bắt chước dung thứ lỗi của não theo nghĩa các hệ thống có thể tiếp tục làm việc và điều chỉnh khi nhận tín hiệu vào một phần thông tin bị sai lệch hoặc bị thiếu. Mạng có thể chấp nhận những dữ liệu mẫu không hoàn toàn chính xác tuyệt đối mà vẫn đảm bảo được phần nhiều tính đúng đắn của bài toán. Điều này làm giảm nhẹ rất nhiều quá trình sàng lọc.

– Xử lý các quá trình phi tuyến: Đặc trưng này rất quan trọng, ví dụ như trong xấp xỉ mạng, miễn nhiễu (chấp nhận nhiễu) và khả năng phân lớp. Với ưu điểm này cho phép chúng ta dễ dàng xây dựng các mô hình thích nghi mà trong đó có sự thay đổi liên tục về quy luật dữ liệu có thể dễ dàng được cập nhật trong quá trình học lại của mạng.

Phần II

MỘT SỐ MẠNG NƠRON THƯỜNG SỬ DỤNG TRONG NHẬN DẠNG PHÂN LOẠI

Chương 2

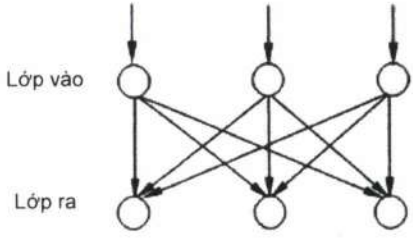
MẠNG NƠRON

2.1. MẠNG NƠRON MỘT LỚP

2.1.1. Mạng hopfield

2.1.1.1. Mô hình mạng

Năm 1982 nhà vật lý người Mỹ J.J. Hopfield đã đề xuất mô hình mạng nơ ron một lớp như hình 2.1.



Hình 2.1. **Mạng Hopfield**

Hàm kích hoạt được dùng tại các nơ ron là hàm:

$$out_j = sign\left(Net_j\right) = sign\left(\sum_{i=1}^{m} w_{ji} x_i\right) \tag{2.1}$$

2.1.1.2. Huấn luyện mạng

Mạng Hopfield (HF) học dựa trên nguyên tắc có giám sát. Giả sử có p mẫu học tương ứng với các vectơ tín hiệu vào X_s, s = 1, p. Mạng sẽ xác định bộ trọng số W sao cho

$$X_s = \text{Tinh}\,(X_s\,,\,W) \text{ với mọi } s = 1, p \tag{2.2}$$

Ta xây dựng ma trận trọng số W như sau : $W = (w_{ij})$ với

$$w_{ji} = \begin{cases} \sum_{s=1}^{p} x_{sj}x_{si} & \text{Nếu } i \neq j \\ 0 & \text{Nếu } i = j \end{cases} \tag{2.3}$$

Trong đó $X_s = (x_{s1}, ..., x_{sm})$. (2.4)

Một cách trực quan, trọng số liên kết ω_{ji} sẽ tăng thêm một lượng là 1 (tương ứng với số hạng $x_{sj}.x_{si}$) nếu cả hai thành phần thứ i và thứ j của mẫu học X_s bằng nhau. Khi có mẫu học mới X_{p+1} ta chỉ cần xét các thành phần thứ i và thứ j của nó để cập nhật giá trị cho w_{ji}. Có thể chứng minh được với ma trận W được xác định như trong biểu thức 2.4, ta sẽ có được biểu thức 2.4. Nói cách khác, mạng đã "học thuộc" các ví dụ mẫu $\{X_s\}$.

2.1.1.3. Ứng dụng và đặc điểm sử dụng mạng

– Ứng dụng mạng: sử dụng trong các bài toán nhận dạng ảnh.

– Đặc điểm sử dụng:

+ Mạng Hopfield cho phép tạo ánh xạ tự kết hợp trên các tập dữ liệu;

+ Dữ liệu vào, ra có giá trị lưỡng cực;

+ Học có giám sát;

+ Mạng cho phép phục hồi dữ liệu;

+ Khả năng nhớ mẫu phụ thuộc vào số nơ ron của mạng.

2.1.2. Mạng ABAM (Adaptive Bidirectional Associative Memory Neural Network)

2.1.2.1. Mô hình mạng

Theo một nghĩa nào đó, mạng ABAM có những nét giống mạng Hopfield.

– Chúng cùng là mạng 1 lớp.

– Tín hiệu vào có thể là nhị phân, hoặc lưỡng cực.

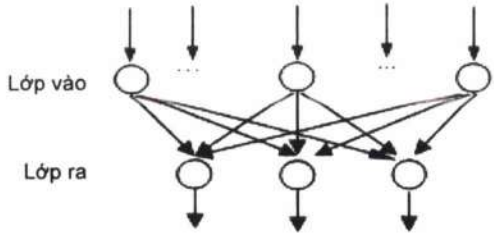
– Việc xác định ma trận trọng số ban đầu giống nhau.

Điểm khác giữa 2 loại mạng chính là ở phạm vi bài toán có thể giải quyết và cách xác định các trọng số cho phù hợp với các bài toán đó.

Mạng Hopfield được xác định đúng một lần và được dùng cho tất cả các bước tính toán. Kích thước của ảnh (số điểm ảnh trong mỗi mẫu) sẽ xác định số nơ ron và số trọng số liên kết, trong khi đó số mẫu học và hình dạng của chúng sẽ xác định giá trị các trọng số.

$$W_{ji}^{(0)} = \sum_{s=1}^{p} x_{si} y_{sj} \tag{2.5}$$

– Với mạng ABAM, ma trận trọng số không bắt buộc phải vuông. Thông thường, số nơron ra ít hơn nhiều số nơron vào. Ban đầu, ma trận trọng số được xác định dựa trên các tập mẫu $\{(X_s,Y_s)\}$ giống như đối với mạng Hopfield nghĩa là: Ở các bước tiếp theo trong quá trình học, ma trận trọng số $W^{(t)}$ được thay đổi cho phù hợp sao cho tạo ra sự kết hợp thực sự 2 chiều giữa tín hiệu vào và tín hiệu ra trong tập mẫu học. Mô hình mạng có dạng như hình 2.2.



Hình 2.2. **Mạng ABAM**

2.1.2.2. Huấn luyện mạng

Giả sử có tập mẫu học $\{(X_s,Y_s)\}$, sơ đồ quá trình học được thể hiện như sau:

Lặp $\{(X_s,Y_s)\} \rightarrow$ ma trận $W^{(0)}$

$$W^{(0)T} \rightarrow Y_s^{(1)}$$

$$Y_s W^{(0)} \rightarrow X_s^{(1)}$$

$$\{(X_s^{(1)}, Y_s^{(1)})\} \rightarrow \text{ma trận } W^{(1)}$$

$$X_s W^{(1)T} \rightarrow Y_s^{(2)}$$

$$Y_s W^{(1)} \rightarrow X_s^{(2)}$$

cho đến khi $\{(X_s^{(t)}, Y_s^{(t)})\} = \{(X_s, Y_s)\}$

Từ tập các mẫu học $\{(X_s^{(t)}, Y_s^{(t)})\}$ xác định ma trận $W^{(t)}$ theo 2.5

Ban đầu $X_s^{(0)} = X_s$ và $Y_s^{(0)} = Y_s$.

+ Tạo tập mẫu học mới $\{(X_s^{(t+1)}, Y_s^{(t+1)})\}$ nhờ nhân ma trận $W^{(t)}$ và chuyển vị của nó $W^{(t)T}$ với các mẫu học gốc $\{(X_s, Y_s)\}$.

+ So sánh tập mẫu học mới và mẫu học gốc. Nếu trùng nhau thì dừng. Ngược lại, tiếp tục quá trình lặp.

2.1.2.3. Ứng dụng và đặc điểm sử dụng mạng

– Ứng dụng: Được sử dụng trong nhận dạng và phân loại ảnh.

– Đặc điểm: Mạng có thể sử dụng theo 2 chiều.

+ Cấu trúc mạng đơn giản, có thể dùng để phân loại, nhận dạng đối tượng;

+ Dữ liệu vào; Nhị phân hoặc lưỡng cực;

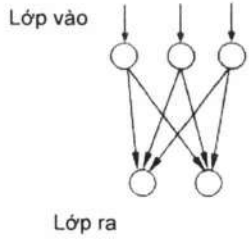
+ Cơ chế học có giám sát;

+ Có thể phục hồi và nhận biết các đối tượng bị biến dạng.

2.1.3. Mạng Perceptron

2.1.3.1. Mô hình mạng

Mạng Perceptron do F Rosenblatt đề xuất năm 1960. Đây là mạng truyền thẳng một lớp có một hoặc nhiều đầu ra. Để đơn giản trong cách trình bày ta giả sử mạng có một đầu ra.



Hình 2.3. **Mạng Perceptron**

Hàm kích hoạt

$$out = \begin{cases} 1 \text{ nếu } \Sigma w_j s_j \geq 0 \\ 0 \text{ nếu ngược lại} \end{cases} \quad (2.6)$$

2.1.3.2. Huấn luyện mạng

Mạng học dựa trên nguyên tắc có giám sát với tập mẫu $\{(X_s, Y_s)\}$. Ý tưởng cơ bản của quá trình huấn luyện mạng là xác định bộ trọng số W sao cho:

$$out_s = Tinh(X_s, W) = Y_s \text{ đối với mọi mẫu học s.}$$

Ban đầu các trọng số được gán ngẫu nhiên một giá trị. Sau đó hiệu chỉnh các trọng số sao cho phù hợp với các mẫu học, làm giảm sai số giữa giá trị quan sát Y_s với giá trị tính toán out_s.

Các bước tiến hành như sau:

– Xác định ngẫu nhiên bộ trọng số.

– Với mỗi mẫu học (X_s, Y_s), $X_s = (X_{s1},..., X_{sn})$, thực hiện các bước:

+ Tính giá trị out_s theo công thức (2.6);

+ Xác định $Err = Y_s - out_s$. Hiệu chỉnh các trọng số $W_j = W_j + \alpha X_{sj}*Err$, trong đó α là hằng số học.

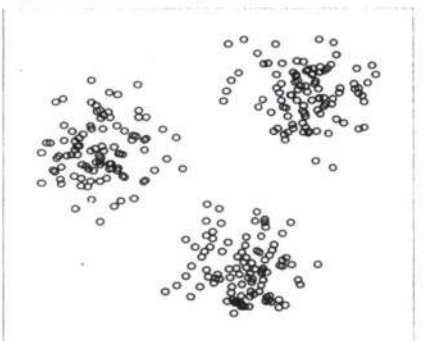
Ứng dụng: Sử dụng trong nhận dạng ảnh và làm công cụ tính toán các hàm lôgic.

2.1.4. Mạng Kohonen

2.1.4.1. Cấu trúc và nguyên lý hoạt động

Các mạng Kohonen kinh điển được sử dụng một cách rộng rãi để sắp xếp và phân loại dữ liệu trên cơ sở của các vectơ đầu vào x đưa vào mạng nơron. Đầu vào ta có một bộ số liệu gồm *p* vectơ đa chiều:

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iN}] \in \mathbb{R}^N; i = 1 \to p. \quad (2.7)$$



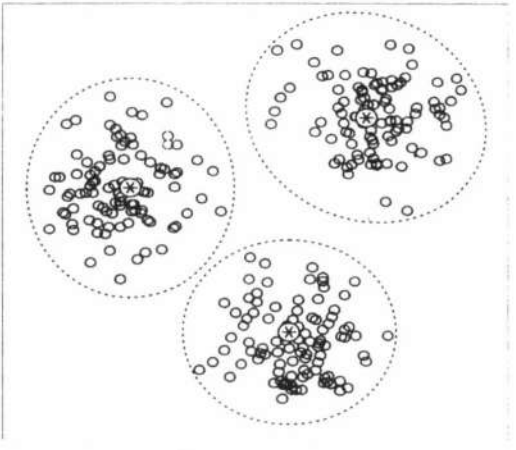
Hình 2.4. **Phân bố số liệu ví dụ trên không gian hai chiều**

Ví dụ trên hình 2.4 là một bộ các điểm trên mặt phẳng hai chiều có xu hướng tập trung thành 3 nhóm.

Nhiệm vụ phân nhóm các số liệu này thành M nhóm, mỗi nhóm được đặc trưng bởi một trọng tâm (*centre*)

$$\mathbf{c}_j = \left[c_{j1}, c_{j2}, \ldots, c_{jN}\right] \in \mathbb{R}^N; j = 1 \to p \tag{2.8}$$

Ví dụ phân chia và nhóm các số liệu từ hình 2.4 thành 3 nhóm ta có kết quả như hình 2.5.



Hình 2.5. **Các số liệu được chia thành 3 nhóm (đặc trưng bởi các viền bao và các trọng tâm '*')**

Với bài toán thuộc dạng "tự tổ chức" (*self-organizing*) thì ta chỉ có thông tin về **x**, chứ không có các thông tin khác, trọng tâm của các nhóm được xác định chủ yếu trên nguyên tắc: "các vectơ có khoảng cách gần nhau sẽ được ưu tiên ghép vào cùng một nhóm". Thước đo khoảng cách giữa các vectơ chủ yếu là sử dụng công thức Ơ-clít:

$$x, c \in \mathbb{R}^N : d(x,c) = \|x - c\| = \sqrt{\sum_{i=1}^{N} (x_i - c_i)^2} \tag{2.9}$$

Tuy nhiên trong các công trình về mạng tự tổ chức ta cũng có thể gặp các công thức tính khoảng cách khác như:

1. Khoảng cách tích vô hướng:

$$d(x,c) = 1 - x \cdot c = 1 - \|x\| \cdot \|c\| \cdot \cos(x,c)$$

2. Khoảng cách Manhattan: $d(x,c) = \sum_{i=1}^{N} |x_i - c_i|$

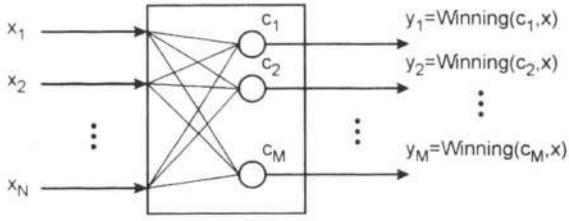
3. Khoảng cách Chebyshev: $d(x,c) = \max_{i=1 \to N} |x_i - c_i|$

4. Khoảng cách Minkowski: $d(x,c) = \sqrt[m]{\sum_{i=1}^{N} |x_i - c_i|^m}$

Trong trường hợp ta có M trọng tâm $c_i, i = 1 \to M$ và một vectơ **x** thì trong quá trình hoạt động cạnh tranh, trọng tâm chiến thắng là trọng tâm có khoảng cách ngắn nhất tới điểm đang xét.

$$\|x - c_{win}\| = \min_{i=1 \to M} \|x - c_i\| \tag{2.10}$$

Ta có thể mô tả cấu trúc xử lý các vectơ đầu vào gồm N thành phần và hệ M trọng tâm $\mathbf{c}_i$ như trên hình 2.6.



Hình 2.6. **Cấu trúc xử lý các vectơ đầu vào của mạng Kohonen kinh điển**

Đây là cấu trúc mạng truyền thẳng một lớp. Tất cả N đầu vào được nối với tất cả M đầu ra thông qua trọng số c_{ij}. Số lượng đầu vào bằng với số chiều của vectơ **x**, trong khi số lượng đầu ra bằng với số lượng của nhóm mà dữ liệu được chia thành. Tọa độ thứ j của trọng tâm thứ i c_{ij} được coi là hệ số đặc trưng của kênh nối đầu vào thứ j (là x_j) tới trọng tâm. Hệ số đặc trưng này trong các nghiên cứu về mạng nơron thường được gọi là trọng số ghép nối (*connection weight*) hay đơn giản là trọng số (*weight*). Vectơ đầu vào $x = [x_1, x_2, \ldots, x_N]$ và $c = [c_1, c_2, \ldots, c_N]$ thường được chuẩn hóa về độ dài 1 (tuy nhiên đây là giải pháp không bắt buộc). Để dễ dàng mô tả quá trình hoạt động của mạng, ngoài khái niệm khoảng cách ta còn dùng khái niệm "mức độ kích hoạt" của trọng tâm thứ j. Mức độ kích hoạt được xác định trên cơ sở một hàm nghịch biến với khoảng cách giữa trọng tâm đang xét và vectơ đầu vào đang xét. Khoảng cách càng nhỏ thì mức độ kích hoạt

càng lớn. Như vậy trọng tâm chiến thắng là trọng tâm có mức độ kích hoạt lớn nhất. Một số hàm kích hoạt thường được sử dụng là:

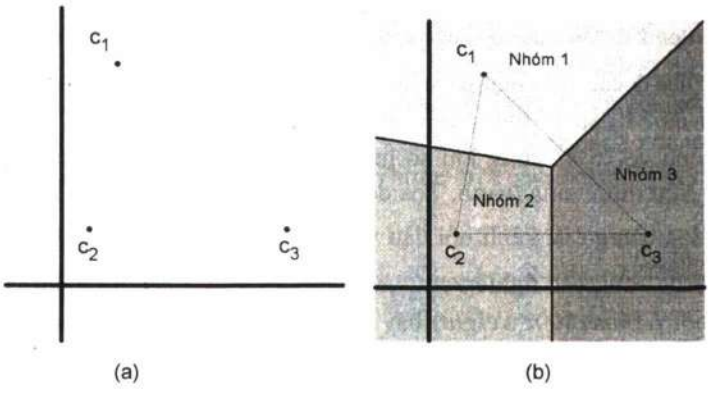
1. Hàm chuông: $\text{activation}_c(x) = \exp\left(-\frac{d^2(x,c)}{\sigma^2}\right)$

2. Hàm Gauss mở rộng: $\text{activation}_c(x) = \frac{1}{1+\left\|\frac{x-c}{\sigma}\right\|^{2b}}$

3. Hàm tam giác: $\text{activation}_c(x) = \begin{cases} 0 & \text{khi} \quad \|x-c\| > a \\ 1-\frac{1}{a}\|x-c\| & \text{khi} \quad \|x-c\| \le a \end{cases}$

Mỗi một trọng tâm sẽ xác định một vùng hoạt động của mình, đó là vùng tập hợp các điểm trong không gian mà khoảng cách tới trọng tâm đó là bé nhất so với khoảng cách tới các trọng tâm khác.

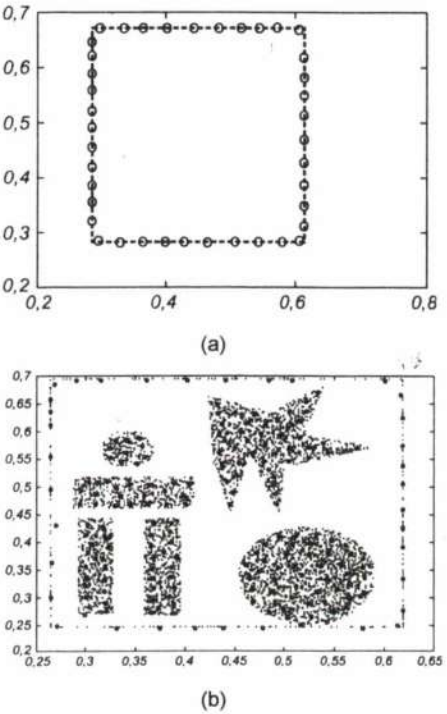
Ví dụ trên hình 2.7a với 3 trọng tâm, ta thấy không gian được chia thành 3 vùng như trên hình 2.7b. Các phân chia này (còn được gọi là phân chia Voronoi do tác giả Voronoi đề xuất lần đầu) có các đường biên giới là các đường trung trực giữa các cặp điểm trọng tâm (nếu ta sử dụng công thức khoảng cách Ơ-clít).



Hình 2.7. **Không gian với 3 trọng tâm được xác định (a) và các vùng "chiến thắng" của mỗi trọng tâm xác định theo công thức Ơ-clít và đồ thị Voronoi (b)**

Một phân chia trọng tâm tốt là trường hợp ta có các trọng tâm được đặt giữa các vùng có mật độ vectơ các điểm đầu vào cao. Trên hai hình 2.8

là ví dụ các phân bố như vậy. Hình 2.8a có các điểm đầu vào (dấu '.') phân bố theo các cạnh của hình vuông, các trọng tâm (dấu 'o') cũng nằm dọc theo các cạnh đó. Hình 2.8b có các điểm đầu vào (dấu '.') phân bố theo các hình dạng 2-D và ta cũng dễ dàng nhận thấy các trọng tâm (dấu '•') cũng được phân bố như vậy.



Hình 2.8. **Ví dụ phân bố các trọng tâm phân tán theo các vùng số liệu**

2.1.4.2. Các thuật toán học cơ bản của mạng Kohonen kinh điển

Trong mạng Kohonen, ta cần xác định vị trí của M trọng tâm khi được cho trước một tập hợp điểm **x** và số trọng tâm M. Các thuật toán xác định vị trí này được gọi là các thuật toán *học* của mạng Kohonen và được chia thành hai nhóm chính: Các thuật toán học trực tuyến (*online*) và các thuật toán học ngoại tuyến (*offline*). Đồng thời các trọng tâm còn được gọi là các *nơron* do việc tham gia vào quá trình học để điều chỉnh các thông số của mình.

2.1.4.2.1. Các thuật toán phân nhóm trực tuyến

Các giải pháp nơron cổ điển cho vấn đề phân nhóm, đã được bắt đầu trong các công trình của Kohonen, thường áp dụng cách tiếp cận trực tuyến, trong đó quá trình cập nhật của các trọng số được thực hiện sau mỗi quá trình biểu diễn của mỗi vectơ dữ liệu đầu vào x.

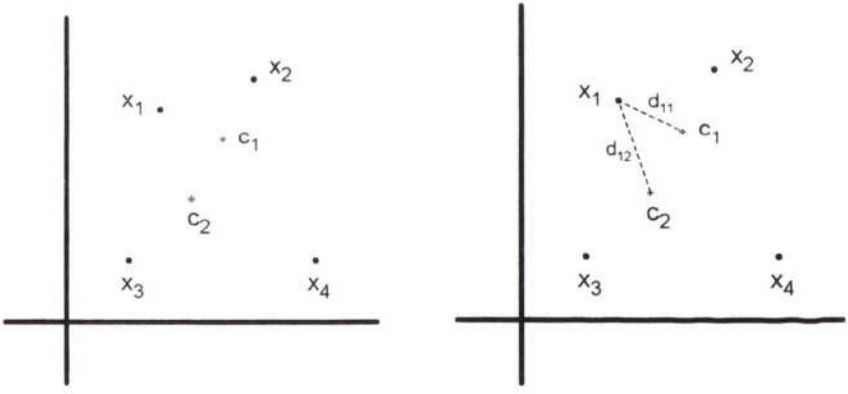
Trong quá trình học, các nơron tự sắp xếp, trong đó thuật toán tự sắp xếp được hình thành bằng trình tự của các hoạt động sau:

Bước 1: Đặt i = 1.

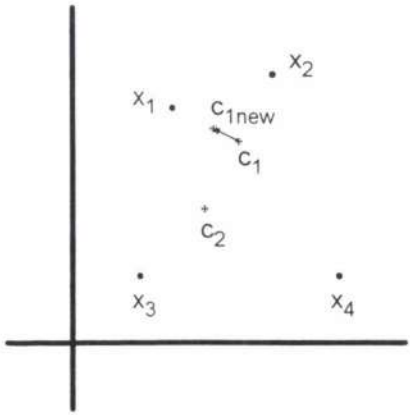
Bước 2: Biểu diễn vectơ $\mathbf{x}_i$ tới mạng.

Bước 3: Xác định các khoảng cách từ vectơ $\mathbf{x}_i$ tới các nơron trọng tâm c_j từ đó xác định nơron chiến thắng N_W là nơron có khoảng cách ngắn nhất tới vectơ x_i.

Bước 4: Cập nhật các trọng số của nơron chiến thắng và các nơron có liên quan tới nơron chiến thắng theo hướng dịch chuyển về gần phía vectơ đầu vào $\mathbf{x}_i$.



Hình 2.9. **Vị trí 4 vectơ đầu vào x_1, x_2, x_3, x_4 và hai trọng tâm c_1, c_2. Khi xét vectơ x_1 thì khoảng cách $d_{11} < d_{12}$ nên trọng tâm c_1 chiến thắng**



Hình 2.10. **Và trọng tâm c_1 được dịch chuyển về phía vectơ x_1 còn trọng tâm c_2 đứng yên**

Lặp lại trình tự này nhiều lần đưa mạng tới một trạng thái được sắp xếp, trong đó mỗi nơron biểu diễn một nhóm dữ liệu riêng biệt. Mô tả minh hoạ một bước học cho trường hợp 4 vectơ mẫu và hai trọng tâm được thể hiện trên hình 2.9 và hình 2.10.

Trong thuật toán trực tuyến chuẩn, chúng ta cập nhật trọng số của các nơron được tìm thấy trong lân cận nơron chiến thắng N_w, theo luật sau:

$$c_j(t+1) = c_j(t) + \eta_j(t)G(c_j, x(t))[x(t) - c_j(t)] \qquad (2.11)$$

trong đó: c_j – vectơ trọng số của nơron thứ j được tìm thấy trong lân cận của nơron chiến thắng;

η_j – hệ số học;

$G(c_j, x(t))$ – hàm lân cận của nơron thứ j và t là chỉ số thời gian rời rạc.

Các công thức khác nhau trong việc lựa chọn hàm lân cận $G(c_j, x(t))$ dẫn tới nhiều thuật toán học khác nhau. Một thuật học nổi tiếng là thuật toán Kohonen với hàm Gaussian, trong đó:

$$G(c_j, x) = \exp\left[-\frac{d^2(c_j, c_{Nw})}{\sigma^2}\right] \qquad (2.12)$$

trong đó: $d^2(c_j, x)$ – khoảng cách Ơ-clit giữa vectơ trọng số của nơron thứ j và nơron chiến thắng N_W;

σ – hệ số vùng lân cận và sẽ giảm theo quá trình học.

Một thuật toán mạnh khác được gọi là khí nơron, trong đó hàm lân cận được định nghĩa theo khoảng cách giữa vectơ đầu vào **x** và vectơ trọng số của nơron. Trong cách tiếp cận này, chúng ta sắp xếp các nơron theo những khoảng cách này, tức là

$$d_1 < d_2 ... < d_{M-1} < d_M \tag{2.13}$$

trong đó $d_i = \|\mathbf{x} - \mathbf{c}_{m(i)}\|$ với $i = 1 \rightarrow M$. Khi đó, giá trị của hàm lân cận được xác định như sau:

$$G(c_j, x) = e^{-\frac{m(j)}{\lambda}} \tag{2.14}$$

trong đó: m(i) là vị trí của nơron thứ *i* sau khi phân loại và λ là tham số giảm theo thời gian

$$\lambda(k) = \lambda_{max}\left(\frac{\lambda_{min}}{\lambda_{max}}\right)^{k/k_{max}} \tag{2.15}$$

Hệ số học η trong tất cả các cách tiếp cận thường giảm theo thời gian hoặc theo hàm mũ hoặc tuyến tính. Thuật toán khí nơron được đánh giá là một trong những phương pháp hiệu quả nhất đào tạo mạng Kohonen, cho phép thu được sự sắp xếp mạng tốt.

2.1.4.2.2. Thuật toán phân nhóm ngoại tuyến

Chế độ học ngoại tuyến (*offline*) còn được gọi là chế độ học toàn bộ (*batch-mode*) của thuật toán tự tổ chức, cũng thường được gọi là thuật toán K- trung bình hay thuật toán K- trung bình cứng, đưa ra một cơ chế đơn giản để tối thiểu hóa tổng các bình phương sai số với các nhóm K, với mỗi nhóm chứa một tập n mẫu tương tự nhau. Thuật toán này có thể được giới thiệu như sau:

– Chọn một tập các nhóm khởi đầu $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_M$, tùy ý.

– Gán các mẫu $\mathbf{x}_i$ tới các nhóm j sử dụng luật khoảng cách Ơ-clit tối thiểu, theo đó x_i thuộc về nhóm C_j nếu $\|x_i - c_j\| = \min_{k=1\rightarrow M}\|x_i - c_k\|$.

– Tính toán các nguyên mẫu nhóm mới là trung bình cộng của tất cả các vectơ trong nhóm i

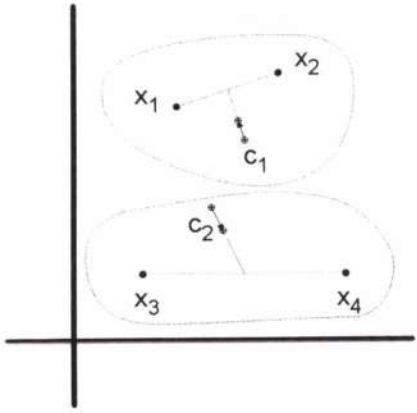
$$c_j = \frac{1}{n_j}\sum_{x_i \in C_j} x_i \tag{2.16}$$

trong đó: n_j chỉ số lượng vectơ $\mathbf{x}_i$ được gán với nhóm thứ j $\left(\sum_{i=1}^{M} n_i = p\right)$.

– Nếu trọng tâm có tọa độ được thay đổi đáng kể trong quá trình cập nhật trên, quay trở lại bước 2; nếu không thì dừng lại.

Thuật toán K- trung bình chia một tập hợp các vectơ **x** thành các **K** nhóm và tìm trọng tâm trong mỗi nhóm theo tiêu chí tối thiểu hóa hàm chi phí đo độ không đồng dạng hay đo khoảng cách. Giả định khoảng cách Ơ-clit được sử dụng, ta có hàm chi phí toàn phần có thể được xác định như sau:

$$E = \sum_{i=1}^{K}\left(\sum_{x_k \in C_i} \|x_k - c_i\|^2\right) \tag{2.17}$$



Hình 2.11. **Trong bước tính toán trên, các véc tơ x_1 và x_2 thuộc về nhóm c_1, các vectơ x_3 và x_4 thuộc về nhóm c_2, do đó trong quá trình cập nhật, c_1 sẽ dịch chuyển về phía trung điểm của x_1 và x_2 còn c_2 sẽ dịch chuyển về phía trung điểm của x_3 và x_4**

Với n điểm dữ liệu và K nhóm, các nhóm đã phân chia có thể được mô tả hoàn chỉnh bằng ma trận liên thuộc nhị phân K × n với phần tử $u_{ij} = 1$ nếu điểm dữ liệu thứ j thuộc nhóm i và bằng 0 trong trường hợp khác, tức là

$$u_{ij} = \begin{cases} 1 & \text{nếu} \quad \|x_j - c_i\|^2 \le \|x_j - c_k\|^2 \\ 0 & \text{ngược lại} \end{cases} \tag{2.18}$$

Trong quá trình tạo nhóm cứng, một điểm dữ liệu cho trước có thể chỉ trong một nhóm, do đó, ma trận liên thuộc U có những tính chất sau

$$\sum_{i=1}^{K} u_{ij} = 1 \tag{2.19}$$

với tất cả các điểm dữ liệu j = 1, 2,..., n và

$$\sum_{i=1}^{K}\sum_{j=1}^{n} u_{ij} = n \tag{2.20}$$

Thuật toán tạo nhóm cứng không tính đến các hoạt động tương đối của các nơron, chỉ thưởng cho nơron chiến thắng. Giá trị liên thuộc của nơron thua luôn luôn bằng 0, bất chấp mức độ hoạt động bên trong của nó.

2.1.4.2.3. Thuật toán tạo nhóm cơ bản

Quá trình tạo nhóm mờ, cũng được gọi là nhóm C- trung bình là một thuật toán trong đó mỗi điểm dữ liệu thuộc về các nhóm cụ thể với các hệ số được gọi là độ liên thuộc μ. Giá trị liên thuộc cụ thể của một vài dữ liệu x tới tập F thường được xác định thông qua sử dụng hàm liên thuộc $\mu(x)$. Ở đây, chúng tôi sẽ sử dụng hàm Gaussian tổng quát hóa được xác định như sau:

$$\mu(x) = \exp\left(-\frac{(x-c)^{2b}}{\sigma^2}\right)$$

Tham số c là tâm của hàm, σ mô tả độ rộng của hàm và b điều khiển hình dạng của hàm. Với b = 1, nó là hàm Gaussian gốc, với b >> 1 nó là hình thang và đặc tính tam giác thu được với các giá trị của b nằm giữa 0.4 và 0.8.

Tương tự trường hợp tạo nhóm cứng, quá trình tạo nhóm mờ chia một tập của n vectơ x_i thành K nhóm mờ và tìm tâm nhóm của mỗi nhóm để hàm chi phí đo độ không đồng dạng được tối thiểu hóa. Vectơ **x** thuộc vào nhóm với độ liên thuộc được mô tả bởi phương trình trên.

Giá trị liên thuộc này phụ thuộc mạnh vào khoảng cách từ dữ liệu **x** tới tâm của nhóm. Lúc này, vectơ **x** có thể thuộc vào một vài nhóm với độ liên thuộc có giá trị từ 0 đến 1. Do đó, đầu vào của ma trận U có giá trị từ 0 đến 1 với tất cả các vectơ dữ liệu. Lúc này, hàm chi phí toàn phần có thể được xác định lại như sau :

$$E = \sum_{i=1}^{K}\sum_{j}^{n} u_{ij}^{m} \left\| c_i - x_j \right\|^2$$

với m là số mũ trọng số, $m \in [1, \infty]$. Để đạt được giá trị tối thiểu của hàm chi phí, chúng ta phải xem xét ràng buộc trên. Với vấn đề này, chúng tôi giới thiệu hàm Lagrange LE, được xác định như sau:

$$LE = \sum_{i=1}^{K}\sum_{j}^{n} u_{ij}^{m} \left\| c_i - x_j \right\|^2 + \sum_{j=1}^{n} \lambda_j \left(\sum_{i=1}^{K} u_{ij} - 1 \right) \tag{2.21}$$

trong đó: λ_j là các nhân tử Lagrange cho n ràng buộc của phương trình trên. Các điều kiện cần thiết để tối thiểu hàm Lagrange như sau :

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} x_j}{\sum_{j=1}^{n} u_{ij}^{m}} \tag{2.22}$$

và

$$u_{ij} = \frac{1}{\sum_{k=1}^{K} \left(\frac{d_{ij}}{d_{kj}} \right)^{1/(m-1)}} \tag{2.23}$$

trong đó: d_{ij} là khoảng cách Ơ-clit giữa tâm $\mathbf{c}_i$ và vectơ dữ liệu $\mathbf{x}_j$. Do đó, thuật toán phân nhóm C-trung bình mờ có thể được phát biểu như sau:

– Khởi tạo ma trận U với các giá trị ngẫu nhiên giữa 0 và 1 theo cách mà thỏa mãn các ràng buộc (2.21).

– Tìm K tâm nhóm mờ $\mathbf{c}_i$ sử dụng phương trình (2.22).

– Tính toán hàm chi phí. Nếu E thấp, giá trị sai số giả định hay sự mở rộng qua vòng lặp trước đó của nó bỏ qua được dừng lại, nếu không, đi đến bước kế tiếp.

– Tính toán các đầu vào mới của ma trận U sử dụng phương trình (2.23) và quay lại bước 2.

Quá trình này được lặp lại nhiều lần dẫn tới một cực tiểu nào đó của E, tuy nhiên, nó không nhất thiết là đạt tới giá trị tối thiểu toàn cục. Chất lượng của giải pháp được quyết định bởi sự lựa chọn các tâm nhóm khởi đầu theo sau từ các giá trị ngẫu nhiên của ma trận U. Các tâm nên được tập trung ở những khu vực này nơi phân bố hầu hết các điểm dữ liệu đa chiều. Chỉ trong trường hợp phân bố đều, sự sắp xếp của các tâm dễ dàng và cũng

nên đều nhau. Trong các trường hợp khác nên áp dụng các phương pháp phân bố mật độ dữ liệu đặc biệt. Phương pháp được biết đến nhiều nhất là phương pháp tạo nhóm theo núi và phân nhóm loại trừ.

2.2. MẠNG MLP (MULTILAYER PERCEPTRONS NETWORK)

2.2.1. Cấu trúc mạng

Là một trong những mạng nơron nhiều lớp được xây dựng đầu tiên và hiện nay đã được ứng dụng rộng rãi trong thực tế. Các ứng dụng mạng MLP trong kỹ thuật có thể kể tới:

– Mô hình hóa hàm phi tuyến.

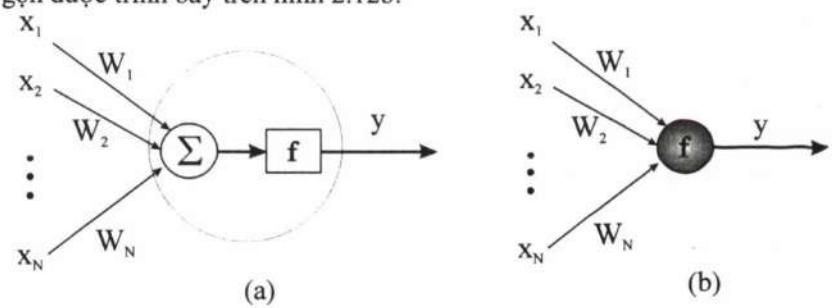
– Nhận dạng đối tượng tuyến tính.

– Nhận dạng đối tượng phi tuyến.

– Phát hiện phần tử sự cố hay bù sai số hệ thống.

...

Mạng MLP được xây dựng từ các phần tử cơ bản gọi là nơron. Mô hình toán học của một nơron được trình bày trên hình 2.12a. Mô hình rút gọn được trình bày trên hình 2.12b:



Hình 2.12. **Mô hình toán học của nơron: mô hình đầy đủ (a) và rút gọn (b)**

Mỗi nơron có thể có nhiều đầu vào x_i, $i = 1 \rightarrow N$. Mỗi đầu vào tín hiệu x_i được khuếch đại bằng trọng số kết nối W_i. Các tín hiệu đầu vào được lấy tổng có trọng lượng và tạo ra tín hiệu đi qua khối hàm f để tạo thành đầu ra tương ứng y. Hàm f được gọi là hàm truyền đạt hoặc hàm kích hoạt (*activation function*). Trong thực tế, ta thường sử dụng các hàm kích hoạt là hàm:

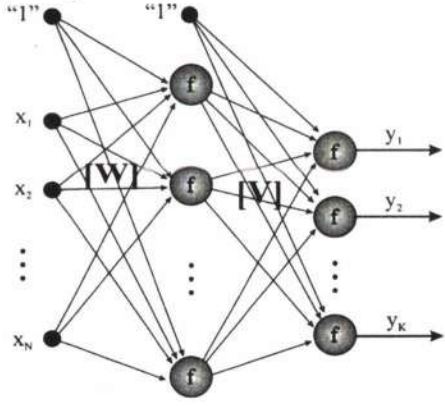
– Logsig: $f(u) = \dfrac{1}{1+e^{-au}}$ (2.24)

– Tansig: $f(u) = \dfrac{1-e^{-au}}{1+e^{-au}}$ (2.25)

– Hàm tuyến tính: f(u) = au (2.26)

Với các phần tử nơron cơ bản này, ta có thể xây dựng mô hình cấu trúc mạng nhiều lớp, bao gồm lớp đầu vào x, lớp đầu ra y và một số lớp ở giữa gọi là lớp ẩn. Tại mỗi lớp (trừ lớp đầu vào) ta có một số nơron, giữa hai lớp bất kỳ của mạng có các kết nối có trọng số. Thông thường ta chỉ cần sử dụng tối đa 2 lớp ẩn là có thể mô hình hóa một hàm phi tuyến với độ chính xác tùy chọn.

Trong các mô hình đã được nghiên cứu, phần lớn chỉ dùng một lớp ẩn. Khi đó mạng MLP sẽ gồm có tổng cộng 3 lớp. Cấu trúc một mạng MLP với 1 lớp ẩn được thể hiện trên hình 2.13 với **W** là ma trận các trọng số kết nối giữa lớp đầu vào và lớp ẩn (W_{ij} – trọng số ghép nối giữa nơron ẩn thứ i và đầu vào thứ j), **V** là ma trận các trọng số kết nối giữa lớp ẩn và lớp đầu ra (V_{ij} – trọng số ghép nối giữa nơron đầu ra thứ *i* và nơron ẩn thứ j).



Hình 2.13. **Cấu trúc mạng MLP với một lớp ẩn**

Mạng MLP với một lớp ẩn có thể được đặc trưng bởi các thông số sau:

– Bộ ba (N, M, K), trong đó: N – số đầu vào; M – số nơron thuộc lớp ẩn; K– số nơron ở lớp đầu ra.

– Các hàm truyền đạt f_1 của lớp ẩn và f_2 của lớp đầu ra.

– Ma trận trọng số $W \in \mathbb{R}^{M\times(N+1)}$ kết nối giữa lớp đầu vào và lớp ẩn, ma trận trọng số kết nối $V \in \mathbb{R}^{K\times(M+1)}$ giữa lớp ẩn và lớp đầu ra.

Khi đó, với vectơ đầu vào x ta có:

– Tổng đầu vào của nơron ẩn thứ i bằng:

$$u_i = \sum_{j=0}^{N} x_j W_{ij} \tag{2.27}$$

Đầu ra của nơron ẩn thứ i: $v_i = f_1(u_i)$ (để phù hợp với công thức, ta coi đầu vào phân cực của các nơron lớp ra là $v_0 = 1$.

– Tổng đầu vào của nơron đầu ra thứ i:

$$g_i = \sum_{j=0}^{M} v_j V_{ij} . \tag{2.28}$$

– Và cuối cùng ta có đầu ra thứ i của mạng sẽ bằng $y_i = f_2(g_i)$. Quá trình tính toán đáp ứng của mạng MLP khi có đầu vào x được gọi là quá trình truyền thẳng.

Nhiệm vụ của quá trình học là xác định giá trị các phần tử của W và V sao cho đáp ứng đầu ra của mạng gần giống với giá trị đích nhất. Ta có thể sử dụng thuật toán bước giảm cực đại để giải bài toán này. Hiện nay mạng MLP đã có nhiều công trình nghiên cứu kỹ cả phần lý thuyết và ứng dụng trong các lĩnh vực nhận dạng, ước lượng, dự báo, các hệ thống tự động điều chỉnh quá trình đạt hiệu quả tốt.

2.2.2. Thuật toán học của mạng MLP

Nội dung của thuật toán bước giảm cực đại có thể được trình bày như sau: Ta có một hàm f(x) bất kỳ và một giá trị khởi đầu $x^{(0)}$. Ta cần tìm điểm $x^{(1)}$ sao cho $f(x^{(1)}) < f(x^{(0)})$. Nếu tiếp tục quá trình tìm kiếm như vậy, ta sẽ thu được kết quả là một chuỗi $\{x^{(k)}\}$ mà giá trị hàm số $f(x^{(k)})$ giảm dần, hay nói cách khác $f(x^{(k)}) \to \min$. Theo thuật toán bước giảm cực đại ta sẽ có công thức xác định bước tiếp theo:

$$x^{(k+1)} = x^{(k)} - \eta \left.\frac{\partial f}{\partial x}\right|_{x=x^{(k)}} \tag{2.29}$$

trong đó: η – hệ số bước học. Công thức này là hệ quả của phép khai triển Taylor của hàm số f lân cận điểm $x^{(k)}$:

$$f(x^{(k)}+\Delta) \approx f(x^{(k)}) + \Delta.f'(x^{(k)}) + O(\Delta^2)$$

từ đó suy ra nếu chọn $\Delta = -\eta.f'(x^{(k)})$ với η đủ nhỏ thì

$$f(x^{(k)}+\Delta) \approx f(x^{(k)}) - \eta.\left(f'(x^{(k)})\right)^2 + O(\Delta^2) < f(x^{(k)}) \qquad (2.30)$$

2.2.3. Thuật toán bước giảm cực đại và ứng dụng trong mạng MLP

Sử dụng thuật toán bước giảm cực đại ta có các công thức thay đổi các trọng số trong hai ma trận W và V để xác định điểm cực trị của hàm mục tiêu E như sau:

$$\begin{cases} W_{\alpha\beta}^{(k+1)} = W_{\alpha\beta}^{(k)} - \eta \dfrac{\partial E}{\partial W_{\alpha\beta}^{(k)}} \\ V_{\alpha\beta}^{(k+1)} = V_{\alpha\beta}^{(k)} - \eta \dfrac{\partial E}{\partial V_{\alpha\beta}^{(k)}} \end{cases} \qquad (2.31)$$

với hàm mục tiêu

$$E = \frac{1}{2}\sum_{i=1}^{p}\left\|y^{(i)} - d^{(i)}\right\|^2$$

Từ các công thức đã nêu trên ta có

$$\frac{\partial E}{\partial V_{\alpha\beta}^{(k)}} = \sum_{i=1}^{p}\sum_{j=1}^{K}\left(y_j^{(i)} - d_j^{(i)}\right)\frac{\partial y_j^{(i)}}{\partial V_{\alpha\beta}} \qquad (2.32)$$

trong đó:

$$\frac{\partial y_j^{(i)}}{\partial V_{\alpha\beta}} = f_2'\left(g_j^{(i)}\right)\frac{\partial g_j^{(i)}}{\partial V_{\alpha\beta}} = f_2'\left(g_j^{(i)}\right)\sum_{k=0}^{N} v_k^{(i)}\frac{\partial V_{jk}}{\partial V_{\alpha\beta}} \qquad (2.33)$$

nhưng $\dfrac{\partial V_{jk}}{\partial V_{\alpha\beta}} = 1$ khi $j = \alpha$ và $k = \beta$, $\dfrac{\partial V_{jk}}{\partial V_{\alpha\beta}} = 0$ trong các trường hợp còn lại.

Từ đó ta có:

$$\frac{\partial E}{\partial V_{\alpha\beta}} = \sum_{i=1}^{p}\left(y_\alpha^{(i)} - d_\alpha^{(i)}\right)f_2'\left(g_\alpha^{(i)}\right)v_\beta^{(i)} \qquad (2.34)$$

Một cách hoàn toàn tương tự ta có:

$$\frac{\partial E}{\partial W_{\alpha\beta}} = \sum_{i=1}^{p}\sum_{j=1}^{K}\left(y_j^{(i)} - d_j^{(i)}\right)f_2'\left(g_j^{(i)}\right)V_{j\alpha}f_1'\left(u_\alpha^{(i)}\right)x_\beta^{(i)} \qquad (2.35)$$

Với các công thức gradient này, ta có thể tiến hành tối ưu hoá các giá trị của hai ma trận **W** và **V**.

Tuy nhiên trong thực tế áp dụng, thuật toán trên chỉ sử dụng thông tin đạo hàm bậc nhất trong khai triển Taylor nên độ chính xác chưa cao, tốc độ hội tụ chậm và xác suất vượt qua được các cực trị địa phương nhỏ. Khi đó ta có thể sử dụng thuật toán Levenberg – Marquadt cũng để tối ưu hóa hàm sai số. Biểu diễn hàm sai số theo các kênh đầu ra của mạng ta có:

$$E = \frac{1}{2}\sum_{i=1}^{K}\left(y_i(x) - d_i\right) = \sum_{i=1}^{K} e_i\left[(x)\right]^2 \tag{2.36}$$

với $e_i(\mathbf{x}) = (y_i(\mathbf{x}) - d_i)$ là sai số trên kênh đầu ra thứ i khi có đầu vào **x**. Ký hiệu:

$$\mathbf{e}(\mathbf{x}) = \begin{bmatrix} e_1(\mathbf{x}) \\ e_2(\mathbf{x}) \\ \dots \\ e_K(\mathbf{x}) \end{bmatrix} \qquad \mathbf{J} = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \dots & \frac{\partial e_1}{\partial w_n} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \dots & \frac{\partial e_2}{\partial w_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_K}{\partial w_1} & \frac{\partial e_K}{\partial w_2} & \dots & \frac{\partial e_K}{\partial w_n} \end{bmatrix} \tag{2.37}$$

khi đó vectơ gradient **g** và ma trận xấp xỉ ma trận Hess là **G** ta có:

$$\mathbf{g} = \mathbf{J}^T\mathbf{e}(\mathbf{x}) \tag{2.38}$$

$$\mathbf{G} = \mathbf{J}^T\mathbf{J} + v\mathbf{1} \tag{2.39}$$

Trong công thức (2.39) thì hệ số v được gọi là tham số điều chỉnh Levenberg – Marquardt. Đây là đại lượng vô hướng và cần giảm về 0 trong quá trình tối ưu hóa. Khi quá trình học bắt đầu các giá trị trọng số **w** vẫn còn khác xa so với điểm tối ưu (hay nói cách khác giá trị hàm sai số (2.36) còn lớn) ta sẽ sử dụng biến v với các giá trị lớn (so với các giá trị riêng của ma trận $\mathbf{J}^T\mathbf{J}$) để ma trận **G** coi như xấp xỉ v**1**, khi đó quá trình điều chỉnh thích nghi sẽ được thực hiện theo thuật toán tương tự như thuật toán bước giảm cực đại:

$$\mathbf{p}_k = -\mathbf{g}_k / v \tag{2.40}$$

Trong quá trình học, khi sai số đã giảm xuống giá trị tương đối nhỏ, tham số v cũng sẽ giảm đi và thành phần $\mathbf{J}^T\mathbf{J}$ sẽ chiếm ưu thế và ma trận Hess bậc 2 sẽ đóng vai trò cao hơn trong quá trình điều chỉnh. Và hướng gradient điều chỉnh khi đó sẽ có thể được xác định theo công thức:

$$\mathbf{p}_k = -\left[\mathbf{J}^T\mathbf{J}\right]^{-1}\mathbf{g}_k \tag{2.41}$$

Lúc này, thuật toán Levenberg – Marquardt trở thành thuật toán kinh điển Gauss – Newton hội tụ bậc 2 trong bài toán tối ưu hóa. Đây là một trong những thuật toán thông dụng nhất để áp dụng trong quá trình học mạng MLP.

Chương 3
MẠNG NƠRON LÔGIC MỜ TSK
(Takaga – Sugeno - Kang)

3.1. LÔGIC MỜ

3.1.1. Khái niệm lôgic mờ

Khái niệm "lôgic mờ" dùng để chỉ việc xử lý các thông tin mà giá trị lôgic không thể xác định rõ, hoặc biến thiên theo điều kiện bên ngoài. Chẳng hạn, với các mệnh đề "Nhiệt độ < 20^0C là lạnh" hay "Tốc độ ôtô khoảng 60km/h là nhanh" thì rất khó xác định được giá trị lôgic vì không có 100% số người cho rằng các mệnh đề này là chính xác. Trong trường hợp này, thường hay gặp dạng phát biểu như sau:

Có 70% số người đồng ý "t < 20^0C là lạnh" hay có 60% số người đồng ý "Tốc độ ô tô khoảng 60 km/h là nhanh".

Trong lôgic kinh điển, khi đưa ra một định nghĩa về tập hợp, ví dụ như: B là một tập hợp gồm các số thực lớn hơn 6

$$B = \{x \in R | x > 6\}$$

thì đây được coi là một định nghĩa chặt chẽ. Theo cách định nghĩa như vậy với một giá trị x bất kì, có thể biết được x có thuộc tập B hay không. Nói cách khác, với mọi số x, thì mệnh đề $x \in B$ chỉ có thể nhận một trong hai giá trị: Đúng (bằng 1), hay sai (bằng 0).

Tuy nhiên, nếu đưa ra một định nghĩa: "C là một tập hợp gồm các số thực có giá trị xấp xỉ (bằng hoặc gần bằng) 3" hay

$$C = \{x \in R | x \approx 3\}$$

thì rõ ràng đây là một định nghĩa không chặt chẽ (hay còn gọi là mờ) vì thực tế không tồn tại một định nghĩa rõ cho khái niệm "xấp xỉ". Với một định nghĩa "mờ" như vậy, không thể khẳng định giá trị x = 2 hay x = 2,9 có thuộc tập C hay không. Nếu đã không khẳng định được x = 2 hay x = 2,9 có thuộc

C (xấp xỉ 3) hay không thì cũng không khẳng định là x = 2 hay x = 2,9 không thuộc C (không xấp xỉ 3).

Khác với lôgic kinh điển chỉ có hai giá trị là 1 nếu $x \in C$ hoặc bằng 0 nếu $x \notin C$, lôgic mờ sẽ đưa ra quan niệm mới có vai trò làm rõ định nghĩa cho tập mờ. Nói một cách khác, với lôgic mờ thì một giá trị x nào đó sẽ có thể thuộc về tập C khoảng bao nhiêu phần trăm? điều đó sẽ được thể hiện thông qua giá trị hàm liên thuộc $\mu(x)$ tại điểm x đó sẽ bằng bao nhiêu. Chẳng hạn có thể nói như sau: "giá trị x = 2,9 thuộc về tập C chín sáu phần trăm" hay "giá trị x = 2 thuộc về tập *C* bốn sáu phần trăm" và giá trị bao nhiêu phần trăm đó sẽ tuỳ thuộc vào cách xây dựng mô hình hàm liên thuộc như thế nào? Như vậy với ví dụ trên, cần có độ tin cậy của mệnh đề $x = 2,9 \in C$ phải cao hơn độ tin cậy của mệnh đề $x = 2 \in C$.

3.1.2. Biểu thức giá trị mờ

Để tìm hiểu về biểu thức giá trị mờ, ta xét 3 dạng biểu thức mờ cơ bản sau:

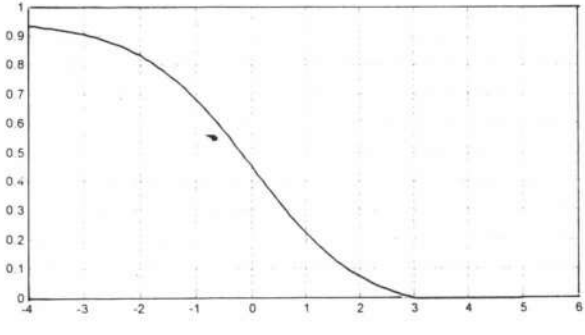
– x nhỏ hơn nhiều so với A : $x \ll A$

– x xấp xỉ bằng A : $x \approx A$

– x lớn hơn nhiều so với A : $x \gg A$

Đối với biểu thức mờ $x \ll A$, hàm liên thuộc (hay còn gọi là hàm phụ thuộc) $\mu_{\ll A}(x)$ có dạng như hình 3.1 với $0 \le \mu_{\ll A}(x) \le 1$, được định nghĩa:

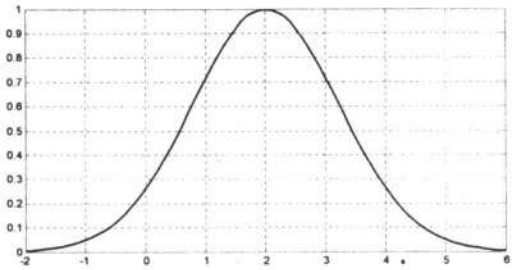
$$y = \mu_{<<A}(x) = \begin{cases} \to 1 & \text{khi} \quad x - A \to -\infty \\ \to 0 & \text{khi} \quad x - A \to +\infty \end{cases} \qquad (3.1)$$



Hình 3.1. **Hàm liên thuộc của biểu thức mờ x << A**

Đối với biểu thức mờ $x \approx A$, hàm liên thuộc $\mu_{\approx A}(x)$ sẽ có dạng hình chuông như hình 3.2 với $0 \le \mu_{\approx A}(x) \le 1$, được định nghĩa:

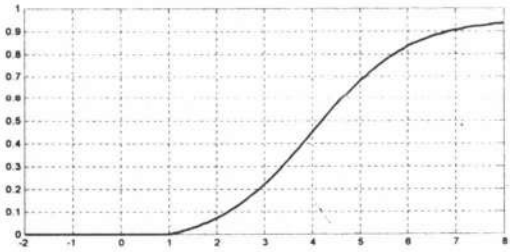
$$y = \mu_{\approx A}(x) = \begin{cases} 1 & \text{khi} \quad x = A \\ \to 0 & \text{khi} \quad |x - A| \to \infty \end{cases} \tag{3.2}$$



Hình 3.2. **Hàm liên thuộc hình chuông của biểu thức mờ x ≈ A**

Đối với biểu thức mờ $x \gg A$, hàm liên thuộc $\mu_{\gg A}(x)$ sẽ có dạng như hình 3.3 với $0 \le \mu_{\gg A}(x) \le 1$, được định nghĩa:

$$y = \mu_{\gg A}(x) = \begin{cases} \to 1 & \text{khi} \quad x - A \to +\infty \\ \to 0 & \text{khi} \quad x - A \to -\infty \end{cases} \tag{3.3}$$



Hình 3.3. **Hàm liên thuộc của biểu thức mờ x>>A**

Các hàm liên thuộc đều có thể được mô tả dưới dạng rời rạc gồm một tập các giá trị liên thuộc. Đối với các hàm liên thuộc hình chuông, thường sử dụng hàm Gauss và được mô tả bằng phương trình:

$$\mu_{c,b,\sigma}(x) = \frac{1}{1 + \left(\frac{\|x - c\|}{\sigma}\right)^{2b}} \tag{3.4}$$

Trong đó: $b, \sigma \in \mathbb{R}$ và x, c là các vectơ. Ba thông số c, b và σ có thể điều chỉnh linh hoạt, nên hình dạng của hàm liên thuộc sẽ được thay đổi bởi ba tham số là: Trọng tâm c, độ mở σ và hệ số mũ b.

Để làm rõ về ảnh hưởng của các tham số đến hình dạng của hàm liên thuộc, xét ví dụ về tập mờ c gồm các số thực gần bằng 3

$$C = \{x \in R | x \approx 3\}$$

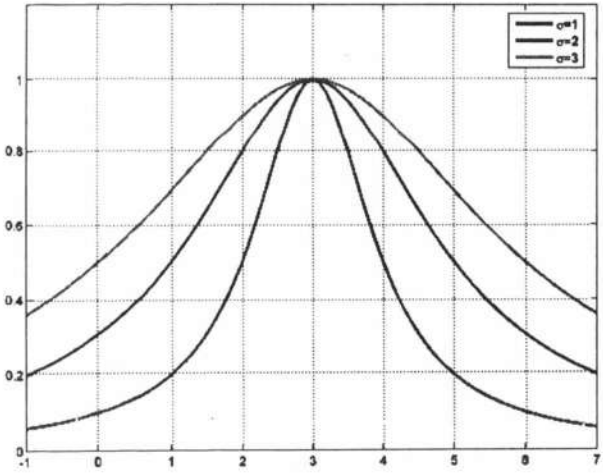
Hàm phụ thuộc $\mu_{\approx 3}(x)$ tại điểm x nào đó phải có giá trị trong khoảng [0,1], tức là:

$$0 \le \mu_{\approx 3}(x) \le 1$$

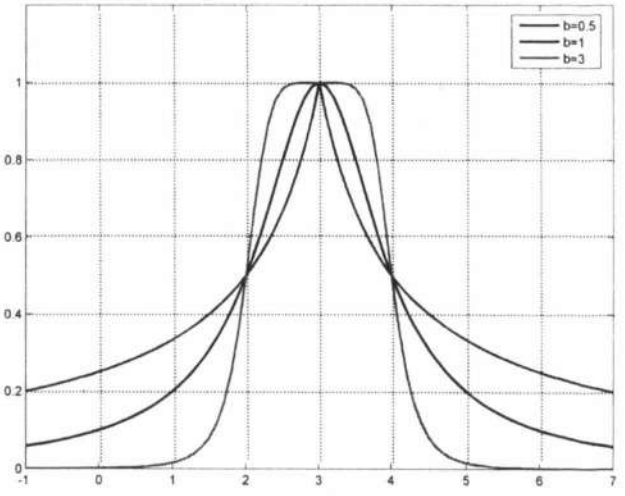
trong đó: $\mu_{\approx 3}(x) \to 1$ khi $|x-3| \to 0$ và $\mu_{\approx 3}(x) \to 0$ khi $|x-3| \to \infty$.

Giả sử chọn giá trị c = 3, độ mở σ = 1, b = 1 thì hàm liên thuộc của tập C như sau:

$$\mu_{\approx 3}(x) = \frac{1}{1+\left(\frac{|x-3|}{1}\right)^2} = \frac{1}{1+(|x-3|)^2}$$



Hình 3.4a. **Hình dạng của hàm liên thuộc $\mu_{\approx 3}(x)$ với ba độ mở khác nhau**



Hình 3.4b. **Hình dạng hàm liên thuộc $\mu_{\approx 3}(x)$ với giá trị khác nhau hệ số b**

Hình 3.4a mô tả hình dạng của hàm liên thuộc $\mu_{\approx 3}(x)$ với hệ số mũ được chọn cố định b = 1 và giá trị độ mở σ lần lượt thay đổi bằng 1, 2 và 3. Trên hình 3.4b, nếu chọn giá trị độ mở σ cố định bằng 1 và giá trị b thay đổi lần lượt bằng 0,5 thì hàm kích hoạt có dạng gần như hình tam giác, b = 1 có dạng hình chuông và b = 3 có dạng gần với hình thang.

Một điểm chung trong các hàm $\mu_{\approx 3}(x)$ là các điểm có giá trị càng gần trọng tâm thì sẽ có giá trị liên thuộc càng lớn (tiến tới 1), đối với các điểm có giá trị càng xa trọng tâm thì có giá trị liên thuộc càng nhỏ (tiến tới 0).

3.1.3. Quy tắc suy luận mờ và giá trị của quy tắc suy luận mờ

Trong lý thuyết điều khiển, các quy tắc điều khiển thường được mô tả dưới dạng mệnh đề điều kiện *IF ...THEN*

IF đầu vào là A THEN đầu ra là B (3.5)

Đối với khái niệm điều khiển chính xác, cần chỉ rõ rằng nếu đại lượng đầu vào có giá trị cụ thể bao nhiêu thì đầu ra cũng bằng một giá trị cụ thể nào đó. Như vậy một quy tắc suy luận chính xác sẽ có cấu trúc như sau:

IF x = A THEN y = B (3.6)

tức là khi giá trị x bằng A thì giá trị y sẽ bằng B.

Tuy nhiên một vấn đề đặt ra: Nếu khi giá trị x "xấp xỉ" bằng A thì giá trị y sẽ bằng bao nhiêu? Và với các giá trị x lân cận quanh giá trị A với độ "xấp xỉ" khác nhau thì có thể tính được giá trị của y hay không? Khái niệm "lôgic mờ" sẽ trả lời được câu hỏi trên thông qua quy tắc suy luận mờ.

Một quy tắc suy luận mờ sẽ có cấu trúc như sau:

$$\text{IF} \quad x \approx A \quad \text{THEN } y \approx B \tag{3.7}$$

tức là nếu x xấp xỉ bằng A thì y sẽ xấp xỉ bằng B. Khi x bằng A thì sẽ có y bằng B.

Ở đây khái niệm "xấp xỉ" sẽ được biểu diễn thông qua hàm liên thuộc $\mu_{\approx A}(x)$ và thông qua giá trị của hàm liên thuộc. Với một giá trị đầu vào x bất kì, có thể đề xuất phương pháp tính được giá trị đầu ra y của quy tắc suy luận mờ như sau:

$$y = B.\mu_{\approx A}(x) \tag{3.8}$$

Trên thực tế, một hệ thống điều khiển hay nhận dạng không chỉ có duy nhất một quy tắc suy luận mờ mà thông thường sẽ bao gồm một tập hợp các quy tắc suy luận mờ. Như vậy đáp ứng đầu vào của hệ thống trong cùng lúc sẽ chịu tác động của nhiều quy tắc suy luận mờ.

Ví dụ: hệ thống gồm có 4 quy tắc mờ R1, R2, R3, R4 như hình 3.5

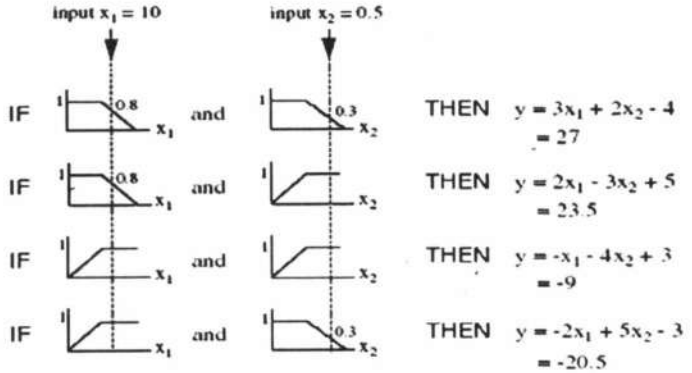
R1: If x_1 is small and x_2 is small Then $y = 3x_1 + 2x_2 - 4$

R2: If x_1 is small and x_2 is big Then $y = 2x_1 - 3x_2 + 5$

R3: If x_1 is big and x_2 is big Then $y = -x_1 - 4x_2 + 3$

R4: If x_1 is big and x_2 is small Then $y = -2x_1 + 5x_2 - 3$

Giả sử vectơ đầu vào là: $x = (x_1; x_2)^T = (10; 0,5)^T$, khi đó các giá trị liên thuộc tương ứng với các quy tắc sẽ được lần lượt tính toán bằng: 0,24; 0,8; 1,0; 0,3, tương ứng với các giá trị đầu ra là: $y_{R1} = 27$; $y_{R2} = 23,5$; $y_{R3} = -9$; $y_{R4} = -20,5$.



Hình 3.5. **Đồ thị hệ thống gồm có 4 quy tắc mờ**

Trong trường hợp này độ mạnh của luật được chọn chính bằng giá trị liên thuộc của x trong từng quy tắc. Giá trị $\mu_i(x)$ và giá trị đầu ra của toàn hệ thống đối với một giá trị x nào đó sẽ được tính dựa trên ảnh hưởng (độ mạnh) của các quy tắc đối với giá trị x đó và được tính như sau:

$$y = \frac{\sum_{i=1}^{N} \mu_i(x).y_i(x)}{\sum_{i=1}^{N} \mu_i(x)} \tag{3.9}$$

$$y = \frac{0,24.27 + 0,8.23,5 + 1,0.(-9) + 0,3.(-20,5)}{0,24 + 0,8 + 0,1 + 0,3} = 4,33$$

trong nhiều trường hợp giá trị $\sum_{i=1}^{N} \mu_i(x)$ được quy chuẩn bằng 1 nên ta có

$$y = \sum_{i=1}^{N} \mu_i(x)\, y_i(x) \tag{3.10}$$

3.2. MẠNG TSK (TAKAGA – SUGENO - KANG)

3.2.1. Mô hình mạng TSK

3.2.1.1. Các luật suy luận TSK

Mạng TSK là mạng nơron lôgic mờ, dùng để mô phỏng các luật suy luận và điều khiển mờ do ba tác giả người Nhật là Takaga, Sugeno và Kang đề xuất. Một quy tắc suy luận mờ TSK có dạng như sau:

$$\text{if } x \approx C \text{ then } y \approx f(x) = a_0 + a_1x_1 + ... + a_Nx_N \qquad (3.11)$$

trong đó $\quad x = [x_1, x_2, ..., x_N]^T, C = [C_1, C_2, ..., C_N]^T \in R^N$

Quy tắc trong phương trình (3.11) có một vectơ đặc trưng C được gọi là trọng tâm của quy tắc. Nếu vectơ đầu vào x càng gần với trọng tâm này thì đầu ra của quy tắc sẽ càng gần với giá trị f(x) với f là một hàm tuyến tính cho trước của vectơ đầu vào (trường hợp đặc biệt, khi x = C thì y = f(x)). Hiểu theo một cách khác, luật suy luận này dùng để tạo đáp ứng đầu ra khi số liệu đầu vào thuộc về lân cận của một điểm $C = [C_1, C_2, ..., C_N]^T \in R^N$ nào đó. Để có thể tạo ra một mô hình suy luận bao trùm được không gian số liệu đầu vào, có thể sử dụng hệ nhiều quy tắc suy luận:

$$\begin{cases} \text{if } x \approx C_1 \text{ then } y_1 \approx f_1(x) \\ \dots\dots\dots\dots\dots\dots \\ \text{if } x \approx C_M \text{ then } y_M \approx f_M(x) \end{cases} \qquad (3.12)$$

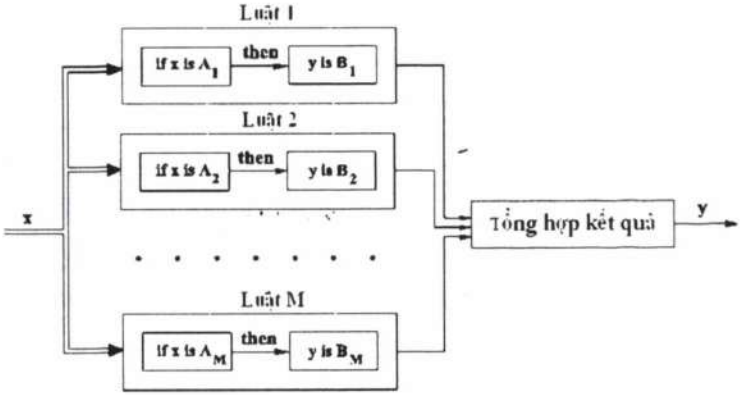
Có thể thay các giá trị phụ thuộc mờ ở hệ trên bằng các hàm chính xác:

$$\begin{cases} y_1 = W_{\approx C_1}(x) f_1(x) \\ \dots\dots\dots\dots\dots\dots \\ y_M = W_{\approx C_M}(x) f_M(x) \end{cases} \qquad (3.13)$$

Trong đó $W_{\approx C_M}$ là hệ số kích hoạt của luật mờ.

Khi đó, ứng với mỗi đầu vào x, mỗi quy tắc suy luận sẽ đưa ra một đáp ứng y_i. Để có thể tổng hợp lại và đưa ra được một đáp ứng duy nhất, các tác giả đã đề xuất lấy trung bình trọng số của các đáp ứng riêng lẻ. Từ đó có:

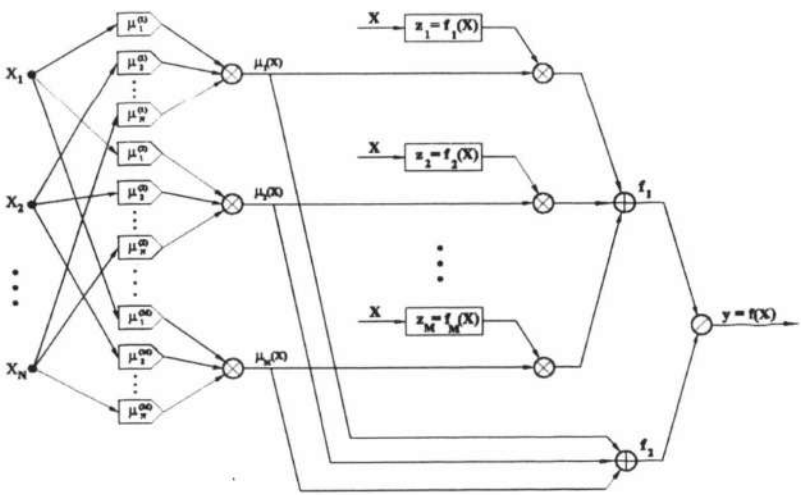
$$y = \frac{\sum_{i=1}^{M} W_{\approx C_i}(x) f_i(x)}{\sum_{i=1}^{M} W_{\approx C_i}(x)} \qquad (3.14)$$



Hình 3.6. **Mô hình hệ nhiều luật**

3.2.1.2. Cấu trúc chung mạng TSK

Phát triển từ hệ suy luận các tác giả Takaga, Sugeno và Kang đã đề xuất mô hình mạng TSK để mô phỏng hệ suy luận. Mạng này thuộc hệ thống các hệ suy luận mờ, ngày nay được áp dụng rộng rãi trong kỹ thuật. Để mô phỏng hoạt động của hệ thống ta có cấu trúc mạng được trình bày cụ thể như hình 3.7, cấu trúc gồm 5 lớp:



Hình 3.7. **Cấu trúc mạng TSK kinh điển**

Lớp thứ nhất bao gồm các khối $\mu_{ij}(x_j)$ còn được gọi là khối mờ hoá cho thành phần thứ j của vectơ đầu vào x. Mỗi khối mờ hoá thường sử dụng hàm Gauss mở rộng như biểu thức (3.15)

$$\mu_{ij}(x_j) = \frac{1}{1 + \left(\frac{\|x_j - c_{ij}\|}{\sigma_{ij}}\right)^{2b_{ij}}} \tag{3.15}$$

được đặc trưng bởi 3 tham số $c_{ij}, b_{ij}, \sigma_{ij}$, trong đó: C là số trọng tâm, b là hệ số mũ, σ là độ mở, i là số luật, j số kênh đầu vào. Hàm Gauss mở rộng này có tính chất là $\mu_{ij}(x_j) \to 1$ khi $\|x_j - c_{ij}\| \to 0$ và $\mu_{ij}(x_j) \to 0$ khi $\|x_j - c_{ij}\| \to \infty$.

Lớp thứ hai là khối nhân, dùng để tính tích đầu ra của các khối mờ hoá

$$\mu_i(x) = \prod_{j=1}^{N} \mu_{ij}(x_j) \tag{3.16}$$

Đầu ra của lớp thứ hai sẽ là các hệ số kích hoạt $\mu_i(x)$ của các kênh suy luận f_i phía sau.

Lớp thứ ba là khối tính các giá trị hàm f_i đầu ra của mạng TSK, đây là các hàm tuyến tính.

Lớp thứ tư là khối tính toán các thành phần tử số f_1 và mẫu số f_2.

$$f_1 = \sum_{i=1}^{M} w_i f^{(i)}(x) \tag{3.17}$$

$$f_2 = \sum_{i=1}^{M} w_i \tag{3.18}$$

Lớp thứ năm là khối tính đáp ứng cuối cùng của mạng TSK

$$y = \frac{\sum_{i=1}^{M} W_i f^{(i)}(x)}{\sum_{i=1}^{M} W_i} = \frac{f_1}{f_2} \tag{3.19}$$

Trong cấu trúc này ta có lớp 2, 4 và 5 là các lớp tính toán và hoạt động cố định. Các lớp 1 và 3 là các lớp có các tham số có thể thay đổi thích nghi để xây dựng mô hình tối ưu. Lớp 1 đó là các tham số c_i, b_i, σ_i của các khối mờ hoá, trong lớp 3 đó là các hệ số a_{ij} của hàm tuyến tính.

3.2.1.3. Cải tiến cấu trúc kinh điển và thuật toán xây dựng mạng TSK

Trong mẫu truyền thống, độ mạnh của quy tắc mờ thứ i phụ thuộc khoảng cách giữa vectơ đầu vào và mẫu của quy tắc và được tính toán bằng:

$$\mu_i(x) = \prod_{j=1}^{N} \mu_{ij}(x_j) = \prod_{j=1}^{N} \frac{1}{1 + \left(\frac{\left(x_j - c_{ij}\right)}{\sigma_{ij}} \right)^{2b_{ij}}} \tag{3.20}$$

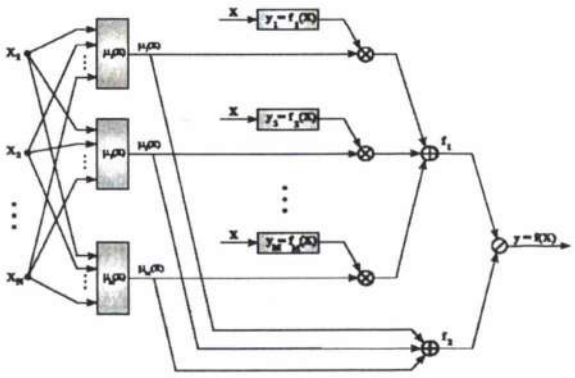
Điều này đòi hỏi hệ số mũ và độ mở được xác định cho mọi tín hiệu đầu vào. Những hệ số này là cần thiết để xác định dữ liệu theo từng chiều, nhưng tại cùng một thời điểm chúng làm phức tạp cấu trúc của mạng và làm tăng số lượng các tham số phi tuyến. Để giảm số lượng các tham số phi tuyến ta sử dụng công thức xác định khoảng cách. Phương pháp được thể hiện dưới dạng tổng quát như sau:

$$d^2(x,c) = (x-c)^T.S.(x-c) \tag{3.21}$$

Trong đó S là ma trận xác định dương, đối xứng. Có thể dễ dàng nhận ra rằng (3.21) là công thức tổng quát hơn công thức xác định khoảng cách Eucildes kinh điển, trong đó S = I – ma trận đơn vị. Với phương pháp này, mức độ kích hoạt của quy tắc thứ i có thể chỉ chứa một cặp hệ số mũ và độ rộng cho mọi biến N. Hàm mờ hiệu chỉnh được xác định là:

$$\mu_i(x) = \frac{1}{1 + \left(\frac{\left\| x - c_i \right\|}{\sigma_i} \right)^{2b_i}} \tag{3.22}$$

Mẫu hiệu chỉnh mạng TSK chỉ có $M \times (N+2)$ tham số điều chỉnh phi tuyến.



Hình 3.8. **Cấu trúc mạng TSK cải tiến**

Để điều chỉnh các tham số của mạng ta có thể chia các tham số thành 2 nhóm: Tham số a_{ij} của các hàm trong mạng TSK gọi là các tham số tuyến tính, và các tham số c_i, b_i, σ_i gọi là các tham số phi tuyến. Thuật toán điều chỉnh được thực hiện như sau:

1. Hiệu chỉnh các tham số tuyến tính a_{ij} các hàm của mạng TSK tại các giá trị cố định các tham số phi tuyến.

2. Hiệu chỉnh các tham số phi tuyến tại các giá trị cố định các tham số tuyến tính.

Các bước 1 và 2 được lặp lại nhiều lần cho đến khi thiết lập được các tham số của mạng. Tại N biến đầu vào và M quy tắc độc lập có $M \times (N+1)$ tham số tuyến tính có thể điều chỉnh được của hàm TSK cho mỗi đầu ra.

Trong bước 1, các tham số phi tuyến là cố định và với sự kích thích của hệ thống bởi vectơ **x**, tín hiệu đầu ra có thể viết dưới dạng:

$$y(x) = \sum_{i=1}^{M} \mu_i(x) \left[a_{i0} + \sum_{j=1}^{N} a_{ij} x_j \right] \tag{3.23a}$$

Khi có p cặp mẫu $\{x_i, d_i\}$ $i = 1, \ldots, p$, ta ký hiệu $W_{ij} = \mu_i(x_j)$ tính cho mẫu đầu vào $x_{(j)}$. Các cặp mẫu này cho phép xây dựng được hệ p phương trình tuyến tính, từ đó có thể xác định được các thông số tuyến tính tối ưu (ứng với sai số nhỏ nhất). Hệ p phương trình tuyến tính được khái quát như sau:

$$\begin{bmatrix} W_{11} & W_{11}x_{11} & \cdots & W_{11}x_{1N} & \cdots & W_{M1} & W_{M1}x_{11} & \cdots & W_{M1}x_{1N} \\ W_{12} & W_{12}x_{21} & \cdots & W_{12}x_{2N} & \cdots & W_{M2} & W_{M2}x_{21} & \cdots & W_{M2}x_{2N} \\ & & \cdots & & \cdots & & & \cdots & \\ W_{1p} & W_{1p}x_{p1} & \cdots & W_{1p}x_{pN} & \cdots & W_{Mp} & W_{Mp}x_{p1} & \cdots & W_{Mp}x_{pN} \end{bmatrix} \times \begin{bmatrix} a_{10} \\ \vdots \\ a_{1N} \\ \vdots \\ a_{M0} \\ \vdots \\ a_{MN} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_p \end{bmatrix} \tag{3.23b}$$

Hoặc dưới dạng ma trận

$$W \times A = D \tag{3.24}$$

Ma trận W có kích thước là $p \times (N+1)M$. Hệ phương trình này được xác lập bằng cách sử dụng ma trận nghịch đảo giả của W.

$$A = W^{+} \times D \tag{3.25}$$

với W^{+} là ma trận nghịch đảo giả của W, được tính bằng cách sử dụng thuật toán SVD.

Sau khi cố định giá trị của các tham số phi tuyến, bước thứ 2 bắt đầu với việc tính giá trị đầu ra cần thiết bằng cách sử dụng mối quan hệ tuyến tính.

$$Y = W \times A \tag{3.26}$$

Có các giá trị đầu ra có thể xác định giá trị của các hàm mục tiêu.

$$E = \frac{1}{2}\sum_{l=1}^{p}\left(y(x_l) - d_l\right)^2 \tag{3.27}$$

Các tham số phi tuyến được hiệu chỉnh bằng cách sử dụng phương pháp bước giảm cực đại:

$$c_{kj}(t+1) = c_{kj}(t) - \eta_c \frac{\partial E(t)}{\partial c_{kj}} \tag{3.28}$$

$$\sigma_{kj}(t+1) = \sigma_{kj}(t) - \eta_\sigma \frac{\partial E(t)}{\partial \sigma_{kj}} \tag{3.29}$$

$$b_{kj}(t+1) = b_{kj}(t) - \eta_b \frac{\partial E(t)}{\partial b_{kj}} \tag{3.30}$$

Các phương trình từ (3.28) đến (3.30) sử dụng độ dốc của hàm mục tiêu E với các tham số phi tuyến, độ dốc đối với mẫu $c_{\alpha\beta}$ của biến thứ β và quy tắc thứ α có thể biểu diễn trong mối quan hệ sau:

$$\frac{\partial E}{\partial c_{\alpha\beta}} = \sum_{l=1}^{p} \frac{y(x_l) - d_l}{\left(\sum_{i=1}^{M} W_{li}\right)^2} \frac{\partial W_{\alpha l}}{\partial c_{\alpha\beta}} \left[\sum_{m=1}^{M} W_{ml}\left(g_\alpha(x_l) - g_m(x_l)\right)\right] \tag{3.31}$$

với $g_\alpha(x_l) = a_{\alpha 0} + \sum_{j=1}^{N} a_{\alpha j} x_{lj}$ ta có

$$\frac{\partial W_{\alpha l}}{\partial c_{\alpha\beta}} = \frac{4b_\alpha \times W_{\alpha l} \times (1 - W_{\alpha l}) \times \sum_{j=1}^{N} S_{\beta j} \times (x_{lj} - c_{\alpha j})}{d^2(x, c_\alpha)} \tag{3.32}$$

Hàm dốc liên quan tới $c_\beta^{(\alpha)}$ là

$$\frac{\partial W_\alpha}{\partial c_{\alpha\beta}} = \sum_{l=1}^{p}\left\{\frac{4e_l b_\alpha W_\alpha(1-W_\alpha).\sum_{j=1}^{N} S_{\beta j}(x_{lj}-c_{\alpha j})}{\left(\sum_{j=1}^{M} w_j\right)^2 . d^2\left(x, c^{(r)}\right)}\left[\sum_{k=1}^{M} w_{kl}\left(g_{\alpha l}-g_{kl}\right)\right]\right\}$$

(3.33)

Tương tự các hàm dốc quan hệ tới σ i b như sau:

$$\frac{\partial E}{\partial \sigma_\alpha} = \sum_{l=1}^{p}\left\{\frac{e^{(l)}}{\left(\sum_{j=1}^{M} w_{lj}\right)^2}\frac{2b^{(\alpha)}}{\sigma_\alpha}. W_\alpha(1-W_\alpha)\left[\sum_{k=1}^{M} W_{kl}\left(g_{\alpha l}-g_{kl}\right)\right]\right\} \quad (3.34)$$

$$\frac{\partial E}{\partial b_\alpha} = -\sum_{l=1}^{p}\left\{\frac{2e^{(l)}}{\left(\sum_{j=1}^{M} w_j\right)^2}\ln\left|\frac{d(x_l, c_\alpha)}{\sigma_\alpha}\right|(1-W_\alpha)\left[\sum_{k=1}^{M} w_{kl}\left(g_{\alpha l}-g_{kl}\right)\right]\right\} \quad (3.35)$$

3.2.2. Khởi tạo tự động của các quy tắc suy luận mờ, thuật toán Gustafson – Kessel

Một trong những vấn đề quan trọng nhất trong mạng TSK là xác định số lượng các quy tắc có thể sử dụng được trong mẫu dữ liệu. Càng nhiều quy tắc có nghĩa là biểu diễn dữ liệu càng tốt, nhưng nó cũng làm tăng độ phức tạp của mạng, làm tăng chi phí xử lý dữ liệu. Mạng điều khiển quá phức tạp có thể dẫn tới giảm khả năng tổng quát hoá, làm giảm chất lượng hoạt động của mạng. Để giải quyết với nhiều đầu ra, có thể viết các đẳng thức tương tự cho mỗi đầu ra. Tuy nhiên, điều đó có nghĩa là làm tăng độ phức tạp của giải pháp. Vì vậy vấn đề quan trọng nhất là giảm số lượng các quy tắc suy luận bằng cách loại bỏ sự kết hợp với những vùng dữ liệu trống. Giải quyết vấn đề này, áp dụng thuật toán tự tổ chức mờ của dữ liệu bằng thuật toán Gustafson – Kessel để xác định vị trí của các mẫu.

Ký hiệu các vectơ dữ liệu trong tập hợp là $x_i \in R^N$ với $i = 1, 2, ..., p$. Ta sẽ phân chia các vectơ này thành M nhóm, mỗi nhóm được đại diện bởi vectơ trọng tâm $c_i = \left[c_{i1}, c_{i2}, ..., c_{iN}\right]^T$. Đặt $U = \left[u_{ij}\right] \in R^{M\times p}$ là ma trận các

hệ số phụ thuộc u_{ij} - mức độ phụ thuộc của x_j $(j = 1, 2, ...p)$ vào nhóm thứ $i (i = 1, 2, ..., M)$. Thuật toán chia nhóm được thực hiện với hàm mục tiêu E được định nghĩa theo công thức:

$$E = \sum_{i=1}^{M} \sum_{j=1}^{p} u_{ij}^{m} d^2(x_j, c_i) \tag{3.36}$$

Một phân chia tốt sẽ có giá trị hàm E nhỏ.

Các thuật toán tối ưu hóa E được xây dựng với các ràng buộc $\sum_{i=1}^{M} u_{ij} = 1$ với j = 1, 2,..., p và $0 \le \sum_{j=1}^{p} u_{ij} \le p$ với i = 1, 2,..., M. Tham số m dùng để điều chỉnh tính mờ của các nhóm (thường thì m = 2). Hàm $d(x_j, c_i)$ xác định khoảng cách giữa vectơ dữ liệu x_j và mẫu c_i của nhóm thứ i. Một trong những thuật toán mờ hiệu quả nhất là thuật toán Gustaffson – Kessel (G-K). Thuật toán G-K được thể hiện với các bước sau :

1. Khởi tạo tạm thời một cách ngẫu nhiên các trọng tâm c_i với $i = 1, 2, ..., M$, tính ma trận U.

2. Xác định vị trí các trọng tâm theo công thức

$$c_i = \frac{\sum_{j=1}^{p} u_{ij}^{m} \mathbf{x}_j}{\sum_{j=1}^{p} u_{ij}^{m}} \tag{3.37}$$

3. Tính các hiệp biến nhóm $F^{(i)}$ và ma trận S_i $(i = 1, 2, ..., M)$ theo

$$F_i = \frac{\sum_{j=1}^{p} u_{ij}^{m} (x_j - c_i)(x_j - c_i)^T}{\sum_{j=1}^{p} u_{ij}^{m}} \tag{3.38}$$

$$S_i = \sqrt[N]{\det(F_i)}.[F_i]^{-1} \tag{3.39}$$

4. Ước tính khoảng cách $d_{ij}^2 (i = 1, 2..., p)$ giữa vectơ đầu vào x_j và các mẫu nhóm c_i

$$d_{ij}^2 = (x_j - c_i)^T.S_i.(x_j - c_i) \tag{3.40}$$

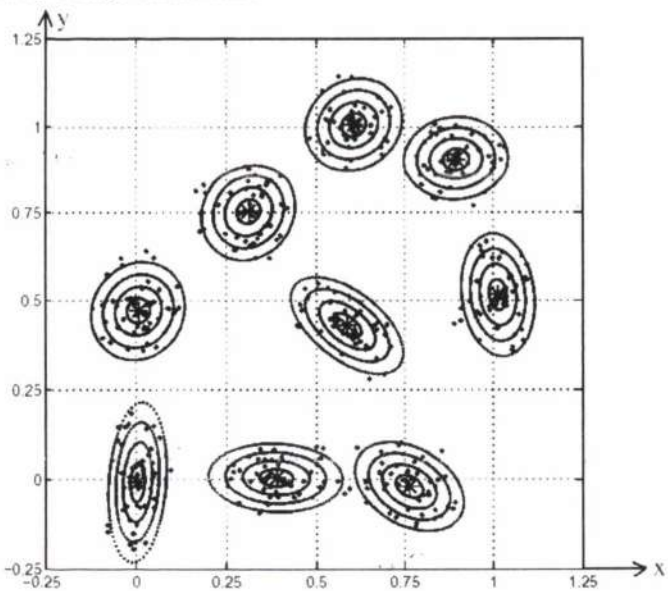
5. Xác định các ma trận đầu vào $u_{ij}(i = 1,2,\ldots,M$ và $j = 1,2,\ldots,p)$ theo quy tắc

$$u_{ij} = \frac{1}{\sum_{k=1}^{M}\left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{m-1}}} \tag{3.41}$$

Nếu $d_{ij} = 0$ trong trường hợp $i = I$, lấy $u_{Ij} = 1$ và $u_{ij} = 0$ cho $I \neq i$. Lặp lại tới khi $\left\|U^{l} - U^{l-1}\right\| \leq \varepsilon$ cho hai bước lặp lại liên tiếp.

Hình 3.9 là một ví dụ về tổ chức không gian dữ liệu thành các nhóm dữ liệu bằng thuật toán G-K, trong đó các điểm trong một nhóm nằm trên cùng một đường elip sẽ có giá trị liên thuộc bằng nhau.

Thuật toán G-K được dùng để khởi tạo quá trình học của mạng nơron. Sau khi áp dụng thuật toán G-K, các vùng dữ liệu sẽ được thiết lập, đồng thời xác định được các giá trị liên thuộc ban đầu của các điểm dữ liệu trong các tập dữ liệu học. Quá trình học tiếp theo của mạng TSK sẽ điều chỉnh lại các giá trị liên thuộc của các điểm dữ liệu cũng như các giá trị tuyến tính a_{ij} để kết quả học được chính xác.



Hình 3.9. **Phân bố các nhóm dữ liệu bằng thuật toán G-K**

3.2.3. Xác định số lượng nhóm thông qua việc phối hợp 6 chỉ số thống kê

Nhiệm vụ ước lượng số nhóm tập trung của số liệu và vị trí phân bố của các nhóm là một trong những vấn đề cơ bản của các bài toán xử lý tín hiệu nói chung và xác suất thống kê nói riêng. Đã có rất nhiều các công trình nghiên cứu về nhiệm vụ này và các công trình nghiên cứu đã đề xuất nhiều công thức chỉ số thống kê khác nhau. Tuy nhiên, cũng theo các nghiên cứu thì mỗi chỉ số thống kê thường có tác dụng cho những trường hợp phân bố số liệu nhất định. Do đó các đánh giá vẫn chủ yếu là theo tính kinh nghiệm, chưa có kết luận khẳng định chính thức về sự vượt trội của chỉ số nào so với các chỉ số còn lại trong mọi trường hợp phân bố số liệu.

Với thực tế như vậy, một hướng đề xuất mới hiện nay là cần sử dụng phối hợp nhiều chỉ số đồng thời để nâng cao được độ tin cậy của việc ước lượng. Một số công trình trước đây đã sử dụng phối hợp đến 4 chỉ số, trong nội dung này chúng tôi đề xuất sử dụng thêm 2 chỉ số nữa tổng cộng là 6 chỉ số để ước lượng số nhóm trong phân bố số liệu.

Thông qua một số ví dụ minh chứng hiệu quả cao hơn của việc sử dụng 6 chỉ số so với sử dụng 4 chỉ số. 6 chỉ số được lựa chọn là:

1. Chỉ số thể tích mờ của nhóm V_h

$$V_h = \sum_{i=1}^{M} \sqrt{\det(\mathbf{F}_i)} \tag{3.42}$$

2. Chỉ số mật độ phân bố mờ trung bình D_A

$$D_A = \frac{1}{M} \sum_{i=1}^{M} \frac{SS_i}{\sqrt{\det(\mathbf{S}_i)}} \tag{3.43}$$

Với các tham số SS_i $(i = 1, 2, \ldots, M)$ chỉ được tính toán cho các vectơ x_k, với phạm vi là độ lệch chuẩn của các đặc tính nhóm và được xác định như sau:

$$SS_i = \sum_k u_{ik} \text{ cho từng biến } k \text{ mà } (x_k - c_i)^T.\left[S^{(i)}\right]^{-1}.(x_k - c_i) < 1.$$

3. Chỉ số trung bình khoảng cách nhóm D_w

$$D_w = \frac{1}{M}\sum_{i=1}^{M}\frac{\sum_{k=1}^{p} u_{ik}^{m} d_{ik}^{2}}{\sum_{k=1}^{p} u_{ik}^{m}} \tag{3.44}$$

4. Chỉ số trung bình độ phẳng của nhóm t_A

$$t_A = \frac{1}{M}\sum_{i=1}^{M} t_i \tag{3.45}$$

Trong đó $t_i = \lambda_{in} / \lambda_{il}$ với λ_{ij} là giá trị thứ j ma trận hiệp biến F_i, sắp xếp theo thứ tự giảm dần, ví dụ: $\lambda_{i1} \geq \lambda_{i2} \geq ... \geq \lambda_{iN}$. Một phân chia tốt khi chỉ số của V_h và t_A đạt giá trị min, các chỉ số của D_A và D_W đạt giá trị max.

5. Chỉ số PBM

$$PBM(k) = \left[\frac{1}{k}\frac{E_1}{E_k}D_k\right]^2 \tag{3.46}$$

Trong đó: k là số trọng tâm;

E_1 là tổng khoảng cách có trọng số từ mọi điểm số liệu đến một trọng tâm C_0 duy nhất.

$$c_0 = \frac{1}{p}\sum_{i=1}^{p} x_i \tag{3.47}$$

$$E_1 = \sum_{i=1}^{p}\|x_i - c_0\| \tag{3.48}$$

p là số điểm dữ liệu

E_k là tổng khoảng cách nội nhóm được định nghĩa bởi:

$$E_k = \sum_{i=1}^{k}\left(\sum_{i=1}^{p} u_{ij}\|x_i - c_j\|^2\right) \tag{3.49}$$

với u_{ij} là hệ số phụ thuộc của vectơ x_i vào trọng tâm C_j và được tính bằng

$$u_{ij} = \frac{\frac{1}{d_{ij}^2}}{\sum_{k=1}^{M}\frac{1}{d_{kj}^2}} \tag{3.50}$$

D_k là khoảng cách lớn nhất giữa hai trọng tâm

$$D_k = \max_{i,j=1,\dots k} \|c_i - c_j\| \tag{3.51}$$

Số trọng tâm tối ưu (một phân chia tốt) sẽ được tìm cho giá trị tương ứng với PBM đạt cực đại.

6. Chỉ số *DN*:

Có bộ mẫu $\{x_i\}$ với i=1,2,...,p; có M là số trọng tâm, từ đó ta có ma trận các hệ số phụ thuộc mờ của vectơ x_i vào trọng tâm C_j, tiếp sau ta tính được các đại lượng

$$D_j(U;C) = \frac{1}{p}\sum_{i=1}^{p}\left|u_{ij} - (u_{ij})_T\right| \tag{3.52}$$

với

$$\left(u_{ij}\right)_T = \begin{cases} 1 & \text{khi} \quad u_{ij} > \dfrac{1}{M} \\ 0 & \text{khi} \quad u_{ij} \le \dfrac{1}{M} \end{cases} \tag{3.53}$$

$$D(U;C) = \sum_{j=1}^{M} D_j(U,C) \tag{3.54}$$

$$DC = \frac{1}{M-1} D(U,C) \tag{3.55}$$

Sau đó ta tính chỉ số Lattice

$$N(\tilde{F}_i, \tilde{F}_k) = \min\left[(\tilde{F}_i \circ \tilde{F}_k), 1 - (\tilde{F}_i \hat{\circ} \tilde{F}_k)\right] \tag{3.56}$$

trong đó theo định nghĩa ta có

$$\tilde{F}_i \circ \tilde{F}_k = \min_{j=1,\dots,M}\left[\max(u_{ij}, u_{kj})\right] \tag{3.57}$$

$$\tilde{F}_i \hat{\circ} \tilde{F}_k = \max_{j=1,\dots,M}\left[\min(u_{ij}, u_{kj})\right] \tag{3.58}$$

Từ các giá trị $N(\tilde{F}_i, \tilde{F}_k)$ ta có:

$$NM = \max_{i,k} N(\tilde{F}_i, \tilde{F}_k) \tag{3.59}$$

Cuối cùng ta có

$$DN(U,C) = \begin{cases} 0 & DC = NM = 0 \\ \dfrac{2 * DC * NM}{DC + NM} & \text{Các trường hợp khác} \end{cases} \tag{3.60}$$

Một phân chia tốt sẽ có giá trị DN nhỏ.

Để phối hợp sử dụng đồng thời 6 chỉ số này, chúng tôi đề xuất một công thức gọi là "chỉ số tổng hợp": α

$$\alpha = a_1 V_h - a_2 D_A - a_3 D_W + a_4 t_A - a_5 PBM + a_6 DN \qquad (3.61)$$

Xuất phát từ lý do như đã trình bày ở trên, mặt khác hiện nay cũng chưa có đề xuất nào mới về việc xác định thêm các chỉ số thống kê khác (ngoài 6 chỉ số thống kê nêu trên), nên chúng tôi đề xuất việc phối hợp 6 chỉ số để tính "chỉ số tổng hợp" α. Đồng thời chưa có kết luận chính thức nào trong các công trình nghiên cứu về hiệu quả của các chỉ số thống kê trong trường hợp tổng quát, vì vậy việc phối hợp các chỉ số tạm dựa trên nguyên tắc là *các chỉ số có ảnh hưởng đồng đều*. Do đó các hệ số a_i của từng chỉ số thống kê được chọn là nghịch đảo của giá trị lớn nhất của chỉ số đó, để đảm bảo trong công thức tính chỉ số tổng hợp mỗi chỉ số thống kê sẽ có khoảng giá trị biến thiên giống nhau là đoạn [0,1]. Ví dụ $a_1 = \dfrac{1}{\max\{V_h\}}$ (max được tính cho tất cả các chỉ số thống kê đang được xét), khi đó ta có $a_1 V_h \in [0, 1]$. Đồng thời, tiêu chí "số nhóm tốt" của mỗi chỉ số được ứng với giá trị min (V_h, t_A và DN) hoặc giá trị max (D_W, D_A và PBM), vì vậy để thuận tiện cho việc tính toán và lựa chọn, nên thống nhất sử dụng tiêu chí giá trị của "chỉ số tổng hợp" đạt: "*giá trị min ứng với số nhóm tốt*". Để có được giá trị α_{min} thì các chỉ số V_h, t_A và DN lấy hệ số với dấu "+", chỉ số D_W, D_A và PBM lấy hệ số với dấu "–".

3.2.4. Đặt giá trị ban đầu cho các hàm suy luận

Sau khi số lượng trọng tâm đã được ước lượng và vị trí ban đầu của các trọng tâm được tính toán từ thuật toán G-K, ta cần khởi tạo các giá trị còn lại của các luật suy luận mờ. Một luật mờ có ba thông số cơ bản là $c_j^{(i)}, b_j^{(i)}, \sigma_j^{(i)}$, một thông số $c_j^{(i)}$ đã được xác định, do đó còn hai thông số $b_j^{(i)}, \sigma_j^{(i)}$ cần phải tiếp tục khởi tạo. Một giá trị nhỏ của σ cũng làm cho quy tắc "chính xác" hơn, nhưng giá trị lớn sẽ làm cho quy tắc trở nên linh hoạt và thiết thực hơn đối với phạm vi rộng của đầu vào. Giá trị của nó sẽ phụ

thuộc vào việc phân bố dữ liệu và khoảng cách giữa các vị trí mẫu. Phương pháp đặt giá trị ban đầu cho các giá trị của σ được thực hiện như sau:

1. Đối với mỗi một trọng tâm c_i, tính khoảng cách cho tất cả các mẫu khác sử dụng.

2. Tính hệ số tỷ lệ $R_{ef}^{(i)}$ bằng cách lấy khoảng cách trung bình từ trọng tâm c_i tới K (thường chọn K = 5) mẫu số liệu gần nhất khi có hơn 5 trọng tâm hoặc K = M – 1 khi số trọng tâm M < 6.

3. Đặt giá trị ban đầu σ_i tới $R_{ef}^{(i)}/k$, hệ số b_i chọn bằng 1 để có được hàm dạng hình chuông.

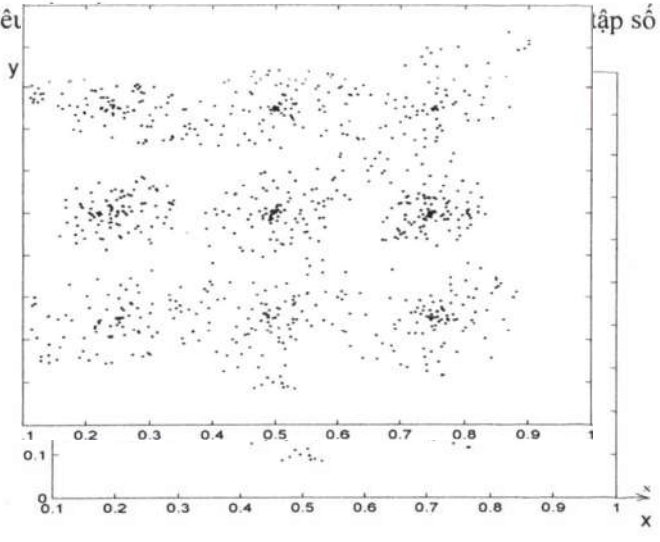
Để làm rõ cho các đề xuất nêu trên ta tiến hành thử nghiệm trên một số tập số liệu phân bố trên mặt phẳng.

3.2.5. Kết quả thử nghiệm

Thử nghiệm 1: Có tập số liệu mẫu phân bố trên mặt phẳng như hình 3.10.

– Nhiệm vụ phân vùng tập số liệu mẫu thành các nhóm.

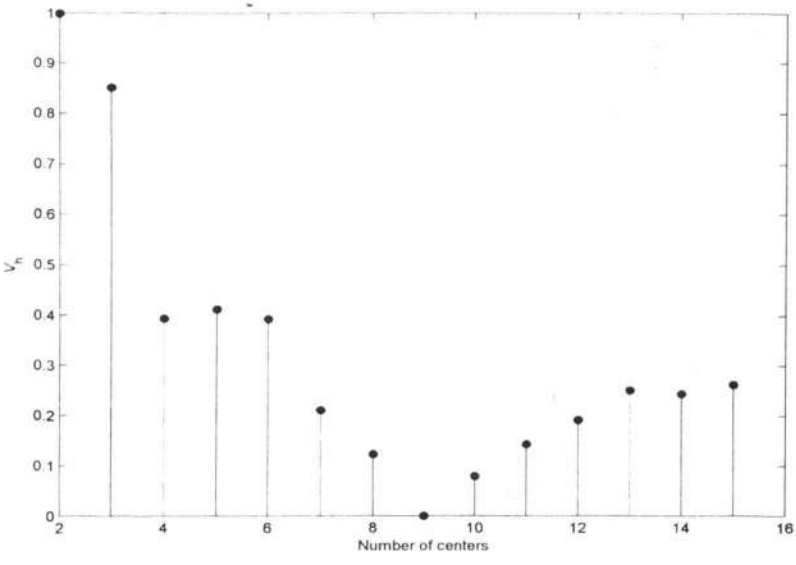
– Yêu tập số liệu.



Hình 3.10. **Tập số liệu mẫu phân bố trên mặt phẳng**

– Cho số nhóm C biến thiên từ 2 đến 15 với 6 chỉ số: V_h, D_A, D_W, t_A, PBM, DN.

– Vẽ biểu đồ biến thiên của chỉ số V_h theo công thức (3.42). Sử dụng hàm function res = Vh_all (X, M_max) được xây dựng trong Matlab.

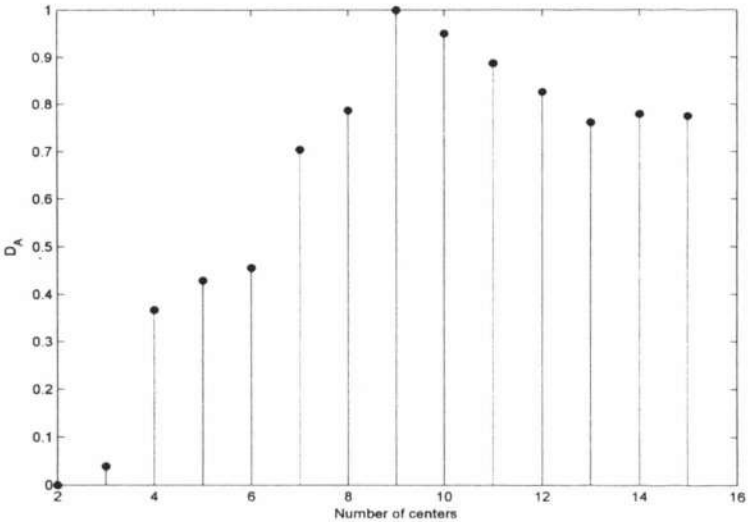


Hình 3.11. **Biểu đồ biến thiên chỉ theo số V_h**

Biểu đồ hình 3.11 cho thấy khi V_h nhận giá trị nhỏ nhất tương ứng số nhóm bằng 9. Căn cứ vào chỉ số V_h chọn số nhóm bằng 9 (tương ứng với 9 luật).

– Vẽ biểu đồ biến thiên của chỉ số D_A biến thiên theo công thức 3.43. Sử dụng hàm function res = DA_all (X, M_max) được xây dựng trong Matlab.

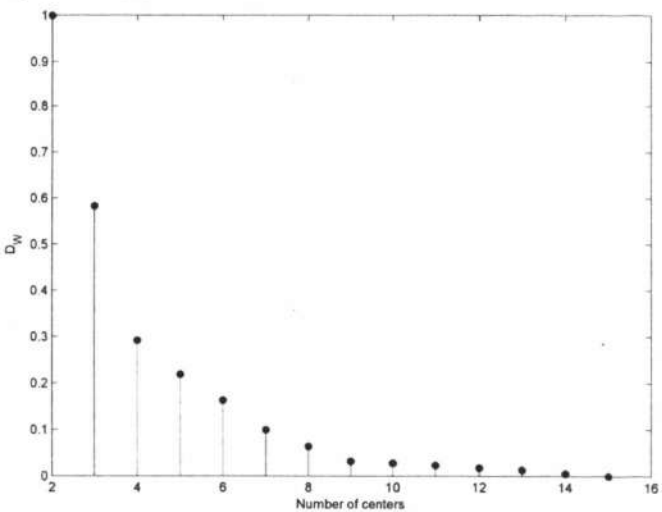
Qua biểu đồ hình 3.12 thấy D_A có giá trị gần bằng 1 là giá trị lớn nhất thì số nhóm tương ứng bằng 9. Căn cứ vào chỉ số D_A chọn số nhóm bằng 9 (tương ứng với 9 luật).



Hình 3.12. **Biểu đồ biến thiên theo chỉ số D_A**

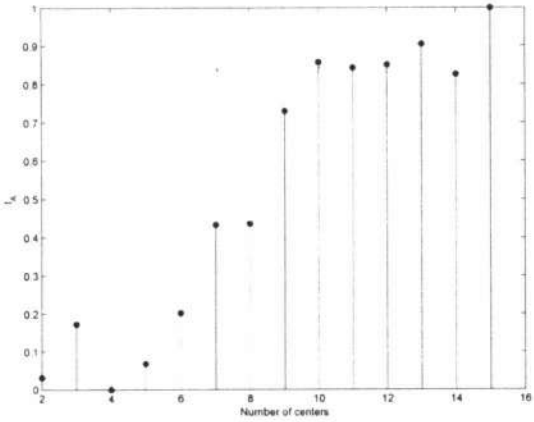
– Vẽ biểu đồ biến thiên của chỉ số D_w biến thiên theo công thức 3.44. Sử dụng hàm function res = DW_all (X, M_max) được vẽ trong Matlab.

Qua biểu đồ hình 3.13 thấy D_w xấp xỉ gần bằng 0,6 là giá trị lớn nhất thì số nhóm tương ứng bằng 2. Căn cứ vào chỉ số D_w chọn số nhóm bằng 2 (tương ứng với 2 luật).



Hình 3.13. **Biểu đồ biến thiên theo chỉ số D_w**

Vẽ biểu đồ biến thiên của chỉ số t_A theo công thức (3.45). Sử dụng hàm function res = tA_all (X, M_max) được vẽ trong Matlab.

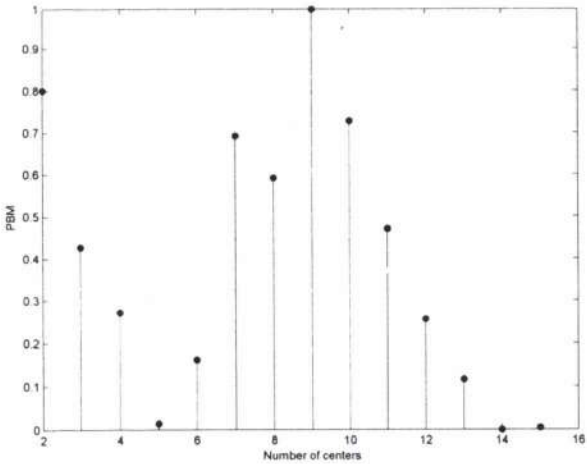


Hình 3.14. **Biểu đồ biến thiên theo chỉ số t_A**

Qua biểu đồ hình 3.14 thấy $t_A = 0$ là giá trị nhỏ nhất thì số nhóm tương ứng bằng 4. Căn cứ vào chỉ số t_A chọn số nhóm bằng 4 (tương ứng với 4 luật).

– Vẽ biểu đồ biến thiên chỉ số PBM theo công thức (3.46). Sử dụng hàm function res = PBM_all (X, M_max) được vẽ trong Matlab.

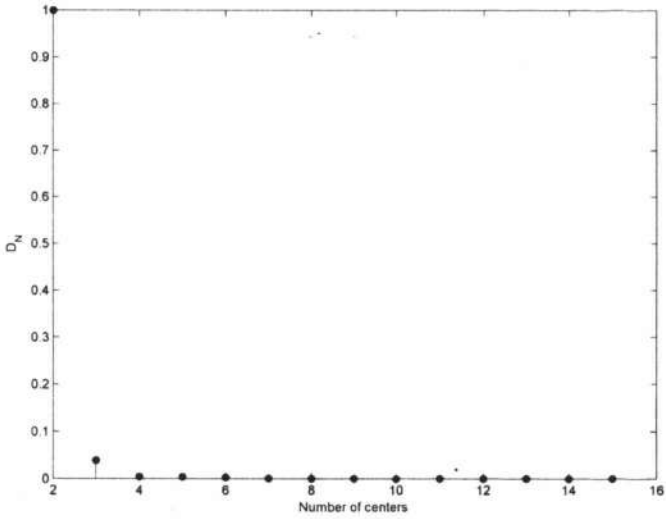
Qua biểu đồ hình 3.15 thấy giá trị PBM xấp xỉ bằng 1 tương ứng với số nhóm bằng 9 là phân chia tốt



Hình 3.15. **Biểu đồ biến thiên theo chỉ số PBM**

– Vẽ biểu đồ biến thiên chỉ số DN theo công thức (3.60). Sử dụng hàm function res = DN_all (X, M_max) được vẽ trong Matlab.

Biểu đồ hình 3.16 cho thấy chỉ số DN đạt giá trị nhỏ xấp xỉ bằng 0 là phân chia tốt (tương ứng với số nhóm trong khoảng từ 4 – 15).



Hình 3.16. **Biểu đồ biến thiên theo chỉ số DN**

– Vẽ biểu đồ biến thiên của chỉ số tổng hợp α theo công thức (3.61). Sử dụng hàm function res = Alpha_all (Vh_all, DA_all, DW_all, tA_all, PBM_all, DN_all,M_max)

Tham số đầu vào:

X - ma trận hai chiều chứa tập các vectơ đầu vào cần phân nhóm.

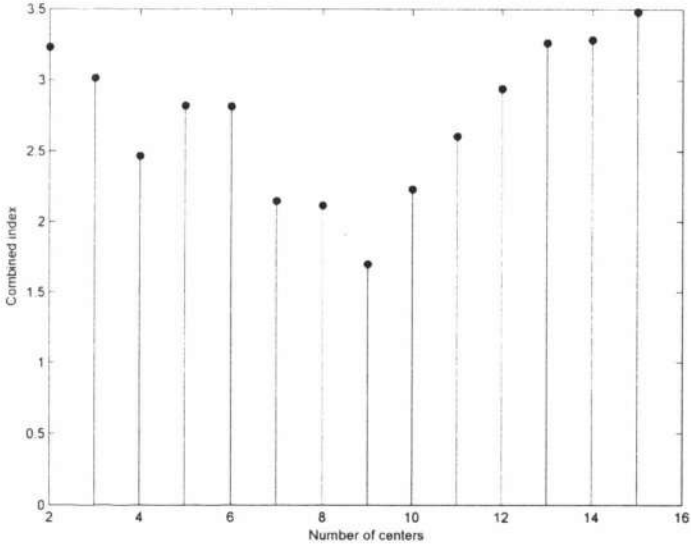
Vh_all – vectơ chứa các giá trị của chỉ số Vh ứng với số trọng tâm biến thiên từ 2 đến M_max.

M_max – số trọng tâm lớn nhất cần xác định chỉ số (hàm sẽ tính cho số trọng tâm trong khoảng từ 2 đến M_max).

Kết quả đầu ra:

res – vectơ chứa các giá trị của chỉ số α ứng với số trọng tâm biến thiên từ 2 đến M_max.

Qua biểu đồ hình 2.17 cho thấy α = 1,75 là giá trị nhỏ nhất thì số nhóm tương ứng bằng 9. Căn cứ vào chỉ số α chọn số nhóm bằng 9 (tương ứng với 9 luật).



Hình 3.17. **Biểu đồ biến thiên theo chỉ số tổng hợp α**

– Xác định trọng tâm của các vùng số liệu theo phương pháp G-K. Sử dụng hàm function [C, U] = gk_one_c(X,M)

Tham số đầu vào:

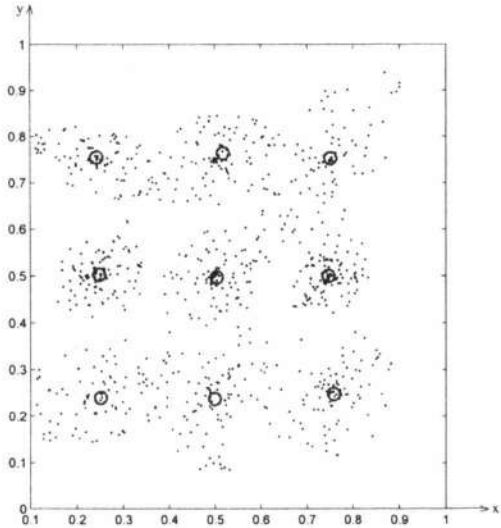
X – ma trận hai chiều chứa tập các vectơ đầu vào cần phân nhóm.

M – số trọng tâm cần tìm.

Kết quả đầu ra:

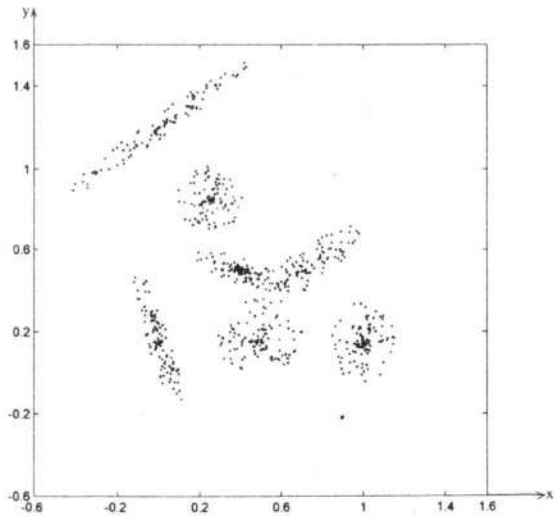
U – ma trận hệ số phụ thuộc của vectơ đầu vào x_i vào trọng tâm c_j.

Phát triển trong Matlab ta có vùng số liệu phân bố như hình 3.18. Nếu phân chia tập số liệu mẫu theo 9 nhóm tương ứng với 9 quy tắc sẽ bao quát hết tập số liệu, đạt được độ chính xác cao trong nhận dạng.



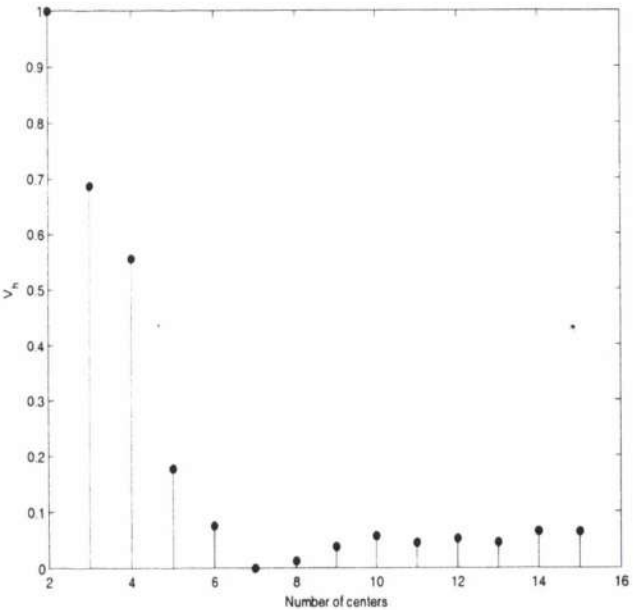
Hình 3.18. **Phân chia tập số liệu mẫu theo 9 nhóm**

Thử nghiệm 2: Xét tập số liệu với phân bố như trên hình 3.19.

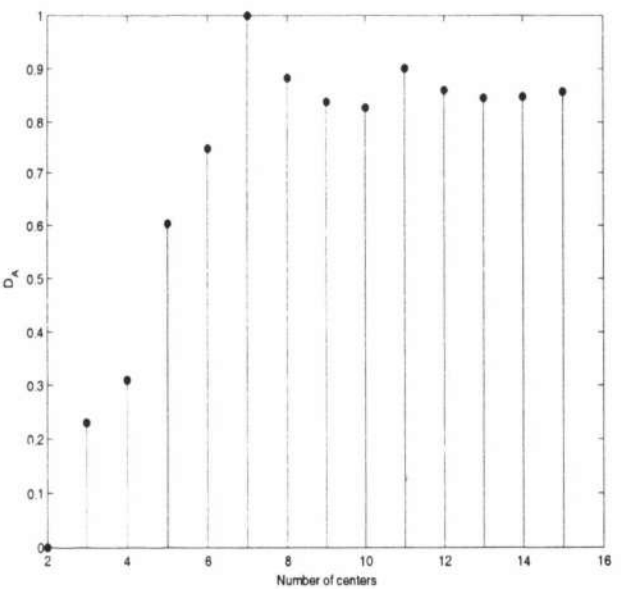


Hình 3.19. **Tập số liệu mẫu với 7 nhóm số liệu**

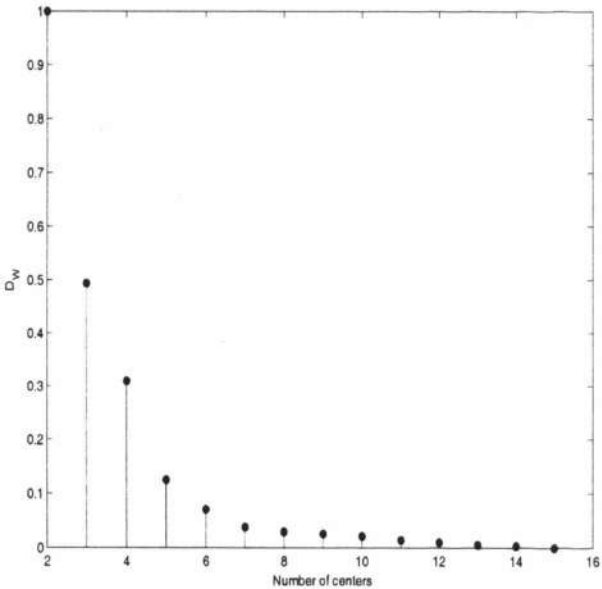
Với bộ số liệu trên, thuật toán tính toán các chỉ số thống kê đã được áp dụng cho số trọng tâm biến thiên từ 2 đến 15. Kết quả thu được trình bày trên hình 3.20.



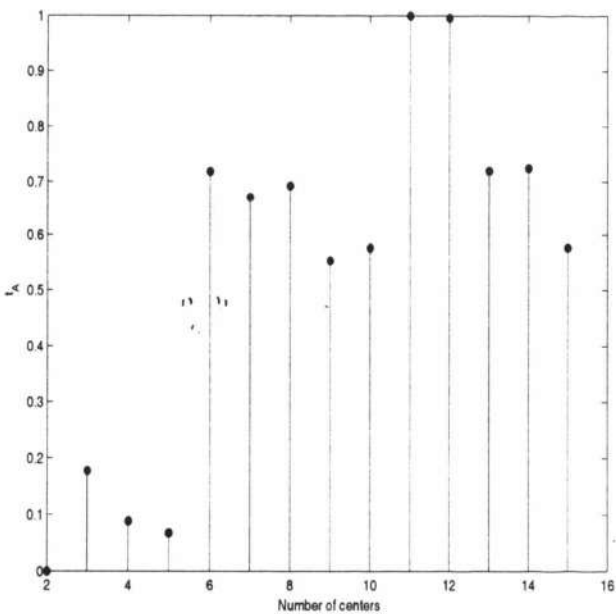
Hình 3.20 (a)



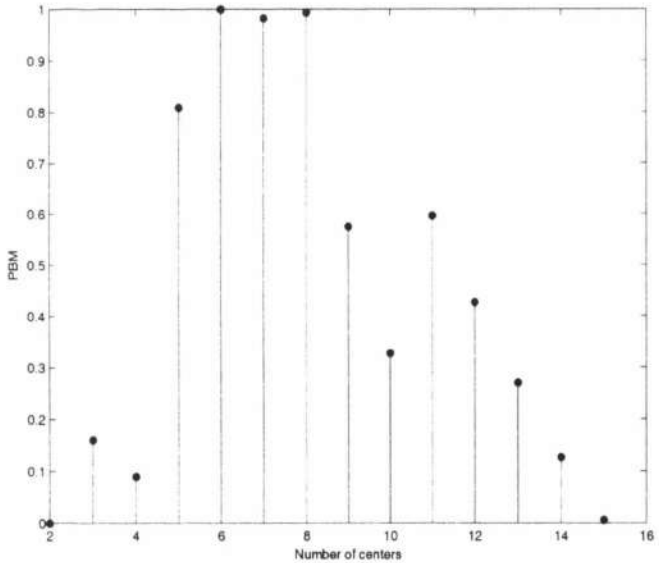
Hình 3.20 (b)



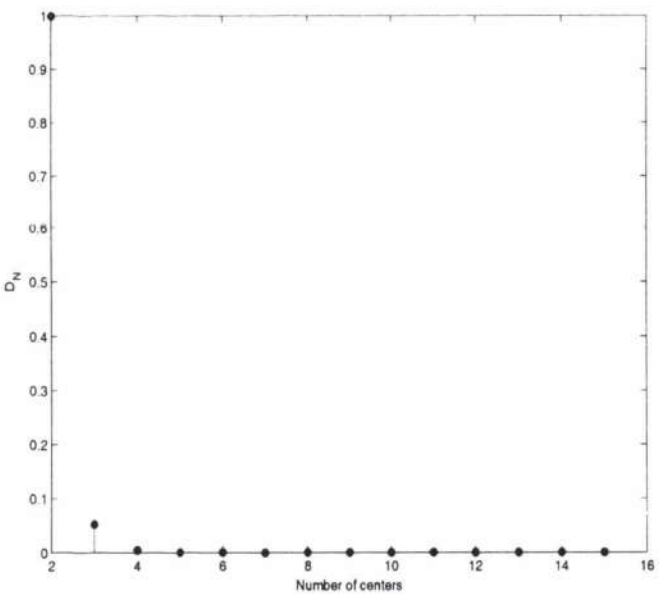
Hình 3.20 (c)



Hình 3.20 (d)



Hình 3.20 (e)

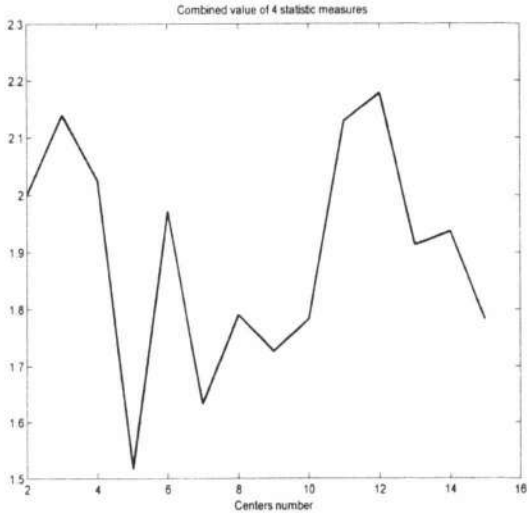


Hình 3.20 (f)

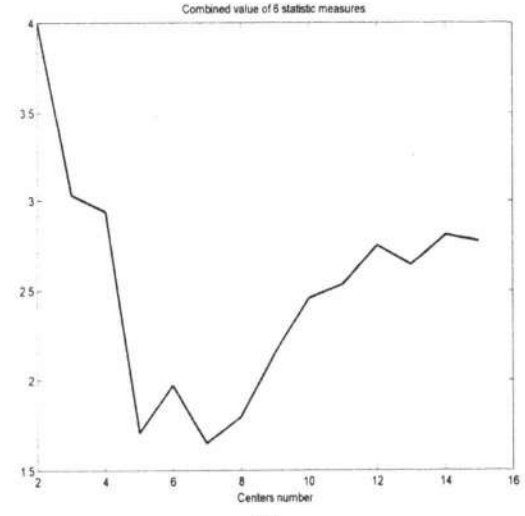
Hình 3.20. **Giá trị các chỉ số V_h (a), D_A (b), D_W (c), t_A (d), PBM (e) và DN (f)**

Để thuận tiện cho việc so sánh về chất lượng sử dụng phối hợp các chỉ số thống kê, ta xét hai trường hợp:

– Biểu đồ biến thiên chỉ số tổng hợp ứng với việc phối hợp 4 chỉ số và 6 chỉ số (hình 3.21).



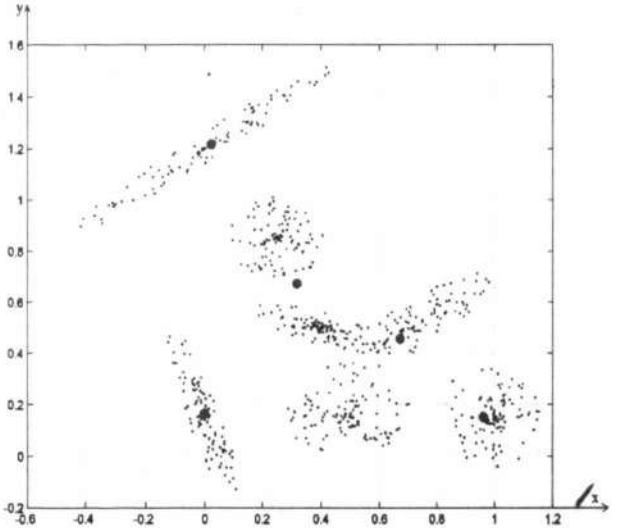
(a)



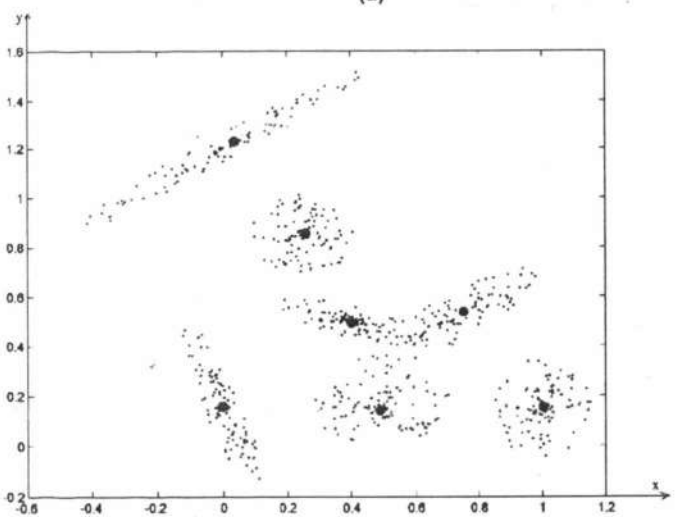
(b)

Hình 3.21. **Sự biến thiên chỉ số tổng hợp khi phối hợp các chỉ số khác nhau: Phối hợp 4 chỉ số số (a) và phối hợp 6 chỉ số (b)**

– Hình 3.22 cho thấy việc phối hợp 6 chỉ số đã phát hiện chính xác số vùng số liệu là 7 (ứng với giá trị cực tiểu bé nhất của chỉ số). Tuy nhiên, khi phối hợp 4 chỉ số thì số vùng số liệu tốt nhất phát hiện được là 5 (số vùng 7 chỉ đứng ở vị trí thứ 1 nếu xét theo mức độ ưu tiên).



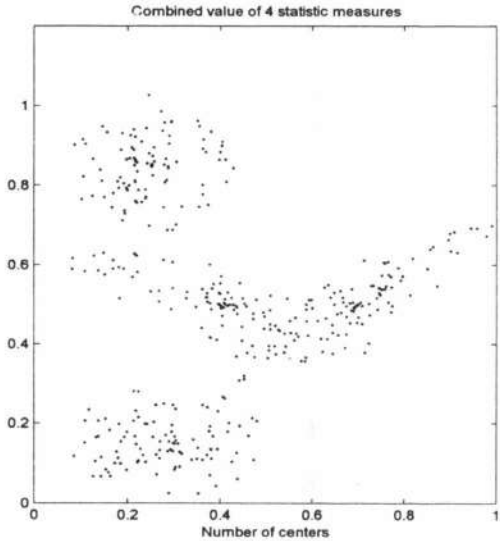
(a)



(b)

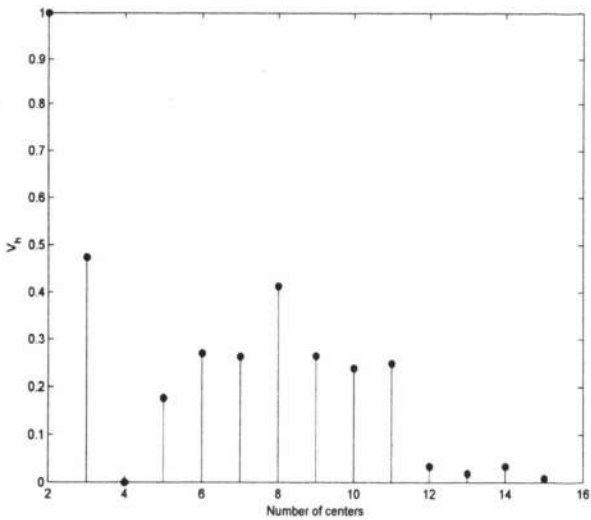
Hình 3.22. **Bộ số liệu ứng với số trọng tâm bằng 5 (a) và trọng tâm bằng 7 (b)**

Thử nghiệm 3: Xét tập số liệu với phân bố như trên hình 3.23.

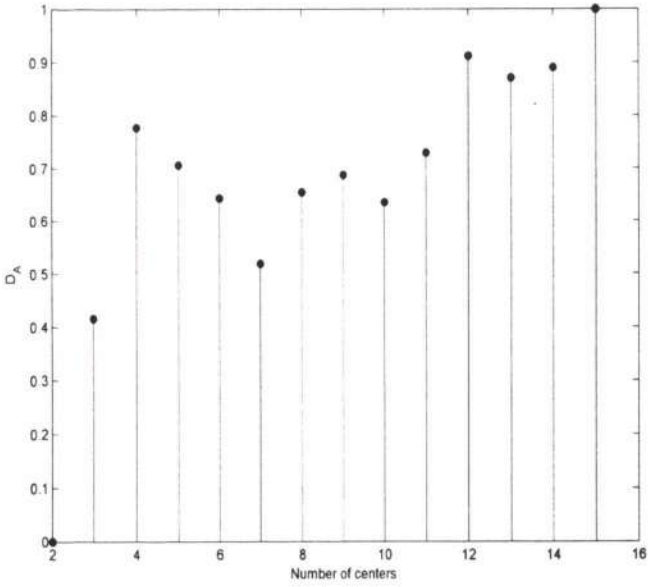


Hình 3.23. **Tập số liệu mẫu với 4 nhóm số liệu**

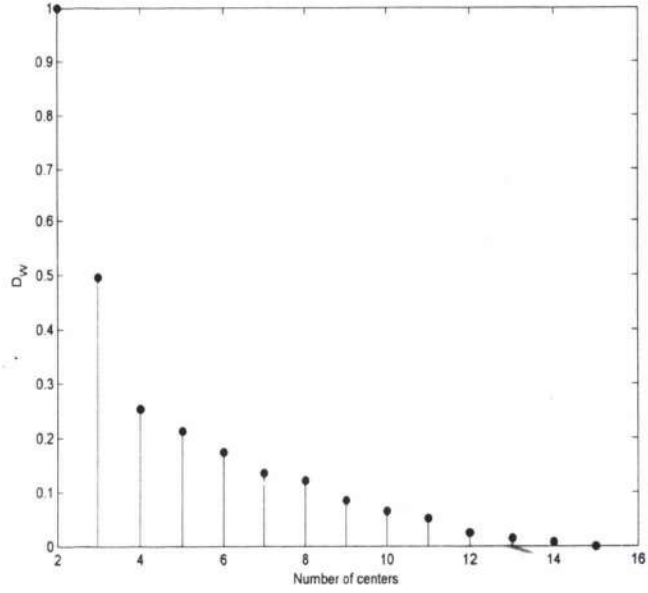
Với bộ số liệu trên, thuật toán tính toán các chỉ số thống kê đã được áp dụng cho số trọng tâm biến thiên từ 2 đến 15. Kết quả thu được được trình bày trên hình 3.24.



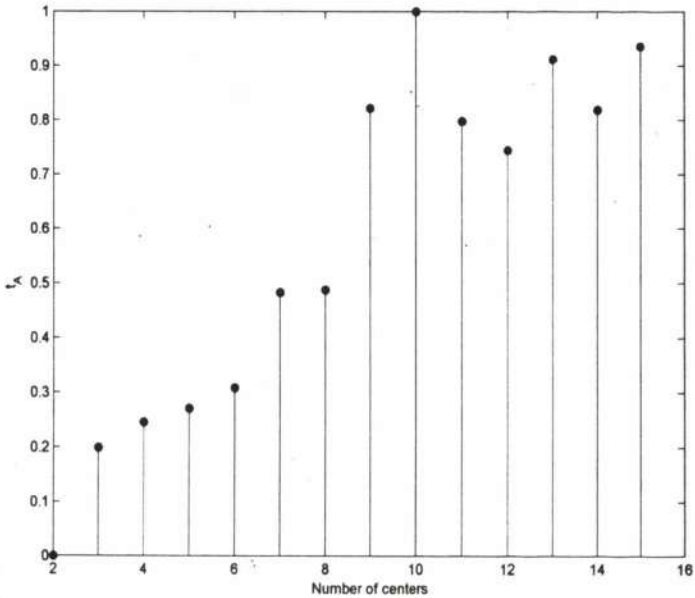
Hình 3.24 (a)



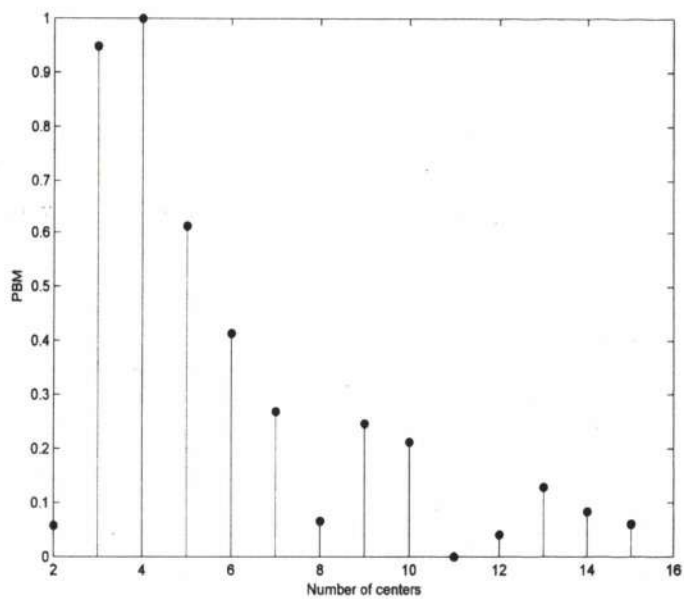
Hinh 3.24 (b)



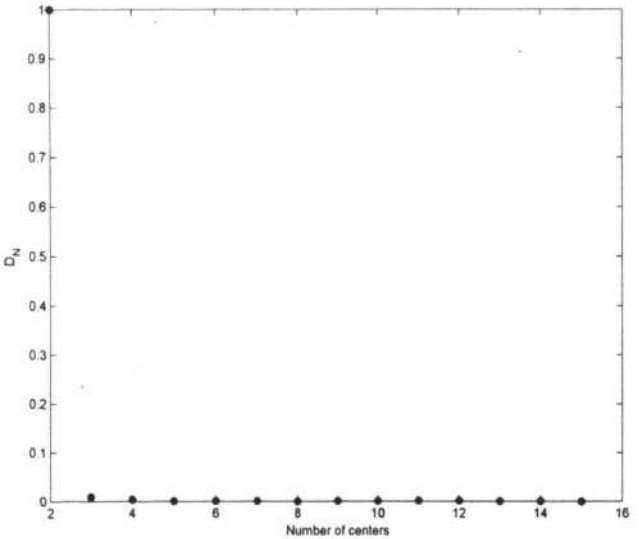
Hinh 3.24 (c)



Hình 3.24 (d)



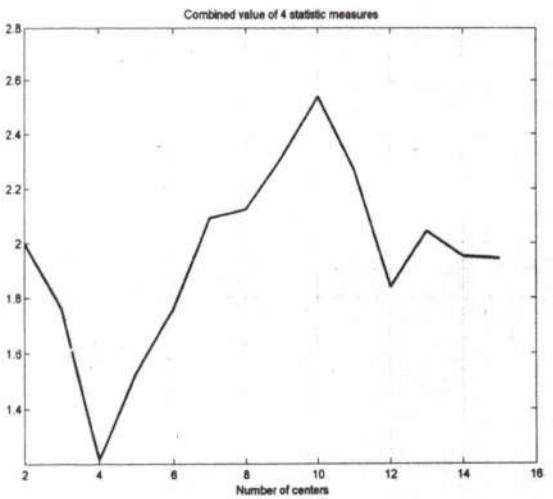
Hình 3.24 (e)



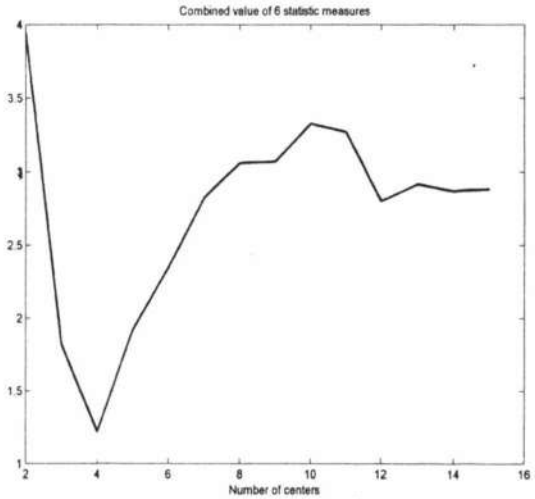
Hình 3.24 (f)

Hình 3.24. **Giá trị chỉ số thống kê V_h(a), D_A(b), D_W(c), t_A(d), PBM (e), DN(f)**

Để thuận tiện cho việc so sánh về chất lượng sử dụng phối hợp các chỉ số thống kê, phương pháp sử dụng phối hợp 6 chỉ số sẽ được so sánh với phương pháp sử dụng phối hợp 4 chỉ số. Giá trị biến thiên của chỉ số tổng hợp trong hai trường hợp này được trình bày trên hình 3.25.



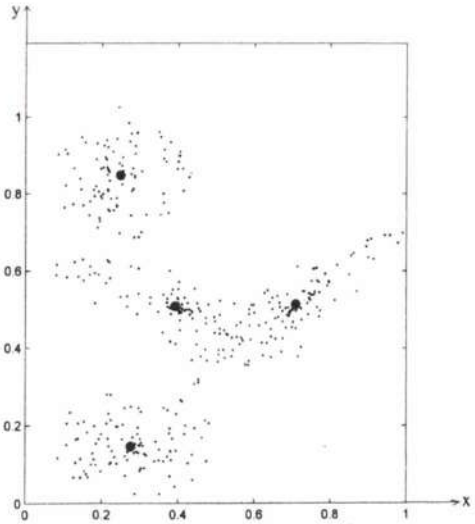
Hình 3.25 (a)



Hình 3.25 (b)

Hình 3.25. **Sự biến thiên chỉ số tổng hợp khi phối hợp các chỉ số khác nhau: Phối hợp 4 chỉ số (a) và phối hợp 6 chỉ số (b)**

Theo hình 3.25 ta nhận thấy trong trường hợp này phối hợp 4 chỉ số và 6 chỉ số đều đã phát hiện chính xác số vùng số liệu là 4 (ứng với giá trị cực tiểu của chỉ số tổng hợp). Hình 3.26 là sự phân bố 4 trọng tâm theo tập số liệu mẫu hình 3.23.



Hình 3.26. **Phân bố 4 trọng tâm cho bộ số liệu của hình 3.23.**

Qua một số kết quả thử nghiệm đã được minh họa như trên, ta có thể thấy hiệu quả của phương pháp xác định số các nhóm số liệu trong không gian đầu vào thông qua việc sử dụng phối hợp nhiều chỉ số thống kê. Đồng thời phương pháp sử dụng phối hợp nhiều (6 chỉ số thống kê) sẽ hiệu quả hơn việc sử dụng ít các chỉ số thống kê thành phần.

Phần III
MỘT SỐ ỨNG DỤNG MẠNG NƠRON TRONG NHẬN DẠNG VÀ DỰ BÁO

Chương 4
ỨNG DỤNG MẠNG NƠRON TRONG NHẬN DẠNG VÀ DỰ BÁO

4.1. NHẬN DẠNG VÀ CÁC PHƯƠNG PHÁP TIẾP CẬN

4.1.1. Khái niệm

Nhận dạng là quá trình phân loại các đối tượng được biểu diễn theo một mô hình nào đó và gán cho chúng vào một lớp dựa theo các quy luật và các mẫu chuẩn. Quá trình nhận dạng dựa vào những mẫu học biết trước gọi là nhận dạng có thầy hay học có thầy, trường hợp ngược lại dựa vào những mẫu học không biết trước gọi là nhận dạng không có thầy hay học không có thầy.

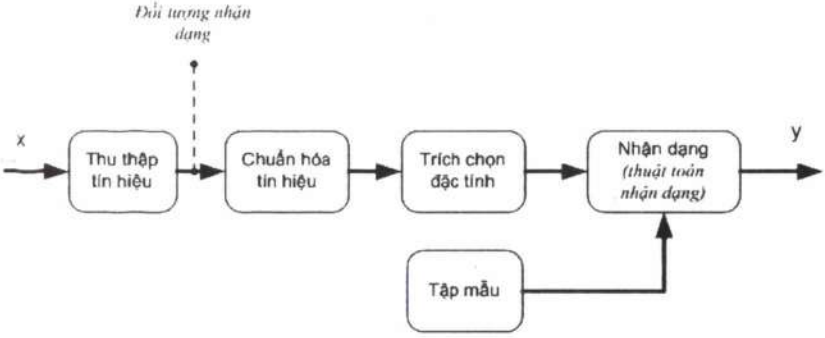
Bản chất của quá trình nhận dạng gồm 3 giai đoạn chính:

+ Lựa chọn mô hình biểu diễn đối tượng.

+ Lựa chọn luật ra quyết định (phương pháp nhận dạng) và suy diễn quá trình học.

+ Học nhận dạng.

Mô hình tổng quát hệ thống nhận dạng được thể hiện như hình 4.1.



Hình 4.1. **Mô hình tổng quát của hệ thống nhận dạng**

– Khối thu thập tín hiệu: Nhiệm vụ thu thập tín hiệu cần nhận dạng, được thực hiện bởi các bộ thu thập tín hiệu như camera số.

– Chuẩn hóa tín hiệu: Đưa tín hiệu về dạng chuẩn được quy định trước.

– Trích chọn đặc tính: Lựa chọn một số đặc tính đặc trưng cho đại lượng cần nhận dạng của tín hiệu, nhằm hạn chế số lượng thông tin thừa.

– Nhận dạng: So sánh đối chiếu đặc tính của đối tượng cần nhận dạng và các mẫu, kết quả đưa ra là mẫu có "khoảng cách" gần nhất tới đối tượng.

4.1.2. Các phương pháp tiếp cận

Trong lý thuyết nhận dạng nói chung, nhận dạng ảnh nói riêng có 3 cách tiếp cận khác nhau:

– Nhận dạng dựa vào phân hoạch không gian: Trong kỹ thuật này, các đối tượng nhận dạng là các đối tượng định lượng, mỗi đối tượng được biểu diễn bởi một vectơ nhiều chiều.

– Nhận dạng cấu trúc.

– Nhận dạng dựa vào kỹ thuật mạng nơron.

Nhận dạng dựa vào phân hoạch không gian và nhận dạng cấu trúc là các kỹ thuật kinh điển. Các đối tượng ảnh quan sát và thu nhận được phải trải qua giai đoạn tiền xử lý nhằm tăng cường chất lượng, làm nổi các chi tiết và cuối cùng mới qua giai đoạn nhận dạng.

Nhận dạng dựa vào kỹ thuật mạng nơron hoàn toàn khác, nó dựa vào cơ chế đoán nhận, lưu trữ và phân biệt đối tượng mô phỏng theo hoạt động của hệ thần kinh con người. Do cơ chế đặc biệt, các đối tượng thu nhận không cần qua giai đoạn cải thiện mà chuyển ngay sang giai đoạn tổng hợp, đối sánh với các mẫu đã lưu trữ để nhận dạng.

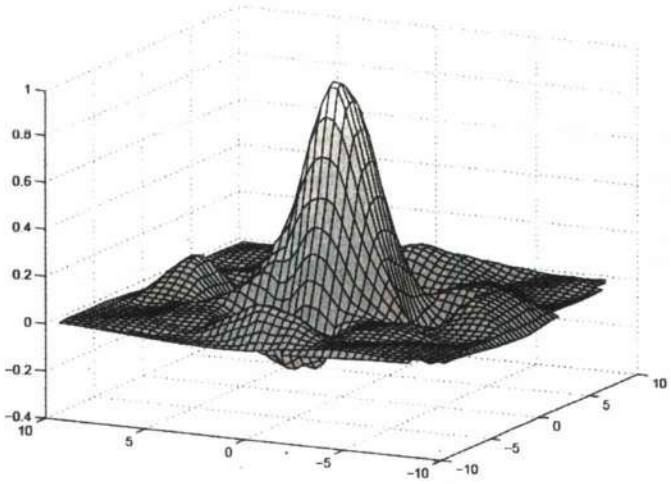
4.2. ỨNG DỤNG TRONG NHẬN DẠNG HÀM PHI TUYẾN

4.2.1. Ứng dụng mạng TSK trong xấp xỉ hàm phi tuyến

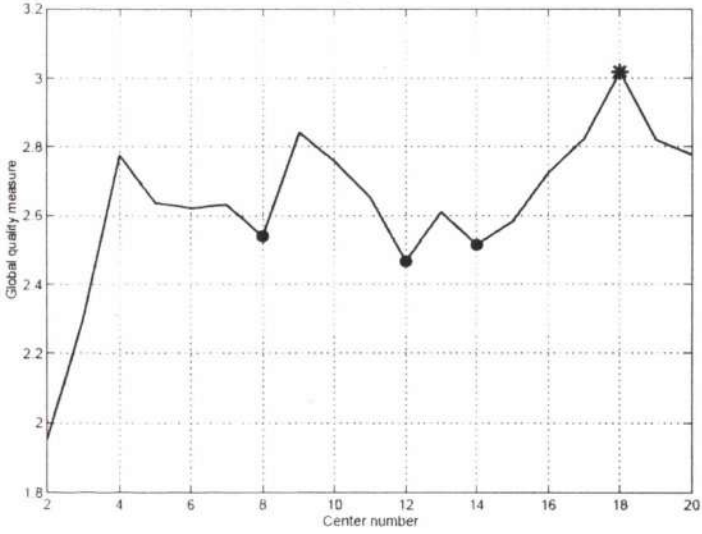
Ví dụ 1: Ứng dụng trong xấp xỉ hàm

$$y(x_1, x_2) = \frac{\sin(x_1).\sin(x_2)}{x_1.x_2} \tag{4.1}$$

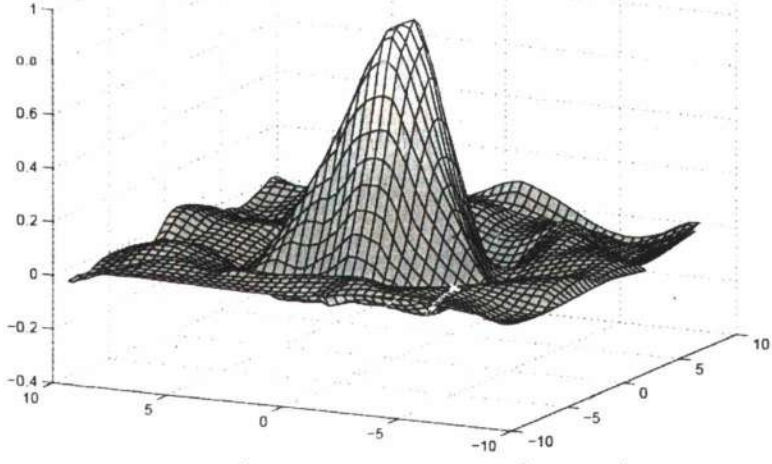
Việc làm xấp xỉ đúng đã thực hiện bằng các bước nêu trên và kết quả đã xác nhận khả năng hoạt động tốt của chúng. Công việc đầu tiên được ghi nhận ở đây là làm xấp xỉ đúng các hàm tĩnh học của hai biến số xác định trong công thức (4.1). Trong phạm vi biến đổi đầu vào [−10,10] cho cả x_1, x_2.



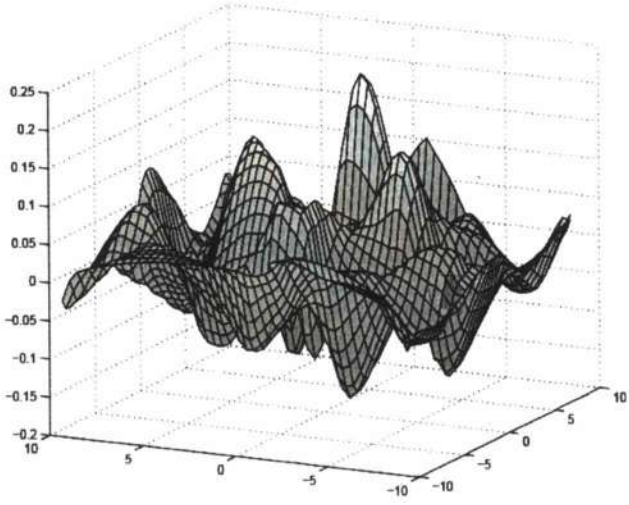
Hình 4.2a. **Hàm mẫu 3-D**



Hình 4.2b. **Biểu đồ chỉ số tổng hợp của hàm mẫu**



Hình 4.3. **Kết quả tái tạo ánh xạ phi tuyến hàm gốc**



Hình 4.4. **Lỗi xấp xỉ tại 12 quy tắc suy luận**

Bảng kết quả kiểm tra mạng TSK với một số nhóm khác nhau:

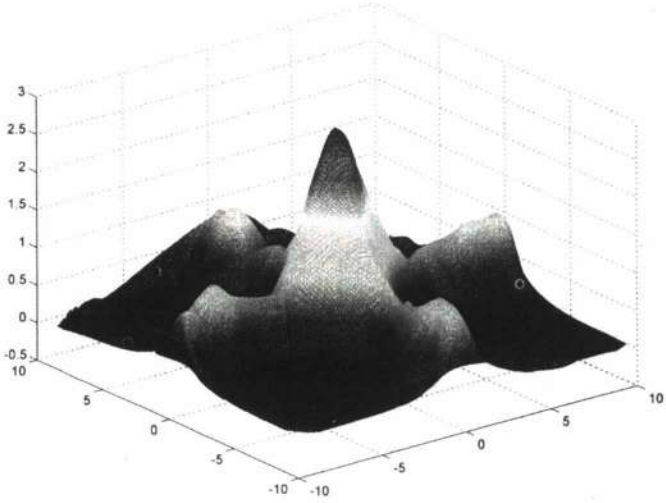
Số nhóm	*Sai số học*	*Sai số kiểm tra*
8	0,3642	0,6324
12	0,2406	0,3422
18	0,0543	0,7753

Ví dụ 2: Ứng dụng trong xấp xỉ hàm phi tuyến

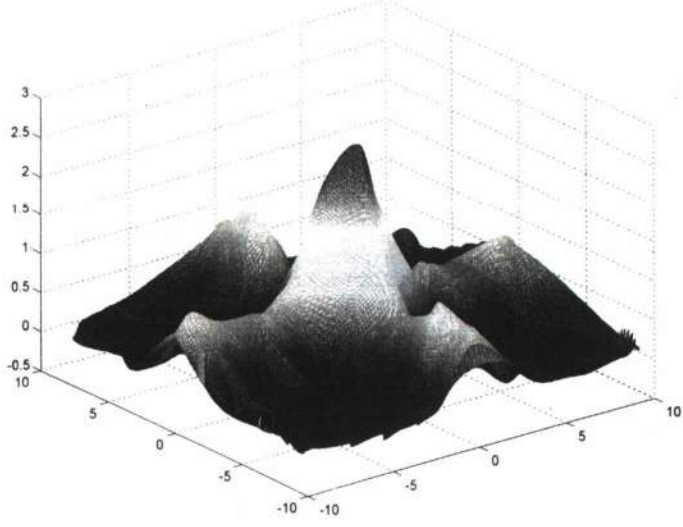
$$z(x,y) = \frac{\sin(x)\cdot\sin(y)}{x\cdot y} + \frac{1}{1+x^2} + \frac{1}{1+y^2} \tag{4.2}$$

Tập số liệu học: 200 điểm (x, y) ngẫu nhiên trong đoạn $[-10,10]\times[-10,10]$.

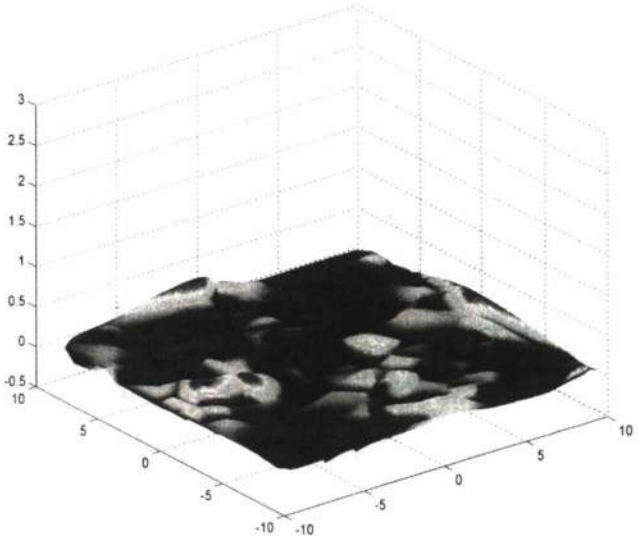
Tập số liệu kiểm tra: 200 điểm (x, y) ngẫu nhiên trong đoạn $[-10,10]\times[-10,10]$ khác với tập số liệu học. Kết quả:



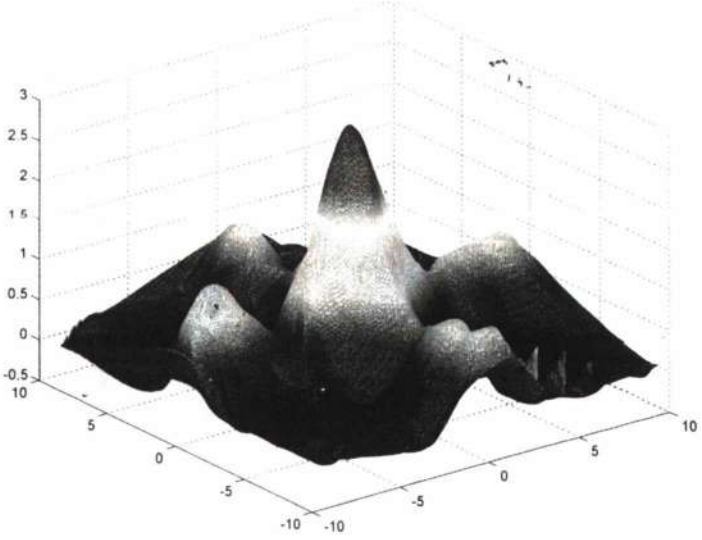
Hình 4.5. **Mẫu hàm gốc ban đầu**



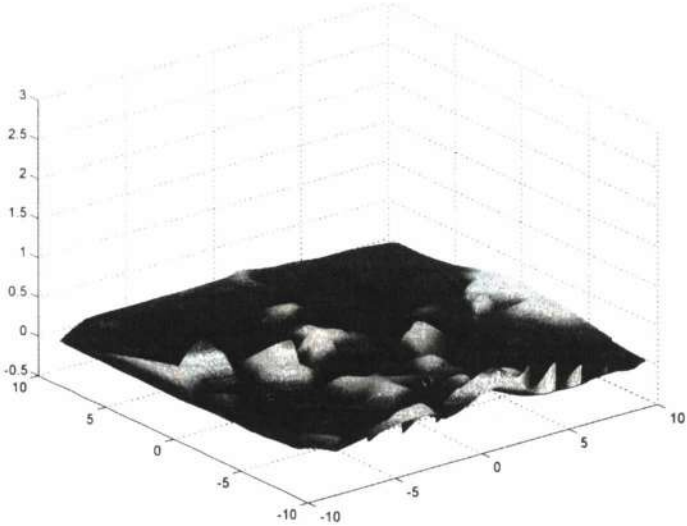
Hình 4.6. **Hàm sau khi học**



Hình 4.7. **Sai số trên bộ số liệu học**



Hình 4.8. **Kết quả kiểm tra trên bộ số liệu test**



Hình 4.9. **Sai số trên bộ số liệu test**

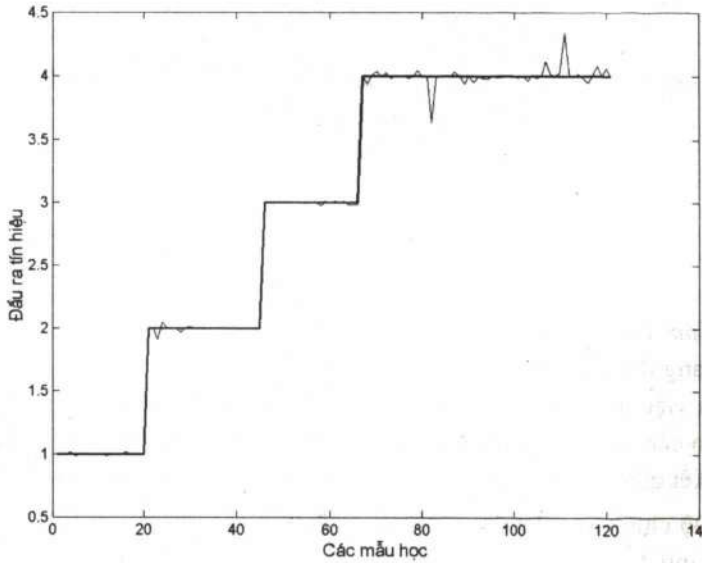
Tóm lại: Trong nội dung phần này chúng ta quan tâm đến cấu trúc mới mạng điều khiển mờ TSK và thuật toán của nó trong nghiên cứu, bao gồm cả việc lựa chọn số lượng tối ưu của các mẫu nhóm và gán giá trị ban đầu cho các tham số của các hàm mờ. Mạng hiệu chỉnh đã được hoàn thiện và đạt kết quả kiểm tra tốt.

Độ chính xác đạt được của phép lấy xấp xỉ tốt hơn so với trường hợp mạng kinh điển. Về vấn đề phân loại, số lượng các tham số không tuyến tính đã giảm đáng kể, điều này làm cho quá trình nghiên cứu dễ dàng hơn và đạt kết quả tốt hơn. Kết quả thực nghiệm đã chứng minh mạng hiệu chỉnh hoạt động tốt.

Mạng TSK có khả năng xấp xỉ hàm phi tuyến với độ chính xác bất kỳ. Để ứng dụng mạng TSK trong bài toán nhận dạng với các đầu ra mẫu chỉ nhận một trong các giá trị rời rạc đặc trưng cho các lớp, còn đầu ra của mạng có thể nhận các giá trị liên tục. Ta cần biến đổi đầu ra y của mạng thành giá trị d gần nhất. Nếu giá trị này khác với giá trị đích cần có thì mạng đã nhận dạng sai. Ngược lại ta có trường hợp mạng nhận dạng chính xác mẫu.

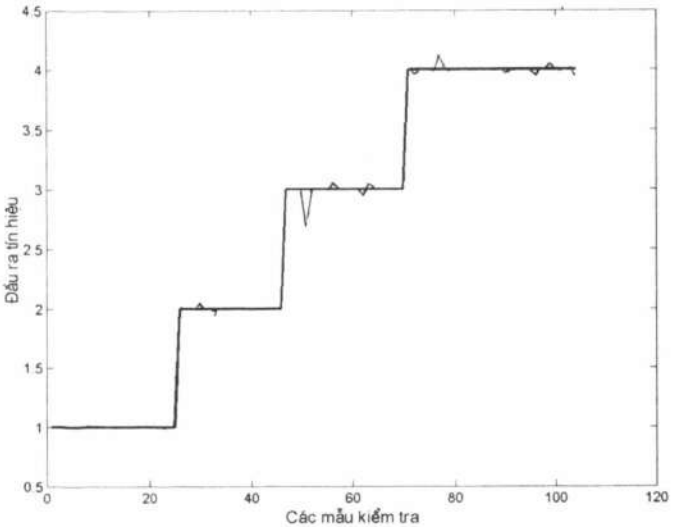
Mô hình nhận dạng bằng mạng TSK đã được xây dựng trong môi trường Matlab và được thử nghiệm với tập số liệu với 225 mẫu, trong đó

121 mẫu được sử dụng để tạo mô hình, còn 104 mẫu để kiểm tra mô hình. Mạng TSK được sử dụng với $M = 15$ quy tắc suy luận. Quá trình học được thực hiện với 100 lần lặp, kết quả của quá trình học được thể hiện trên hình 4.10. Các giá trị đích có thể là 1, 2, 3 hoặc 4. Trên hình 4.10 ta có đầu ra của mạng TSK đã bám sát với các giá trị đích, các sai số đều nhỏ hơn 0,5 là giá trị ngưỡng của sai số nhận dạng.



Hình 4.10. **Đầu ra tín hiệu đích và đầu ra tín hiệu từ mạng TSK cho các số liệu "học"**

Mạng sau khi được huấn luyện đã được kiểm tra với tập 104 số liệu khác với các mẫu đã học. Hình 4.11 mô tả đầu ra của mạng cùng với các giá trị đích. Cũng tương tự như trong trường hợp kết quả học, khi đưa vào mạng TSK các số liệu từ các mẫu của tập kiểm tra, ta có thể dễ dàng nhận thấy các tín hiệu đầu ra bám sát với giá trị đích, sai số tuyệt đối không vượt quá 0,5 – giá trị ngưỡng của sai số nhận dạng. Với các kết quả như trên, cả trong trường hợp xây dựng cũng như kiểm tra, mô hình TSK đều cho kết quả nhận dạng chính xác 99%.



Hình 4.11. **Đầu ra tín hiệu đích và đầu ra tín hiệu từ mạng TSK cho các số liệu "kiểm tra"**

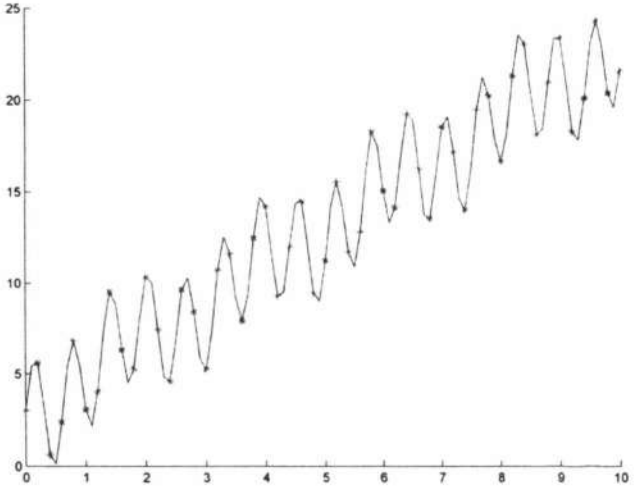
Đây là một kết quả rất khả quan và hứa hẹn khả năng xây dựng được một mô hình thực tế với độ chính xác cao.

4.2.2. Ứng dụng mô hình hỗn hợp nhận dạng hàm phi tuyến của đối tượng

Để kiểm chứng chất lượng của mô hình hỗn hợp, ta sẽ ứng dụng mô hình trong việc xấp xỉ hàm phi tuyến. Để lấy ví dụ ta sẽ sử dụng một hàm phi tuyến được mô tả theo quan hệ.

$$d = 2x + 3 + 3\sin(10x) + \sin(60x) \tag{4.3}$$

trong đoạn $x \in [0,10]$. Đây là một hàm quan hệ có chứa cả hai thành phần tuyến tính ($2x+3$) và thành phần phi tuyến ($3\sin(10x)+\sin(60x)$) với các thành phần tần số và biên độ tương ứng khác nhau. Tập số liệu học sẽ là các điểm hàm số lấy theo lưới đều với bước lấy số liệu $h = 0,2$: $x_i = i*h = 0,2i; d_i = f(x_i)$. Các giá trị kiểm tra sẽ được lấy trên cùng đoạn với bước $h = 0,1$, có nghĩa là bao gồm cả những điểm mới, chưa xuất hiện trong tập số liệu học $x_i^{kt} = i*h^{kt} = 0,1i; d_i^{kt} = f(x_i^{kt})$. Các điểm số liệu của hàm được thể hiện trên hình 4.12 là các dấu '*':



Hình 4.12. **Hàm phi tuyến dùng để kiểm tra phương pháp**

4.2.2.1. Kết quả cho thành phần tuyến tính

Ta ước lượng thành phần tuyến tính bằng cách chuyển bài toán ban đầu về bài toán tối ưu hóa sai số tuyến tính $d \approx \alpha \cdot x + \beta$. Từ các điểm số liệu ban đầu ta xây dựng được hệ phương trình tuyến tính:

$$\mathbf{A} \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \approx \mathbf{d} \text{ với } \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0,2 & 1 \\ \dots & \dots \\ 10 & 1 \end{bmatrix} \in \mathbb{R}^{51\times 2}; \tag{4.4}$$

Từ đây sử dụng thuật toán khai triển SVD ta có:

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T \tag{4.5}$$

$$\text{với } \mathbf{U} \in \mathbb{R}^{51\times 51}; \tag{4.6}$$

$$\mathbf{S} = \begin{bmatrix} 41,895 & 0 \\ 0 & 3,584 \\ 0 & 0 \\ \dots & \dots \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{51\times 2};$$

$$\mathbf{V} = \begin{bmatrix} -0,989 & 0,148 \\ -0,148 & -0,989 \end{bmatrix} \in \mathbb{R}^{2\times 2};$$

Khi đó ta có thể dễ dàng tính được ma trận giả nghịch đảo của **A** (hàm SVD trong thư viện *Signal Processing* của Matlab) và nghiệm tối ưu của hệ phương trình trên:

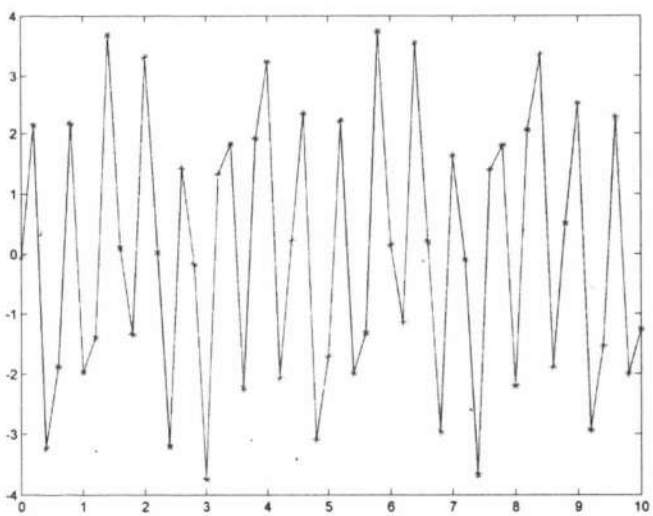
$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \mathbf{A}^{+}\mathbf{d} = \begin{bmatrix} 1,971 \\ 3,065 \end{bmatrix}$$

Dễ dàng nhận thấy nghiệm tìm được có giá trị xấp xỉ rất tốt với các hệ số thực tế là $\alpha = 2$ và $\beta = 3$.

4.2.2.2. Kết quả cho thành phần phi tuyến

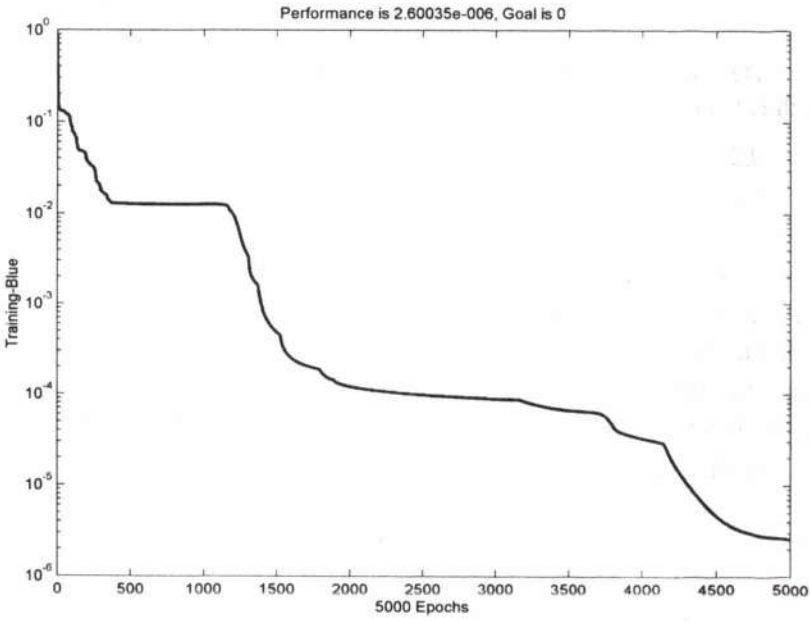
Sau khi đã ước lượng được thành phần tuyến tính, lượng sai số còn lại $d - (\alpha x + \beta) = 2x + 3 + 3\sin(10x) + \sin(60x) - (1,971x + 3,065)$ sẽ được ước lượng tiếp bằng mô hình phi tuyến mạng MLP. Bằng phương pháp thử với các số lượng nơron lớp ẩn khác nhau (lựa chọn trong khoảng từ 5 tới 30 nơron ẩn), ta đã tìm được mạng có 20 nơron ẩn có thể cho sai số nhỏ ở mức chấp nhận được. Sau đây là các kết quả học và kiểm tra mô hình mạng nơron một lớp ẩn với 20 nơron (với các hàm truyền đạt trên cả hai lớp là hàm tansig).

Số liệu đầu vào cho bài toán phi tuyến:



Hình 4.13. **Sai số (trên bộ số liệu học) còn lại sau khi đã loại trừ thành phần ước lượng tuyến tính**

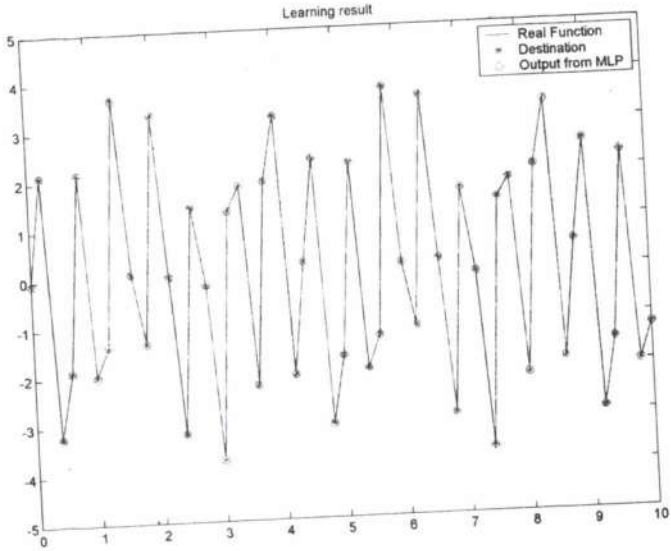
Mạng MLP được khởi tạo ngẫu nhiên (hàm newff trong thư viện Neural Network Toolbox của Matlab), sau quá trình học với 5000 bước lặp bằng thuật toán Levenberg – Marquadt (hàm trainlm trong thư viện Neural Network Toolbox của Matlab) sai số đã giảm từ khoảng 1 ($=10^0$) xuống chỉ còn khoảng $2,6.10^{-6}$ (hình 4.14). Giá trị sai số này rất thấp so với mức tín hiệu giao động trong khoảng từ $[-4,4]$.



Hình 4.14. **Đồ thị phụ thuộc của giá trị hàm sai số biến thiên trong quá trình học của mạng MLP**

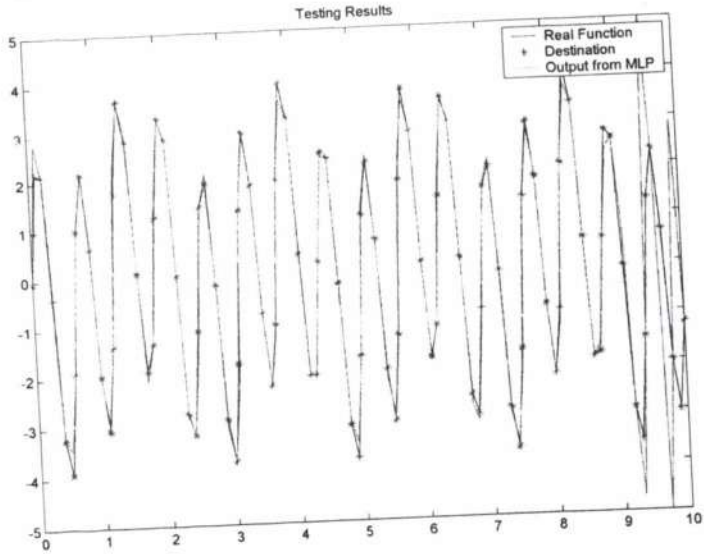
Kết quả học được thể hiện trên hình 4.15. Ta có thể nhận thấy các điểm '*' và các điểm 'o' trùng khít lên nhau, điều này thể hiện sai số học đã đạt rất thấp. Sai số trung bình:

$$E_{learn} = \sqrt{\frac{1}{51}\sum_{i=1}^{51}(y_i - d_i)^2} = 0,0081$$



Hình 4.15. **Kết quả kiểm tra mạng MLP với bộ số liệu học**

Để kiểm tra chất lượng của mô hình đối với các điểm số liệu khác của hàm số, ta tiến hành kiểm tra trên tập điểm mật độ dày hơn (khoảng cách lấy mẫu h = 0,1 như đã nêu). Kết quả thu được như sau:



Hình 4.16. **Kết quả kiểm tra mạng MLP với bộ số liệu không trùng với bộ số liệu học**

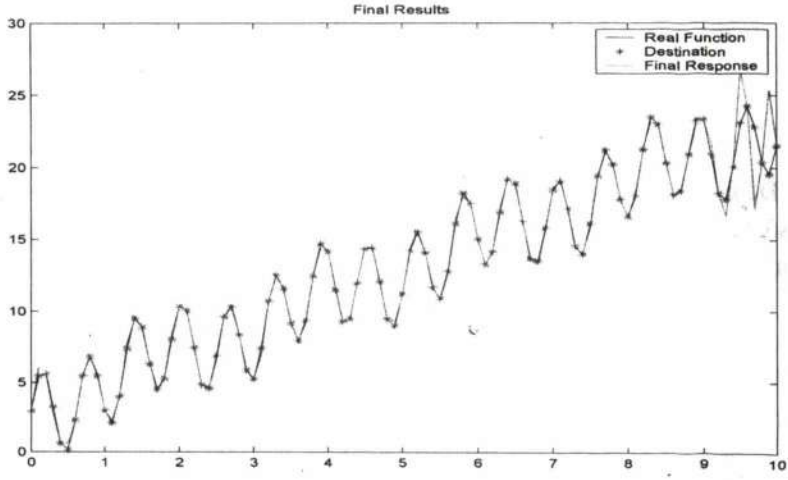
Từ hình 4.16 ta nhận thấy kết quả kiểm tra hai hàm số có độ bám sát nhau rất cao, thể hiện được khả năng tái tạo lại quan hệ phi tuyến tốt của mạng MLP.

4.2.2.3. Kết quả tổng hợp

Tổng hợp hai kết quả đạt được từ khối ước lượng thành phần tuyến tính và khối ước lượng thành phần phi tuyến ta có:

$$y_{final} = y_{SVD} + y_{MLP} \approx d$$

Hàm truyền đạt của mạng hỗn hợp tổng hợp từ khối tuyến tính và khối phi tuyến được trình bày trên hình 4.17 (đường màu đỏ). Đặc tính này bám rất sát đặc tính của hàm ban đầu (đường màu xanh).



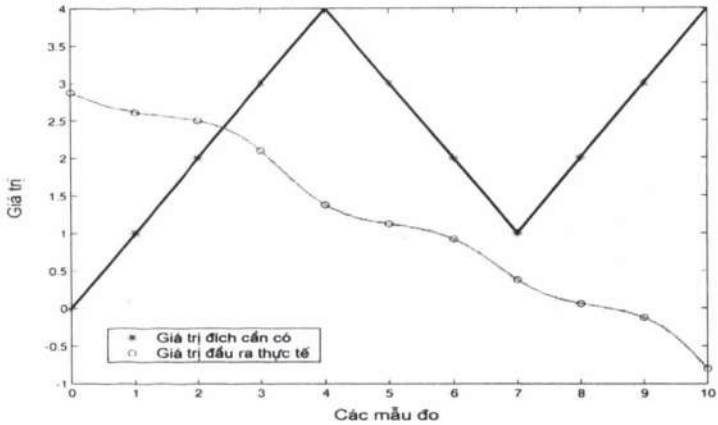
Hình 4.17. **Đáp ứng đầu ra của mạng khi tổng hợp hai thành phần tuyến tính và phi tuyến**

4.3. ỨNG DỤNG TRONG NHẬN DẠNG, PHÂN LOẠI

4.3.1. Ứng dụng mạng MLP trong nhận dạng

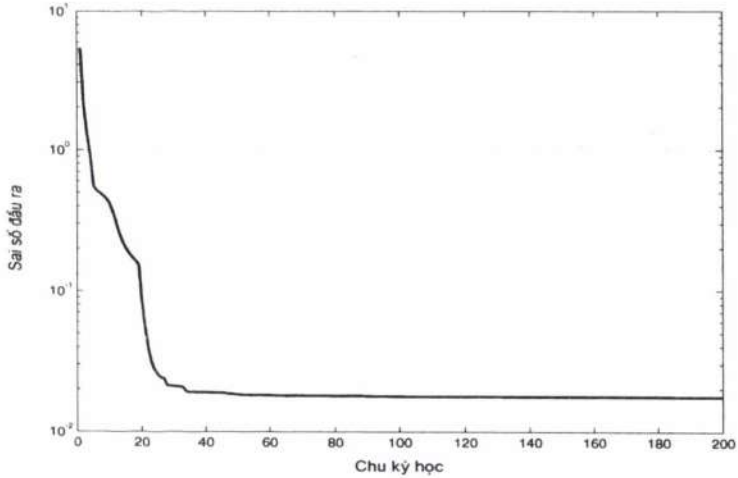
Để xác định tính hiệu quả của thuật toán trên, ta sẽ kiểm nghiệm trên một ví dụ đơn giản. Ta có tập học là 10 điểm đo của một tín hiệu hình tam giác (đường đậm nét). Trên hình 4.18 đó là các điểm hình sao (*). Tập trên sẽ được đưa vào một mạng MLP với một lớp ẩn gồm có 4 nơron ẩn. Do tập đầu vào và tập đầu ra đều chỉ có một thành phần, nên cấu trúc của mạng MLP cho bài toán này sẽ là 1 – 4 – 1. Trước quá trình học, mạng MLP có

ác trọng số ghép nối với giá trị ngẫu nhiên. Do đó, với tập đầu vào như rên, tập đầu ra tương ứng sai lệch rất lớn so với đích ban đầu như ta thấy rên hình 3.12 (đầu ra của mạng là các điểm hình tròn tại các hoành độ lựa họn nói trên, và đường nét nhạt là hàm truyền của mạng).



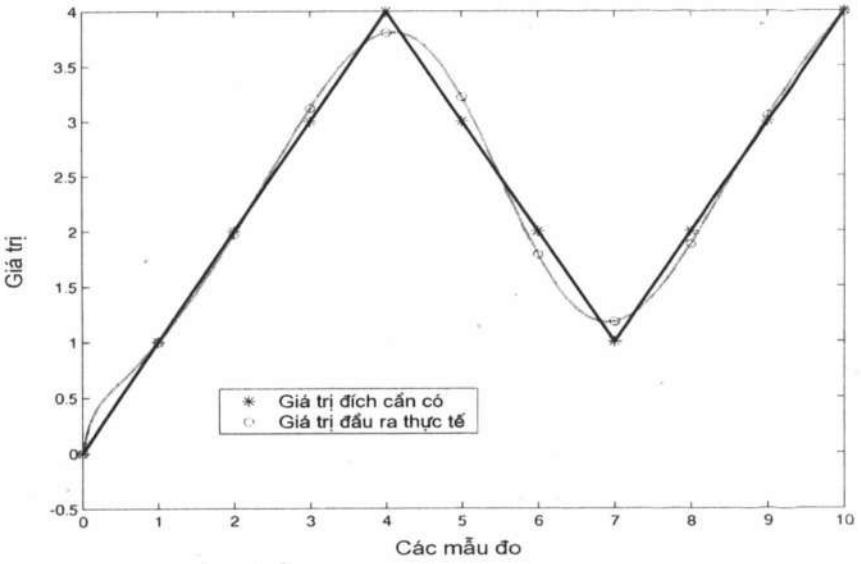
Hình 4.18. **Các giá trị mẫu (*) và các giá trị đầu ra của mạng trước khi huấn luyện (o)**

Mạng MLP đã được huấn luyện với thuật toán bước giảm cực đại với 00 chu kỳ. Đồ thị hình 4.19 cho thấy quá trình giảm của hàm mục tiêu E rong quá trình học này. Kết thúc quá trình học, E đạt giá trị bằng 0,018 nhỏ ơn rất nhiều so với ban đầu là 5,49.



Hình 4.19. **Quá trình giảm sai số trong 200 chu kỳ học đầu tiên**

Tương ứng với kết quả học này trên đồ thị hình 4.20 ta thấy các đầu ra của mạng MLP đã gần trùng khít với các giá trị đích ban đầu. Không chỉ tại những điểm này mà hàm truyền của mạng cũng gần với hàm tín hiệu hình tam giác ban đầu. Ta có thể kết luận là thuật toán học được sử dụng đã hoạt động hiệu quả.



Hình 4.20. **Các giá trị mẫu (*) và các giá trị đầu ra của mạng sau khi huấn luyện (o)**

4.3.2. Ứng dụng trong mô hình nhận dạng phân loại sản phẩm công nghiệp

Các thuật toán tự động xây dựng mô hình nhận dạng đã được kiểm nghiệm trên hai loại gạch ốp lát. Mẫu hoa văn của hai loại gạch này được thể hiện trên hình 4.21.

(a)

(b)

Hình 4.21. **Hai mẫu gạch ốp lát được sử dụng thí nghiệm**

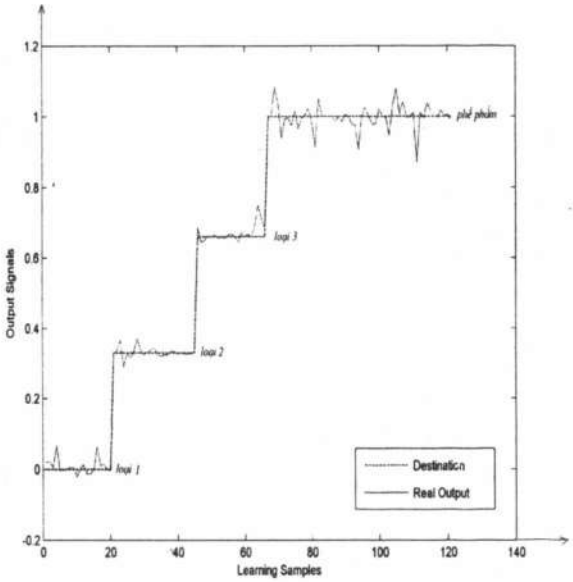
Đối với mỗi mẫu gạch, được phân thành 4 loại phụ thuộc vào chất lượng. Đó là gạch loại A1 (chất lượng cao nhất), A2, A3 và A4 (phế phẩm). Trong các tính toán mô phỏng, gạch loại A1 sẽ được mã hóa tương ứng với đáp ứng đầu ra là y = 0, gạch loại A2 sẽ tương ứng với đáp ứng đầu ra là $y = 0,33$, gạch loại A3 sẽ tương ứng với đáp ứng đầu ra là $y = 0,66$ và các viên gạch phế phẩm sẽ tương ứng với đáp ứng đầu ra $y = 1$.

Loại gạch (hoa văn 4.21a) có 225 mẫu gạch, được chia thành 2 tập hợp: Ảnh của 121 viên được sử dụng để xây dựng mô hình (tập số liệu học), ảnh của 104 viên còn lại được sử dụng để kiểm tra mô hình (tập số liệu kiểm tra).

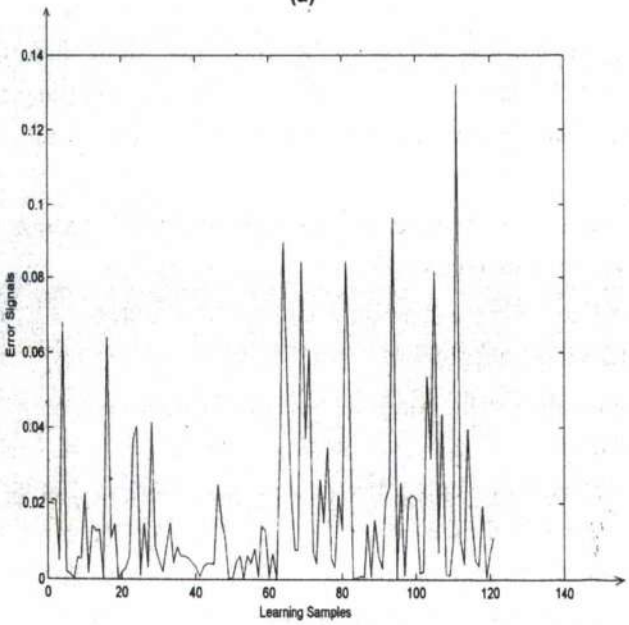
Tập số liệu học 121 mẫu bao gồm: 20 mẫu loại 1, 25 mẫu loại 2, 21 mẫu loại 3 và 55 mẫu loại 4.

Tập số liệu kiểm tra 104 mẫu bao gồm: 34 mẫu loại 1, 21 mẫu loại 2, 34 mẫu loại 3 và 24 mẫu loại 4.

Mô hình nhận dạng bằng mạng TSK đã được xây dựng trong môi trường Matlab được sử dụng với $M = 14$ quy tắc suy luận. Quá trình học được thực hiện với 100 lần lặp, kết quả của quá trình học được thể hiện trên hình 4.22. (Các giá trị đích có thể là 0; 0,33; 0,66 hoặc 1 ứng với 4 loại gạch). Từ hình 4.22 ta có đầu ra của mạng TSK đã bám sát với các giá trị đích, các sai số đều nhỏ hơn 0,16 là giá trị ngưỡng của sai số nhận dạng (do tín hiệu đầu ra của mạng ta sẽ làm tròn về giá trị đích gần nhất).



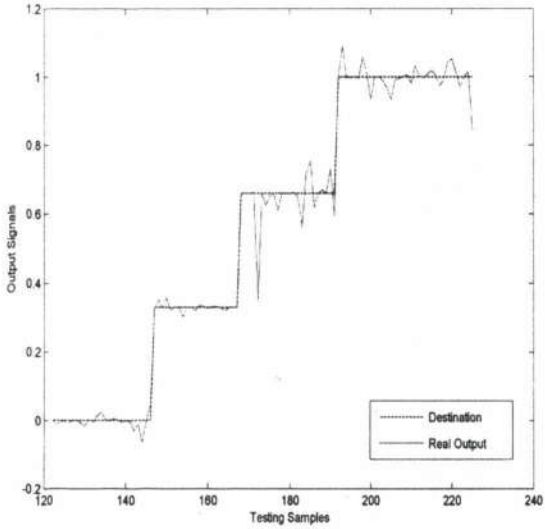
(a)



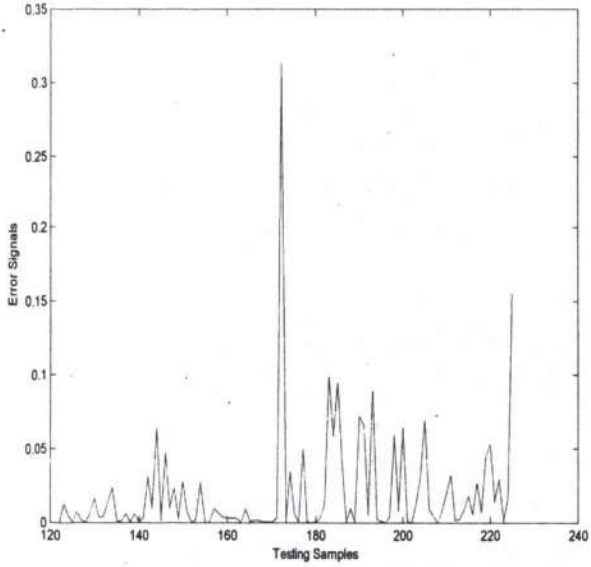
(b)

Hình 4.22. **Đồ thị đầu ra tín hiệu đích và đầu ra tín hiệu từ mạng TSK cho các số liệu "học" (a) và sai số giữa hai tín hiệu (b)**

Mạng sau khi được huấn luyện đã được kiểm tra với tập 104 số liệu khác với các mẫu đã học. Hình 4.23 mô tả đầu ra của mạng cùng với các giá trị đích và sai số giữa hai tín hiệu này.



(a)



(b)

Hình 4.23. **Đồ thị đầu ra tín hiệu đích và đầu ra tín hiệu từ mạng TSK cho các số liệu "kiểm tra" (a) và sai số giữa hai tín hiệu (b)**

Cũng tương tự như trong trường hợp kết quả học, khi đưa vào mạng TSK các số liệu từ các mẫu của tập kiểm tra, ta có thể dễ dàng nhận thấy đối với đa số trường hợp các tín hiệu đầu ra bám theo giá trị đích. Từ đồ thị sai số ở hình 4.23b, ta thấy chỉ có 2 trường hợp tín hiệu vượt ngưỡng 0,16, đó là trường hợp nhận dạng nhầm gạch A3 thành gạch A2 và nhận dạng nhầm gạch A4 thành A3. Với các kết quả như trên, cả trong trường hợp xây dựng cũng như kiểm tra, mô hình TSK đều cho kết quả nhận dạng chính xác cao. Cụ thể sai số học $E_{học} = \frac{0}{121} \times 100\% = 0\%$, sai số kiểm tra bằng $E_{kt} = \frac{2}{104} \cdot 100\% = 1,92\%$. Đây là một kết quả rất khả quan và hứa hẹn khả năng xây dựng được một mô hình thực tế với chất lượng cao.

Hoàn toàn tương tự với loại gạch thứ hai (hoa văn 4.21b). Ta có bộ số liệu tổng cộng 304 mẫu, được chia làm hai tập hợp. Tập dùng để xây dựng mô hình bao gồm 165 mẫu, tập dùng để kiểm tra mô hình gồm 139 mẫu. Các mẫu được chia thành hai tập hợp số liệu theo tỷ lệ tương đối là 55% mẫu để học, 45% mẫu để kiểm tra, cụ thể là:

Tập 165 mẫu học bao gồm: 33 mẫu loại A1, 50 mẫu loại A2, 39 mẫu loại A3 và 43 mẫu loại A4.

Tập 139 mẫu kiểm tra bao gồm: 27 mẫu loại A1, 42 loại mẫu A2, 33 mẫu loại A3 và 37 mẫu loại A4.

Sau khi xây dựng mô hình mạng TSK cho bộ số liệu học ta có kết quả học cho tập 165 mẫu có 2 mẫu bị nhận dạng sai, ứng với sai số học $E_{học} = \frac{2}{165} \times 100\% = 1,21\%$. Mạng TSK này được kiểm tra trên 139 mẫu còn lại ta có 5 mẫu bị nhận dạng sai, tương ứng với $E_{ktra} = \frac{5}{139} \times 100\% = 3,59\%$.

Tóm lại, đối với cả hai loại mẫu hoa văn, ta có sai số nhận dạng khá tốt, độ chính xác đạt 96,4%. Đồng thời, sau khi có được vectơ đặc tính cho các mẫu gạch, quá trình tính toán mô hình TSK đã được thực hiện gần như hoàn toàn tự động.

4.4. ỨNG DỤNG TRONG DỰ BÁO SỰ CỐ ĐỘNG CƠ

Trong nội dung này, chúng tôi đưa ra ví dụ minh họa sử dụng mô hình mạng nơron TSK dự báo sự cố động cơ điện một chiều.

4.4.1. Mô phỏng các chế độ làm việc của động cơ

Nhằm đơn giản hóa các khối đo lường, trong nội dung này đặt mục tiêu xây dựng mô hình phát hiện sự cố chỉ dựa trên các số liệu đo vận tốc góc tức thời của động cơ. Để có được các giá trị vận tốc góc tức thời này, một mô hình của động cơ một chiều kích từ độc lập sẽ được xây dựng sử dụng công cụ Simulink (Matlab) để mô phỏng. Đồng thời, trong nội dung này ta cũng tạm xét các trường hợp mà thời điểm xảy ra sự cố được xác định trước để mô hình chỉ tập trung vào nhiệm vụ phân loại sự cố.

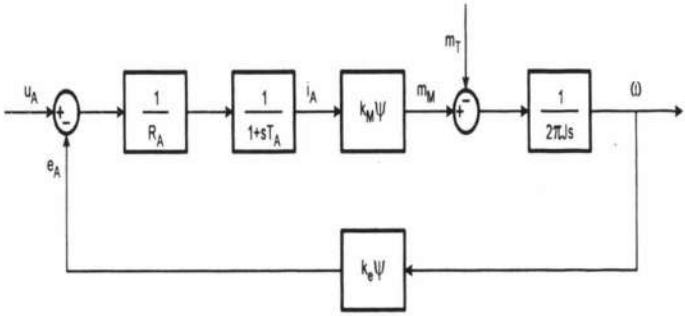
4.4.1.1. Mô tả toán học động cơ điện một chiều kích từ độc lập

Từ các phương trình cơ bản của động cơ điện một chiều kích từ độc lập biến đổi Laplace ta có hệ phương trình sau:

$$\begin{aligned} u_A(s) &= e_A(s) + R_A \cdot i_A(s) + L_A \cdot s \cdot i_A(s) \\ e_A(s) &= k_e \cdot \psi \cdot \omega(s) \\ m_M(s) &= k_M \cdot \psi \cdot i_A(s) \\ m_M(s) - m_T(s) &= 2\pi J \cdot s \cdot \omega(s) \end{aligned} \tag{4.7}$$

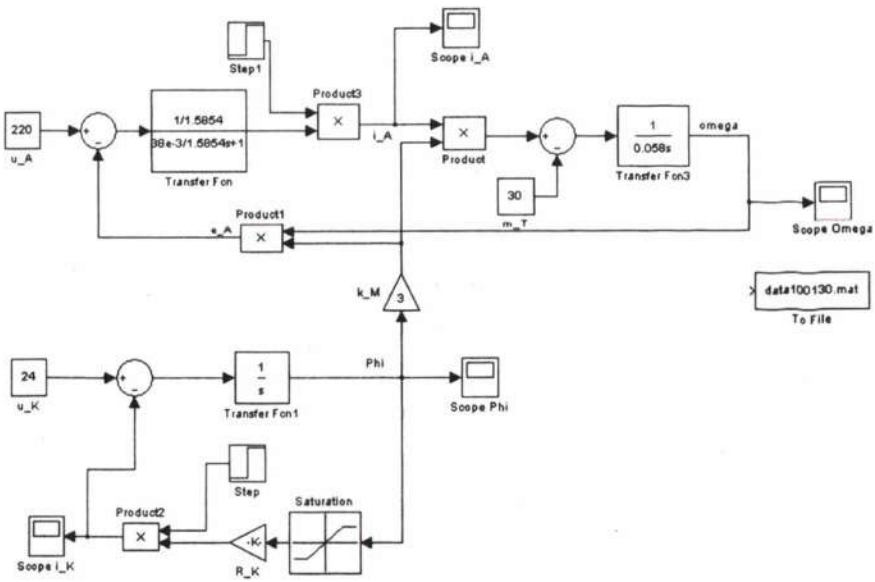
Từ hệ phương trình (4.7) ta xây dựng được sơ đồ khối động cơ điện một chiều kích từ độc lập khi từ thông kích từ không đổi như hình 4.24. Để mô phỏng các sự cố xảy ra khi động cơ đang làm việc ta sử dụng số liệu động cơ một chiều kích từ độc lập nghiên cứu có các thông số sau:

Điện áp phần ứng: $u_A = 220V$; điện áp kích từ: $u_K = 24V$; điện cảm phần ứng: $L_A = 0,038H$; điện trở phần ứng: $R_A = 1,5854\Omega$; mô-men quán tính của rôto: $J = 0,058kg/m^2$; hằng số mô-men: $k_M = 3$. Trước thời điểm xảy ra sự cố coi như động cơ đang ở trạng thái xác lập (trong quá trình mô phỏng, xuất phát từ các sơ kiện bằng 0, ta quan sát thấy ở thời điểm 2s các mô hình động cơ đã đạt trạng thái xác lập nên các mô phỏng sẽ lấy thời điểm 2s là thời điểm xảy ra sự cố).



Hình 4.24. **Sơ đồ khối động cơ điện một chiều kích từ độc lập khi từ thông kích từ không đổi**

Để có được giá trị tức thời của vận tốc góc, sử dụng mô hình mô phỏng của động cơ điện một chiều kích từ độc lập trên SIMULINK như hình 4.25, trong đó sự cố sẽ được mô phỏng bằng cách điều chỉnh giá trị dòng i_A và i_K nhân với các hệ số tương ứng (tại thời điểm xảy ra quá độ), giá trị tải của động cơ được thay đổi ở tham số m_T.

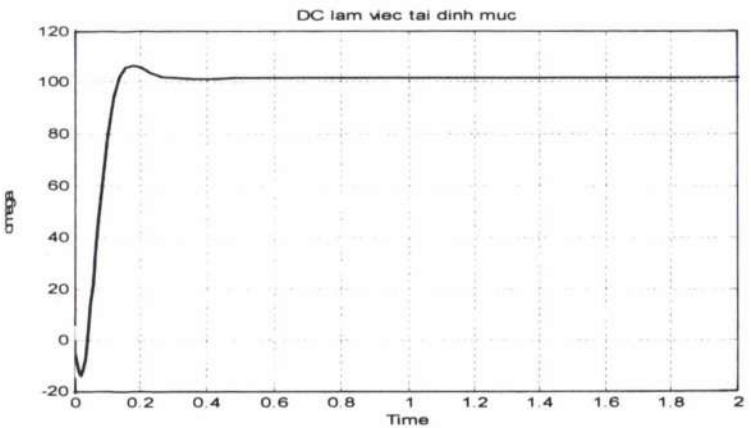


Hình 4.25. **Mô hình mô phỏng động cơ điện một chiều kích từ độc lập sử dụng SIMULINK**

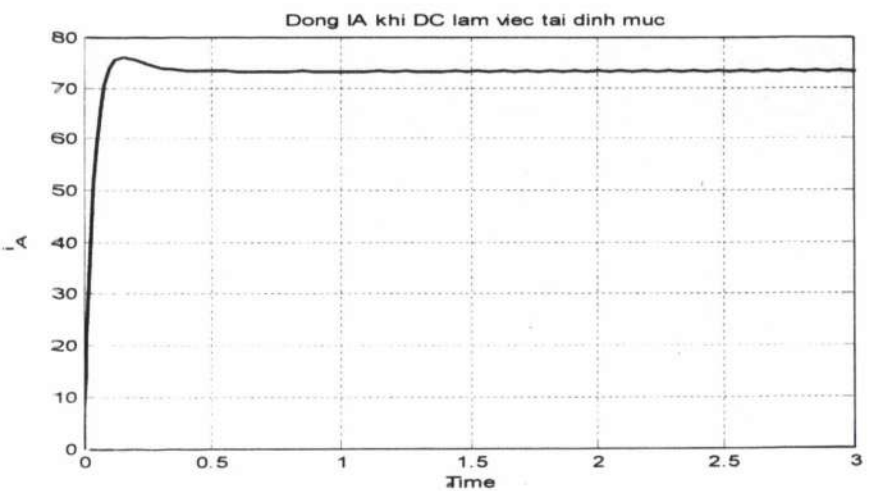
4.4.1.2. Mô phỏng các chế độ làm việc

a. Động cơ khi làm việc bình thường:

Từ hình 4.26 ta thấy động cơ điện một chiều kích từ độc lập trên khi làm việc bình thường, tải định mức có đáp ứng đầu ra là tốc độ tốt. Hệ dao động ít, thời gian quá độ ngắn $t_{qd} = 0,3$ s, độ quá điều chỉnh nhỏ.



(a)



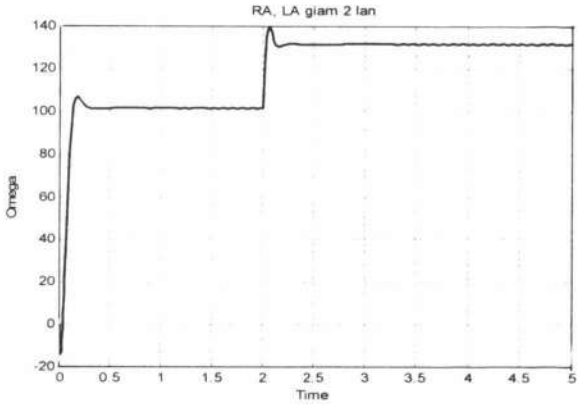
(b)

Hình 4.26. **Đồ thị đáp ứng tốc độ (a), dòng điện phần ứng (b) động cơ làm việc bình thường khi tải định mức**

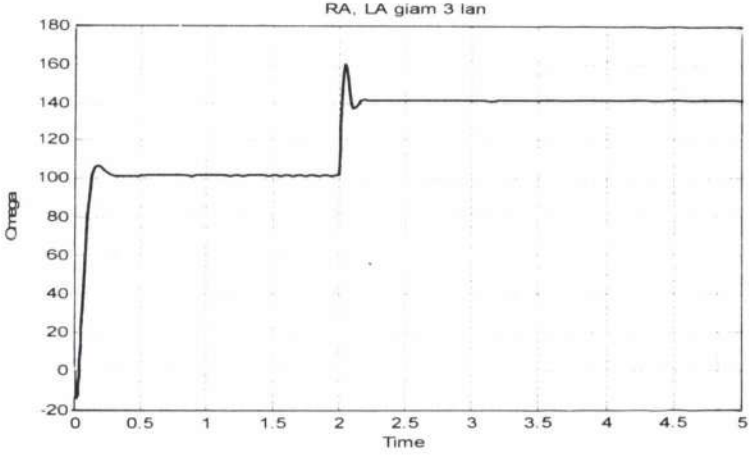
b. Động cơ khi xảy ra sự cố:

– Sự cố chạm tắt các vòng dây phần ứng:

Động cơ đang làm việc bình thường, tải định mức, sau 2s xảy ra sự cố R_A, L_A giảm, đáp ứng đầu ra tốc độ có dạng như trên hình 4.27. Nhìn vào đồ thị ta thấy, khi R_A, L_A giảm càng nhiều thì tốc độ động cơ càng tăng cao, số lần dao động ít, thời gian quá độ khi xảy ra sự cố $t_{qđ} = 0,15$ s.



(a)

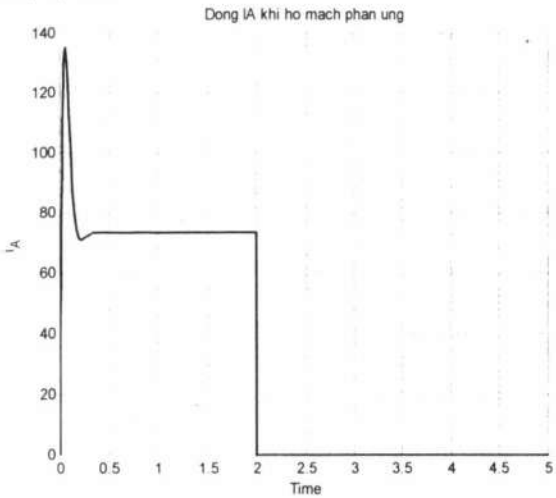


(b)

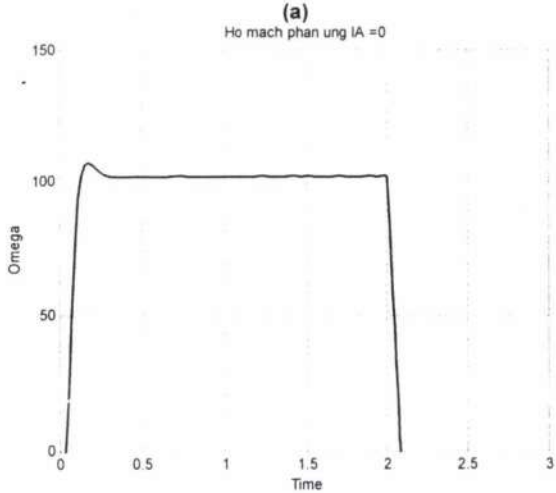
Hình 4.27. **Đồ thị đáp ứng tốc độ khi xảy ra sự cố chạm tắt vòng dây phần ứng R_A, L_A giảm 2 lần (a), giảm 3 lần (b)**

– Hở mạch phần ứng:

Khi động cơ đang làm việc bình thường tải định mức, giả sử sau 2s xảy ra sự cố hở mạch phần ứng, dòng điện phần ứng $i_A = 0$, đáp ứng ra tốc độ có dạng như hình 4.28. Nhìn vào đồ thị hình 4.28 ta thấy, khi xảy ra sự cố hở mạch phần ứng dòng điện phần ứng $i_A = 0$ ngay, tốc độ giảm về 0 sau thời gian quá độ 0,15s.



(a)

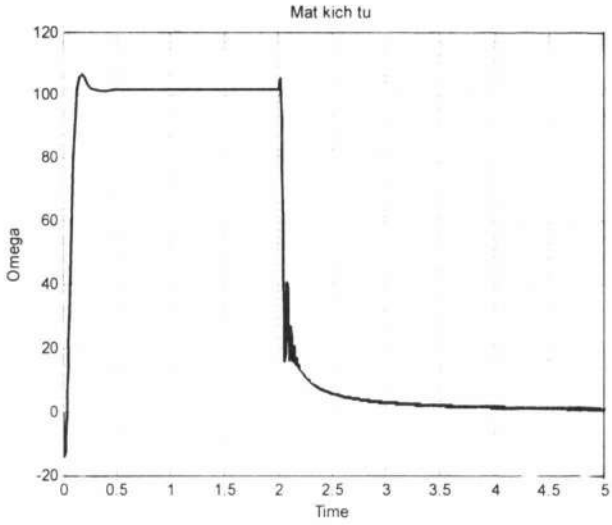


(b)

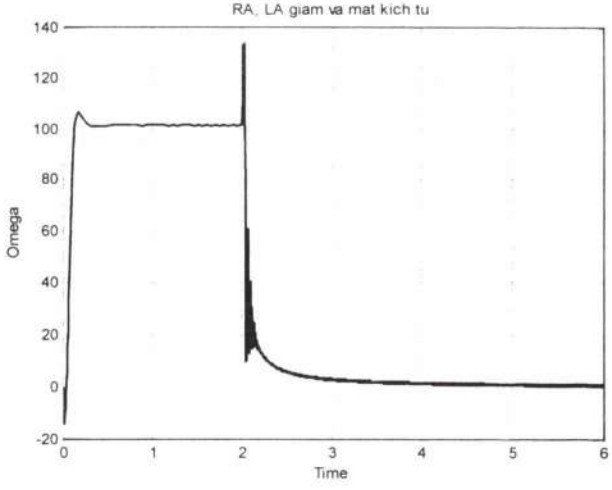
Hình 4.28. **Đồ thị đáp ứng dòng điện (a), tốc độ (b) khi xảy ra sự cố hở mạch phần ứng**

– Sự cố chạm tắt các vòng dây phần ứng và mất kích từ:

Tương tự ta có các đồ thị kết quả như trên hình 4.29.



(a)



(b)

Hình 4.29. **Đồ thị đáp ứng tốc độ khi xảy ra sự cố mất kích từ (a), khi xảy ra sự cố vừa chạm tắt dây quấn phần ứng vừa mất kích từ (b)**

Từ các đồ thị mô phỏng động cơ điện một chiều kích từ độc lập khi làm việc bình thường, khi đang làm việc bình thường sau 2s xảy ra các sự cố ta nhận thấy:

+ Với mỗi loại sự cố có độ quá điều chỉnh ở đồ thị đáp ứng tốc độ khác nhau;

+ Khi xảy ra sự cố, thời gian quá độ đều nằm trong khoảng 2 đến 2,15s.

Ý tưởng của giải pháp nhận dạng sẽ là tự động phân tích số liệu để có được mô hình mô tả sự khác nhau giữa các đường quá độ, từ trên cơ sở đó ta có thể suy ngược lại được sự cố đã gây ra quá trình quá độ đang xét.

4.4.2. Phân tích số liệu để xây dựng mô hình nhận dạng sự cố

Để có được bộ số liệu mẫu, mô hình mô phỏng trên hình 4.25 đã được sử dụng với nhiều thông số khác nhau đại diện cho những trường hợp làm việc và sự cố khác nhau của động cơ. Cụ thể:

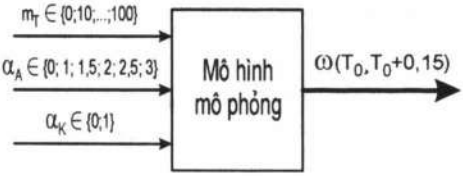
– Khi động cơ đang làm việc bình thường nhưng tải thay đổi thì cũng dẫn tới tốc độ động cơ biến thiên. Mạng TSK phải phân biệt được trường hợp động cơ có sự cố và trường hợp biến thiên của tải. Vì vậy ta xây dựng bộ số liệu với thông số thứ nhất biến thiên là: m_T (tải). Ta chọn tải có khối lượng dao động trong khoảng từ 0 đến 100kg với bước thay đổi 10kg. Vậy ta có bộ số liệu theo tải là 11 trường hợp [0; 10; 20; 30; 40; 50; 60; 70; 80; 90; 100].

– Khi hở mạch dây quấn phần ứng ta có dòng điện phần ứng $i_A = 0$, khi chạm tắt dây quấn phần ứng ta có dòng điện phần ứng tăng lên có thể 1,5; 2; 2,5; 3 lần so với định mức. Vậy ta có bộ số liệu theo sự thay đổi của hệ số dòng phần ứng là α_A có 6 trường hợp [0; 1; 1,5; 2; 2,5; 3].

– Khi động cơ mất kích từ thì hệ số của dòng $\alpha_K = 0$. Khi động cơ làm việc bình thường ta xét trường hợp dòng kích từ không đổi hay hệ số dòng $\alpha_K = 1$. Vậy ta có thông số thứ hai của bộ số liệu là α_K với hai giá trị 0 hoặc 1.

Tổng hợp ba trường hợp biến thiên của các số liệu trên, ta xây dựng được bộ số liệu tổng quát gồm $11 \cdot 6 \cdot 2 = 132$ mẫu. Tuần tự từng bộ 3 giá trị của các tham số m_T, α_A, α_K được đưa vào mô hình mô phỏng trên hình 4.25

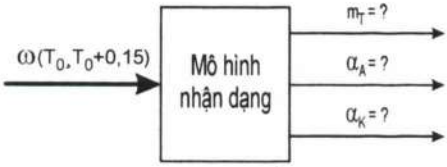
để tính toán đưa ra được vận tốc góc đáp ứng của động cơ trong trường hợp đó như trên hình 4.30.



Hình 4.30. **Mô hình tạo bộ số liệu**

Theo như các khảo sát đã trình bày trên từ hình 4.27 đến 4.29 ta thấy quá trình quá độ cho mẫu động cơ đang khảo sát diễn ra trong khoảng 150ms sau thời điểm quá độ (xét đồng nhất là tại $T_0 = 2s$), do đó trong khi xây dựng vectơ đặc tính ta sẽ chỉ cần lấy các giá trị tức thời trong khoảng $\omega_{T0} \div \omega_{T0+0,15}$.

Mục đích của mô hình nhận dạng là xây dựng được mô hình ngược, từ các giá trị tức thời của vận tốc góc cần xác định ngược lại 3 giá trị trạng thái của động cơ là m_T, α_A, α_K. Quá trình này được mô tả như trên hình 4.31.



Hình 4.31. **Mô hình tổng quát nhận dạng sự cố**

Để giảm bớt kích thước của mô hình nhận dạng, ta đã khảo sát và lựa chọn bước rời rạc hóa của giá trị tức thời của vận tốc góc là 0,01s. Bước rời rạc hóa này cho phép thu được các kết quả có độ chính xác chấp nhận được như trình bày dưới đây. Với bước rời rạc này ta có 16 giá trị tức thời của tốc độ góc $\omega_{T_0}, \omega_{T_0+0,01}, \ldots, \omega_{T_0+0,15}$ để lập thành vectơ đặc tính $\mathbf{x}$ của một mẫu gồm 16 thành phần: $\mathbf{x} = [x_1, \ldots, x_{16}]$. Như vậy 1 mẫu số liệu bao gồm cặp 2 véc tơ $\{\mathbf{x}, \mathbf{d}\}$ trong đó $\mathbf{x} \in \mathbb{R}^{16}$ - vectơ đầu vào chứa 16 giá trị tức thời của vận tốc góc của động cơ, $\mathbf{d} \in \mathbb{R}^3$ - vectơ đầu ra tương ứng chứa 3 giá trị tham số m_T, α_A, α_K.

4.4.3. Ứng dụng mạng TSK nhận dạng sự cố trong động cơ điện một chiều

Từ bộ số liệu 132 mẫu ta tách thành 2 tập số liệu: 75% số mẫu (tương ứng với 99 mẫu) được dùng để xây dựng mô hình nhận dạng (bộ số liệu học), 25% (tương ứng với 33 mẫu) được dùng để kiểm tra mô hình nhận dạng (bộ số liệu kiểm tra). Tập mẫu kiểm tra ta chọn các mẫu có chỉ số 2, 6, 10, ..., 132 (cách 4); tập mẫu học 99 mẫu là các mẫu còn lại.

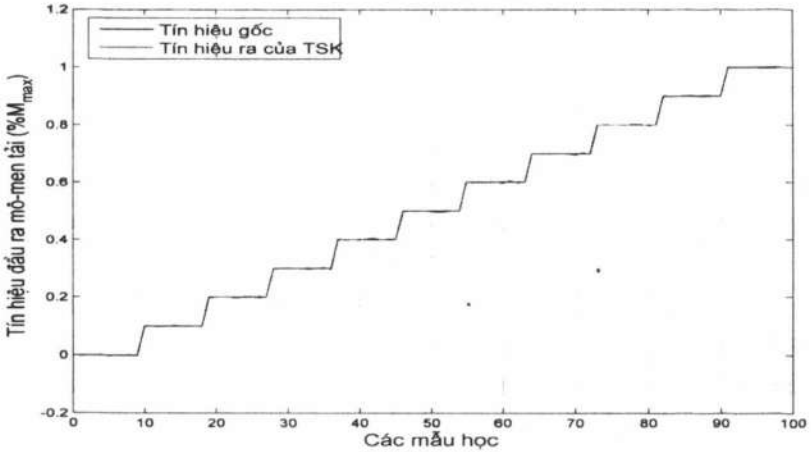
Với các thử nghiệm khác nhau, ta đã chọn số luật suy luận của mạng TSK thích hợp cho tập số liệu đã tạo, từ đó xây dựng được sơ đồ cấu trúc mạng TSK nhận dạng sự cố động cơ theo hình 1 với các đáp ứng đầu ra

$$\mathbf{y}(\mathbf{x}) = \mathrm{TSK}(\mathbf{x}) \tag{4.8}$$

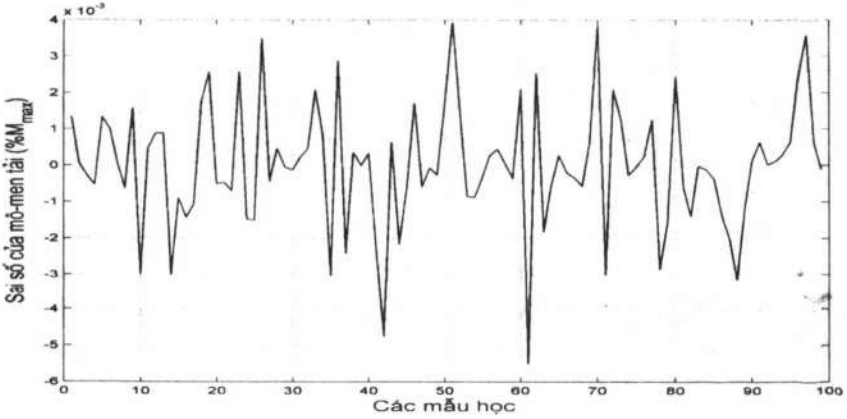
Vậy bài toán đặt ra là đi xác định tám trọng tâm $\mathbf{c}_1,\ldots,\mathbf{c}_8$, các độ mở $\boldsymbol{\sigma}_1,\ldots,\boldsymbol{\sigma}_8$, các hệ số hình dạng $\mathbf{b}_1,\ldots,\mathbf{b}_8$, và các hàm TSK $f_1(\mathbf{x}),\ldots,f_8(\mathbf{x})$ sao cho ta có $\forall \mathbf{x} : \mathbf{y} = \mathrm{TSK}(\mathbf{x}) \approx \mathbf{d}$ (đáp ứng đầu ra của mạng xấp xỉ với giá trị đích cần có) với tất cả các mẫu trong bộ số liệu học (99 mẫu) và bộ số liệu kiểm tra (33 mẫu). Việc xác định các thông số của mạng này sẽ được thực hiện bằng thuật toán bước giảm cực đại để tối ưu hóa sai số giữa đầu ra của mạng và đầu ra đích cần có.

Đối với bộ số liệu đã có, quá trình học theo thuật toán bước giảm cực đại được thực hiện với 50 bước lặp. Các kết quả học (với bộ 99 mẫu) của mạng TSK đối với kênh 1 (nhận dạng giá trị m_T) được thể hiện trên hình 4.32. Các giá trị đích là 0; 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1 tương ứng với 11 trường hợp tải dao động.

Từ hình 4.32a ta có đầu ra của mạng TSK đã bám sát với các giá trị đích, các sai số đều nhỏ hơn giá trị ngưỡng của sai số nhận là 0,1 (sai số lớn nhất trên hình 4.32b không vượt quá 6.10^{-3}). Mạng sau khi được huấn luyện đã được kiểm tra với tập 33 số liệu khác với các mẫu đã học.

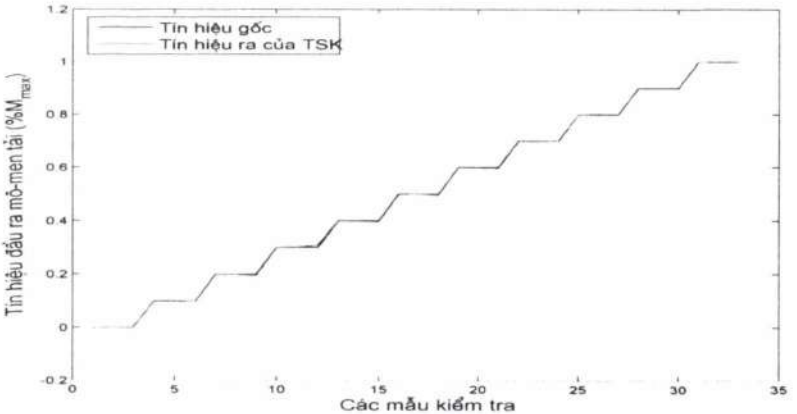


(a)

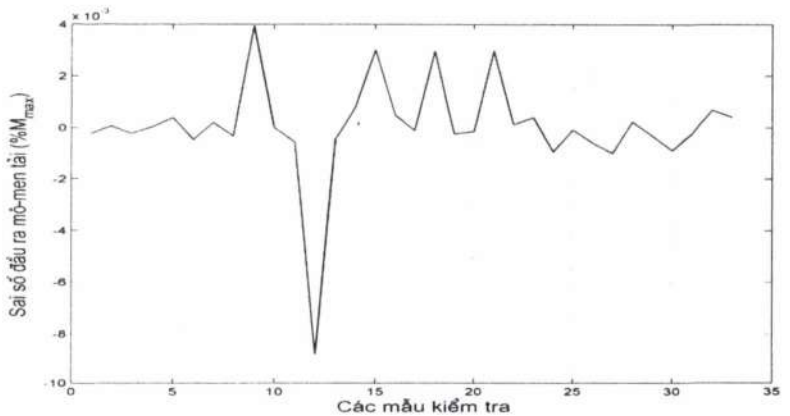


(b)

Hình 4.32. **Kết quả học trên 99 mẫu cho nhận dạng trạng thái của mômen tải (biến thiên từ 0 đến 100% mômen max)(a), và sai số học giữa kết quả đầu ra TSK và giá trị đích (cho mômen tải) (b)**



(a)

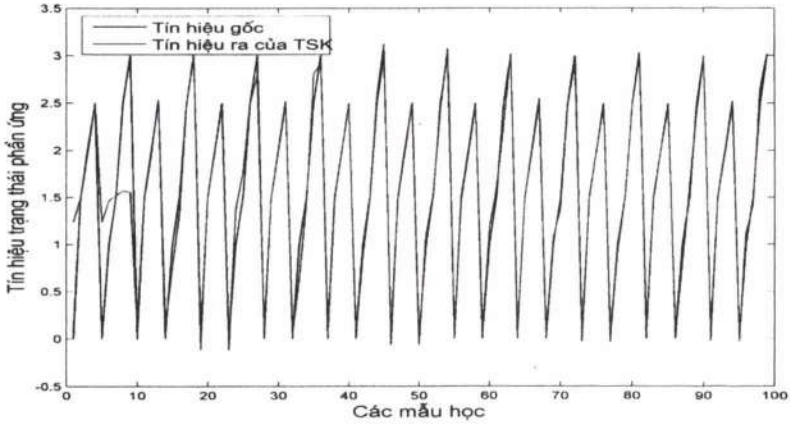


(b)

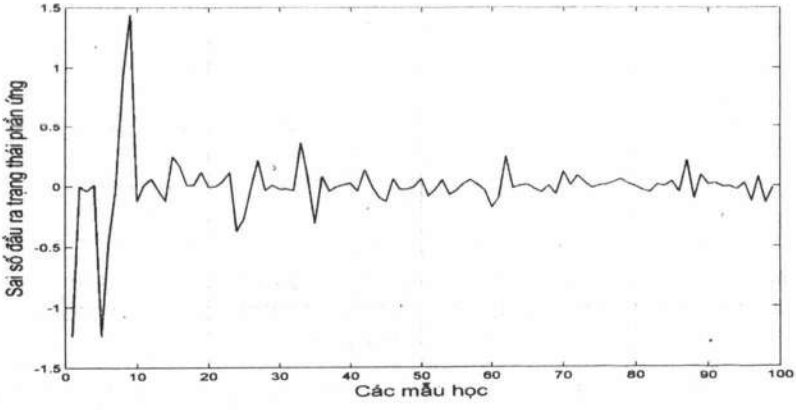
Hình 4.33. **Kết quả kiểm tra trên 33 mẫu nhận dạng giá trị mômen tải biến thiên từ 0 đến 100% mômen max(a) và Sai số học giữa kết quả đầu ra TSK và giá trị đích cho mômen tải (b)**

Hình 4.33 mô tả đầu ra m_T (đầu ra thứ nhất) của mạng cùng với các giá trị đích và sai số giữa hai tín hiệu này. Cũng tương tự như trong trường hợp kết quả học, khi đưa vào mạng TSK các số liệu từ các mẫu của tập kiểm tra, ta có thể dễ dàng nhận thấy các tín hiệu đầu ra bám sát theo giá trị đích. Từ đồ thị sai số ở hình 4.33b, ta thấy không có trường hợp tín hiệu vượt ngưỡng 0,1 (sai số lớn nhất không vượt quá 10^{-2}).

Với các kết quả như trên, cả trong trường hợp xây dựng cũng như kiểm tra, mô hình TSK đều cho kết quả nhận dạng chính xác cao. Cụ thể sai số học $E_{học}$ = 0% sai số kiểm tra bằng E_{kt} = 0%. Hoàn toàn tương tự, ta xét đầu ra thứ hai của mạng TSK (nhận dạng giá trị α_A). Các kết quả học và kiểm tra cho kênh này được thể hiện trên các hình 4.34 và hình 4.35.

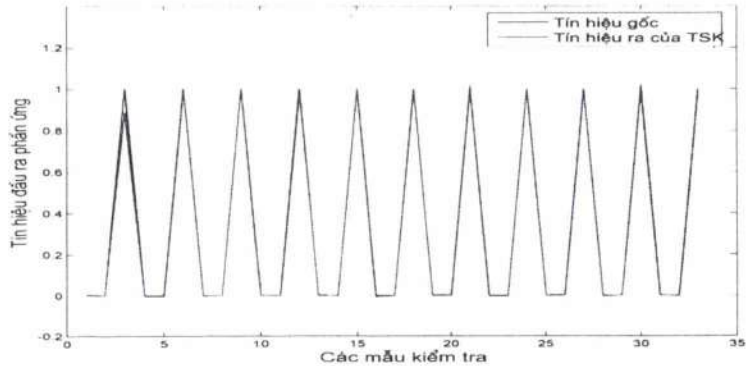


(a)

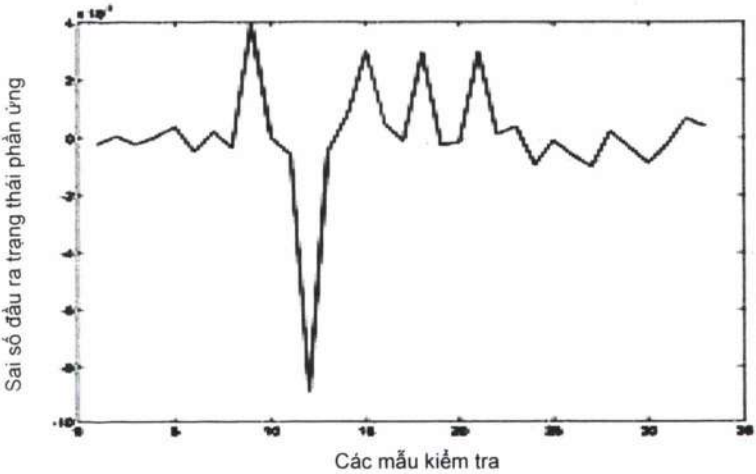


(b)

Hình 4.34. **Kết quả học trên 99 mẫu cho nhận dạng trạng thái của dòng phản ứng (a) và sai số học trên 99 mẫu cho nhận dạng trạng thái của dòng phản ứng (b)**



(a)

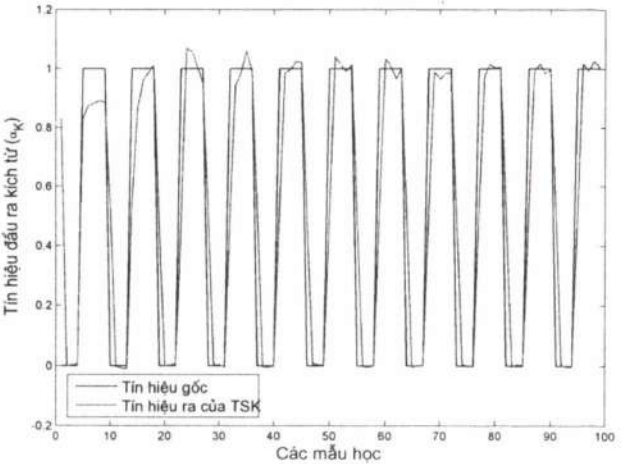


(b)

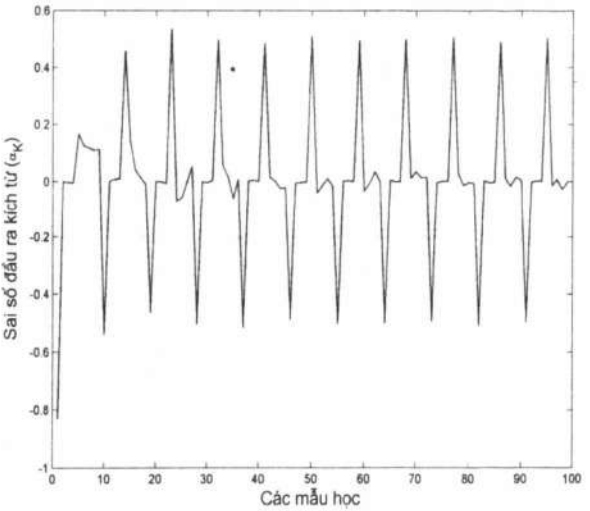
Hình 4.35. **Kết quả kiểm tra trên 33 mẫu cho nhận dạng trạng thái của dòng phản ứng (a), sai số trên 33 mẫu cho nhận dạng trạng thái của dòng phản ứng (b)**

Có thể nhận thấy trên hình 4.34b là có 4 trường hợp sai số vượt ngưỡng 0,25, ứng với sai số học $E_{học} = \frac{4}{99} \cdot 100\% = 4,04\%$, còn trên hình 4.35b sai số kiểm tra vượt ngưỡng bằng 0 ứng với $E_{kt} = 0\%$.

Tương tự đầu ra thứ ba của mạng TSK (nhận dạng giá trị α_K) có các kết quả học và kiểm tra được thể hiện tương ứng trên hình 4.36 và hình 4.37.

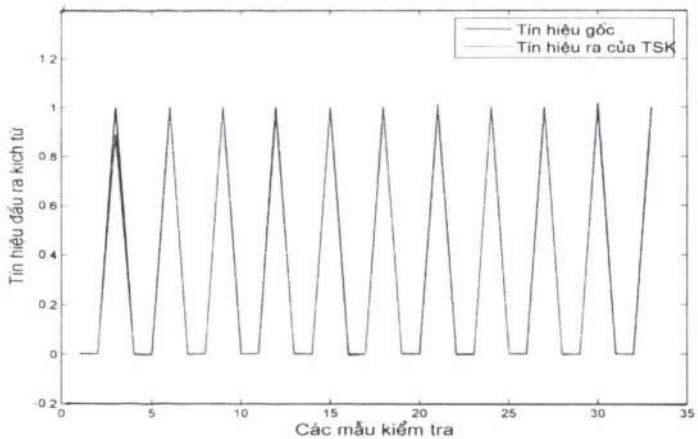


(a)

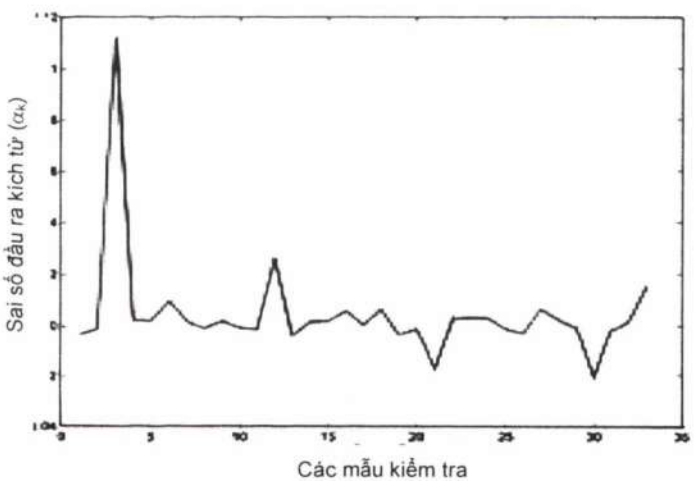


(b)

Hình 4.36. **Kết quả học trên 99 mẫu cho nhận dạng trạng thái của dòng kích từ (a), sai số học giữa đầu ra của mạng và giá trị đích cho nhận dạng trạng thái của dòng kích từ (b)**



(a)



(b)

Hình 4.37. **Kết quả kiểm tra trên 33 mẫu cho nhận dạng trạng thái của dòng kích từ (a), sai số kiểm tra trên 33 mẫu cho nhận dạng trạng thái của dòng kích từ (b)**

Trên hình 4.36b ta có thể quan sát được 5 trường hợp sai số vượt ngưỡng 0,5, ứng với sai số học $E_{học} = \frac{5}{99} \cdot 100\% = 5\%$; còn trên hình 4.37b, tất cả các mẫu kiểm tra đều có sai số nhỏ hơn 0,5, ứng với sai số kiểm tra bằng $_{kt} = 0\%$.

Mô hình mạng nơron lôgic mờ TSK có khả năng ứng dụng cho nhiều bài toán nhận dạng và dự báo. Thông qua kết quả khảo sát này cho thấy đối với các loại sự cố thay đổi của tải động cơ điện một chiều kích từ độc lập ta có độ chính xác cao gần 100%. Sau khi có được vectơ đặc tính cho các tốc độ động cơ thì quá trình tính toán các thông số mô hình mạng TSK thực hiện gần như hoàn toàn tự động.

TÀI LIỆU THAM KHẢO

Bùi Công Cường, Nguyễn Doãn Phước (2001), *Hệ mờ mạng Nơron và ứng dụng*, NXB Khoa học và Kỹ thuật.

Phan Xuân Minh, Nguyễn Doãn Phước (2004), *Lý thuyết điều khiển mờ*, NXB Khoa học và Kỹ thuật.

Nguyễn Phùng Quang (2005), *Matlab & Simalink*, NXB Khoa học và Kỹ thuật.

Nguyễn Trọng Thuần (2000), *Điều khiển Lôgic và ứng dụng*, NXB Khoa học và Kỹ thuật.

Nguyễn Hữu Tình, Lê Tấn Hưng, Phạm Thị Ngọc Yến, Nguyễn Thị Lan Hương (1999), *Cơ sở kỹ thuật Matlab và ứng dụng*, NXB Khoa học và Kỹ thuật.

Trần Hoài Linh, Đinh Văn Nhượng, Đặng Văn Tuệ, *Ứng dụng mạng nơron lôgic mờ trong phát hiện sự cố của động cơ điện một chiều*, Tạp chí tin học và điều khiển học, T27,S4 (2011).

http://www.vnchannet.net/news/kinh-te/200709/thi-truong-gom-su

http://www.taichinhVietnam.com/taichinhvietnam/modules.php

http://www.sieuthixaydung.com.vn/modules.php?name=News&opcase=detailsnews&mid=751&mcid=204&pid=&menuid=

http://baigiang.violet.vn/present/show?entry_id=218839

http://elib.deeforum.net/ebooks/GiaoTrinh/DKTD/DKHD/ Chapter4.pdf

Tổng quan xử lý ảnh http://vocw.edu.vn/content/m 11079/latest

A.Cichocki., R,Unbehauen. (1993), "*Neural Networks for Optimization and Signal Processing*", Wiley, N.Y.

Alsina, C, Trillas, E, Valverde, L. (1983), *On some Logical Connectives for Fuzzy Set Theory*, J. Math. Appl.

15. Araki, S., Nomura, H., Wakami, N. (1993), *Segmentation of Thermal Images using Fuzzy Cmeans Algoritthm*, in Proc. Second IEEE International Conference on Fuzzy Systems, pp. 719-724.

16. Arbib, Michaela (Ed). (1995), *The Hanbook of Brain Theory and Neural Networks*.

17. B. Kosko.(1992), *Neural Networks and Fuzzy Systems*, Prentice - Hall, NJ.

18. Backer, E. (1978), *Cluster Analysis by Optimal Decomposition of Induced Fuzzy Sets*, Delft University Press, Delft, The Netherlands.

19. Bellman. R, Giertz. M. (1973), *On the Analytic Formalism of the Theory of Fuzzy Set*, Inf. Sciences.

20. Bensaid, A. (1994), *Improved Fuzzy Clustering for Pattern Recognition with Image Segmentation*, PhD Dissertation, Univ. of South Florida, Tampa.

21. Bertsekas, Dimitrip. (1999), *Nonlinear Progamming*.

22. Bezdek.J.(1974), *Cluster Vvalidity with Fuzzy Sets*, J. Cybernetics, 3, pp. 58-73.

23. Bezdek.J. (1974), *Numerical Taxonomy with Fuzzy Sets*, J. Mathematical Biology, 1, pp. 57- 71.

24. Bezdek,J., (1975), *Mathematical models for systematics and taxonomy*, in Proc. 8th Int. Conf. on Numerical Taxonomy, G. Estabrook (Ed.), Freeman, San Francisco, pp. 143-166.

25. Bezdek. J., Windham. M., Ehrlich. R., (1980), *Statistical prameters of cluster validity functionals*, Internat. J. Comput. and Inform. Sci., 9, pp. 385- 394.

26. Bezdek. J., Windham. M., Ehrlich. R., (1980), *Statistical parameters of fuzzy cluster validity functionals*, Int. J. Computer and Information Sciences, 9, pp. 323-336.

27. Bezdek. J. (1981), *Pattern Recognition with Fuzzy Objective function Algorithms*, Plenum Press, New York.

28. Bezdek, j.C., (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York.

29. Bezdek, j.C., (1993), *A review of probabilistic, fuzzy and neural models for pattern recognition*, J. of Intelligent and Fuzzy Systems, pp. 1-25.

30. Bock, H., (1979), *Simultaneous clustering of objects and variables*, Cours de Communautes Europeenes, Analyse de donnees et Informatique, INRIA, Fontainbleau.

31. Butnariu, D. (1983), *Additive fuzzy measure and integrals*, Math.J.Anal.Appl, pp. 436-452.

32. Butnariu. D. Klement, E.P. (1993), *Triangular Norm Based Measures and Games with Fuzzy Coalitions*, Kluwer, Dordrecht.

33. C.T. Lin and C.S. G. Lee. (1996) *Neural Fuzzy Systems*, prentice Hall, Lon don.

34. Cheo Duo, Li Xue, Cui Du-Wu.(2007), "*An adaptive cluster validity index for the fuzzy C-mean*", IJCSNS,vol. 7, No. 2.

35. D.Gustafson., W, Kessel., *Fuzzy clustering with a fuzzy convariance matrix*, Proe. IEEE CDC, San Diego, pp. 761-766.

36. D.P.Filev and R.R.Yager. (1998), *On the isue of obtaining OWA weights*, Fuzzy sets and Systems.

37. Driankov, D. (1993), *An Introduction to Fuzzy Control*, SpringerVerlag.

38. Dubois, D. and Prade, H.(1980), *Fuzzy sest and System*: Theory and Applicantion: Academic press, NewYork.

39. Dumitrescu, D, (1983), *On fuzzy partitions, in ltinerant Seminar on functional Equation*, Approxinmation and Convexity, Cluj-Napoca, pp. 57-60.

40. Dumitrescu, D., (1983), *A method for cluster analysis*, Proc. Nat. Symp. Methods, Models and Techniques and Physics and Related Fields, Cluj- Napoca, pp. 57- 60.

41. Dumitrescu, D., (1983), *Hierarchical pattern classification*, in Colloq. Informatics, Iassy, pp. 349- 355.

42. Dumitrescu, D., (1984), *Fuzzy hierarchy in pattern recognition*, in Colloq. on Approximation and Optimization, Cluj- Napoca, pp. 75- 82.

43. Dumitrescu, D, Barbu, M., (1985), *Fuzzy entropy and processes*, in Itinerant Seminar on Functionnal Equations, Approximation and Convexity, Cluj-Napoka, pp. 71-74.

44. Dumitrescu, D., Kekedy, L., (1987), *Classifitation of mineral waters by pattern processing of chemical composition data*, Studia, Univ. Babes-bolyai Ser. Chemia, 32(2), pp. 68- 73.

45. Dumitrescu, D, (1988), *Hierarchical pattern classification*, Fuzzy Sets and Systems.

46. Dumitrescu, D., (1988), Hierarchical pattern classification, Fuzzy sets and Systems, 28, pp. 145- 162.

47. Dumitrescu, D., Lazarovici, G., (1990), *Fuzzy divisive clustering in archaeclogy, in Archeometry* in Romania, P. Frangopol, V. Moraru(Eds.), IAP Press, Bucharest, pp. 87- 92.

48. Dumitrescu, D., (1991), *Classification Theory*, University of Cluj-Napoca Press.

49. Dumitrescu, D., (1997), *Fuzzy hierarchical classification methods in analytical chemistry,* Fuzzy Logic in Chemistry, D.H.Rouvray(Ed.), Academic Press, New York, pp. 321-356.

50. E.Kamen., B.Heck. (1977), *"Fundamentals of singnals and systems using Matlab"*, Prentice Hall, N.Y

51. Follinger, O., (1992), *Regelungstechnik.* Huthig Buch Verlag Heidelberg.

52. Fossard, A. and Gueguen, C. (1997) *Multivariable System control.* North - Holand publishing company.

53. Frigui, H., Krishnapuram, R., (1997), *Clustering by competitive agglomeration*, Pattern Recognition, 30, pp. 1223- 1232.

54. Frigui, H., Krishnapuram, R., (1999), *A robust competitive clustering algorithm with applications in computer vision*, IEEE Trans. on Pattern analysis and Machine Intelligence, 21, pp. 450- 465.

55. Fung, L.W, Fu, K.S, (1975), *An axionmatic approach to rationnal decision making in a fuzzy environment*, in Fuzzy Sets and Their Applications to Cognitive and Decision Process,L.A. Zadeh, K.S. Fu.K.Tanaka, M.Shimura (Eds), Academic Press, New York.

56. G.Golub., C. Van Loan. (1991), *"Matrix computations"*, Academie Press, NY.

57. H. Demuth, M. Beale. (1992), *Neural Networks Toolbox for use with Matlab*, the Math work, Inc, , Natick.

58. Hellen doorn, (1994). *H. fuzzy control*. An overview Vieweg Verlag.

59. Hellendoorn, H. (1994), *Fuzzy Control:* An Overview. Vieweg Verlag.

60. Images Processing Toolbox for Use with Matlab, www.Mathworks.com.

61. J.S, Jang., C.T,Sun., E,Mizutani. (1997), *Neuro-fuzzy and soft comput-ing*, Prentice Hall, N.J.

62. K.Brudzewski., S,Oswski. (1999), "*Gas analysis system composed of a solid state sensor array and hybrid neural network stucture*". Sensors and Actuators-B55, pp. 38-46.

63. K.Godfrey, R.jones. (1986), *Singnal processing for control*, Springer-Verlag Press.

64. K.J.Hunt, D.Sbarbaro, R.Zbikowski and P.J.Gawthrop. (1992), *Neural Networks for Control Systems,* A survey, Automatica, V28, N6, pp. 1083-1112.

65. K.J.Hunt, G.lrwin. (1996), *Neural networks Engineering in dynamic Control Systems*, Springer - Verlag London Limited.

66. K.S.Narendra and K.Parthasarathy. (1990), *Identification and Control of dynamica systems using Neural Networks*.

67. K.S. Narendra and S. Mukhopadhyay. (1992), *Intelligent Control Vsing Neural networks*, IEEE Control System.

68. K.Warwick. (1995), *An overview of neural networks in control applications,* In "Neural networks in Robotic control theory and applications", Ellis Horwood.

69. Kahier, J. (1995), *Fuzzy control fur Ingenieure Vieweg Verlag Wiesbaden.*

70. Klir, G.J., (1987), Where do we stand on measures of uncertainty, ambiguity, fuzziness, and the like? Fuzzy Sets and Systems, pp. 141-160.

71. Klir, G.J and Yuan. B. (1995), *Fuzzy sest and fuzzy logic theory and Applicantions Prentice Hall.*

72. Kosko, B. (1991): Neural Networks and Fuzzy Control, Prentice Hall.

73. Kouvatsos, D. D., (1984), *A new decomposition algorithm for computer systems design*, in Methods of Operations Research, vol. 52, G. Hammer, D. Pal- laschke (Eds.), Verlagsgruppe Athenaum/ Hain/ Hanstein, Konigstein, pp. 497- 512.

74. Kruse, R. Gebhard, J. and Klawonn, F (1994): *Foundations of fuzzy Systems.* John Wiley & sons.

75. L. A, Zadeh. (1976), *The Concept of a linguistic Variable and its Application to Approximate Reasoning*, Part 3, Information Sciences 9, pp. 43-80.

76. L. A. Zadeh. (1976), *The Concept of a linguistic Variable and its Application to Approximate Reasoning*, Part 3, Information Sciences , pp. 43-80.

77. L.O.Chuna and L.Yang. (1988), *Celluler Neural networks*, IEEE Tran. Circuits and Systems Theory, pp. 1257-1290.

78. L.Zadeh. (1979), *Fuzzy sets and information granularity*, in M.M.Gupta et al, Eds, Advances in Fuzzy Sets Theory and Applications, North-Holland, London.

79. M.K, Pakhira., S, Bandyopadhyay., H, Maulik.(2004), *"Validity index for crisp and fuzzy clusters"*, Pattern Recognition, vol,37, pp. 487-501.

80. Mark S. Nixon and Alberto S. Aguado. (2002), *Feature Extraction and Image processing*, Linacre House, Jordan Hill, Oxford OX2 8DP.

81. N.Pal, J.C.Bezdek, E.C.Tsao. (1993), *Generalized clustering networks and Kohonen selforganizing scheme,* IEEE Trans, Networks, vol.4.

82. N.R,Pal., J.C, Bezdek. (1995), *On cluster validity for the fuzzy c-means model,* IEEE Trans, Fuzzy Systems, vol.3, pp. 370-379.

83. S.Haykin. (1994), *Neural Networks, a comprehensive foundation*, Macmillan College Publishing company, N, Y.

84. S.Osowski., M, Stodolski., P, Bojarczak. (1996), *"Fast second order learning Neural Networks*, vol.9, pp.1583-1596.

85. S.Osowski., K, Brudzewski. (2000), *Fuzzy self-organizing Hybrid Neural Network for gas analysis system,* IEEE Tran, on Instrumentation and Measurement, vol.49, pp. 424-429.

86. S.Osowski., T.H.Linh., K, Bruduzewski. (2004), *"Neuro-fuzzy networks for flavour recognition and classification"* IEEE Trans.on Instrumentation and Measurements, (in processing) (also in IEEE Instrumentation and Measurement Technology Conference, Anchorage, USA, May 21-22, pp. 165-170).

87. T.H.Linh., S, Osowski., K, Brudzewski. (2000), "*Supervised fuzzy neural networks for gas measurements"*, XXIII KKTOIUE, pp. 561-566.

88. T.H.Linh. (2002), *The Modification of TSK Network In Neuro-Fuzzy Systems*, XXV IC SPETO, pp. 525-528.

89. T.H.Linh. (2002), *Effective Algorithm of Atomatic Building TSK Network in Fuzzy Reasoning Systems*, IV CPEE Workshop, Zakopane, pp. 85-88.

90. T.Kohonen. (1995) *Self-organizing maps*, Springer Verlag, Berlin.

91. Yali Amt. (2002), *2D Object Detection and Recognition*, The MIT Press Publish.

BỘ CÔNG THƯƠNG – TRƯỜNG ĐẠI HỌC SAO ĐỎ

TS. ĐINH VĂN NHƯỢNG

GIÁO TRÌNH
MỘT SỐ ỨNG DỤNG MẠNG NƠRON XÂY DỰNG MÔ HÌNH NHẬN DẠNG VÀ DỰ BÁO

Chịu trách nhiệm xuất bản : PHẠM NGỌC KHÔI
Biên tập : HỒNG THỦY
Trình bày bìa : NGỌC TUẤN
Thiết kế sách và chế bản : THÁI SƠN

NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
70 TRẦN HƯNG ĐẠO, HÀ NỘI

In 500 bản, khổ 16 × 24 cm, tại Xí nghiệp in NXB Văn hóa Dân tộc.
Số ĐKKHXB: 384–2013/CXB/138–20/KHKT, ngày 27/3/2013.
Quyết định XB số: 96/QĐXB-NXBKHKT, ngày 20/6/2013.
In xong và nộp lưu chiểu Quý III năm 2013.