

**BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC SAO ĐỎ**



LÊ NGỌC TUẤN

**ỨNG DỤNG MẠNG NƠ-RÔN NHÂN TẠO DỰ BÁO PHỤ TẢI ĐIỆN
THÀNH PHỐ HẠ LONG - TỈNH QUẢNG NINH**

LUẬN VĂN THẠC SĨ
CHUYÊN NGÀNH: KỸ THUẬT ĐIỆN TỬ

NGƯỜI HƯỚNG DẪN KHOA HỌC:

- 1. PGS.TSKH TRẦN HOÀI LINH**
- 2. TS. ĐỖ VĂN ĐÌNH**

HẢI DƯƠNG – NĂM 2019

LỜI CAM ĐOAN

Tôi xin cam các kết quả nghiên cứu trong luận văn tốt nghiệp này là các kết quả thu được trong quá trình nghiên cứu của riêng học viên với sự hướng dẫn của PGS.TSKH Trần Hoài Linh và TS. Đỗ Văn Đỉnh. Không sao chép bất kỳ kết quả nghiên cứu nào của các tác giả khác.

Nội dung nghiên cứu có tham khảo và sử dụng một số thông tin, tài liệu từ các nguồn tài liệu đã được liệt kê trong danh mục tài liệu tham khảo.

Nếu sai tôi xin chịu mọi hình thức kỷ luật theo quy định.

Hải Dương, ngày 28 tháng 12 năm 2019

Tác giả luận văn

Lê Ngọc Tuấn

LỜI CẢM ƠN

Với lòng kính trọng và biết ơn, đầu tiên em xin chân thành gửi lời cảm ơn tới PGS.TSKH Trần Hoài Linh và TS. Đỗ Văn Đình, hai thầy đã tận tình hướng dẫn, giúp đỡ em rất nhiều về kiến thức cũng như tài liệu kỹ thuật và cho em nhiều ý kiến quý báu trong suốt quá trình làm luận văn.

Em xin chân thành cảm ơn các quý thầy cô đã trực tiếp giảng dạy tác giả trong toàn khóa học; Cảm ơn tập thể các thầy cô giáo Khoa Điện, trường Đại học Sao Đỏ; Cảm ơn tập thể, lãnh đạo công ty điện lực thành phố Hạ Long, tỉnh Quảng Ninh đã hỗ trợ, cung cấp số liệu để học viên nghiên cứu; Cảm ơn anh, em, bạn bè, đồng nghiệp và gia đình đã động viên, hỗ trợ, đóng góp ý kiến giúp em hoàn thành luận văn này.

Trong thời gian không dài, em đã rất nỗ lực và cố gắng để hoàn thành luận văn tốt nghiệp này nhưng sự hiểu biết và thời gian nghiên cứu thực tế có hạn nên không tránh khỏi những thiếu sót, rất mong nhận được sự góp ý của các thầy, cô và bạn đọc để luận văn của tác giả được hoàn thiện hơn.

Em xin trân trọng cảm ơn!

MỤC LỤC

	Trang
LỜI CAM ĐOAN	i
LỜI CẢM ƠN.....	ii
MỤC LỤC	iii
DANH MỤC CÁC KÝ HIỆU, CÁC TỪ VIẾT TẮT	v
DANH MỤC CÁC BẢNG.....	vi
DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ	vii
MỞ ĐẦU	1
1. Lý do chọn đề tài	1
2. Tính cấp thiết của đề tài.....	1
3. Mục tiêu nghiên cứu	2
4. Đối tượng và phạm vi nghiên cứu	2
5. Phương pháp nghiên cứu.....	2
6. Ý nghĩa khoa học và thực tiễn của đề tài.....	2
6.1. Ý nghĩa khoa học	2
6.2. Ý nghĩa thực tiễn.....	3
7. Cấu trúc của đề tài	3
CHƯƠNG 1. TỔNG QUAN VỀ DỰ BÁO PHỤ TẢI ĐIỆN.....	4
1.1. Đặt vấn đề.....	4
1.2. Tình hình nghiên cứu ngoài nước.....	4
1.3. Tình hình nghiên cứu trong nước	5
1.4. Định hướng nghiên cứu của đề tài.....	6
1.5. Kết luận chương 1	6
CHƯƠNG 2. TỔNG QUAN VỀ ỨNG DỤNG MẠNG NƠ-RÔN NHÂN TẠO TRONG DỰ BÁO PHỤ TẢI ĐIỆN.....	7
2.1. Trí tuệ nhân tạo [23, 27].	7
2.2. Nơ-rôn sinh học	7
2.3. Mạng nơ rôn nhân tạo.....	8
2.3.2. Mô hình mạng nơ rôn nhân tạo [25, 2].	10
2.3.3. Các hàm kích hoạt [1, 2].	11
2.3.4. Phân loại các mạng nơ rôn [25, 5].	13
2.4. Huấn luyện mạng nơ rôn [25, 2].....	14
2.5. Một số ứng dụng của mạng nơ rôn.....	15

2.6. Kết luận chương 2.	16
CHƯƠNG III: MÔ HÌNH LAI VÀ ỨNG DỤNG TRONG ƯỚC LƯỢNG PHI TUYẾN	17
3.1. Mô hình lai [1, 3].....	17
3.2. Chọn đặc tính đầu vào của mô hình dự báo [1, 3].....	24
3.2.1. Phương pháp phân tích thành phần chính – PCA	26
3.2.2. Phương pháp phân tích sự khác biệt tuyến tính LDA.....	27
3.2.3. Phương pháp phân tích giá trị kỳ dị (SVD)	28
3.3. Ứng dụng giải thuật SVD xây dựng khối tuyến tính [3].....	28
3.4. Ứng dụng mạng MLP xây dựng khối phi tuyến [1, 3].....	31
3.5. Kết luận chương III	38
CHƯƠNG IV: XÂY DỰNG MÔ HÌNH, TÍNH TOÁN VÀ MÔ PHỎNG.....	39
4.1. Xây dựng mô hình dự báo công suất phụ tải.....	39
4.1.1. Mô hình dự báo công suất phụ tải [1,3]	39
4.1.2. Quy trình xây dựng các mô hình dự báo trong luận văn [1].....	39
4.2. Kết quả tính toán và mô phỏng.....	46
4.2.1. Bộ số liệu đầu vào sử dụng cho mô hình dự báo	46
4.2.2. Kết quả mô hình dự báo công suất đỉnh phụ tải P_{max}	47
4.2.3. Kết quả mô hình dự báo công suất đáy phụ tải P_{min}	51
KẾT LUẬN VÀ KIẾN NGHỊ.....	56
1. Kết luận.....	56
2. Kiến nghị	56
DANH MỤC CÁC TÀI LIỆU THAM KHẢO	57
PHỤ LỤC	1

DANH MỤC CÁC KÝ HIỆU, CÁC TỪ VIẾT TẮT

Từ, cụm từ viết tắt	Tiếng Anh	Tiếng Việt
AI	Artificial Intelligence	Trí thông minh nhân tạo
BDPT		Biểu đồ phụ tải
HTĐ		Hệ thống điện
MLP	Multi Layer Perceptron	Mạng nơ- rôn truyền thẳng nhiều lớp
MNR		Mạng nơ- rôn
SOM	Self Organization Map	Kỹ thuật trực quan hoá dữ liệu
TTNT		Trí tuệ nhân tạo

DANH MỤC CÁC BẢNG

	Trang
Bảng 4.1. Kết quả ước lượng công suất đỉnh phụ tải (P_{max}) dùng phương pháp khai triển tuyến tính SVD.....	47
Bảng 4.2. Một số kết quả trên bộ số liệu học của bài toán ước lượng P_{min}	49
Bảng 4.3. Kết quả ước lượng công suất đáy phụ tải (P_{min}) dùng phương pháp khai triển tuyến tính SVD	51
Bảng 4.4. Kết quả sai số học và sai số kiểm tra khi ước lượng P_{min} bằng MLP	53

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

	Trang
Hình 2.1. Cấu tạo nơ- rôn sinh học [28].	8
Hình 2.2. Mô hình mạng nơ- rôn nhân tạo[25]	10
Hình 2.3. Hàm đồng nhất (Identity function)	11
Hình 2.4. Hàm bước nhị phân (Binary step function)	11
Hình 2.5. Hàm Sigmoid	12
Hình 2.6. Hàm sigmoid lưỡng cực	12
Hình 2.7. Mạng nơ- rôn truyền thẳng nhiều lớp (Feed-forward neural network)[25]	13
Hình 2.8. Mạng nơ- rôn hồi quy (Recurrent neural network)[25]	14
Hình 2.9. Mô hình Học có thầy (Supervised learning model)[25]	15
Hình 3.1. Sơ đồ khối tổng thể của mô hình lai [1]	17
Hình 3.2. Khối phi tuyến ở dạng tổng quát[1]	20
Hình 3.3. Khối phi tuyến kết hợp với sai số đầu ra của khối tuyến tính [1]	21
Hình 3.4. Mô hình mạng nơ-rôn truyền thẳng với hai lớp ẩn [3]	23
Hình 3.5. Mô hình mạng nơ-rôn truyền thẳng với một lớp ẩn [3]	23
Hình 3.6. Cấu trúc khối trích chọn đặc tính [1]	25
Hình 3.7. Cấu trúc tổng quát các bước thực hiện trích chọn đặc tính	25
Hình 3.8. Cấu trúc mô hình theo hướng Filter	26
Hình 3.9. Cấu trúc mô hình theo hướng Wrapper	26
Hình 3.10. Không gian mới được tạo ra qua phép biến đổi PCA [1]	27
Hình 3.11. Xác định mạng MLP xấp xỉ hàm truyền của đối tượng [1]	31
Hình 4.1. Đồ thị biểu diễn bộ số liệu mẫu (P_{max} , P_{min})	46
Hình 4.2. Đồ thị sai số học khi khai triển thành phần tuyến tính cho mô hình dự báo P_{max}	48
Hình 4.3. Đồ thị sai số quá trình kiểm tra khi khai triển thành phần tuyến tính cho mô hình dự báo P_{max}	48
Hình 4.4. Quý trình học của mạng MLP khi ước lượng P_{max}	49
Hình 4.5. Đồ thị sai số quá trình học khi ước lượng P_{max} bằng mạng MLP	50
Hình 4.6. Đồ thị sai số quá trình kiểm tra khi ước lượng P_{max} bằng mạng MLP	50
Hình 4.7. Đồ thị sai số học khi khai triển thành phần tuyến tính cho mô hình dự báo P_{min}	52
Hình 4.8. Đồ thị sai số quá trình kiểm tra khi khai triển thành phần tuyến tính cho mô hình dự báo P_{min}	52

Hình 4.9. Quý trình học của mạng MLP khi ước lượng P_{\min} 53
Hình 4.10. Đồ thị sai số quá trình học khi ước lượng P_{\min} bằng mạng MLP54
Hình 4.11. Đồ thị sai số quá trình kiểm tra khi ước lượng P_{\min} bằng mạng MLP54

MỞ ĐẦU

1. Lý do chọn đề tài

Hiện nay, quá trình Công nghiệp hoá, Hiện đại hoá đất nước và những ứng dụng thành tựu khoa học kỹ thuật ngày càng được triển khai rộng rãi đã đẩy nhanh sự phát triển của nền kinh tế, dẫn đến nhu cầu sử dụng điện tăng cao, do vậy xây dựng công cụ dự báo về phụ tải điện, làm cơ sở cho việc vận hành và khai thác hiệu quả nguồn và lưới điện, tiến tới hỗ trợ quy hoạch phát triển nguồn là vấn đề cần thiết.

Việc dự báo nhu cầu điện năng ngắn hạn nhằm hỗ trợ cho quá trình vận hành, điều độ lưới xác định quy luật và đặc điểm của phụ tải điện nhằm phát hiện và giải quyết các vấn đề tiềm ẩn trong hệ thống điện.

Trên biểu đồ phụ tải có hai giá trị đặc biệt là phụ tải cao điểm P_{\max} và phụ tải thấp điểm P_{\min} và dạng của biểu đồ phụ tải là yếu tố quan trọng hàng đầu của người làm công tác dự báo. Để đạt được hiệu quả kinh tế cũng như hướng phát triển tương lai của HTĐ thì phụ thuộc nhiều vào dự báo chính xác hai giá trị P_{\max} và P_{\min} .

Đến nay đã có rất nhiều mô hình giải pháp được đề xuất và ứng dụng cho dự báo phụ tải điện ngắn hạn, tuy nhiên cho tới thời điểm này vẫn chưa có một mô hình nào được coi là chuẩn và áp dụng hiệu quả.

Trong luận văn này tác giả đã đề xuất một mô hình dự báo phụ tải điện ngắn hạn sử dụng mô hình hỗn hợp (là mô hình sử dụng phối hợp cả hai giải pháp phi tuyến và tuyến tính) và trình bày khả năng ứng dụng của mô hình này trong thực tiễn dự báo phụ tải điện ở thành phố Hạ Long tỉnh Quảng Ninh.

2. Tính cấp thiết của đề tài.

Trong những năm gần đây, quá trình công nghiệp hóa, hiện đại hóa đất nước ngày càng được nâng cao nhằm thúc đẩy mạnh mẽ sự phát triển của nền kinh tế quốc dân, các đề tài nghiên cứu khoa học được áp dụng nhiều vào thực tế cuộc sống đặc biệt là các ngành sản xuất nhằm nâng cao hiệu quả sản xuất. Đồng nghĩa với việc ngày càng áp dụng các máy móc để thay thế con người vào dây chuyền sản xuất thì nhu cầu sử dụng điện năng ngày càng cao do vậy việc dự báo phụ tải là một trong những đề tài được quan tâm chủ yếu trong lĩnh vực phát điện, truyền tải và phân phối điện năng. Có những phương án để tăng nguồn điện năng phục vụ cho sản xuất ngày càng phát triển là xây dựng các nhà máy phát điện, nâng cấp đường dây tải điện, trạm biến áp nhưng có phần hạn chế là thiếu sự đồng bộ giữa các hệ thống vì hiện nay việc nâng cấp hệ thống không dựa trên quy hoạch tổng thể mà còn mang tính tự phát. Do vậy đã xảy ra tình trạng như: mất cân đối giữa nguồn và lưới điện, giữa cung và cầu, chất lượng điện năng bị giảm xuống, độ tin cậy của hệ thống không cao, tổn thất điện năng khi truyền tải lớn. Có nhiều nguyên nhân dẫn đến các hậu quả trên nhưng nguyên nhân chủ yếu là lập kế hoạch để vận hành hệ thống điện (HTĐ). Trong kế hoạch vận hành hệ thống điện như phân phối một cách kinh tế, hiệu quả, lên kế hoạch bảo dưỡng và sửa chữa,...

hiều hoạt động thường được thực hiện dựa trên việc dự báo phụ tải nhằm giảm hạn chế tối đa việc ngừng cung cấp điện. Dự báo phụ tải có thể được phân thành: Dự báo dài hạn từ 10 đến 30 năm, dự báo trung hạn từ 3 đến 10 năm và dự báo ngắn hạn (giờ, ngày, tháng, mùa năm). Dự báo ngắn hạn chủ yếu phục vụ việc điều hành sản xuất, truyền tải và phân phối năng lượng phục vụ cho nhu cầu trực tiếp của sản xuất và đời sống, lập kế hoạch sản xuất và kinh doanh.

Dự báo phụ tải điện là điều hết sức quan trọng, nếu dự báo thừa quá nhiều so với nhu cầu thực tế khiến cho phải phát huy số lượng nguồn phát lớn, nhưng không sử dụng hết công suất của chúng sẽ gây ra lãng phí, nếu dự báo quá thấp so với nhu cầu thực tế thì sẽ làm thiếu hụt nguồn điện do vậy phải cắt điện làm ảnh hưởng đến việc sản xuất. Do vậy việc dự báo tốt sẽ giúp phát huy tối đa được các nguồn phát và hạn chế được việc phải cắt điện làm ảnh hưởng đến sản xuất, sinh hoạt đời sống.

3. Mục tiêu nghiên cứu

- Xây dựng mô hình dự báo công suất phụ tải điện dựa trên kết quả tiêu thụ điện năng trong quá khứ từ số liệu tổng hợp của các Công ty điện lực.
- Nghiên cứu, áp dụng mạng nơ- rôn trong xây dựng các mô hình dự báo phi tuyến.
- Đề xuất mô hình dự báo công suất phụ tải điện cho thành phố Hạ Long, tỉnh Quảng Ninh.

4. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu là bài toán dự báo phụ tải điện của hệ thống điện cụ thể ứng dụng cho phụ tải thành phố Hạ Long- tỉnh Quảng Ninh.
- Phạm vi nghiên cứu: dự báo công suất đỉnh phụ tải (P_{max}) và công suất đáy phụ tải (P_{min}) trong ngày.

5. Phương pháp nghiên cứu

- Nghiên cứu tài liệu: Các mô hình, phương pháp dự báo phụ tải điện đã được công bố trên sách, tạp chí,...
- Nghiên cứu thực nghiệm: tính toán, mô phỏng trên phần mềm Matlab dự báo công suất phụ tải điện

6. Ý nghĩa khoa học và thực tiễn của đề tài

6.1. Ý nghĩa khoa học

Bài toán dự báo phụ tải của hệ thống điện đã có nhiều nghiên cứu trên thực tế tại nhiều nước trên thế giới. Tuy nhiên trong điều kiện Việt Nam là nước đang có nền kinh tế phát triển với tốc độ cao, các mô hình đã được xây dựng chuyên dụng cho các nước có nền kinh tế đã phát triển không đáp ứng được độ phức tạp của bài toán dự báo tại Việt Nam. Ở nước ta, các giải pháp dự báo sử dụng mạng nơ- rôn nhân tạo cũng chỉ dừng ở mức lý thuyết hoặc để giải một số bài toán đơn giản, có lượng thông tin cần xử

lý thấp hoặc độ phức tạp của mô hình không cao. Đến nay đã có một vài nghiên cứu ứng dụng mạng nơ- rôn trong dự báo phụ tải hệ thống điện và so sánh với những phương pháp hiện tại. Tuy nhiên, các kết quả mới chỉ là bước đầu, chưa thực sự trở thành hướng nghiên cứu phổ biến. Do đó, việc xây dựng mô hình ứng dụng mạng nơ- rôn nhân tạo trong dự báo là một hướng nghiên cứu nên được triển khai và thử nghiệm.

6.2. Ý nghĩa thực tiễn

Kết quả nghiên cứu của đề tài có thể áp dụng vào thực tiễn để dự báo công suất phụ tải điện ở các khu vực, vùng kinh tế để có những định hướng điều chỉnh quy hoạch lưới điện và quy hoạch phát triển kinh tế.

7. Cấu trúc của đề tài

Cấu trúc của luận văn gồm 04 chương, ngoài ra còn mục lục, danh sách các ký hiệu, từ viết tắt; bảng/hình vẽ, đồ thị; các tài liệu tham khảo; cụ thể:

Chương 1: Tổng quan về dự báo phụ tải điện

Chương 2: Tổng quan về ứng dụng mạng nơ-rôn nhân tạo

Chương 3: Mô hình lai và ứng dụng trong ước lượng phi tuyến

Chương 4: Xây dựng mô hình, tính toán và mô phỏng

Kết luận và kiến nghị

CHƯƠNG 1. TỔNG QUAN VỀ DỰ BÁO PHỤ TẢI ĐIỆN

1.1. Đặt vấn đề

Hiện nay, trong đời sống sinh hoạt, phát triển kinh tế, xã hội của mỗi quốc gia thì điện năng là nguồn năng lượng không thể thiếu được và sử dụng càng nhiều hơn nữa trong tương lai. Nhưng có một vấn đề đặt ra là điện năng tiêu thụ không phải lúc nào cũng giống nhau mà phụ thuộc chủ yếu vào nhu cầu sử dụng như: số giờ làm việc, số giờ sản xuất kinh doanh, nhiệt độ môi trường Việc sản xuất điện năng khác với các ngành khác sản xuất khác là khả năng đáp ứng nhu ngay nhu cầu sử dụng và gần như không thể dự trữ được. Do vậy, dự báo phụ tải điện là một yếu tố rất quan trọng để tính toán sản xuất, truyền tải, phân phối điện năng nhằm đáp ứng được nhu cầu sử dụng của khách hàng.

1.2. Tình hình nghiên cứu ngoài nước.

Ngày 13/6/2018, Tập đoàn BP công bố "Đánh giá thống kê năng lượng thế giới" (BP Statistical Review of World Energy) trình bày dữ liệu cập nhật mới nhất của mình về tình hình năng lượng trên thế giới. Đây là lần đầu tiên báo cáo thường niên này đưa thêm các số liệu về điện [22]. 85,2% nhiên liệu hóa thạch trong hỗn hợp năng lượng toàn cầu.

Vào tháng 3/2018, Cơ quan Năng lượng Quốc tế (IEA) đã báo cáo rằng, mức tiêu thụ năng lượng toàn cầu vào năm 2017 đã tăng 2,1% so với cùng kỳ năm trước. Trong báo cáo của mình, BP cho rằng mức tăng thực chất là 2,2% và xác nhận rằng, tất cả các nguồn năng lượng đã được sử dụng vào năm ngoái trên thế giới, bao gồm cả than (+ 1%, do nhu cầu tăng ở Ấn Độ). Theo BP, đây lần đầu tiên kể từ năm 2013, toàn bộ các nguồn năng lượng trên thế giới đều được sử dụng [22].

Trong tổng mức tiêu thụ đó, nhiên liệu hóa thạch vẫn chiếm 85,2%, giảm rất ít so với mức 85,5% của năm 2016. Dầu vẫn là nguồn cung cấp năng lượng chính trong hỗn hợp năng lượng toàn cầu (34,2%), với mức tăng 1,7 triệu thùng/ngày vào năm 2017.

Việc phát triển các nguồn năng lượng tái tạo ngoại trừ thủy điện đặc biệt tăng mạnh (+ 16,6%) nhưng vẫn chỉ chiếm 3,5% tổng mức tiêu thụ năng lượng sơ cấp của thế giới vào năm 2017. Tỷ trọng năng lượng hạt nhân vẫn tương đối ổn định (4,4% trong năm 2017) [22].

Đánh giá thống kê của BP về năng lượng thế giới đã cung cấp các dữ liệu chi tiết về dầu khí. Tiêu thụ dầu và khí đốt tự nhiên tăng lần lượt 1,4% và 2,7% trong năm 2017 [22].

Hỗn hợp điện vẫn bị thống trị bởi điện than

Sản lượng điện thế giới tăng 2,8% trong năm 2017. Đa phần mức tăng này đến từ các nước mới nổi trong khi sản xuất điện ở các nước thuộc Tổ chức Hợp tác và Phát triển Kinh tế (OCDE) vẫn tương đối ổn định từ năm 2010. Năng lượng tái tạo và than

lần lượt chiếm 49% và 44% tổng mức tăng của năng lượng điện toàn cầu trong năm 2017 [22].

Tuy nhiên, than vẫn còn lâu mới là nguồn cung cấp điện lớn nhất cho thế giới: nhiên liệu này chiếm tới 38% sản lượng điện thế giới, tương đương mức của năm 1998 mặc dù có sự sụt giảm nhẹ trong những năm gần đây. Tỷ trọng năng lượng “phi hóa thạch” thậm chí còn thấp hơn mức của 20 năm trước, sự gia tăng của năng lượng tái tạo (chiếm 25% sản lượng điện toàn cầu vào năm 2017) chỉ bù đắp được một phần cho sự sụt giảm của năng lượng hạt nhân (giảm 10% vào năm 2017) [22].

Một bước thụt lùi lớn

Lượng phát thải CO₂ toàn cầu liên quan đến sử dụng năng lượng tăng 1,6% trong năm 2017 sau 3 năm liên tiếp ở mức ổn định. Sự suy giảm cường độ năng lượng đã chậm lại trong bối cảnh tăng trưởng kinh tế mạnh mẽ và sự gia tăng về nhu cầu than đã ảnh hưởng đến lượng phát thải này [22].

Sự thay đổi này, được BP mô tả là “bước lùi lớn”, khẳng định khoảng cách rất quan trọng giữa các tham vọng thể hiện ở Hội nghị COP21 về chống biến đổi khí hậu và xu hướng hiện tại. IPCC (nhóm các chuyên gia liên chính phủ về thay đổi khí hậu) ước tính rằng, để đạt được mục tiêu của Hiệp định Paris tại COP21 thì thế giới cần phải giảm phát thải khí nhà kính toàn cầu 40-70% vào năm 2050 (so với mức của năm 2010) và nền kinh tế toàn cầu phải gần như trung hòa được lượng khí carbon trong nửa sau của thế kỷ XXI [22].

1.3. Tình hình nghiên cứu trong nước

Ở nước ta dự báo phụ tải ngắn hạn cũng thu hút sự quan tâm nhất định trong ngành điện, các trường Đại học và Viện nghiên cứu. Một vài công ty điện lực, cơ quan điều độ quốc gia đã bắt đầu xây dựng chương trình dự báo phụ tải điện. Ngoài một số nghiên cứu ứng dụng mô hình hồi quy. Có thể thấy rằng nhu cầu dự báo phụ tải ngắn hạn ở các công ty điện lực chỉ được thực hiện bằng phương pháp chuyên gia dựa trên kinh nghiệm của các kỹ sư điều độ. Nên mang tính chất rủi ro và độ chính xác không cao.

Nếu dự báo phụ tải điện quá thừa so với nhu cầu thực tế thì dẫn đến việc phải huy động một số lượng lớn nguồn phát, nhưng thực tế không dùng hết công suất của chúng sẽ gây lãng phí. Nếu kết quả dự báo phụ tải điện quá thấp so với nhu cầu thực tế sẽ dẫn đến tình trạng thiếu nguồn điện, ảnh hưởng đến phát triển của nền kinh tế quốc dân. Khi kết quả dự báo tốt sẽ giúp cho việc huy động nguồn tốt và điều độ cũng tốt có lợi ích về kinh tế.

Do vậy, việc áp dụng mạng nơ- rôn nhân tạo cho dự án báo phụ tải điện cho một thành phố du lịch là yếu tố rất quan trọng.

1.4. Định hướng nghiên cứu của đề tài

- Tìm hiểu về các mô hình thu thập, xử lý, dự báo công suất phụ tải điện.
- Nghiên cứu về khả năng ứng dụng mạng nơ- rôn trong xây dựng các mô hình dự báo phi tuyến.
- Đề xuất ứng dụng mô hình lai gồm một khối tuyến tính mắc song song với một khối phi tuyến (sử dụng mạng MLP) để ước lượng tốt hơn trong dự công suất phụ tải điện chỉ dựa trên các kết quả đo công suất phụ tải trong ngày ở quá khứ từ số tổng hợp của Công ty điện lực.
- Đánh giá, lựa chọn các thông số quá khứ cho mô hình dự báo trên cơ sở các hệ số khai triển tuyến tính.
- Triển khai, thử nghiệm mô hình dự báo công suất đỉnh phụ tải và công suất đáy phụ tải trong ngày cho khu vực Thành phố Hạ Long, tỉnh Quảng Ninh.
- Tính toán mô phỏng để thử nghiệm độ chính xác của mô hình được đề xuất.

1.5. Kết luận chương 1

Nhu cầu sử dụng năng lượng điện hiện nay ngày càng tăng cao đưa ra không ít những khó khăn về sản xuất, truyền tải và phân phối điện năng nhằm đáp ứng đủ cho nhu cầu sử dụng điện hiện nay. Việc ứng dụng mạng nơ- rôn nhân tạo để dự báo phụ tải điện là một trong những giải pháp dự báo lượng điện năng tiêu thụ từ đó đưa ra được các phương thức sản xuất cho phù hợp với nhu cầu thực tiễn. Tiếp theo chương 2 sẽ đi tìm hiểu về mạng nơ- rôn nhân tạo.

CHƯƠNG 2. TỔNG QUAN VỀ ỨNG DỤNG MẠNG NƠ-RÔN NHẬN TẠO TRONG DỰ BÁO PHỤ TẢI ĐIỆN

2.1. Trí tuệ nhân tạo [23, 27].

Trí tuệ nhân tạo là trí thông minh của máy móc do con người tạo ra có khả năng thực hiện các công việc mà con người thường phải xử lý và khi những dáng vẻ, khả năng xử lý một số công việc tốt hơn hoặc tương đương với con người thì đó được gọi là máy thông minh hay máy đó có trí thông minh. Sự thông minh của máy móc được đánh giá trên kết quả của nó có giống với con người hay không.

(AI: Artificial Intelligence) có thể được định nghĩa như một ngành của khoa học máy tính liên quan đến việc tự động hóa các hành vi thông minh. AI là một bộ phận của khoa học máy tính và do đó nó phải được đặt trên những nguyên lý lý thuyết vững chắc, có khả năng ứng dụng được của lĩnh vực này [23, 27].

Khái niệm về công nghệ AI xuất hiện đầu tiên bởi John McCarthy, một nhà khoa học máy tính Mỹ, vào năm 1956 tại Hội nghị The Dartmouth. Ngày nay, công nghệ AI là một thuật ngữ bao gồm tất cả mọi thứ từ quá trình tự động hoá robot đến người máy thực tế [23, 27].

Công nghệ AI gần đây trở nên nổi tiếng, nhận được sự quan tâm của nhiều người là nhờ Dữ liệu lớn (Big Data), mối quan tâm của các doanh nghiệp về tầm quan trọng của dữ liệu cùng với công nghệ phân cứng đã phát triển mạnh mẽ hơn, cho phép xử lý công nghệ AI với tốc độ nhanh hơn bao giờ hết [23].

Ở thời điểm hiện tại, Thuật ngữ này thường dùng để nói đến các máy tính có mục đích không nhất định và ngành khoa học nghiên cứu về các lý thuyết và ứng dụng của trí tuệ nhân tạo. Tức là mỗi loại trí tuệ nhân tạo hiện nay đang dừng lại ở mức độ những máy tính hoặc siêu máy tính dùng để xử lý một loại công việc nào đó như điều khiển một ngôi nhà, nghiên cứu nhận diện hình ảnh, xử lý dữ liệu của bệnh nhân để đưa ra phác đồ điều trị, xử lý dữ liệu để tự học hỏi, khả năng trả lời các câu hỏi về chẩn đoán bệnh, trả lời khách hàng về các sản phẩm của một công ty,...

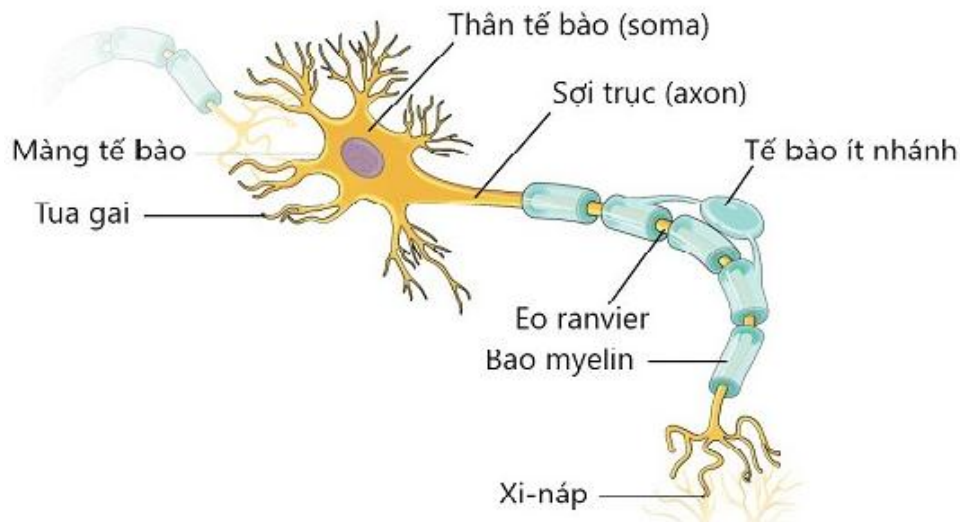
2.2. Nơ-rôn sinh học

Nơ-rôn sinh học là những tế bào thần kinh chính thức có chức năng cảm ứng và truyền dẫn xung điện. Nó là đơn vị cơ bản cấu tạo nên hệ thống thần kinh và là phần quan trọng nhất của bộ não đóng vai trò rất quan trọng trong cơ thể. Khi chúng bị tổn thương, con người bị rối loạn vận động và cảm giác, dẫn đến hiện tượng run rẩy chân tay hoặc tê liệt một phần hoặc toàn bộ cơ thể.

Nơ- rôn là đơn vị cấu tạo của hệ thần kinh, trong hệ thần kinh có khoảng 1000 tỷ nơ- rôn. Mỗi nơ- rôn là một tế bào đặc biệt, có thể truyền các tín hiệu điện. Nơ- rôn có cấu trúc rẽ ngõ vào, thân tế bào và cấu trúc rẽ ngõ ra (sợi thần kinh). Các sợi thần kinh của một tế bào kết nối với các tế bào khác thông qua synapse. Khi một nơ- rôn kích hoạt, nó tạo ra xung điện hóa học dọc theo sợi thần kinh. Tín hiệu này đi qua các

synapse đến các nơ- rôn khác, và tiếp tục bị kích hoạt. Nơ- rôn hoạt động chỉ khi tất cả các tín hiệu nhận được ở thân tế bào thông qua rễ ngõ vào vượt quá một mức nào đó (ngưỡng hoạt động). Cường độ tín hiệu thu được của nơron phụ thuộc vào độ nhạy của synapse (tế bào thu nhận) [24].

Cấu tạo của mỗi nơ- rôn điển hình gồm các bộ phận: thân, đuôi gai và sợi trục.



Hình 2.1. Cấu tạo nơ- rôn sinh học [28].

2.3. Mạng nơ rôn nhân tạo

2.3.1. Lịch sử hình thành và phát triển.

Những năm gần đây, khá nhiều sách, báo, công trình nghiên cứu khoa học đề cập đến các kỹ thuật tính toán, người ta hay nhắc đến nhiều thuật ngữ như: máy tính thông minh, máy tính thế hệ V, hệ chuyên gia, mạng ngữ nghĩa,... Các ngôn ngữ lập trình như LISP, PROLOG mở đường cho việc áp dụng hàng loạt các hệ thống chương trình có khả năng “thông minh”.

Trước đây, mỗi khi nói đến Trí tuệ nhân tạo (TTNT) người ta thường quan tâm đến việc tạo lập các máy tính có khả năng “suy nghĩ”, thậm chí trong một số phạm vi hẹp nào đó, có thể cạnh tranh hoặc vượt quá khả năng của bộ não con người. Những hy vọng này trong một thời gian dài đã ảnh hưởng rất nhiều đến các nghiên cứu trong phòng thí nghiệm. Mặc dù những mô hình tương tự các máy tính thông minh đã được đưa ra nhiều năm trước, nhưng chỉ từ khi Alan Turing công bố những kết quả nghiên cứu quan trọng đầu tiên, người ta mới bắt đầu thực sự nghiên cứu đến các vấn đề TTNT một cách nghiêm túc. Phát hiện của Turing cho rằng chương trình có thể được lưu trữ trong bộ nhớ để sau đó được thực hiện trên cơ sở các phép toán cơ bản thao tác với các bit “0”, “1”. Điều này đã tạo nên nền tảng của những máy tính hiện đại. Việc lưu trữ chương trình trong máy cho phép thay đổi chức năng của nó một cách nhanh chóng và dễ dàng thông qua việc nạp một chương trình mới vào bộ nhớ. Theo một nghĩa nào đó, khả năng này làm cho máy tính có khả năng học và suy nghĩ. Đó cũng

chính là một trong những biểu hiện quan trọng đầu tiên của những máy tính được trang bị TTNT.

Năm 1956, chương trình dẫn xuất kết luận trong hệ hình thức đã được công bố. Tiếp theo đó, năm 1959 chương trình chứng minh các định lý hình học phẳng và chương trình giải quyết bài toán vạn năng (GPS - General Problem Solving) đã được đưa ra. Tuy vậy chỉ cho đến khoảng năm 1960 khi McCarthy ở MIT (Massachusetts Institute of Technology) đưa ra ngôn ngữ lập trình đầu tiên dùng cho trí tuệ nhân tạo LISP (list processing), các nghiên cứu về TTNT mới bắt đầu phát triển mạnh mẽ. Thuật ngữ TTNT do Marvin Minsky một chuyên gia nổi tiếng cũng ở MIT đưa ra năm 1961 trong bài báo “Steps Forwards To Artificial Intelligence”. Những năm 60 có thể xem là một mốc quan trọng trong quá trình xây dựng các máy có khả năng suy nghĩ. Các chương trình chơi cờ và các chương trình chứng minh định lý toán học đầu tiên cũng được công bố trong khoảng thời gian này [23].

Những bế tắc, hạn chế thành công của các công trình nghiên cứu TTNT trong những năm 60 chính là do giới hạn khả năng của các thiết bị, bộ nhớ và đặc biệt là yếu tố thời gian thực hiện. Chính những yếu tố này không cho phép tổng quát hóa những thành công bước đầu đạt được trong các hệ chương trình TTNT đã xây dựng. Tuy rằng vào giữa những năm 70, bộ nhớ máy tính và thời gian tính toán đã được nâng cao đáng kể về chất, song những cách tiếp cận khác nhau đến TTNT vẫn chưa đem tới những thành công thật sự do sự bùng nổ tổ hợp trong quá trình tìm kiếm lời giải cho các bài toán đặt ra.

Cuối những năm 70, một số nghiên cứu cơ bản trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, biểu diễn tri thức, lý thuyết giải quyết vấn đề đã đem lại diện mạo mới cho TTNT. Thị trường tin học đã bắt đầu đón nhận những sản phẩm TTNT ứng dụng đầu tiên mang tính thương mại. Đó là các hệ chuyên gia được áp dụng trong các lĩnh vực khác nhau. Hệ chuyên gia là các phần mềm máy tính, chứa các thông tin và tri thức về một lĩnh vực cụ thể nào đó, có khả năng giải quyết những yêu cầu của người dùng ở một mức độ nào đó với trình độ như một chuyên gia có kinh nghiệm lâu năm. Một trong những hệ chuyên gia đầu tiên được sử dụng thành công trong thực tế là hệ MYCIN, được thiết kế và cài đặt tại trường Đại học Tổng Hợp Stanford [23].

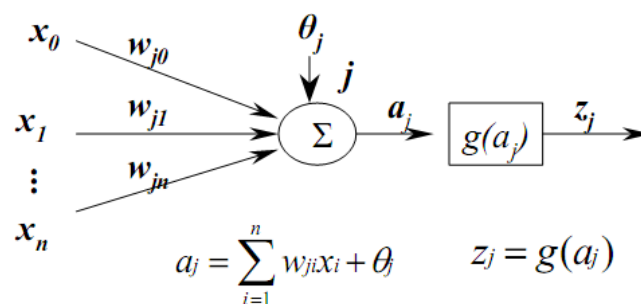
Một sự kiện quan trọng trong sự phát triển của khoa học TTNT là sự ra đời của ngôn ngữ PROLOG, do Alain Colmerauer đưa ra năm 1972. Năm 1981, dự án của Nhật Bản xây dựng các máy tính thế hệ thứ V lấy ngôn ngữ PROLOG như là ngôn ngữ cơ sở đã làm thay đổi khá nhiều tình hình phát triển TTNT ở Mỹ cũng như châu Âu.

Giai đoạn 1981 trở đi người ta cảm nhận khá rõ nét rằng các chuyên gia về TTNT đang dần chuyển các kết quả nghiên cứu từ phòng thí nghiệm sang cài đặt các ứng dụng cụ thể. Có thể nói đây cũng là giai đoạn cạnh tranh ráo riết của các công ty,

các viện nghiên cứu hàng đầu nhằm đưa ra thị trường các sản phẩm phần mềm ứng dụng kỹ thuật TTNT [23].

Cuối những năm 80, đầu những năm 90 thị trường các sản phẩm dân dụng đã có khá nhiều sản phẩm ở trình độ cao như máy giặt, máy ảnh,... sử dụng TTNT. Các hệ thống nhận dạng và xử lý hình ảnh, tiếng nói đang ngày càng thúc đẩy sự phát triển kỹ thuật mạng Neuron. Sự xích lại của hai cách tiếp cận: Tiếp cận mờ trong lập luận xấp xỉ và kỹ thuật mạng Neuron đã và đang gây được sự quan tâm đặc biệt của các chuyên gia tin học. Bên cạnh sự xuất hiện của các hệ chuyên gia, các ứng dụng công nghiệp và quản lý xã hội, quản lý kinh tế cũng đòi hỏi sự ra đời của các hệ thống xử lý tri thức – dữ liệu tích hợp.

2.3.2. Mô hình mạng nơ-ron nhân tạo [25, 2].



Hình 2.2. Mô hình mạng nơ-ron nhân tạo[25]

trong đó:

x_i : các đầu vào

w_{ji} : các trọng số tương ứng với các đầu vào θ_j : độ lệch (bias)

a_j : đầu vào mạng (net-input) z_j : đầu ra của nơ-ron

$g(x)$: hàm chuyển (hàm kích hoạt).

Trong một mạng nơ-ron có ba kiểu đơn vị:

Các đơn vị đầu vào (Input units), nhận tín hiệu từ bên ngoài.

Các đơn vị đầu ra (Output units), gửi dữ liệu ra bên ngoài.

Các đơn vị ẩn (Hidden units), tín hiệu vào (input) và ra (output) của nó nằm trong mạng.

Mỗi đơn vị j có thể có một hoặc nhiều đầu vào: $x_0, x_1, x_2, \dots, x_n$, nhưng chỉ có một đầu ra z_j . Một đầu vào tới một đơn vị có thể là dữ liệu từ bên ngoài mạng, hoặc đầu ra của một đơn vị khác, hoặc là đầu ra của chính nó.

Nơ-ron nhân tạo nhận một số các ngõ vào từ dữ liệu gốc, hay từ ngõ ra các nơron khác trong mạng. Mỗi kết nối đến ngõ vào có một cường độ (hay trọng số), những trọng số này tương ứng với tác dụng synapse trong nơron sinh học. Mỗi nơron cũng có một giá trị ngưỡng.

Tín hiệu được truyền qua hàm kích hoạt (hay còn gọi là hàm truyền) tạo giá trị ngõ ra nơ- rôn.

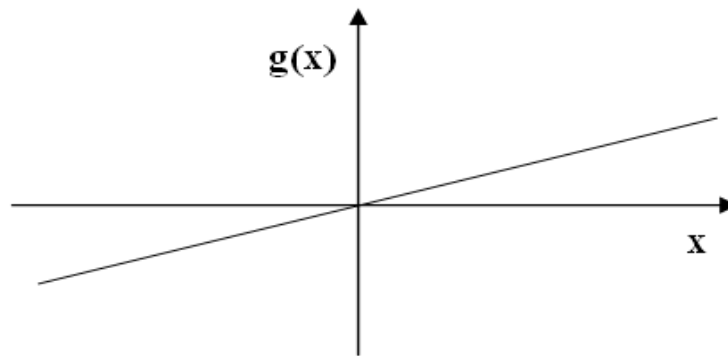
2.3.3. Các hàm kích hoạt [1, 2].

Phần lớn các đơn vị trong mạng nơ- rôn chuyển net input bằng cách sử dụng một hàm vô hướng (scalar-to-scalar function) gọi là hàm kích hoạt, kết quả của hàm này là một giá trị gọi là mức độ kích hoạt của đơn vị (unit's activation). Loại trừ khả năng đơn vị đó thuộc lớp ra, giá trị kích hoạt được đưa vào một hay nhiều đơn vị khác. Các hàm kích hoạt thường bị ép vào một khoảng giá trị xác định, do đó thường được gọi là các hàm bẹp (squashing). Các hàm kích hoạt hay được sử dụng là:

- Hàm đồng nhất (Linear function, Identity function).

$$g(x) = x \tag{2.1}$$

Nếu coi các đầu vào là một đơn vị thì chúng sẽ sử dụng hàm này. Đôi khi một hằng số được nhân với net-input để tạo ra một hàm đồng nhất.



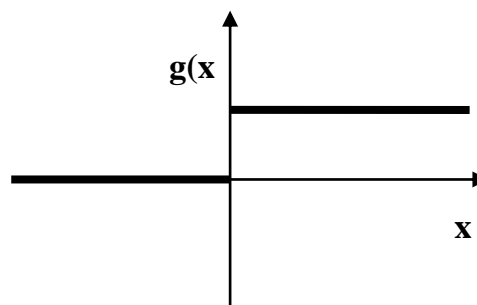
Hình 2.3. Hàm đồng nhất (Identity function)

- Hàm bước nhị phân (Binary step function, Hard limit function).

Hàm này cũng được biết đến với tên "Hàm ngưỡng" (Threshold function hay Heaviside function). Đầu ra của hàm này được giới hạn vào một trong hai giá trị:

$$g(x) = \begin{cases} 1, & \text{nếu } (x \geq \theta) \\ 0, & \text{nếu } (x < \theta) \end{cases} \tag{2.2}$$

Dạng hàm này được sử dụng trong các mạng chỉ có một lớp. Trong hình vẽ sau, θ được chọn bằng “1”.

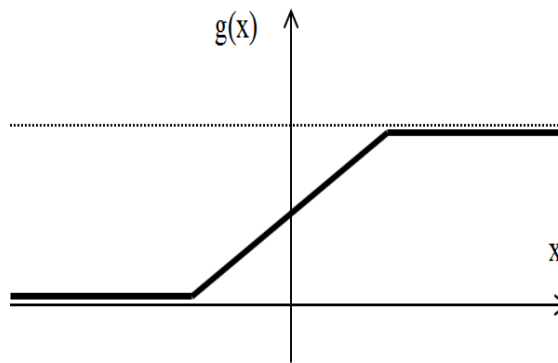


Hình 2.4. Hàm bước nhị phân (Binary step function)

- Hàm sigmoid (Sigmoid function (logsig))

$$g(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

Hàm này đặc biệt thuận lợi khi sử dụng cho các mạng được huấn luyện (trained) bởi thuật toán lan truyền ngược (back-propagation), bởi vì nó dễ lấy đạo hàm, do đó có thể giảm đáng kể tính toán trong quá trình huấn luyện. Hàm này được ứng dụng cho các chương trình ứng dụng mà các đầu ra mong muốn rơi vào khoảng [0,1].

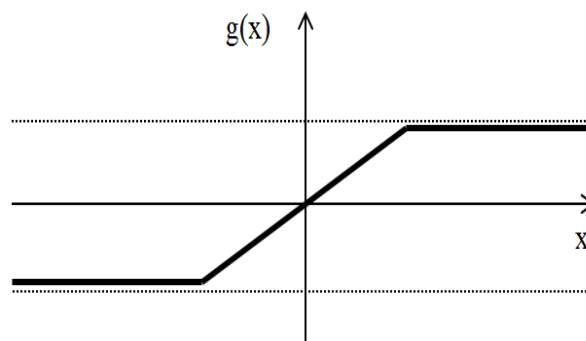


Hình 2.5. Hàm Sigmoid

- Hàm sigmoid lưỡng cực (Bipolar sigmoid function (tansig))

$$g(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{2.4}$$

Hàm này có các thuộc tính tương tự hàm sigmoid. Nó làm việc tốt đối với các ứng dụng có đầu ra yêu cầu trong khoảng [-1,1].



Hình 2.6. Hàm sigmoid lưỡng cực

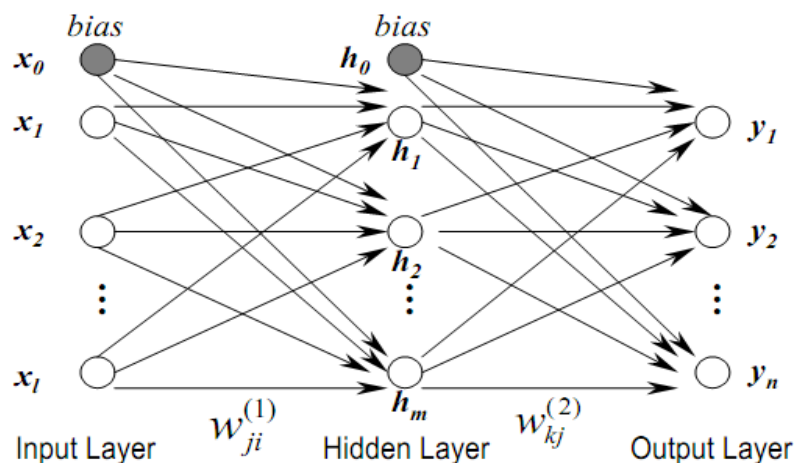
Các hàm chuyển của các đơn vị ẩn (hidden units) là cần thiết để biểu diễn sự phi tuyến vào trong mạng. Lý do là hợp thành của các hàm đồng nhất là một hàm đồng nhất. Mặc dù vậy nhưng nó mang tính chất phi tuyến (nghĩa là, khả năng biểu diễn các hàm phi tuyến) làm cho các mạng nhiều tầng có khả năng rất tốt trong biểu diễn các ánh xạ phi tuyến. Tuy nhiên, đối với luật học lan truyền ngược, hàm phải khả vi (differentiable) và sẽ có ích nếu như hàm được gán trong một khoảng nào đó. Do vậy, hàm sigmoid là lựa chọn thông dụng nhất.

Đối với các đơn vị đầu ra (output units), các hàm chuyển cần được chọn sao cho phù hợp với sự phân phối của các giá trị đích mong muốn. Chúng ta đã thấy rằng đối với các giá trị ra trong khoảng $[0,1]$, hàm sigmoid là có ích, đối với các giá trị đích mong muốn là liên tục trong khoảng đó thì hàm này cũng vẫn có ích, nó có thể cho ta các giá trị ra hay giá trị đích được căn trong một khoảng của hàm kích hoạt đầu ra. Nhưng nếu các giá trị đích không được biết trước khoảng xác định thì hàm hay được sử dụng nhất là hàm đồng nhất (identity function). Nếu giá trị mong muốn là dương nhưng không biết cận trên thì nên sử dụng một hàm kích hoạt dạng mũ (exponential output activation function).

2.3.4. Phân loại các mạng nơ-ron [25, 5].

- Mạng truyền thẳng (Feed-forward neural network):

Dòng dữ liệu từ đơn vị đầu vào đến đơn vị đầu ra chỉ được truyền thẳng. Việc xử lý dữ liệu có thể mở rộng ra nhiều lớp, nhưng không có các liên kết phản hồi. Nghĩa là, các liên kết mở rộng từ các đơn vị đầu ra tới các đơn vị đầu vào trong cùng một lớp hay các lớp trước đó là không cho phép.



Hình 2.7. Mạng nơ-ron truyền thẳng nhiều lớp (Feed-forward neural network)[25]

Trong đó:

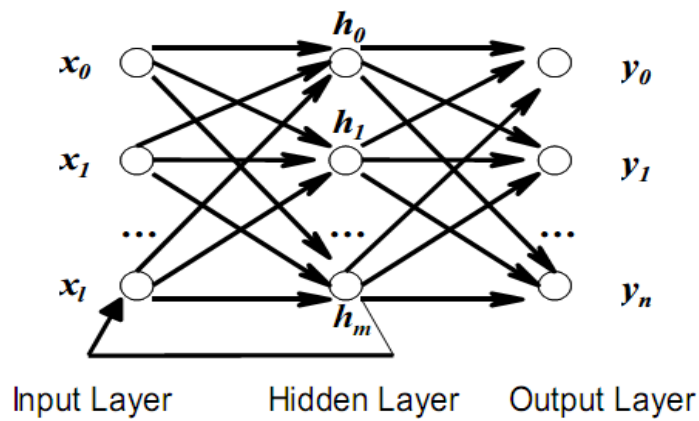
Input Layer: Lớp đầu vào

Hidden Layer: Lớp ẩn

Output Layer: Lớp đầu ra

- Mạng hồi quy (Recurrent neural network):

Có chứa các liên kết ngược. Khác với mạng truyền thẳng, các thuộc tính động của mạng mới quan trọng. Trong một số trường hợp, các giá trị kích hoạt của các đơn vị trải qua quá trình nói lỏng (tăng giảm số đơn vị và thay đổi các liên kết) cho đến khi mạng đạt đến một trạng thái ổn định và các giá trị kích hoạt không thay đổi nữa. Trong các ứng dụng khác mà cách chạy động tạo thành đầu ra của mạng thì những sự thay đổi các giá trị kích hoạt là đáng quan tâm.



Hình 2.8. Mạng nơ- rôn hồi quy (Recurrent neural network)[25]

Trong đó:

Input Layer: Lớp đầu vào

Hidden Layer: Lớp ẩn

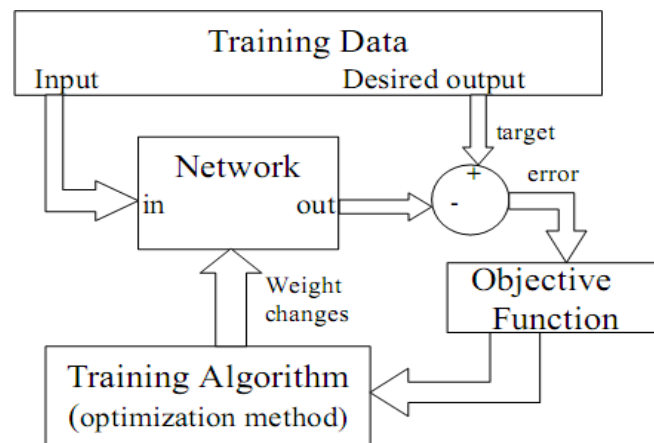
Output Layer: Lớp đầu ra

2.4. Huấn luyện mạng nơ- rôn [25, 2].

Chức năng của một mạng nơ- rôn được quyết định bởi các nhân tố như: hình trạng mạng (số lớp, số đơn vị trên mỗi tầng, và cách mà các lớp được liên kết với nhau) và các trọng số của các liên kết bên trong mạng. Hình trạng của mạng thường là cố định, và các trọng số được quyết định bởi một thuật toán huấn luyện (training algorithm). Tiến trình điều chỉnh các trọng số để mạng “nhận biết” được quan hệ giữa đầu vào và đích mong muốn được gọi là học (learning) hay huấn luyện (training). Rất nhiều thuật toán học đã được phát minh để tìm ra tập trọng số tối ưu làm giải pháp cho các bài toán. Các thuật toán đó có thể chia làm hai nhóm chính: Học có thầy (Supervised learning) và Học không có thầy (Unsupervised Learning).

- Học có thầy (Supervised learning):

Mạng được huấn luyện bằng cách cung cấp cho nó các cặp mẫu đầu vào và các đầu ra mong muốn (target values). Các cặp được cung cấp bởi "thầy giáo", hay bởi hệ thống trên đó mạng hoạt động. Sự khác biệt giữa các đầu ra thực tế so với các đầu ra mong muốn được thuật toán sử dụng để thích ứng các trọng số trong mạng. Điều này thường được đưa ra như một bài toán xấp xỉ hàm số - cho dữ liệu huấn luyện bao gồm các cặp mẫu đầu vào x , và một đích tương ứng t , mục đích là tìm ra hàm $f(x)$ thoả mãn tất cả các mẫu học đầu vào.



Hình 2.9. Mô hình Học có thầy (Supervised learning model)[25]

Trong đó:

Training Data: Dữ liệu học

Input: Đầu vào

Desired output: Đầu ra mong muốn

Network: Mạng nơ ron

Target: Giá trị mong muốn

Objective function: Hàm mục tiêu

Training Algorithm: Thuật toán

Weight changers (optimization method): Phương pháp tối ưu hóa

- Học không có thầy (Unsupervised Learning)

Với cách học không có thầy, không có phản hồi từ môi trường để chỉ ra rằng đầu ra của mạng là đúng. Mạng sẽ phải khám phá các đặc trưng, các điều chỉnh, các mối tương quan, hay các lớp trong dữ liệu vào một cách tự động. Trong thực tế, đối với phần lớn các biến thể của học không có thầy, các đích trùng với đầu vào. Nói một cách khác, học không có thầy luôn thực hiện một công việc tương tự như một mạng tự liên hợp, cô đọng thông tin từ dữ liệu vào.

2.5. Một số ứng dụng của mạng nơ ron

Theo đà phát triển của công nghệ, ứng dụng AI luôn là xu hướng công nghệ tương lai mà các hãng công nghệ trên toàn thế giới đua nhau sáng tạo, nó là nền tảng cốt lõi của cuộc cách mạng 4.0.

Ngày nay, các hệ thống trí tuệ nhân tạo được dùng thường xuyên trong kinh tế, y dược, các ngành kỹ thuật và quân sự, cũng như trong các phần mềm máy tính thông dụng trong gia đình và trò chơi điện tử.

Bước sơ khởi của AI trên các thiết bị nghe nhìn như tivi bắt đầu từ giữa năm 2017, khi mà AI giống như một trợ lí của người dùng, học hỏi cách mà người dùng sử dụng chính chiếc tivi của mình.

Với nền tảng trí tuệ nhân tạo ThinQ được tích hợp sẵn trên các dòng tivi của LG, người dùng dễ dàng tận hưởng những tính năng tiện lợi từ công nghệ hỗ trợ giọng nói tiên tiến, mọi thao tác với tivi được thực hiện một cách nhanh chóng chỉ bằng những mệnh lệnh bằng lời nói qua bộ điều khiển.

Samsung cũng đã tích hợp tính năng AI này thông qua SmartHub trên các dòng Tivi QLED của mình, cũng như IoT (Internet of Thing) trên các dòng tivi trước đó. Với công nghệ AI, Tivi QLED Q900R của Samsung có khả năng nâng cấp hình ảnh và âm thanh tương xứng với mức 8K từ các nguồn tín hiệu đầu vào bất kể chất lượng và định dạng.

Trí tuệ nhân tạo (AI) đã được tích hợp trên smartphone từ lâu kể từ khi trợ lý ảo Siri trên iPhone ra đời. Năm 2017 chứng kiến cuộc đua tích hợp trí tuệ nhân tạo dưới dạng trợ lý ảo của các hãng smartphone như trên Google Pixel, HTC U Ultra, LG G6 và sắp tới là Galaxy S8.

Siri có thể giúp iPhone thông minh hơn nhờ có những tác vụ thông minh qua điều khiển giọng nói như: nhắc nhở; đọc, soạn và gửi tin nhắn; thông báo thời tiết; tìm thông tin; thiết lập một cuộc hẹn; gửi email; chỉ đường; bật một bản nhạc; tán gẫu những câu cơ bản với Siri.

Sense Companion - trợ lý ảo mới được giới thiệu trên HTC U Ultra cũng có thể thực hiện các hành động thông minh: nếu trời mưa, Sense Companion sẽ hiện thông báo trước khi người dùng ra khỏi nhà; nếu đang trong giờ nghỉ và không có ghi chú cấp bách nào, trợ lý ảo này cũng sẽ tự động tắt các chuông báo giờ cài đặt trước đó.

Thiết bị âm thanh như hệ thống loa hay các loại loa mini được các hãng công nghệ sáng tạo nhằm đáp ứng nhu cầu của người dùng về sự tiện ích của các loại loa thông minh trên thị trường.

Cụ thể có thể kể đến tại sự kiện IFA 2018, Sony đã cho ra mắt chiếc loa nhỏ gọn thuộc dòng Extra Bass mang tên XB510G. Nó được tích hợp microphone để có thể gọi Google Assistant (một trợ lý cá nhân ảo có thể tham gia trò chuyện hai chiều, điều khiển qua giọng nói), biến nó trở thành một chiếc Google Home (chiếc loa thông minh mini của Google) loại lớn, và có khả năng kết nối với các loa khác để mở rộng tầm nhạc.

Và còn nhiều hơn thế những ứng dụng của mạng nơ-ron nhân vào vào đời sống con người.

2.6. Kết luận chương 2.

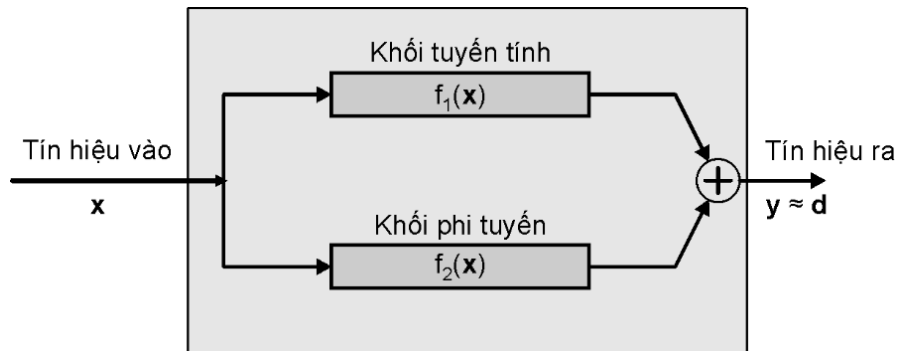
Nội dung chương này đã giới thiệu về trí tuệ nhân tạo, sự hình thành nơ-ron nhân tạo từ đó có thể phân loại, xây dựng các mạng nơ-ron bằng các hàm kích hoạt nhằm mục đích ứng dụng mạng nơ-ron để tính phụ tải điện ở chương tiếp theo.

CHƯƠNG III: MÔ HÌNH LAI VÀ ỨNG DỤNG TRONG ƯỚC LƯỢNG PHI TUYẾN

Trong chương III sẽ trình bày chi tiết về mô hình lai và làm rõ việc xây dựng khối ước lượng khối tuyến tính và phi tuyến cũng như đề xuất về việc lựa chọn các đặc tính đầu vào cho mô hình.

3.1. Mô hình lai [1, 3]

Mô hình lai hay mô hình hỗn hợp là sự kết hợp giữa hai hay nhiều mô hình, thuật toán riêng lẻ với nhau để tăng khả năng giải quyết một nội dung phức tạp một cách chính xác và tốt hơn. Sự kết hợp các mô hình phù hợp sẽ tích hợp và bổ trợ các đặc tính ưu việt của từng mô hình và góp phần làm giảm thiểu đi các đặc tính xấu không mong muốn xuất hiện ở từng mô hình riêng lẻ. Trong luận văn này, tác giả sử dụng mô hình lai kết hợp đồng thời 2 khối là khối tuyến tính và khối phi tuyến như hình 2.1.



Hình 3.1. Sơ đồ khối tổng thể của mô hình lai [1]

Khi có tín hiệu đầu vào x thì tín hiệu đầu ra tổng hợp mô hình sẽ được tính bằng:

$$d \approx y = f(\mathbf{x}) = \text{Linear}(\mathbf{x}) + \text{Nonlinear}(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x}) \quad (3.1)$$

với $f_1(\mathbf{x})$ là khối tuyến tính trong mô hình

$f_2(\mathbf{x})$ là khối phi tuyến trong mô hình

a) Khối tuyến tính

Khối tuyến tính trong mô hình được giả sử có véc-tơ tín hiệu đầu vào (\mathbf{x}), với $\mathbf{x} = [x_1, x_2, \dots, x_n]$ là các số liệu quá khứ, với các hệ số tuyến tính ai được xác định để tính toán tín hiệu đầu ra là d .

$$a_0 + \sum_{i=1}^N x_i \cdot a_i \approx d$$

Ta có:

Khi ta có tập mẫu ($p > 1$) của khối tuyến tính sẽ có dạng:

$$\begin{cases} \mathbf{x}_1 = [x_{11}, x_{12}, \dots, x_{1n}], d_1 \\ \vdots \\ \mathbf{x}_p = [x_{p1}, x_{p2}, \dots, x_{pn}], d_p \end{cases} \quad (3.2)$$

Đảm bảo điều kiện khi ta đưa \mathbf{x}_1 với các hệ số a_i ta thu được tín hiệu đầu ra của khối tuyến tính xấp xỉ bằng tín hiệu d_1 . Tương tự như với $\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_p$. Ta thể hiện tín hiệu đầu ra (d) dưới dạng hệ phương trình tuyến tính tổng quát:

$$\begin{cases} a_0 + x_{11}a_1 + x_{12}x_2 + \dots + x_{1n}a_n \approx d_1 \\ a_0 + x_{21}a_1 + x_{22}x_2 + \dots + x_{2n}a_n \approx d_2 \\ \dots \\ a_0 + x_{p1}a_1 + x_{p2}a_2 + \dots + x_{pn}a_n \approx d_p \end{cases} \quad (3.3)$$

Chuyển hóa hệ phương trình (3.3) về dạng ma trận ta sẽ có:

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix} \approx \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_p \end{bmatrix} \quad (3.4)$$

Ta đặt: $\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}; \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix}; d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_p \end{bmatrix}$

Khi đó (3.4) sẽ tương đương với: $\mathbf{X} \cdot \mathbf{a} \approx \mathbf{d}$

Bài toán đặt ra phải xác định các hệ số a_i sao cho giá trị xấp xỉ tốt nhất. Khi đó nghiệm của hệ phương trình này được xác định từ bài toán tối ưu hóa sai số.

$$\|\mathbf{d} - \mathbf{X} \cdot \mathbf{a}_{opt}\| = \min_x \|\mathbf{d} - \mathbf{X} \cdot \mathbf{a}\| \quad (3.5)$$

Luận văn này sẽ sử dụng phương pháp bình phương cực tiểu xác định giá trị sai số tối thiểu.

Phương pháp bình phương cực tiểu là phương pháp thường được sử dụng để giải xấp xỉ các hệ phương trình mà trong đó số phương trình nhiều hơn số ẩn. Cơ sở lý thuyết của phương pháp là xác định xấp xỉ nghiệm bằng cách tối thiểu hóa sai số thường được xác định bằng một hàm bậc 2.

Xét hệ phương trình tuyến tính với n ẩn và m phương trình ta có điều kiện để thực hiện phương pháp ($m > n$).

$$\{\mathbf{x} \in \mathbf{R}^n \mid \mathbf{X} \cdot \mathbf{a} = \mathbf{d}\}, \text{ với } \mathbf{X} \in \mathbf{R}^{m \times n}, \mathbf{d} \in \mathbf{R}^m$$

ở đây \mathbf{X} là ma trận $m \times n$, \mathbf{a} và \mathbf{d} các là véc-tơ.

Với bài toán ở dạng này được xem là vô nghiệm. Vậy để $\mathbf{X} \cdot \mathbf{a} = \mathbf{d}$ thì véc-tơ \mathbf{d} cần phải là tổ hợp tuyến tính của các cột trong ma trận \mathbf{X} , hay $\mathbf{d} \in \text{range}(\mathbf{X})$. Trong đó véc-tơ \mathbf{d} thuộc không gian m chiều, trong đó, số chiều lớn nhất của ma trận \mathbf{X} chỉ là n . Nên ta có, $\mathbf{d} \in \text{range}(\mathbf{X})$ được coi là trường hợp đặc biệt được gán giá trị.

Vì vậy trong trường hợp tổng quát, ta sẽ không đi tìm nghiệm của hệ phương trình, thay vào đó ta cực tiểu hóa véc-tơ sai số. Véc-tơ sai số này được tổng quát hóa dưới dạng sau:

$$\|\mathbf{d} - \mathbf{X} \cdot \mathbf{a}_{opt}\|^2 = \mathbf{r} = \min_{\mathbf{x}} \|\mathbf{d} - \mathbf{X} \cdot \mathbf{a}\|^2 \quad (3.6)$$

với $\mathbf{r} \in \mathbf{R}^m$, véc-tơ \mathbf{r} được lựa chọn sao cho có giá trị nhỏ nhất bằng cách chọn giá trị \mathbf{a} thích hợp.

Để giải bài toán bình phương cực tiểu ta có rất nhiều phương pháp tác giả xin đưa ra một số phương pháp thường được áp dụng hiện nay như: giải bài bình phương cực tiểu theo phương pháp Cholesky, theo phân tích QR và theo phân tích SVD.

- Thuật toán phân tích Cholesky phân tích ma trận \mathbf{X} xác định dương thành tích của hai ma trận.

$$\mathbf{X}' \cdot \mathbf{X} = \mathbf{R}' \cdot \mathbf{R} \quad (3.7)$$

với \mathbf{R} là ma trận tam giác trên. Phương pháp này có ưu điểm tốc độ xử lý tính toán nhanh, có khối lượng tính toán lớn, quá trình thực hiện phức tạp, yêu cầu điều kiện tích của hai ma trận $\mathbf{X}^T \cdot \mathbf{X}$ phải đủ hạng và có tính ổn định. Sai số của phương pháp này sẽ cao hơn hai phương pháp QR và SVD do sai số được làm tròn.

- Phương pháp QR: là dựa trên việc phân tích ma trận

$$\mathbf{X} = \mathbf{Q} \cdot \mathbf{R} \quad (3.8)$$

với \mathbf{Q} là ma trận trực giao, \mathbf{R} là ma trận tam giác trên. Phương pháp QR có tốc độ xử lý chậm hơn so với phương pháp Cholesky, nhưng phương pháp này có thể xử lý tốt các bài toán có ma trận \mathbf{X} không đủ hạng đồng thời tính ổn định cao hơn so với phương pháp Cholesky.

- Phương pháp SVD (Singular Value Decomposition) được hiểu như sau: với mọi ma trận \mathbf{X} có phân tích SVD là:

$$\mathbf{X} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}' \tag{3.9}$$

thì ma trận $\mathbf{X}^+ = \mathbf{U} \cdot \mathbf{S}^+ \cdot \mathbf{V}'$ là ma trận nghịch đảo của mở rộng của \mathbf{X} . Với \mathbf{S}^+ là ma trận nghịch đảo của \mathbf{S} . Phương pháp là một công cụ rất mạnh để giải quyết các bài toán khi ma trận \mathbf{X} có độ ổn định thấp. Việc sử dụng phương pháp này sẽ cho kết quả chính xác hơn và ổn định cao hơn.

Trong luận văn tác giả đề xuất sử dụng phương pháp bình phương cực tiểu để xây dựng khối tuyến tính và sử dụng phương pháp phân tích giá trị kì dị SVD để giải quyết bài toán bình phương tối thiểu.

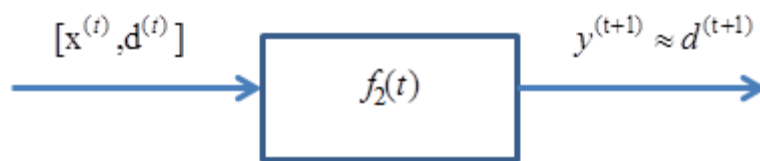
Khối tuyến tính $f_1(x)$ được đề xuất xác định đầu tiên. Các số liệu đầu vào có ảnh hưởng lớn, quyết định đến kết quả ước lượng tuyến tính sẽ được sử dụng làm số liệu đầu vào để của mô hình phi tuyến $f_2(x)$.

b) Khối phi tuyến

Giải quyết bài toán phi tuyến là việc xác định giá trị gần đúng. Việc ước lượng và xác định các thông số của mô hình phi tuyến: Quá trình xác định các thông số của mô hình phi tuyến là một quá trình tối ưu hóa sai số trên các số liệu mẫu cho trước.

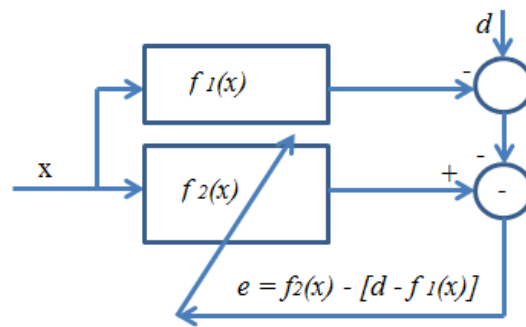
Khối phi tuyến trong mô hình đề xuất được giả sử với tín hiệu đầu vào là (x) , với $\mathbf{x} = [x_1, x_2, \dots, x_n]$ là các số liệu quá khứ và bằng tín hiệu đầu ra của khối tuyến tính là (d) với $\mathbf{d} = [d_1, d_2, \dots, d_p]$ vậy ta xây dựng khối phi tuyến $f_2(x)$ với đầu ra tín hiệu là $y^{(t+1)} \approx d^{(t+1)}$

$$f_2(x) \approx d - f_1(x) \tag{3.10}$$



Hình 3.2. Khối phi tuyến ở dạng tổng quát[1]

Khi xác định được giá trị $f_1(x)$ ta thực hiện loại thành phần này ra khỏi các số liệu đầu vào để nhằm chỉ giữ lại khối phi tuyến trong tín hiệu của đối tượng đầu vào. Tín hiệu còn lại này sẽ được dùng để huấn luyện mô hình phi tuyến.



Hình 3.3. Khối phi tuyến kết hợp với sai số đầu ra của khối tuyến tính [1]

Sai số đầu ra của khối tuyến tính được xác định như sau:

$$E = \frac{1}{2} \sum_{i=1}^p \|f_2(\mathbf{x}_i) - (d - f_1(\mathbf{x}_i))\|^2 \rightarrow \min \quad (3.11)$$

Hiện này, có rất nhiều phương pháp ước lượng khối phi tuyến được sử dụng nhưng mỗi phương pháp đều có những ưu việt và nhược điểm về độ phức tạp trong tính toán, mức độ chính xác. Việc lựa chọn ứng dụng mạng nơ-rôn vào xử lý bài toán khối phi tuyến dựa trên các ưu điểm sau: tính mềm dẻo trong tính toán, dễ thích nghi... Ngoài ra, mạng có tính ứng dụng đa dạng trong nhiều lĩnh vực. Đặc biệt là các bài toán nhận dạng mẫu, xử lý, lọc dữ liệu, điều khiển và dự báo. Ứng dụng của mạng nơ-rôn được chia thành các hướng sau:

- Xử lý ngôn ngữ như: Xử lý ngôn ngữ tự nhiên
- Nhận dạng mẫu
 - Nhận dạng đối tượng tuyến tính
 - Nhận dạng đối tượng phi tuyến
 - ...
- Xử lý tín hiệu: Điều khiển tự động...
- Lọc và phân loại dữ liệu
 - Chẩn đoán bệnh
 - Tìm kiếm
- Dự báo:
 - Dự báo các yếu tố môi trường
 - Dự báo phụ tải điện

Nhận thấy được những hướng và khả năng ứng dụng rộng rãi của mạng nơ-rôn. Tác giả đã sử dụng mạng nơ-rôn nhân tạo cụ thể là mạng nơ-rôn truyền thẳng nhiều lớp làm công cụ xử lý khối phi tuyến của mô hình đề xuất.

c) Mạng MLP

Mạng nơ-rôn nhân tạo ANN (*Artificial Neural Network*) được thiết kế dựa trên một số tính chất của mạng nơ-rôn sinh học được ứng dụng rộng rãi trong nhiều vấn đề kỹ thuật. Đây là một phương pháp tốt để giải quyết các bài toán trong xử lý tín hiệu. Mạng nơ-rôn nhân tạo có các đặc tính ưu việt sau:

- Có khả năng tạo hàm truyền phi tuyến đa biến với độ chính xác cho trước.
- Cấu trúc mạng nơ-rôn dạng truyền thẳng và cấu trúc có phản hồi dễ dàng lựa chọn cho các đối tượng có đặc tính có nhớ hoặc không có nhớ.
- Trang bị nhiều thuật toán học cho mạng giúp tăng khả năng điều chỉnh thích nghi các thông số để thích nghi với các bộ số liệu tĩnh hoặc động phù hợp hơn với yêu cầu bài toán.
- Có khả năng tổng quát hóa để hoạt động hiệu quả với các tín hiệu đầu vào mới, chưa xuất hiện trong quá trình xây dựng mô hình.

Bên cạnh những ưu điểm trên mạng còn tồn tại một số những nhược điểm sau:

- Đối với mô hình mạng lớn nhiều lớp độ phức tạp lớn dẫn tới thời gian xử lý lâu.
- Bộ trọng số thu được thường chưa tối ưu toàn cục
- Quá trình học thu được có thể xảy ra trường hợp không hoặc hội tụ chậm do vậy cần kết hợp thêm các thuật toán hỗ trợ để tăng khả năng hội tụ cho mô hình.

Mạng nơ- rôn với ưu điểm có khả năng học dưới nhiều hình thức, tổng quát hóa dữ liệu thông qua mẫu học và cấu trúc tính toán dựa trên liên kết nhiều lớp xử lý đa dạng được liên kết chặt chẽ với nhau tạo nên tính mềm dẻo cho mô hình. Bộ số liệu học mẫu của mạng nơ-rôn là những số liệu cụ thể.

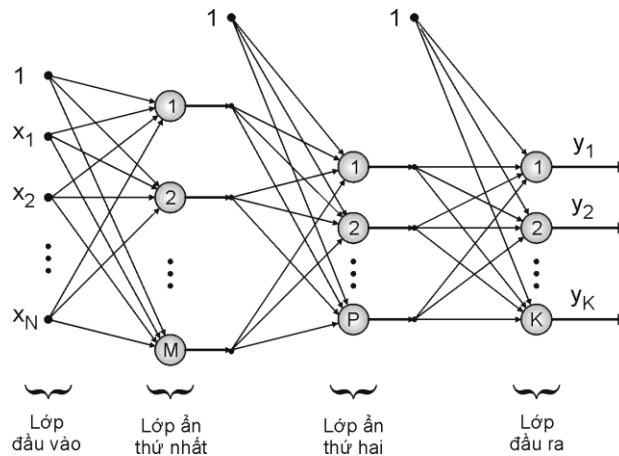
Mạng nơ-rôn MLP là một mạng truyền thẳng với các khối cơ bản là các nơ-rôn MCCulloch-Pitts. Một số yêu cầu về cấu trúc một mạng MLP:

- Một mạng MLP có n lớp với ($n \geq 2$). Trong đó gồm có một lớp các tín hiệu đầu vào (Input layer). Một lớp các tín hiệu đầu ra (Output layer) và một hoặc nhiều lớp ẩn trung gian giữa đầu vào và đầu ra (Hidden layers)
- Trên cùng một lớp sẽ không có sự liên kết giữa các nơ-rôn. Các ghép nối liên kết sẽ được tạo sao cho tín hiệu truyền thẳng từ đầu vào đến đầu ra.
- Các nơ-rôn trên cùng một lớp có cùng chung một hàm truyền đạt.

Hoạt động cơ bản của mạng MLP: Lớp đầu vào nhận tín hiệu được khuếch đại, xác định tổng trọng số kết nối W_i và gửi tới hàm kích hoạt f (*activation function*). Kết quả của hàm truyền được truyền tới các nơ-rôn của lớp đầu tiên. Tại đây các nơ-rôn sẽ nhận tín hiệu, xử lý và tiếp tục gửi kết quả đến lớp nơ-rôn tiếp theo cho đến lớp cho tín hiệu đầu ra.

Qua các báo cáo và nghiên cứu đã chứng minh được thông thường chỉ cần sử dụng đa số 1 lớp ẩn hoặc tối đa 2 lớp ẩn để mô hình hóa một hàm phi tuyến với độ

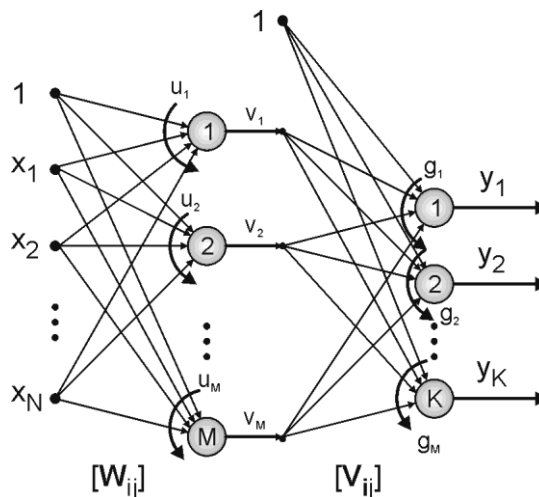
chính xác tùy chọn. Việc sử dụng quá nhiều lớp ẩn sẽ phức tạp mô hình. Trong khuôn khổ đề tài tác giả đề xuất sử dụng mạng MLP với một lớp ẩn.



Hình 3.4. Mô hình mạng nơ-rôn truyền thẳng với hai lớp ẩn [3]

Một đặc điểm cơ bản của mạng MLP một lớp ẩn:

- Mạng MLP 1 lớp ẩn gồm có: N đầu vào tín hiệu, M số nơ-rôn thuộc lớp ẩn và K số nơ-rôn lớp đầu ra.
- Trọng số ghép nối lớp đầu vào lớp ẩn được quy ước là W_{ij} với $i=1,2,\dots,M$ và $j=1,2,\dots,N$. Trọng số ghép nối lớp ẩn với lớp đầu ra là V_{ij} với $i=1,2,\dots,K$ và $j=1,2,\dots,M$. Từ đó ta xác định được ma trận trọng số các ghép nối đầu vào lớp ẩn là $\mathbf{W} \in \mathbf{R}^{M \times (N+1)}$ và ma trận ghép nối lớp ẩn và lớp đầu ra là $\mathbf{V} \in \mathbf{R}^{K \times (M+1)}$.
- Hàm kích hoạt lớp ẩn f_1 , hàm kích hoạt lớp đầu ra f_2 .



Hình 3.5. Mô hình mạng nơ-rôn truyền thẳng với một lớp ẩn [3]

Ta có véc-tơ đầu vào $x = [x_1, x_1, \dots, x_n] \in \mathbf{R}^N$ với đầu vào phân cực cố định $x_0=1$. Đầu ra của tín hiệu được xác định tuần tự theo chiều lan truyền thuận như sau:

1. Tổng các kích thích đầu vào của nơ-rôn ẩn thứ i (với $i=1,2,\dots,M$):

$$u_i = \sum_{j=0}^M x_j W_{ij} \quad (3.12)$$

2. Tổng đầu ra của nơ-rôn lớp ẩn thứ i (với $i=1,2,\dots,M$):

$$v_i = f_1(u_i) \quad (3.13)$$

Ta coi đầu vào phân cực cho các nơ-rôn lớp ra là $v_0 = 1$ để thuận cho việc biểu diễn công thức.

3. Tổng kích thích đầu vào của nơ-rôn đầu ra thứ i (với $i=1,2,\dots,K$):

$$g_i = \sum_{j=0}^M v_j V_{ij} \quad (3.14)$$

4. Tính toán đầu ra thứ i của mạng (với $i=1,2,\dots,K$)

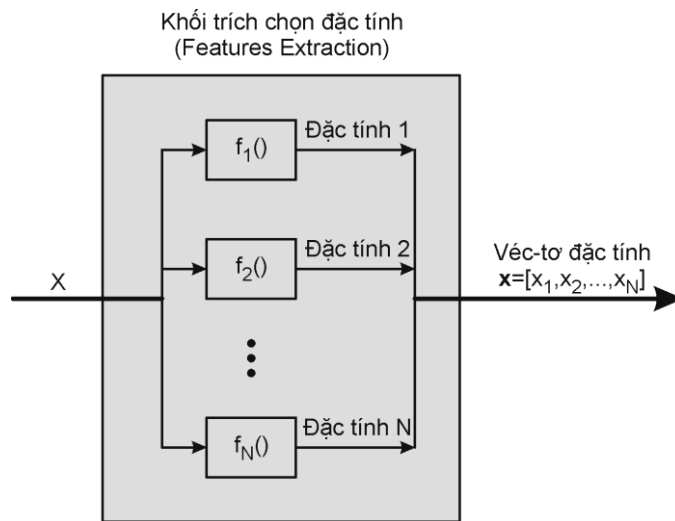
Ta xác định được hàm truyền đạt của mạng MLP là một hàm phi tuyến:

$$y_i = f_2(g_i) = f_2\left(\sum_{j=0}^M v_j V_{ij}\right) = f_2\left(\sum_{j=0}^M f_1(u_j) V_{ij}\right) = f_2\left(\sum_{j=0}^M \left[f_1\left(\sum_{k=0}^N x_k W_{jk}\right) V_{ij} \right]\right) \quad (3.15)$$

3.2. Chọn đặc tính đầu vào của mô hình dự báo [1, 3]

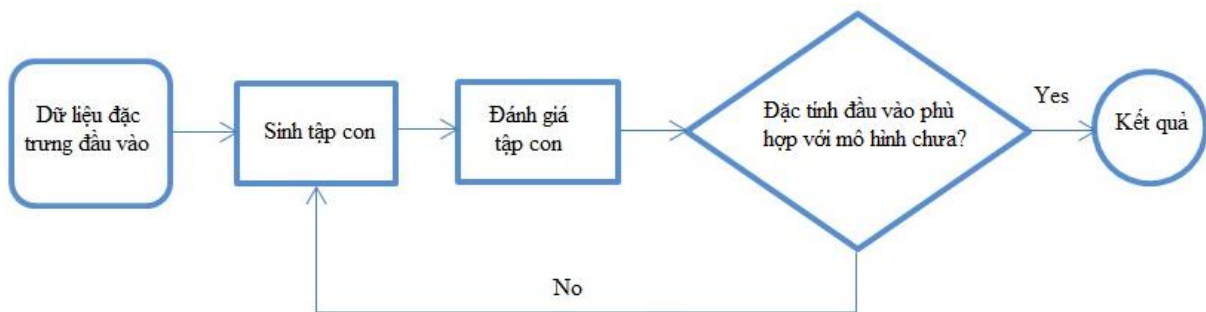
Trích chọn đặc tính đầu vào là một bước rất quan trọng trong quá trình nhận dạng, điều khiển tín hiệu hay dự báo... sau khi được thu thập bằng cách loại bỏ những thông tin đặc trưng có rất ít hoặc không có thông tin dự đoán. Các giá trị được chọn lọc hỗ trợ rất lớn cho quá trình xử lý, nhận dạng tín hiệu. Trích chọn tín hiệu đầu vào giúp làm giảm kích cỡ của không gian đặc trưng, loại bỏ các tín hiệu nhiễu, thừa từ quá trình đo mang lại. Trong đa số các phương pháp chọn đặc tính đầu vào tín hiệu để đảm bảo tăng độ chính xác và làm giảm tính phức tạp của các thuật toán phải đảm bảo yêu cầu sau:

- Các véc-tơ đặc trưng của đối tượng được chọn cần phải đảm bảo nhỏ và gọn hơn nhiều so với đối tượng gốc nhưng vẫn đảm bảo được thông tin.
- Thời gian tính toán các đặc trưng phải phù hợp với thực tế áp dụng, không quá phức tạp để giảm hiệu quả hoạt động và tăng thời gian đáp ứng của hệ thống.
- Tập thông tin được chọn có thể phân biệt tương đối rõ ràng với các tập thông tin khác.
- Tín hiệu thu thập được từ các thiết bị đo phải có giá trị tương đồng, dải giá trị phân tán ít, gọn cụm.



Hình 3.6. Cấu trúc khối trích chọn đặc tính [1]

Quá trình trích chọn đặc trưng bao gồm các bước cơ bản sau: Bộ dữ liệu đặc trưng đầu vào, sinh tập con, đánh giá tập con, điều kiện dừng trích chọn và kết quả. Các bước trên được thể hiện theo hình 3.7.

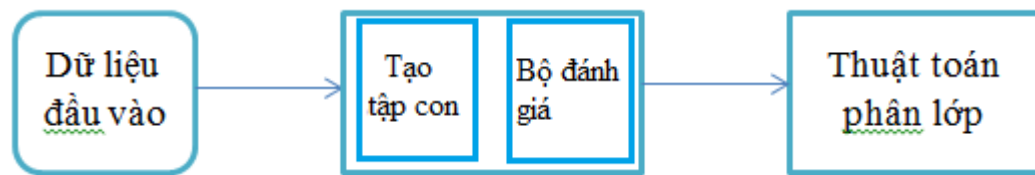


Hình 3.7. Cấu trúc tổng quát các bước thực hiện trích chọn đặc tính

Các phương pháp chọn lọc đặc tính được tổng quát hóa bằng các bước sau: lựa chọn trọng số của các đặc tính. Trong đó, đặc tính có trọng số tốt nhất sẽ được chọn để áp dụng vào bài toán của mô hình. Phương pháp trích chọn thuộc tính phụ thuộc vào cấu trúc của bộ dữ liệu và thuật toán được dùng để sử dụng để khởi tạo mô hình. Các kỹ thuật chọn đặc tính thường được sử dụng trong các bài toán biến đổi phi tuyến. Hiện nay, có rất nhiều phương pháp sử dụng để trích chọn đặc tính ta có hai hướng tiếp cận điển hình là: Filter và Wrapper.

a) Mô hình theo hướng Filter:

Hướng tiếp cận Filter là việc đánh giá từng đặc tính bằng một số tiêu chuẩn, sau đó chọn ra tập con các giá trị có được đánh giá cao nhất. Hướng tiếp cận này, thực hiện việc trích chọn đặc tính trước sau đó mới sử dụng thuật toán để phân lớp.



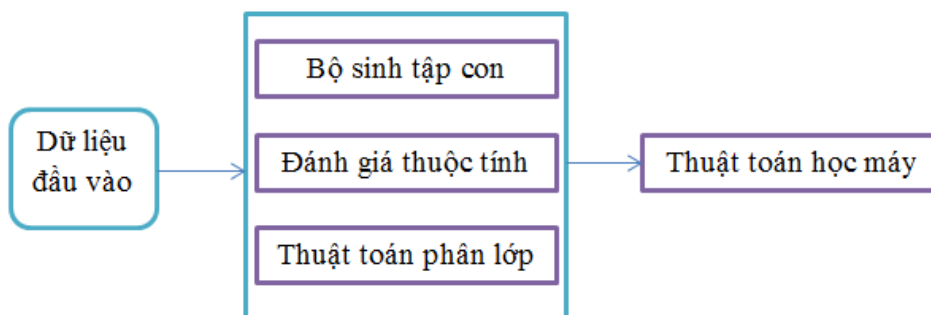
Hình 3.8. Cấu trúc mô hình theo hướng Filter

- Ưu điểm của hướng tiếp cận này là: dễ dàng nhận diện và thời gian thực hiện ít hơn mô hình Wrapper, không có quá trình xử lý học máy trong việc lựa chọn đặc tính.
- Nhược điểm: Khả năng tạo ra các tập con đặc trưng ở mức thấp. Các hàm đánh giá ở các phạm vi nhất định dẫn tới kết quả thu được giảm độ chính xác của việc học trong giai đoạn sau.

b) Mô hình theo hướng Wrapper

Hướng tiếp cận Wrapper sử dụng một thuật toán tìm kiếm để đánh giá một nhóm (tập con) các thuộc tính thay vì đánh giá từng thuộc tính như Filter. Thông qua việc xác định tỷ lệ thực hiện đánh giá độ tốt, xấu của tập con đặc trưng.

- Ưu điểm: hiệu suất học cao hơn hướng Filter.
- Nhược điểm: Thời gian học lớn, kết quả đánh giá phụ thuộc vào thuật toán học. ít được áp dụng hơn hướng tiếp cận Filter.



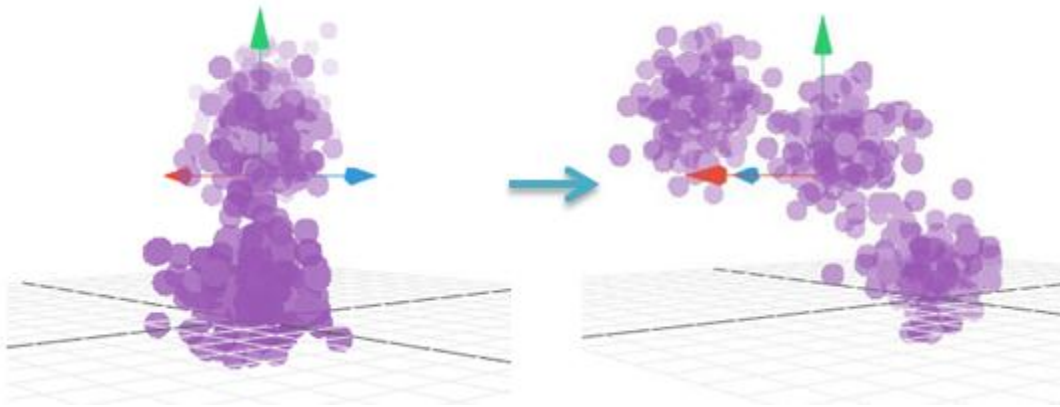
Hình 3.9. Cấu trúc mô hình theo hướng Wrapper

Sau đây là một số phương pháp trích chọn đặc tính thường được sử dụng trong các bài toán xây dựng mô hình dự báo hiện nay:

3.2.1. Phương pháp phân tích thành phần chính – PCA

Một trong những phương pháp phân tích dữ liệu nhiều biến đơn giản nhất hiện nay không thể không nhắc tới phương pháp phân tích thành phần chính PCA (Principal Component Analysis). Đây là một phương pháp cho phép người dùng đưa dữ liệu từ không gian có số chiều lớn không thể xây dựng mô hình dựa trên bộ số liệu đó về không gian có số chiều nhỏ hơn. Hơn thế nữa PCA còn có một số đặc tính ưu việt sau:

- PCA xây dựng một không gian mới ít chiều hơn, nhưng vẫn đảm bảo khả năng diễn dịch dữ liệu tốt tương đương không gian cũ. Hoặc có thể nói PCA đảm bảo độ biến thiên của dữ liệu trên chiều mới xây dựng.
- Trục tọa độ của không gian mới chính là tổ hợp tuyến tính của không gian cũ.
- Các liên kết tiềm ẩn của bộ dữ liệu có thể được chọn lọc, phân tác ra ở không gian mới tốt hơn và dễ dàng hơn không gian cũ.



Hình 3.10. Không gian mới được tạo ra qua phép biến đổi PCA [1]

Bên cạnh những đặc tính tốt PCA cũng mang một số khuyết điểm sau:

- PCA dựa toàn bộ trên các biến đổi tuyến tính.
- Nhạy cảm với các điểm riêng biệt và vô cùng (ourlier/extreme).
- Chỉ sử dụng trong môi trường dữ liệu số (numeric).
- Xây dựng thuật toán tương đối phức tạp.

3.2.2. Phương pháp phân tích sự khác biệt tuyến tính LDA

Phương pháp phân tích sự khác biệt tuyến tính LDA (*Linear Discriminat Analysis*) thường được sử dụng để giải quyết các bài toán phân lớp đối tượng đầu vào trong một hay nhiều lớp đã được xác định đặc trưng. Với hai lớp phân tích khác biệt trở lên phương pháp này sẽ được gọi là phân tích đa khác biệt (*Multiple Discriminat Analysis - MDA*).

Mục đích của phương pháp là phân loại đối tượng (người, vật,...) vào một hoặc nhiều lớp được xác định trước dựa trên các đặc trưng dùng để mô tả đối tượng. Hay có thể nói từ tập các thuộc tính mô tả đối tượng ban đầu tìm ra các thuộc tính tốt để biểu diễn cho đối tượng làm giảm sai số biểu diễn đối tượng.

Một số ưu nhược điểm của phương pháp:

- Ưu điểm: phương sai thấp, làm việc tốt khi số mẫu lớn. Đạt hiệu quả cao hơn khi đưa ra được mô hình phù hợp.
- Nhược điểm: Khi số lượng mẫu huấn luyện ít hơn so với kích thước không gian đặc trưng sẽ gặp khó khăn trong việc huấn luyện.

3.2.3. Phương pháp phân tích giá trị kỳ dị (SVD)

Như những phân tích sơ bộ ở (phần 3.1 của Chương III). SVD là một trong những phương pháp tính toán mạnh trong đại số tuyến tính, được ứng dụng rất nhiều trong việc phân tích, xử lý tín hiệu và thống kê.

Một số đặc điểm ưu việt của phương pháp

- Giải thuật SVD cho kết quả có sai số nhỏ
- Giảm số chiều mô hình.
- Hơn thế nữa phương pháp còn hỗ trợ trong việc xác định giá trị cực tiểu toàn cục trong hệ đa phương trình tuyến tính.
- Phân tích tốt mối quan hệ khác nhau các phần tử trong bộ dữ liệu gốc
- Giải quyết hiệu quả các bài toán liên quan đến tính toán ma trận.
- SVD dùng để xác định và sắp xếp các chiều véc-tơ sao cho các dữ liệu có sự chênh lệch nhiều nhất.
- Đối với nội dung luận văn phương pháp SVD cho phép xác định các sử dụng SVD để xác định hệ số a trong mô hình dự báo tuyến tính và để phân tích thành phần đầu vào cho mô hình dự báo.

Từ những phân tích khả năng, ưu nhược điểm của các phương pháp trích chọn đặc tính, trong luận văn này tác giả sử dụng phương pháp SVD để chọn đặc tính cho mô hình dự báo.

3.3. Ứng dụng giải thuật SVD xây dựng khối tuyến tính [3]

Trong đại số tuyến tính, giải thuật phân tích giá trị kỳ dị SVD là một trong những phương pháp tính toán mạnh được ứng dụng rất nhiều trong việc phân tích, xử lý tín hiệu và thống kê.

Ta có dạng tổng quát của một hệ phương trình tuyến tính như sau:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1k}x_k = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2k}x_k = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nk}x_k = b_n \end{cases} \quad (3.16)$$

Hệ phương trình tuyến tính trên có thể được viết dưới dạng phương trình ma trận:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \quad (3.17)$$

với \mathbf{A} là ma trận chứa các hệ số a_{ij} (i là thành phần cột, còn j là thành phần hàng của ma trận \mathbf{A} , \mathbf{x} là véc-tơ có chứa các biến x_j , \mathbf{b} là véc-tơ có chứa các hằng số b_i . Qua đó phương trình (2.2) được viết dưới dạng ma trận như sau:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nk} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (3.18)$$

Xét bài toán hệ phương trình tuyến tính đặt ra của mô hình dự báo đề xuất trong luận văn ta có số biến k nhỏ hơn so với hạng r của ma trận. Vậy ta sẽ có hệ phương trình tuyến tính đa ẩn của mô hình đề xuất giải bằng phương pháp phân tích giá trị kỳ dị SVD.

Giải thuật SVD dựa vào định lý định lý sau của đại số tuyến tính: Cho ma trận \mathbf{A} có kích thước $m \times n$ gồm m hàng và n cột ($\mathbf{A} \in \mathbf{R}^{m \times n}$) với điều kiện $m \geq n$. Ta có: $\lambda \geq 0$ là giá trị riêng của ma trận $\mathbf{A}^T \mathbf{A}$ và véc-tơ riêng tương ứng.

Giá trị kỳ dị của ma trận \mathbf{A} được xác định như sau:

$$\sigma = \sqrt{\lambda_i} = \|\mathbf{A}v_i\| \quad (3.19)$$

với: λ_i là giá trị riêng thứ i , v_i là giá trị véc-tơ riêng tương ứng thứ i .

Từ điều kiện $\lambda_i \geq 0$ vậy ta sẽ có với mọi $\exists r: \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0, \lambda_i = 0$ và $i > r$ ta suy ra: $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ và $\sigma_{r+1} = \sigma_{r+1} = \dots = \sigma_n = 0$.

Ma trận $\mathbf{A}_{m \times n}$ sẽ tồn tại 2 ma trận trực giao $\mathbf{U} \in \mathbf{R}^{m \times n}$ và $\mathbf{V} \in \mathbf{R}^{m \times n}$ sao cho

$$\mathbf{U}^T \cdot \mathbf{A} \cdot \mathbf{V} = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbf{R}^{m \times n} \quad (3.20)$$

Các cột của ma trận \mathbf{U} và \mathbf{V} là các véc-tơ kỳ dị trái và phải tương ứng. Với

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T \quad (3.21)$$

Với $\mathbf{S} \in \mathbf{R}^{m \times n}$ là ma trận đường chéo $\text{diag}(\sigma_1, \dots, \sigma_r)$ có dạng sau:

$$\Rightarrow \mathbf{S}_{m \times n} = \begin{bmatrix} \mathbf{D}_{r \times r} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix} \text{ với } \mathbf{D}_{r \times r} = \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_r \end{bmatrix}$$

Ma trận \mathbf{V} xác định bằng véc-tơ riêng của $\mathbf{A}^T \mathbf{A}$: $\mathbf{V} = [v_1, v_2, \dots, v_n]$

Ma trận \mathbf{U} được xác định như sau.

Các σ_i có giá trị $i \in (1, r)$. Ta xác định giá trị u_i như sau:

$$u_i = \frac{1}{\sigma_i} \mathbf{A} \cdot v_i \tag{3.22}$$

Ta xác định được các giá trị ma trận $\mathbf{U} = [u_1, u_2, \dots, u_m]$ với $(m > r)$.

Ta sẽ giải quyết bài toán tối ưu hóa sai số:

$$x = \bar{x} = \mathbf{A}^+ \cdot b \tag{3.23}$$

với \mathbf{A}^+ là ma trận giả nghịch đảo của $\mathbf{A}^{m \times n}$ thì ta có $\|\mathbf{A} \cdot x - b\|$ đạt giá trị nhỏ nhất và $\|\bar{x}\|$ có độ lớn nhỏ nhất. Ma trận $\mathbf{A}^{m \times n}$ với $m > n$ ta xác định được ma trận giả nghịch đảo \mathbf{A}^+ . Theo phân tích SVD của ma trận A theo công thức (3.20) ta sẽ có:

$$\mathbf{A}^+ = \mathbf{V} \cdot \mathbf{S}^+ \cdot \mathbf{U}^T \tag{3.24}$$

với $\mathbf{S}^+ = \text{diag}\left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_r}\right) \in \mathbf{R}^{n \times m}$ là ma trận giả nghịch đảo của ma trận S.

Để xác định được giá trị $\|\bar{x}\|$ ta thực hiện theo 5 bước cơ bản dưới đây:

- Bước 1: Tìm các giá trị riêng λ_i và xác định các giá trị kỳ dị σ_i của tích 2 ma trận $\mathbf{A}^T \mathbf{A}$ bằng cách giải phương trình $\det(\mathbf{A} - \lambda \cdot \mathbf{I}) = 0$
- Bước 2: Thực hiện giải phương trình $(\mathbf{A}^T \mathbf{A} - \lambda_i \cdot \mathbf{I}) \cdot \mathbf{x} = 0$ tìm các véc-tơ riêng vì từ đó ta xác định được ma trận V và VT
- Bước 3: Xác định các véc-tơ ui ma trận U qua công thức số (2.21)
- Bước 4: Xác định được dạng khai triển của ma trận A theo SVD: $\mathbf{A} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T$ từ đó xác định được ma trận giả nghịch đảo \mathbf{A}^+ .
- Bước 5: Tìm giá trị nghiệm xấp xỉ tốt nhất của phương trình:

$$\bar{x} = \mathbf{A}^+ \cdot b \tag{3.25}$$

Ví dụ 2.1: giải hệ phương trình với 3 phương trình và 2 ẩn số:

$$\begin{cases} x + 2y = 6 \\ -4x + 2y = 1 \text{ (I)} \\ 3x - y = 1 \end{cases}$$

Lời giải:

Hệ phương trình (I) tương đương với dạng phương trình:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \text{ với } \mathbf{A} = \begin{bmatrix} 1 & 2 \\ -4 & 2 \\ 3 & -1 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 6 \\ 1 \\ 1 \end{bmatrix}$$

Giá trị riêng λ của $\mathbf{A}^T \mathbf{A}$ là: $\begin{cases} \lambda_1 = 29,88 \\ \lambda_2 = 5,12 \end{cases}$

Ta có giá trị kỳ dị của ma trận \mathbf{A} là: $\sigma_1 = \sqrt{\lambda_1} = \sqrt{29,88}; \sigma_2 = \sqrt{\lambda_2} = \sqrt{5,12}$

$$\mathbf{S}_{2 \times 3} = \begin{bmatrix} 29,88 & 0 \\ 0 & 5,12 \end{bmatrix} \Rightarrow \mathbf{V}^T = \begin{bmatrix} -0,40 & -0,92 \\ -0,92 & -0,40 \end{bmatrix}; \quad \mathbf{U} = \begin{bmatrix} -0,41 & -0,76 \\ -0,04 & 1,24 \\ -0,05 & -1,04 \end{bmatrix}$$

Vậy ta có ma trận

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T = \begin{bmatrix} -0,41 & -0,76 \\ -0,04 & 1,24 \\ -0,05 & -1,04 \end{bmatrix} \begin{bmatrix} 29,88 & 0 \\ 0 & 5,12 \end{bmatrix} \begin{bmatrix} -0,40 & -0,92 \\ -0,92 & -0,40 \end{bmatrix}$$

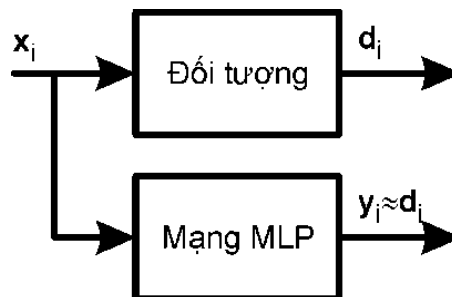
Ma trận giả nghịch đảo \mathbf{A}^+ được viết như sau:

$$\mathbf{A}^+ = \mathbf{V} \cdot \mathbf{S}^+ \cdot \mathbf{U}^T = \begin{bmatrix} -0,40 & -0,92 \\ -0,92 & -0,40 \end{bmatrix} \begin{bmatrix} \frac{1}{29,88} & 0 \\ 0 & \frac{1}{5,12} \end{bmatrix} \begin{bmatrix} -0,41 & -0,04 & -0,05 \\ -0,76 & 1,24 & -1,04 \end{bmatrix}$$

$$\mathbf{A}^+ = \begin{bmatrix} 0,14 & -0,22 & 0,19 \\ 0,07 & -0,1 & 0,083 \end{bmatrix} \Rightarrow \bar{\mathbf{x}} = \begin{bmatrix} 0,82 \\ 0,42 \end{bmatrix}$$

3.4. Ứng dụng mạng MLP xây dựng khối phi tuyến [1, 3]

Trước hết ta xét bộ số liệu mẫu của mạng. Với mạng MLP bộ số liệu mẫu được cho dưới dạng véc-tơ đầu vào và véc-tơ đầu ra tương ứng $\{\mathbf{x}_i, \mathbf{d}_i\}$ với $i=1,2,\dots,p, \mathbf{x}_i \in \mathbf{R}^N; \mathbf{d}_i \in \mathbf{R}^K$.



Hình 3.11. Xác định mạng MLP xấp xỉ hàm truyền của đối tượng [1]

Khi cho véc-tơ đầu vào \mathbf{x}_i thì đáp ứng đầu ra của mạng sẽ xấp xỉ giá trị đích cho trước với giá trị sai số tiến tới một giá trị cực tiểu nào đó của hàm mục tiêu được thể hiện qua công thức (2.12) với $\|\cdot\|$ là chuẩn khoảng cách.

$$E = \frac{1}{2} \sum_{i=1}^p \|MLP(\mathbf{x}_i) - \mathbf{d}_i\|^2 \rightarrow \min \tag{3.26}$$

Một số chuẩn khoảng cách được sử dụng trong mạng nơ-rôn như sau:

Công thức Ô-clit:	$\mathbf{x}, \mathbf{c} \in \mathbb{R}^N : d(\mathbf{x}, \mathbf{c}) = \ \mathbf{x} - \mathbf{c}\ = \sqrt{\sum_{i=1}^N (x_i - c_i)^2}$	(3.27)
Tích vô hướng	$d(\mathbf{x}, \mathbf{c}) = 1 - \mathbf{x} \cdot \mathbf{c} = 1 - \ \mathbf{x}\ \cdot \ \mathbf{c}\ \cdot \cos(\mathbf{x}, \mathbf{c})$	(3.28)
Khoảng cách Manhattan	$d(\mathbf{x}, \mathbf{c}) = \sum_{i=1}^N x_i - c_i $	(3.29)
Khoảng cách Chebyshev	$d(\mathbf{x}, \mathbf{c}) = \max_{i=1 \rightarrow N} x_i - c_i $	(3.30)
Khoảng cách Minkowshi	$d(\mathbf{x}, \mathbf{c}) = \sqrt[m]{\sum_{i=1}^N x_i - c_i ^m}$	(3.31)

Để xác định mạng MLP để tái tạo lại bộ số liệu bao gồm xác định các thông số cấu trúc và trọng số ghép nối ta thực hiện theo các bước sau:

- Xác định số kênh đầu vào N bằng kích thước của các véc-tơ xi, xác định số kênh đầu ra bằng kích thước của véc-tơ di.
- Số lớp ẩn được sử dụng: trong luận văn xét trường hợp mạng có 1 lớp ẩn hạn chế độ phức tạp và bậc phi tuyến cao.

	Hàm truyền	Mục đích sử dụng
Lớp đầu vào	tansig: $f(u) = \frac{1 - e^{-au}}{1 + e^{-au}}$	Tạo ra các tín hiệu trong đoạn [-1,1]
Lớp ẩn	tansig: $f(u) = \frac{1 - e^{-au}}{1 + e^{-au}}$	
Lớp đầu ra	tansig: $f(u) = \frac{1 - e^{-au}}{1 + e^{-au}}$	
	logsig: $f(u) = \frac{1}{1 + e^{-au}}$	Tạo ra các tín hiệu trong đoạn [0,1]
	Hàm tuyến tính	Tạo tín hiệu đầu ra ở

		ngoài đoạn [-1,1]
--	--	-------------------

- Trọng số ghép nối giữa các lớp của mạng MLP: ta xét mạng với 1 lớp ẩn vậy ta xác định 2 ma trận trọng số ghép nối \mathbf{W} (giữa lớp đầu vào và lớp ẩn) và \mathbf{V} (giữa lớp ẩn và lớp đầu ra). Việc điều chỉnh các trọng số ghép nối sao cho phù hợp và đưa ra được đáp ứng đầu ra của mạng gần giống với giá trị mong muốn. Ta có thể sử dụng một số thuật toán sau:

a) Thuật toán bước giảm cực đại:

Với thuật toán bước giảm cực đại, các giá trị trọng số được khởi tạo bằng giá trị ngẫu nhiên: $[\mathbf{W}] = [\mathbf{W}]^{(0)}; [\mathbf{V}] = [\mathbf{V}]^{(0)}$. Ta xác định điểm cực trị của hàm mục tiêu E qua công thức thay đổi các trọng số trong hai ma trận \mathbf{W} và \mathbf{V} như sau:

$$\begin{cases} W_{\alpha\beta}^{(t+1)} = W_{\alpha\beta}^{(t)} - \eta \frac{\partial E}{\partial W_{\alpha\beta}} \\ V_{\alpha\beta}^{(t+1)} = V_{\alpha\beta}^{(t)} - \eta \frac{\partial E}{\partial V_{\alpha\beta}} \end{cases} \quad (3.32)$$

Độ lệch giữa véc-tơ đầu ra y_i của mạng và véc-tơ giá trị đích d_i được tính bằng các công thức tính chuẩn khoảng cách đã nêu ở trên.

Theo công thức (3.12) đến (3.15) và (3.26) thực hiện biến đổi ta được phương trình sau:

$$\frac{\partial E}{\partial V_{\alpha\beta}} = \sum_{i=1}^p \sum_{j=1}^K (y_{ij} - d_{ij}) \frac{\partial y_{ij}}{\partial V_{\alpha\beta}} \quad (3.33)$$

Với
$$\frac{\partial y_{ij}}{\partial V_{\alpha\beta}} = f_2'(g_{ij}) \frac{\partial g_{ij}}{\partial V_{\alpha\beta}} = f_2'(g_{ij}) \sum_{k=0}^N v_{ik} \frac{\partial V_{jk}}{\partial V_{\alpha\beta}} \quad (3.34)$$

Ta có $j = \alpha$ và $k = \beta$ thì $\frac{\partial y_{ij}}{\partial V_{\alpha\beta}} = 1; \frac{\partial y_{ij}}{\partial V_{\alpha\beta}} = 0$ trong các trường hợp còn lại.

Vậy công thức (3.34) sẽ được viết lại như sau:

$$\frac{\partial E}{\partial V_{\alpha\beta}} = \sum_{i=1}^p (y_{i\alpha} - d_{i\alpha}) f_2'(g_{i\alpha}) v_{i\beta} \quad (3.35)$$

Phương trình (2.33) được viết lại như sau:

$$\frac{\partial E}{\partial W_{\alpha\beta}} = \sum_{i=1}^p \sum_{j=1}^K (y_{ij} - d_{ij}) f_2'(g_{ij}) V_{j\alpha} f_1'(u_{i\alpha}) x_{i\beta} \quad (3.36)$$

Từ công thức (2.36), ta tiến hành điều chỉnh các giá trị của hai ma trận \mathbf{W} và \mathbf{V} để xác định cực tiểu của hàm sai số.

b) Thuật toán Levenberg – Marquadt

Trong bài toán tuyến tính của mô hình được đề xuất tác giả sử dụng phương pháp bình phương cực tiểu để tối ưu hóa sai số mô hình. Trong trường hợp khối phi tuyến bài toán có thể giải bằng thuật toán Levenberg – Marquadt.

Thuật toán Levenberg – Marquadt có khả năng cải thiện tốc độ hội tụ hay tốc độ học của mạng nơ-rôn. Ta có công thức điều chỉnh các trọng số của mạng như sau:

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \left[\mathbf{H}(\mathbf{W}^{(t)}) \right]^{-1} \mathbf{g}(\mathbf{W}^{(t)}) \quad (3.37)$$

$$\mathbf{V}^{(t+1)} = \mathbf{V}^{(t)} - \eta \left[\mathbf{H}(\mathbf{V}^{(t)}) \right]^{-1} \mathbf{g}(\mathbf{V}^{(t)}) \quad (3.38)$$

trong đó:

- $\mathbf{W}^{(t)}, \mathbf{V}^{(t)}$ là các ma trận chứa trọng số ghép nối $\mathbf{W}_{ij}, \mathbf{V}_{ij}$ tại bước lặp thứ (t) trong quá trình điều chỉnh thích nghi.
- $\mathbf{g}(\mathbf{W})$ và $\mathbf{g}(\mathbf{V})$ là các véc-tơ gradient của hàm sai số theo từng trọng số trong ma trận.
- $\mathbf{H}(\mathbf{W})$ và $\mathbf{H}(\mathbf{V})$ là các ma trận gradient bậc 2 của hàm sai số theo từng cặp trọng số từ các ma trận W và V tương ứng.

Để xây dựng một mạng nơ-rôn đáp ứng đủ yêu cầu công nghệ của bài toán phi tuyến cần đầy đủ các bước cơ bản sau đây:

1. Mẫu số liệu chia làm hai bộ: bộ số liệu học và bộ số liệu kiểm tra. Để đơn giản cho quá trình chia số liệu ta có thể thực hiện chia xấp xỉ ngẫu nhiên 2 bộ số liệu học và kiểm tra. Nhưng đối với các trường hợp phương pháp chia ngẫu nhiên không cho kết quả cao, ta thực hiện chia bộ số liệu theo một trong các phương pháp sau: phương pháp Cross-Validation, Bootstrapping...

2. Lựa chọn thuật toán phù hợp nhất để xây mạng cho tập số liệu học. Trong quá trình xây dựng mạng ta cần giải quyết 2 vấn đề sau:

- Xây dựng hàm ánh xạ phi tuyến giữa không gian đầu vào và đầu ra sao cho hàm ánh xạ bao quát đầy đủ các không gian của đầu vào.
- Mạng có tính tổng quát hóa để đảm bảo tránh trường hợp học quá khớp và học ít.

1. Kiểm tra mạng xây dựng ở bước 2 với bộ số liệu kiểm tra. Nếu kết quả kiểm tra đạt yêu cầu thì kết thúc. Nếu không đạt thực hiện bước 4.

2. Thực hiện xây dựng mạng mới cho phù hợp với bộ số liệu. Hoặc xây dựng lại bộ số liệu cho phù hợp.

Trên đây là trình tự các bước để xây dựng một mô hình mạng phi tuyến cơ bản để đảm bảo tính chính xác của mạng ta cần lưu ý tới các yếu tố sau:

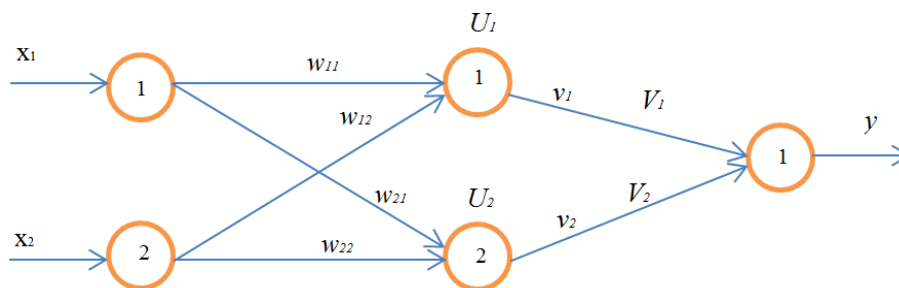
- Số đầu vào và đầu ra của mạng;
- Số lượng cặp mẫu trong bộ số liệu học;

- Lượng nhiễu có trong tín hiệu đầu vào;
- Độ phức tạp của hàm ánh xạ;
- Cấu trúc của mạng MLP đang xây dựng;
- Dạng của hàm truyền đạt;
- Thuật toán học;
- Các phương pháp hỗ trợ quá trình học.

Ví dụ 2.3: Xét bài toán xét bộ số liệu $x_1 = 0,5; x_2 = 0,2, d = 0,7$

- Xét mạng hai đầu vào, một lớp ẩn và một lớp đầu ra;
- Số lượng nơ-rôn lớp đầu vào là 2;
- Số lượng nơ-rôn trong lớp ẩn là 2;
- Số nơ-rôn đầu ra là 1.

Mạng sẽ có cấu trúc 2-2-1 như sau:



Hàm truyền sử dụng là hàm logsig: $f(x) = \frac{1}{1 + e^{-tx}}$

Thực hiện chọn bộ trọng số ngẫu nhiên như sau:

$$w_{11} = 0,1; w_{12} = 0,2; w_{21} = 0,3; w_{22} = 0,4;$$

$$v_1 = 0,5; v_2 = 0,6$$

Ta có hàm sai số như sau:

$$E = \frac{1}{2}(y - d)^2$$

Xác định tổng đầu vào: $\begin{cases} u_1 = 0,11 \\ u_2 = 0,27 \end{cases}$

Xác định đầu ra các nơ-rôn:

$$\begin{cases} v_1 = 0,528 \\ v_2 = 0,567 \end{cases}$$

$$\rightarrow g = v_1 V_1 + v_2 V_2 = 0,604$$

$$\rightarrow y = \text{logsig}(g) = \frac{1}{1 + e^{-g}} = 0,647$$

Ta tính được sai số như sau:

$$E = \frac{1}{2}(y - d)^2 = \frac{1}{2}(0,647 - 0,7)^2 = 0,014$$

Thực hiện điều chỉnh sai số theo phương pháp bước giảm cực đại công thức số (3.32) với $\forall \alpha, \beta$.

$$\begin{cases} W_{\alpha\beta}^{(t+1)} &= W_{\alpha\beta}^{(t)} - \eta \frac{\partial E}{\partial W_{\alpha\beta}} \\ V_{\alpha}^{(t+1)} &= V_{\alpha}^{(t)} - \eta \frac{\partial E}{\partial V_{\alpha}} \end{cases}$$

Thực hiện đạo hàm 2 vế của phương trình (I) theo V_{α}

$$\begin{aligned} \frac{\partial E}{\partial V_{\alpha}} &= (y - d) \frac{\partial y}{\partial V_{\alpha}} \\ \rightarrow \frac{\partial E}{\partial V_{\alpha}} &= (y - d) y (1 - y) V_{\alpha} \end{aligned}$$

Ta có
$$\frac{\partial y}{\partial g} = \frac{\partial \text{logsig}(g)}{\partial g} = \frac{(-1)e^{-g}}{(1 + e^{-g})^2}$$

Đạo hàm phương trình (I) theo $w_{\alpha\beta}$ ta có:

$$\frac{\partial E}{\partial W_{\alpha\beta}} = (y - d) \frac{\partial y}{\partial W_{\alpha\beta}}$$

Thực hiện thay $y = \text{logsig}(g)$ vào phương trình (II) ta có:

$$\begin{aligned} \frac{\partial E}{\partial W_{\alpha\beta}} &= (y - d) \cdot \frac{\partial \text{logsig}(g)}{\partial g} \cdot \frac{\partial g}{\partial \alpha\beta} \\ \Leftrightarrow \frac{\partial E}{\partial W_{\alpha\beta}} &= (y - d) y (1 - y) V_{\alpha} \cdot \frac{\partial V_{\alpha}}{\partial \alpha\beta} \end{aligned}$$

Với $v_1 = \text{logsig}(u_1), v_2 = \text{logsig}(u_2)$ ta có:

$$\frac{\partial E}{\partial W_{\alpha\beta}} = (y - d) y (1 - y) V_{\alpha} \cdot \text{logsig}(u_{\alpha}) (1 - \text{logsig}(u_{\alpha})) \frac{\partial V_{\alpha}}{\partial \alpha\beta}$$

Từ phương trình (I) đạo hàm hai vế theo V_i ta có:

$$\frac{\partial E}{\partial V_1} = (y - d) \frac{\partial y}{\partial V_1}$$

$$\Leftrightarrow \frac{\partial E}{\partial V_1} = (y - d) g(1 - g) \frac{\partial g}{\partial V_1}$$

Thay các giá trị $y, d, g, \partial g, \partial v_1$ ta có:

$$\frac{\partial E}{\partial V_1} = (0,647 - 0,7)0,604(1 - 0,604)0,528 = -0,669 \cdot 10^{-2}$$

Hoàn toàn tương tự ta xác định được giá trị đạo hàm của sai số E theo v_2 :

$$\frac{\partial E}{\partial V_2} = -7,19 \cdot 10^{-3}$$

Từ công thức (2.32) ta có:

$$V_i^{(1)} = V_i^{(0)} - \eta \frac{\partial E}{\partial V_i}$$

Chọn hệ số $\eta = 14,947 \approx 15$

$$\Rightarrow V_1 = 0,5 - 15(-0,669 \cdot 10^{-2}) \approx 0,6$$

$$\Rightarrow V_2 = 0,6 - 15(-7,19 \cdot 10^{-3}) \approx 0,708$$

Từ phương trình (II) ta có:

$$\frac{\partial E}{\partial W_{\alpha\beta}} = (y - d) \frac{\partial y}{\partial W_{\alpha\beta}} = (y - d) f'(g) \frac{\partial g}{\partial W_{\alpha\beta}}$$

$$\rightarrow \frac{\partial E}{\partial W_{\alpha\beta}} = (y - d) g(1 - g) \frac{\partial (v_1 V_1 + v_2 V_2)}{\partial W_{\alpha\beta}}$$

$$\rightarrow \frac{\partial E}{\partial W_{\alpha\beta}} = (y - d) g(1 - d) v_\alpha f'(v_\alpha) x_\beta$$

Từ đó ta có đạo hàm theo trọng số $w_{11}, w_{12}, w_{21}, w_{22}$ như sau:

$$\frac{\partial E}{\partial W_{11}} = (y - d) g(1 - d) v_1 f'(u_1) x_1$$

$$= 3,35 \cdot 10^{-4}$$

Hoàn toàn tương tự:

$$\frac{\partial E}{\partial W_{12}} = -4,03 \cdot 10^{-4}; \frac{\partial E}{\partial W_{21}} = -8,845 \cdot 10^{-4}; \frac{\partial E}{\partial W_{22}} = -5,307 \cdot 10^{-4}$$

Chọn $\eta = 100$ ta tính được các trọng số vào của mạng như sau:

$$w_{11} = x_1 - \eta \frac{\partial E}{\partial W_{11}} = 0,1 - 100 \cdot 3,35 \cdot 10^{-4} = 0,0665$$

Tương tự ta có: $w_{12} = 0,24; w_{21} = 0,39, w_{22} = 0,45$

Vậy sau khi xác định được các giá trị của trọng số trong mạng ta thử kết quả với bộ trọng số mới.

Cho $x_1=0,5; x_2=0,3$

Ta có:

$$\begin{cases} u_1 = w_{11}x_1 + w_{12}x_2 = 0,105 \\ u_2 = w_{21}x_1 + w_{22}x_2 = 0,33 \end{cases} \Rightarrow \begin{cases} v_1 = \log \text{sig}(u_1) = 0,526 \\ v_2 = \log \text{sig}(u_2) = 0,582 \end{cases}$$

$$\Rightarrow g = 0,6v_1 + 0,708v_2 = 0,728$$

với $y = \text{logsig}(g) \Rightarrow y = 0,67 \approx d = 0,7$

3.5. Kết luận chương III

Qua những phân tích về mô hình lai, xây dựng các khối tuyến tính, phi tuyến và qua các ví dụ cụ thể trong chương III có thể thấy được việc sử dụng mô hình lai giữa mạng truyền thẳng nhiều lớp và phương pháp lựa chọn giá trị tuyệt đối là phù hợp sử dụng trong mô hình dự báo công suất phụ tải điện. Ước lượng giá trị của khối tuyến tính sẽ được ứng dụng bằng phương pháp bình phương cực tiểu được phân tích dựa vào phương pháp bình phương cực tiểu cụ thể được khai triển bằng giải phương pháp phân tích giá trị kỳ dị SVD. Sau đó thực hiện loại bỏ thành phần này ra khỏi bộ số liệu đầu vào với mục đích chỉ giữ lại khối phi tuyến trong tín hiệu. Các bước thực hiện để xác định tín hiệu sẽ được trình bày ở chương IV.

CHƯƠNG IV: XÂY DỰNG MÔ HÌNH, TÍNH TOÁN VÀ MÔ PHỎNG

Từ những phân tích về mô hình lai trong chương 3 tập trung phân tích việc chọn bộ số liệu đầu vào, đầu ra, mô hình đề xuất của luận văn. Từ đó xác định quy trình xây dựng mô hình dự báo trong luận văn, xây dựng khối tuyến tính và phi tuyến trong mô hình lai thực hiện dự báo công suất đỉnh phụ tải P_{\max} và đáy phụ tải P_{\min} . Sử dụng các công thức kiểm chứng sai số của mô hình.

4.1. Xây dựng mô hình dự báo công suất phụ tải

4.1.1. Mô hình dự báo công suất phụ tải [1,3]

4.1.1.1. Các số liệu đầu vào

Mô hình dự báo được tác giả lựa chọn và được trình bày ở các chương trên sử dụng bộ số liệu công suất phụ tải điện đã được đo với d ngày. Luận văn xây dựng theo hướng dự báo công suất phụ tải P_{\max} , P_{\min} theo ngày. Bộ số liệu xây dựng mẫu tín hiệu:

$$\text{Công suất đỉnh phụ tải của ngày được xác định: } P_{\max}(d) = \max_{i=1 \rightarrow 24} P_i(d) \quad (4.1)$$

$$\text{Công suất đáy phụ tải của ngày được xác định: } P_{\min}(d) = \min_{i=1 \rightarrow 24} P_i(d) \quad (4.2)$$

Giả sử với bộ số liệu d ngày trên, ta sẽ sử dụng số liệu $(d-x)$ ngày đầu làm bộ số liệu học, xây dựng mô hình dự báo với $(d-x)$ ngày đầu và số liệu (x) còn lại được sử dụng làm bộ số liệu kiểm tra.

4.1.1.2. Mô hình đề xuất của luận văn

Mô hình lai có đầu vào là P_{\max} , P_{\min} với đầu ra là 2 mô hình dự báo công suất phụ tải điện cho P_{\max} , P_{\min} . Với số liệu đầu vào của mô hình là số liệu đã được đo đạc. Tín hiệu đầu ra được xác định là tổng hợp của khối ước lượng tuyến tính và khối ước lượng phi tuyến.

4.1.2. Quy trình xây dựng các mô hình dự báo trong luận văn [1]

Để thuận tiện cho việc tính toán và mô phỏng tác giả đã ứng dụng phần mềm Matlab 2010b bản 7.10. Việc xây dựng mô hình dự báo công suất phụ tải dựa trên các bước sau:

1. Xây dựng mô hình dự báo P_{\max} dựa trên mô hình lai giữa mô hình tuyến tính sử dụng thuật toán SVD và mô hình phi tuyến sử dụng mạng MLP.
2. Tương tự ta xây dựng mô hình P_{\min} dựa trên mô hình lai như trên.

Sau đây là những phân tích chi tiết để xây dựng mô hình dự báo công suất phụ tải được đề xuất trong luận văn.

4.1.2.1. Xây dựng mô hình dự báo công suất đỉnh phụ tải P_{\max}

a) Xây dựng khối tuyến tính

Giả sử với bộ số liệu mẫu $\{x_i, d_i\}$ gồm p giá trị vậy ta, $i=1, 2, \dots, p$. Khối tuyến tính trong mô hình được xác định dựa trên cơ sở tối ưu hóa sai số E trong bộ số liệu mẫu p . Vậy ta sẽ có:

$$\forall i: f_1(x_i) \approx d \Leftrightarrow E = \frac{1}{2} \sum_{i=1}^p \|f_1(x_i) - d_i\|^2 \Rightarrow \min \quad (4.4)$$

Từ đây ta thực hiện xác định giá trị công suất đỉnh phụ tải P_{\max} như sau:

$$P_{\max}(d) \approx f_2(P_{\max}(d-i), P_{\min}(d-i), T_{tb}(d-i)) + \sum_{i=1}^K [a_{i1} \cdot P_{\max}(d-i) + a_{i2} \cdot P_{\min}(d-i) + a_{i3} \cdot T_{tb}(d-i)] \quad (4.5)$$

với $f_2()$ là hàm phi tuyến ($i=1, 2, \dots, K$), a_{ij} là hệ số mô hình tuyến tính; P_{\min} là giá đáy của công suất phụ tải của các ngày có ảnh hưởng lớn nhất tới P_{\max} của ngày dự báo d là $d-i_1, d-i_2, \dots$, T_{tb} là giá trị nhiệt độ trung bình của ngày ảnh hưởng nhất đến P_{\max} của ngày d .

Từ phương trình trên ta xác định được P_{\max} của các ngày trong quá khứ và được xác định qua hệ các phương trình tuyến tính sau:

$$\begin{cases} a_1 \cdot P_{\max}(K) + a_2 \cdot P_{\max}(K-1) + \dots + a_K P_{\max}(d-K) \approx P_{\max}(K+1) \\ a_1 \cdot P_{\max}(K-1) + a_2 \cdot P_{\max}(K-2) + \dots + a_K P_{\max}(d-K+1) \approx P_{\max}(K+2) \\ \dots \\ a_1 \cdot P_{\max}(N_{\max}-1) + a_2 \cdot P_{\max}(N_{\max}-2) + \dots + P_{\max}(N_{\max}-K) \approx P_{\max}(N_{\max}) \end{cases} \quad (4.6)$$

với $a = [a_1, a_2, \dots, a_K]$ là hệ số của mô hình tuyến tính.

Biến đổi công thức (4.6) về dạng ma trận

$$\begin{bmatrix} P_{\max}(K) & P_{\max}(K-1) & \dots & P_{\max}(1) \\ P_{\max}(K-1) & P_{\max}(K-2) & \dots & P_{\max}(2) \\ P_{\max}(K-2) & P_{\max}(K-3) & \dots & P_{\max}(3) \\ P_{\max}(N_{\max}-1) & P_{\max}(N_{\max}-2) & \dots & P_{\max}(N_{\max}-K) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{bmatrix} \approx \begin{bmatrix} P_{\max}(K+1) \\ P_{\max}(K+2) \\ \vdots \\ P_{\max}(N_{\max}) \end{bmatrix} \quad (4.7)$$

Quá trình tìm P_{\max} thực hiện trong 3 bước sau:

1. Thực hiện lựa chọn số liệu trong quá khứ K sao cho véc-tơ \mathbf{a} đủ lớn. Trong khuôn khổ luận văn này, tác giả đề xuất sử dụng $K = 60$ tương đương với 2 tháng số liệu quá khứ trước ngày cần dự báo.

2. Sau khi có được K quá khứ, ta thực hiện xác định véc-tơ $\mathbf{a} = [a_1, a_2, \dots, a_K]^T$ sao cho đạt cực tiểu hàm sai số ước lượng. Ta có:

$$T_{\max}(d) = \sum_{i=1}^K [a_i \cdot T_{\max}(d-i)] \quad (4.8)$$

bằng phương pháp phân tích theo các giá trị kỳ dị SVD.

Bài toán mô hình tuyến tính trên ta có thể đưa về giải phương trình tìm nghiệm \mathbf{x} của phương trình, theo công thức (3.16) ta có:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

trong đó \mathbf{A} là ma trận cỡ $K \times N$, \mathbf{x} tương đương với a_i ($i = 1, 2, \dots, K$) và \mathbf{b} tương đương với giá trị P_{\max} của các ngày $K + 1, K + 2, \dots, N_{\max}$. Do hệ phương trình có số phương trình nhiều hơn số ẩn vì vậy phương trình sẽ không có nghiệm duy nhất. Ta sẽ giải quyết bài toán trên dựa vào bài toán xác tối ưu hóa sai số, theo công thức số (3.23):

$$\mathbf{x} = \bar{\mathbf{x}} = \mathbf{A}^+ \cdot \mathbf{b}$$

với \mathbf{A}^+ là ma trận giả nghịch đảo của $\mathbf{A}^{m \times n}$ thì ta có $\|\mathbf{A} \cdot \mathbf{x} - \mathbf{b}\|$ đạt giá trị nhỏ nhất và $\|\bar{\mathbf{x}}\|$ có độ lớn nhỏ nhất.

Theo phân tích SVD của ma trận \mathbf{A} , từ công thức số (3.23) ta sẽ có:

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

với $\mathbf{S}^+ = \text{diag}(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_r}) \in \mathbf{R}^{n \times m}$. Qua các bước tính toán được trình bày chi tiết

ở mục 3.3 ta xác định được giá trị $\bar{\mathbf{x}} = \mathbf{A}^+ \cdot \mathbf{b} \approx \mathbf{a}_i$ theo công thức (3.23).

3. Sau khi xác định được véc-tơ \mathbf{a} ta thực hiện tìm và loại bỏ giá trị nhỏ nhất của véc-tơ \mathbf{a} ứng với ngày trong quá khứ có ảnh hưởng ít nhất tới dự báo. Loại ngày có ảnh hưởng ít này ra khỏi bộ số liệu quá khứ.

Ta thực hiện lặp lại bước 2 và bước 3 để loại bỏ dần số K xuống một số phù hợp để mô hình không quá phức tạp.

$$P_{\max}(d) \approx \sum_{i=1}^K [a_{i1} \cdot P_{\max}(d-i) + a_{i2} \cdot P_{\min}(d-i) + a_{i3} \cdot T_{tb}(d-i)]$$

(4.9)

Giả sử ở khối tuyến tính xác định được 8 số liệu quá khứ $P_{\max}(d-1)$, $P_{\max}(d-2)$, $P_{\max}(d-3)$, $P_{\max}(d-6)$, $P_{\min}(d-1)$, $P_{\min}(d-2)$, $P_{\min}(d-7)$, $T_{tb}(d-1)$ tương ứng với $a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8$ ảnh hưởng đến P_{\max} của ngày dự báo d .

Từ đó ta có mô hình ước lượng khối tuyến tính là:

$$P_{\max}(d) = a_1 P_{\max}(d-1) + a_2 P_{\max}(d-2) + a_3 P_{\max}(d-3) + a_4 P_{\max}(d-6) + a_5 P_{\min}(d-1) + a_6 P_{\min}(d-2) + a_7 P_{\max}(d-7) + a_8 T_{tb}(d-1)$$

Để kiểm chứng chất lượng của mô hình dự báo được đề xuất trong luận văn ta dựa vào sai số của phương pháp dự báo. Sai số dự báo là những sai lệch giữa giá trị đo được và giá trị dự báo được qua đó giúp ta đánh giá được chất lượng của mô hình và là cơ sở để điều chỉnh mô hình dự báo. Tác giả đề xuất sử dụng các công thức dưới đây để đánh giá sai số:

- Trung bình của sai số tương đối: MRE (Mean Relative Error)

$$MRE = \frac{1}{N} \left| \frac{y-d}{y} \right| \cdot 100\% \quad (4.10)$$

- Trung bình của sai số tuyệt đối: MAE (Mean Absolute Error)

$$MAE = \frac{1}{N} |y-d| \quad (4.11)$$

- Giá trị lớn nhất của sai số tuyệt đối: MaxAE (Maximum Absolute Error)

$$MaxAE = \max |y-d| \quad (4.12)$$

trong đó: y là giá trị thực tại thời điểm đo; d là giá trị tính toán dự báo; N là số lượng thời điểm đo (số mẫu)

b) Xây dựng khối phi tuyến

Sau khi đã xác định được mối quan hệ giữa công suất đỉnh phụ tải trong ngày dự báo d với K_{min} ngày trong quá khứ, ta tính sai số chênh lệch giữa số liệu thực tế và số liệu ước lượng.

$$f_2(d) = P_{\max}(d) - \sum_{i=1}^K \left[a_{i1} \cdot P_{\max}(d-i) + a_{i2} \cdot P_{\min}(d-i) + a_{i3} \cdot T_{tb}(d-i) \right] \quad (4.13)$$

với $f_2(d)$ là phần phụ thuộc phi tuyến còn lại giữa công suất đỉnh phụ tải trong ngày với các ngày trong quá khứ.

Khối phi tuyến sẽ được xác định sau khi tìm được các thông số khối phi tuyến. Thành phần này được xây dựng bằng mạng MLP. Qua đó ta ước lượng sai số còn lại. Số đầu vào của mạng MLP được chọn là số ngày có ảnh hưởng xác định được ở khối và số nơ-rôn trong lớp ẩn được xác định qua quá trình lập trình tính toán trên phần mềm Matlab.

Như ở phần xây dựng khối tuyến tính đã giả sử với p ngày số liệu. Với $p-x$ ngày đầu sử dụng làm số liệu học và x ngày số liệu còn lại được sử dụng làm bộ số liệu kiểm tra.

Ta xác lập một mạng MLP với 09 đầu vào như với mô hình ước lượng khối tuyến tính (ứng với 3 giá trị công suất phụ tải đỉnh của 03 ngày trong quá khứ là $d-1, d-2, d-6$; 03 giá trị công suất đáy phụ tải của 03 ngày trong quá khứ là $d-1, d-2, d-12$; 3 giá trị nhiệt độ trung bình ngày trong quá khứ tại $d-1, d-4, d-5$). Có 01 đầu ra (ứng với giá trị công suất đỉnh phụ tải của ngày d cần dự báo).

Số liệu đầu vào của mạng là sai số của quá trình học của mô hình tuyến tính được tính toán trong phần trên. Tiếp sau quá trình học, ta thực hiện kiểm tra mạng MLP với bộ số liệu được lựa chọn làm bộ số liệu kiểm tra.

Ta thực hiện kiểm tra chất lượng dự báo của khối ước lượng phi tuyến với các công thức (4.10) đến (4.12) giống như đối với khối tuyến tính.

4.1.2.2. Xây dựng mô hình dự báo giá trị đáy phụ tải P_{min}

a) Xây dựng khối tuyến tính

Tương tự với việc xây dựng dự báo giá trị đỉnh phụ tải P_{max} , ta thực hiện xét với bộ số liệu mẫu $\{x_i, d_i\}$ gồm p giá trị vậy ta, $i = 1, 2, 3, \dots, p$ như trên. Khối tuyến tính trong mô hình được xác định dựa trên cơ sở tối ưu hóa sai số E trong bộ số liệu mẫu p . Vậy theo công thức (4.4) ta sẽ có:

$$\forall i: f_t(x_i) \approx d \Leftrightarrow E = \frac{1}{2} \sum_{i=1}^p \|f_t(x_i) - d_i\|^2 \Rightarrow \min$$

Từ đây ta thực hiện xác định P_{min} Với giả thiết giá trị công suất đáy phụ tải P_{min} được ước lượng như sau:

$$P_{min}(d) \approx f_{pt}(P_{min}(d-i), P_{max}(d-i), T_{tb}(d-i)) + \sum_{i=1}^K \left[a_{i1} \cdot P_{max}(d-i) + a_{i2} \cdot P_{min}(d-i) + a_{i3} \cdot T_{tb}(d-i) \right] \quad (4.14)$$

với $f_{pt}()$ là hàm phi tuyến ($i = 1, 2, 3, \dots, K$), a_{ij} là hệ số mô hình tuyến tính;

P_{\max} là giá trị công suất đầy phụ tải của các ngày có ảnh hưởng lớn nhất tới P_{\min} của ngày dự báo d là $d - i_1, d - i_2, \dots$

T_{tb} là giá trị nhiệt độ trung bình của ngày ảnh hưởng nhất đến P_{\min} của ngày d ;

Từ phương trình trên ta xác định được P_{\min} của các ngày trong quá khứ và được xác định qua hệ các phương trình tuyến tính sau:

$$\begin{cases} a_1 \cdot P_{\min}(K) + a_2 \cdot P_{\min}(K-1) + \dots + a_K P_{\min}(d-K) \approx P_{\min}(K+1) \\ a_1 \cdot P_{\min}(K-1) + a_2 \cdot P_{\min}(K-2) + \dots + a_K P_{\min}(d-K+1) \approx P_{\min}(K+2) \\ \dots \\ a_1 \cdot P_{\min}(N_{\min}-1) + a_2 \cdot P_{\min}(N_{\min}-2) + \dots + P_{\min}(N_{\min}-K) \approx P_{\min}(N_{\min}) \end{cases} \quad (4.15)$$

với $a = [a_1, a_2, \dots, a_K]$ là hệ số của mô hình tuyến tính.

Biến đổi công thức (3.15) về dạng ma trận

$$\begin{bmatrix} P_{\min}(K) & P_{\min}(K-1) & \dots & P_{\min}(1) \\ P_{\min}(K-1) & P_{\min}(K-2) & \dots & P_{\min}(2) \\ P_{\min}(K-2) & P_{\min}(K-3) & \dots & P_{\min}(3) \\ P_{\min}(N_{\min}-1) & P_{\min}(N_{\min}-2) & \dots & P_{\min}(N_{\min}-K) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{bmatrix} \approx \begin{bmatrix} P_{\min}(K+1) \\ P_{\min}(K+2) \\ \vdots \\ T_{\min}(N_{\min}) \end{bmatrix} \quad (4.16)$$

Quá trình tìm khối tuyến tính trong ước lượng P_{\min} thực hiện tương tự như với P_{\max}

1. Thực hiện lựa chọn số liệu trong quá khứ K sao cho véc-tơ a đủ lớn. Với việc sử dụng $K = 60$ tương đương với 2 tháng số liệu quá khứ trước ngày cần dự báo.

2. Sau khi có được K quá khứ, ta thực hiện xác định véc-tơ $a = [a_1, a_2, \dots, a_K]^T$ sao cho đạt cực tiểu hàm sai số ước lượng. Ta có:

$$P_{\min}(d) = \sum_{i=1}^K [a_i \cdot P_{\min}(d-i)] \quad (4.17)$$

bằng phương pháp phân tích điểm kỳ dị SVD.

3. Sau khi xác định được véc-tơ a ta thực hiện tìm và loại bỏ giá trị nhỏ nhất của véc-tơ a ứng với ngày trong quá khứ có ảnh hưởng ít nhất tới dự báo. Loại ngày có ảnh hưởng ít này ra khỏi bộ số liệu quá khứ.

Ta thực hiện lặp lại bước 2 và bước 3 để loại bỏ dần số K xuống một số phù hợp để mô hình không quá phức tạp.

$$P_{\min}(d) \approx \sum_{i=1}^K \left[a_{i1} \cdot P_{\min}(d-i) + a_{i2} \cdot P_{\max}(d-i) + a_{i3} \cdot T_{tb}(d-i) \right] \quad (4.18)$$

Giả sử ở khối tuyến tính xác định được 8 số liệu quá khứ $P_{\min}(d-1)$, $P_{\min}(d-2)$, $P_{\min}(d-3)$, $P_{\min}(d-7)$, $P_{\max}(d-1)$, $P_{\max}(d-2)$, $P_{\max}(d-6)$, $T_{tb}(d-1)$ tương ứng với: $a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8$ ảnh hưởng đến P_{\min} của ngày dự báo d .

Từ đó ta có mô hình ước lượng khối tuyến tính là:

$$P_{\min}(d) = a_1 P_{\min}(d-1) + a_2 P_{\min}(d-2) + a_3 P_{\min}(d-3) + a_4 P_{\min}(d-7) + a_5 P_{\max}(d-1) + a_6 P_{\max}(d-2) + a_7 P_{\max}(d-6) + a_8 T_{tb}(d-1) \quad (4.19)$$

Để kiểm chứng chất lượng dự báo của khối tuyến tính ta sử dụng các công thức (4.10) đến (4.12) tương tự trong mục xác định khối tuyến tính P_{\max}

b) Xây dựng khối phi tuyến

Sau khi đã xác định được mối quan hệ giữa công suất đáy phụ tải trong ngày dự báo d với K_{\min} ngày trong quá khứ, ta tính sai số chênh lệch giữa số liệu thực tế và số liệu ước lượng.

$$f_2(d) = P_{\min}(d) - \sum_{i=1}^K \left[a_{i1} \cdot P_{\min}(d-i) + a_{i2} \cdot P_{\max}(d-i) + a_{i3} \cdot T_{tb}(d-i) \right] \quad (4.20)$$

với $f_2(d)$ là phần phụ thuộc phi tuyến còn lại giữa công suất đáy phụ tải trong ngày với các ngày trong quá khứ.

Tương tự như hướng giải quyết xác định khối phi tuyến của công suất đỉnh phụ tải P_{\max} khối này được xây dựng bằng mạng MLP. Qua đó ta ước lượng sai số còn lại. Số đầu vào của mạng MLP được chọn là ngày có ảnh hưởng xác định được ở khối tuyến tính và số nơ-rôn trong lớp ẩn được xác định qua quá trình lập trình tính toán trên phần mềm Matlab.

Như ở phần xây dựng khối tuyến tính của P_{\min} như đã giả sử với p ngày số liệu. Với $p-x$ ngày đầu sử dụng làm số liệu học và x ngày số liệu còn lại được sử dụng làm bộ số liệu kiểm tra.

Ta xác lập một mạng MLP với 9 đầu vào như với mô hình ước lượng khối tuyến tính (ứng với 3 giá trị công suất đáy phụ tải của 3 ngày trong quá khứ là $d-1$, $d-2$, $d-10$; 2 giá trị công suất đỉnh phụ tải của 2 ngày trong quá khứ là $d-1$, $d-2$; 4 giá trị

hiệt độ trung bình ngày trong quá khứ tại $d-1$, $d-3$, $d-5$, $d-9$). Có 01 đầu ra (ứng với giá trị công suất đáy phụ tải của ngày d cần dự báo).

Tiếp sau quá trình học, ta thực hiện kiểm tra mạng MLP với bộ số liệu được lựa chọn làm bộ số liệu kiểm tra.

Ta cũng thực hiện kiểm tra dự báo đối với khối ước lượng phi tuyến với các công thức (4.10) đến (4.12).

4.2. Kết quả tính toán và mô phỏng

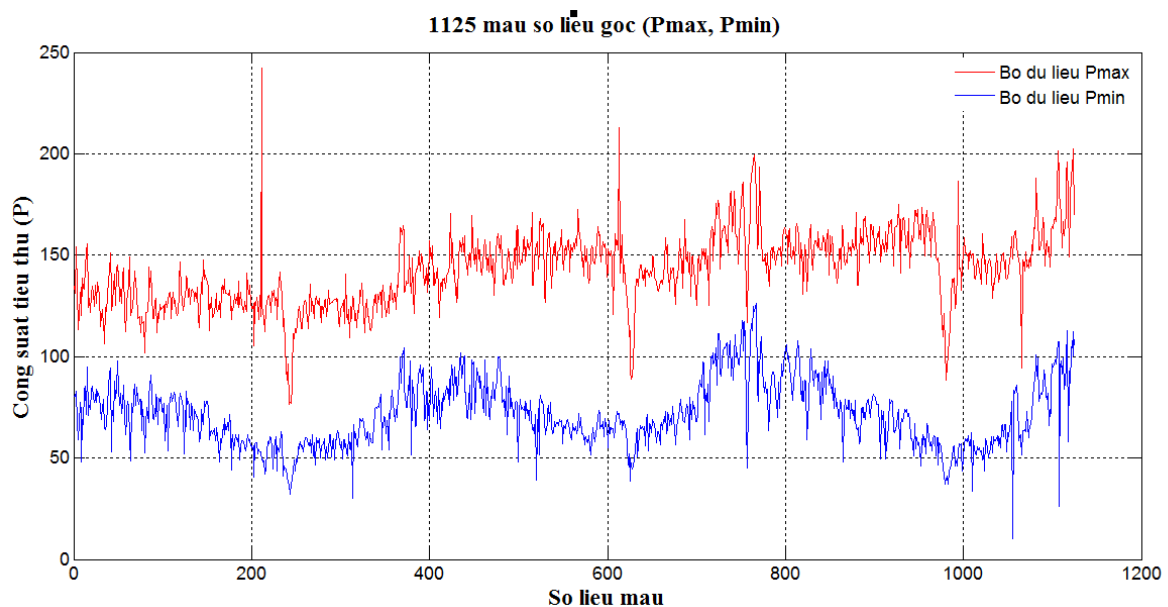
Sử dụng bộ số liệu do Công ty điện lực thành phố Hạ Long và trạm quan trắc khí tượng thủy văn thành phố Hạ Long cung cấp. Việc tính toán và mô phỏng tác giả đã ứng dụng phần mềm Matlab 2010b bản 7.10. Code của các chương trình (xem phụ lục).

4.2.1. Bộ số liệu đầu vào sử dụng cho mô hình dự báo

Mô hình dự báo được tác giả lựa chọn và được trình bày ở các chương trên sử dụng bộ số liệu là công suất đỉnh, đáy phụ tải điện hàng ngày tổng hợp tại Công ty điện lực thành phố Hạ Long tỉnh Quảng Ninh và bộ số liệu nhiệt độ trung bình ngày đo tại trạm quan trắc của thành phố Hạ Long.

Bộ số liệu được lấy trong thời gian từ 01/6/2016 đến 30/06/2019. Bộ số liệu xây dựng mẫu tín hiệu: được tính bằng công thức (4.1) và (4.2).

Với 1125 ngày dữ liệu ta sẽ sử dụng 750 ngày đầu tiên (tương đương 2/3 bộ số liệu) để làm bộ số liệu học, dùng để xây dựng và điều chỉnh thích nghi các tham số của các mô hình ước lượng. Sử dụng 375 ngày còn lại (tương đương 1/3 bộ số liệu) sẽ được dùng để kiểm tra chất lượng của mô hình. Bộ số liệu gồm 1125 mẫu số liệu được miêu tả như hình 4.1.



Hình 4.1. Đồ thị biểu diễn bộ số liệu mẫu (P_{max} , P_{min})

4.2.2. Kết quả mô hình dự báo công suất đỉnh phụ tải P_{max}

4.2.2.1. Ước lượng thành phần tuyến tính

Dùng phương pháp khai triển thành phần tuyến tính như được trình bày ở mục 4.1.2.1.a, kết quả như sau:

Đầu tiên, ta xác định giá trị P_{max} trong quá khứ ảnh hưởng đến P_{max} dự báo. Mô hình ước lượng tuyến tính cho P_{max} được điều chỉnh từ khoảng giá trị $K_{max} = 60$ ngày (tương đương với 2 tháng) giảm xuống với $K_{min} < 5$. Lần lượt sử dụng số liệu P_{max} của 60 ngày và dùng SVD để phân tích, kết quả 3 ngày có ảnh hưởng lớn (hệ số phụ thuộc a_{i1} cao) tới giá trị ngày cần dự báo là: $d - 1, d - 2, d - 6$.

Tiếp theo, ta xác định giá trị P_{min} trong quá khứ ảnh hưởng đến P_{max} dự báo. Kết quả ta 3 ngày có ảnh hưởng lớn nhất đến P_{max} của ngày d đó là các ngày $d-1, d-2$ và $d-12$. Các ngày còn lại có hệ số tuyến tính nhỏ hoặc ở xa ngày dự báo lên ta loại bỏ.

Tương tự, ta xét ảnh hưởng của nhiệt độ trung bình ngày ảnh hưởng đến P_{max} của ngày dự báo. Kết quả ta xác định được 3 ngày mà T_{tb} ảnh hưởng lớn đến P_{max} của ngày dự báo, đó là các ngày $d-1, d-4$ và $d-5$. Tổng hợp lại ta có mô hình được lựa chọn để dự báo giá trị P_{max} của ngày thứ d sẽ gồm 09 số liệu từ quá khứ:

$$P_{max}(d) = f \left(\begin{matrix} P_{max}(d-1), P_{max}(d-2), P_{max}(d-6) \\ P_{min}(d-1), P_{min}(d-2), P_{min}(d-12), T_{tb}(d-1), T_{tb}(d-4), T_{tb}(d-5) \end{matrix} \right) \quad (4.21)$$

Dùng phương pháp khai triển hệ số tuyến tính (SVD) ta xác định được các hệ số tuyến tính ; Cụ thể như phương trình 4.22.

$$P_{max}(d) = 0,57 \cdot P_{max}(d-1) + 0,17 \cdot P_{max}(d-2) + 0,098 \cdot P_{max}(d-6) + 0,40 \cdot P_{min}(d-1) + 0,167 \cdot P_{min}(d-2) + 0,106 \cdot P_{min}(d-12) + 0,082 \cdot T_{tb}(d-1) + 0,042 \cdot T_{tb}(d-4) + 0,036 \cdot T_{tb}(d-5) \quad (4.22)$$

Bảng 4.1 là sai số học và sai số kiểm tra khi ước lượng thành phần tuyến tính trong mô hình dự báo công suất đỉnh phụ tải P_{max} bằng phương pháp khai triển các hệ số tuyến tính SVD, qua bảng 4.1 ta nhận thấy sai số học và sai số kiểm tra tương đối nhỏ cỡ 7%.

Bảng 4.1. Kết quả ước lượng công suất đỉnh phụ tải (P_{max}) dùng phương pháp khai triển tuyến tính SVD

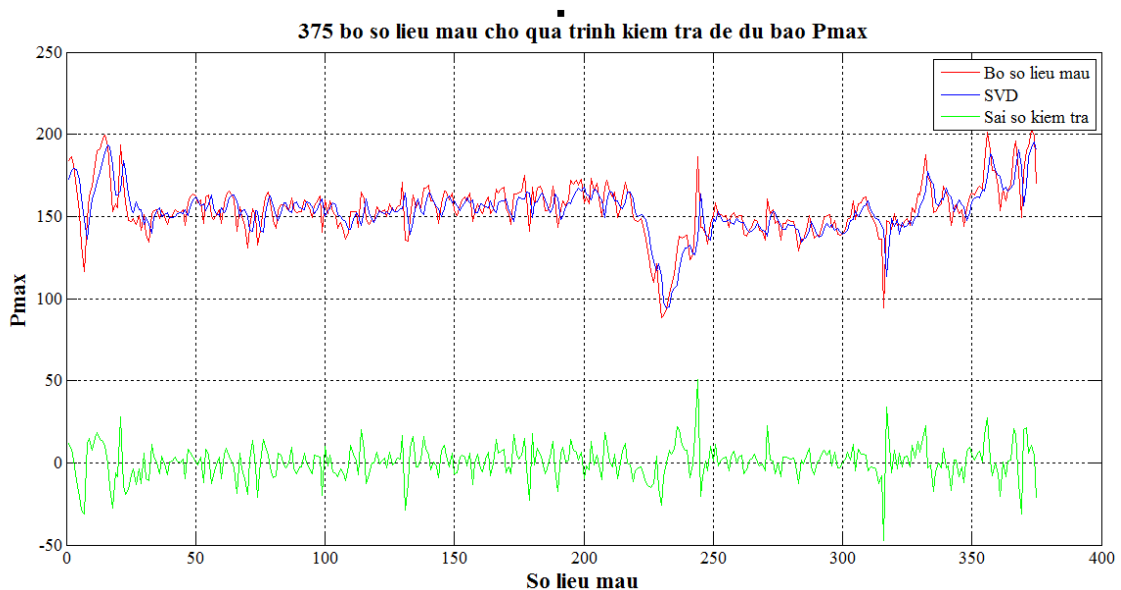
Quá trình	Sai số	SVD
Học	$MRE = \frac{1}{N} \left \frac{y-d}{y} \right \cdot 100\%$	6,67%
	$MAE = \frac{1}{N} y-d $	4,87
	$MaxAE = \max y-d $	38,62

Kiểm tra	$MRE = \frac{1}{N} \left \frac{y-d}{y} \right \cdot 100\%$	7,50%
	$MAE = \frac{1}{N} y-d $	4,97
	$MaxAE = \max y-d $	50,62

Hình 4.2 và hình 4.3 thể hiện quá trình học và quá trình kiểm tra khi triển khai thành phần tuyến tính cho mô hình dự báo P_{max} , đường màu đỏ là bộ số liệu mẫu, đường màu xanh nước biển là kết quả học/kiểm tra, đường màu xanh là cây là sai số quá trình học/kiểm tra.



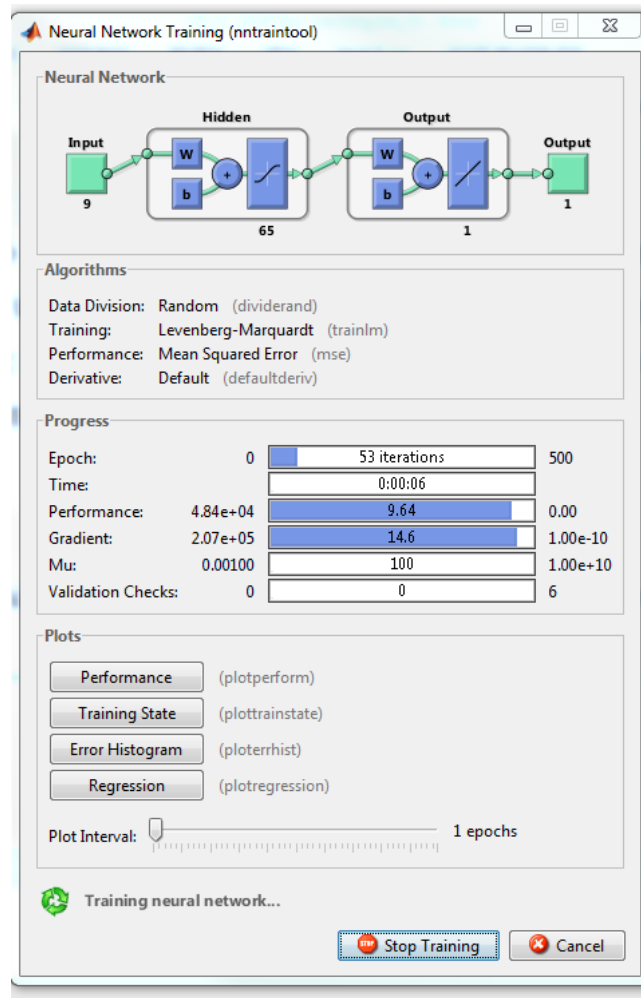
Hình 4.2. Đồ thị sai số học khi khai triển thành phần tuyến tính cho mô hình dự báo P_{max}



Hình 4.3. Đồ thị sai số quá trình kiểm tra khi khai triển thành phần tuyến tính cho mô hình dự báo P_{max}

4.2.2.2. Ước lượng thành phần phi tuyến bằng mạng nơ-rôn MLP

Thực hiện các bước theo mục 4.1.2.1.b, ta có kết quả như sau: mô hình mạng MLP được xây dựng có cấu trúc 09 đầu vào ứng với 09 đại lượng quá khứ (công thức 4.22), 1 lớp ẩn với 500 nơ-rôn và 1 đầu ra ứng với giá trị cần dự báo cho ngày tiếp theo. Mô hình huấn luyện mạng MLP để ước lượng P_{max} như hình 4.4.



Hình 4.4. Quy trình học của mạng MLP khi ước lượng P_{max}

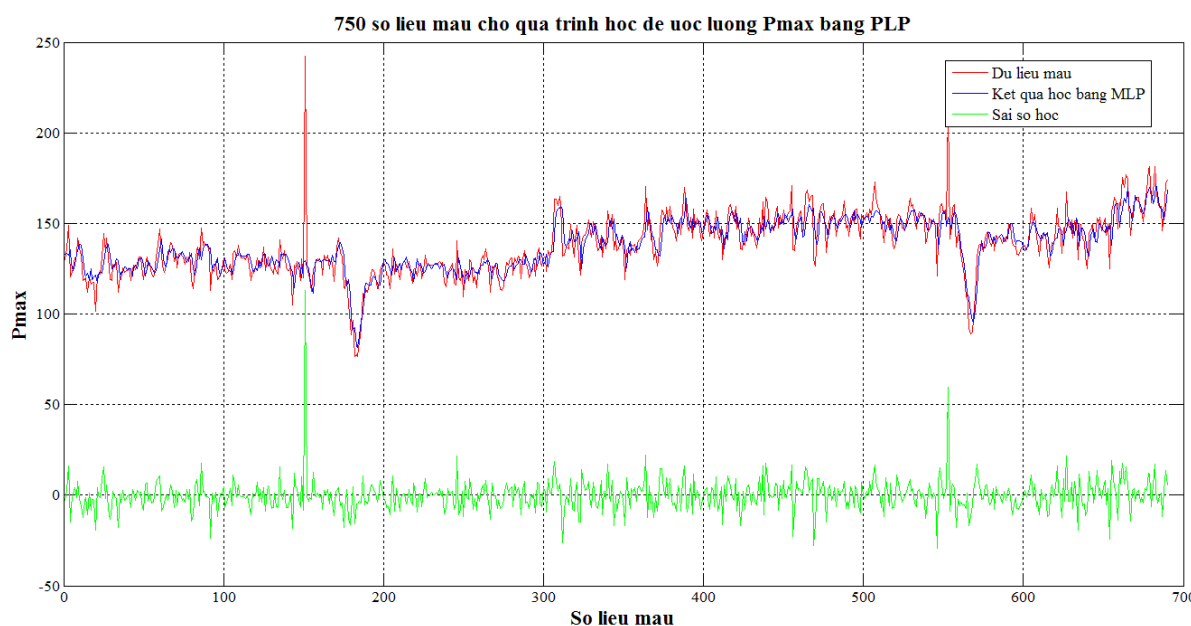
Bảng 4.2 là kết quả sai số học và sai số kiểm tra của các mô hình cho bài toán dự báo phụ tải đỉnh. Qua bảng 4.2 cho thấy sai số học cỡ 9% còn sai số kiểm tra cỡ 11%.

Bảng 4.2. Một số kết quả trên bộ số liệu học của bài toán ước lượng P_{min}

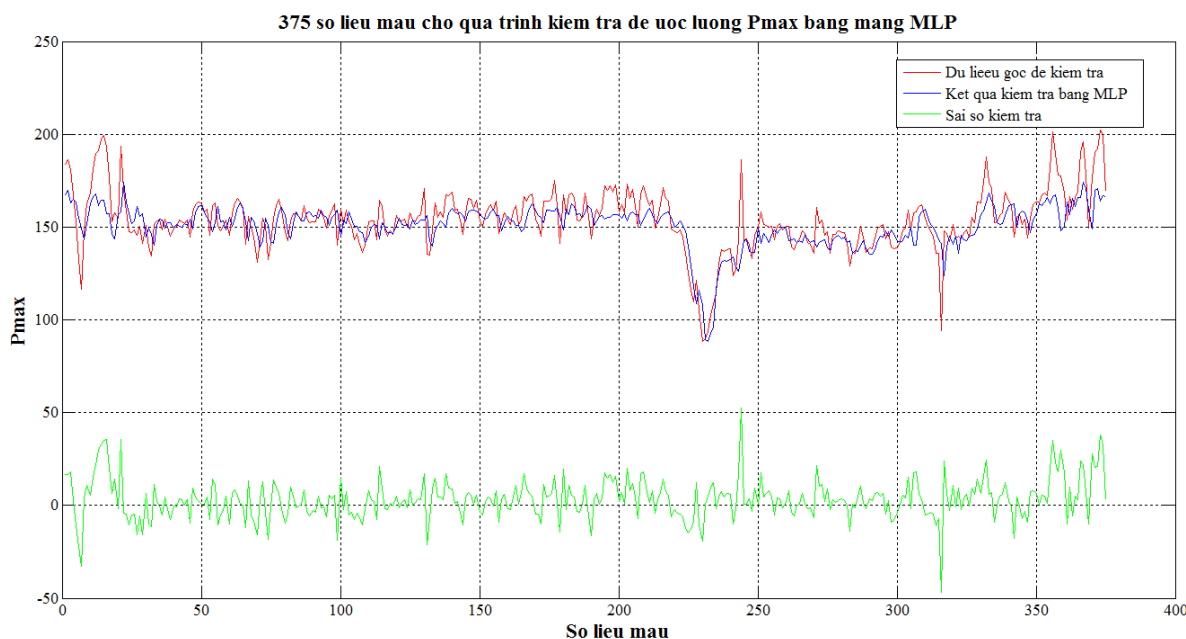
Quá trình	Sai số	Sai số của mạng MLP
Học	$MRE = \frac{1}{N} \left \frac{y - d}{y} \right \cdot 100\%$	9,03%
	$MAE = \frac{1}{N} y - d $	6,65
	$MaxAE = \max y - d $	25,92

Kiểm tra	$MRE = \frac{1}{N} \left \frac{y-d}{y} \right \cdot 100\%$	11,09%
	$MAE = \frac{1}{N} y-d $	7,28
	$MaxAE = \max y-d $	38,42

Hình 4.5 và hình 4.6 là đồ thị sai số quá trình học và kiểm tra khi ước lượng P_{max} bằng mạng nơ-ron MLP, đường màu đỏ là bộ số liệu mẫu, đường màu xanh nước biển là kết quả học bằng MLP, đường màu xanh lá cây là sai số học/kiểm tra.



Hình 4.5. Đồ thị sai số quá trình học khi ước lượng P_{max} bằng mạng MLP



Hình 4.6. Đồ thị sai số quá trình kiểm tra khi ước lượng P_{max} bằng mạng MLP

4.2.3. Kết quả mô hình dự báo công suất đáy phụ tải P_{min}

4.2.3.1. Ước lượng thành phần tuyến tính

Dùng phương pháp khai triển thành phần tuyến tính như được trình bày ở mục 4.1.2.2.a, kết quả như sau:

Đầu tiên, ta xác định giá trị P_{min} trong quá khứ ảnh hưởng đến P_{min} dự báo. Mô hình ước lượng tuyến tính cho P_{min} được điều chỉnh từ khoảng giá trị $K_{max} = 60$ ngày (tương đương với 2 tháng) giảm xuống với $K_{min} < 5$. Lần lượt sử dụng số liệu P_{min} của 60 ngày và dùng SVD để phân tích, kết quả 3 ngày có ảnh hưởng lớn (hệ số phụ thuộc a_{i1} cao) tới giá trị ngày cần dự báo là: $d-1, d-2, d-10$. Ngày tiếp theo có hệ số tuyến tính lớn ảnh hưởng đến ngày dự báo P_{min} là ngày $d-20$ nhưng ở xa ngày dự báo lên bỏ qua.

Tiếp theo, ta xác định giá trị P_{max} trong quá khứ ảnh hưởng đến P_{min} dự báo. Kết quả ta 2 ngày có ảnh hưởng lớn nhất đến P_{min} của ngày d đó là các ngày $d-1, d-2$, Các ngày tiếp theo có hệ số tuyến tính lớn là ngày $d-56$ và $d-60$ nhưng ở xa ngày dự báo lên không xét.

Trương tự, ta xét ảnh hưởng của nhiệt độ trung bình ngày ảnh hưởng đến P_{min} của ngày dự báo. Kết quả ta xác định được 4 ngày mà T_{tb} ảnh hưởng lớn đến P_{min} của ngày dự báo, đó là các ngày $d-1, d-3, d-5$ và $d-9$. Tổng hợp lại ta có mô hình được lựa chọn để dự báo giá trị P_{min} của ngày thứ d sẽ gồm 09 số liệu từ quá khứ:

$$P_{min}(d) = f \left(\begin{matrix} P_{min}(d-1), P_{min}(d-2), P_{min}(d-10), \\ P_{max}(d-1), P_{max}(d-2), T_{tb}(d-1), T_{tb}(d-3), T_{tb}(d-5), T_{tb}(d-9) \end{matrix} \right) \quad (4.23)$$

Hay

$$\begin{aligned} P_{min}(d) = & 0,462 \cdot P_{min}(d-1) + 0,167 \cdot P_{min}(d-2) + 0,149 \cdot P_{min}(d-10) + \\ & + 0,64 \cdot P_{max}(d-1) + 0,32 \cdot P_{max}(d-2) + \\ & + 0,088 \cdot T_{tb}(d-1) + 0,064 \cdot T_{tb}(d-3) + 0,067 \cdot T_{tb}(d-5) + 0,066 \cdot T_{tb}(d-9) \end{aligned} \quad (4.24)$$

Bảng 4.3. Kết quả ước lượng công suất đáy phụ tải (P_{min}) dùng phương pháp khai triển tuyến tính SVD

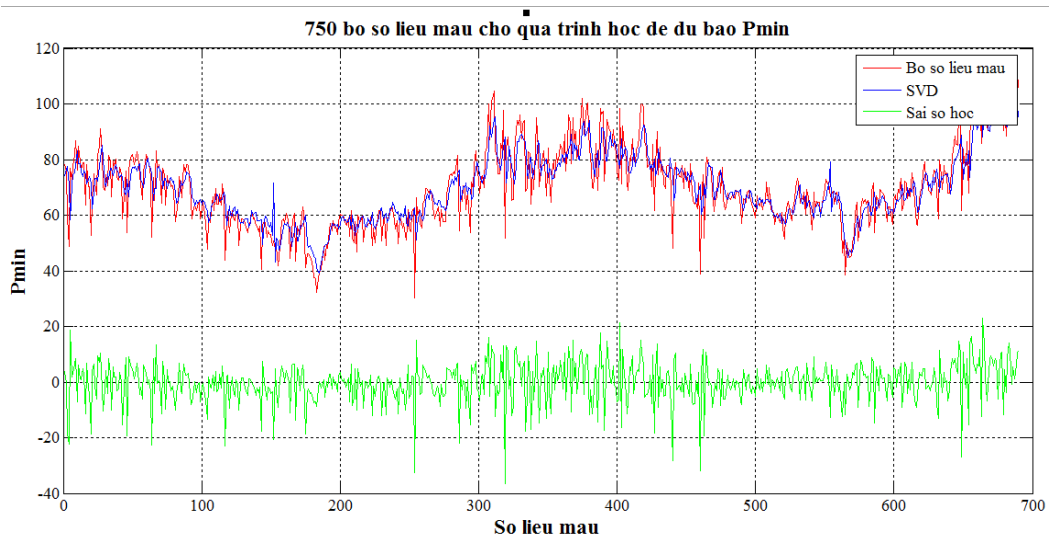
Quá trình	Sai số	SVD
Học	$MRE = \frac{1}{N} \left \frac{y-d}{y} \right \cdot 100\%$	5,33%
	$MAE = \frac{1}{N} y-d $	8,03
	$MaxAE = \max y-d $	36,36
Kiểm tra	$MRE = \frac{1}{N} \left \frac{y-d}{y} \right \cdot 100\%$	6,73%

	$MAE = \frac{1}{N} y - d $	11,52
	$MaxAE = \max y - d $	73,36

Bảng 4.3 là kết quả ước lượng công suất đầy phụ tải dùng phương pháp khai triển thành phần tuyến tính SVD, qua bảng 3.2 ta nhận thấy sai số học và sai số kiểm tra tương đối nhỏ cỡ 6%.



Hình 4.7. Đồ thị sai số học khi khai triển thành phần tuyến tính cho mô hình dự báo P_{min}



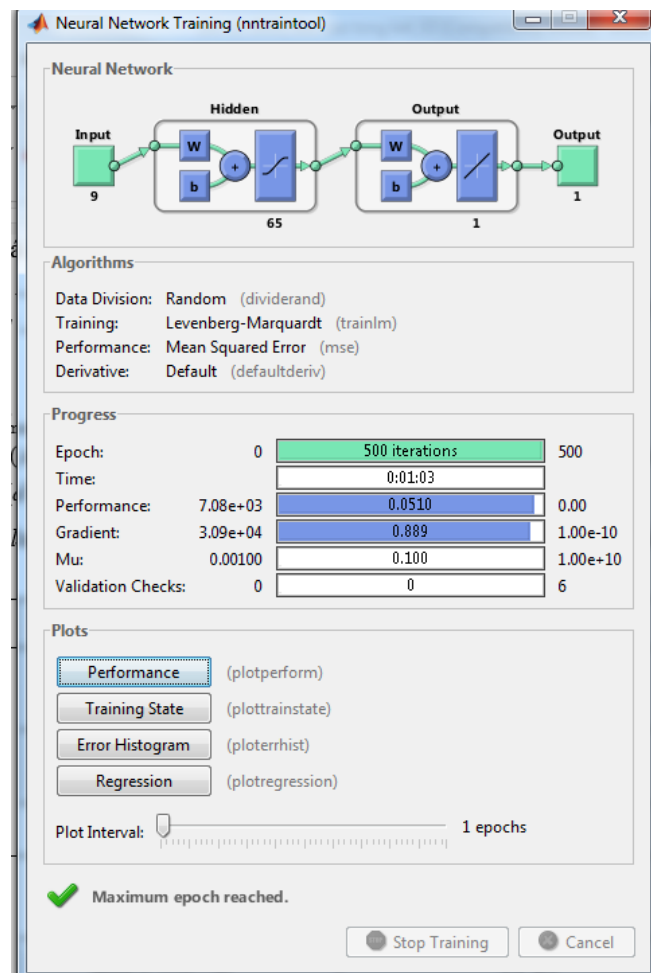
Hình 4.8. Đồ thị sai số quá trình kiểm tra khi khai triển thành phần tuyến tính cho mô hình dự báo P_{min}

Hình 4.7 và hình 4.8 thể hiện quá trình học và quá trình kiểm tra khi triển khai thành phần tuyến tính cho mô hình dự báo P_{min} , đường màu đỏ là bộ số liệu mẫu,

đường màu xanh nước biển là kết quả học/kiểm tra, đường màu xanh là cây là sai số quá trình học/kiểm tra.

4.2.3.2. Ước lượng thành phần phi tuyến bằng mạng nơ-rôn MLP

Thực hiện các bước theo mục 4.1.3.1.b, ta có kết quả như sau: mô hình mạng MLP được xây dựng có cấu trúc 09 đầu vào ứng với 09 đại lượng quá khứ (công thức 4.24), 1 lớp ẩn với 500 nơ-rôn và 1 đầu ra ứng với giá trị cần dự báo cho ngày tiếp theo. Mô hình huấn luyện mạng MLP để ước lượng P_{min} như hình 4.9.



Hình 4.9. Quy trình học của mạng MLP khi ước lượng P_{min}

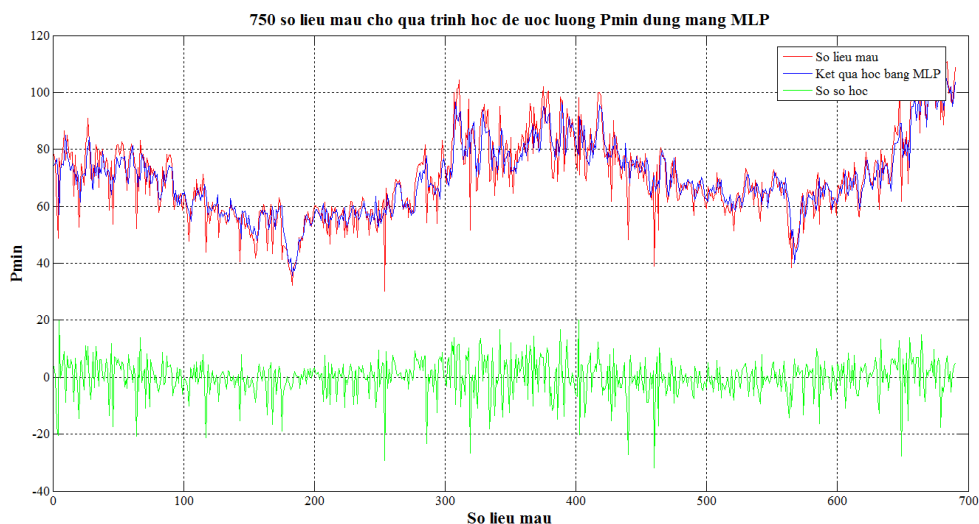
Bảng 4.4 là kết quả sai số học và sai số kiểm tra khi ước lượng P_{min} bằng mạng nơ-rôn MLP, qua số liệu trên bảng ta thấy sai số khi ước lượng P_{min} tương đối nhỏ (Sai số học 7,34%; sai số kiểm tra 10,26%)

Bảng 4.4. Kết quả sai số học và sai số kiểm tra khi ước lượng P_{min} bằng MLP

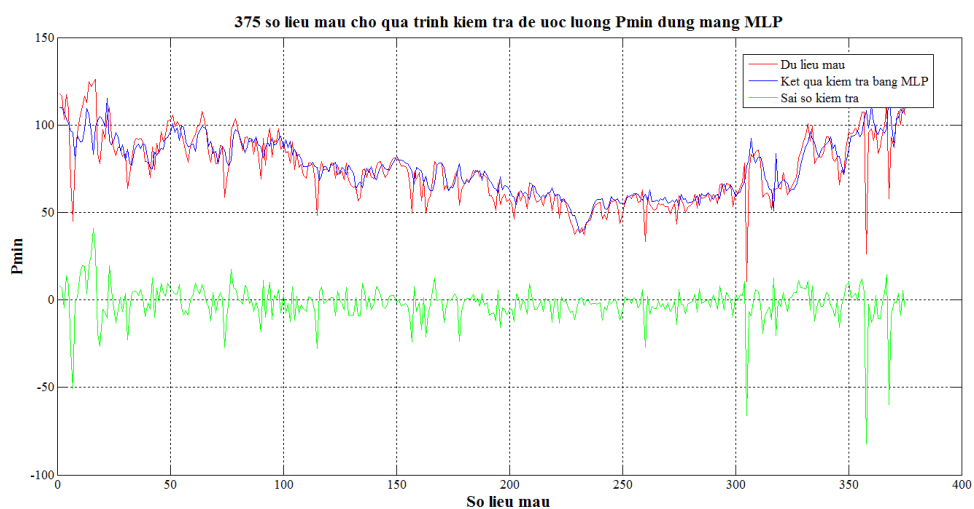
Quá trình	Sai số	Sai số của mạng MLP
Học	$MRE = \frac{1}{N} \left \frac{y-d}{y} \right \cdot 100\%$	7,34%

Kiểm tra	$MAE = \frac{1}{N} y - d $	11,00
	$MaxAE = \max y - d $	39,91
	$MRE = \frac{1}{N} \left \frac{y - d}{y} \right \cdot 100\%$	10,26
	$MAE = \frac{1}{N} y - d $	15,00
	$MaxAE = \max y - d $	45,38

Hình 3.16, 3.17 thể hiện quá trình học và quá trình kiểm tra khi triển khai thành phần tuyến tính cho mô hình dự báo P_{min} , đường màu đỏ là bộ số liệu mẫu, đường màu xanh nước biển là kết quả học/kiểm tra, đường màu xanh là cây là sai số quá trình học/kiểm tra.



Hình 4.10. Đồ thị sai số quá trình học khi ước lượng P_{min} bằng mạng MLP



Hình 4.11. Đồ thị sai số quá trình kiểm tra khi ước lượng P_{min} bằng mạng MLP

Dựa trên các kết quả thu được ta có thể kết luận rằng nhiệm vụ dự báo đỉnh và đáy phụ tải ngắn hạn đã được thực hiện khá hiệu quả nhờ vào việc sử dụng mạng nơ-ron MLP. Sai số trung bình khoảng 7% là mức sai số không lớn, thực tế chấp nhận được. Tuy nhiên nếu sử dụng các mô hình mạng nơ-ron khác, ta còn có thể tiếp tục giảm sai số này xuống thấp hơn.

4.3. Kết luận chương IV

Trong chương IV tác giả trình bày kết quả của ứng dụng việc xây dựng khối tuyến tính và khối phi tuyến trong mô hình lai dự báo công suất phụ tải ngày. Bộ số liệu sử dụng trong mô hình (P_{\max} , P_{\min}) do Công ty điện lực thành phố Hạ Long, tỉnh Quảng Ninh cung cấp và nhiệt độ trung bình ngày lấy tại trạm quan trắc khí tượng thủy văn thành phố Hạ Long, tỉnh Quảng Ninh cung cấp, thời gian lấy từ tháng 01/6/2016 đến 30/6/2019. Với 1150 số liệu mẫu, chia thành bộ số liệu học với 750 bộ số liệu đầu tiên (tương đương 2/3 bộ số liệu mẫu) và 375 số liệu còn lại (tương đương 1/3 bộ số liệu mẫu) làm bộ số liệu kiểm tra mô hình.

Tổng hợp khối ước lượng tuyến tính và khối ước lượng phi tuyến ta xây dựng lên mô hình dự báo công suất đỉnh phủ tải điện (P_{\max}) và công suất đáy phụ tải điện (P_{\min}) trong ngày.

Khối tuyến tính được xác định hệ số bằng phương pháp phân tích giá trị kì dị (khai triển theo hệ số tuyến tính) SVD kết hợp với phương pháp bình phương cực tiểu. Tác giả xây dựng khối tuyến tính của mô hình dự báo công suất đỉnh phụ tải P_{\max} với 09 số liệu quá khứ: tương ứng với các giá trị phụ thuộc P_{\max} , P_{\min} , T_{tb} . Tương tự xây dựng mô hình dự báo P_{\min} cũng với 09 số liệu quá khứ.

Khối phi tuyến của mô hình dự báo cho P_{\max} , P_{\min} được xây dựng bằng mạng nơ-ron truyền thẳng nhiều lớp MLP. Với mô hình báo cho P_{\max} , mạng MLP có 09 đầu vào tương đương với các đầu vào của khối tuyến tính Số lớp ẩn là 1, số nơ-ron ẩn là 65. Tương tự mô hình báo cho P_{\min} , mạng MLP có 09 đầu vào. Số lớp ẩn là 1, số nơ-ron ẩn là 65; hàm truyền được sử dụng chung cho cả hai mô hình là tansig, quá trình học của thuật toán sử dụng Levenberg – Marquardt với 500 lần lặp.

Qua quá trình xây dựng mô hình dự báo kết quả thu được sau khi tổng hợp 2 khối ước lượng tuyến tính và phi tuyến ta thu được kết quả ước lượng được đánh giá trên 3 phương pháp trung bình của sai số tương đối, trung bình của sai số tuyệt đối và giá trị lớn nhất của sai số tuyệt đối để đánh giá chất lượng quá trình học, quá trình kiểm tra của từng khối ước lượng và đánh giá tổng hợp chung. Theo bảng 4.2 và bảng 4.4. Từ kết quả và bảng đánh giá nhận xét ta thấy rằng mô hình lai giữa mạng MLP và thuật toán SVD cho giá trị dự báo tốt hơn so với mô hình mạng MLP độc lập và mô hình tuyến tính độc lập.

KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết luận

Luận văn “*Ứng dụng mạng nơ-rôn nhân tạo dự báo phụ tải điện thành phố Hạ Long – tỉnh Quảng Ninh*” dựa trên những nghiên cứu lý thuyết mạng nơ-rôn nói riêng và các mô hình xử lý tín hiệu tiên tiến nói chung để đề xuất ứng dụng mô hình lai vào bài toán dự báo công suất phụ tải điện với độ chính xác cao.

Luận văn đã thực hiện được các công việc cụ thể như sau:

1. Tìm hiểu về các mô hình thu thập, xử lý, dự báo công suất phụ tải điện.
2. Nghiên cứu về khả năng ứng dụng mạng nơ- rôn trong xây dựng các mô hình dự báo phi tuyến.
3. Đề xuất ứng dụng mô hình lai gồm một khối tuyến tính mắc song song với một khối phi tuyến (sử dụng mạng MLP) để ước lượng tốt hơn trong dự báo công suất phụ tải điện chỉ dựa trên các kết quả đo công suất phụ tải trong ngày ở quá khứ từ số tổng hợp của Công ty điện lực.
4. Đánh giá, lựa chọn các thông số quá khứ cho mô hình dự báo trên cơ sở các hệ số khai triển tuyến tính.
5. Triển khai, thử nghiệm mô hình dự báo công suất đỉnh phụ tải và công suất đáy phụ tải trong ngày cho khu vực Thành phố Hạ Long, tỉnh Quảng Ninh.
6. Tính toán mô phỏng để thử nghiệm độ chính xác của mô hình được đề xuất.

2. Kiến nghị

Trong luận văn này, phương pháp đề xuất đã đạt được một số kết quả bước đầu khả quan, áp dụng vào dự báo công suất phụ tải điện tiêu thụ trong ngày của thành phố Hạ Long, tỉnh Quảng Ninh. Tuy nhiên, trong lĩnh vực nghiên cứu còn một số tồn tại cần được tiếp tục được nghiên cứu và giải quyết: cần đưa thêm vào mô hình mạng nơron các thông số thời tiết có ảnh hưởng lớn đến phụ tải (lượng mưa, số giờ nắng...), đặc điểm đặc thù của các ngày đặc biệt (ngày lễ, tết...) để có thể nâng cao hơn độ chính xác, mức độ tin cậy của dự báo.

Một số hướng để phát triển đề tài:

1. Xây dựng mô hình dự báo 24h trong ngày.
2. Thu thập thêm số liệu để tăng độ tin cậy của mô hình; Xét thêm các yếu tố như (lượng mưa, số giờ nắng).
3. Sử dụng mô hình mạng nơ-rôn khác ước lượng thành phần phi tuyến trong mô hình lai như SVM, Deep learning,.. để cho kết quả sai số tốt hơn.

DANH MỤC CÁC TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1]. Trần Hoài Linh (2009), Mạng nơ-rôn và ứng dụng trong xử lý tín hiệu, NXB Bách Khoa.
- [2]. Vũ Thanh Nguyên (2006), Ứng dụng logic mờ, mạng nơron mờ, hệ các luật mờ phân tích dự báo các mặt hàng chiến lược, Hội thảo khoa học Hệ mờ, mạng nơron và ứng dụng, lần 1, Hà nội 8-9/11/2006.
- [3]. Nguyễn Quân Nhu (2009), Luận văn tốt nghiệp cao học, Trường ĐHBK-HN.
- [4]. Nguyễn Đình Thúc (2000), Mạng nơron phương pháp và ứng dụng.
- [5]. Trần Kỳ Phúc (2006), Nghiên cứu ứng dụng mạng Nơ ron nhân tạo trong dự báo ngắn hạn phụ tải điện (giai đoạn I), Viện năng lượng.
- [6]. Trần Đình Long (1998), Quy hoạch phát triển năng lượng và điện lực, Trường ĐH Bách khoa Hà Nội.
- [7]. Trần Kỳ Phúc, Monique Polit, Nguyễn Tiên Phong, Lê Thị Thanh Hà (2006), Mạng Nơ ron – dự báo phụ tải điện ngắn hạn ứng dụng tại Hà Nội, ảnh hưởng của các thông số thời tiết, tạp chí Tự động hoá ngày nay.
- [8]. Nguyễn Hoàng Phương, Bùi Công Cường, Nguyễn Doãn Phước, Phan Xuân Minh, Chu Văn Hỷ (2009), Hệ mờ và ứng dụng, NXB Khoa học kỹ thuật.
- [9]. Đinh Văn Nhượng (2012), trường Đại học Sao Đỏ, Một số ứng dụng mạng nơron xây dựng mô hình nhận dạng và dự báo, NXB Khoa học và kỹ thuật.
- [10]. LM Markovits, Trần Đình Long, Bùi Ngọc Thư, Bùi Thiên Dụ, Hà Học Trạc (dịch, 2007), Các chế độ của hệ thống năng lượng, NXB KH-KT.
- [11]. Nguyễn Lân Tráng (2007), Quy hoạch phát triển hệ thống điện, NXB Khoa học và kỹ thuật.

Tiếng Anh

- [12]. S. Osowski, K. Siwek (1998), Selforganising neural networks for short term load forecasting in power system, EANN'98, pp. 253 - 256, Gibraltar.
- [13]. S. Osowski, K. Siwek (1999), The selforganising neural networks approach to load forecasting in power system. IJCNN'99, vol.1, pp. 1032 - 1036, Washington.
- [14]. Pauli Murto (1998), Neural network models for short-term load forecasting, Helsinki university of technology.
- [15]. E.G.Swee, Terence, S.Elangovan: Wavelets Based Analysis of Non-Uniformly Sampled Data Power Forecasting. Department of Electrical engineering, National University of Singapore.
- [16]. D.C. Park, M.A. El-Sharkawi, R.J. Marks, L.E. Atlas and M.J. Damborg (1991), Electric Load Forecasting Using An Artificial Neural Network. IEEE Transactions on Power Systems, Vol. 6, No. 2, pp 442-449.
- [17]. A. Wang, B. Ramsay (1998), A neural network based estimator for electricity spot-pricing with particular reference to weekend and public holiday, Neurocomputing, vol. 23, pp. 47 - 57.
- [18]. J. Zielinski, N. Hatziargyriou, J. Pecas Lopers (1996), AI in power systems - selected applications, Colloquium in Artificial Intelligence, CAI'96, Poland.

- [19]. M. Gavrilas (2010), Neural Network Based Forecasting for Electricity Markets. Technical University of Iasi, Romania.
- [20]. Hyan Xu, Wei Ji Chen (1999), Artificial Neural Network Short-term Electrical Load Forecasting Techniques. IEEE TENCON, pp.1458-1461.
- [21]. A.K.Sinha (2000), Short Term Load Forecasting Using Artificial Neural Networks, In Proc. of IEEE Int. Conf on Industrial Technology, Goa, India, vol. 1, pp. 548-553.

Các trang Website

- [22]. <https://petrotimes.vn/nang-luong-toan-cau-va-nhung-con-so-dang-chu-y-506971.html>, cập nhật 02/6/2019.
- [23]. <https://fptshop.com.vn/tin-tuc/danh-gia/cong-nghe-ai-la-gi-tim-hieu-ve-cong-nghe-ai-63382>, cập nhật 03/06/2019
- [24]. <https://www.vinmec.com/vi/co-the-nguoi-te-bao-than-kinh-165/>, cập nhật ngày 10/06/2019.
- [25]. <http://kdientu.duytan.edu.vn/media/50176/ly-thuyet-mang-neural.pdf>, cập nhật ngày 25/07/2019.
- [26]. <http://bis.net.vn/forums/t/507.asp>, cập nhật ngày 15/07/2019.
- [27]. <https://khoaoc.tv/ai-tri-tue-nhan-tao>, cập nhật ngày 26/07/2019.
- [28]. <https://cunghocvui.com/bai-viet/gioi-thieu-chung-kien-thuc-ve-he-than-kinh-sinh-hoc-lop-8.html>. cập nhật ngày 27/07/2019.

PHỤ LỤC

```

%%% ===== READ DATA DIEN LUC THANH PHO HA LONG ===== %%%
clear all
clc
Pmin_QN=xlsread('Tong hop so lieu.xls','d3:d1127');
Pmax_QN=xlsread('Tong hop so lieu.xls','f3:f1127');
Ttb_QN=xlsread('Tong hop so lieu.xls','i3:i1127');

Data_QN_All=[Pmin_QN Pmax_QN Ttb_QN];
save Data_QN_All;
%%%% XAY DUNG MA TRAN 60 COT
clear all
clc
load Data_QN_All
K=60;
Nmax = length(Pmax_QN);
BigA_Pmax = zeros(Nmax-K,K);
for i=1:K
    BigA_Pmax(:,i)=Pmax_QN((K-i+1):(Nmax-i));
end;
B_Pmax = Pmax_QN((K+1):Nmax);

BigA_Pmin = zeros(Nmax-K,K);
for i=1:K
    BigA_Pmin(:,i)=Pmin_QN((K-i+1):(Nmax-i));
end;
B_Pmin = Pmin_QN((K+1):Nmax);

BigA_Tmax = zeros(Nmax-K,K);
for i=1:K
    BigA_Tmax(:,i)=Tmax_QN((K-i+1):(Nmax-i));
end;
B_Tmax = Tmax_QN((K+1):Nmax);

BigA_Ttb = zeros(Nmax-K,K);
for i=1:K
    BigA_Ttb(:,i)=Ttb_QN((K-i+1):(Nmax-i));
end;
B_Ttb = Ttb_QN((K+1):Nmax);
save Data_QN_SVD

%%%% Linear model for Pmax
%%% Phan tich SVD de xac dinh yeu to anh huong
%Pmax=f(Pmax)
clear all
clc
load Data_QN_SVD;

```

```

index=1:60;
CoeffAll_max=[]; % Tao ma tran rong de chua cac he so
for K=60:-1:5
    K
    Coeff=BigA_Pmax\B_Pmax
    a=abs(Coeff);
    [val,ind]=min(a);
    vitri_min=ind % vi tri xac dinh min
    Stt=index(ind) % So thu tu trong vector 60 phan tu
    tmp2=zeros(60,1);
    tmp2(index)=Coeff;
    CoeffAll_max=[CoeffAll_max tmp2];% Tong hop thanh ma tran chua cac he so
    if (ind==1)
        index=index(2:end);
        BigA_Pmax=BigA_Pmax(:,2:end);
    else
        if (ind==length(a))
            index=index(1:end-1);
            BigA_Pmax=BigA_Pmax(:,1:end-1);
        else
            tmpind=[1:ind-1,ind+1:length(a)];
            index=index(tmpind);
            BigA_Pmax=BigA_Pmax(:,tmpind);
        end;
    end;
    index
end;

%%%%%%%%%%%%%% Linear model for Pmax
%%% Phan tich SVD de xac dinh yeu to anh huong
%Pmax=f(Pmin)
clear all
clc
load Data_QN_SVD;
index=1:60;
CoeffAll_max=[]; % Tao ma tran rong de chua cac he so
for K=60:-1:5
    K
    Coeff=BigA_Pmax\B_Pmin
    a=abs(Coeff);
    [val,ind]=min(a);
    vitri_min=ind % vi tri xac dinh min
    Stt=index(ind) % So thu tu trong vector 60 phan tu
    tmp2=zeros(60,1);
    tmp2(index)=Coeff;
    CoeffAll_max=[CoeffAll_max tmp2];% Tong hop thanh ma tran chua cac he so
    % loai cot ind ra khoi bang BigA
    if (ind==1)

```

```

    index=index(2:end);
    BigA_Pmax=BigA_Pmax(:,2:end);
else
    if (ind==length(a))
        index=index(1:end-1);
        BigA_Pmax=BigA_Pmax(:,1:end-1);
    else
        tmpind=[1:ind-1,ind+1:length(a)];
        index=index(tmpind);
        BigA_Pmax=BigA_Pmax(:,tmpind);
    end;
end;
index
% pause;
end;

%%%%%%%%%%%%%% Linear model for Pmax
%%% Phan tich SVD de xac dinh yeu to anh huong
%Pmax=f(Ttb)
clear all
clc
load Data_QN_SVD;
index=1:60;
CoeffAll_max=[]; % Tao ma tran rong de chua cac he so
for K=60:-1:5
    K
    Coeff=BigA_Pmax\B_Ttb
    a=abs(Coeff);
    [val,ind]=min(a);
    vitri_min=ind % vi tri xac dinh min
    Stt=index(ind) % So thu tu trong vector 60 phan tu
    tmp2=zeros(60,1);
    tmp2(index)=Coeff;
    CoeffAll_max=[CoeffAll_max tmp2];% Tong hop thanh MT chua cac he so
    if (ind==1)
        index=index(2:end);
        BigA_Pmax=BigA_Pmax(:,2:end);
    else
        if (ind==length(a))
            index=index(1:end-1);
            BigA_Pmax=BigA_Pmax(:,1:end-1);
        else
            tmpind=[1:ind-1,ind+1:length(a)];
            index=index(tmpind);
            BigA_Pmax=BigA_Pmax(:,tmpind);
        end;
    end;
end;
index

```

```

end;

%% Ma tran tong hop sau khi thuc hien SVD
clear all
clc
load Data_QN_SVD
BigBigA_Pmax_SVD = [BigA_Pmax(:,[1 2 6]) BigA_Pmin(:,[1 2 12]) BigA_Ttb(:,[1
3 5 9])];
whos BigBigA_Pmax_SVD
TestNr = 375; % Su dung 30% bo so lieu cuoi de kiem tra mo hinh

BigBigA_Pmax_Learn = BigBigA_Pmax_SVD(1:(end-TestNr),:);
B_Pmax_Learn = B_Pmax(1:(end-TestNr));
BigBigA_Pmax_Test = BigBigA_Pmax_SVD((end-TestNr+1):end,:);
B_Pmax_Test = B_Pmax((end-TestNr+1):end);
PmaxLinearCoeff_Learn = BigBigA_Pmax_Learn\B_Pmax_Learn;

% He so hoc cua mo hinh tuyen tinh
ErrorLearnAfterLinear_Pmax = B_Pmax_Learn -
BigBigA_Pmax_Learn*PmaxLinearCoeff_Learn; % Sai so con lai sau khi thuc hien
ErrorTestAfterLinear_Pmax = B_Pmax_Test -
BigBigA_Pmax_Test*PmaxLinearCoeff_Learn; % Sai so con lai sau khi thuc hien
%% Tinh sai so hoc
MAE_Learn_Linear_Pmax_QN_SVD=mean(abs(ErrorLearnAfterLinear_Pmax))
MRE_Learn_Linear_Pmax_QN_SVD=mean(abs(ErrorLearnAfterLinear_Pmax./B_P
max_Learn)*100)
Maxae_Learn_Linear_Pmax_QN_SVD=max(abs(ErrorLearnAfterLinear_Pmax))

%% Tinh sai so Test
MAE_Test_Linear_Pmax_QN_SVD=mean(abs(ErrorTestAfterLinear_Pmax))
MRE_Test_Linear_Pmax_QN_SVD=mean(abs(ErrorTestAfterLinear_Pmax./B_Pmax
_Test)*100)
Maxae_Test_Linear_Pmax_QN_SVD=max(abs(ErrorTestAfterLinear_Pmax))
figure(1)
tt=length(B_Pmax_Learn);
plot((1:tt),B_Pmax_Learn,'r',(1:tt),BigBigA_Pmax_Learn*PmaxLinearCoeff_Learn,'-
b',(1:tt),ErrorLearnAfterLinear_Pmax,'-g');
figure(2)
tt1=length(B_Pmax_Test);
plot((1:tt1),B_Pmax_Test,'r',(1:tt1),BigBigA_Pmax_Test*PmaxLinearCoeff_Learn,'-
b',(1:tt1),ErrorTestAfterLinear_Pmax,'-g');

%%%%% LINEAR MODEL FOR Pmin
%%%% Phan tich SVD de xac dinh yeu to anh huong
%Pmin=f(Pmin)
clear all
clc
load Data_QN_SVD;

```

```

index=1:60;
CoeffAll_max=[]; % Tao ma tran rong de chua cac he so
for K=60:-1:5
    K
    Coeff=BigA_Pmin\B_Pmin
    a=abs(Coeff);
    [val,ind]=min(a);
    %index(ind);
    vitri_min=ind % vi tri xac dinh min
    Stt=index(ind) % So thu tu trong vector 60 phan tu
    tmp2=zeros(60,1);
    tmp2(index)=Coeff;
    CoeffAll_max=[CoeffAll_max tmp2];% Tong hop thanh ma tran chua cac he so
    % loai cot ind ra khoi bang BigA
    if (ind==1)
        index=index(2:end);
        BigA_Pmin=BigA_Pmin(:,2:end);
    else
        if (ind==length(a))
            index=index(1:end-1);
            BigA_Pmin=BigA_Pmin(:,1:end-1);
        else
            tmpind=[1:ind-1,ind+1:length(a)];
            index=index(tmpind);
            BigA_Pmin=BigA_Pmin(:,tmpind);
        end;
    end;
    index
    %pause;
end;

%%% Linear model for Pmin
%%% Phan tich SVD de xac dinh yeu to anh huong
%Pmin=f(Pmax)
clear all
clc
load Data_QN_SVD;
index=1:60;
CoeffAll_max=[]; % Tao ma tran rong de chua cac he so
for K=60:-1:5
    K
    Coeff=BigA_Pmin\B_Pmax
    a=abs(Coeff);
    [val,ind]=min(a);
    vitri_min=ind % vi tri xac dinh min
    Stt=index(ind) % So thu tu trong vector 60 phan tu
    tmp2=zeros(60,1);

```



```

tmp2(index)=Coeff;
CoeffAll_max=[CoeffAll_max tmp2];% Tong hop thanh ma tran chua cac he so
if (ind==1)
    index=index(2:end);
    BigA_Pmin=BigA_Pmin(:,2:end);
else
    if (ind==length(a))
        index=index(1:end-1);
        BigA_Pmin=BigA_Pmin(:,1:end-1);
    else
        tmpind=[1:ind-1,ind+1:length(a)];
        index=index(tmpind);
        BigA_Pmin=BigA_Pmin(:,tmpind);
    end;
end;
index
end;
%%% Phan tich SVD de xac dinh yeu to anh huong
%Pmin=f(Ttb)
clear all
clc
load Data_QN_SVD;
index=1:60;
CoeffAll_max=[]; % Tao ma tran rong de chua cac he so
for K=60:-1:5
    K
    Coeff=BigA_Pmin\B_Ttb
    a=abs(Coeff);
    [val,ind]=min(a);
    vitri_min=ind % vi tri xac dinh min
    Stt=index(ind) % So thu tu trong vector 60 phan tu
    tmp2=zeros(60,1);
    tmp2(index)=Coeff;
    CoeffAll_max=[CoeffAll_max tmp2];% Tong hop thanh ma tran chua cac he so
    if (ind==1)
        index=index(2:end);
        BigA_Pmin=BigA_Pmin(:,2:end);
    else
        if (ind==length(a))
            index=index(1:end-1);
            BigA_Pmin=BigA_Pmin(:,1:end-1);
        else
            tmpind=[1:ind-1,ind+1:length(a)];
            index=index(tmpind);
            BigA_Pmin=BigA_Pmin(:,tmpind);
        end;
    end;
end;
index

```

```

end;
%% Ma tran tong hop sau khi thuc hien SVD
clear all
clc
load Data_QN_SVD
BigBigA_Pmin_SVD = [BigA_Pmin(:,[1 2 10]) BigA_Pmax(:,[1 2]) BigA_Ttb(:,[1 3
5 9])];
TestNr = 375; % Su dung 30% bo so lieu cuoi de kiem tra mo hinh
BigBigA_Pmin_Learn = BigBigA_Pmin_SVD(1:(end-TestNr),:);
B_Pmin_Learn = B_Pmin(1:(end-TestNr));
BigBigA_Pmin_Test = BigBigA_Pmin_SVD((end-TestNr+1):end,:);
B_Pmin_Test = B_Pmin((end-TestNr+1):end);
PminLinearCoeff_Learn = BigBigA_Pmin_Learn\B_Pmin_Learn;

%% He so hoc cua mo hinh tuyen tinh
ErrorLearnAfterLinear_Pmin = B_Pmin_Learn -
BigBigA_Pmin_Learn*PminLinearCoeff_Learn; % Sai so con lai sau khi thuc hien
SVD
ErrorTestAfterLinear_Pmin = B_Pmin_Test -
BigBigA_Pmin_Test*PminLinearCoeff_Learn; % Sai so con lai sau khi thuc hien
%% Tinh sai so hoc
MAE_Learn_Linear_Pmin_QN_SVD=mean(abs(ErrorLearnAfterLinear_Pmin))
MRE_Learn_Linear_Pmin_QN_SVD=mean(abs(ErrorLearnAfterLinear_Pmin./B_Pm
in_Learn)*100)
Maxae_Learn_Linear_Pmin_QN_SVD=max(abs(ErrorLearnAfterLinear_Pmin))

%% Tinh sai so Test
MAE_Test_Linear_Pmin_QN_SVD=mean(abs(ErrorTestAfterLinear_Pmin))
MRE_Test_Linear_Pmin_QN_SVD=mean(abs(ErrorTestAfterLinear_Pmin./B_Pmin_
Test)*100)
Maxae_Test_Linear_Pmin_QN_SVD=max(abs(ErrorTestAfterLinear_Pmin))
figure(1)
tt=length(B_Pmin_Learn);
plot((1:tt),B_Pmin_Learn,'-r',(1:tt),BigBigA_Pmin_Learn*PminLinearCoeff_Learn,'-
b',(1:tt),ErrorLearnAfterLinear_Pmin,'-g');
figure(2)
tt1=length(B_Pmin_Test);
plot((1:tt1),B_Pmin_Test,'-r',(1:tt1),BigBigA_Pmin_Test*PminLinearCoeff_Learn,'-
b',(1:tt1),ErrorTestAfterLinear_Pmin,'-g');

%% UOC LUONG THANH PHAN PHI TUYEN
%% Uoc luong Pmax
clear all
clc
load Data_QN_SVD
BigBigA_Pmax_SVD = [BigA_Pmax(:,[1 2 6]) BigA_Pmin(:,[1 2 12]) BigA_Ttb(:,[1
4 5])];
whos BigBigA_Pmax_SVD

```

```

TestNr = 375; % Su dung 30% bo so lieu cuoi de kiem tra mo hinh
BigBigA_Pmax_Learn = BigBigA_Pmax_SVD(1:(end-TestNr),:);
B_Pmax_Learn = B_Pmax(1:(end-TestNr));
BigBigA_Pmax_Test = BigBigA_Pmax_SVD((end-TestNr+1):end,:);
B_Pmax_Test = B_Pmax((end-TestNr+1):end);
PmaxLinearCoeff_Learn = BigBigA_Pmax_Learn\B_Pmax_Learn;
ErrorLearnAfterLinear_Pmax = B_Pmax_Learn -
BigBigA_Pmax_Learn*PmaxLinearCoeff_Learn; % Sai so con lai sau khi thuc hien
SVD
ErrorTestAfterLinear_Pmax = B_Pmax_Test -
BigBigA_Pmax_Test*PmaxLinearCoeff_Learn; % Sai so con lai sau khi thuc hien
BestResultError = inf;
for HiddenNeuron = 5:20:65
    PmaxNN = feedforwardnet(HiddenNeuron,'trainlm');

    RepeatCount = 5;
    for i = 1:RepeatCount
        net2 = configure(PmaxNN, BigBigA_Pmax_Learn',
ErrorLearnAfterLinear_Pmax');
        net2.divideParam.trainRatio =1;
        net2.divideParam.valRatio =0;
        net2.divideParam.testRatio =0;
        net2.trainParam.min_grad=1e-10;
        NumberEpoch = 500;
        net2.trainParam.epochs = NumberEpoch;
        [net3,tr] = train(net2, BigBigA_Pmax_Learn', ErrorLearnAfterLinear_Pmax');

        ytest_MLP=sim(net3, BigBigA_Pmax_Test');
        err_test = sum((ytest_MLP - ErrorTestAfterLinear_Pmax').^2);
        if (err_test<BestResultError)
            BestResultError = err_test
            best_net = net3;
        end;
    end;
end
ylearn_MLP=sim(best_net, BigBigA_Pmax_Learn');
ytest_MLP=sim(best_net, BigBigA_Pmax_Test');
err_learn = sqrt(mean((ylearn_MLP - ErrorLearnAfterLinear_Pmax').^2)
err_test = sqrt(mean((ytest_MLP - ErrorTestAfterLinear_Pmax').^2))
return

% Sai so hoc NL
MAE_Learn_NN_Pmax_MLP = mean(abs(err_learn))
MRE_Learn_NN_Pmax_MLP = mean(abs(err_learn./B_Pmax_Learn)*100)
MaxAE_Learn_NN_Pmax_MLP = max(abs(err_learn))

% Sai so kiem tra NL
MAE_Test_NN_Pmax_MLP=mean(abs(err_test))

```

```

MRE_Test_NN_Pmax_MLP=mean(abs(err_test./B_Pmax_Test)*100)
MaxAE_Test_NN_Pmax_MLP=max(abs(err_test))
figure(3)
tt=length(ylearn_MLP);
plot((1:tt),ylearn_MLP,'-r',(1:tt),B_Pmax_Learn,'-b',(1:tt),ylearn_MLP -
B_Pmax_Learn,'-g');
figure(4)
tt1=length(ytest_MLP);
plot((1:tt1),ytest_MLP,'-r',(1:tt1),B_Pmax_Test,'-b',(1:tt1),ytest_MLP -
B_Pmax_Test,'-g');

%%% UOC LUONG THANH PHAN PHI TUYEN
%%% Uoc luong Pmin

clear all
clc
load Data_QN_SVD
BigBigA_Pmin_SVD = [BigA_Pmin(:,[1 2 10]) BigA_Pmax(:,[1 2]) BigA_Ptb(:,[1 3
5 9])];
whos BigBigA_Ptb_SVD
TestNr = 375; % Su dung 30% bo so lieu cuoi de kiem tra mo hinh

BigBigA_Pmin_Learn = BigBigA_Pmin_SVD(1:(end-TestNr),:);
B_Pmin_Learn = B_Pmin(1:(end-TestNr));

BigBigA_Pmin_Test = BigBigA_Pmin_SVD((end-TestNr+1):end,:);
B_Pmin_Test = B_Pmin((end-TestNr+1):end);

PminLinearCoeff_Learn = BigBigA_Pmin_Learn\B_Pmin_Learn;% He so hoc cua mo
%hinh tuyen tinh

ErrorLearnAfterLinear_Pmin = B_Pmin_Learn -
BigBigA_Pmin_Learn*PminLinearCoeff_Learn; % Sai so con lai sau khi thuc %hien
SVD
ErrorTestAfterLinear_Pmin = B_Pmin_Test -
BigBigA_Pmin_Test*PminLinearCoeff_Learn; % Sai so con lai sau khi thuc hien
SVD test tren cac mau so lieu con lai
% t=1:1504;
% t=1:1504;
% Tim so lop an N
% results = [];
BestResultError = inf;
for HiddenNeuron = 5:20:65
    PminNN = feedforwardnet(HiddenNeuron,'trainlm');
    RepeatCount = 5;
    for i = 1:RepeatCount
        net2 = configure(PminNN, BigBigA_Pmin_Learn',
ErrorLearnAfterLinear_Pmin');

```

```

net2.divideParam.trainRatio =1;
net2.divideParam.valRatio =0;
net2.divideParam.testRatio =0;
net2.trainParam.min_grad=1e-10;
NumberEpoch = 500;
net2.trainParam.epochs = NumberEpoch;
[net3,tr] = train(net2, BigBigA_Pmin_Learn', ErrorLearnAfterLinear_Pmin');

ytest_MLP=sim(net3, BigBigA_Pmin_Test');
err_test = sum((ytest_MLP - ErrorTestAfterLinear_Pmin').^2);
if (err_test<BestResultError)      %% ket qua test moi tot hon thi luu lai
    BestResultError = err_test
    best_net = net3;
end;
end;
end
ylearn_MLP=sim(best_net, BigBigA_Pmin_Learn');
ytest_MLP=sim(best_net, BigBigA_Pmin_Test');
err_learn = sqrt(mean((ylearn_MLP - ErrorLearnAfterLinear_Pmin').^2))   %%% Tai
day tinh sai so trung binh de don vi cua sai so cung la do C nhe
err_test = sqrt(mean((ytest_MLP - ErrorTestAfterLinear_Pmin').^2))
return

% Tinh Sai so hoc NL
MAE_Learn_NN_Pmin_MLP = mean(abs(err_learn))
MRE_Learn_NN_Pmin_MLP = mean(abs(err_learn./B_Pmin_Learn)*100)
MaxAE_Learn_NN_Pmin_MLP = max(abs(err_learn))

% Tinh Sai sokiem tra NL
MAE_Test_NN_Pmin_MLP=mean(abs(err_test))
MRE_Test_NN_Pmin_MLP=mean(abs(err_test./B_Pmin_Test)*100)
MaxAE_Test_NN_Pmin_MLP=max(abs(err_test))

```