

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

**ỨNG DỤNG MẠNG NEURON NHÂN TẠO
TRONG VIỆC DỰ BÁO DỮ LIỆU CHUỖI THỜI
GIAN CÓ TÍNH XU HƯỚNG VÀ TÍNH MÙA**

HỘI ĐỒNG: HỆ THỐNG THÔNG TIN 4

GVHD: PGS.TS Dương Tuấn Anh

GVPB: TS. Võ Thị Ngọc Châu

---o0o---

SVTH 1: Đoàn Ngọc Bảo 50800107

SVTH 2: Ngô Duy Khánh Vy 50802706

TP. HỒ CHÍ MINH, 01/2013

LỜI CAM ĐOAN

Chúng tôi cam đoan rằng ngoại trừ các kết quả tham khảo từ các công trình khác như đã ghi rõ trong luận văn, các công việc trình bày trong luận văn này là do chính chúng tôi thực hiện và chưa có phần nội dung nào của luận văn này được nộm để lấy một bằng cấp ở trường đại học nào khác.

Hồ Chí Minh, ngày tháng năm.....

Ký tên

LỜI CẢM ƠN

Trước hết, chúng tôi xin gửi lời cảm ơn chân thành và sâu sắc đến PGS.TS Dương Tuấn Anh, người đã trực tiếp hướng dẫn chúng tôi trong quá trình làm luận văn này. Sự hướng dẫn chu đáo, tận tình, cùng với những tài liệu và lời khuyên quý giá của thầy trong suốt khoảng thời gian qua là nhân tố không thể thiếu để chúng tôi có thể hoàn thành đề tài này.

Chúng tôi cũng xin gửi lời cảm ơn đến gia đình và bạn bè, những người luôn sát cánh, động viên và tạo mọi điều kiện tốt nhất để chúng tôi có thể học tập và hoàn tất được luận văn tốt nghiệp này.

Chúng tôi chân thành biết ơn sự tận tình dạy dỗ và sự giúp đỡ của tất cả quý thầy cô khoa Khoa học và Kỹ thuật Máy tính trường Đại học Bách khoa.

Luận văn này khó tránh khỏi thiếu sót do tầm nhìn của chúng tôi còn hạn hẹp, rất mong nhận được sự góp ý của quý thầy cô cùng các bạn.

TÓM TẮT LUẬN VĂN

Bài toán dự báo chuỗi thời gian ngày càng nhận được nhiều sự quan tâm của cộng đồng khoa học. Nhiều phương pháp, mô hình dự đoán đã được đề xuất và hiện thực. Trong đó, mạng neuron nhân tạo nhờ vào khả năng xấp xỉ hàm phi tuyến nên được áp dụng nhiều trong việc dự báo dữ liệu chuỗi thời gian. Tuy nhiên, nhiều nghiên cứu cho thấy đối với chuỗi dữ liệu có tính mùa và tính xu hướng thì chất lượng dự báo của mạng neuron nhân tạo không được cao.

Trong luận văn này, chúng tôi xây dựng và hiện thực hai mô hình cải tiến cho mạng neuron nhân tạo. Mô hình thứ nhất dựa trên ý tưởng khử thành phần mùa và xu hướng ra khỏi chuỗi dữ liệu và tiến hành huấn luyện mạng neuron trên chuỗi dữ liệu mới. Trong giai đoạn dự báo, kết quả do mạng neuron sinh ra sẽ được thêm vào thành phần mùa và xu hướng để tạo ra kết quả dự báo sau cùng. Mô hình thứ hai tận dụng khả năng xấp xỉ tốt thành phần mùa và xu hướng của kỹ thuật làm trơn theo hàm mũ và khả năng xấp xỉ hàm phi tuyến của mạng neuron nhân tạo để tạo ra mô hình lai có khả năng tạo ra kết quả dự đoán tốt hơn.

Trong quá trình thực nghiệm dự đoán với dữ liệu thực tế có tính mùa và tính xu hướng, hai mô hình này cho kết quả dự đoán tốt hơn nhiều so với mô hình mạng neuron nhân tạo thuần túy.

ABSTRACT

Nowadays, time series forecasting get more and more attention from the scientific community. There are several methods which are introduced to model and forecast for time series data. In that, artificial neuron network has been used as a widely promising method thanks to its nonlinear approximate capacity but many studies found that neuron network is not able to model seasonal and trend time series.

In this work, we use two ways to improve neuron network for forecasting seasonal and trend time series. The first way, we remove seasonality and trend from the time series then train the neuron network in the new time series. Outputs of neuron network will be added trend and seasonality to generate final forecast results. The second way, we combine neuron network and exponential smoothing to create a hybrid model. The model attempts to incorporate the good modeling capacity for linear trend and seasonal time series of exponential smoothing and the good nonlinear approximate capacity of neuron network to generate better forecast result than neuron network.

We also implement and use five real-world trend and seasonal time series to test the two models and find that their forecast values are better than neuron network.

MỤC LỤC

LỜI CAM ĐOAN	ii
LỜI CẢM ƠN.....	iii
TÓM TẮT LUẬN VĂN.....	iv
ABSTRACT	v
MỤC LỤC	vi
DANH MỤC HÌNH.....	x
DANH MỤC BẢNG	xii
Chương 1 GIỚI THIỆU.....	1
1.1. Đặt vấn đề	1
1.2. Mục tiêu của đề tài.....	2
1.3. Cấu trúc báo cáo	2
Chương 2 MẠNG NEURON NHÂN TẠO: CẤU TRÚC, NGUYÊN TẮC HOẠT ĐỘNG VÀ CÁC GIẢI THUẬT HUẤN LUYỆN	4
2.1. Sơ lược về mạng neuron nhân tạo	4
2.2. Cấu trúc về mạng neuron nhân tạo	5
2.3. Nguyên tắc hoạt động và các giải thuật huấn luyện của mạng neuron.....	9
2.3.1. Perceptron.....	9
2.3.2. Mạng nhiều lớp và giải thuật lan truyền ngược	15
2.3.3. Giải thuật RPROP	19
2.3.4. Hiện tượng quá khớp.....	24
Chương 3 ỨNG DỤNG MẠNG NEURON NHÂN TẠO VÀO CÔNG TÁC DỰ BÁO DỮ LIỆU CHUỖI THỜI GIAN	25

3.1.	Dữ liệu chuỗi thời gian	25
3.1.1.	Dữ liệu chuỗi thời gian là gì.....	25
3.1.2.	Các thành phần của dữ liệu chuỗi thời gian	26
3.1.1.	Các mô hình của chuỗi thời gian	28
3.2.	Áp dụng mạng neuron nhân tạo vào dự báo dữ liệu chuỗi thời gian.....	29
3.3.	Các bước xây dựng một mô hình mạng neuron nhân tạo để dự báo dữ liệu chuỗi thời gian	31
3.3.1.	Lựa chọn các biến.....	31
3.3.2.	Thu thập dữ liệu	32
3.3.3.	Tiền xử lý dữ liệu	32
3.3.4.	Phân chia tập dữ liệu	33
3.3.5.	Xây dựng cấu trúc mạng	34
3.3.6.	Xác định tiêu chuẩn đánh giá	36
3.3.7.	Huấn luyện mạng.....	36
3.3.8.	Dự đoán và cải tiến.....	37
3.4.	Hạn chế của việc sử dụng mạng Neuron trong việc dự báo đối với dữ liệu chuỗi thời gian có tính xu hướng và tính mùa	37
Chương 4	MÔ HÌNH KHỬ MÙA, KHỬ XU HƯỚNG KẾT HỢP MẠNG NEURON NHÂN TẠO.....	40
4.1.	Kỹ thuật khử tính xu hướng cho dữ liệu chuỗi thời gian.....	40
4.1.1.	Kỹ thuật khử xu hướng tuyến tính	40
4.1.2.	Kỹ thuật khử xu hướng bằng lấy hiệu	41
4.2.	Kỹ thuật khử tính mùa cho dữ liệu chuỗi thời gian	41
4.2.1.	Kỹ thuật lấy hiệu theo mùa	42

4.2.1.	Kỹ thuật khử mùa bằng RTMA.....	43
4.3.	Mô hình dự báo chuỗi thời gian bằng mạng neuron nhân tạo kết hợp với khử mùa và khử xu hướng	43
4.3.1.	Ý tưởng.....	43
4.3.2.	Hiện thực.	44
Chương 5	MÔ HÌNH LAI GIỮA MẠNG NEURON VÀ KỸ THUẬT LÀM TRƠN THEO HÀM MŨ.....	49
5.1.	Kỹ thuật làm trơn theo hàm mũ	49
5.1.1.	Kỹ thuật làm trơn theo hàm mũ đơn giản (Simple Exponential Smoothing).....	49
5.1.2.	Kỹ thuật làm trơn theo hàm mũ Holt	50
5.1.3.	Kỹ thuật làm trơn theo hàm mũ Winters (mô hình Holt-Winters).....	50
5.2.	Mô hình lai giữa mạng neuron nhân tạo và kỹ thuật làm trơn theo hàm mũ .	51
5.2.1.	Ý tưởng.....	51
5.2.2.	Hiện thực	52
Chương 6	THỰC NGHIỆM VÀ ĐÁNH GIÁ	57
6.1.	Cách thức thực nghiệm	57
6.2.	Kết quả thực nghiệm và đánh giá	58
6.2.1.	Bộ dữ liệu số lượng khách hàng đặt chỗ hàng tháng của hãng hàng không Pan Am từ năm 1946 đến năm 1960	58
6.2.2.	Bộ dữ liệu mật độ khí cacbonic trong khí quyển hàng tháng ở Mauna Loa (Hawaii).	62
6.2.3.	Bộ dữ liệu số người chết trung bình hàng tháng vì bệnh phổi ở Anh.	66
6.2.4.	Bộ dữ liệu doanh số bán hàng hàng tháng của một cửa hàng bán đồ lưu niệm ở Queensland, Australia, từ năm 1987 đến năm 1993.	70

6.2.5.	Bộ dữ liệu lượng tiêu thụ khí đốt trung bình theo quý tại Anh.	75
6.2.6.	Nhận xét chung.....	78
Chương 7	KẾT LUẬN	80
7.1.	Đánh giá kết quả	80
7.1.1.	Những công việc làm được	80
7.1.2.	Những đúc kết về mặt lý luận	81
7.1.3.	Mặt hạn chế	82
7.2.	Hướng phát triển	82
TÀI LIỆU THAM KHẢO		83
Phụ lục A Bảng thuật ngữ Anh-Việt		A1
Phụ lục B Hướng dẫn sử dụng chương trình thực nghiệm.....		B1

DANH MỤC HÌNH

Hình 2.1: Đơn vị mạng neuron	5
Hình 2.2: Mạng neuron truyền thẳng	7
Hình 2.3: Mạng neuron hồi quy	7
Hình 2.4: Mô hình học có giám sát	8
Hình 2.5: Đơn vị mạng Neuron	9
Hình 2.6: Mặt quyết định biểu diễn bởi perceptron hai đầu nhập	10
Hình 2.7: Hàm lỗi của một đơn vị tuyến tính	13
Hình 2.8: Đơn vị sigmoid	15
Hình 2.9: Giải thuật lan truyền ngược	20
Hình 2.10: Giải thuật RPROP	23
Hình 3.1: Số khách hàng đặt chỗ hàng tháng của hãng Pan Am	26
Hình 3.2: Độ tăng nhiệt độ trung bình hàng năm từ 1856 đến 2005	27
Hình 3.3: Hàm tự tương quan của chuỗi tăng nhiệt độ trung bình hàng năm (1856 - 2005)	28
Hình 3.4: Chuỗi thời gian có tính mùa	29
Hình 3.5: Mô hình học với chuỗi thời gian	30
Hình 3.6: Thủ tục sử dụng phương pháp walk-forward chia tập dữ liệu	34
Hình 4.1: (a) Chuỗi thời gian ban đầu. (b) Chuỗi thời gian sau khi khử xu hướng tuyến tính. (c) Chuỗi thời gian sau khi lấy hiệu.	42
Hình 4.2: (a) Chuỗi thời gian ban đầu. (b) Chuỗi thời gian sau khi khử mùa bằng RTMA. (c) Chuỗi thời gian sau khi lấy hiệu theo mùa.	44
Hình 4.3: Mô hình mạng neuron nhân tạo kết hợp với khử mùa và xu hướng.	45
Hình 4.4: Định dạng của file dữ liệu.	45

Hình 4.5: Đồ thị của chuỗi thời gian được biểu diễn bởi module đọc và biểu diễn dữ liệu.	46
Hình 4.6: Hàm tự tương quan của chuỗi thời gian được biểu diễn bởi module đọc và biểu diễn dữ liệu.	47
Hình 5.1: Cấu trúc mô hình lai	52
Hình 6.1: Lượng khách hàng đặt chỗ hàng tháng của hãng hàng không Pan Am từ năm 1946 đến năm 1960.....	58
Hình 6.2: Kết quả dự đoán của 3 mô hình cho chuỗi số lượng khách hàng đặt chỗ hàng tháng của hãng hàng không Pan Am.	60
Hình 6.3: Mật độ khí cacbonic trong khí quyển hàng tháng ở Mauna Loa (Hawaii) ...	63
Hình 6.4: Kết quả dự đoán của 3 mô hình cho chuỗi mật độ khí cacbonic trong khí quyển hàng tháng ở Mauna Loa (Hawaii).	65
Hình 6.5: Số người chết trung bình hàng tháng vì bệnh phổi ở Anh	67
Hình 6.6: Kết quả dự đoán của 3 mô hình cho chuỗi số người chết trung bình hàng tháng vì bệnh phổi ở Anh.	69
Hình 6.7: Doanh số bán hàng hàng tháng của một cửa hàng bán đồ lưu niệm ở Queensland, Australia, từ năm 1987 đến năm 1993.....	72
Hình 6.8: Kết quả dự đoán của 3 mô hình cho chuỗi doanh số bán hàng hàng tháng của một cửa hàng bán đồ lưu niệm ở Queensland, Australia.....	74
Hình 6.9: Lượng tiêu thụ khí đốt trung bình theo quý tại Anh	75
Hình 6.10: Kết quả dự đoán của 3 mô hình cho chuỗi lượng tiêu thụ khí đốt trung bình theo quý tại Anh.	78

DANH MỤC BẢNG

Bảng 6.1: Kết quả thực nghiệm cho chuỗi số lượng khách hàng đặt chỗ hàng tháng của hãng hàng không Pan Am từ năm 1946 đến năm 1960.....	59
Bảng 6.2: Kết quả thực nghiệm cho chuỗi mật độ khí cacbonic trong khí quyển hàng tháng ở Mauna Loa (Hawaii).....	64
Bảng 6.3: Kết quả thực nghiệm cho chuỗi số người chết trung bình hàng tháng vì bệnh phổi ở Anh	68
Bảng 6.4: Kết quả thực nghiệm cho chuỗi doanh số bán hàng hàng tháng của một cửa hàng bán đồ lưu niệm ở Queensland, Australia, từ năm 1987 đến năm 1993	73
Bảng 6.5: Kết quả thực nghiệm cho chuỗi lượng tiêu thụ khí đốt trung bình theo quý tại Anh	77

Chương 1

GIỚI THIỆU

1.1. Đặt vấn đề

Ngày nay khi mà hầu hết các tổ chức đều hoạt động trong môi trường không chắc chắn, kế hoạch lập ra hôm nay sẽ ảnh hưởng đến sự sống còn của tổ chức trong ngày mai thì việc dự đoán trước một cách chính xác trở nên rất quan trọng đối với các nhà ra quyết định. Các nhà đầu tư cần phải dự đoán được nhu cầu thị trường, sự biến động của nền kinh tế trong tương lai để có thể đầu tư hiệu quả. Các nhà hoạt động chính sách quốc gia cần dự đoán được về môi trường kinh doanh quốc tế, tỷ lệ lạm phát, tỷ lệ thất nghiệp... trong nhiều năm tới để đưa ra các chính sách phù hợp.

Để đưa ra dự báo chính xác và có cơ sở người ta tiến hành thu nhập dữ liệu về các yếu tố liên quan đến vấn đề mình quan tâm. Một kiểu dữ liệu thu nhập thường thấy là kiểu *dữ liệu chuỗi thời gian* (time series data). Dữ liệu chuỗi thời gian, tức là dữ liệu được thu nhập, lưu trữ và quan sát theo sự tăng dần của thời gian. Ví dụ, số lượng thí sinh dự thi đại học vào Trường Đại Học Bách Khoa thành phố Hồ Chí Minh được lưu trữ theo từng năm, hay số lượng hàng hóa đã bán được của một siêu thị được lưu trữ theo từng quý là các dữ liệu chuỗi thời gian.

Việc dự báo dữ liệu chuỗi thời gian ngày càng chiếm vị trí quan trọng trong hoạt động của các đơn vị tổ chức. Có rất nhiều phương pháp được xây dựng để dự báo chuỗi thời gian, nhiều phương pháp (ví dụ: phương pháp hồi quy) đã được xây dựng từ thế kỷ 19 và nhiều phương pháp (ví dụ phương pháp mạng neuron nhân tạo) được phát triển gần đây. Cơ bản có hai kỹ thuật chủ yếu trong việc dự báo chuỗi thời gian là các phương pháp thống kê: *hồi quy* (regression), *làm trơn* (smoothing), phương pháp luận Box-Jenkins... và phương pháp dùng mạng neuron nhân tạo.

Mạng neuron nhân tạo là một mô hình toán học đã được nghiên cứu từ lâu và được ứng dụng nhiều vào các bài toán mô phỏng, nhận dạng, dự đoán. Gần đây mạng neuron nhân tạo được quan tâm và ứng dụng ngày càng nhiều vào các bài toán dự báo dữ liệu chuỗi thời gian nhờ vào khả năng xấp xỉ các hàm phi tuyến của mình. Tuy nhiên những nghiên cứu của Zhang [4] [1] [2] và Virili [7] cho thấy mạng neuron nhân tạo không xấp xỉ tốt được các chuỗi thời gian có tính xu hướng và tính mùa. Vấn đề đặt ra là phải cải tiến mạng neuron để có thể áp dụng được cho các chuỗi thời gian có tính xu hướng và tính mùa.

1.2. Mục tiêu của đề tài

Mục tiêu của luận văn này là áp dụng mạng neuron nhân tạo vào việc dự báo dữ liệu chuỗi thời gian có tính mùa và tính xu hướng. Vì mạng neuron nhân tạo thuần túy không có khả năng dự báo tốt cho chuỗi thời gian có tính mùa và xu hướng nên trong luận văn này chúng tôi thực hiện việc cải tiến mạng neuron bằng hai phương pháp là kết hợp việc khử mùa, khử xu hướng với mạng neuron nhân tạo và kết hợp mô hình làm trơn theo hàm mũ với mạng neuron nhân tạo để tăng khả năng dự báo của nó cho chuỗi thời gian, đặc biệt là những chuỗi có tính mùa và tính xu hướng.

1.3. Cấu trúc báo cáo

Bài báo cáo chia làm 7 chương:

Chương 1: Giới thiệu về bài toán và nhiệm vụ đề tài.

Chương 2: Giới thiệu về cấu trúc, nguyên tắc hoạt động của mạng neuron nhân tạo cùng với hai giải thuật huấn luyện: lan truyền ngược và RPROP.

Chương 3: Giới thiệu về cách ứng dụng mạng neuron nhân tạo vào công tác dự báo dữ liệu chuỗi thời gian.

Chương 4: Trình bày cách hiện thực mô hình mạng neuron nhân tạo kết hợp với khử mùa và xu hướng.

Chương 5: Trình bày ý tưởng và cách hiện thực mô hình lai giữa mạng neuron nhân tạo và kỹ thuật làm trơn theo hàm mũ.

Chương 6: Trình bày thực nghiệm và đánh giá độ chính xác của hai mô hình mạng neuron nhân tạo kết hợp với khử mùa, khử xu hướng và mô hình lai giữa mạng neuron nhân tạo với kỹ thuật làm trơn theo hàm mũ trong việc dự báo cho chuỗi thời gian có tính mùa và tính xu hướng.

Chương 7: Đánh giá kết quả đạt được, các mặt hạn chế và hướng phát triển của đề tài.

Chương 2

MẠNG NEURON NHÂN TẠO: CẤU TRÚC, NGUYÊN TẮC HOẠT ĐỘNG VÀ CÁC GIẢI THUẬT HUẤN LUYỆN

Chương này chúng tôi sẽ giới thiệu về cấu trúc, nguyên tắc hoạt động và hai giải thuật huấn luyện mạng neuron nhân tạo: lan truyền ngược và RPROP.

2.1. Sơ lược về mạng neuron nhân tạo

Mạng neuron nhân tạo (Artificial Neural Network) là một mô hình toán học định nghĩa một hàm số từ một tập đầu vào đến một tập đầu ra. Mạng neuron nhân tạo được mô phỏng theo mạng neuron sinh học trong bộ não người.

Theo các nhà sinh lý học thì bộ não con người chứa khoảng 10^{11} các phần tử liên kết chặt chẽ với nhau gọi là các neuron. Mỗi neuron được cấu tạo bởi các thành phần: tế bào hình cây, tế bào thân và sợi trục thần kinh. Tế bào hình cây có nhiệm vụ mang tín hiệu điện sinh học tới tế bào thân, tế bào thân sẽ thực hiện tính tổng và *phân ngưỡng* (thresholds) các tín hiệu đến. Sợi trục thần kinh có nhiệm vụ đưa tín hiệu từ tế bào thân ra ngoài. Điểm tiếp xúc giữa sợi trục thần kinh của neuron này với tế bào hình cây của neuron kia gọi là *khớp thần kinh* (synapse). Sự sắp xếp của các neuron và độ mạnh yếu của các khớp thần kinh quyết định khả năng của mạng neuron. Cấu trúc của mạng neuron sinh học luôn luôn thay đổi và phát triển theo quá trình học tập và lao động của con người, các liên kết mới được tạo ra và các liên kết cũ bị loại bỏ, hay tăng giảm độ mạnh yếu của các liên kết thông qua các khớp thần kinh.

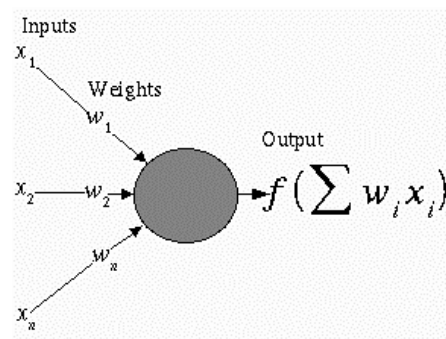
Mạng neuron nhân tạo là một sự mô phỏng đơn giản mạng neuron sinh học. Cấu trúc của mạng bao gồm các đơn vị tính toán đơn giản (tượng trưng cho các neuron) được liên kết với nhau bằng các cạnh có trọng số (tượng trưng cho các khớp thần kinh). Khả năng xấp xỉ hàm số của mạng neuron nhân tạo phụ thuộc vào hình dạng và độ mạnh yếu của các liên kết (giá trị của các trọng số).

Trong quá trình phát triển của mình, mạng neuron nhân tạo đã được ứng dụng thành công trong nhiều bài toán thực tế như nhận dạng chữ viết, nhận dạng tiếng nói, điều khiển tự động, dự báo chuỗi thời gian...

2.2. Cấu trúc về mạng neuron nhân tạo

Mạng neuron nhân tạo là một mạng gồm một tập các *đơn vị* (unit) được kết nối với nhau bằng các cạnh có trọng số.

Một đơn vị thực hiện một công việc rất đơn giản: nó nhận tín hiệu vào từ các đơn vị phía trước hay một nguồn bên ngoài và sử dụng chúng để tính tín hiệu ra. Mỗi đơn vị có thể có nhiều tín hiệu đầu vào nhưng chỉ có một tín hiệu đầu ra duy nhất. Đôi khi các đơn vị còn có một giá trị gọi là *độ lệch* (bias) được gộp vào các tính hiệu đầu vào để tính tín hiệu ra. Để đơn giản ký hiệu, độ lệch của một đơn vị được xem như là trọng số nối từ một đơn vị giả có giá trị xuất luôn là 1 đến đơn vị đó. Hình 2.1 minh họa đầu vào và đầu ra ứng với một đơn vị của mạng neuron nhân tạo.



Hình 2.1: Đơn vị mạng neuron

Trong một mạng neuron có ba kiểu đơn vị:

- Các đơn vị đầu vào, nhận tín hiệu từ bên ngoài.
- Các đơn vị đầu ra, gửi dữ liệu ra bên ngoài.
- Các đơn vị ẩn, tín hiệu vào của nó được truyền từ các đơn vị trước nó và tín hiệu ra được truyền đến các đơn vị sau nó trong mạng.

Khi nhận được các tín hiệu đầu vào, một đơn vị sẽ nhân mỗi tín hiệu với trọng số tương ứng rồi lấy tổng các giá trị vừa nhận được. Kết quả sẽ được đưa vào một hàm số gọi là *hàm kích hoạt* (Activation function) để tính ra tín hiệu đầu ra. Các đơn vị khác nhau có thể có các hàm kích hoạt khác nhau.

Có 4 loại hàm kích hoạt thường dùng:

- Hàm đồng nhất:

$$g(x) = x$$

- Hàm ngưỡng:

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

- Hàm sigmoid:

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

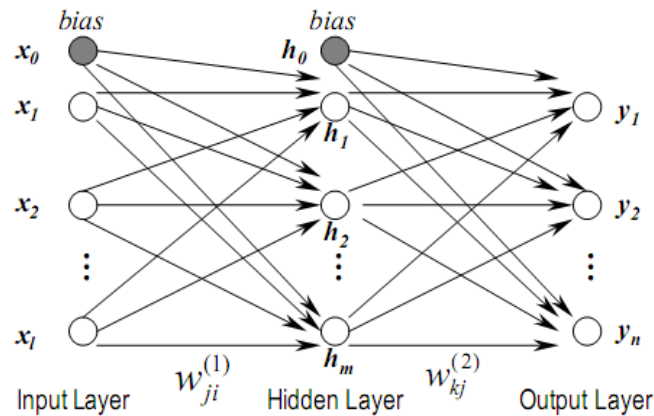
- Hàm sigmoid lưỡng cực

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

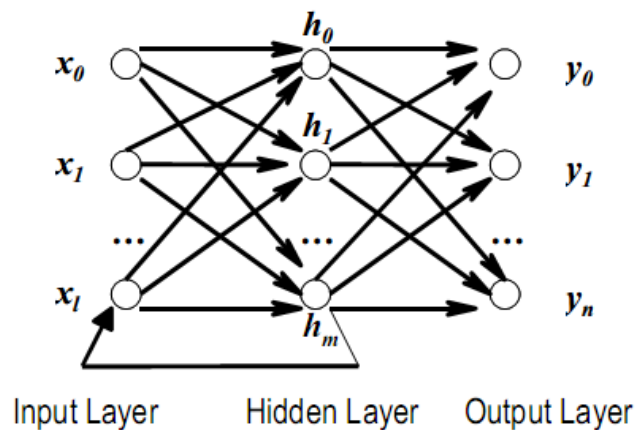
Các đơn vị liên kết với nhau qua các cạnh có trong sơ đồ tạo thành mạng neuron nhân tạo. Tùy theo số lượng các đơn vị và cách thức liên kết của chúng mà tạo thành các mạng neuron khác nhau có khả năng khác nhau. Có hai loại hình dạng mạng neuron nhân tạo cơ bản là mạng truyền thẳng và mạng hồi quy:

- *Mạng truyền thẳng* (Feed-forward neural network): Một đơn vị ở lớp đứng trước sẽ kết nối với tất cả các đơn vị ở lớp đứng sau. Tín hiệu chỉ được truyền theo một hướng từ lớp đầu vào qua các lớp ẩn (nếu có) và đến lớp đầu ra. Nghĩa là tín hiệu ra của một đơn vị không được phép truyền cho các đơn vị trong cùng lớp hay ở lớp trước. Đây là loại mạng rất phổ biến và được dùng nhiều trong việc dự báo dữ liệu chuỗi thời gian. Trong luận văn này chúng tôi chỉ sử dụng mô hình mạng truyền thẳng. Hình 2.2 minh họa cấu trúc của một mạng neuron truyền thẳng.

- *Mạng hồi quy* (Recurrent neural network): Khác với mạng truyền thẳng, mạng hồi quy có chứa các liên kết ngược từ một đơn vị đến các đơn vị ở lớp trước nó. Giá trị đầu ra của một đơn vị sẽ được truyền ngược lại cho các đơn vị ở lớp trước để điều chỉnh cấu hình của mạng. Hình 2.3 minh họa cấu trúc của một mạng neuron hồi quy.



Hình 2.2: Mạng neuron truyền thẳng

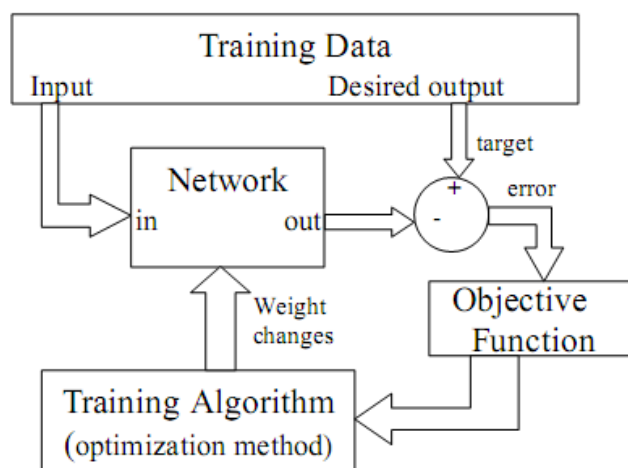


Hình 2.3: Mạng neuron hồi quy

Chức năng của một mạng neuron được quyết định bởi các nhân tố như: hình dạng mạng (số lớp, số đơn vị trên mỗi lớp, cách mà các lớp được liên kết với nhau) và

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa các trọng số của các liên kết bên trong mạng. Đối với mạng truyền thẳng hình dạng 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 đầu ra mong muốn được gọi là *học* (learning) hay *huấn luyện* (training). Rất nhiều thuật toán huấn luyện 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 giám sát* (Supervised learning) và *Học không có giám sát* (Unsupervised Learning) [6].

- **Học có giám sát:** 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 này có sẵn trong quá trình thu nhập dữ liệu. Sự khác biệt giữa các đầu ra theo tính toán trên mạng 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 $f(x)$ thỏa mãn tất cả các mẫu học đầu vào [6]. Đây là mô hình học rất phổ biến trong việc áp dụng mạng neuron vào bài toán dự báo dữ liệu chuỗi thời gian. Hình 2.4 minh họa cấu trúc hoạt động của mô hình học này. Hai giả thuật được chúng tôi hiện thực trong luận văn này, giả thuật *lan truyền ngược* (Backpropagation) và *RPROP* (resilient propagation) là hai giả thuật học thuộc mô hình này.



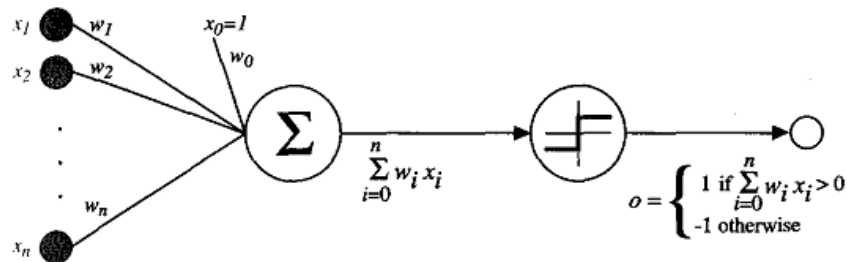
Hình 2.4: Mô hình học có giám sát

- **Học không có giám sát:** Với cách học không có giám sát, 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 giám sát, các đích trùng với đầu vào. Nói một cách khác, học không có giám sát luôn thực hiện một công việc tương tự như một mạng tự liên hợp, cô đọng thông tin từ dữ liệu vào [6]

2.3. Nguyên tắc hoạt động và các giải thuật huấn luyện của mạng neuron

2.3.1. Perceptron

Để hiểu rõ về nguyên tắc hoạt động và cách huấn luyện các mạng neuron nhân tạo trước hết ta khảo sát một mô hình mạng neuron đơn giản được xây dựng trên một đơn vị gọi là perceptron. Một perceptron nhận một vector các giá trị thực, tính tổ hợp tuyến tính của chúng và xuất ra 1 nếu kết quả lớn hơn một ngưỡng nào đó và xuất ra -1 trong các trường hợp còn lại. Hình 2.5 minh họa cách dữ liệu đi vào và đi ra tại một đơn vị của mạng neuron nhân tạo.



Hình 2.5: Đơn vị mạng Neuron

Một cách hình thức, khi nhận một vector đầu vào n chiều gồm các giá trị x_1 đến x_n , giá trị xuất sẽ được tính như sau:

$$o(x_1, \dots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n > 0 \\ -1 & \text{otherwise} \end{cases}$$

Ở đây các số thực w_i là các trọng số biểu diễn mức độ đóng góp của giá trị nhập x_i vào giá trị xuất của perceptron. Đại lượng $(-w_0)$ là ngưỡng mà tổ hợp tuyến các giá

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa
trị nhập phải vượt qua để kết quả xuất là 1. Đặt $x_0 = 1$, ta viết lại phương trình trên
dưới dạng vector như sau

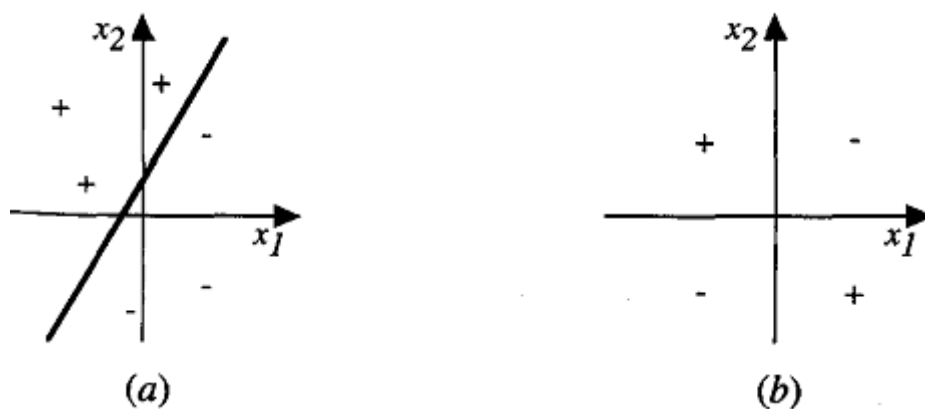
$$o(\vec{x}) = \text{sgn}(\vec{w} \cdot \vec{x})$$

Ở đây \vec{x} và \vec{w} là các vector có $n + 1$ chiều. Hàm $\text{sgn}(y)$ được định nghĩa như
sau:

$$\text{sgn}(y) = \begin{cases} 1 & \text{if } y > 0 \\ -1 & \text{otherwise} \end{cases}$$

Nếu xem các vector nhập (x_0, x_1, \dots, x_n) là các điểm trên không gian $n + 1$ chiều
(x_0 luôn là 1) thì perceptron biểu diễn một *mặt quyết định* (decision surface) xem một
điểm có nằm trên một *siêu phẳng* (hyperplane) có phương trình là $\vec{w} \cdot \vec{x} = 0$ hay không.
Perceptron sẽ xuất ra giá trị 1 cho các điểm nằm trên siêu phẳng này và xuất ra -1 cho
các điểm còn lại.

Trong thực tế, ta thường có sẵn một bộ dữ liệu mẫu gồm một tập các điểm được
gán nhãn dương và âm. Bài toán huấn luyện perceptron là bài toán xác định vector \vec{w}
sau cho siêu phẳng $\vec{w} \cdot \vec{x} = 0$ phân chia các điểm trong tập mẫu một cách chính xác theo
các nhãn của nó. Thực tế có một số bộ dữ liệu mà không thể tìm thấy bất kỳ siêu
phẳng nào có thể phân chia đúng các điểm của nó, các bộ dữ liệu đó được gọi là tập dữ
liệu không *khả phân tuyến tính* (linearly separable). Ngược lại nếu một bộ dữ liệu có
thể được phân chia đúng bởi một siêu phẳng nào đó thì gọi là khả phân tuyến tính.



Hình 2.6: Mặt quyết định biểu diễn bởi perceptron hai đầu nhập

Hình 2.6 (a) là một tập mẫu khả phân tuyến tính có thể được phân ra bởi một mặt quyết định của perceptron. Hình 2.6 (b) là một tập mẫu không khả phân tuyến tính.

Quá trình huấn luyện một perceptron là một quá trình tìm kiếm một vector \vec{w} trên một không gian thực $n + 1$ chiều sao cho nó có khả năng phân xuất ra các giá trị $+1, -1$ một cách đúng đắn cho một tập dữ liệu nào đó. Có hai giải thuật huấn luyện cơ bản là *luật huấn luyện perceptron* (perceptron training rule) và *luật delta* (delta rule).

- a) **Luật huấn luyện perceptron:** Để tìm một vector \vec{w} thích hợp, trước hết ta áp dụng một perceptron với trọng số \vec{w} ngẫu nhiên qua từng mẫu của tập dữ liệu huấn luyện và hiệu chỉnh các trọng số này khi có sự phân loại sai tập mẫu. Quá trình này được lặp đi lặp lại cho đến khi perceptron đã phân loại đúng tất cả các mẫu của tập huấn luyện. Các trọng số được cập nhập theo luật

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i = \eta(t - o)x_i$$

Ở đây o là giá trị xuất của perceptron, t là giá trị đích của mẫu huấn luyện hiện thời, x_i là giá trị nhập thứ i , η là *hệ số học* (learning rate) có vai trò điều tiết mức độ thay đổi của trọng số trong các bước cập nhập. Nó thông thường được gán một giá trị dương nhỏ (ví dụ 0.1) và được điều chỉnh giảm khi số lần cập nhập trọng số tăng lên [9]. Giải thuật học này được chứng minh hội tụ sau một số hữu hạn lần cập nhập các trọng số đối với các tập dữ liệu mẫu khả phân tuyến tính và một hệ số học đủ nhỏ nhưng đối với các tập dữ liệu không khả phân tuyến tính thì sự hội tụ là không chắc [9].

- b) **Luật delta:** Luật perceptron không đảm bảo tính hội tụ đối với các tập mẫu khả phân tuyến tính do đó người ta thiết kế giải thuật luật delta để vượt qua khó khăn này. Luật delta sẽ hội tụ về một xấp xỉ tốt nhất cho các tập không khả phân tuyến tính. Ý tưởng chính của luật delta là áp dụng phương pháp *giảm độ dốc* (gradient descent) để tìm kiếm vector trọng số đáp ứng tốt nhất tập huấn luyện. Xét một perceptron thực hiện việc lấy tổ hợp tuyến tính các giá trị nhập nhưng không phân ngưỡng kết quả.

Perceptron này gọi là *perceptron không phân ngưỡng* (unthresholded perceptron) hay còn gọi là *đơn vị tuyến tính* (linear unit). Giá trị xuất của perceptron được tính như sau

$$o(\vec{x}) = \vec{w} \cdot \vec{x}$$

Để áp dụng luật delta ta cần định nghĩa một hàm đánh giá, hay còn gọi là *hàm lỗi* (training error function). Có nhiều hàm lỗi được sử dụng nhưng thường dùng nhất là hàm sau

$$E(\vec{w}) \equiv \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

Ở đây D là tập dữ liệu huấn luyện, d là một mẫu trong tập D , t_d là giá trị đích của mẫu d , o_d là giá trị xuất của perceptron. Mục đích của luật delta là tìm vector \vec{w} sau cho $E(\vec{w})$ đạt giá trị nhỏ nhất. Hình 2.7 là một biểu diễn hàm lỗi của một đơn vị tuyến tính. Trục thẳng đứng của đồ thị là giá trị hàm lỗi, hai trục ở mặt phẳng ngang là giá trị của các trọng số.

Phương pháp giảm độ dốc bắt đầu tìm với một vector trọng số ngẫu nhiên và duyệt qua các mẫu của tập huấn luyện, mỗi lần duyệt qua các trọng số sẽ được cập nhập theo hướng làm giảm giá trị hàm lỗi. Quá trình này được lặp đi lặp lại cho đến khi đạt được giá trị cực tiểu của hàm lỗi.

Hướng cập nhập các trọng số để làm giảm giá trị hàm lỗi được xác định theo *vector độ dốc* (gradient) của hàm lỗi E theo \vec{w} , ký hiệu là $\nabla E(\vec{w})$

$$\nabla E(\vec{w}) \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right]$$

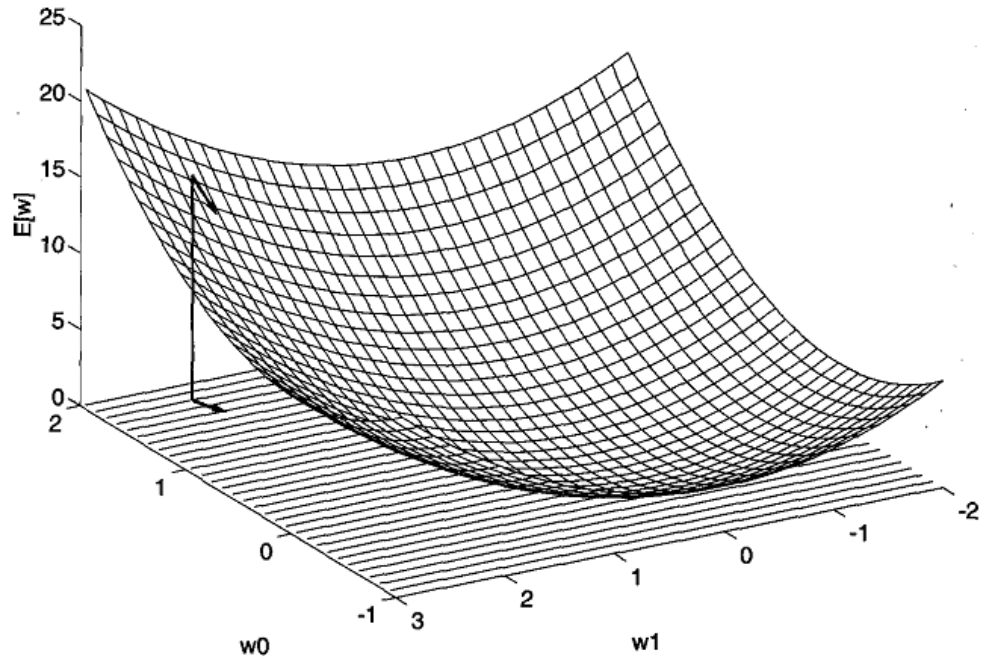
Về mặt toán học vector độ dốc biểu diễn hướng làm tăng giá trị hàm E trong không gian trọng số, do đó $-\nabla E(\vec{w})$ sẽ là hướng làm giảm giá trị hàm E . Trong hình 2.7 nó được biểu diễn bằng dấu mũi tên. Các trọng số sẽ được cập nhập theo quy luật sau:

$$\begin{aligned} \vec{w} &\leftarrow \vec{w} + \Delta \vec{w} \\ \Delta \vec{w} &= -\eta \nabla E(\vec{w}) \end{aligned}$$

Luật huấn luyện này có thể được viết lại cho từng trọng số như sau:

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} \quad (2.1)$$



Hình 2.7: Hàm lỗi của một đơn vị tuyến tính

Để thực hiện cập nhật các trọng số, ta thực hiện tính đạo hàm riêng phần của hàm E theo từng trọng số:

$$\begin{aligned} \frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i} \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 \\ &= \frac{1}{2} \sum_{d \in D} \frac{\partial}{\partial w_i} (t_d - o_d)^2 \\ &= \frac{1}{2} \sum_{d \in D} 2(t_d - o_d) \frac{\partial}{\partial w_i} (t_d - o_d) \\ &= \sum_{d \in D} (t_d - o_d) \frac{\partial}{\partial w_i} (t_d - \vec{w} \cdot \vec{x}_d) \\ \frac{\partial E}{\partial w_i} &= \sum_{d \in D} (t_d - o_d)(-x_{id}) \quad (2.2) \end{aligned}$$

Thay (2.2) vào (2.1) ta được giá nhập cập nhật trọng số qua từng bước ta được

$$\Delta w_i = \eta \sum_{d \in D} (t_d - o_d) x_{id} \quad (2.3)$$

Ở đây giá trị x_{id} là giá trị đầu vào thứ i của mẫu d . Phương pháp giảm độ dốc có hai hạn chế chính là tốc độ hội tụ đôi khi khá chậm và nếu có nhiều *cực tiểu cục bộ* (local minimum) trên bề mặt của hàm lỗi thì giải thuật dễ rơi vào cực tiểu cục bộ mà không đạt được *cực tiểu toàn cục* (global minimum). Để giải quyết các khó khăn này người ta đã phát triển phương pháp giảm độ dốc thành phương pháp *giảm độ dốc tăng cường* (incremental gradient descent). Khác với phương pháp giảm độ dốc ở trên phương pháp giảm độ dốc tăng cường thực hiện việc tính toán lỗi và cập nhập các trọng số ngay khi duyệt qua một mẫu của tập dữ liệu. Giá trị cập nhập cho các trọng số của phương pháp giảm độ dốc tăng cường là

$$\Delta w_i = \eta(t - o) x_i \quad (2.4)$$

Ở đây các giá trị t , o , x_i lần lượt là giá trị đích, giá trị xuất của mạng và giá trị nhập thứ i của mẫu huấn luyện hiện hành. Hàm lỗi của phương pháp giảm độ dốc tăng cường không phải là hàm lỗi toàn cục cho toàn bộ dữ liệu huấn luyện như phương pháp giảm độ dốc thông thường mà là hàm lỗi cho từng mẫu trong tập dữ liệu

$$E_d(\vec{w}) = \frac{1}{2}(t_d - o_d)^2$$

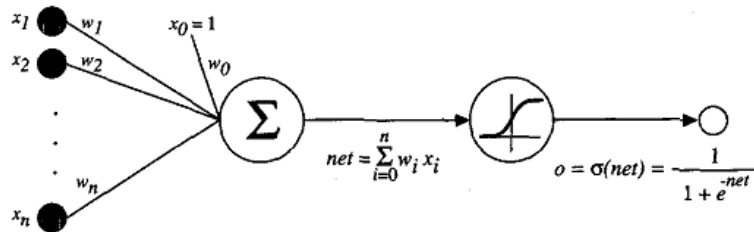
Ở đây giá trị t_d , o_d lần lượt là giá trị đích và giá trị xuất của mạng cho mẫu d trong tập dữ liệu. Theo Tom Mitchell [9] phương pháp giảm độ dốc tăng cường khác với phương pháp giảm độ dốc thông thường ở ba điểm sau. Thứ nhất, giải thuật thực hiện việc tính toán lỗi và cập nhập các trọng số cho mỗi mẫu trong tập huấn luyện chứ không đợi duyệt qua hết các mẫu trong tập huấn luyện. Thứ hai, phương pháp giảm độ dốc thông thường cần nhiều tính toán để cập nhập các trọng số vì nó cần phải tính toán hàm lỗi thực sự cho toàn bộ tập dữ liệu huấn luyện và mỗi lần cập nhập các trọng số được cập nhập một bước lớn hơn phương pháp giảm độ dốc tăng cường. Thứ ba phương pháp giảm độ dốc tăng cường có khả năng không bị rơi vào cực tiểu cục bộ vì nó sử dụng $\nabla E_d(\vec{w})$ thay cho $\nabla E(\vec{w})$ để tìm kiếm trạng thái tối ưu.

Sự khác biệt giữa hai giải thuật huấn luyện luật delta và luật huấn luyện perceptron khác nhau ở tính chất hội tụ của chúng. Luật huấn luyện perceptron hội tụ sau một số lần lặp hữu hạn và tìm ra một mặt phẳng phân loại hoàn hảo một tập dữ liệu huấn luyện khả phân tuyến tính trong khi giải thuật luật delta sẽ hội tụ về một điểm cực tiểu của hàm lỗi với một thời gian khá lâu (có thể là vô hạn) nhưng sự hội tụ của nó không bị ảnh hưởng bởi tính khả phân tuyến tính của tập dữ liệu huấn luyện [9].

2.3.2. Mạng nhiều lớp và giải thuật lan truyền ngược

Mạng neuron đơn giản như perceptron chỉ biểu diễn được các hàm tuyến tính, nhưng trong thực tế ta cần biểu diễn các hàm phi tuyến như trong các bài toán dự báo chuỗi thời gian. Để làm được điều này ta sử dụng các mạng neuron nhiều lớp, tức là mạng gồm một lớp đầu vào, một lớp đầu ra và một hay nhiều lớp ẩn.

Các mạng neuron nhiều lớp ít khi sử dụng các đơn vị tuyến tính hay đơn vị phân ngưỡng mà chúng sử dụng các đơn vị có các hàm kích hoạt là các hàm khả vi. Một trong những đơn vị hay dùng nhất là *đơn vị sigmoid* (sigmoid unit) (hình 2.8). Một đơn vị sigmoid sẽ tính tổ hợp tuyến tính các giá trị đầu vào và đưa kết quả này vào hàm sigmoid để tính giá trị đầu ra.



Hình 2.8: Đơn vị sigmoid

Công thức tính giá trị đầu ra của đơn vị sigmoid là

$$o = \sigma(\vec{w} \cdot \vec{x})$$

Với

$$\sigma(y) = \frac{1}{1 + e^{-y}}$$

Một thuận lợi khi sử dụng các đơn vị sigmoid là nhờ đạo hàm của hàm sigmoid rất dễ tính ($\sigma'(y) = \sigma(y) * (1 - \sigma(y))$). Điều này làm cho việc áp dụng phương pháp giảm độ dốc được dễ dàng.

Giải thuật lan truyền ngược tìm tập các trọng số thích hợp cho một mạng neuron truyền thẳng nhiều lớp. Nó áp dụng phương pháp giảm độ dốc để tối thiểu hóa bình phương sai số giữa kết quả xuất của mạng với kết quả xuất mong muốn. Ý tưởng chính của giải thuật là giá trị lỗi sẽ được lan truyền ngược từ tầng xuất về tầng nhập để tính $\nabla E(\vec{w})$

Hàm lỗi của giải thuật lan truyền ngược được định nghĩa tổng quát như sau

$$E(\vec{w}) \equiv \frac{1}{2} \sum_{d \in D} \sum_{k \in \text{outputs}} (t_{kd} - o_{kd})^2$$

Ở đây *outputs* là tập các đầu ra của mạng neuron, t_{kd} và o_{kd} lần lượt là giá trị đích và giá trị xuất của đầu ra thứ k của mẫu huấn luyện d

Giải thuật lan truyền ngược áp dụng phương pháp giảm độ dốc để tìm ra điểm tối ưu của hàm lỗi. Với mỗi mẫu trong tập huấn luyện, mạng neuron được áp dụng để tính đầu ra sau đó giá trị độ dốc của hàm lỗi được tính cho từng đơn của mạng. Cuối cùng giải thuật áp dụng phương pháp giảm độ dốc để cập nhập các giá trị trọng số

Để áp dụng phương pháp giảm độ dốc trước hết ta cần thông tin về đạo hàm riêng phần của hàm lỗi cho từng trọng số

Ta tính đạo hàm riêng phần này như sau:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_i} \frac{\partial o_i}{\partial w_{ij}} \quad (2.5)$$

$$\text{với } \frac{\partial o_i}{\partial w_{ij}} = \frac{\partial o_i}{\partial \text{net}_i} \frac{\partial \text{net}_i}{\partial w_{ij}} = f'(\text{net}_i) o_j \quad (2.6)$$

$$\text{net}_i = \sum_{j \in \text{pred}(i)} o_j w_{ij} - \theta_i \quad (2.7)$$

Ở đây:

- w_{ij} là trọng số của cạnh nối đơn vị j đến đơn vị i
- o_j là kết quả xuất của đơn vị j

- $f()$ là hàm kích hoạt của các đơn vị
- $pred(i)$ là các đơn vị đứng trước đơn vị i trong mạng

Giá trị $\frac{\partial E}{\partial o_i}$ được tính theo hai trường hợp tùy theo đơn vị i là đơn vị ở tầng xuất

hay tầng ẩn:

- Nếu đơn vị i là đơn vị ở tầng xuất thì:

$$\frac{\partial E}{\partial o_i} = \frac{\partial}{\partial o_i} \frac{1}{2} \sum_{k \in outputs} (t_k - o_k)^2$$

Đạo hàm $\frac{\partial}{\partial o_j} (t_k - o_k)^2$ bằng 0 đối với mỗi giá trị k khác i nên

$$\begin{aligned} \frac{\partial E}{\partial o_i} &= \frac{\partial}{\partial o_i} \frac{1}{2} (t_i - o_i)^2 \\ &= \frac{1}{2} 2(t_i - o_i) \frac{\partial (t_i - o_i)}{\partial o_i} \\ &= -(t_i - o_i) \end{aligned} \quad (2.8)$$

Thay (2.8) và (2.6) vào (2.5) ta được công thức tính đạo hàm riêng phần của hàm lỗi theo trọng số w_{ij} của đơn vị xuất i

$$\frac{\partial E}{\partial w_{ij}} = -(t_i - o_i) * f'(net_i) o_j \quad (2.9)$$

- Nếu đơn vị i là đơn vị ở tầng ẩn ở tầng ẩn thì việc tính toán phức tạp hơn bởi vì giá trị xuất của i không ảnh hưởng trực tiếp lên giá trị xuất của mạng neuron mà ảnh hưởng gián tiếp thông qua các đơn vị ở sau nó.

$$\begin{aligned} \frac{\partial E}{\partial o_i} &= \sum_{k \in succ(i)} \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial o_i} \\ &= \sum_{k \in succ(i)} \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial o_i} = \sum_{k \in succ(i)} \frac{\partial E}{\partial o_k} f'(net_k) w_{ki} \end{aligned} \quad (2.10)$$

Thay (2.10) và (2.6) vào (2.5) ta được công thức tính đạo hàm riêng phần của hàm lỗi theo trọng số w_{ij} của đơn vị ẩn i

$$\frac{\partial E}{\partial w_{ij}} = \left(\sum_{k \in succ(i)} \frac{\partial E}{\partial o_k} f'(net_k) w_{ki} \right) * f'(net_i) o_j \quad (2.11)$$

Ở đây $succ(i)$ là các đơn vị ở lớp ngay sau đơn vị i . Các công thức này cho phép ta xây dựng một thủ tục tính đạo hàm riêng của hàm lỗi E theo các trọng số w_{ij} như sau: Bắt đầu tính toán từ các đơn vị ở tầng xuất, sau đó sử dụng kết quả vừa tính được vào việc tính toán ở các đơn vị ở tầng trước. Nói cách khác thông tin về độ dốc được lan truyền từ tầng xuất đến tầng nhập. Do đó giả thuật này được gọi là giải thuật lan truyền ngược.

Mỗi khi thông tin về đạo hàm riêng phần đã biết, bước tiếp theo trong giải thuật lan truyền ngược là cập nhập các trọng số w_{ij} .

$$\Delta w(t) = -\eta * \nabla E(t) = -\eta * \frac{\partial E(t)}{\partial w(t)} \quad (2.12)$$

$$w(t+1) = w(t) + \Delta w(t) \quad (2.13)$$

Ở đây η là hệ số học có vai trò điều tiết mức độ thay đổi của trọng số trong các bước cập nhập.

Cơ bản có hai phương pháp cập nhập các trọng số phân loại theo thời điểm cập nhập: *học theo mẫu* (learning by pattern) và *học theo epoch* (learning by epoch). Một epoch là một lần học duyệt qua tất cả các mẫu trong tập dữ liệu mẫu dùng để học.

Trong phương pháp học theo mẫu đôi khi còn được gọi là *học trực tuyến* (online learning) áp dụng phương pháp giảm độ dốc tăng cường, cứ mỗi lần một mẫu trong tập dữ liệu được duyệt qua thì các trọng số sẽ được cập nhập. Phương pháp này cố gắng tối thiểu hàm *lỗi tổng thể* (overall error) bằng cách tối ưu hàm lỗi cho từng mẫu trong tập dữ liệu học. Phương pháp này làm việc tốt cho các tập dữ liệu mẫu có kích cỡ lớn và chứa đựng nhiều thông tin dư thừa [11].

Phương pháp học theo epoch thực hiện lấy tổng tất cả thông tin về *độ dốc* (gradient) cho toàn bộ *tập mẫu* (pattern set) sau đó mới cập nhập các trọng số theo phương pháp giảm độ dốc thông thường, nghĩa là nó thực hiện việc cập nhập trọng số sau khi đã duyệt qua hết các mẫu trong tập dữ liệu. Phương pháp này còn có tên gọi khác là *học theo bó* (batch learning).

Mặc dù giải thuật lan truyền ngược tương đối đơn giản nhưng trong thực tế việc lựa chọn một hệ số học phù hợp là không hề đơn giản. Hệ số học quá nhỏ sẽ dẫn đến thời gian hội tụ của giải thuật quá lâu, ngược lại hệ số học quá lớn sẽ dẫn đến hiện tượng *giao động* (oscillation), ngăn không cho giá trị hàm mục tiêu hội tụ về một điểm nhất định. Hơn nữa, mặc dù điểm tối ưu cục bộ có thể được chứng minh là luôn có thể đạt được ở một vài trường hợp cụ thể nhưng không có gì đảm bảo giải thuật sẽ tìm được điểm tối ưu toàn cục của hàm lỗi [11]. Một vấn đề khác nữa là kích cỡ của đạo hàm cũng ảnh hưởng đến sự cập nhật các trọng số. Nếu đạo hàm riêng phần quá nhỏ thì Δw nhỏ, nếu đạo hàm riêng phần lớn thì Δw lớn. Độ lớn của đạo hàm riêng phần thay đổi không thể biết trước được theo hình dạng của hàm lỗi E trong mỗi lần lặp. Do đó quá trình học không ổn định.

Để cho quá trình học ổn định người ta thêm vào một *hệ số quán tính* (momentum term)

$$\Delta w_{ij}(t) = -\eta \frac{\partial E}{\partial w_{ij}}(t) + \mu * \Delta w_{ij}(t-1) \quad (2.14)$$

Hệ số quán tính μ có tác dụng điều chỉnh mức độ ảnh hưởng của giá trị $\Delta w_{ij}(t-1)$ ở bước lặp trước lên giá trị $\Delta w_{ij}(t)$. Hệ số này có tác dụng giúp cho giải thuật không bị dừng ở tối ưu cục bộ và các vùng phẳng của bề mặt lỗi. Nó cũng giúp tăng giá trị cập nhật ở những vùng mà độ dốc không đổi, do đó tăng tốc độ hội tụ [9].

Hình 2.9 là mã giả cho giải thuật lan truyền ngược theo phương pháp học trực tuyến có áp dụng hệ số quán tính.

Giải thuật lan truyền ngược cần hai thông số nhập vào đó là hệ số học và hệ số quán tính. Đối với mỗi bài toán khác nhau các thông số này cần có các giá trị khác nhau để đạt được sự hiệu quả trong quá trình học. Việc xác định các thông số này một cách đúng đắn không phải là một việc dễ dàng cần nhiều công sức và kinh nghiệm.

2.3.3. Giải thuật RPROP

Giải thuật lan truyền ngược gặp một vấn đề ở chỗ giá trị cập nhật trọng số ($\Delta w_{ij}(t)$) không những phụ thuộc vào dấu của đạo hàm riêng phần mà còn bị ảnh hưởng

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa bởi độ lớn của nó, điều này làm cho quá trình học không được ổn định. Việc thêm vào hệ số quán tính không giải quyết trọn vẹn vấn đề bởi vì ta không biết giá trị tối ưu cho hệ số này là bao nhiêu.

1. Khởi tạo tất cả các trọng số bằng các số nhỏ ngẫu nhiên
2. **Loop until** điều kiện dừng thỏa
 - 2.1. **For each** mỗi mẫu trong tập dữ liệu, **do**
 - 2.1.1. Nhập mẫu vào mạng và tính toán giá trị đầu ra.
 - 2.1.2. **For each** mỗi giá trị xuất của đơn vị k

$$\text{Tính } \frac{\partial E}{\partial w_{kj}}$$
 - 2.1.3. **For each** đơn vị ẩn h , từ lớp ẩn cuối cùng đến lớp ẩn đầu tiên
 - 2.1.3.1. $\text{Tính } \frac{\partial E}{\partial w_{hj}}$
 - 2.1.4. **For each** w_{ij} trong mạng
 - 2.1.4.1. $\text{Tính } \Delta w_{ij}(t) = -\varepsilon \frac{\partial E}{\partial w_{ij}}(t) + \mu * \Delta w_{ij}(t-1)$
 - 2.1.4.2. $\text{Tính } w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$
 - 2.1.5. **End for**
 - 2.2. **End for**
3. **End loop**

Hình 2.9: Giải thuật lan truyền ngược

Nhiều giải thuật đã được phát minh để giải quyết các vấn đề trên, chúng có thể được phân ra làm hai loại: *chiến lược toàn cục* (global strategy) và *chiến lược cục bộ* (local strategy). Chiến lược toàn cục sử dụng kiến thức về trạng thái của toàn bộ mạng để hiệu chỉnh các thông số toàn cục, trong khi chiến lược cục bộ dùng các thông tin riêng của từng trọng số một để thích nghi các thông số đặc biệt cho từng trọng số [13].

RPROP là viết tắt của từ ‘resilient propagation’, nghĩa là lan truyền đàn hồi là một phương pháp thích nghi cục bộ. RPROP thực hiện cập nhập các trọng số w_{ij} dựa vào thông tin về dấu của các đạo hàm riêng phần điều này giúp nó tránh được sự ảnh hưởng của độ lớn của các đạo hàm riêng phần này. Để thực hiện điều này các trọng số

sẽ có một giá trị cập nhập riêng Δ_{ij} chỉ phụ thuộc vào dấu của $\frac{\partial E}{\partial w_{ij}}$. Giá trị này được

cập nhập trong quá trình học theo quy luật sau:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ * \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ \eta^- * \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)}, & \text{else} \end{cases} \quad (2.15)$$

Ở đây $0 < \eta^- < 1 < \eta^+$ là các hệ số cố định của quá trình học dùng để hiệu chỉnh các giá trị cập nhập cho từng trọng số tùy theo hình dạng của hàm lỗi.

Mỗi lần đạo hàm riêng phần theo trọng số w_{ij} của hàm lỗi E đổi dấu, nghĩa là giá trị cập nhập vừa thực hiện là quá lớn và giải thuật đã nhảy vượt qua điểm tối ưu cục bộ thì giá trị cập nhập Δ_{ij} sẽ giảm đi theo một thừa số η^- . Ngược lại nếu đạo hàm riêng phần vẫn giữ nguyên dấu thì giá trị cập nhập Δ_{ij} sẽ được tăng lên để tăng tốc độ hội tụ. Cứ mỗi lần giá trị cập nhập được biết thì các trọng số được điều chỉnh theo luật sau: nếu đạo hàm riêng phần dương thì trọng số được giảm đi một lượng bằng với giá trị cập nhập (Δ_{ij}), nếu đạo hàm riêng phần âm thì giá trị cập nhập được cộng thêm vào trọng số.

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)}, & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)}, & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0, & \text{else} \end{cases} \quad (2.16)$$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)} \quad (2.17)$$

Tuy nhiên có một trường hợp đặc biệt đó là khi đạo hàm riêng phần đổi dấu, nghĩa là bước cập nhập trước quá lớn làm cho điểm tối ưu bị nhảy vượt qua. Giá trị

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa
trọng số phải được trả về giá trị trước khi thay đổi, giá trị cập nhập Δ_{ij} sẽ được giảm xuống và ở bước kế sau ta sẽ không cập nhập giá trị này. Thực tế ta có thể làm việc này bằng cách gán

$$\Delta w_{ij}^{(t)} = -\Delta w_{ij}^{(t-1)} \text{ và } \frac{\partial E^{(t)}}{\partial w_{ij}} = 0$$

Giải thuật RPROP thực hiện việc thích nghi các giá trị cập nhập của các trọng số tùy theo độ dốc của hàm lỗi E , mà thông tin về tổng độ dốc của hàm lỗi cho toàn bộ tập dữ liệu mẫu đáng tin hơn thông tin về độ dốc chỉ cho một mẫu trong tập mẫu nên giải thuật RPROP thực hiện theo mô hình học theo bó (học theo epoch). Các thông tin về đạo hàm riêng phần sẽ được cộng dồn qua từng mẫu trong tập huấn luyện và các trọng số sẽ được cập nhập sau khi đã duyệt qua hết các mẫu.

Giải thuật RPROP ban đầu cũng thực hiện các bước giống như giải thuật lan truyền ngược, các thông tin về đạo hàm riêng phần của hàm lỗi theo các trọng số sẽ được lan truyền ngược từ các lớp sau đến các lớp trước. Khi các thông tin về các đạo hàm riêng phần này có đủ thì giải thuật sẽ thực hiện việc cập nhập các trọng số theo các quy tắc nêu ở trên. Hình 2.10 là mã giả cho phần cập nhập trọng số của giải thuật RPROP.

Ở đây hàm số *minimum* và *maximum* lần lượt là hai hàm trả về giá trị nhỏ nhất và lớn nhất của hai số. Hàm số *sign(x)* trả về 1 nếu x dương, trả về -1 nếu x âm và trả về 0 trong các trường hợp còn lại.

Ban đầu các giá trị cập nhập Δ_{ij} sẽ được khởi tạo một giá trị dương ban đầu Δ_0 . Lựa chọn tốt cho Δ_0 là 0.1 nhưng theo các nghiên cứu thì việc lựa chọn tham số này không ảnh hưởng nhiều đến tốc độ hội tụ của giải thuật [13]. Các thông số về Δ_{min} và Δ_{max} để tránh vấn đề tràn số của các biến thực. Giá trị Δ_{min} được thiết lập thường là 1.0e-6, còn giá trị Δ_{max} là 50.0. Thông thường độ hội tụ của giải thuật không bị ảnh hưởng bởi các thông số này nhưng đôi khi thông số Δ_{max} được chọn là một giá trị nhỏ (ví dụ 1.0) để ngăn giải thuật không rơi quá nhanh vào một cực tiểu cục bộ [13]. Hai thông số η^+ và η^- được cố định ở hai giá trị lần lượt là 1.2 và 0.5, để việc lựa chọn các

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa tham số cho giải thuật được đơn giản. Trong thực tế, hai thông số cần được lựa chọn cho giải thuật RPROP là Δ_0 và Δ_{max} .

Một trong các thuận tiện của RPROP là trong nhiều bài toán không cần phải lựa chọn các tham số một cách cẩn thận cũng đạt được tốc độ hội tụ tối ưu hay gần tối ưu [13].

$$\begin{aligned}
 & \textbf{For} \text{ mọi trọng số và độ lệch } \{ \\
 & \quad \textbf{if} \left(\frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \right) \textbf{ then } \{ \\
 & \quad \quad \Delta_{ij}^{(t)} = \text{minimum} \left(\Delta_{ij}^{(t-1)} * \eta^+, \Delta_{max} \right) \\
 & \quad \quad \Delta w_{ij}^{(t)} = - \text{sign} \left(\frac{\partial E^{(t)}}{\partial w_{ij}} \right) * \Delta_{ij}^{(t)} \\
 & \quad \quad w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)} \\
 & \quad \} \textbf{ else if } \left(\frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \right) \textbf{ then } \{ \\
 & \quad \quad \Delta_{ij}^{(t)} = \text{maximum} \left(\Delta_{ij}^{(t-1)} * \eta^-, \Delta_{min} \right) \\
 & \quad \quad w_{ij}^{(t+1)} = w_{ij}^{(t)} - \Delta w_{ij}^{(t-1)} \\
 & \quad \quad \frac{\partial E^{(t)}}{\partial w_{ij}} = 0 \\
 & \quad \} \\
 & \quad \textbf{else if} \left(\frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} = 0 \right) \textbf{ then } \{ \\
 & \quad \quad \Delta w_{ij}^{(t)} = - \text{sign} \left(\frac{\partial E^{(t)}}{\partial w_{ij}} \right) * \Delta_{ij}^{(t)} \\
 & \quad \quad w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)} \\
 & \quad \} \\
 & \}
 \end{aligned}$$

Hình 2.10: Giải thuật RPROP

Trong giải thuật lan truyền ngược, giá trị cập nhật trọng số phụ thuộc vào độ lớn của đạo hàm riêng phần của hàm lỗi theo trọng số, mà giá trị này lại giảm theo khoảng cách của các trọng số đối với lớp xuất. Do đó các trọng số ở xa lớp xuất sẽ ít

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa được hiệu chỉnh hơn và việc huấn luyện các trọng số này sẽ chậm hơn các trọng số gần lớp xuất. Tuy nhiên khi dùng RPROP thì giá trị cập nhật trọng số chỉ phụ thuộc vào dấu nên sự huấn luyện sẽ trải đều trên toàn bộ mạng: những trọng số gần lớp nhập cũng có cơ hội được cập nhật và phát triển ngang với các trọng số gần lớp xuất [13].

2.3.4. Hiện tượng quá khớp

Trong việc huấn luyện mạng neuron, đôi khi ta gặp phải hiện tượng mạng xấp xỉ rất tốt tập dữ liệu huấn luyện nhưng cho ra kết quả dự đoán thiếu chính xác, giảm khả năng tổng quát của mạng. Đây được gọi là hiện tượng *quá khớp* (overfitting).

Để đảm bảo tính tổng quát hóa của mạng và tránh hiện tượng quá khớp, ta cần chuẩn bị một tập dữ liệu kiểm tra. Tập dữ liệu này được sử dụng trong giai đoạn huấn luyện, sau khi huấn luyện xong một cấu hình mạng, ta cần tiến hành kiểm tra trên tập dữ liệu này để xem mạng có xấp xỉ tốt tập dữ liệu kiểm tra này hay không. Nếu sai số kiểm tra nhỏ thì mô hình mạng vừa được huấn luyện có khả năng tổng quát hóa tốt và có thể được sử dụng để dự báo. Ngược lại, nếu sai số kiểm tra lớn, ta buộc phải thực hiện việc huấn luyện mạng lại.

Chương 3

ỨNG DỤNG MẠNG NEURON NHÂN TẠO VÀO CÔNG TÁC DỰ BÁO DỮ LIỆU CHUỖI THỜI GIAN

Trong chương này, chúng tôi sẽ giới thiệu về dữ liệu chuỗi thời gian và các đặc tính, mô hình của nó cùng với cách ứng dụng mạng neuron nhân tạo trong việc dự báo dữ liệu chuỗi thời gian.

3.1. Dữ liệu chuỗi thời gian

3.1.1. Dữ liệu chuỗi thời gian là gì

Trong bài toán dự báo, một kiểu dữ liệu thường gặp là dữ liệu chuỗi thời gian, tức là dữ liệu được thu nhập, lưu trữ và quan sát theo sự tăng dần của thời gian. Ví dụ, số lượng thí sinh dự thi đại học vào Trường Đại Học Bách Khoa thành phố Hồ Chí Minh được lưu trữ theo từng năm, hay số lượng hàng hóa đã bán được của một siêu thị được lưu trữ theo từng quý là các dữ liệu chuỗi thời gian.

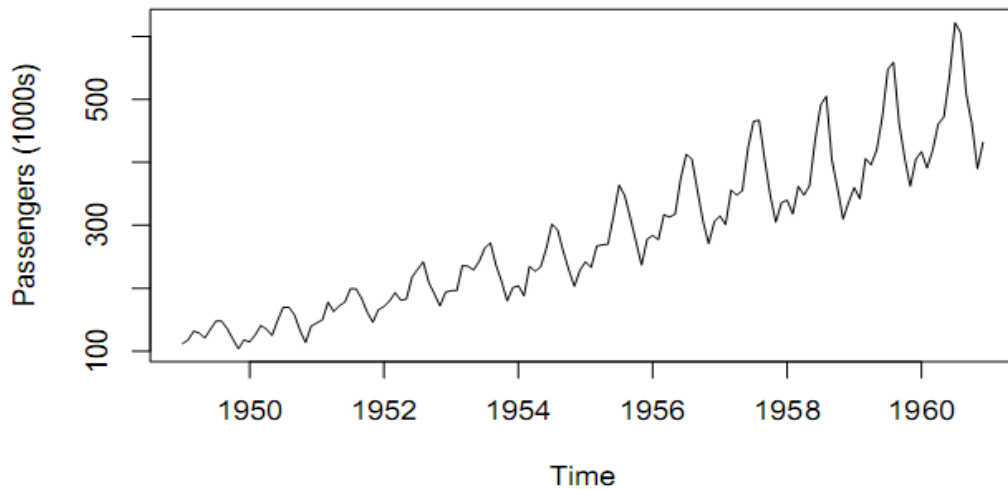
Ta ký hiệu chuỗi thời gian là $\{X_t\}$ với t là các số tự nhiên. X_t là các *biến ngẫu nhiên* (random variable) rút ra từ một *phân bố xác suất* (probability distribution) nào đó. Các chuỗi thời gian thường được biểu diễn bằng một đồ thị với trục hoành là biến thời gian. Hình 3.1 là một ví dụ về chuỗi thời gian, số hành khách đặt chỗ hàng tháng của hãng Pan Am.

Trong chuỗi thời gian các giá trị ở những thời điểm khác nhau có mối tương quan với nhau. Sự tương quan này được đánh giá bằng *hệ số tự tương quan* (autocorrelation).

Tự tương quan là sự tương quan của một biến với chính nó theo những độ trễ thời gian khác nhau [5]. Hệ số tự tương quan là đại lượng biểu diễn mức độ tự tương quan và được tính theo công thức sau:

$$\rho_k = \frac{E[(X_t - \mu)(X_{t+k} - \mu)]}{\sqrt{\text{Var}(X_t)\text{Var}(X_{t+k})}} \quad (3.1)$$

Ở đây ρ_k là hệ số tự tương quan của X ở độ trễ k , μ là giá trị trung bình của X_t . Ký hiệu $E(X)$ là kỳ vọng của biến ngẫu nhiên X , $\text{Var}(X)$ là phương sai của biến ngẫu nhiên X .



Hình 3.1: Số khách hàng đặt chỗ hàng tháng của hãng Pan Am

Các hệ số tự tương quan của một biến ngẫu nhiên theo các độ trễ khác nhau được biểu diễn trên một đồ thị có trục hoành là các độ trễ và trục tung là các hệ số tự tương quan làm thành *hàm tự tương quan* (autocorrelation function) của biến đó. Hàm tự tương quan là một công cụ quan trọng giúp xác định các thành phần cơ bản của chuỗi thời gian.

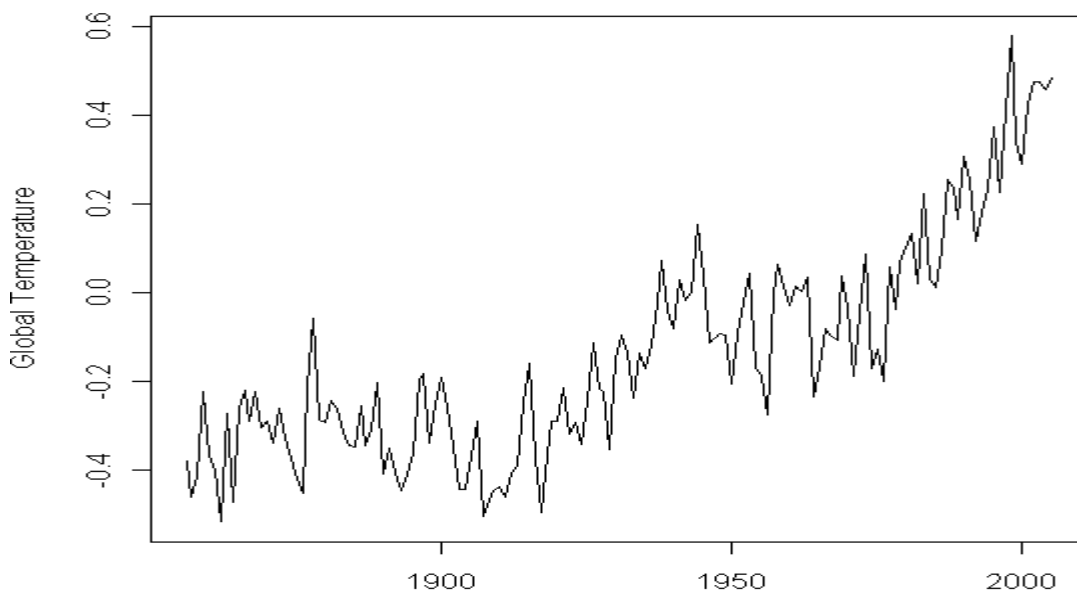
3.1.2. Các thành phần của dữ liệu chuỗi thời gian

Trong thực tế, khi quan sát chuỗi thời gian ta nhận thấy bốn thành phần ảnh hưởng lên mỗi giá trị của chuỗi thời gian đó là *xu hướng* (trend), *chu kỳ* (cyclical), *mùa* (seasonal), *bất quy tắc* (irregular).

Thành phần xu hướng

Là thành phần thể hiện sự tăng hay giảm giá trị của chuỗi thời gian trong một giai đoạn dài hạn nào đó [5]. Ta có thể xác định một chuỗi thời gian có chứa thành phần xu

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa hướng hay không bằng việc kiểm tra hàm tự tương quan của nó. Nếu một chuỗi thời gian có thành phần xu hướng sẽ có hệ số tự tương quan rất lớn ở những độ trễ đầu tiên và giảm dần về 0 khi độ trễ tăng lên. Hình 3.2 và 3.3 là một minh họa về chuỗi thời gian có thành phần xu hướng. Ở đây dù mức tăng nhiệt độ toàn cầu có biến đổi theo từng năm nhưng nhìn chung mức tăng nhiệt độ trung bình có xung hướng tăng theo thời gian. Hệ số tự tương quan rất lớn ở những độ trễ đầu tiên và giảm dần theo sự tăng của độ trễ.



Hình 3.2: Độ tăng nhiệt độ trung bình hàng năm từ 1856 đến 2005

Thành phần chu kỳ

Là chuỗi biến đổi dạng sóng quanh xu hướng [5]. Trong thực tế thành phần này rất khó xác định và người ta thường xem nó như là một phần của thành phần xu hướng.

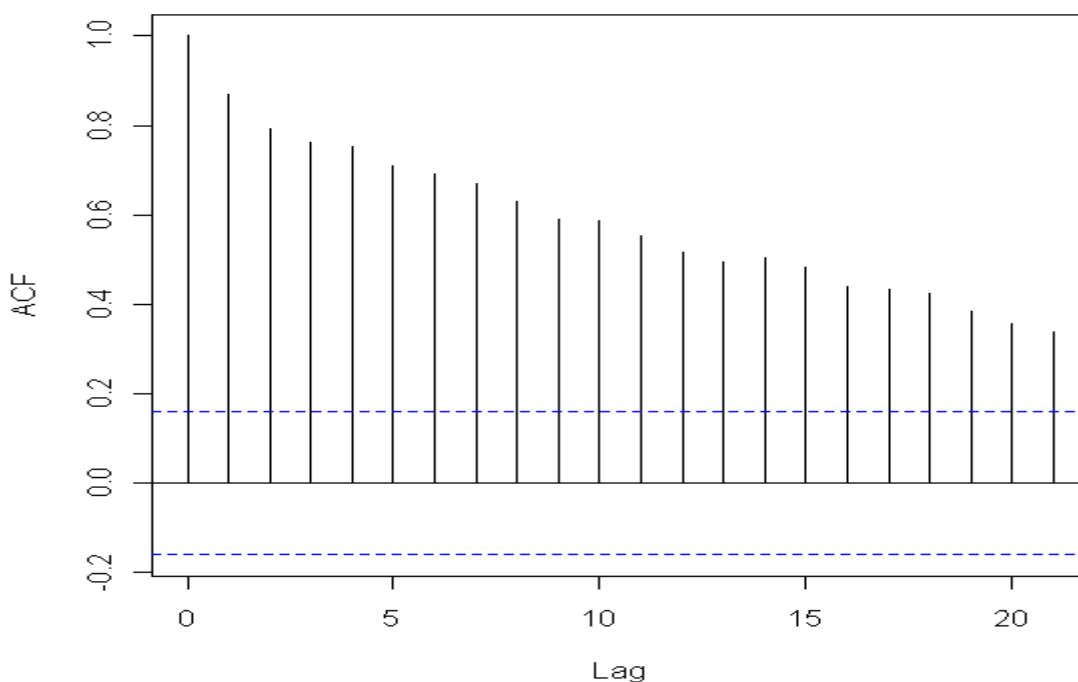
Thành phần bất quy tắc

Là thành phần thể hiện sự biến đổi ngẫu nhiên không thể đoán được của chuỗi thời gian [5].

Thành phần mùa

Là thành phần thể hiện sự biến đổi lặp đi lặp lại tại từng thời điểm cố định theo từng năm của chuỗi thời gian [5]. Đối với chuỗi thời gian có thành phần mùa thì giá trị

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa tại những thời điểm cố định theo từng năm sẽ có sự tương quan lớn với nhau. Ví dụ một chuỗi thời gian được ghi nhận theo từng quý có tính chất mùa thì hệ số tự tương quan ở độ trễ là 4 sẽ khác không một cách có ý nghĩa. Hình 3.4 là đồ thị của một chuỗi thời gian có tính mùa.



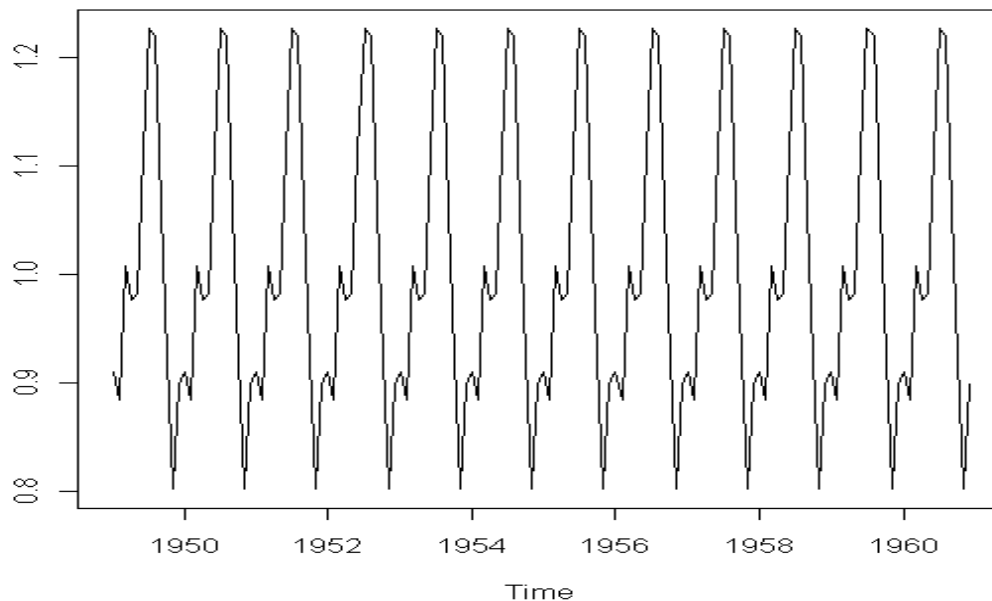
Hình 3.3: Hàm tự tương quan của chuỗi tăng nhiệt độ trung bình hàng năm (1856 - 2005)

Việc xác định một chuỗi thời gian có thành phần xu hướng hay thành phần mùa hay không rất quan trọng trong bài toán dự đoán chuỗi thời gian. Nó giúp ta lựa chọn được mô hình dự đoán phù hợp hay giúp cải tiến mô hình đã có chính xác hơn.

3.1.1. Các mô hình của chuỗi thời gian

Trong việc phân tích dữ liệu chuỗi thời gian, người ta thường cố gắng xác định các thành phần của chuỗi thời gian, xem các thành phần này có ảnh hưởng như thế nào đến chuỗi thời gian ban đầu và từ đó có thể dùng các thành phần này để dự đoán. Tuy nhiên trong thực tế, giá trị của các thành phần của chuỗi thời gian không quan sát được mà chỉ có thể ước lượng từ giá trị của chuỗi thời gian. Để làm được điều này ta phải xây dựng một mô hình toán học mô tả chuỗi thời gian theo các thành phần của nó. Có hai mô hình thường dùng để mô tả chuỗi thời gian theo các thành phần của nó là *mô*

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa
hình có tính cộng (additive components model) và *mô hình có tính nhân* (multiplicative components model).



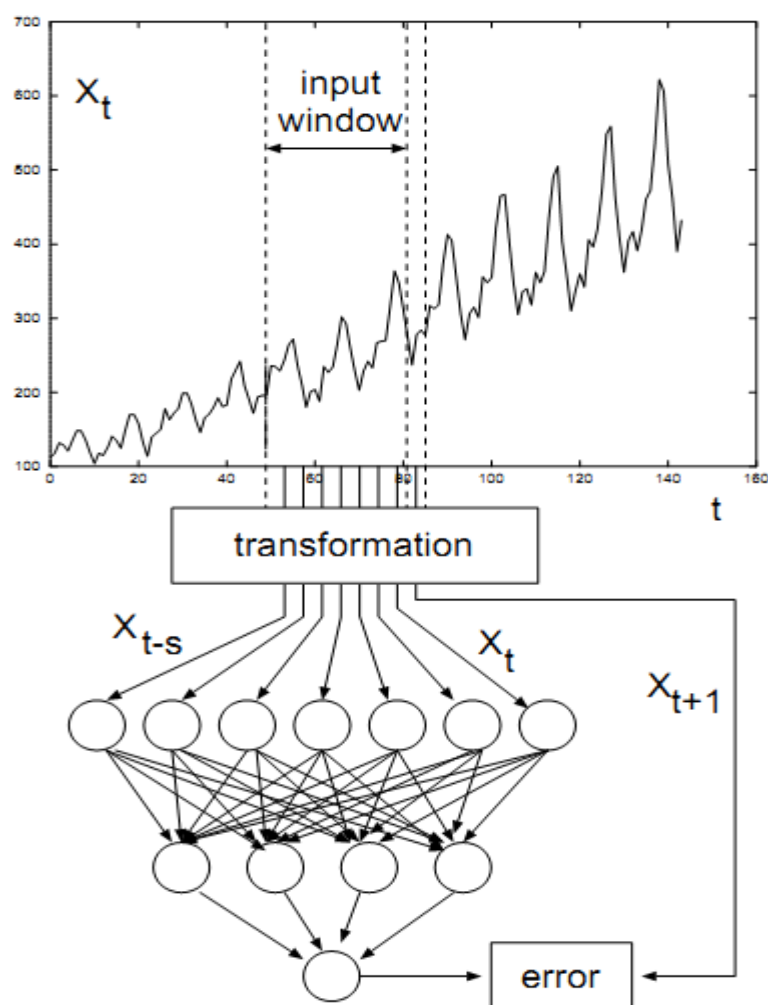
Hình 3.4: Chuỗi thời gian có tính mùa

Mô hình có tính cộng mô tả giá trị của chuỗi thời gian là tổng các thành phần của nó. Mô hình này phù hợp cho việc phân tích chuỗi thời gian mà sự biến đổi của nó đều đặn và hầu như không đổi theo chiều dài của chuỗi thời gian [5]. Chuỗi thời gian như vậy còn gọi là chuỗi thời gian có tính cộng.

Mô hình có tính nhân mô tả giá trị của chuỗi thời gian là tích các thành phần của nó. Mô hình này phù hợp cho việc phân tích chuỗi thời gian mà sự biến đổi của nó tăng theo xu hướng [5]. Chuỗi thời gian như vậy còn gọi là chuỗi thời gian có tính nhân.

3.2. Áp dụng mạng neuron nhân tạo vào dự báo dữ liệu chuỗi thời gian

Việc sử dụng mạng neuron nhân tạo vào việc dự báo dữ liệu chuỗi thời gian dựa chủ yếu vào dữ liệu mà ta thu nhập. Mạng neuron nhân tạo truyền thẳng với ít nhất một lớp ẩn và đủ số đơn vị cho lớp ẩn có thể xấp xỉ bất kỳ hàm khả đánh giá (measurable function) tuyến tính hay phi tuyến nào [12]. Hình 3.5 minh họa cho quá trình sử dụng mạng neuron nhân tạo đối với dữ liệu chuỗi thời gian.



Hình 3.5: Mô hình học với chuỗi thời gian

Như đã đề cập ở trên, dữ liệu chuỗi thời gian là dữ liệu được thu nhập, lưu trữ và quan sát theo sự tăng dần của thời gian X_1, X_2, \dots, X_n .

Mạng neuron học cấu hình mạng từ dữ liệu chuỗi thời gian bằng cách ánh xạ từ một vector dữ liệu đầu vào sang dữ liệu đầu ra. Một số lượng dữ liệu ở thời đoạn trước (thường là liên tiếp) của dữ liệu chuỗi thời gian (cửa sổ đầu vào $X_{t-s}, X_{t-s+1}, \dots, X_t$) được ánh xạ sang khoảng thích hợp (ví dụ $[0,1]$ hoặc $[-1,1]$) và được sử dụng như dữ liệu đầu vào của tầng nhập. Giá trị s của “cửa sổ đầu vào” tương ứng với số đơn vị ở tầng nhập. Trong giai đoạn truyền tiến, những giá trị đó được truyền qua tầng ẩn rồi đến các đơn vị đầu ra. Khi truyền tới đơn vị đầu ra, giá trị lỗi được tính toán dựa vào sự khác biệt giữa giá trị đầu ra với giá trị của dữ liệu chuỗi thời gian tại thời điểm $t+1$. Sau đó,

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa giá trị lỗi này được truyền ngược lại tới các kết nối giữa tầng ẩn và tầng đầu ra, kết nối giữa tầng đầu vào và tầng ẩn để cập nhập lại trọng số của các kết nối này.

Các cửa sổ đầu vào có thể được chọn một cách ngẫu nhiên hoặc liên tiếp nhau từ dữ liệu chuỗi thời gian. Chọn cửa sổ đầu vào một cách ngẫu nhiên sẽ phức tạp hơn, tuy nhiên sẽ đảm bảo cấu hình mạng tốt hơn và tránh được lỗi tối ưu cục bộ [12].

3.3. Các bước xây dựng một mô hình mạng neuron nhân tạo để dự báo dữ liệu chuỗi thời gian

Theo Kaastra và các cộng sự [10], quá trình xây dựng một mô hình mạng neuron cho bài toán dự báo thường gồm 8 bước:

- Lựa chọn các biến
- Thu thập dữ liệu
- Tiền xử lý dữ liệu
- Phân chia tập dữ liệu
- Xây dựng cấu trúc mạng
- Xác định tiêu chuẩn đánh giá
- Huấn luyện mạng
- Dự đoán và cải tiến

Quá trình này thường không phải là một quá trình liên tiếp các bước, một số bước có thể được lặp lại đặc biệt là: lựa chọn các biến và huấn luyện mạng

3.3.1. Lựa chọn các biến

- Thành công trong việc xây dựng một mạng neuron phụ thuộc vào việc hiểu rõ ràng vấn đề cần giải quyết. Biết được những biến nào cần được xem xét là điểm mấu chốt.
- Trong bài toán dự báo các dữ liệu thương mại thì các học thuyết kinh tế có thể giúp chọn lựa các biến là các chỉ số kinh tế quan trọng. Đối với một bài toán cụ thể cần thực hiện xem xét các vấn đề lý thuyết mà từ đó sẽ xác định được các nhân tố ảnh hưởng đến bài toán. Tại bước này trong quá trình thiết kế, điều cần quan tâm đó là

các dữ liệu thô từ đó có thể phát triển thành các chỉ số quan trọng. Các chỉ số này sẽ tạo ra các đầu vào cho mạng.

- Khi lựa chọn các biến, ta có thể chọn biến kỹ thuật hoặc biến cơ bản. Biến kỹ thuật bao gồm các giá trị cũ, trong quá khứ của biến đó hoặc các chỉ số được tính toán từ các giá trị cũ đó. Biến cơ bản bao gồm dữ liệu của các biến khác mà ảnh hưởng đến biến đang xem xét. Mô hình neuron đơn giản nhất sử dụng các dữ liệu của biến kỹ thuật hoặc *lấy hiệu* (differencing) của nó như dữ liệu đầu vào của mạng. Hiệu của một chuỗi thời gian $\{X_t\}$ cũng là một chuỗi thời gian $\{Y_t\}$, với các giá trị $Y_t = X_{t+1} - X_t$. Việc lấy hiệu có thể loại bỏ tính xu hướng hay tính mùa của một chuỗi thời gian và làm cho việc xấp xỉ nó đơn giản hơn. Một mô khác cũng được áp dụng phổ biến là sử dụng dữ liệu của các biến cơ bản trong quá khứ để dự đoán.
- Tần suất của dữ liệu được ghi nhận phụ thuộc vào mục đích của nhà dự báo. Nếu dùng để dự đoán tình hình giao dịch chứng khoán thì dữ liệu được ghi nhận hằng ngày. Đối với các vấn đề đầu tư dài hạn thì các dữ liệu hàng tuần, hàng tháng được dùng làm đầu vào cho mạng neuron.

3.3.2. Thu thập dữ liệu

- Ta cần phải xem xét chi phí và khả năng có thể thu thập được dữ liệu của các biến đã chọn ra ở bước trước. Các dữ liệu kỹ thuật có thể thu thập được dễ dàng và chi phí ít tốn kém hơn là các dữ liệu cơ bản. Để đảm bảo tính chính xác của mạng neuron, ta phải đảm dữ liệu có chất lượng cao. Sau khi được thu thập, các dữ liệu phải được kiểm tra để đảm tính hợp lệ, tính nhất quán và tránh các dữ liệu bị thiếu sót.
- Các dữ liệu bị thiếu sót thường xuyên xuất hiện và có thể được xử lý bằng nhiều cách khác nhau. Các dữ liệu bị thiếu sót có thể được bỏ qua hoặc chúng có thể xem như không thay đổi so với dữ liệu trước nó, và được tính toán bằng phương pháp nội suy hoặc trung bình các giá trị lân cận.

3.3.3. Tiền xử lý dữ liệu

- Tiền xử lý dữ liệu liên quan đến việc phân tích và chuyển đổi giá trị các tham số đầu vào, đầu ra mạng để tối thiểu hóa nhiễu, nhấn mạnh các đặc trưng quan

trọng, phát hiện các xu hướng và cân bằng phân bố của dữ liệu. Bởi vì, mạng neuron dùng để học mẫu từ tập dữ liệu, sự biểu diễn dữ liệu có vai trò quyết định trong việc học các mẫu thích hợp. Các dữ liệu dùng cho đầu vào, đầu ra của mạng neuron hiếm khi được đưa trực tiếp vào mạng dưới dạng dữ liệu thô. Chúng thường được chuẩn hóa vào khoảng giữa cận trên và cận dưới của hàm chuyển (thường là giữa đoạn $[0;1]$ hoặc $[-1;1]$).

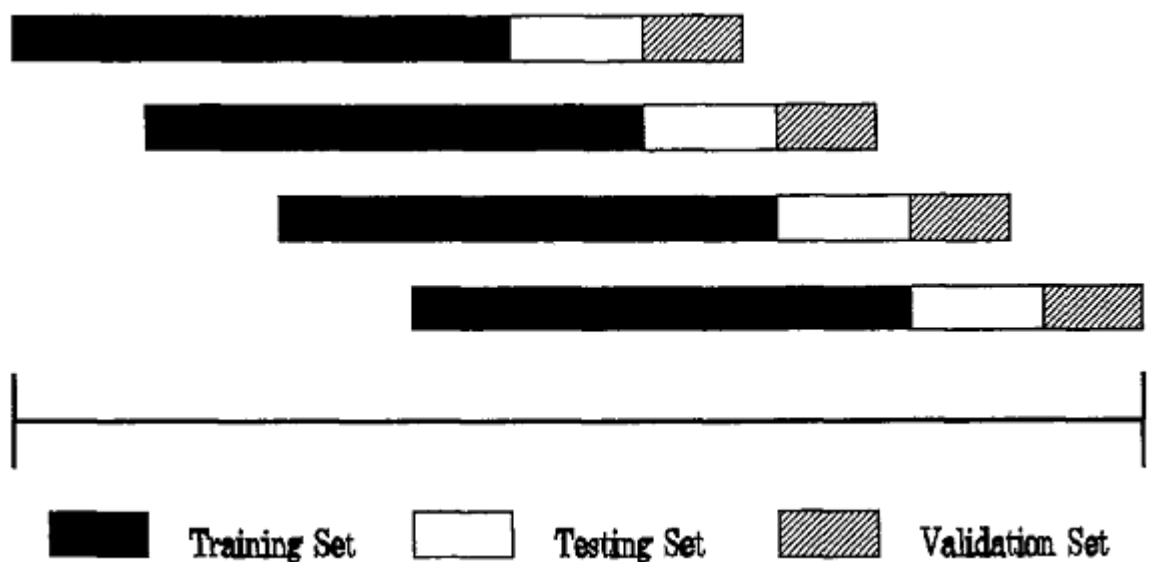
- Hai phương pháp chuyển đổi dữ liệu thường dùng nhất là lấy hiệu và lấy logarit tự nhiên của biến số. Lấy hiệu sử dụng sự thay đổi trong giá trị của biến số, nó có thể được sử dụng để loại bỏ khuynh hướng tuyến tính của dữ liệu. Việc lấy logarit tự nhiên của biến số là hữu ích trong trường hợp biến số lấy các giá trị rất khác nhau, sự thay đổi trong giá trị rất lớn. Việc lấy logarit tự nhiên đồng thời cũng có thể chuyển từ mối liên hệ tỷ lệ sang mối liên hệ cộng giữa các biến dự báo.
- Ngoài phương pháp lấy hiệu và lấy logarit tự nhiên của biến số, ta có thể sử dụng tỉ số của biến đầu vào, trung bình di động. Ta có thể kết hợp các phương pháp để hạn chế dư thừa dữ liệu và cung cấp mạng với tính chính xác cao.

3.3.4. Phân chia tập dữ liệu

- Trong thực tế, khi huấn luyện, người ta thường chia tập dữ liệu thành các tập: huấn luyện, kiểm tra và kiểm định (ngoài các mẫu). Tập huấn luyện thường là tập lớn nhất được sử dụng để huấn luyện cho mạng. Tập kiểm tra thường chứa khoảng 10% đến 30% tập dữ liệu huấn luyện, được sử dụng để kiểm tra mức độ tổng quát hóa của mạng sau khi huấn luyện. Kích thước của tập kiểm định cần được cân bằng giữa việc cần có đủ số mẫu để có thể kiểm tra mạng đã được huấn luyện và việc cần có đủ các mẫu còn lại cho cả pha huấn luyện và kiểm tra. Tập kiểm định nên bao gồm các giá trị liên tục mới nhất.
- Có hai cách thực hiện xác định tập kiểm tra. Một là lấy ngẫu nhiên các mẫu từ tập huấn luyện ban đầu. Lợi điểm của cách này là có thể tránh được nguy hiểm khi mà đoạn dữ liệu được chọn có thể chỉ diễn hình cho một tính chất của dữ liệu (đang tăng hoặc đang giảm). Hai là chỉ lấy các dữ liệu ở phần sau của tập

huấn luyện, trong trường hợp các dữ liệu gần với hiện tại là quan trọng hơn các dữ liệu quá khứ.

- Tập dữ liệu kiểm tra ngẫu nhiên không nên lặp lại trong tập huấn luyện, bởi vì điều này có thể làm mất khả năng tổng quát hóa của mạng neuron, đặc biệt trong trường hợp kích thước của tập kiểm tra tương đối lớn so với tập huấn luyện (khoảng 30%). Phương pháp tắt định, như sử dụng mỗi dữ liệu thứ n làm dữ liệu kiểm tra, cũng không nên được sử dụng bởi vì nó chịu ảnh hưởng bởi tính chu kỳ của dữ liệu.
- Một phương pháp chặt chẽ dùng để đánh giá mạng neuron nhân tạo là walk-forward. Phương pháp walk-forward chia tập dữ liệu thành một chuỗi các tập dữ liệu nhỏ hơn huấn luyện-kiểm tra-kiểm định gộp chồng lên nhau được minh họa tại hình 3.6.



Hình 3.6: Thủ tục sử dụng phương pháp walk-forward chia tập dữ liệu

3.3.5. Xây dựng cấu trúc mạng

- Phương pháp thực hiện xây dựng cấu trúc mạng neuron bao gồm việc xác định sự liên kết giữa các neuron, đồng thời xác định cấu trúc của mạng bao gồm số lớp ẩn, số neuron trong từng lớp. Ta có thể thực hiện lựa chọn số neuron trong

các lớp ẩn bằng cách bắt đầu bằng một số nào đó dựa trên các luật. Sau khi thực hiện huấn luyện, kiểm tra lỗi tổng quát hóa của từng cấu trúc, có thể tăng hoặc giảm số các neuron.

- Việc thiết kế cấu hình một mạng neuron có ý nghĩa quyết định quan trọng trong việc dự đoán dự đoán dữ liệu chuỗi thời gian. Nếu xây dựng mạng có quá nhiều tầng ẩn, hoặc số lượng đơn vị ở mỗi tầng quá nhiều sẽ dẫn đến vấn đề quá khớp. Tức là khi đó, cấu hình mạng neuron giải thích tập dữ liệu huấn luyện rất tốt, nhưng lại không có khả năng tổng quát hóa, vì thế không thể dùng cấu hình này để dự đoán. Tuy nhiên số tầng hoặc số đơn vị trên mỗi tầng quá ít thì mạng neuron không có khả năng giải thích và dự đoán tốt các chuỗi thời gian phức tạp.
- Thực tế đã chứng minh: một mạng neuron với một tầng đầu vào, một tầng ẩn, một tầng đầu ra cùng với sự thay đổi số đơn vị tại mỗi tầng là đủ để xấp xỉ bất kỳ một hàm liên tục nào [10]. Thông thường các mạng neuron được khởi tạo với một hoặc nhiều nhất là hai lớp ẩn. Nếu kết quả huấn luyện từ mạng trên mà vẫn không thỏa mãn sau khi đã thử với nhiều giá trị khởi tạo ngẫu nhiên của trọng số thì ta nên xem xét hiệu chỉnh lại số đơn vị trên các lớp ẩn hay kiểm tra dữ liệu đầu vào (ví dụ dữ liệu dùng để huấn luyện mạng có phải đã lỗi thời không?) chứ không nên tăng thêm số tầng ẩn. Cả lý thuyết và các kết quả thực nghiệm gần đây đều kết luận rằng các mạng với hơn hai tầng ẩn sẽ không cải thiện được kết quả dự đoán [10].
- Số lượng đơn vị trong mỗi lớp cũng là một vấn đề cần phải xem xét vì nó cũng ảnh hưởng nhiều đến chất lượng của công tác dự báo. Số lượng các đơn vị ở tầng xuất luôn là 1 cho bài toán dự báo chuỗi thời gian. Tuy nhiên việc chọn số đơn vị cho tầng ẩn và tầng nhập là việc không dễ. Số đơn vị ở tầng nhập bằng số giá trị trong cửa sổ nhập, việc lựa chọn này dựa trên giả định của nhà dự báo về giá trị tại thời điểm hiện tại của chuỗi thời gian sẽ bị chi phối chủ yếu bởi giá trị của bao nhiêu thời điểm trước nó. Việc lựa chọn thông số này phụ thuộc vào kinh nghiệm và sự hiểu biết của nhà dự báo vào chuỗi thời gian đang xét. Số lượng đơn vị ở tầng ẩn cũng là một thông số cần phải lựa chọn cẩn thận và cũng không có một thủ tục hình thức nào giúp ta xác định được một cách tối ưu thông

số này. Việc lựa chọn sao cho phù hợp phải dựa vào thực nghiệm. Thông thường có hai cách chủ yếu để tìm giá trị tối ưu cho số đơn vị ở lớp ẩn. Cách thứ nhất ta chuẩn bị một nhóm các mạng neuron chỉ khác nhau số đơn vị ở lớp ẩn (số lượng đơn vị có thể tăng dần theo một, hai hoặc ba), sau đó ta thực hiện huấn luyện và kiểm tra các mạng này trên tập dữ liệu đã chuẩn bị. Mạng neuron có sai số nhỏ nhất là là mạng có cấu hình tốt nhất. Phương pháp này khá tốn thời gian nhưng khá hiệu quả. Cách thứ hai là thay đổi số đơn vị trong lớp ẩn ngay trong quá trình huấn luyện. Cách này không cần phải tạo ra nhiều mạng neuron riêng biệt nhưng lại rất phức tạp. Rất ít các hệ thống thương mại cho phép việc thay đổi số đơn vị trong quá trình huấn luyện [10].

- Nhiều mô hình mạng neuron tầng vào-tầng ẩn-tầng ra đã được sử dụng hiệu quả trong bài toán dự báo chuỗi thời gian như: 8-8-1, 6-6-1, 5-5-1 [12]. Tuy nhiên, trong hiện thực của luận văn này, chúng tôi sử dụng mô hình mạng ứng với chu kỳ của chuỗi dữ liệu có tính mùa. Nghĩa là số nút ẩn sẽ bằng số nút đầu vào và bằng chu kỳ của chuỗi dữ liệu.

3.3.6. Xác định tiêu chuẩn đánh giá

Để đánh giá khả năng xấp xỉ một chuỗi thời gian của mạng neuron người ta thường dùng hàm *tổng bình phương lỗi* (sum of squared errors) sau:

$$SSE = \sum_{k=1}^n (t_k - o_k)^2 \quad (3.2)$$

Ở đây n là số điểm trong tập dữ liệu dùng để kiểm tra mạng, t_k và o_k lần lượt là giá trị mong muốn trong bộ dữ liệu và giá trị xuất của mạng neuron. Mạng neuron có tổng bình phương lỗi càng nhỏ càng tốt.

Ngoài ra người ta còn dùng các hàm khác là hàm *độ lệch tuyệt đối nhỏ nhất* (least absolute deviation), *hiệu phần trăm* (percentage differences).

3.3.7. Huấn luyện mạng

- Huấn luyện mạng để học các mẫu từ dữ liệu bằng cách lần lượt đưa các mẫu vào cùng với những giá trị mong muốn. Mục tiêu của việc huấn luyện mạng đó là

tìm ra tập các trọng số cho ta giá trị nhỏ nhất toàn cục của chỉ số hiệu năng hay hàm lỗi.

- Một vấn đề quan trọng trong quá trình huấn luyện mạng neuron là xác định điều kiện dừng của quá trình huấn luyện. Có ba cách thường dùng để dừng một quá trình huấn luyện. Cách thứ nhất nhấn mạnh vào việc tránh bị rơi vào điểm tối ưu cục bộ, nhà dự báo chỉ dừng quá trình học khi không có một sự cải thiện đáng kể nào của hàm lỗi. Điểm mà mạng neuron không còn cải thiện được nữa gọi là điểm hội tụ. Cách thứ hai là sử dụng một thông số cố định là số lần lặp tối đa, quá trình huấn luyện sẽ dừng nếu số số lần lặp (epochs) vượt quá thông số này. Mạng neuron sẽ được kiểm tra, nếu kết quả không tốt thì quá trình học sẽ được tiếp tục lại. Cách thứ ba là ta sử dụng một tập dữ liệu ngoài dữ liệu huấn luyện gọi là *tập dữ liệu kiểm định* (validation set). Trong quá trình huấn luyện, cứ mỗi lần vector trọng số của mạng neuron thay đổi, tập dữ liệu xác thực này sẽ được đưa vào mạng và tính ra sai số. Giải thuật huấn luyện sẽ dừng khi sai số này nhỏ hơn một ngưỡng mà nhà dự đoán mong muốn. Phương pháp này có khả năng tránh được quá khớp tuy nhiên không phải lúc nào cũng hội tụ do mạng neuron có thể không đáp ứng được ngưỡng sai số mong muốn của người dùng.

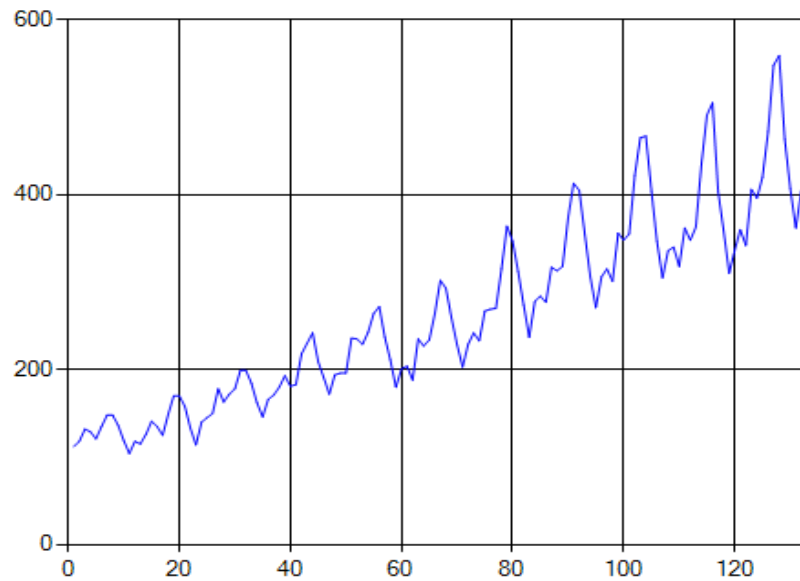
3.3.8. Dự đoán và cải tiến

Sau khi đã thực hiện các bước trên, ta có được một mô hình mạng neuron dùng để dự đoán. Các giá trị dự đoán của mạng được lưu lại và so sánh với các giá trị thực tế khi chúng xuất hiện. Sau một thời gian, có thể mô hình mạng không còn đúng nữa thể hiện qua việc kết quả dự đoán ngày càng xa các giá trị thật, ta cần phải tiến hành cải tiến mạng hoặc học lại và xây dựng mạng mới theo các bước trước.

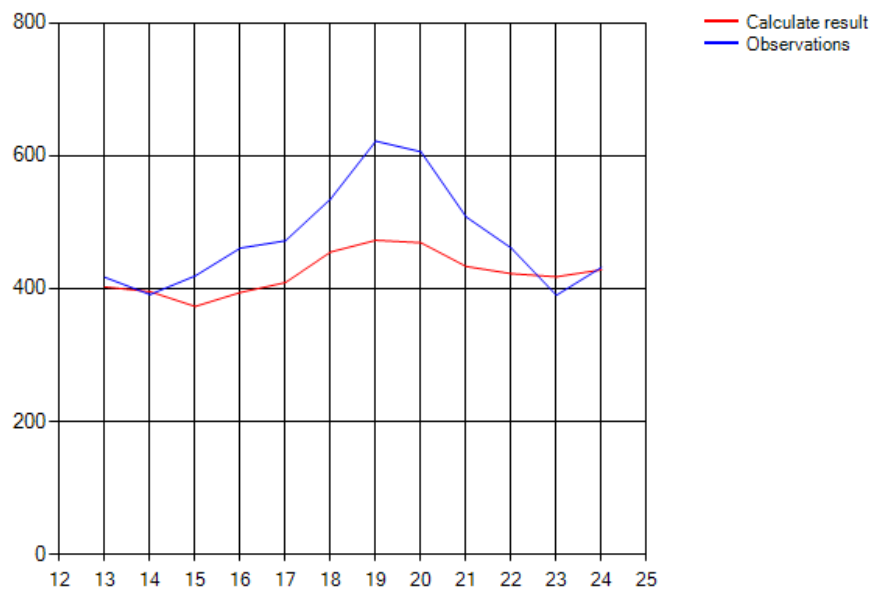
3.4. Hạn chế của việc sử dụng mạng Neuron trong việc dự báo đối với dữ liệu chuỗi thời gian có tính xu hướng và tính mùa

Tuy rằng mạng neuron nhân tạo có khả năng xấp xỉ tốt các hàm phi tuyến và có thể dự báo cho các chuỗi thời gian mà không cần nhiều kiến thức về các tính chất của chuỗi

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa cần dự báo nhưng nhiều kết quả nghiên cứu thực nghiệm đặc biệt là của Zhang [4] [1] [2] đã chứng minh rằng mạng neuron nhân tạo thuần túy không có khả năng dự đoán tốt được cho chuỗi thời gian có tính xu hướng và tính mùa.



(a)



(b)

Hình 3.7: (a) Chuỗi thời gian có tính mùa và xu hướng. (b) Kết quả dự đoán của mạng neuron nhân tạo sinh ra cho chuỗi thời gian trên hình a, đường màu đỏ là kết quả do mạng sinh ra, đường màu xanh là đường của dữ liệu quan sát được.

Hình 3.7a là đồ thị của một chuỗi thời gian quan sát theo tháng có tính mùa và tính xu hướng, hình 3.7b là kết quả dự báo do mạng neuron nhân tạo ba lớp có 12 đơn vị đầu vào, 12 đơn vị tầng ẩn và 1 đơn vị đầu ra huấn luyện bằng giải thuật RPROP sinh ra cho chuỗi này. Ta thấy rõ ràng độ chính xác dự báo của mạng neuron sinh ra không chính xác, do đó cần có sự cải tiến mạng neuron nhân tạo để tăng khả năng dự báo cho những chuỗi thời gian có tính xu hướng và tính mùa. Trong luận văn này chúng tôi sử dụng hai mô hình là mạng neuron nhân tạo kết hợp với khử mùa và làm trơn theo hàm mũ kết hợp với mạng neuron nhân tạo.

Chương 4

MÔ HÌNH KHỬ MÙA, KHỬ XU HƯỚNG KẾT HỢP MẠNG NEURON NHÂN TẠO

Mạng neuron nhân tạo thuần không có khả năng đưa ra dự báo tốt cho dữ liệu chuỗi thời gian có tính xu hướng và tính mùa. Để cải thiện điều này, ta có thể tìm cách loại bỏ thành phần mùa và xu hướng trong dữ liệu rồi thực hiện huấn luyện và dự báo đối với chuỗi dữ liệu mới sau khi loại bỏ hai thành phần trên. Trong chương này, chúng tôi sẽ giới thiệu về kỹ thuật để khử tính mùa, tính xu hướng ra khỏi dữ liệu chuỗi thời gian và một mô hình kết hợp mạng neuron nhân tạo với kỹ thuật này.

4.1. Kỹ thuật khử tính xu hướng cho dữ liệu chuỗi thời gian

Các chuỗi thời gian trong thực tế thường có thành phần xu hướng, thành phần này thể hiện sự tăng hay giảm giá trị của chuỗi thời gian trong một thời đoạn dài hạn. Có nhiều kỹ thuật để khử đi tính xu hướng này của chuỗi thời gian, trong đó có hai kỹ thuật thường được sử dụng do tính đơn giản và hiệu quả của chúng là *lấy hiệu* (differencing) và *khử xu hướng tuyến tính* (linear detrend).

4.1.1. Kỹ thuật khử xu hướng tuyến tính

Trong kỹ thuật khử xu hướng tuyến tính, ta xấp xỉ chuỗi thời gian bằng một đường thẳng hồi quy $at + b$ với t là biến thời gian. Các hệ số a và b được ước lượng dựa trên việc tối thiểu hàm sai số sau:

$$E(a, b) = \sum_{t=1}^n (Y_t - (at + b))^2$$

Trong đó n là số giá trị quan sát của chuỗi thời gian, Y_t là giá trị của chuỗi thời gian tại thời điểm t .

Trong giải tích toán học, người ta đã chứng minh được rằng để hàm số $E(a, b)$ đạt giá trị cực tiểu thì a và b phải là nghiệm của hệ sau:

$$\begin{cases} \frac{\partial}{\partial a} \sum_{t=1}^n (at+b-Y_t)^2 = 2 \sum_{t=1}^n (at+b-Y_t)t = 2(\sum_{t=1}^n t)b + 2(\sum_{t=1}^n t^2)a - 2 \sum_{t=1}^n tY_t = 0 \\ \frac{\partial}{\partial b} \sum_{t=1}^n (at+b-Y_t)^2 = 2 \sum_{t=1}^n (at+b-Y_t) = 2nb + 2(\sum_{t=1}^n t)a - 2 \sum_{t=1}^n Y_t = 0 \end{cases} \quad (4.1)$$

Từ đó ta tính được a, b

$$\begin{cases} a = \frac{2(2n+1)}{n(n-1)} \sum_{t=1}^n Y_t - \frac{6}{n(n-1)} \sum_{t=1}^n tY_t \\ b = \frac{12}{n(n^2-1)} \sum_{t=1}^n tY_t - \frac{6}{n(n-1)} \sum_{t=1}^n Y_t \end{cases} \quad (4.2)$$

Sau khi đã có đường thẳng $at+b$, ta khử xu hướng bằng cách ứng với mỗi t , lấy Y_t trừ đi $at+b$. Khi đó ta sẽ được chuỗi mới là chuỗi Y'_t là chuỗi thời gian đã khử xu hướng. Để phục hồi một chuỗi đã khử xu hướng bằng phương pháp này về chuỗi ban đầu chỉ cần lấy giá trị của nó tại mỗi thời điểm t cộng với $at+b$.

4.1.2. Kỹ thuật khử xu hướng bằng lấy hiệu

Một phương pháp khác để khử đi tính xu hướng của chuỗi thời gian là phương pháp lấy hiệu. Với chuỗi thời gian $\{Y_t\}$ có tính xu hướng, đặt $X_t = Y_{t+1} - Y_t$ thì chuỗi thời gian $\{X_t\}$ sinh ra là một chuỗi không có tính xu hướng.

Để phục hồi một chuỗi thời gian $\{X_t\}$ đã được khử tính xu hướng theo cách này về chuỗi ban đầu ta thực hiện $Y_{t+1} = X_t + Y_t$. Nếu ta biết được Y_1 thì ta sẽ dễ dàng phục hồi lại toàn chuỗi $\{Y_t\}$. Hình 4.1 là đồ thị minh họa cho dữ liệu chuỗi thời gian trước và sau khi khử xu hướng bằng phương pháp tuyến tính và lấy hiệu.

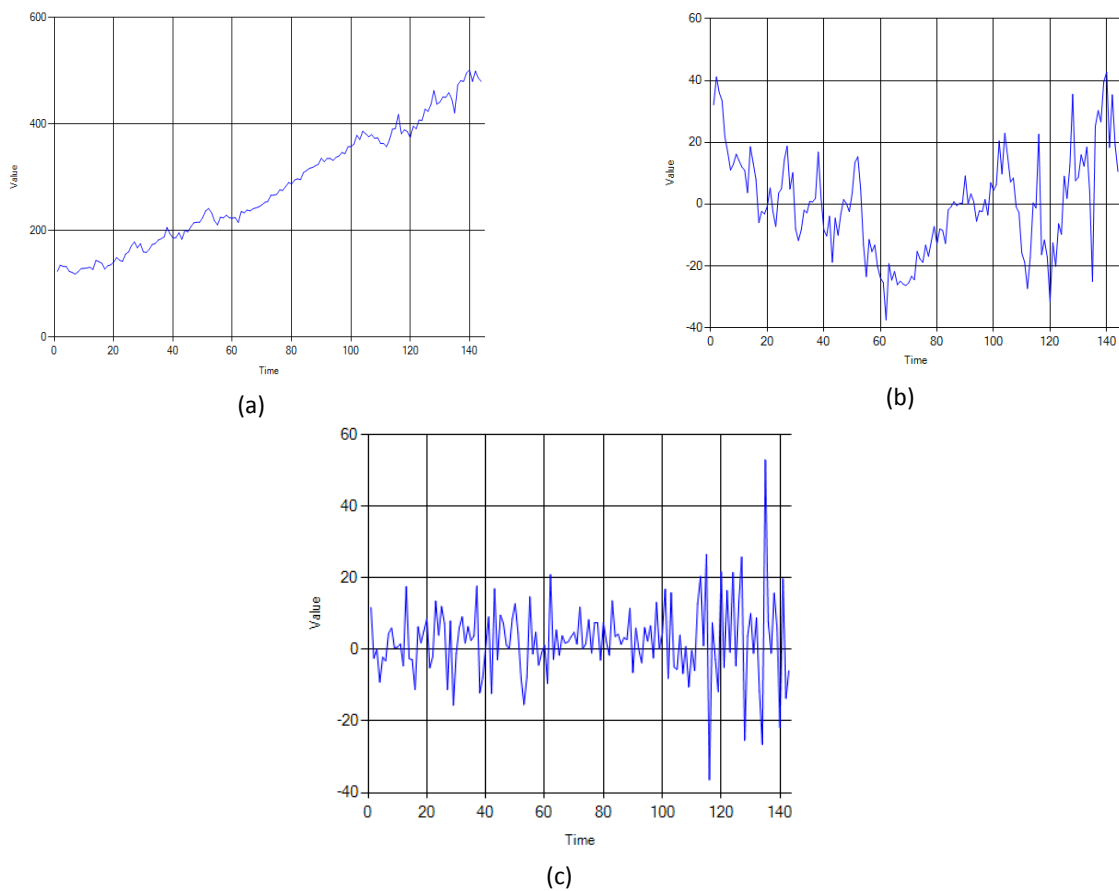
4.2. Kỹ thuật khử tính mùa cho dữ liệu chuỗi thời gian

Trong thực tế có rất nhiều chuỗi thời gian có tính mùa ví dụ như hàng năm doanh số bán hàng của một cửa hàng thực phẩm tăng cao trong các tháng 11, 12. Mạng neuron không có khả năng dự đoán tốt cho chuỗi thời gian có tính mùa nên cần phải khử mùa cho chuỗi thời gian trước khi sử dụng mạng neuron để dự báo. Có nhiều phương pháp

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa
khử mùa cho chuỗi thời gian, ở đây chúng tôi chọn hai phương pháp là *lấy hiệu theo mùa* (seasonal differencing) và *RTMA* (ratio to moving average) do tính hiệu quả và đơn giản cho việc phục hồi lại chuỗi ban đầu từ chuỗi đã khử mùa.

4.2.1. Kỹ thuật lấy hiệu theo mùa

Kỹ thuật này được sử dụng trong phương pháp luận Box-Jenkins và được đề cập trong [5]. Kỹ thuật này thực hiện việc biến đổi chuỗi thời gian $\{Y_t\}$ thành chuỗi $\{X_t\}$ như sau $X_t = Y_{t+s} - Y_t$, với s là độ lớn một chu kỳ của chuỗi thời gian. Nếu chuỗi thời gian được quan sát theo đơn vị là tháng thì $s = 12$, nếu chuỗi quan sát theo quý thì $s = 4$.



Hình 4.1: (a) Chuỗi thời gian ban đầu. (b) Chuỗi thời gian sau khi khử xu hướng tuyến tính. (c) Chuỗi thời gian sau khi lấy hiệu.

4.2.1. Kỹ thuật khử mùa bằng RTMA

Kỹ thuật này hay được đề cập trong [5]. Ý tưởng của giải thuật này là ta sẽ ước lượng *chỉ số mùa* (seasonal index) của các thời đoạn trong một chu kỳ của chuỗi thời gian, nghĩa là nếu chuỗi thời gian có tính mùa theo năm và giá trị được quan sát theo tháng thì ta sẽ tính chỉ số mùa cho từng tháng, nếu giá trị được quan sát theo quý thì ta sẽ tính chỉ số mùa cho từng quý. Để tính chỉ số mùa ta cần phải ước lượng và loại bỏ thành phần xu hướng và thành phần mùa của chuỗi thời gian bằng phương pháp *trung bình di động trung tâm* (centered moving average).

Giả sử ta có một chuỗi thời gian quan sát theo tháng thì chỉ số mùa theo tháng sẽ được tính như sau: Bắt đầu từ tháng 7 của năm đầu tiên ta tính tổng di động 12 tháng cho từng tháng. Với mỗi giá trị trong chuỗi tổng di động vừa tính ta cộng nó với giá trị kế sau và chia cho 24 để được chuỗi các giá trị trung bình di động trung tâm cho các thời đoạn. Chia giá trị của các thời đoạn trong chuỗi thời gian với giá trị trung bình di động trung tâm tương ứng ta được một chuỗi các giá trị ước lượng chỉ số mùa cho các tháng. Vì ứng với mỗi tháng có nhiều giá trị ước lượng như vậy nên giá trị trung vị của các giá trị ước lượng sẽ là chỉ số tạm thời mùa cho tháng. Lấy mỗi chỉ số mùa tạm thời này nhân với 12 và chia cho tổng chỉ số mùa tạm thời của 12 tháng ta sẽ có các chỉ số mùa thật sự của 12 tháng trong năm.

Để khử tính mùa trong chuỗi thời gian ta chỉ việc lấy giá trị của mỗi thời đoạn chia cho chỉ số mùa tương ứng của nó. Hình 4.2 là đồ thị minh họa cho dữ liệu chuỗi thời gian trước và sau khi khử mùa.

4.3. Mô hình dự báo chuỗi thời gian bằng mạng neuron nhân tạo kết hợp với khử mùa và khử xu hướng

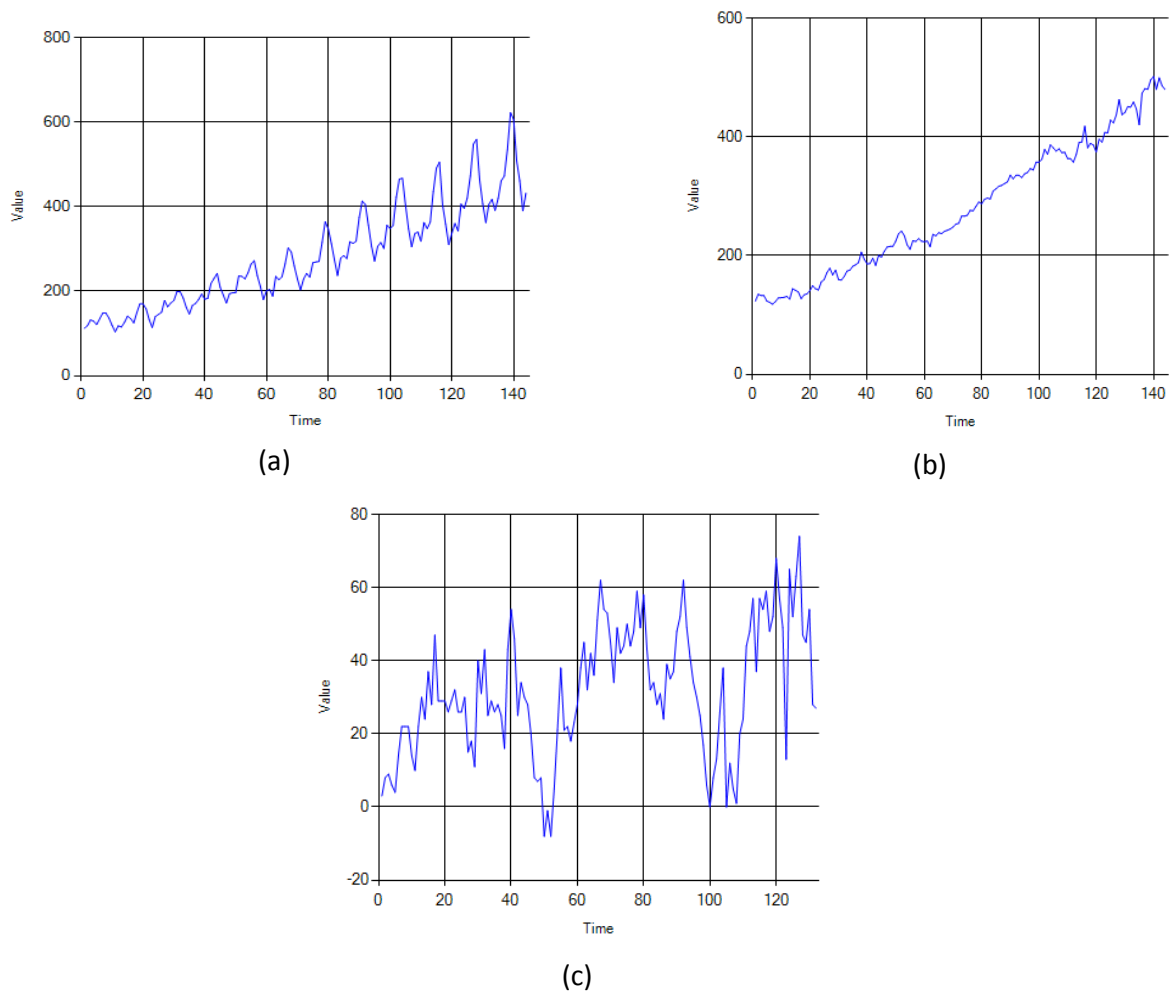
4.3.1. Ý tưởng

Mô hình mạng neuron nhân tạo kết hợp với khử mùa và khử xu hướng được xây dựng dựa theo ý tưởng thay vì cho mạng neuron học trên trực tiếp trên bộ dữ liệu có tính mùa và tính xu hướng, ta sẽ tiến hành khử mùa và khử xu hướng trước, sau đó neuron sẽ học trên tập dữ liệu mới này. Trong bước dự báo, các giá trị dự báo do mạng neuron

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa sinh ra sẽ được thêm vào tính mùa và tính xu hướng đã khử trước đó để cho ra kết quả dự báo sau cùng.

4.3.2. Hiện thực.

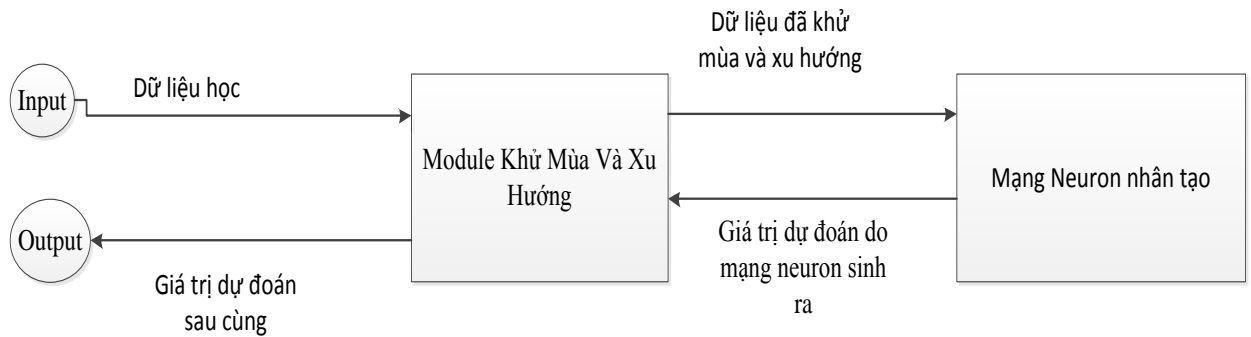
Chúng tôi hiện thực mô hình trên bằng ngôn ngữ lập trình C#, trên nền .NET framework 4.0 với bộ hỗ trợ lập trình Visual C# 2010 express. Chương trình gồm có các module chính được minh họa tại hình 4.3.



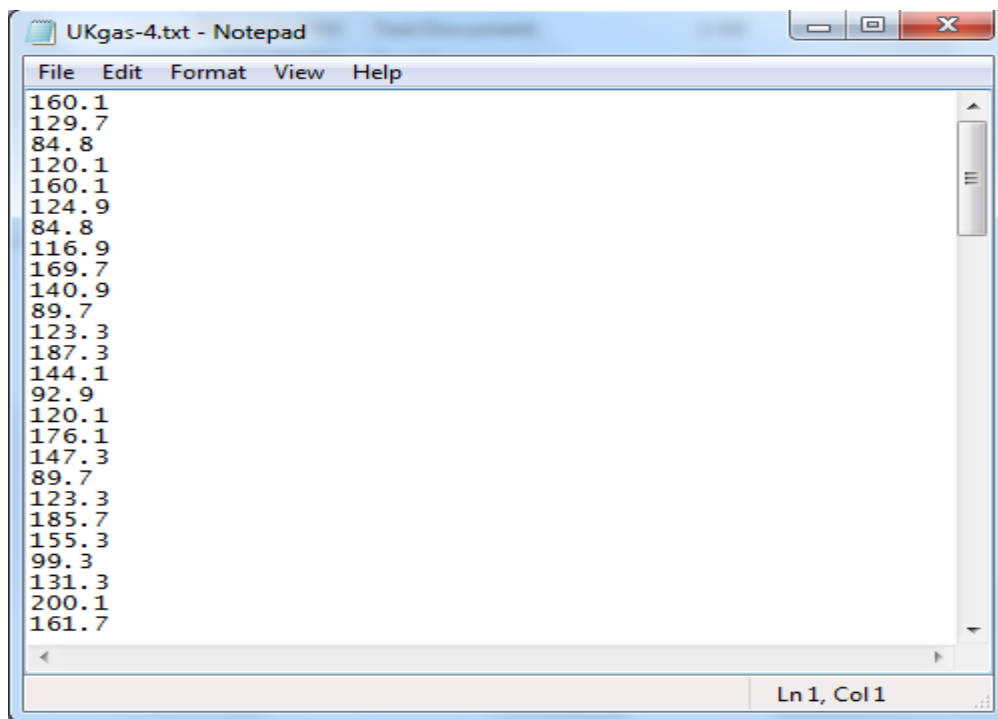
Hình 4.2: (a) Chuỗi thời gian ban đầu. (b) Chuỗi thời gian sau khi khử mùa bằng RTMA. (c) Chuỗi thời gian sau khi lấy hiệu theo mùa.

Module đọc và biểu diễn dữ liệu: Module này thực hiện đọc dữ liệu từ file với định dạng như hình dưới và biểu diễn dữ liệu cho người dùng bằng đồ thị trực quan. Module này còn có chức năng tính và biểu diễn hàm tự tương quan của chuỗi thời gian

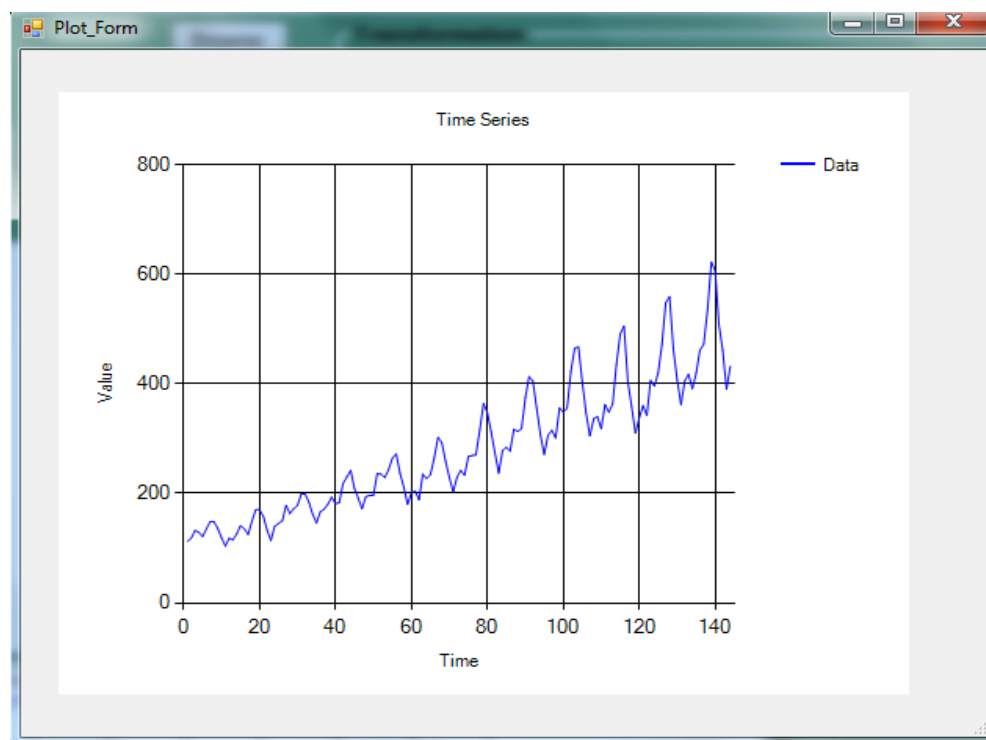
Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa bằng đồ thị để hỗ trợ người dùng xác định các thành phần trong chuỗi thời gian. File chứa dữ liệu có định dạng gồm một hoặc nhiều cột các số, nếu file có nhiều cột số thì các số ở các cột khác nhau trong cùng một hàng phân cách nhau bởi dấu “,” hoặc ký tự trống. Hình 4.4, 4.5, 4.6 lần lượt minh họa định dạng file dữ liệu, việc vẽ đồ thị và đồ thị tự tương quan đối với một chuỗi dữ liệu.



Hình 4.3: Mô hình mạng neuron nhân tạo kết hợp với khử mùa và xu hướng.



Hình 4.4: Định dạng của file dữ liệu.

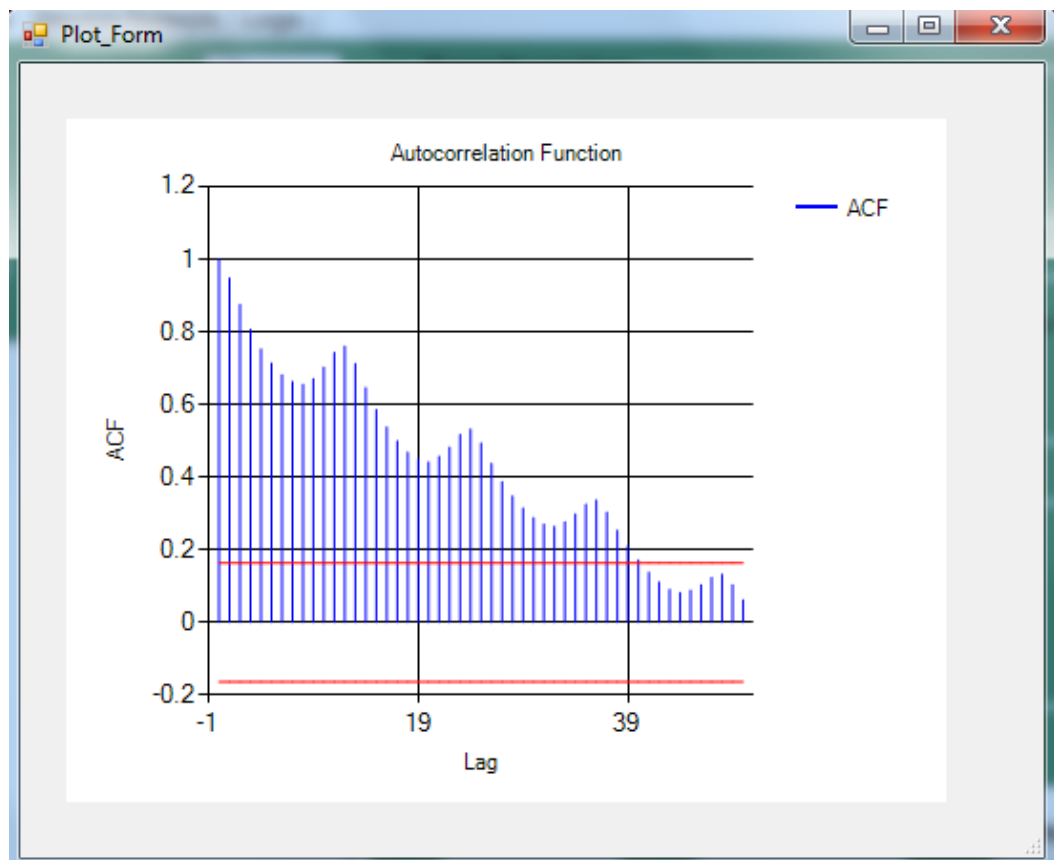


Hình 4.5: Đồ thị của chuỗi thời gian được biểu diễn bởi module đọc và biểu diễn dữ liệu.

Module khử mùa và khử xu hướng: Module này có nhiệm vụ khử tính mùa và tính xu hướng khỏi một chuỗi thời gian để đưa vào mạng neuron đồng thời thêm thành phần mùa và xu hướng vào chuỗi dự báo của mạng neuron để đưa ra giá trị dự báo sau cùng. Chúng tôi hiện thực hết 4 giải thuật khử mùa và khử xu hướng đã nêu ở trên trong module này và cho phép người dùng chọn phương pháp phù hợp với dữ liệu của mình.

Module mạng neuron: là module thực hiện chức năng của mạng neuron trong mô hình là học và sinh ra dự báo. Mô hình mạng được hiện thực là mô hình mạng truyền thẳng gồm ba lớp với một đơn vị đầu ra. Số lượng đơn vị ở tầng xuất và tầng ẩn do người dùng tùy chọn. Các giá trị đầu vào của mạng là các giá trị chuỗi thời gian ở các thời đoạn trước. Hàm kích hoạt được sử dụng ở tầng ẩn và tầng xuất là hàm Sigmoid, hàm kích hoạt ở tầng nhập là hàm đồng nhất. Ở đây chúng tôi hiện thực cả hai giải thuật huấn luyện mạng là lan truyền ngược và RPROP. Chương trình cho phép người dùng lựa chọn giải thuật để huấn luyện mạng. Điều kiện dừng của quá trình học được

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa xác định bởi hai thông số: số epoch tối đa và độ lệch sai số tối thiểu giữa hai epoch. Số epoch tối đa là một số do người dùng chọn, khi số epoch của quá trình học lớn hơn số epoch tối đa thì chương trình sẽ dừng. Độ lệch sai số tối thiểu là một số nhỏ do người dùng chọn, khi độ chênh lệch của hàm sai số trong hai epoch học kế tiếp nhỏ hơn giá trị này thì điều đó có nghĩa là việc học tiếp không cải thiện được hàm sai số một cách đáng kể, quá trình học sẽ dừng. Việc chọn điều kiện dừng như thế sẽ giúp người dùng chủ động trong việc kiểm soát quá trình học và tránh hiện tượng quá khớp. Ban đầu người dùng nên chọn số epoch tối đa nhỏ và độ lệch sai số tối thiểu lớn để quá trình học dừng sau một số ít epoch. Mạng vừa học xong sẽ được kiểm tra, nếu sai số dự báo quá lớn thì người dùng tăng số epoch tối đa và giảm độ lệch sai số tối thiểu rồi học tiếp.



Hình 4.6: Hàm tự tương quan của chuỗi thời gian được biểu diễn bởi module đọc và biểu diễn dữ liệu.

Module scaling: Module này biến đổi dữ liệu đầu vào về một giá trị trong khoảng (0, 1) và chuyển giá trị xuất của mạng neuron (một giá trị nằm trong khoảng (0, 1)) về giá trị thực. Module này được hiện thực theo hai công thức sau:

$$scaleValue = (value - minValue) / (maxValue - minValue) * (0.9999 - (0.0001)) + (0.0001) \quad (4.3)$$

$$value = (Scalevalue - 0.0001) / (0.9999 - 0.0001) * (maxValue - minValue) + minValue \quad (4.4)$$

Trong đó: scaleValue là giá trị biến đổi nằm trong khoảng (0,1), value là giá trị nằm trong khoảng xác định (*minValue*, *maxValue*). Các thông số *minValue* và *maxValue* là cận dưới và cận trên của các giá trị trong chuỗi thời gian. Hai thông số này được chọn bởi người dùng tùy theo từng chuỗi thời gian.

Chương 5

MÔ HÌNH LAI GIỮA MẠNG NEURON VÀ KỸ THUẬT LÀM TRƠN THEO HÀM MŨ

Mạng neuron nhân tạo thuần không có khả năng đưa ra dự báo tốt cho dữ liệu chuỗi thời gian có tính xu hướng và tính mùa. Ngoài cách tiếp cận như chương 4 đã đề cập, một cách khác là sử dụng kết hợp mô hình làm trơn theo hàm mũ và mạng neuron nhân tạo. Trong chương này, chúng tôi sẽ đề cập đến các kỹ thuật làm trơn theo hàm mũ cũng như cách kết hợp với mạng neuron nhân tạo để tạo thành mô hình lai.

5.1. Kỹ thuật làm trơn theo hàm mũ

Các phương pháp làm trơn là tập hợp các phương pháp, trong đó các giá trị dự báo mới ở một thời đoạn sẽ được ước lượng bằng việc kết hợp giá trị dự báo và giá trị quan sát ở thời đoạn ngay trước đó. Đây là một trong những kỹ thuật đơn giản nhưng hiệu quả để sử dụng dự báo đối với dữ liệu chuỗi thời gian có tính chất tuyến tính. Gồm có các hiện thực sau:

5.1.1. Kỹ thuật làm trơn theo hàm mũ đơn giản (Simple Exponential Smoothing)

Trong kỹ thuật này, giá trị ước lượng mới sẽ được tính theo công thức:

$$\hat{Y}_{t+1} = \alpha Y_t + (1 - \alpha) \hat{Y}_t \quad (5.1)$$

Với α được gọi là hệ số làm trơn ($0 < \alpha < 1$).

Đây là hiện thực giản đơn nhất trong các phương pháp làm trơn dữ liệu. Từ công thức (5.1) ta có thể biến đổi thành:

$$\begin{aligned} \hat{Y}_{t+1} &= \alpha Y_t + \alpha(1 - \alpha)Y_{t-1} + \alpha(1 - \alpha)^2 Y_{t-2} + \alpha(1 - \alpha)^3 Y_{t-3} + \dots \\ \rightarrow \hat{Y}_{t+1} &= \alpha[Y_t + (1 - \alpha)Y_{t-1} + (1 - \alpha)^2 Y_{t-2} + (1 - \alpha)^3 Y_{t-3} + \dots] \end{aligned}$$

Vì thế kỹ thuật này mới được gọi là kỹ thuật làm trơn theo hàm mũ. Đối với giá trị α ta hoàn toàn có thể ước lượng bằng cách sử dụng các phương pháp ước lượng tối thiểu các hàm sai số SSE (sum squared error), MSE (mean squared error), MAE (mean absolute error)....

5.1.2. Kỹ thuật làm trơn theo hàm mũ Holt

Đối với dữ liệu chuỗi thời gian có tính xu hướng rõ rệt, việc sử dụng kỹ thuật làm trơn giản đơn không còn hiệu quả. Vì thế, một kỹ thuật cải tiến đã được sử dụng đó là kỹ thuật làm trơn Holt. Trong kỹ thuật này, dữ liệu chuỗi thời gian được tách thành hai thành phần là thành phần *bậc* (level) và thành phần *xu hướng* (trend), khi đó giá trị Y tại thời điểm $(t+p)$ sẽ được ước lượng bởi

$$\hat{Y}_{t+p} = L_t + pT_t$$

Trong đó, L_t và T_t là hai thành phần bậc và xu hướng được ước lượng như sau:

$$L_t = \alpha Y_t + (1 - \alpha)(L_{t-1} + T_{t-1})$$

$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$$

Đây chính là một sự mở rộng từ kỹ thuật làm trơn giản đơn nếu cho thành phần xu hướng trong dữ liệu bằng 0.

5.1.3. Kỹ thuật làm trơn theo hàm mũ Winters (mô hình Holt-Winters)

Đặc biệt, đối với dữ liệu chuỗi thời gian có tính mùa, một phương pháp được sử dụng là kỹ thuật làm trơn Winters. Kỹ thuật này có hai mô hình hiện thực: mô hình có tính nhân và mô hình có tính cộng. Công thức cho từng mô hình này như sau:

Mô hình nhân: thành phần mùa được xem là một thành phần nhân trong dữ liệu

$$Y_{n+k} = (L_n + k.T_n).S_{n+k-s}$$

$$L_t = \alpha \frac{Y_t}{S_{t-s}} + (1 - \alpha)(L_{t-1} + T_{t-1})$$

$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$$

$$S_t = \gamma \frac{Y_t}{L_t} + (1 - \gamma)S_{t-s}$$

Mô hình cộng: thành phần mùa được xem là một thành phần cộng trong dữ liệu

$$Y_{n+k} = (L_n + k.T_n) + S_{n+k-s}$$

$$L_t = \alpha(Y_t - S_{t-s}) + (1 - \alpha)(L_{t-1} + T_{t-1})$$

$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$$

$$S_t = \gamma(Y_t - L_t) + (1 - \gamma)S_{t-s}$$

Trong đó:

L, T, S lần lượt là các thành phần bậc, xu hướng, mùa của chuỗi thời gian.

α, β, γ là các hệ số làm trơn ($0 \leq \alpha, \beta, \gamma \leq 1$) tương ứng với các thành phần L, T, S

Nếu hệ số γ trên bằng 0, nghĩa là thành phần mùa không tồn tại trong dữ liệu, kĩ thuật này trở thành kĩ thuật làm trơn Holt. Nếu hệ số β cũng bằng 0, mô hình này trở thành mô hình làm trơn theo hàm mũ giản đơn. Như vậy, đây chính là kĩ thuật tổng quát nhất trong 3 kĩ thuật (giải quyết trọn vẹn cho ba kiểu dữ liệu).

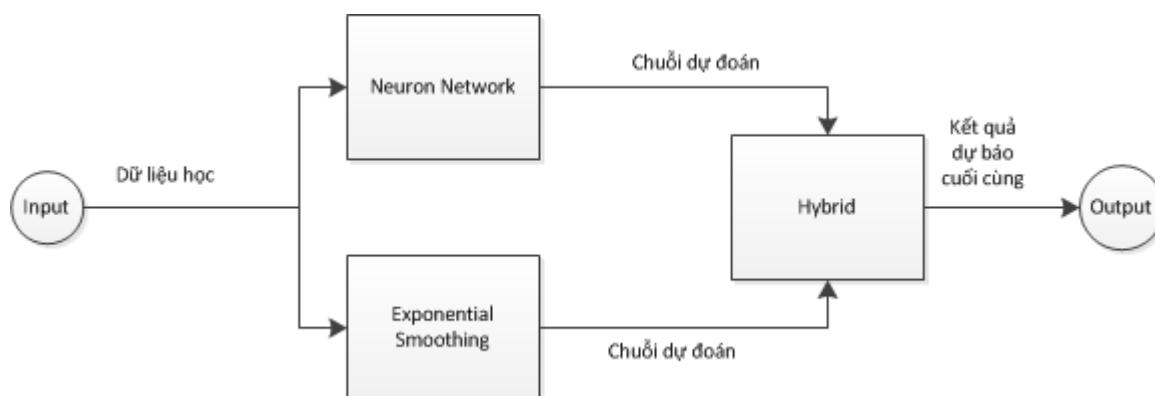
5.2. Mô hình lai giữa mạng neuron nhân tạo và kỹ thuật làm trơn theo hàm mũ

5.2.1. Ý tưởng

Dữ liệu chuỗi thời gian, đặc biệt là dữ liệu chuỗi thời gian có tính mùa và xu hướng rõ rệt thì kĩ thuật làm trơn theo hàm mũ xử lí khá tốt và cho kết quả với độ sai số nhỏ. Tuy nhiên trong trường hợp thành phần ngoại lai của chuỗi dữ liệu chiếm nhiều thì mô hình neuron network lại xử lí tốt hơn. Vì thế, để cải tiến hiệu suất của dự báo, một mô hình mới, *mô hình lai* (hybrid) giữa mạng neuron nhân tạo và kĩ thuật làm trơn theo hàm mũ được đưa ra bởi Lai và các cộng sự. [3]

Cấu trúc mô hình lai được mô tả ở hình 5.1.

Mô hình lai gồm ba phần tách biệt: mạng neuron, thành phần làm trơn theo hàm mũ và thành phần lai. Trong đó, hai thành phần mạng neuron và làm trơn theo hàm mũ là hai thành phần hoàn toàn độc lập nhau.



Hình 5.1: Cấu trúc mô hình lai

Mỗi thành phần sẽ được huấn luyện trên cùng một tập dữ liệu (tìm các trọng số giữa các đơn vị và độ lệch tương ứng của chúng đối với mô hình mạng neuron nhân tạo và tìm ba hệ số α, β, γ đối với mô hình làm trơn theo hàm mũ). Sau đó, cả hai mô hình này sẽ truyền toàn bộ các kết quả dự báo của mình cho mô hình lai, tại đây sẽ cho ra kết quả cuối cùng bằng cách kết hợp hai kết quả từ mạng neuron và làm trơn theo hàm mũ theo công thức:

$$Y_{Hybrid} = \omega Y_{NN} + (1 - \omega) Y_{ES}$$

Trong đó, ω được gọi là trọng số lai ($0 \leq \omega \leq 1$). Nếu ω bằng 0, nghĩa là kết quả dự đoán bằng kỹ thuật làm trơn theo hàm mũ là tốt nhất, nếu ω bằng 1, nghĩa là kết quả dự đoán bằng mô hình mạng neuron là tốt nhất.

5.2.2. Hiện thực

Chúng tôi hiện thực mô hình lai bằng ngôn ngữ lập trình C#, trên nền .NET framework 4.0 với bộ hỗ trợ lập trình Visual C# 2010 express. Chương trình gồm có các module chính sau:

Module mạng neuron: sử dụng lại Module mạng từ hiện thực phía trên (mô hình kết hợp giữa mạng neuron nhân tạo và khử mùa, khử xu hướng)

Module làm tròn theo hàm mũ: Đối với module này, công việc khó khăn nhất chính là việc xác định ba hệ số α , β và γ trong kĩ thuật làm tròn theo hàm mũ Winters. Qua quá trình tìm hiểu và nghiên cứu chúng tôi đã có một số giải pháp sau đây:

a. Phương pháp vét cạn (Brute force)

Ta biết rằng cả ba hệ số α , β và γ đều thuộc $[0;1]$. Do đó, nếu độ chính xác của ba hệ số trên khoảng 0.1 hoặc 0.01 thì tổng số trường hợp cho mỗi hệ số là 10 hoặc 10^2 . Như vậy, tổng số trường hợp nếu ta thực hiện vét cạn là 10^3 hoặc 10^6 trường hợp. Tại mỗi trường hợp, ta sẽ tính giá trị sai số MSE (mean of square errors) theo công thức sau:

$$MSE = \frac{1}{n} (\sum_{i=1}^n (\hat{Y}_i - Y_i)^2) \quad (5.2)$$

Ta sẽ tìm bộ ba hệ số ứng với giá trị MSE nhỏ nhất.

b. Phương pháp leo đồi (Hill climbing)

Gọi d là độ chính xác yêu cầu của ba hệ số α , β , γ . Như vậy, ứng với phương pháp vét cạn ta nên sử dụng ứng với $d = 0.1$ hoặc 0.01. Còn đối với các giá trị d nhỏ hơn (nghĩa là yêu cầu độ chính xác của ba hệ số cao hơn) thì phương pháp vét cạn sẽ không thể làm nổi (do tổng số trường hợp quá lớn). Vì thế, để giải quyết việc này, ta sẽ sử dụng một phương pháp khác là phương pháp leo đồi.

Về mặt cơ bản, hai phương pháp này đều xem một bộ ba hệ số là một trạng thái trong không gian trạng thái của bộ ba hệ số này. Đối với vét cạn, ta sẽ tính tất cả các giá trị MSE của không gian trạng thái. Còn đối với phương pháp leo đồi, ta sẽ làm như sau:

- Gọi T_0 là trạng thái hiện tại của bộ ba hệ số này. Tính giá trị MSE của T_0 theo công thức 5.2.
- Tiếp đây ta sẽ tìm trạng thái tốt hơn tiếp theo của T_0 . Gọi α_0 là giá trị của α ứng với trạng thái T_0 . Ở trạng thái tốt hơn tiếp theo, giá trị α có ba khả năng xảy ra: hoặc tăng hoặc giảm với khoảng cách là d , hoặc không thay đổi. Như vậy sẽ có ba trường hợp đối với giá trị α . Như vậy, ta có tổng cộng 27 ($3 \times 3 \times 3$) trạng thái kế tiếp của T_0 . Lúc này ta sẽ vét cạn trong 27 trạng thái này để tìm trạng thái tốt hơn rồi nhảy tới đó.

Có hai điều kiện dừng cho phương pháp này ứng với hai tên gọi:

- Tại trạng thái hiện tại, nếu không tìm được trạng thái tốt hơn tiếp theo, ta sẽ dừng lại quá trình tìm kiếm và kết quả trả về chính là trạng thái hiện tại. Với cách làm này, phương pháp leo đồi được gọi là *leo đồi dốc nhất* (Steepest-Ascent Hill Climbing).
- Trong trường hợp không tìm được trạng thái tốt hơn từ trạng thái hiện tại, thay vì kết thúc, ta sẽ sinh ngẫu nhiên một giá trị trong khoảng $[0;1]$. Nếu giá trị này nhỏ hơn 0.1, ta sẽ nhảy đến trạng thái mới (tức là đối với phương pháp này ta sẽ có 10% xác suất nhảy đến trạng thái mới dù rằng nó không tốt hơn trạng thái cũ). Nếu ngược lại, ta sẽ kết thúc quá trình tìm kiếm. Với cách làm này, phương pháp leo đồi được gọi là *tôi luyện mô phỏng* (Simulated Annealing). Ưu điểm của phương pháp này giúp ta giảm khả năng rơi vào điểm cực trị cục bộ của không gian trạng thái.

Đối với phương pháp này, một điều khá quan trọng chính là việc chọn trạng thái bắt đầu của bộ ba hệ số. Việc này ảnh hưởng nhiều tới khả năng quá trình tìm kiếm rơi vào điểm cực trị cục bộ hay không. Do vậy, trong quá trình chạy thực nghiệm, chúng tôi sẽ kết hợp với giữa vét cạn và tìm kiếm leo đồi như sau:

- Thực hiện vét cạn với $d = 0.1$. Ta sẽ tìm được trạng thái tốt nhất ứng với độ chính xác này.
- Tìm kiếm với độ chính xác cao hơn (với d nhỏ hơn) sử dụng phương pháp leo đồi (leo đồi dốc nhất hoặc tôi luyện mô phỏng) với trạng thái bắt đầu là trạng thái vừa tìm được từ vét cạn.

Với phương thức tiếp cận này, chúng ta có thể tìm được bộ ba hệ số này với độ chính xác cao và thời gian ngắn.

c. Sử dụng phần mềm R

Như chúng ta đã biết phần mềm R là một trong những bộ phần mềm lớn nhất hiện thực hầu hết các lý thuyết mô hình trong lĩnh vực thống kê bao gồm cả làm trơn theo hàm mũ. Trong giới hạn của luận văn này, chúng tôi chỉ đề cập tới phương thức sử dụng và

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa cách lấy các tham số ước lượng α, β, γ trả về từ R cho mô hình làm trơn. Các thông tin chi tiết về phần mềm R có thể tìm trong [14].

Để ước lượng các thông số, chúng tôi sử dụng hàm HoltWinters được hiện thực sẵn trong R [16] và lấy ba tham số α, β, γ trả về từ hàm này.

Chương trình hiện thực sẽ gọi và lấy kết quả từ R thông qua phần mềm trung gian R(D)COM (hay còn gọi là statconnDCOM) được cài đặt và chạy ngoài trường trình. Thông tin về phần mềm này có thể tìm thấy ở [15].

Ngoài ra, việc ước lượng và tính toán sai số, các chuỗi bậc, xu hướng và mùa phụ thuộc khá nhiều vào việc khởi tạo giá trị ban đầu cho ba chuỗi này. Trong phạm vi hiện thực của luận văn này, chúng tôi sử dụng phương pháp sau:

Mô hình cộng:

$$L_s = \frac{1}{s} \left(\sum_{i=1}^s Y_i \right)$$

$$T_s = \frac{1}{s} \left(\frac{Y_{s+1} - Y_1}{s} + \frac{Y_{s+2} - Y_2}{s} + \dots + \frac{Y_{2s} - Y_s}{s} \right)$$

$$S_i = Y_i - L_s, i = 1, \dots, s$$

Mô hình nhân:

$$L_s = \frac{1}{s} \left(\sum_{i=1}^s Y_i \right)$$

$$T_s = \frac{1}{s} \left(\frac{Y_{s+1} - Y_1}{s} + \frac{Y_{s+2} - Y_2}{s} + \dots + \frac{Y_{2s} - Y_s}{s} \right)$$

$$S_i = \frac{Y_i}{L_s}, i = 1, \dots, s$$

Module lai (Hybrid): Đối với module này, công việc chủ yếu là xác định trọng số lai giữa mạng neuron nhân tạo và mô hình làm trơn theo hàm mũ. Giả sử chuỗi dùng để huấn luyện là Y_t , hai chuỗi dự báo từ Y_t qua hai mô hình neuron và làm trơn theo hàm mũ là N_t và S_t . Việc xác định hệ số α sẽ được tính từ việc tối thiểu giá trị

$$\text{MSE} = \sum (Y_t - [\omega N_t + (1 - \omega)S_t])^2 \quad (5.3)$$

Đây chính là một parabol có cực tiểu xác định nên ta hoàn toàn có thể tính chính xác giá trị ω tốt nhất theo công thức:

$$\omega = \frac{\sum (N_t - S_t)(Y_t - S_t)}{\sum (N_t - S_t)^2}. \quad (5.4)$$

Vì ω nhận giá trị trong đoạn $[0;1]$ nên nếu giá trị ω tính ra âm, chúng tôi chọn ω bằng 0, nếu giá trị ω tính ra lớn hơn 1, chúng tôi chọn ω bằng 1.

Chương 6

THỰC NGHIỆM VÀ ĐÁNH GIÁ

Để so sánh tính hiệu quả của hai mô hình đã trình bày ở chương 4 và chương 5 so với mạng neuron thuần, chúng tôi đã tiến hành chạy thực nghiệm trên các bộ dữ liệu thực tế khác nhau. Trong chương này, chúng tôi sẽ trình bày cách thức tiến hành thực nghiệm và kết quả thực nghiệm mà chúng tôi đã thu được.

6.1. Cách thức thực nghiệm

Để minh chứng cho tính hiệu quả của hai mô hình hiện thực ở trên trong việc dự báo cho chuỗi thời gian so với mạng neuron nhân tạo thuần túy và so sánh sự hiệu quả giữa hai mô hình, chúng tôi tiến hành chạy thực nghiệm trên năm bộ dữ liệu thực tế khác nhau. Ứng với mỗi bộ dữ liệu, chúng tôi tiến hành chạy với mạng neuron thông thường, chạy với mô hình mạng neuron kết hợp với làm trơn theo hàm mũ và chạy với mô hình mạng neuron kết hợp với khử mùa và khử xu hướng. Sai số dự báo của các mô hình sẽ được so sánh với nhau. Các sai số dự báo sẽ được quan sát là:

- Sai số tuyệt đối trung bình MAE (mean absolute error) tính theo công thức

$$MAE = \frac{1}{n} \sum_{t=1}^n | \hat{y}_t - y_t |$$

- Sai số trung bình bình phương MSE (mean squared error) tính theo công thức

$$MSE = \frac{1}{n} \sum_{t=1}^n (\hat{y}_t - y_t)^2$$

- Sai số tuyệt đối phần trăm $MAPE$ (mean absolute percentage error)

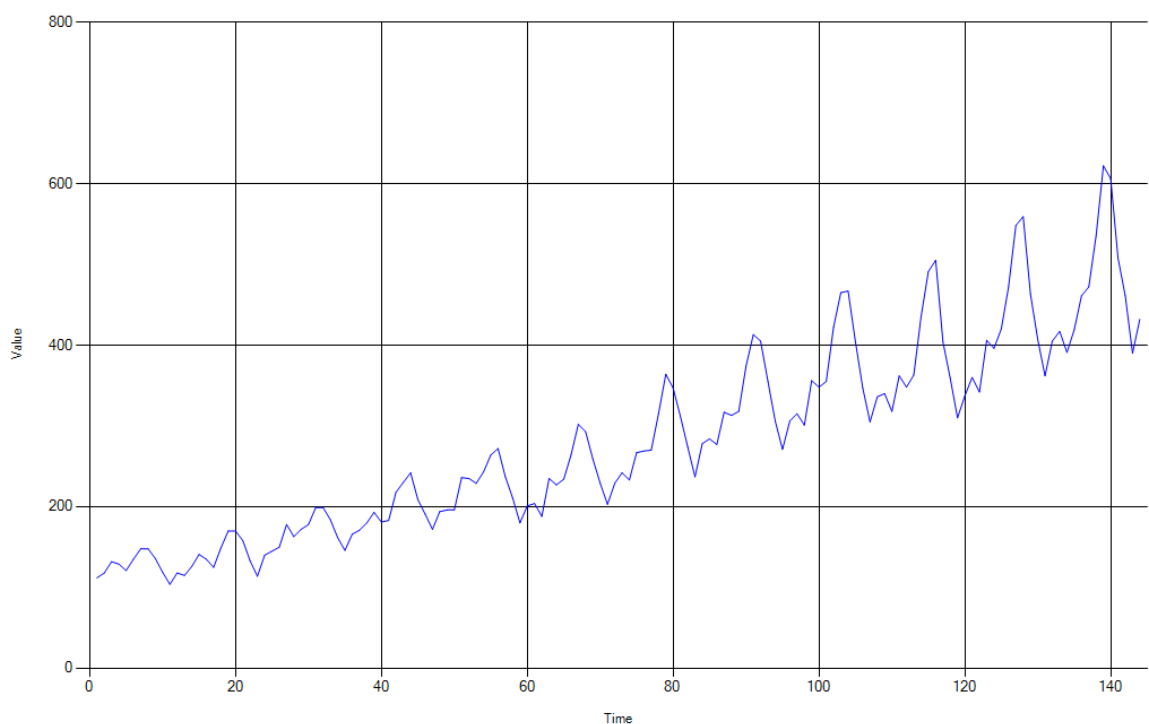
$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{| \hat{y}_t - y_t |}{y_t}$$

Ở đây n là chiều dài chuỗi dự đoán, y_t là giá trị quan sát được dùng để kiểm tra độ chính xác (tập giá trị này không nằm trong tập dùng để huấn luyện mô hình), \hat{y} là giá trị dự báo sinh ra bởi mô hình.

Ứng với mỗi mô hình, chúng tôi thực hiện chạy trên nhiều cấu hình khác nhau, mỗi cấu hình chạy 3 lần để đảm bảo tính khách quan, sai số dự báo của cấu hình là sai số dự báo trung bình của 3 lần chạy. Độ chính xác dự báo của mô hình khi đem so sánh với các mô hình khác sẽ là độ chính xác của cấu hình có sai số dự báo nhỏ nhất. Các thực nghiệm đều được chạy với máy tính có cấu hình: Ram 3GB, CPU Core 2 Duo 2.4 GHz, cài đặt hệ điều hành window 7 professional 32bits và .Net framework 4.0. Hai mô hình chúng tôi đều hiện thực trên ngôn ngữ lập trình C#. Hướng dẫn sử dụng chương trình, người đọc có thể tham khảo tại phụ lục B. Dưới đây là kết quả thực nghiệm chi tiết của các bộ dữ liệu.

6.2. Kết quả thực nghiệm và đánh giá

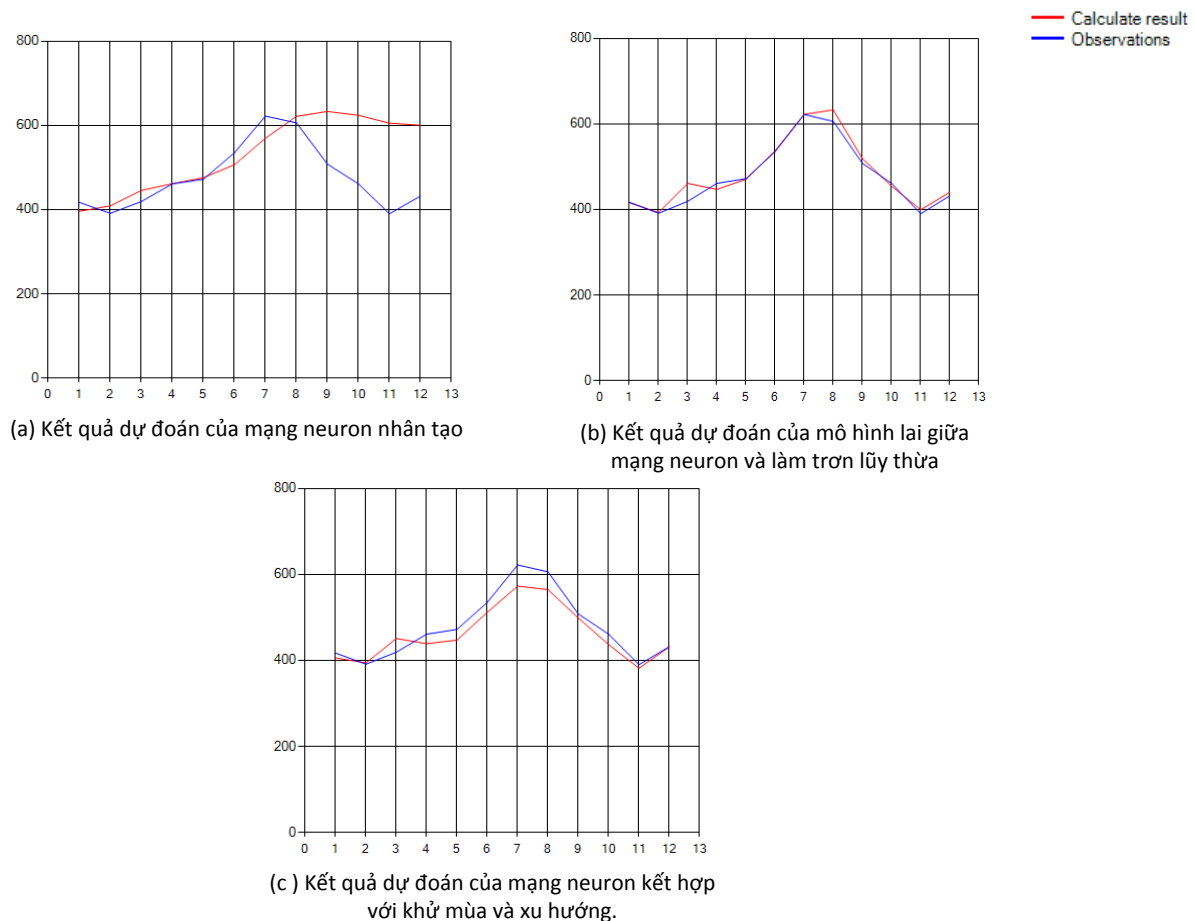
6.2.1. Bộ dữ liệu số lượng khách hàng đặt chỗ hàng tháng của hãng hàng không Pan Am từ năm 1946 đến năm 1960



Hình 6.1: Lượng khách hàng đặt chỗ hàng tháng của hãng hàng không Pan Am từ năm 1946 đến năm 1960

Bảng 6.1: Kết quả thực nghiệm cho chuỗi số lượng khách hàng đặt chỗ hàng tháng của hãng hàng không Pan Am từ năm 1946 đến năm 1960

Mô hình	Cấu hình	Thông số được ước lượng	Sai số dự báo
Mạng neuron nhân tạo(12-12-1)	Giải thuật: lan truyền ngược Số epoch tối đa: 1500. Độ lệch sai số tối thiểu: 0.001 Hệ số học: 0.5. Hệ số quán tính: 0.3.		MAE: 72.38 MSE: 10848.49 MAPE: 16.23%
Mô hình lai giữa mạng neuron nhân tạo(12-12-1) với làm trơn theo hàm mũ	Mô hình có tính nhân. Ước lượng α, β, γ bằng vét cạn và tối luyện mô phỏng	α : 0.30672 β : 0.03413 γ : 0.96873 ω : 0	MAE: 10.43 MSE: 250.64 MAPE: 2.25%
Mạng neuron nhân tạo(12-12-1) kết hợp với khử mùa và khử xu hướng	Giải thuật: lan truyền ngược Số epoch tối đa: 1500. Độ lệch sai số tối thiểu: 0.0001 Hệ số học: 0.5. Hệ số quán tính: 0.3. Giải thuật khử mùa: RTMA Giải thuật khử xu hướng: khử xu hướng tuyến tính		MAE: 23.90 MSE: 898.00 MAPE: 4.69%



Hình 6.2: Kết quả dự đoán của 3 mô hình cho chuỗi số lượng khách hàng đặt chỗ hàng tháng của hãng hàng không Pan Am. Đường màu đỏ là giá trị dự đoán, đường màu xanh là giá trị quan sát thực tế

Đây là bộ dữ liệu mà chúng tôi lấy từ bộ dữ liệu có sẵn trong phần mềm R. Bộ dữ liệu này là một chuỗi thời gian có chiều dài 144 vừa có tính xu hướng vừa có tính mùa (hình 6.1). Chúng tôi dùng các giá trị từ 1 đến 132 để huấn luyện cho mô hình, các giá trị từ 133 đến 144 dùng để kiểm tra độ chính xác dự báo của mô hình.

Vì đây là chuỗi thời gian có tính mùa với chu kỳ 12, chúng tôi chọn mạng neuron có ba tầng với 12 đơn vị tầng nhập, 12 đơn vị tầng ẩn, 1 đơn vị tầng xuất.

Đối với mô hình dùng mạng neuron thuần túy, chúng tôi tiến hành chạy trên các cấu hình sau:

1. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.001, số epoch tối đa lần lượt là 1000 và 1500.
2. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.001, số epoch tối đa lần lượt là 1000 và 1500.

Mỗi cấu hình trên chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình dùng mạng neuron thuần túy.

Đối với mô hình lai giữa mạng neuron với làm trơn theo hàm mũ, chúng tôi chọn cấu hình cho mạng neuron là cấu hình tốt nhất ở mô hình dùng mạng neuron thuần túy. Mô hình làm trơn theo hàm mũ sẽ được áp dụng lần theo mô hình có tính nhân lẫn mô hình có tính cộng. Các thông số α, β, γ cũng được ước lượng theo hai cách là dùng phần mềm R và bằng vét cạn kết hợp với tôi luyện mô phỏng do chúng tôi hiện thực với bước dịch chuyển cho giải thuật vét cạn 0.1 và cho giải thuật tôi luyện mô phỏng là 0.00001. Mỗi cấu hình chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình lai giữa mạng neuron với làm trơn theo hàm mũ.

Đối với mô hình dùng mạng neuron kết hợp với khử mùa và khử xu hướng, chúng tôi tiến hành chạy trên các cấu hình sau:

1. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1500. Phương pháp khử mùa và khử xu hướng lần lượt là lấy hiệu theo mùa và lấy hiệu.
2. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1500. Phương pháp khử mùa và khử xu hướng lần lượt là RTMA và khử xu hướng tuyến tính.

3. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1500. Phương pháp khử mùa và khử xu hướng lần lượt là lấy hiệu theo mùa và lấy hiệu.
4. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1500. Phương pháp khử mùa và khử xu hướng lần lượt là RTMA và khử xu hướng tuyến tính.

Mỗi cấu hình trên chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình dùng mạng neuron kết hợp với khử mùa và khử xu hướng. Kết quả thực nghiệm cho các mô hình cho ở bảng 6.1. Hình 6.2 là kết quả kiểm tra đối với chuỗi dữ liệu bằng mạng neuron thuần, mô hình lai và mô hình khử mùa, khử xu hướng kết hợp mạng neuron.

Nhận xét: Đối với bộ dữ liệu này thì cả hai mô hình cải tiến đều cho kết quả dự báo tốt hơn mạng neuron nhân tạo, trong đó mô hình lai giữa mạng neuron nhân tạo và kỹ thuật làm trơn theo hàm mũ có kết quả dự báo tốt nhất.

6.2.2. Bộ dữ liệu mật độ khí cacbonic trong khí quyển hàng tháng ở Mauna Loa (Hawaii).

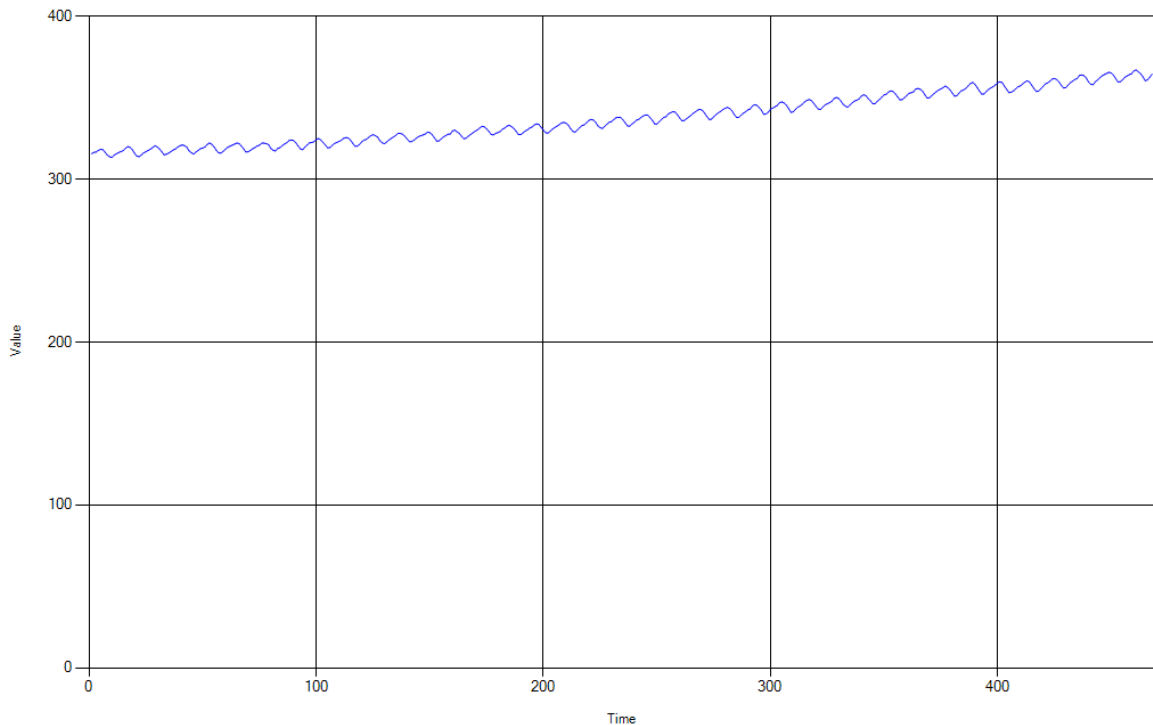
Đây là bộ dữ liệu mà chúng tôi lấy từ bộ dữ liệu có sẵn trong phần mềm R. Bộ dữ liệu này là một chuỗi thời gian có chiều dài 468 vừa có tính xu hướng vừa có tính mùa. Chúng tôi dùng các giá trị từ 1 đến 456 để huấn luyện cho mô hình, các giá trị từ 457 đến 468 dùng để kiểm tra độ chính xác dự báo của mô hình. Hình 6.3 biểu diễn hình dạng của chuỗi thời gian này.

Vì đây là chuỗi thời gian có tính mùa với chu kỳ 12, chúng tôi chọn neuron có ba tầng với 12 đơn vị tầng nhập, 12 đơn vị tầng ẩn, 1 đơn vị tầng xuất.

Đối với mô hình dùng mạng neuron thuần túy, chúng tôi tiến hành chạy trên các cấu hình sau:

1. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.001, số epoch tối đa lần lượt là 1000 và 1500.

2. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.001, số epoch tối đa lần lượt là 1000 và 1500.
3. Mỗi cấu hình trên chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình dùng mạng neuron thuần túy.

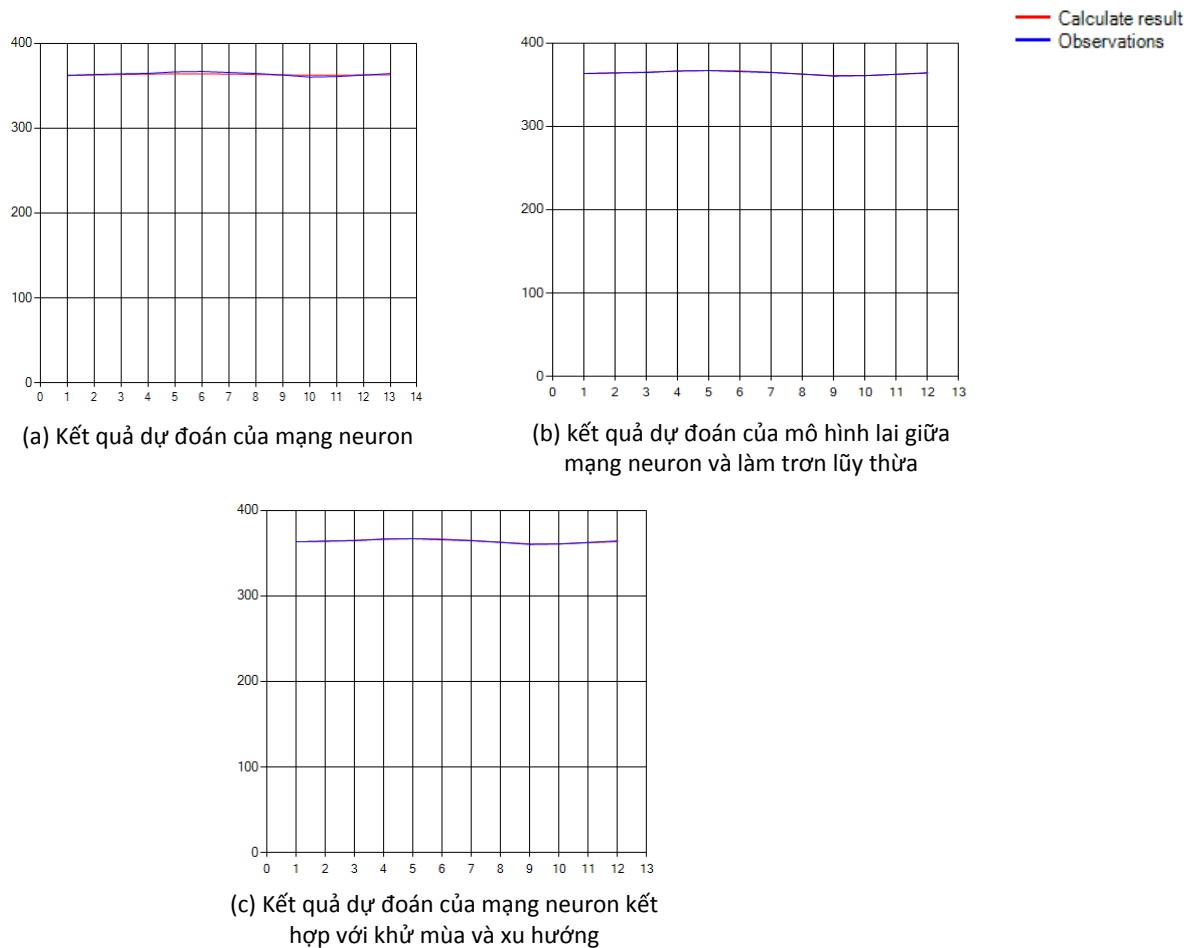


Hình 6.3: Mật độ khí cacbonic trong khí quyển hàng tháng ở Mauna Loa (Hawaii)

Đối với mô hình lai giữa mạng neuron với làm trơn theo hàm mũ, chúng tôi chọn cấu hình cho mạng neuron là cấu hình tốt nhất ở mô hình dùng mạng neuron thuần túy. Mô hình làm trơn theo hàm mũ sẽ được áp dụng lần theo mô hình có tính nhân lẫn mô hình có tính cộng. Các thông số α , β , γ cũng được ước lượng theo hai cách là dùng phần mềm R và bằng vét cạn kết hợp với tôi luyện mô phỏng do chúng tôi hiện thực với bước dịch chuyển cho giải thuật vét cạn 0.1 và cho giải thuật tôi luyện mô phỏng là 0.00001. Mỗi cấu hình chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình lai giữa mạng neuron với làm trơn theo hàm mũ.

Bảng 6.2: Kết quả thực nghiệm cho chuỗi mật độ khí cacbonic trong khí quyển hàng tháng ở Mauna Loa (Hawaii)

Mô hình	Cấu hình	Thông số được ước lượng	Sai số dự báo
Mạng neuron nhân tạo(12-12-1)	Giải thuật: lan truyền ngược Số epoch tối đa: 1000. Độ lệch sai số tối thiểu: 0.001 Hệ số học: 0.5. Hệ số quán tính: 0.3.		MAE: 2.54 MSE: 8.47 MAPE: 0.65%
Mô hình lai giữa mạng neuron nhân tạo(12-12-1) với làm trơn theo hàm mũ.	Mô hình có tính nhân. Ước lượng α, β, γ bằng vét cạn và tối luyện mô phỏng	α : 0.52495 β : 0.00886 γ : 0.49745 ω : 0	MAE: 0.32 MSE: 0.16 MAPE: 0.09%
Mạng neuron nhân tạo(12-12-1) kết hợp với khử mùa và khử xu hướng	Giải thuật: lan truyền ngược Số epoch tối đa: 1000. Độ lệch sai số tối thiểu: 0.0001 Hệ số học: 0.5. Hệ số quán tính: 0.3. Giải thuật khử mùa: RTMA Giải thuật khử xu hướng: khử xu hướng tuyến tính		MAE: 0.33 MSE: 0.19 MAPE: 0.09%



Hình 6.4: Kết quả dự đoán của 3 mô hình cho chuỗi mật độ khí cacbonic trong khí quyển hàng tháng ở Mauna Loa (Hawaii). Đường màu đỏ là giá trị dự đoán, đường màu xanh là giá trị quan sát thực tế

Đối với mô hình dùng mạng neuron kết hợp với khử mùa và khử xu hướng, chúng tôi tiến hành chạy trên các cấu hình sau:

1. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa và khử xu hướng lần lượt là lấy hiệu theo mùa và lấy hiệu.
2. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa và khử xu hướng lần lượt là RTMA và khử xu hướng tuyến tính.

3. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa và khử xu hướng lần lượt là lấy hiệu theo mùa và lấy hiệu.
4. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa và khử xu hướng lần lượt là RTMA và khử xu hướng tuyến tính.

Mỗi cấu hình trên chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình dùng mạng neuron kết hợp với khử mùa và khử xu hướng.

Kết quả thực nghiệm cho các mô hình ở bảng 6.2. Hình 6.4 là kết quả kiểm tra đối với bộ dữ liệu này bằng mô hình neuron thuần, mô hình lai, mô hình khử mùa, khử xu hướng với mạng neuron.

Nhận xét: Đối với bộ dữ liệu này thì kết quả dự báo ở hai mô hình (mô hình lai và mô hình kết hợp giữa mạng neuron nhân tạo với khử xu hướng và khử mùa) đều tốt (MAPE bằng 0.09%) hơn mạng neuron.

6.2.3. Bộ dữ liệu số người chết trung bình hàng tháng vì bệnh phổi ở Anh.

Đây là bộ dữ liệu mà chúng tôi lấy từ bộ dữ liệu có sẵn trong phần mềm R. Bộ dữ liệu này là một chuỗi thời gian có chiều dài 72 có tính mùa. Chúng tôi dùng các giá trị từ 1 đến 60 để huấn luyện cho mô hình, các giá trị từ 61 đến 72 dùng để kiểm tra độ chính xác dự báo của mô hình. Hình 6.5 là đồ thị biểu diễn chuỗi dữ liệu.

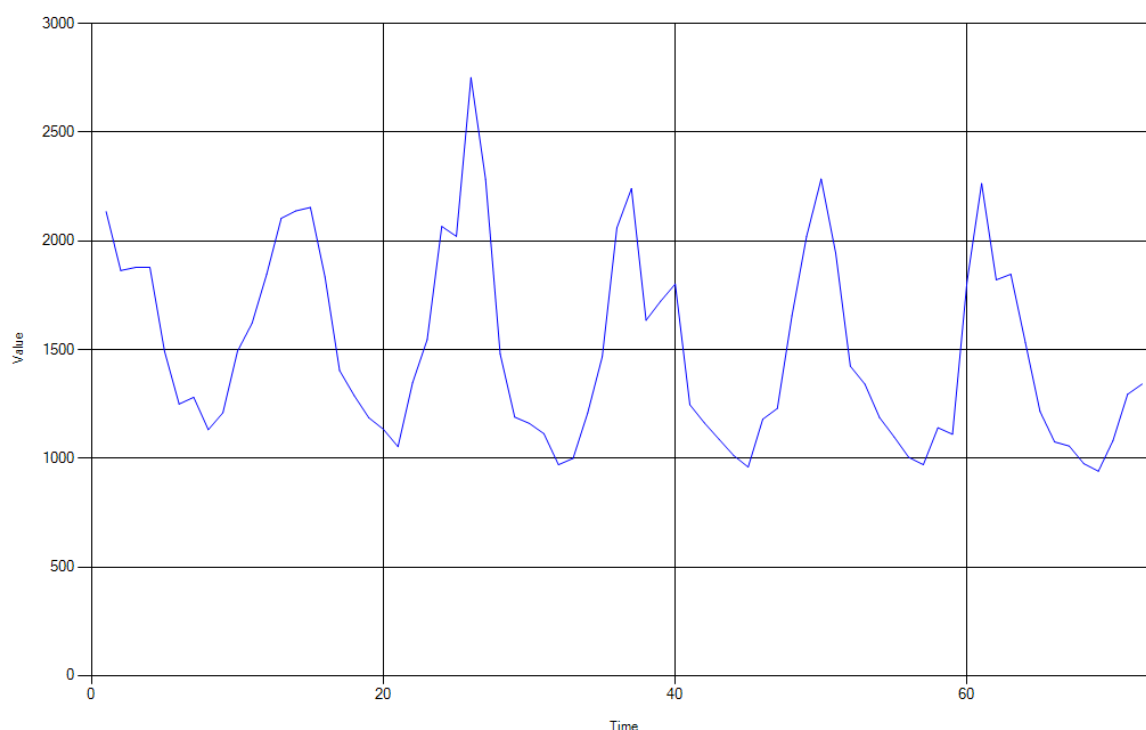
Vì đây là chuỗi thời gian có tính mùa với chu kỳ 12, chúng tôi chọn neuron có ba tầng với 12 đơn vị tầng nhập, 12 đơn vị tầng ẩn, 1 đơn vị tầng xuất.

Đối với mô hình dùng mạng neuron thuần túy, chúng tôi tiến hành chạy trên các cấu hình sau:

1. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.001, số epoch tối đa lần lượt là 1000 và 1500.

2. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.001, số epoch tối đa lần lượt là 1000 và 1500.

Mỗi cấu hình trên chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình dùng mạng neuron thuần túy.

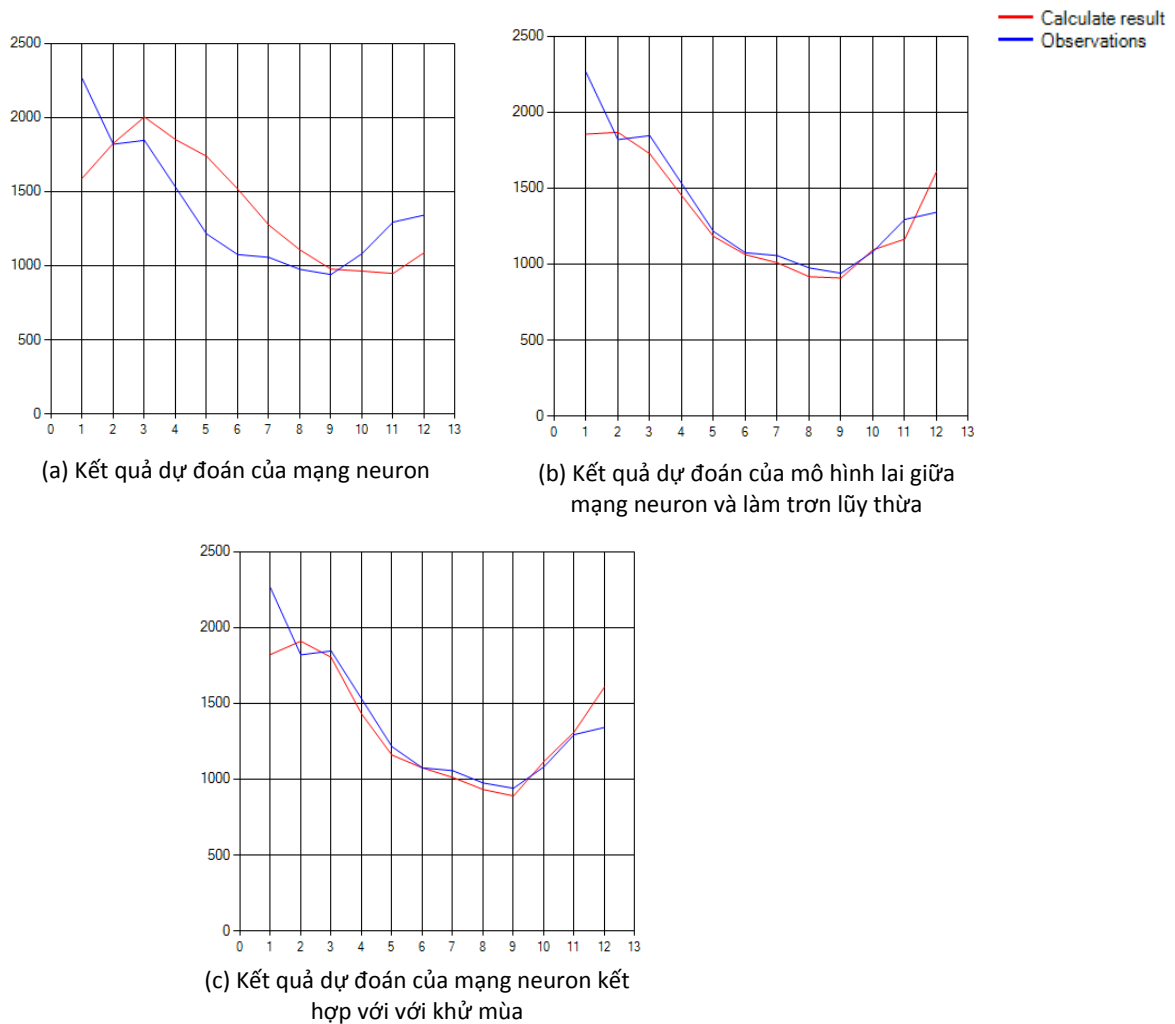


Hình 6.5: Số người chết trung bình hàng tháng vì bệnh phổi ở Anh

Đối với mô hình lai giữa mạng neuron với làm trơn theo hàm mũ, chúng tôi chọn cấu hình cho mạng neuron là cấu hình tốt nhất ở mô hình dùng mạng neuron thuần túy. Mô hình làm trơn theo hàm mũ sẽ được áp dụng lần theo mô hình có tính nhân lẫn mô hình có tính cộng. Các thông số α , β , γ cũng được ước lượng theo hai cách là dùng phần mềm R và bằng vét cạn kết hợp với tôi luyện mô phỏng do chúng tôi hiện thực với bước dịch chuyển cho giải thuật vét cạn 0.1 và cho giải thuật tôi luyện mô phỏng là 0.00001. Mỗi cấu hình chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình lai giữa mạng neuron với làm trơn theo hàm mũ.

Bảng 6.3: Kết quả thực nghiệm cho chuỗi số người chết trung bình hàng tháng vì bệnh phổi ở Anh

Mô hình	Cấu hình	Thông số được ước lượng	Sai số dự báo
Mạng neuron nhân tạo(12-12-1)	Giải thuật: lan truyền ngược Số epoch tối đa: 1000. Độ lệch sai số tối thiểu: 0.001 Hệ số học: 0.5. Hệ số quán tính: 0.3.		MAE: 274.66 MSE: 112914.60 MAPE: 20.37%
Mô hình lai giữa mạng neuron nhân tạo(12-12-1) với làm trơn theo hàm mũ	Mô hình có tính nhân. Ước lượng α, β, γ bằng vét cạn và tối luyện mô phỏng	α : 0.4144 β : 0.09989 γ : 0.33591 ω : 0.07789	MAE: 103.98 MSE: 23829.37 MAPE: 6.77%
Mạng neuron nhân tạo(12-12-1) kết hợp với khử mùa	Giải thuật: lan truyền ngược Số epoch tối đa: 1000. Độ lệch sai số tối thiểu: 0.0001 Hệ số học: 0.5. Hệ số quán tính: 0.3. Giải thuật khử mùa: RTMA		MAE: 98.4 MSE: 23881.2 MAPE: 6.41%



Hình 6.6: Kết quả dự đoán của 3 mô hình cho chuỗi số người chết trung bình hàng tháng vì bệnh phổi ở Anh. Đường màu đỏ là giá trị dự đoán, đường màu xanh là giá trị quan sát thực tế

Đối với mô hình dùng mạng neuron kết hợp với khử mùa và khử xu hướng, chúng tôi tiến hành chạy trên các cấu hình sau:

1. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa là lấy hiệu theo mùa.
2. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa là RTMA.

3. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa là lấy hiệu theo mùa.
4. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa là RTMA.

Mỗi cấu hình trên chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình dùng mạng neuron kết hợp với khử mùa và khử xu hướng.

Kết quả thực nghiệm của ba mô hình ở bảng 6.3. Hình 6.6 là kết quả kiểm tra ứng với mô hình mạng neuron thuần, mô hình lai giữa mạng neuron và kỹ thuật làm trơn theo hàm mũ, mô hình khử mùa, khử xu hướng kết hợp mạng neuron.

Nhận xét: Đối với bộ dữ liệu này thì hai mô hình cải tiến cho kết quả dự báo tốt hơn mạng neuron nhiều, trong đó mô hình kết hợp giữa mạng neuron nhân tạo và khử xu hướng, khử mùa có kết quả dự báo tốt nhất.

6.2.4. Bộ dữ liệu doanh số bán hàng hàng tháng của một cửa hàng bán đồ lưu niệm ở Queensland, Australia, từ năm 1987 đến năm 1993.

Đây là bộ dữ liệu mà chúng tôi lấy từ bộ dữ liệu có sẵn trong phần mềm R. Bộ dữ liệu này là một chuỗi thời gian có chiều dài 84 có tính mùa và một chút xu hướng. Chúng tôi dùng các giá trị từ 1 đến 72 để huấn luyện cho mô hình, các giá trị từ 73 đến 84 dùng để kiểm tra độ chính xác dự báo của mô hình. Hình 6.7 là đồ thị biểu diễn cho chuỗi dữ liệu.

Vì đây là chuỗi thời gian có tính mùa với chu kỳ 12, chúng tôi chọn neuron có ba tầng với 12 đơn vị tầng nhập, 12 đơn vị tầng ẩn, 1 đơn vị tầng xuất.

Đối với mô hình dùng mạng neuron thuần túy, chúng tôi tiến hành chạy trên các cấu hình sau:

1. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.001, số epoch tối đa lần lượt là 1000 và 1500.
2. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.001, số epoch tối đa lần lượt là 1000 và 1500.

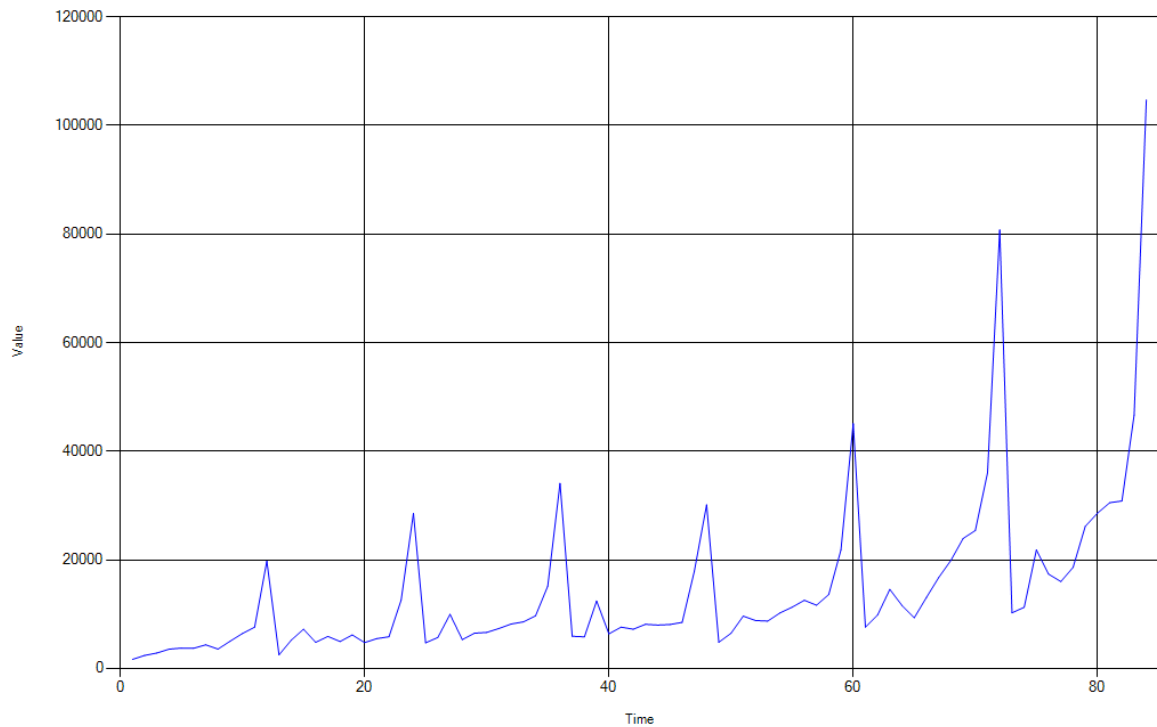
Mỗi cấu hình trên chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình dùng mạng neuron thuần túy.

Đối với mô hình lai giữa mạng neuron với làm trơn theo hàm mũ, chúng tôi chọn cấu hình cho mạng neuron là cấu hình tốt nhất ở mô hình dùng mạng neuron thuần túy. Mô hình làm trơn theo hàm mũ sẽ được áp dụng lần theo mô hình có tính nhân lẫn mô hình có tính cộng. Các thông số α, β, γ cũng được ước lượng theo hai cách là dùng phần mềm R và bằng vét cạn kết hợp với tôi luyện mô phỏng do chúng tôi hiện thực với bước dịch chuyển cho giải thuật vét cạn 0.1 và cho giải thuật tôi luyện mô phỏng là 0.00001. Mỗi cấu hình chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình lai giữa mạng neuron với làm trơn theo hàm mũ.

Đối với mô hình dùng mạng neuron kết hợp với khử mùa và khử xu hướng, chúng tôi tiến hành chạy trên các cấu hình sau:

1. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa và khử xu hướng lần lượt là lấy hiệu theo mùa và lấy hiệu.
2. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa và khử xu hướng lần lượt là RTMA và khử xu hướng tuyến tính.

3. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa và khử xu hướng lần lượt là RTMA và lấy hiệu.

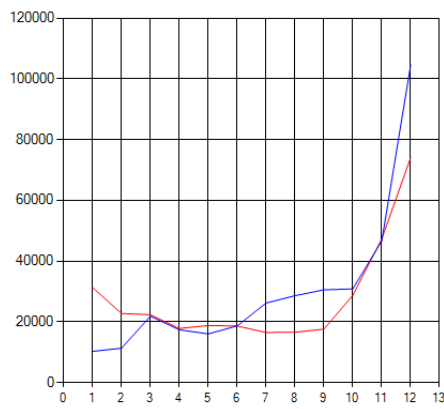


Hình 6.7: Doanh số bán hàng hàng tháng của một cửa hàng bán đồ lưu niệm ở Queensland, Australia, từ năm 1987 đến năm 1993

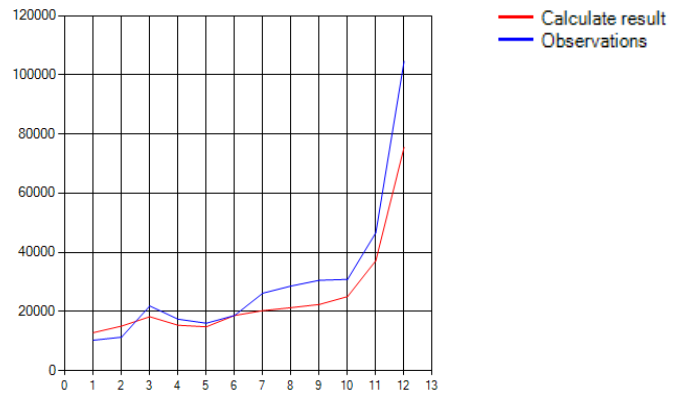
4. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa và khử xu hướng lần lượt là lấy hiệu theo mùa và lấy hiệu.
5. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa và khử xu hướng lần lượt là RTMA và khử xu hướng tuyến tính.
6. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa và khử xu hướng lần lượt là RTMA và lấy hiệu.

Bảng 6.4: Kết quả thực nghiệm cho chuỗi doanh số bán hàng hàng tháng của một cửa hàng bán đồ lưu niệm ở Queensland, Australia, từ năm 1987 đến năm 1993

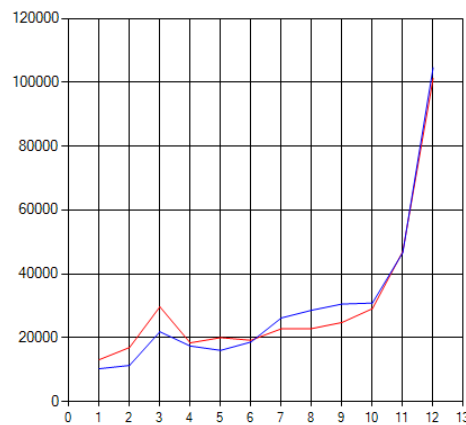
Mô hình	Cấu hình	Thông số được ước lượng	Sai số dự báo
Mạng neuron nhân tạo(12-12-1)	Giải thuật: RPROP Số epoch tối đa: 1500. Độ lệch sai số tối thiểu: 0.001 Hệ số cập nhập mặc định (A_0): 0.1. Hệ số cập nhập lớn nhất (A_{max}): 1.		MAE: 20319.49 MSE: 920433586.8 MAPE: 88.15%
Mô hình lai giữa mạng neuron nhân tạo(12-12-1) với làm trơn theo hàm mũ	Mô hình có tính nhân. Ước lượng α, β, γ bằng vét cạn và tối luyện mô phỏng	α : 0.63418 β : 0.0 γ : 0.63532 ω : 0.211356	MAE: 6584.78 MSE: 97239348.98 MAPE: 19.68%
Mạng neuron nhân tạo(12-12-1) kết hợp với khử mùa	Giải thuật: RPROP Số epoch tối đa: 1000. Độ lệch sai số tối thiểu: 0.001 Hệ số cập nhập mặc định (A_0): 0.1. Hệ số cập nhập lớn nhất (A_{max}): 1. Giải thuật khử mùa: RTMA Giải thuật khử xu hướng: lấy hiệu		MAE: 3817.22 MSE: 20136815 MAPE: 18.1%



(a) Kết quả dự đoán của mạng neuron



(b) Kết quả dự đoán của mô hình lai giữa mạng neuron và làm trơn lũy thừa



(c) Kết quả dự đoán của mô hình mạng neuron kết hợp với khử mùa và xu hướng

Hình 6.8: Kết quả dự đoán của 3 mô hình cho chuỗi doanh số bán hàng hàng tháng của một cửa hàng bán đồ lưu niệm ở Queensland, Australia.

Mỗi cấu hình trên chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình dùng mạng neuron kết hợp với khử mùa và khử xu hướng.

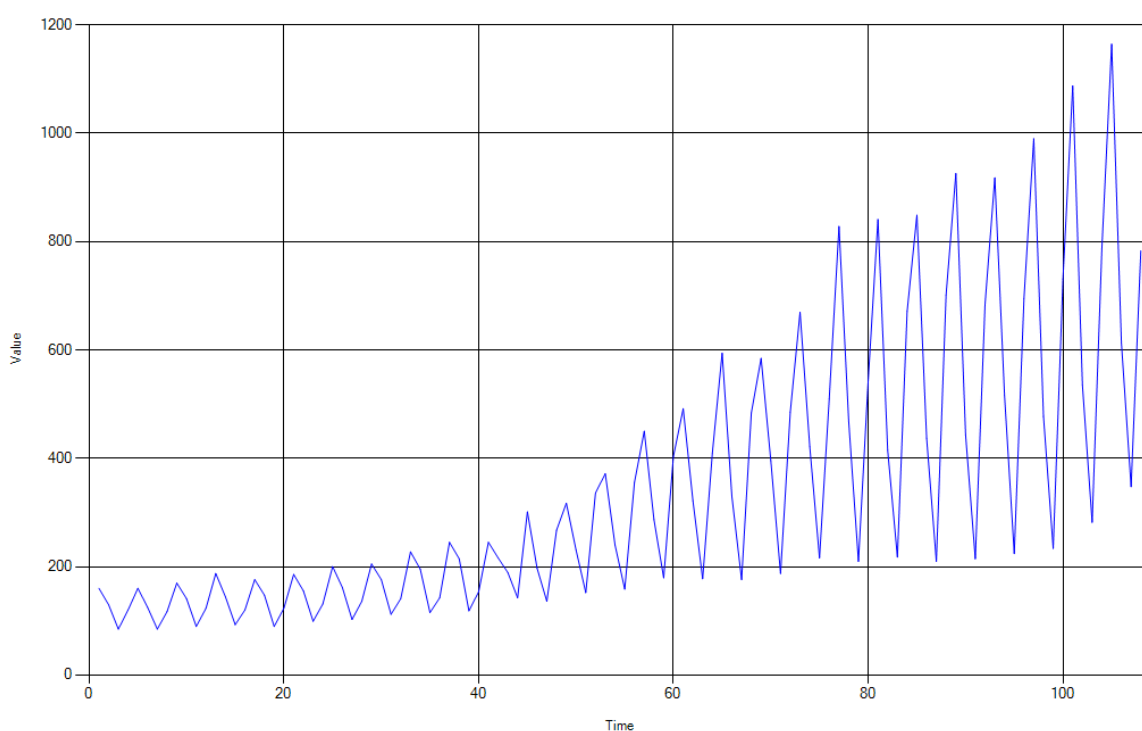
Kết quả thực nghiệm của ba mô hình ở bảng 6.4. Hình 6.8 biểu diễn kết quả kiểm tra của ba mô hình.

Nhận xét: Đối với bộ dữ liệu này thì hai mô hình cải tiến cho kết quả dự báo tốt hơn mạng neuron, trong đó mô hình kết hợp giữa mạng neuron nhân tạo và khử xu hướng, khử mùa có kết quả dự báo tốt nhất.

6.2.5. Bộ dữ liệu lượng tiêu thụ khí đốt trung bình theo quý tại Anh.

Đây là bộ dữ liệu mà chúng tôi lấy từ bộ dữ liệu có sẵn trong phần mềm R. Bộ dữ liệu này là một chuỗi thời gian có chiều dài 108 vừa có tính xu hướng vừa có tính mùa. Chúng tôi dùng các giá trị từ 1 đến 104 để huấn luyện cho mô hình, các giá trị từ 104 đến 108 dùng để kiểm tra độ chính xác dự báo của mô hình. Hình 6.9 là đồ thị biểu diễn của bộ dữ liệu.

Vì đây là chuỗi thời gian có tính mùa với chu kỳ 4, chúng tôi chọn neuron có ba tầng với 4 đơn vị tầng nhập, 4 đơn vị tầng ẩn, 1 đơn vị tầng xuất.



Hình 6.9: Lượng tiêu thụ khí đốt trung bình theo quý tại Anh

Đối với mô hình dùng mạng neuron thuần túy, chúng tôi tiến hành chạy trên các cấu hình sau:

1. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.001, số epoch tối đa lần lượt là 1000 và 1500.
2. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.001, số epoch tối đa lần lượt là 1000 và 1500.

Mỗi cấu hình trên chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình dùng mạng neuron thuần túy.

Đối với mô hình lai giữa mạng neuron với làm trơn theo hàm mũ, chúng tôi chọn cấu hình cho mạng neuron là cấu hình tốt nhất ở mô hình dùng mạng neuron thuần túy. Mô hình làm trơn theo hàm mũ sẽ được áp dụng lần theo mô hình có tính nhân lẫn mô hình có tính cộng. Các thông số α , β , γ cũng được ước lượng theo hai cách là dùng phần mềm R và bằng vết cận kết hợp với tôi luyện mô phỏng do chúng tôi hiện thực với bước dịch chuyển cho giải thuật vết cận 0.1 và cho giải thuật tôi luyện mô phỏng là 0.00001. Mỗi cấu hình chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình lai giữa mạng neuron với làm trơn theo hàm mũ.

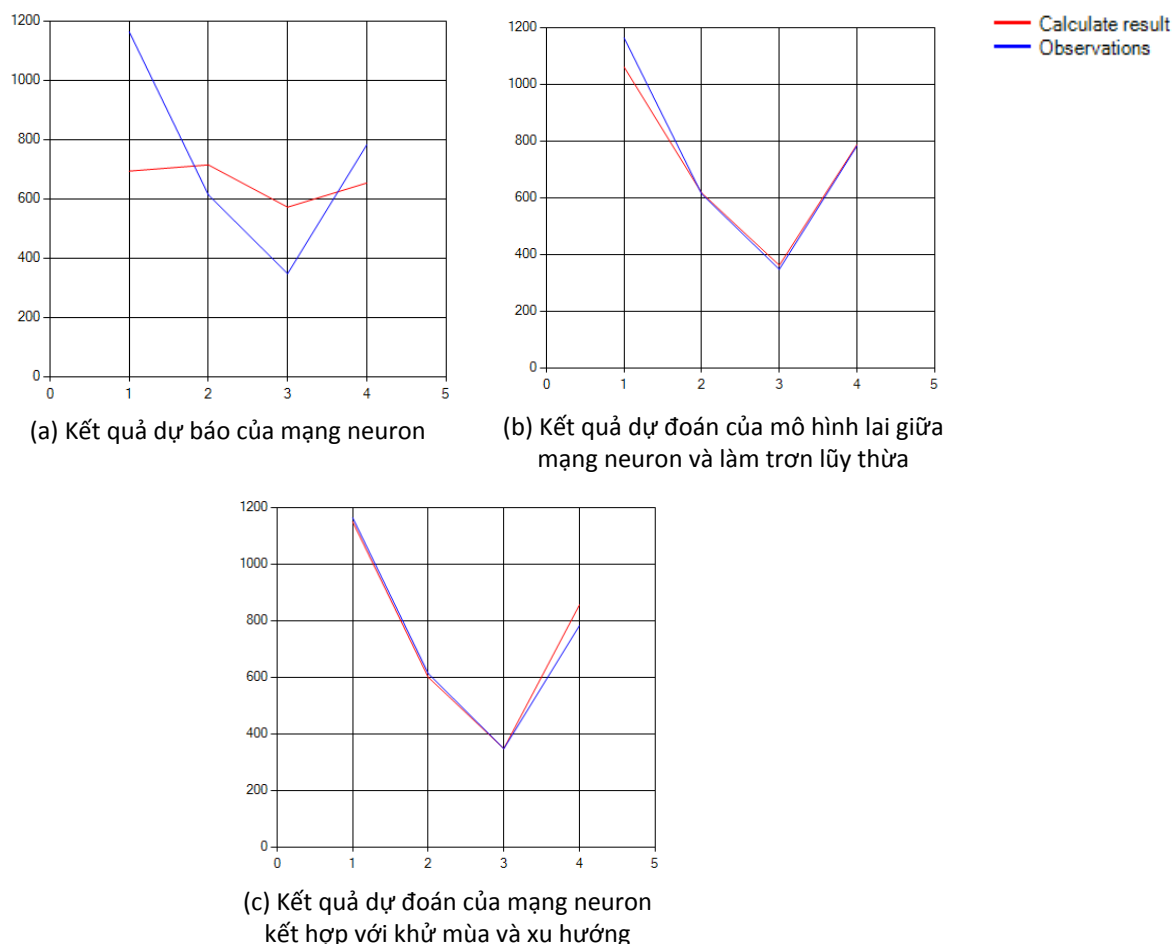
Đối với mô hình dùng mạng neuron kết hợp với khử mùa và khử xu hướng, chúng tôi tiến hành chạy trên các cấu hình sau:

1. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa là lấy hiệu theo mùa.
2. Dùng giải thuật RPROP với giá trị cập nhập mặc định (Δ_0) là 0.1, giá trị cập nhập tối đa (Δ_{max}) là 1.0, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa là RTMA.
3. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa là lấy hiệu theo mùa.
4. Dùng giải thuật lan truyền ngược với hệ số học là 0.5, hệ số quán tính là 0.3, độ lệch sai số tối thiểu là 0.0001, số epoch tối đa là 1000. Phương pháp khử mùa là RTMA.

Mỗi cấu hình trên chúng tôi thực hiện chạy 3 lần và lấy sai số trung bình. Sai số trung bình của cấu hình tốt nhất sẽ dùng làm sai số của mô hình dùng mạng neuron kết hợp với khử mùa và khử xu hướng.

Bảng 6.5: Kết quả thực nghiệm cho chuỗi lượng tiêu thụ khí đốt trung bình theo quý tại Anh

Mô hình	Cấu hình	Thông số được ước lượng	Sai số dự báo
Mạng neuron nhân tạo(4-4-1)	Giải thuật: RPROP Số epoch tối đa: 1000. Độ lệch sai số tối thiểu: 0.001 Hệ số cập nhập mặc định (Δ_0): 0.1. Hệ số cập nhập lớn nhất (Δ_{max}): 1.		MAE: 239.42 MSE: 79434.4 MAPE: 37.73%
Mô hình mạng neuron nhân tạo(4-4-1) với làm trơn theo hàm mũ	Mô hình có tính nhân. Ước lượng α, β, γ bằng phần mềm R	α : 0.025262 β : 1 γ : 0.715679 ω : 0.208328	MAE: 31.23 MSE: 2703.017 MAPE: 3.52%
Mạng neuron nhân tạo(4-4-1) kết hợp với khử mùa	Giải thuật: lan truyền ngược Số epoch tối đa: 1000. Độ lệch sai số tối thiểu: 0.0001 Hệ số học: 0.5. Hệ số quán tính: 0.3. Giải thuật khử mùa:lấy hiệu theo mùa. Giải thuật khử xu hướng: lấy hiệu.		MAE: 25.49 MSE: 1423.97 MAPE: 3.25%



Hình 6.10: Kết quả dự đoán của 3 mô hình cho chuỗi lượng tiêu thụ khí đốt trung bình theo quý tại Anh.

Kết quả thực nghiệm của các mô hình ở bảng 6.5. Hình 6.10 là kết quả kiểm tra đối với ba mô hình.

Nhận xét: Đối với bộ dữ liệu này thì hai mô hình cải tiến cho kết quả dự báo tốt hơn mạng neuron nhiều, trong đó mô hình kết hợp giữa mạng neuron nhân tạo và khử xu hướng, khử mùa có kết quả dự báo tốt nhất.

6.2.6. Nhận xét chung

Qua kết quả thực nghiệm trên 5 bộ dữ liệu ta thấy rằng mạng neuron nhân tạo không thể đưa ra kết quả dự báo chính xác cho các chuỗi thời gian có tính mùa và tính xu hướng. Trong khi đó hai mô hình cải tiến của nó: mô hình mạng neuron nhân tạo kết

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa hợp với làm trơn theo hàm mũ và mô hình mạng neuron nhân tạo kết hợp với khử mùa, khử xu hướng cho ra kết quả dự báo khá chính xác.

Chúng tôi cũng nhận thấy rằng mô hình mạng neuron nhân tạo kết hợp với khử mùa và khử xu hướng đưa ra dự báo ít chính xác hơn mô hình lai giữa mạng neuron nhân tạo với mô hình làm trơn hàm mũ trong hai bộ dữ liệu mật độ khí cacbonic trong khí quyển và số hành khách của hãng hàng không Pan Am nhưng cho kết quả chính xác hơn trong các bộ dữ liệu còn lại. Hai chuỗi thời gian mật độ khí cacbonic trong khí quyển và số hành khách của hãng hàng không Pan Am là hai chuỗi thời gian có tính mùa và tính xu hướng biến đổi một cách đều đặn, nghĩa là sự biến đổi giữa các chu kỳ liên tiếp nhau diễn ra đều đặn với độ thay đổi biên độ gần như không đổi theo thời gian và không có sự đột biến giữa các chu kỳ. Trong khi đó với chuỗi số người chết trung bình do bệnh phổi ở Anh có sự biến đổi đột ngột ở chu kỳ thứ ba so với các chu kỳ khác, chuỗi doanh số bán hàng của cửa hàng lưu niệm có sự biến đổi đột ngột ở chu kỳ thứ sáu và thứ bảy so với các chu kỳ trước, chuỗi lượng khí đốt tiêu thụ ở Anh có sự biến đổi đột ngột biên độ ở nửa sau của chuỗi. Từ đó chúng tôi cho rằng mô hình mạng neuron nhân tạo kết hợp với khử mùa và xu hướng thích hợp với những chuỗi thời gian có mùa và xu hướng mà có sự biến đổi đột ngột ở các chu kỳ, trong khi đó mô hình lai giữa mạng neuron nhân tạo với làm trơn theo hàm mũ sẽ thích hợp hơn với những chuỗi mà sự biến đổi giữa các chu kỳ liên tiếp là đều đặn với độ thay đổi biên độ ít biến đổi theo thời gian.

Trong điều kiện thời gian của luận văn và sự không sẵn có nhiều bộ dữ liệu, chúng tôi chưa thể chạy trên thật nhiều bộ dữ liệu để đưa ra một thống kê đáng tin cậy hơn nên điều khẳng định trên có thể được xem là một kết luận rút ra từ một số thực nghiệm ban đầu.

Chương 7

KẾT LUẬN

Trong chương này, chúng tôi trình bày các kết quả mà chúng tôi đã thực hiện được trong luận văn này cùng với những hạn chế và hướng phát triển của đề tài.

7.1. Đánh giá kết quả

7.1.1. Những công việc làm được

Trong quá trình thực hiện đề tài, chúng tôi đã làm được những công việc sau:

- Tìm hiểu việc áp dụng mạng neuron để dự báo đối với dữ liệu chuỗi thời gian.
- Tìm hiểu các phương pháp khử mùa và khử xu hướng đối với dữ liệu chuỗi thời gian như: khử xu hướng bằng phương pháp lấy hiệu, khử xu hướng bằng phương pháp tuyến tính, lấy hiệu theo mùa, khử mùa bằng phương pháp RTMA.
- Tìm hiểu các kỹ thuật làm trơn theo hàm mũ như: làm trơn theo hàm mũ giản đơn, làm trơn theo hàm mũ Holt, làm trơn theo hàm mũ Winters. Ngoài ra, tiến hành nghiên cứu và tìm ra phương pháp ước lượng các hệ số trong kỹ thuật làm trơn theo hàm mũ Winters bằng việc kết hợp vét cạn và phương pháp tối ưu hóa mô phỏng.
- Tìm hiểu phương thức gọi hàm R bằng R(D)COM trong chương trình C#.NET
- Nghiên cứu việc kết hợp hai kỹ thuật: khử mùa, khử xu hướng và làm trơn theo hàm mũ với mạng neuron nhằm nâng cao chất lượng dự báo đối với dữ liệu chuỗi thời gian có tính mùa và xu hướng.
- Tiến hành hiện thực hai mô hình dự báo từ các nghiên cứu trên.
- Tiến hành chạy thực nghiệm với năm bộ dữ liệu thực tế và đánh giá, kiểm chứng tính đúng đắn của cơ sở lý thuyết cũng như quá trình hiện thực.

7.1.2. Những đúc kết về mặt lý luận

Bản thân mạng neuron nhân tạo không có khả năng dự đoán tốt cho chuỗi thời gian có tính mùa và tính xu hướng. Mô hình lai giữa mạng neuron nhân tạo với làm trơn theo hàm mũ và mô hình mạng neuron nhân tạo kết hợp với khử mùa và xu hướng cho chất lượng dự báo tốt hơn nhiều. Trong quá trình tìm hiểu và thực nghiệm, chúng tôi rút ra được một số kết luận về hai mô hình này

Đối với mô hình lai:

- Mô hình nhân và mô hình cộng trong kỹ thuật làm trơn theo hàm mũ có kết quả phù hợp cho từng trường hợp cụ thể. Mô hình cộng tỏ ra hiệu quả hơn mô hình nhân đối với các chuỗi dữ liệu có tính mùa với biên độ mùa ổn định, trong khi mô hình nhân phù hợp với các chuỗi dữ liệu có tính mùa với biên độ mùa biến đổi.
- Trọng số lai giữa mạng neuron nhân tạo và mô hình làm trơn theo hàm mũ có giá trị phụ thuộc vào kết quả huấn luyện của mạng neuron và tính chất mùa của chuỗi dữ liệu. Nếu chuỗi dữ liệu có tính mùa rõ, ít giá trị ngoại lai thì trọng số lai sẽ nhỏ, tức là nghiêng về phía mô hình làm trơn hay mô hình làm trơn chiếm ưu thế trong mô hình lai này. Ngược lại nếu tính mùa của chuỗi không rõ ràng và có nhiều ngoại lai thì trọng số này càng lớn.

Đối với mô hình khử mùa kết hợp mạng neuron:

- Việc lựa chọn phương pháp khử mùa, khử xu hướng phụ thuộc vào tính chất của chuỗi dữ liệu. Trong quá trình thực nghiệm, đối với các chuỗi dữ liệu ở 6.2.1, 6.2.2, 6.2.3, 6.2.5 thì việc kết hợp lấy hiệu khử mùa với lấy hiệu khử xu hướng và khử xu hướng tuyến tính với khử mùa với RTMA tỏ ra khá hiệu quả. Nhưng đối với chuỗi dữ liệu ở 6.2.4, khử mùa bằng RTMA và lấy hiệu khử mùa với khử xu hướng tuyến tính tỏ ra khá hiệu quả.

Nhìn chung, cả hai mô hình đều đưa ra dự báo chính xác hơn nhiều so với mạng neuron nhân tạo thường đối với dữ liệu có tính mùa và tính xu hướng. Kết quả thực nghiệm từ năm bộ dữ liệu đã chứng tỏ điều trên.

7.1.3. Mặt hạn chế

Kỹ thuật xu hướng tuyến tính do chúng tôi hiện thực chỉ xử xu hướng tốt cho các chuỗi mà xu hướng có dạng đường thẳng.

Trong mô hình lai, việc khởi tạo các thông số ban đầu trong mô hình làm trơn theo hàm mũ tuy khá hiệu quả và dễ hiện thực nhưng kết quả làm trơn sẽ không được như mong đợi nếu chuỗi dữ liệu quá ngắn hoặc có nhiều nhiễu [17]. Gần đây các nhà nghiên cứu có đưa ra một số phương pháp khác để khởi tạo các thông số ban đầu cho mô hình làm trơn Holt-Winters nhưng do hạn chế về thời gian nên chúng tôi chưa thể tìm hiểu và hiện thực.

Chưa có điều kiện chạy thực nghiệm trên thật nhiều bộ dữ liệu có các tính chất khác nhau để đưa ra một kết luận đáng tin cậy về chất lượng dự báo của mô hình lai giữa mạng neuron nhân tạo với làm trơn theo hàm mũ và mô hình mạng neuron nhân tạo kết hợp với xử mùa, xử xu hướng.

Do mục đích của việc hiện thực chương trình chỉ dành cho mục đích thực nghiệm, kiểm tra và so sánh tính hiệu quả của các mô hình lý thuyết nên giao diện còn nhiều hạn chế, có đôi phần khó sử dụng.

7.2. Hướng phát triển

Đối với mô hình lai, thay thế phương thức khởi tạo các thông số đối với mô hình làm trơn theo hàm mũ đang dùng bằng các phương pháp tiên tiến hơn như phương pháp *dựa trên hồi quy* (regression-based procedure) [17] hay *phương pháp dựa trên phân giải* (decomposition-based) [17] để có thể đưa ra dự báo chính xác hơn.

Đối với mô hình xử mùa, xử xu hướng kết hợp mạng neuron, cải tiến phương pháp xử xu hướng tuyến tính để có thể áp dụng tốt cho các chuỗi thời gian có xu hướng mang hình dạng đường cong. Áp dụng các kỹ thuật xử mùa tiên tiến gần đây như kỹ thuật X-12-ARIMA vào chương trình

TÀI LIỆU THAM KHẢO

- [1] G. Zhang, M. Qi. *Trend Time-Series Modeling And Forecasting With Neural Networks*. IEEE Transactions on Neural Network, vol. 19, no. 5, pages 808-816, 2008.
- [2] G. Zhang, D. M. Kline. *Quarterly Time-Series Forecasting With Neural Networks*. IEEE Transactions on Neural Network, vol. 18, no. 6, pages 1800-1814, 2007.
- [3] K. Lai, L. Yu, S. Wang, W. Huang. *Hybridizing Exponential Smoothing And Neural Network For Financial Time Series Predication*. ICCS'06 Proceedings of the 6th international conference on Computational Science, vol. 4, pages 493-500, 2006.
- [4] G. Zhang, M. Qi. *Neural Network Forecasting For Seasonal And Trend Time Series*. European Journal of Operational Research vol. 160, pages 501-514, 2005.
- [5] J. E. Hanke, D. W. Wichern. *Business Forecasting*, Pearson Prentice Hall, ISBN 0-13-141290-6, 2005.
- [6] Trần Đức Minh. Luận văn thạc sĩ *Mạng Neural Truyền Thống Và Ứng Dụng Trong Dự Báo Dữ Liệu*. Đại học quốc gia Hà Nội, 2002
- [7] F. Virili, B. Freisleben. *Preprocessing Seasonal Time Series For Improving Neural Network Predictions*. Proceedings of CIMA 99 Computational Intelligence Methods and Applications, Rochester-NY, pages 622-628, 1999.
- [8] G. Zhang, M. Y. Hu. *Neural Network Forecasting Of The British Pound/US Dollar Exchange Rate*. Omega, International Journal of Management Science, 26, pages 495-506, 1998.
- [9] T. M. Mitchell. *Machine Learning*, McGraw-Hill Science/ Engineering/ Math, ISBN 0070428077, 1997.
- [10] I. Kaastra, M. Boyd. *Designing A Neural Network For Forecasting Financial And Economic Time Series*. Neurocomputing, vol. 10, pages 215-236, 1996.
- [11] M. Riedmiller. *Advanced Supervised Learning In Multi-layer Perceptrons – From Backpropagation To Adaptive Learning Algorithms*. Int. Journal of Computer Standards and Interfaces, 1994.

- [12] T. Kolarik, G. Rudorfer. *Time Series Forecasting Using Neural Networks*. ACM Sigapl Apl Quote Quad, vol. 25, no. 1, pages 86-94, 1994.
- [13] M. Riedmiller, H. Braun. *A Direct Adaptive Method For Faster Backpropagation Learning: The RPROP Algorithm*. Proceedings of the IEEE International Conference on Neural Networks, pages 586-591, 1993.
- [14] The R Project for Statistical Computing at <http://www.r-project.org>.
- [15] StatConn at <http://rcom.univie.ac.at>.
- [16] <http://stat.ethz.ch/R-manual/R-patched/library/stats/html/HoltWinters.html>.
- [17] <http://www.r-bloggers.com/initializing-the-holt-winters-method/>.

Phụ lục A

Bảng thuật ngữ Anh-Việt

Activation Function	Hàm kích hoạt
Artificial Neural Network	Mạng neuron nhân tạo
Autocorrelation	Hệ số tự tương quan
Autocorrelation Function	Hàm tự tương quan
Backpropagation Algorithm	Giải thuật lan truyền ngược
Batch Learning	Học theo bó
Bias	Độ lệch
Cyclical	Tính chu kỳ
Decision Surface	Mặt quyết định
Delta Rule	Luật delta
Differencing	Lấy hiệu
Exponential Smoothing	Làm trơn theo hàm mũ
Feed-forward neural network	Mạng truyền thẳng
Global Minimum	cực tiểu toàn cục
Global Strategy	chiến lược toàn cục
Gradient	Độ dốc
Gradient Descent	Giảm độ dốc
Hyperplane	Siêu phẳng

Incremental Gradient Descent	Giảm độ dốc tăng cường
Irregular	Tính bất quy tắc
Learning By Epoch	Học theo epoch
Learning By Pattern	Học theo mẫu
Learning Rate	Hệ số học
Least Absolute Deviation	Độ lệch tuyệt đối nhỏ nhất
Linear Detrend	Khử xu hướng tuyến tính
Linear Unit	Đơn vị tuyến tính
Linearly Separable	Khả phân tuyến tính
Local Minimum	Tối ưu cục bộ
Local Strategy	Chiến lược cục bộ
Mean Absolute Errors	Trung bình tuyệt đối lỗi
Mean Absolute percentage Error	Trung bình phần trăm tuyệt đối
Mean Squared Errors	Trung bình bình phương lỗi
Measurable Function	Hàm khả đánh giá
Momentum Term	Hệ số quán tính
Online Learning	Học trực tuyến
Oscillation	Giao động
Overall Error	Lỗi tổng thể
Overfitting	Quá khớp

Pattern Set	Tập mẫu
Percentage Differences	Hiệu phần trăm
Perceptron Training Rule	Luật huấn luyện perceptron
Probability Distribution	Phân bố xác suất
Random Variable	Biến ngẫu nhiên
Recurrent Neural Network	Mạng hồi quy
Seasonal	Tính mùa
Seasonal Differencing	Lấy hiệu theo mùa
Sigmoid Unit	Đơn vị sigmoid
Sum Of Squared Errors	Tổng bình phương lỗi
Supervised Learning	Học có giám sát
Synapse	Khớp thần kinh
Thresholds	Phân ngưỡng
Time Series	Chuỗi thời gian
Training Algorithm	Giải thuật huấn luyện
Training Error Function	Hàm lỗi
Trend	Tính xu hướng
Unsupervised Learning	Học không có giám sát
Unthresholded Perceptron	Perceptron không phân ngưỡng

Phụ lục B

Hướng dẫn sử dụng chương trình thực nghiệm

B.1. Chương trình hiện thực mô hình lai

B.1.1. Môi trường sử dụng và cài đặt

Chương trình này được chúng tôi hiện thực bằng ngôn ngữ lập trình C#, trên nền .NET framework 4.0 với bộ hỗ trợ lập trình Visual C# 2010 Express.

Để thực hiện việc sử dụng chương trình này của chúng tôi, các bạn cần phải đảm bảo các điều sau:

- Cài đặt .Net framework 4.0
- Cài đặt chương trình RAndFriends phiên bản 3.2. Việc cài đặt nên được thực hiện theo các cấu hình mặc định để đảm bảo hai chương trình có thể tương tác tốt nhất có thể. Trong quá trình cài đặt này, chương trình R sẽ được cài đặt với phiên bản là 2.15.1. Do vậy kết quả của chương trình đối với phần chạy trên R sẽ tương ứng với phiên bản này

Sau khi khởi động chương trình, ta sẽ có màn hình sau:

The screenshot shows a software window titled "Hybrid Model" with four tabs: "Neuron Network", "Smoothing Model", "Hybrid Model", and "Logs". The "Hybrid Model" tab is active. It is divided into three main sections: "Init Neuron Network", "Algorithm Config", and "Training".

Init Neuron Network: Contains input fields for "Num Input Nodes" (12), "Num Hidden Nodes" (12), and "Num Output Nodes" (1). Below these is a text field for "I-O lags" containing the sequence "1,2,3,4,5,6,7,8,9,10,11". At the bottom are four buttons: "New", "Load", "Save", and "Clear".

Algorithm Config: Contains four input fields: "label13" (0.1), "label14" (1), "Max Epoches" (1000), and "Delta Errors" (0.001).

Training: Contains a "Choose Training File" text box. Below it are fields for "Num Rows" (label7), "Num Columns" (label8), "Max Value", and "Min Value". There are also "Row" and "Column" fields followed by a "to" label and another empty field. At the bottom left, under "Algorithms", are two radio buttons: "Back Propagation" (selected) and "RPROP". To the right are two large buttons: "Train" and "Plot".

Hình B.1: Giao diện khởi động chương trình lai

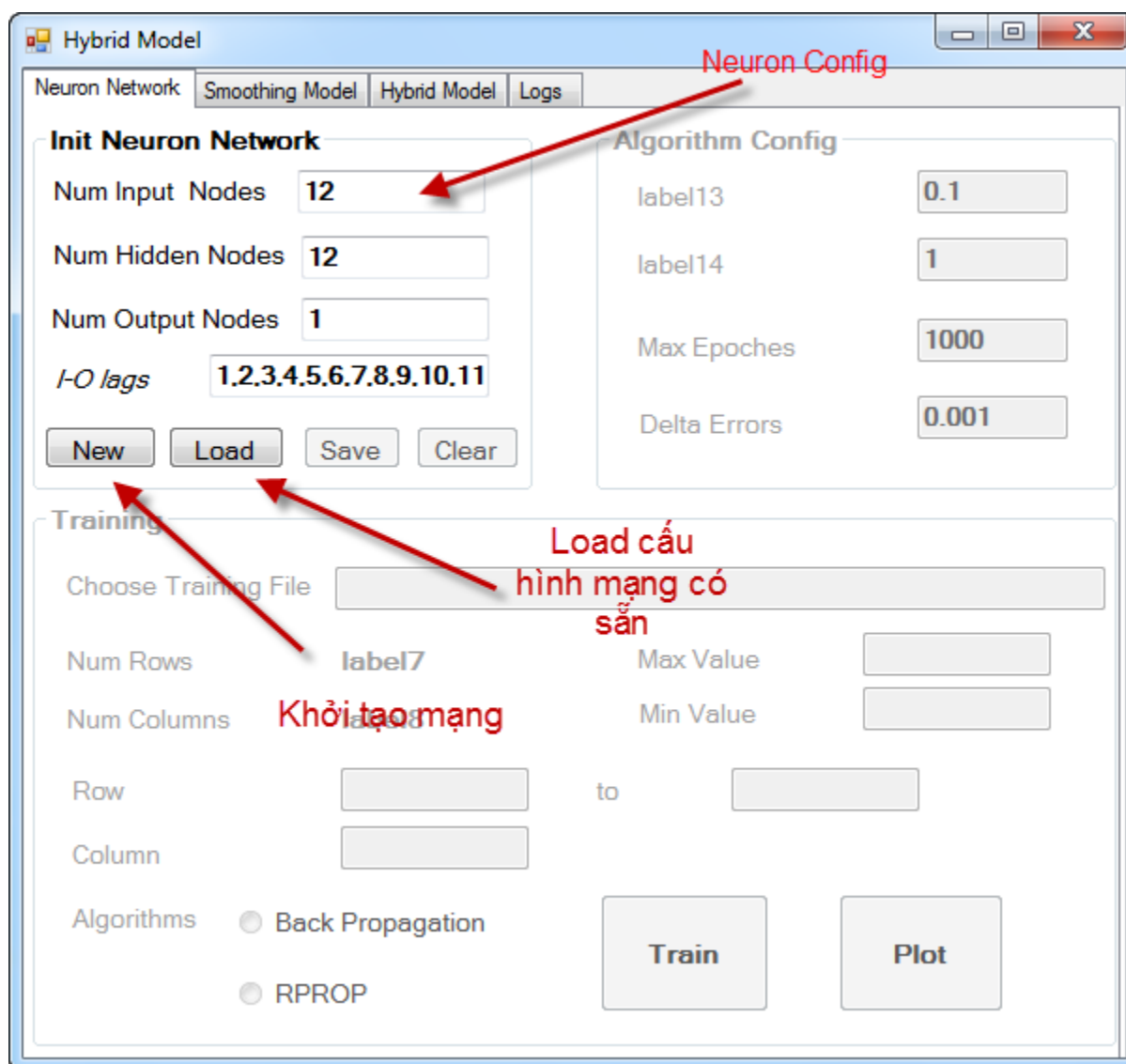
B.1.2. Thực hiện huấn luyện

Để thực hiện việc huấn luyện, ta tiến hành ba bước sau

- Huấn luyện mạng Neuron
- Huấn luyện mô hình làm trơn theo hàm mũ
- Huấn luyện mô hình lai

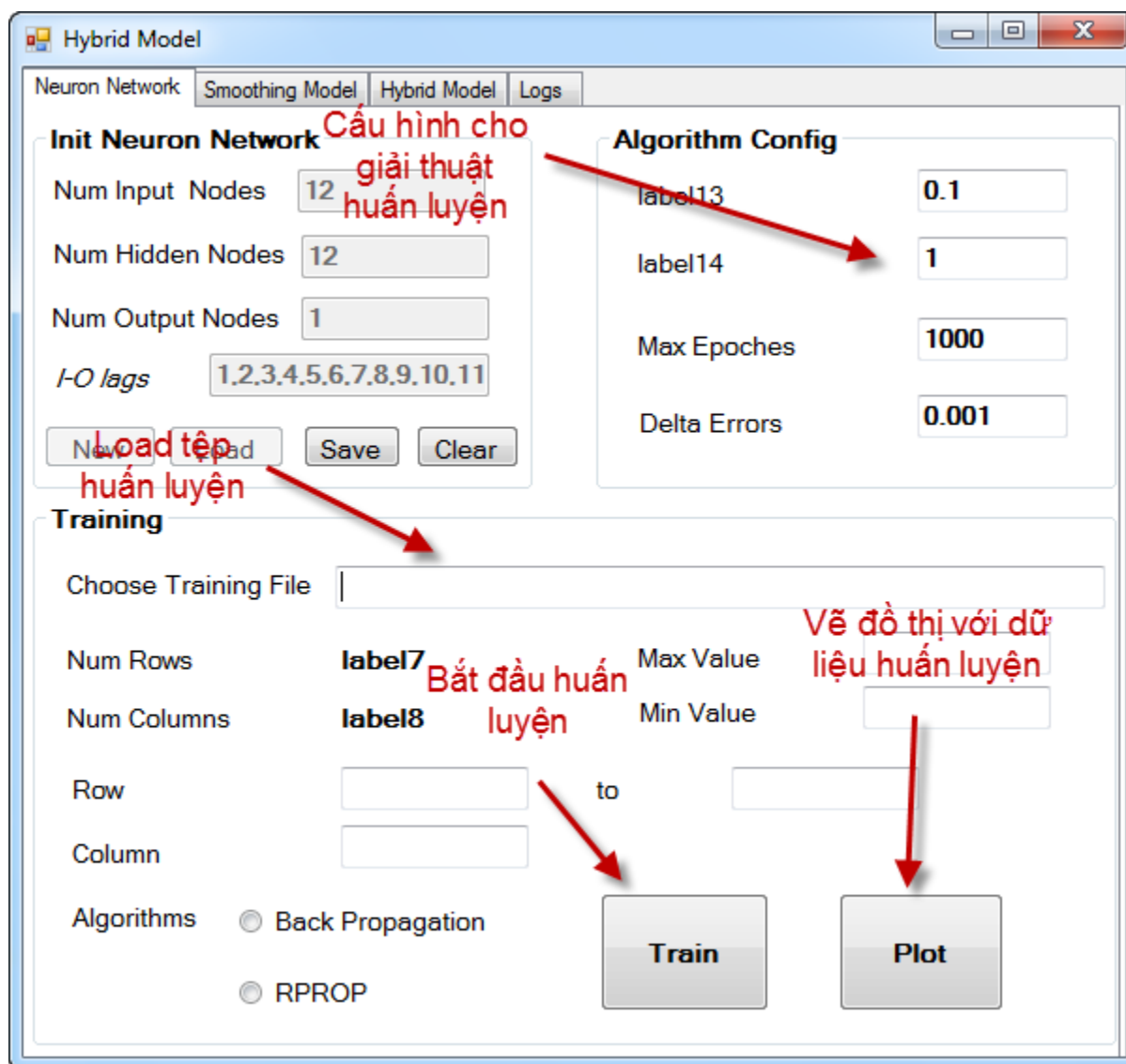
Huấn luyện mạng Neuron

- Khởi tạo thông số mạng hoặc load một file cấu hình mạng có sẵn



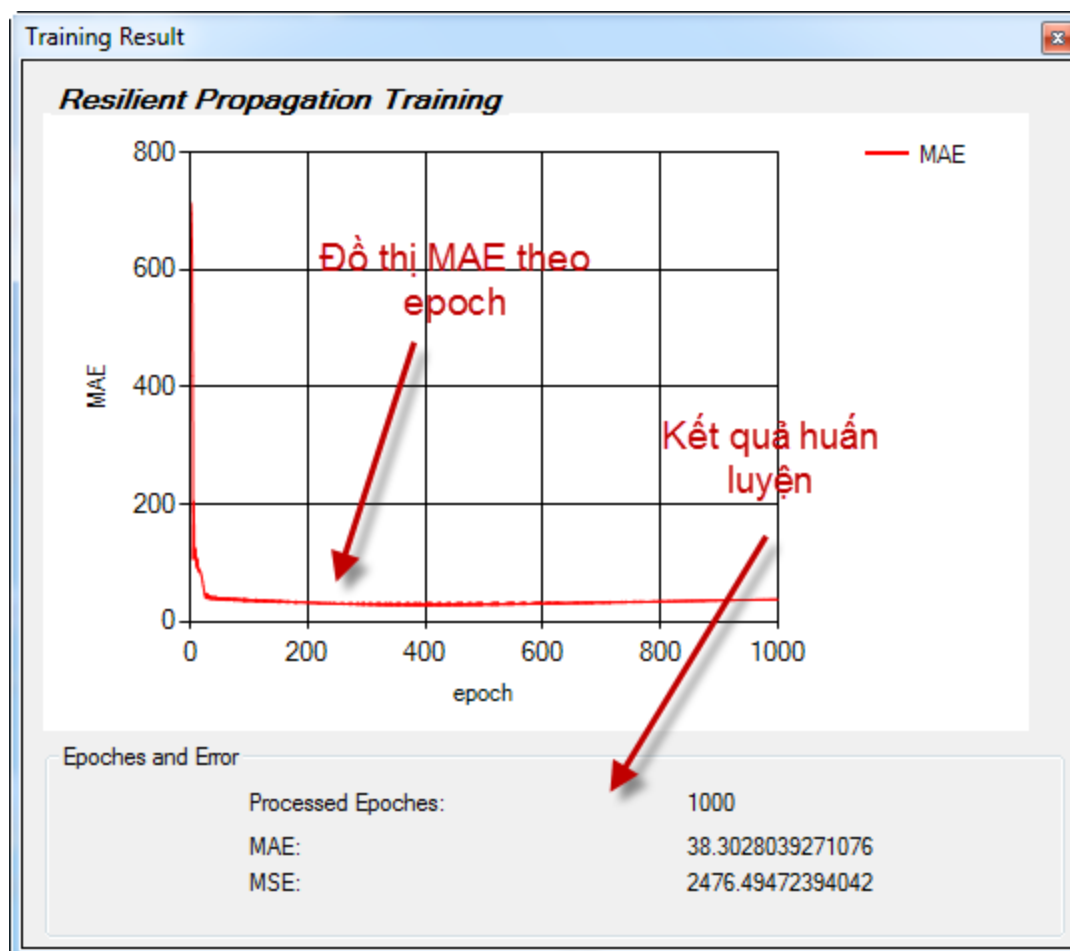
Hình B.2: Giao diện cấu hình mạng neuron nhân tạo

- Load file dữ liệu huấn luyện, chọn giải thuật và cấu hình các thông số cho giải thuật



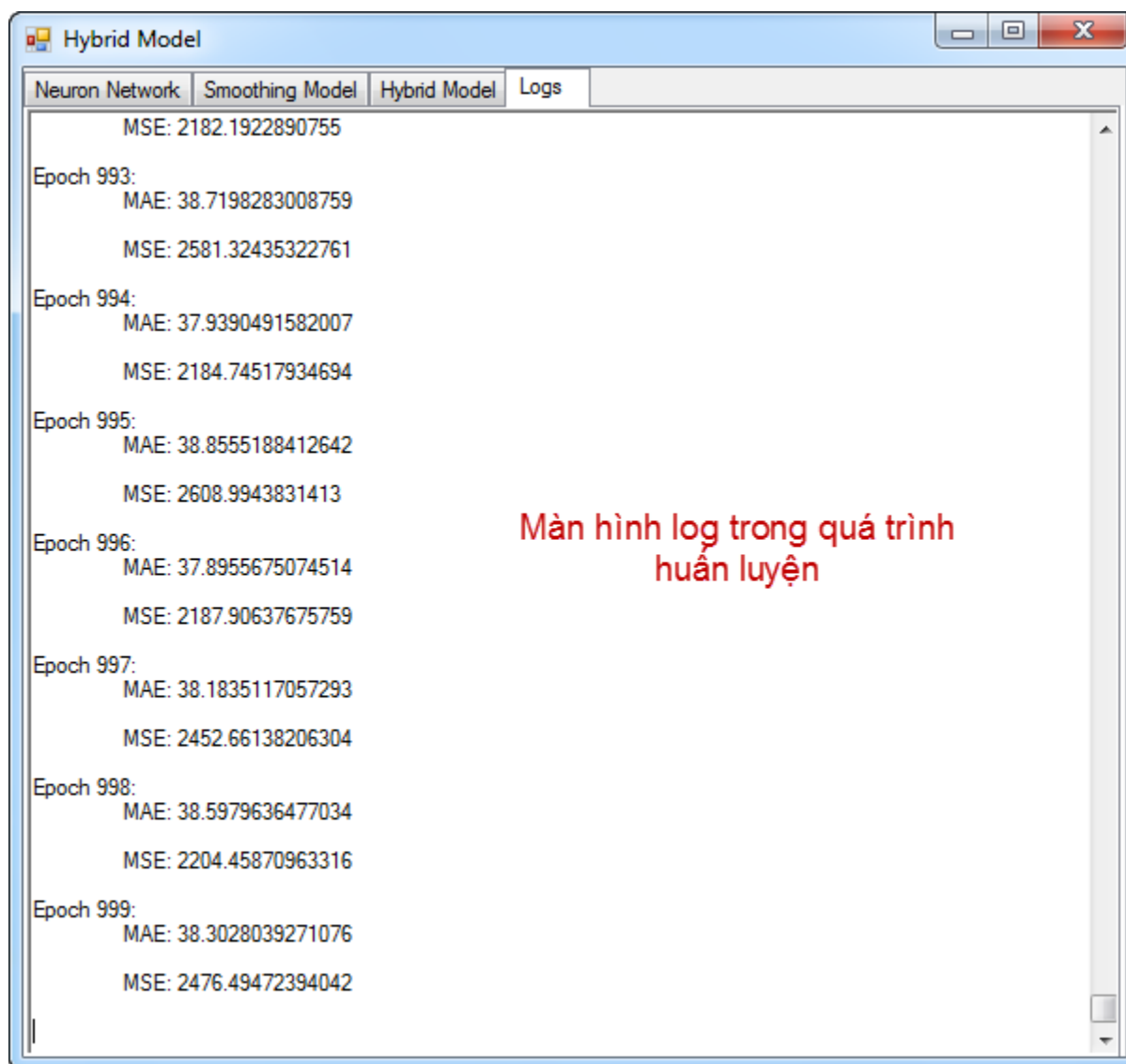
Hình B.3: Giao diện load dữ liệu huấn luyện

- Đây là màn hình kết quả sau khi thực hiện huấn luyện đối với mạng Neuron



Hình B.4: Giao diện kết quả huấn luyện mạng neuron nhân tạo

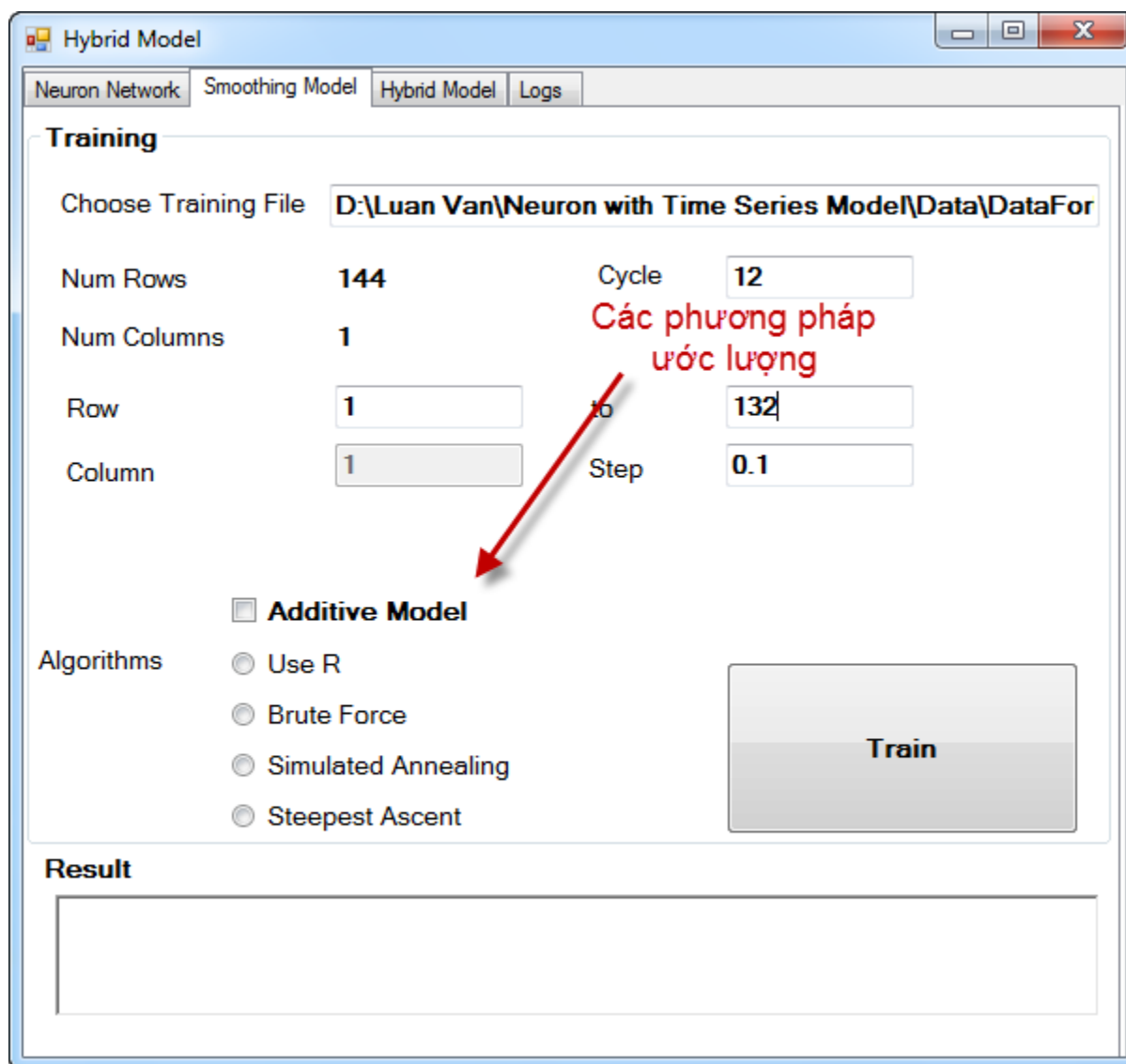
- Màn hình Log trong quá trình huấn luyện



Hình B.5: Giao diện nhật kí huấn luyện

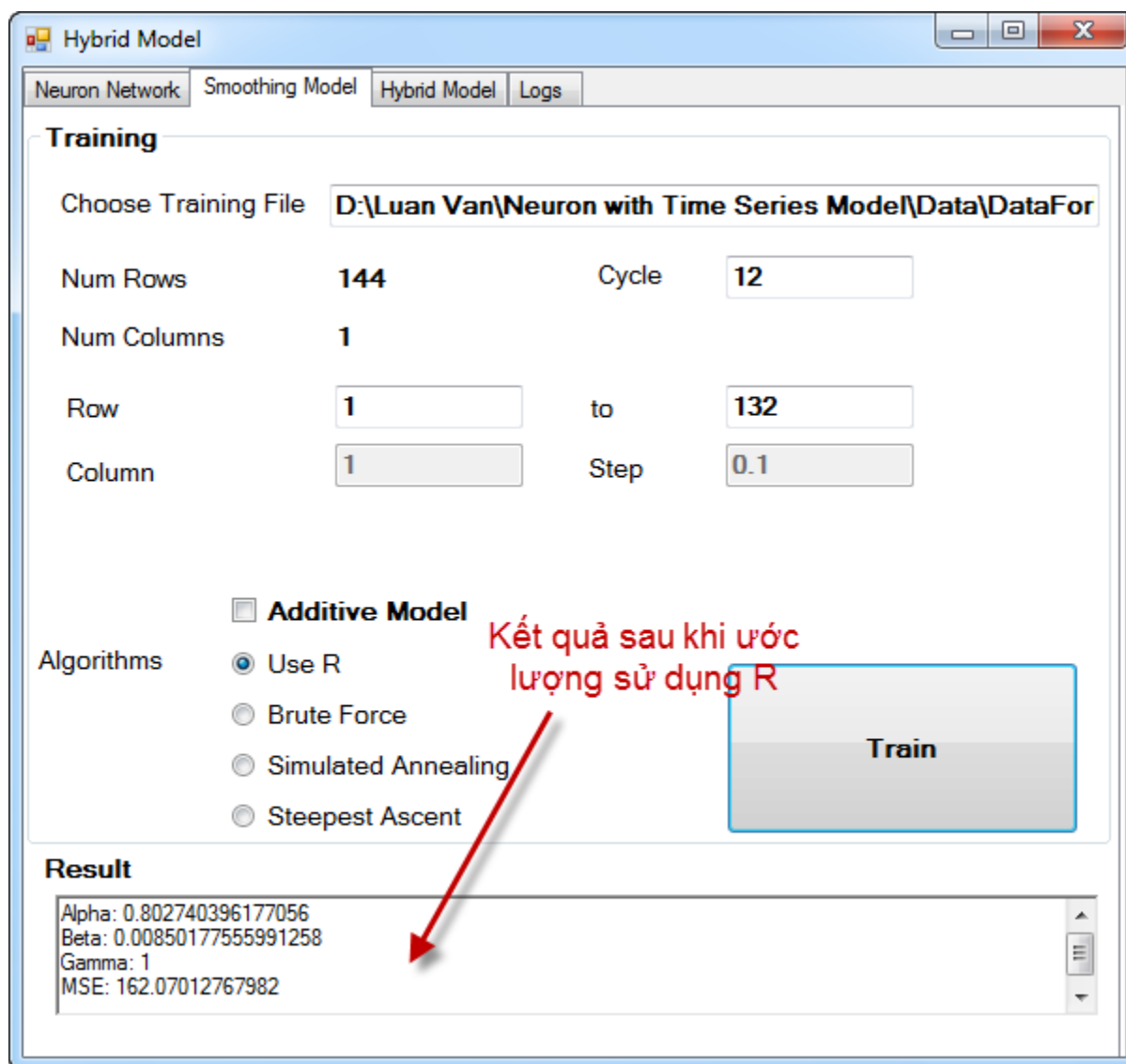
Huấn luyện mô hình làm trơn theo hàm mũ

- Load file dữ liệu (nếu trước đó đã thực hiện load file dữ liệu ở bước huấn luyện mạng Neuron thì ở bước này ta không cần load lại, chỉ cần cấu hình các thống số)
- Chọn phương pháp ước lượng (có 4 phương pháp chính: sử dụng R, Brute Force, Simulated Annealing, Steepest Ascent, ở mỗi phương pháp này ta có thể chọn mô hình đi kèm là mô hình nhân hay mô hình cộng). Màn hình cài đặt phương pháp ước lượng như sau



Hình B.6: Giao diện mô hình làm trơn theo hàm mũ

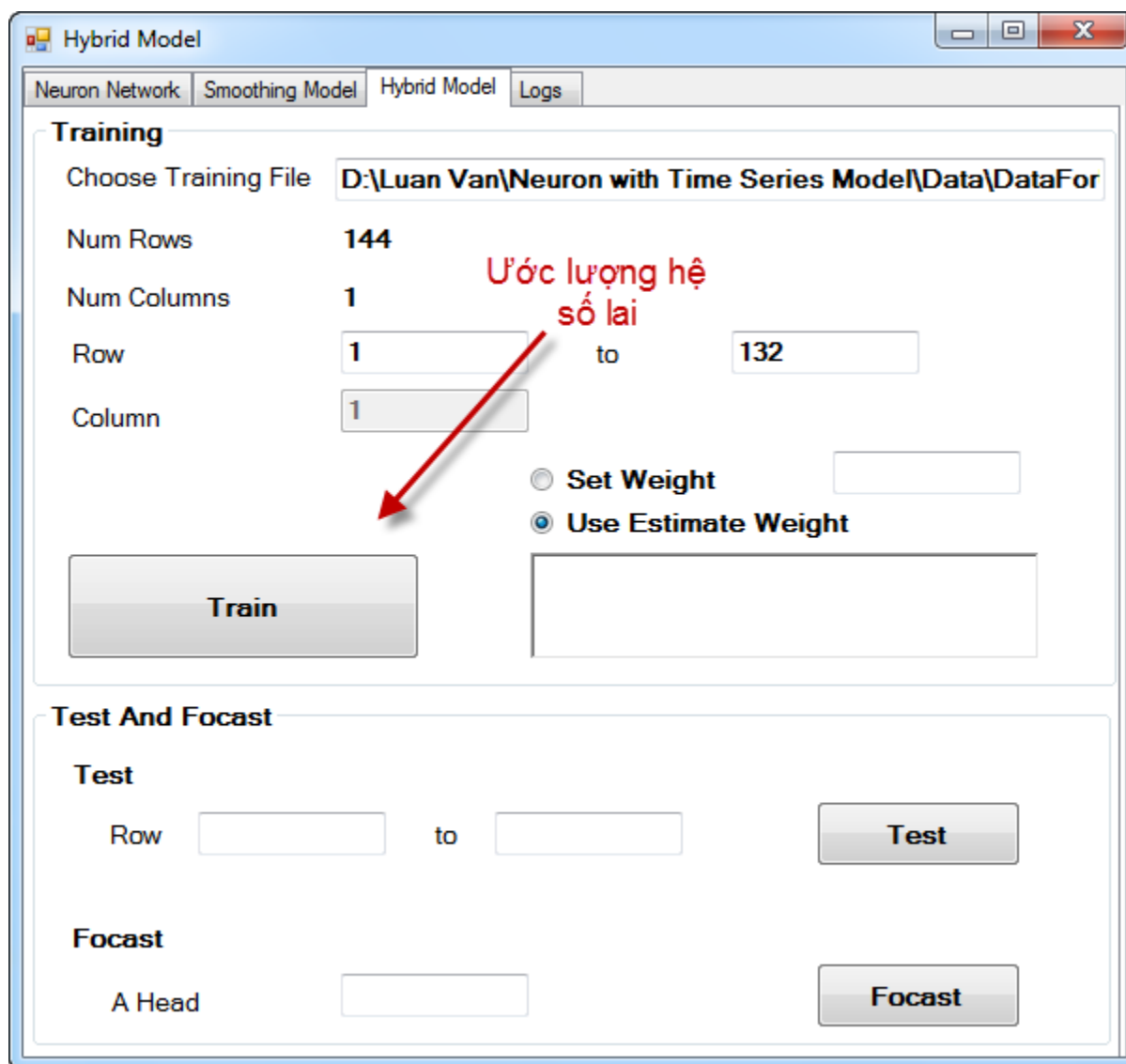
- Và đây là kết quả sau khi ước lượng



Hình B.7: Giao diện kết quả huấn luyện mô hình làm trơn theo hàm mũ

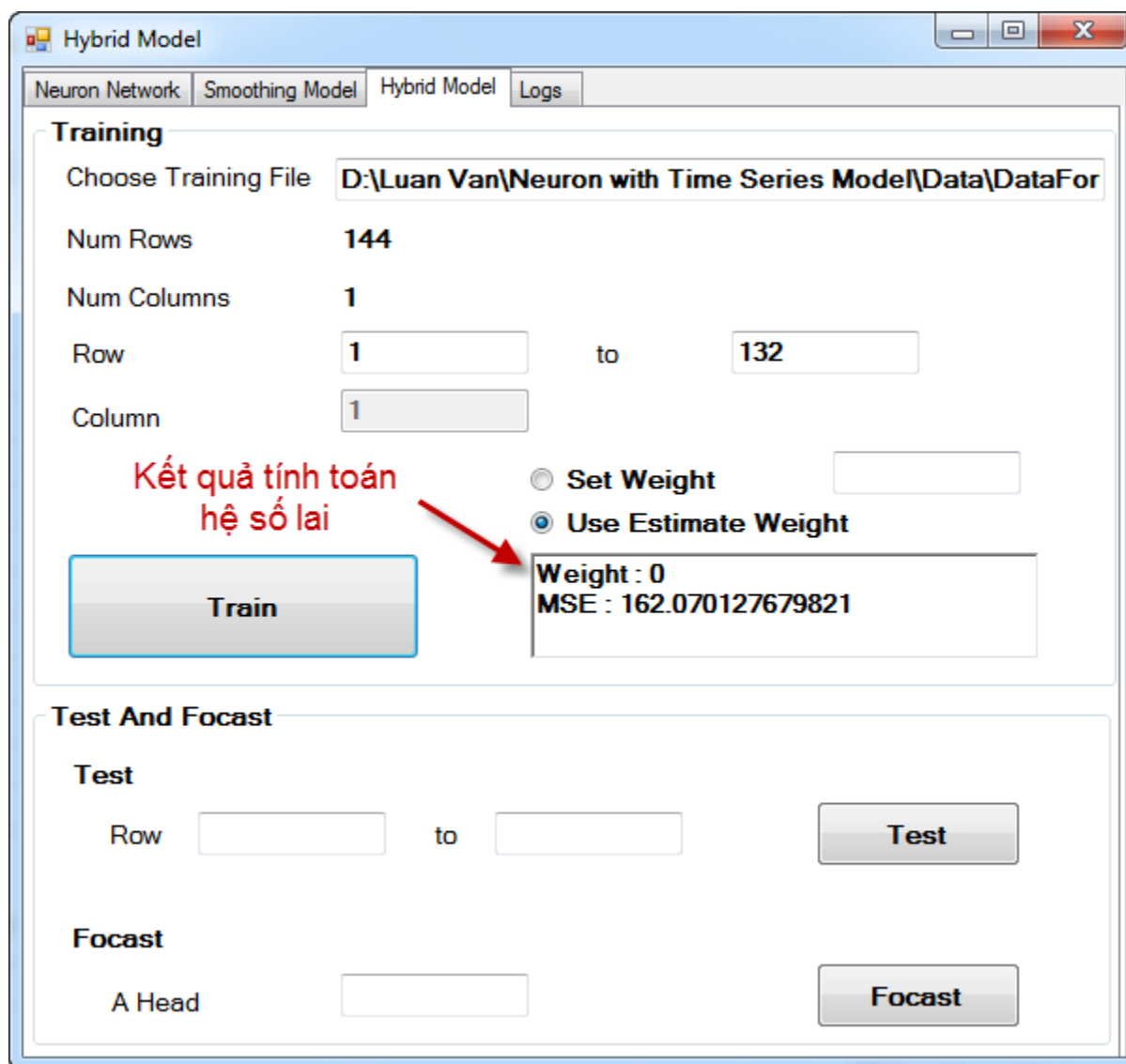
Huấn luyện mô hình lại

- Tương tự phần này phần load dữ liệu đã được định bởi hai phần kia (các bạn cũng có thể đổi file này, tuy nhiên kết quả có thể sẽ không được như mong đợi)
- Tiến hành ước lượng trọng số lại. Màn hình chương trình tại bước nay



Hình B.9: Giao diện mô hình lai

- Và đây là kết quả sau khi thực hiện ước lượng



Hình B.10: Kết quả huấn luyện mô hình lai

- Tuy nhiên, các bạn có thể định trọng số này tại khung Set Weight. Tuy nhiên, MSE sẽ không là tốt nhất đối với tập huấn luyện.

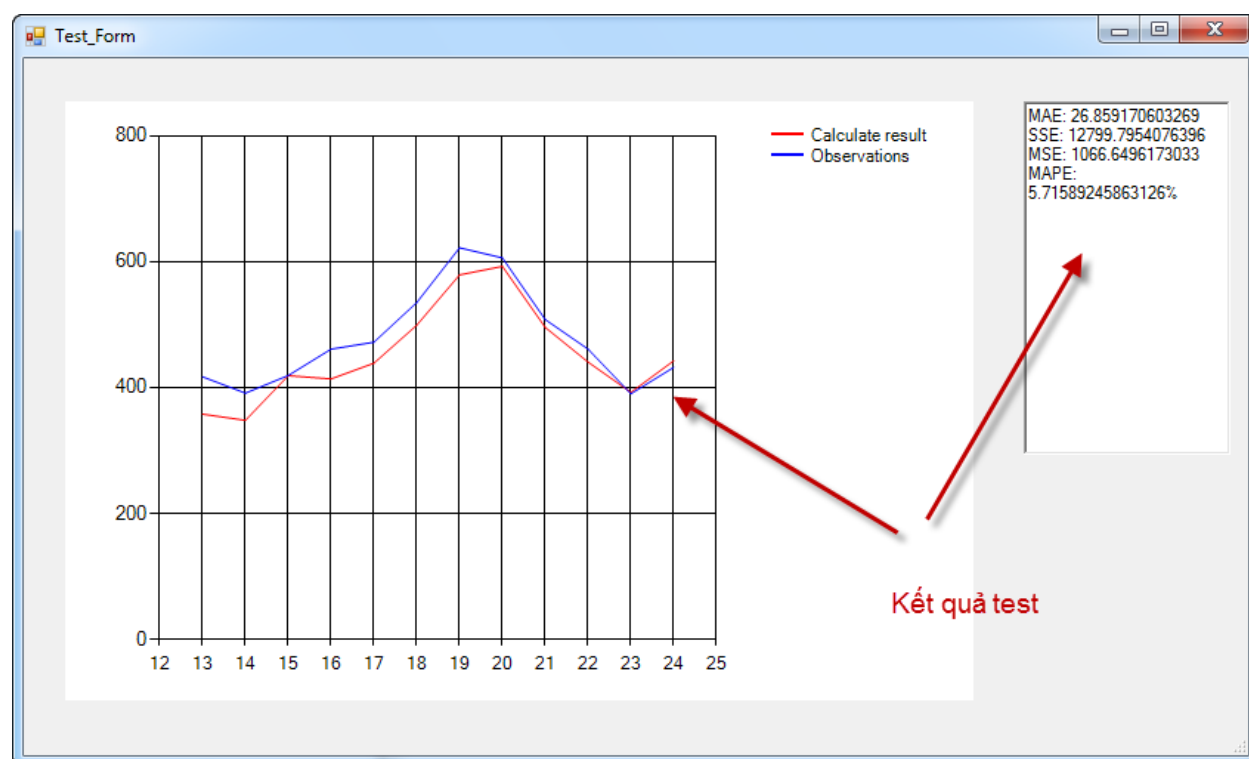
B.1.3. Thực hiện kiểm tra

Sau khi thực hiện huấn luyện và có nhu cầu đánh giá kết quả huấn luyện, các bạn cần phải cài đặt phạm vi thực hiện kiểm tra.

The screenshot shows the 'Hybrid Model' software window with four tabs: 'Neuron Network', 'Smoothing Model', 'Hybrid Model' (selected), and 'Logs'. The 'Training' section includes a 'Choose Training File' field with the path 'D:\Luan Van\Neuron with Time Series Model\Data\DataFor', 'Num Rows' set to 144, 'Num Columns' set to 1, 'Row' range from 1 to 132, and 'Column' set to 1. There are radio buttons for 'Set Weight' and 'Use Estimate Weight' (selected). A 'Train' button is present. A box displays 'Weight : 0' and 'MSE : 162.070127679821'. The 'Test And Focast' section has a 'Test' subsection with 'Row' range from 133 to 144 and a 'Test' button. A red arrow points from the text 'Test mô hình sau khi huấn luyện' to the 'Test' button. The 'Focast' subsection has an 'A Head' field and a 'Focast' button.

Hình B.11: Giao diện test mô hình

Kết quả kiểm tra sẽ như sau



Hình B.12: Giao diện kết quả test mô hình

B.1.4. Thực hiện dự báo

Sau khi huấn luyện, nếu có như cầu dự báo, bạn cần phải cài đặt số lượng thời đoạn muốn dự báo như hình sau

The screenshot displays the 'Hybrid Model' application window. It features a tabbed interface with 'Neuron Network', 'Smoothing Model', 'Hybrid Model', and 'Logs'. The 'Hybrid Model' tab is active, showing a 'Training' section with fields for 'Choose Training File' (D:\Luan Van\Neuron with Time Series Model\Data\DataFor), 'Num Rows' (144), 'Num Columns' (1), 'Row' (1 to 132), and 'Column' (1). There are radio buttons for 'Set Weight' and 'Use Estimate Weight' (selected). A 'Train' button is present. Below the training section, the 'Test And Focast' section includes 'Test' (Row 133 to 144) and 'Focast' (A Head 24) buttons. A red arrow points from the 'Test' button to the 'Focast' button. The text 'Dự báo' is written in red above the 'Focast' button. A status box shows 'Weight : 0' and 'MSE : 162.070127679821'.

Hybrid Model

Neuron Network Smoothing Model Hybrid Model Logs

Training

Choose Training File D:\Luan Van\Neuron with Time Series Model\Data\DataFor

Num Rows 144

Num Columns 1

Row 1 to 132

Column 1

☐ Set Weight

☒ Use Estimate Weight

Train

Weight : 0
MSE : 162.070127679821

Test And Focast

Test

Row 133 to 144

Test

Focast

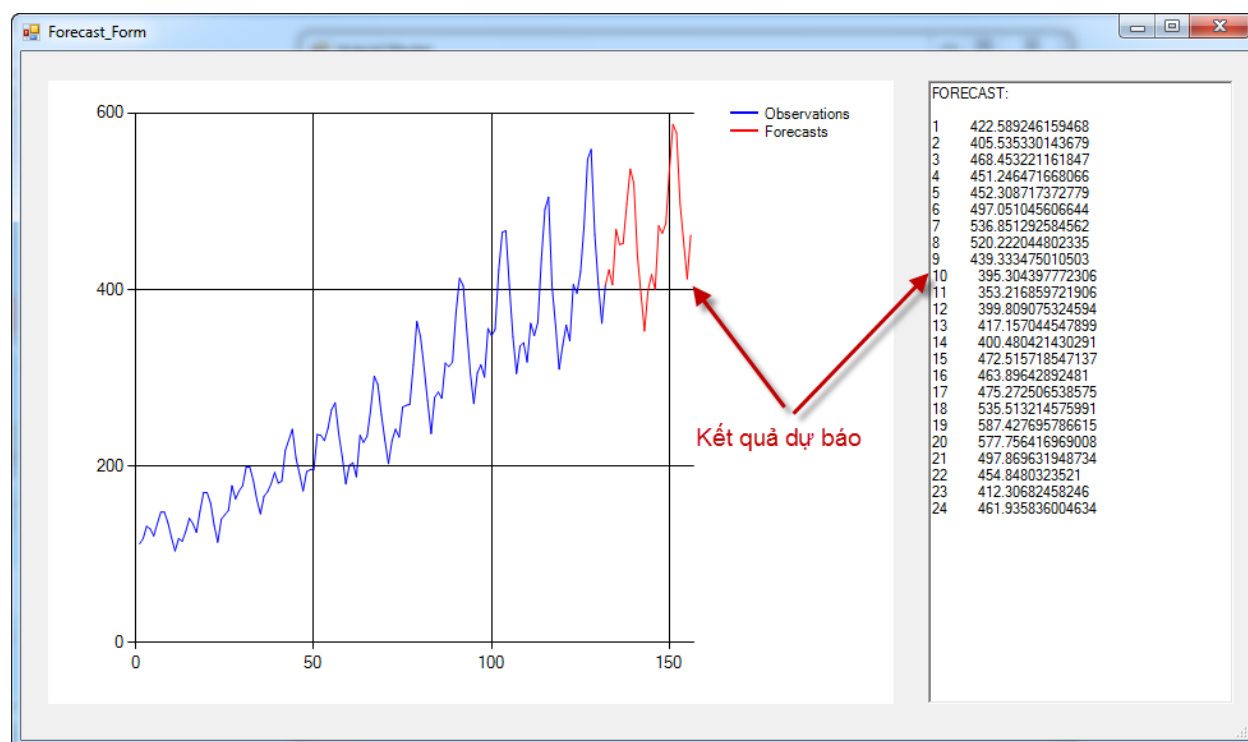
A Head 24

Focast

Dự báo

Hình B.14: Giao diện dự báo mô hình

Kết quả sẽ như sau



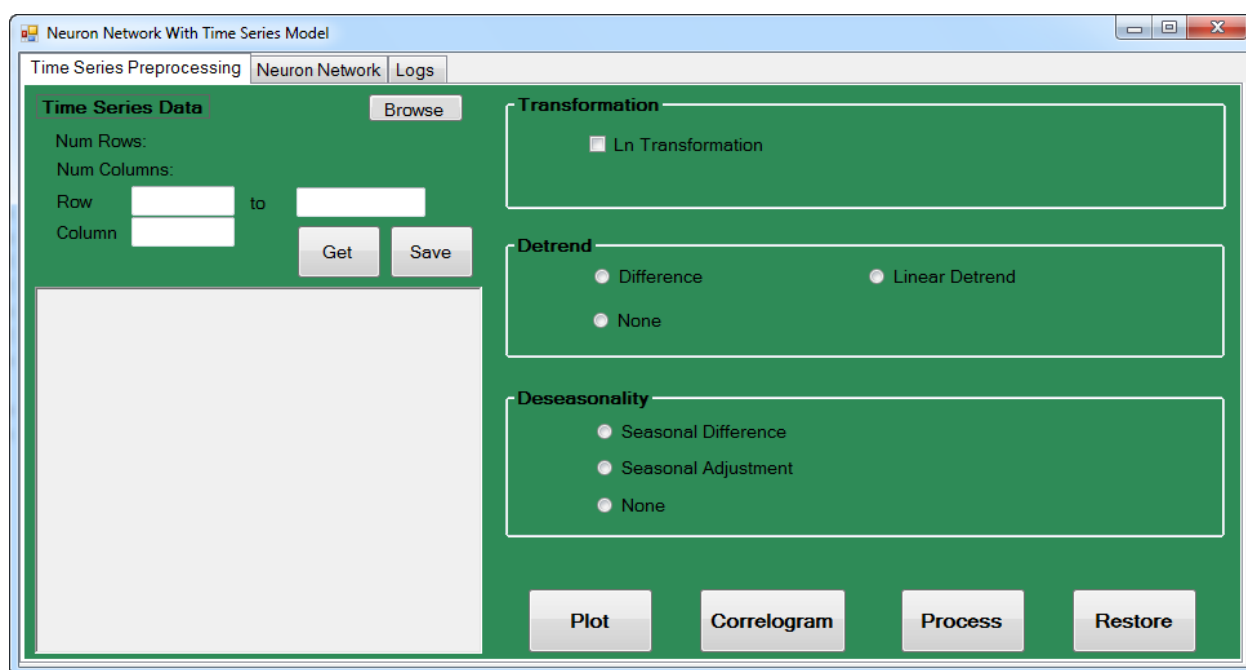
Hình B.15: Giao diện kết quả dự báo

B.2. Chương trình hiện thực mô hình khử mùa

B.2.1. Môi trường cài đặt và sử dụng

Chương trình này toàn bộ do chúng tôi hiện thực và thiết kế bằng ngôn ngữ C# trên nền .Net framework 4.0. Vì thế, để có thể chạy được chương trình, các bạn chỉ cần cài đặt phiên bản .Net framework 4.0

Giao diện khởi động chương trình như sau:



Hình B.16: Giao diện chương trình khử mùa, khử xu hướng

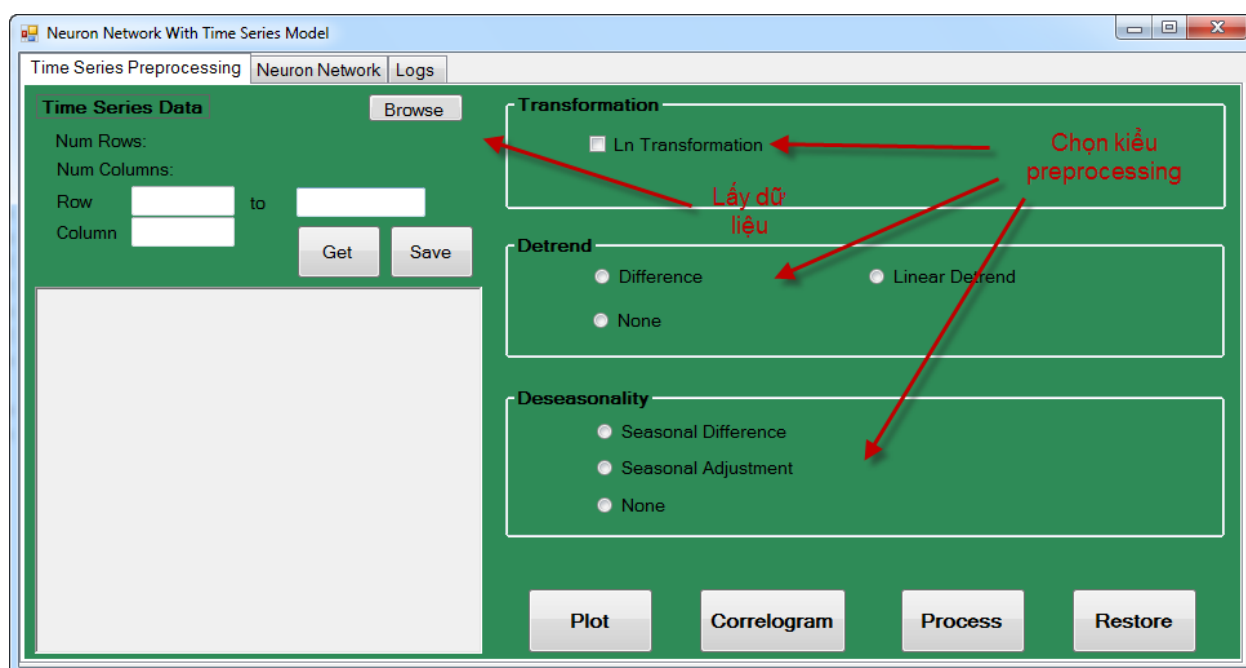
B.2.2. Thực hiện huấn luyện

Để thực hiện huấn luyện ta tiến hành các bước sau:

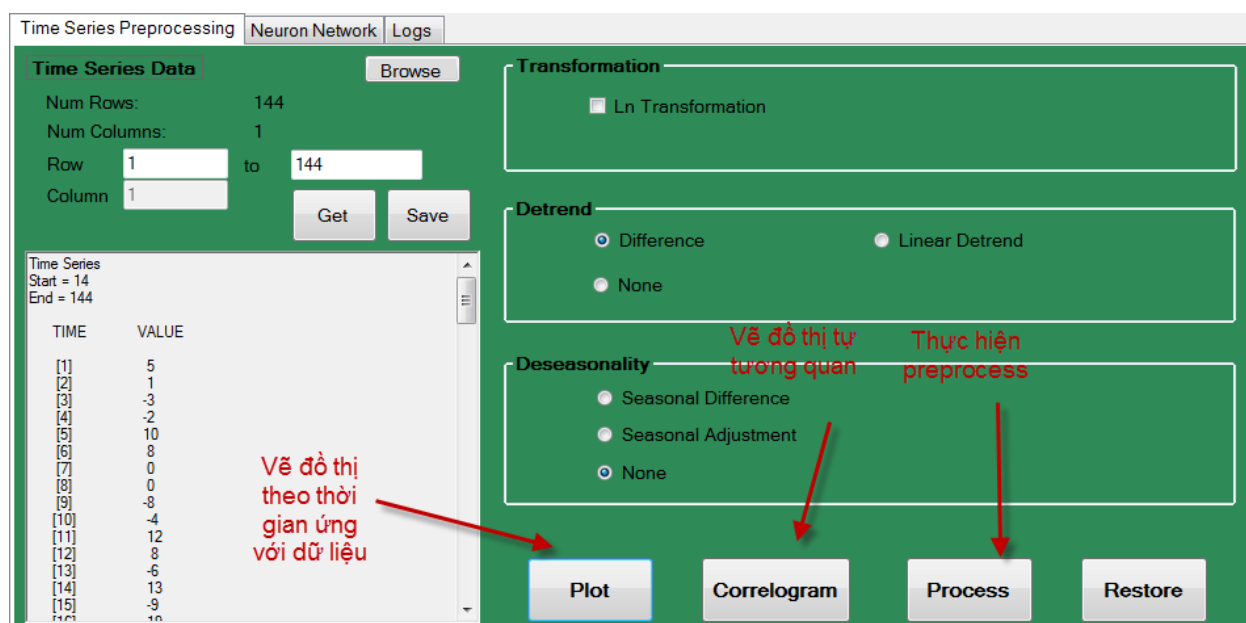
- Kiểm tra dữ liệu để chọn phương pháp khử mùa và xu hướng hợp lý nhất
- Thực hiện huấn luyện mạng Neuron với các phương pháp khử đã chọn

Kiểm tra dữ liệu: để thực hiện việc này, chúng tôi cung cấp cho người sử dụng tab preprocessing. Ở tab này, người sử dụng có thể load tập dữ liệu cho huấn luyện, thực hiện các phép khử hỗ trợ sau đó vẽ biểu đồ và đồ thị tương quan. Từ đó, người sử dụng có thể chọn được phương pháp khử hợp lý nhất.

Giao diện vừa mới khởi động:

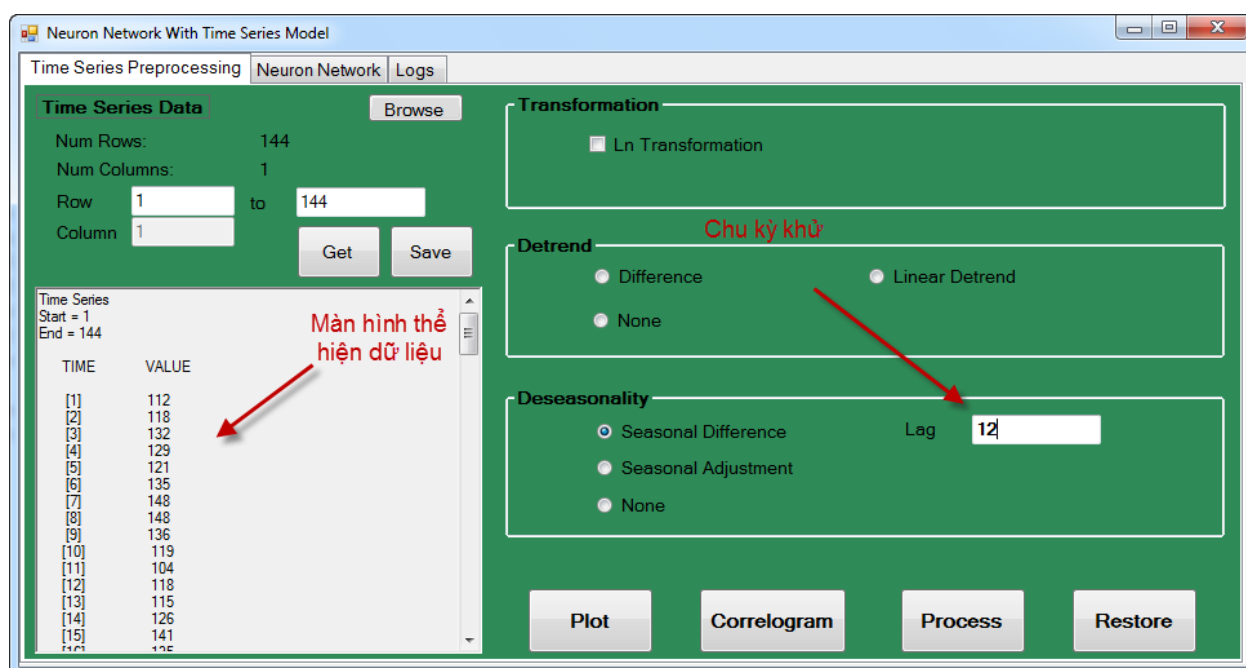


Hình B.17: Giao diện module khử mùa, khử xu hướng



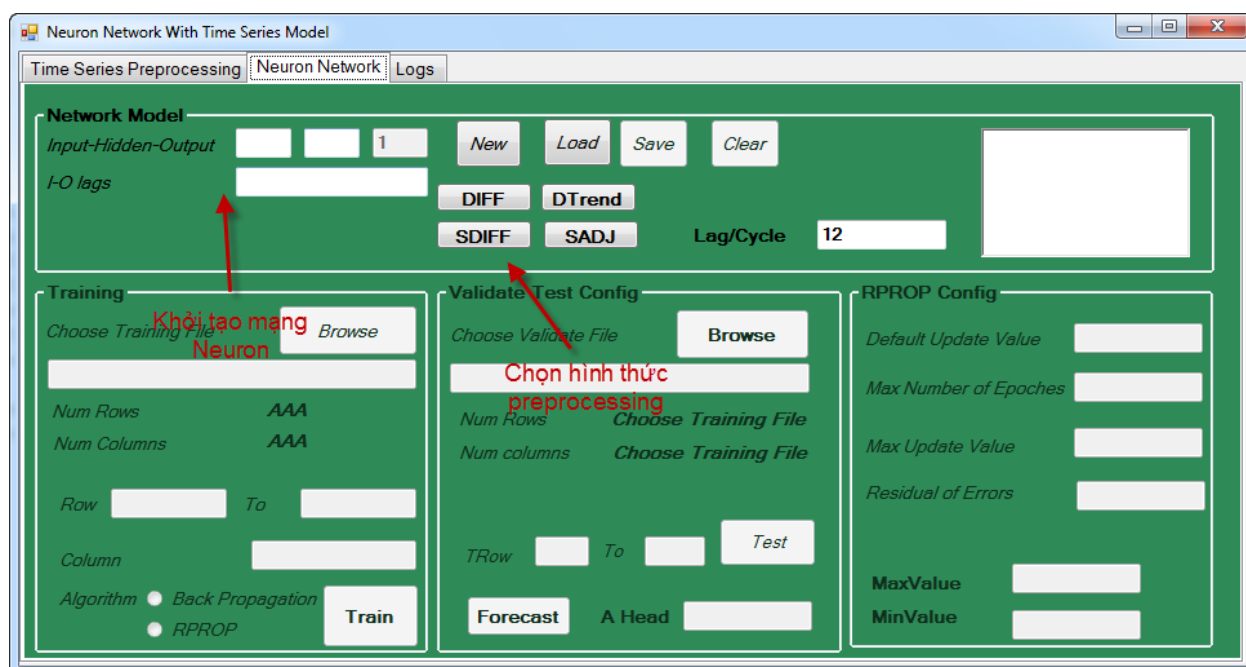
Hình B.18: Giao diện mô hình khử mùa, khử xu hướng load dữ liệu

Tiến hành load dữ liệu và chọn phương thức khử



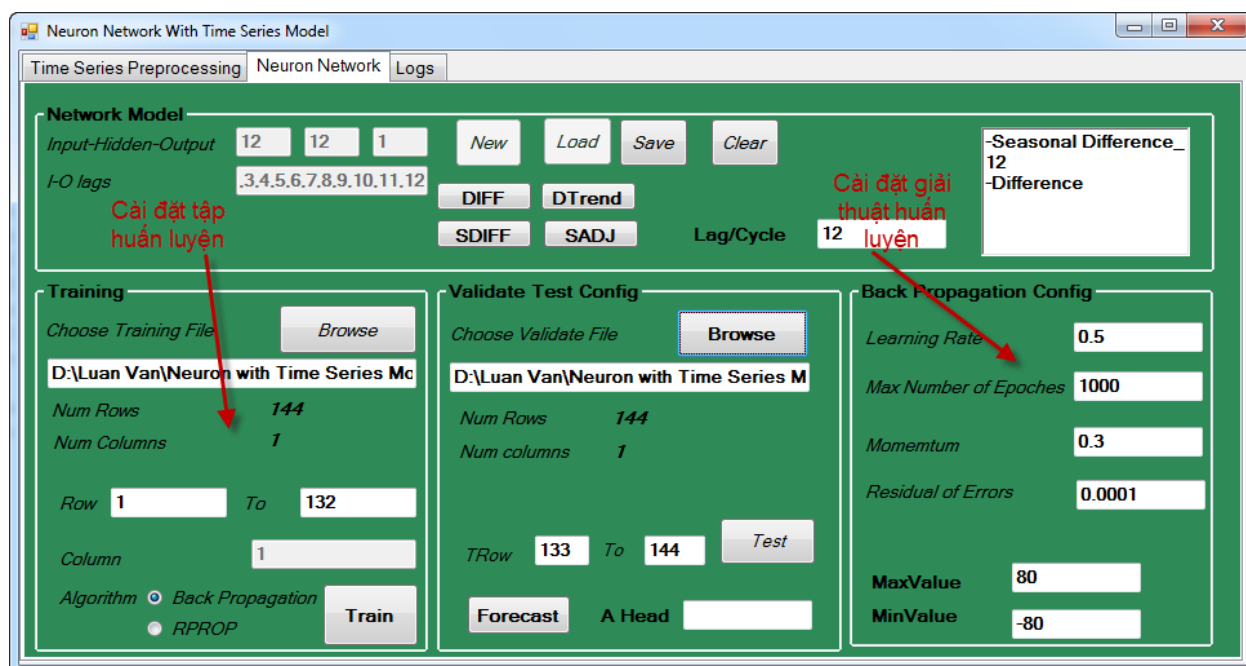
Hình B.19: Giao diện khử mùa, khử xu hướng với cách cấu hình khử mùa

Thực hiện huấn luyện mạng: khởi tạo thông số và chọn phương pháp khử



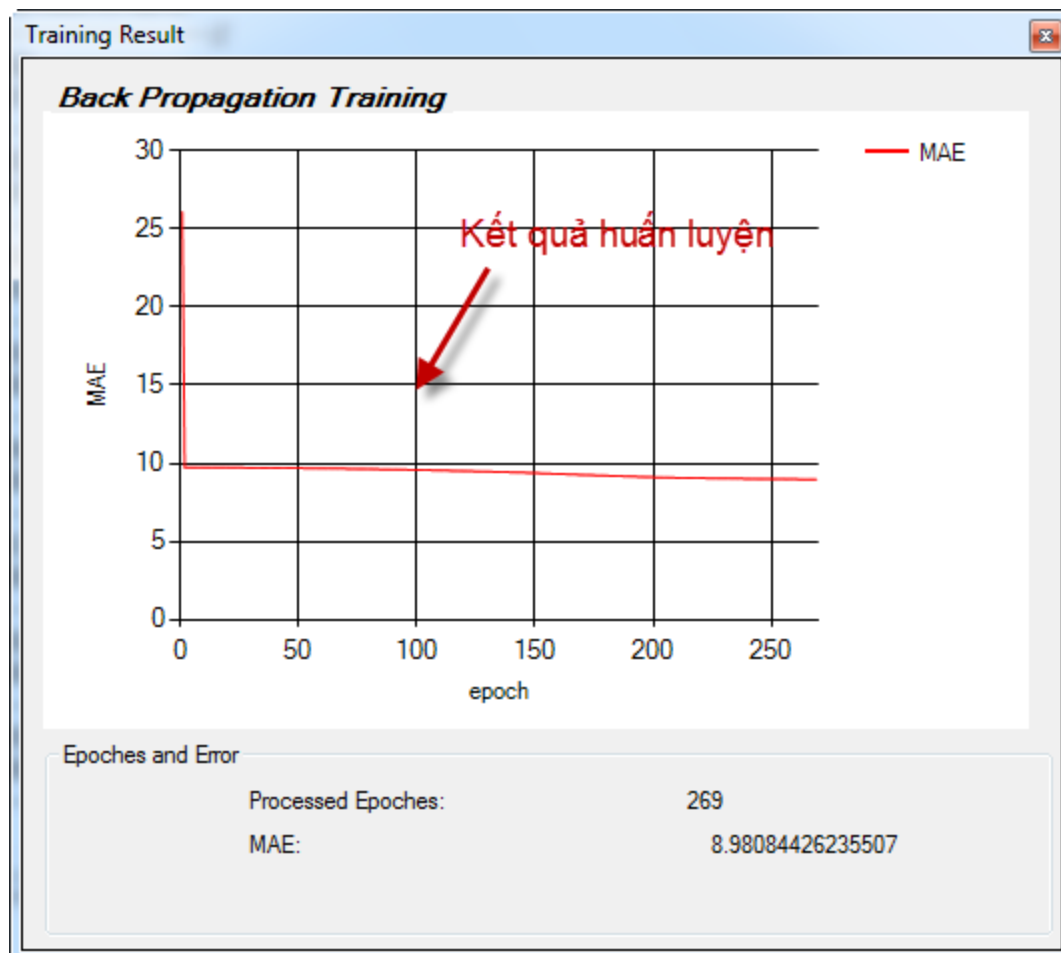
Hình B.20: Giao diện mạng neuron nhân tạo

Load dữ liệu huấn luyện và cài đặt thông số cho giải thuật



Hình B.21: Giao diện cài đặt huấn luyện mạng neuron nhân tạo

Tiến hành huấn luyện, ta sẽ được kết quả như sau:

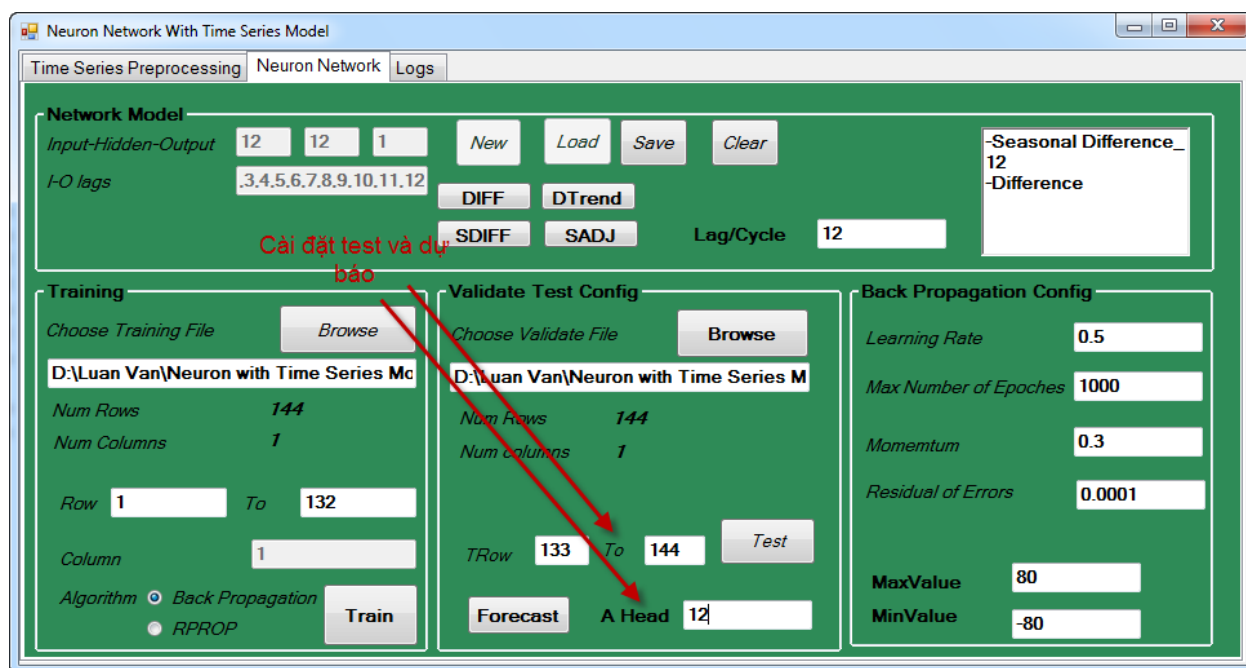


Hình B.22: Giao diện kết quả huấn luyện mạng

B.2.3. Thực hiện kiểm tra và dự báo

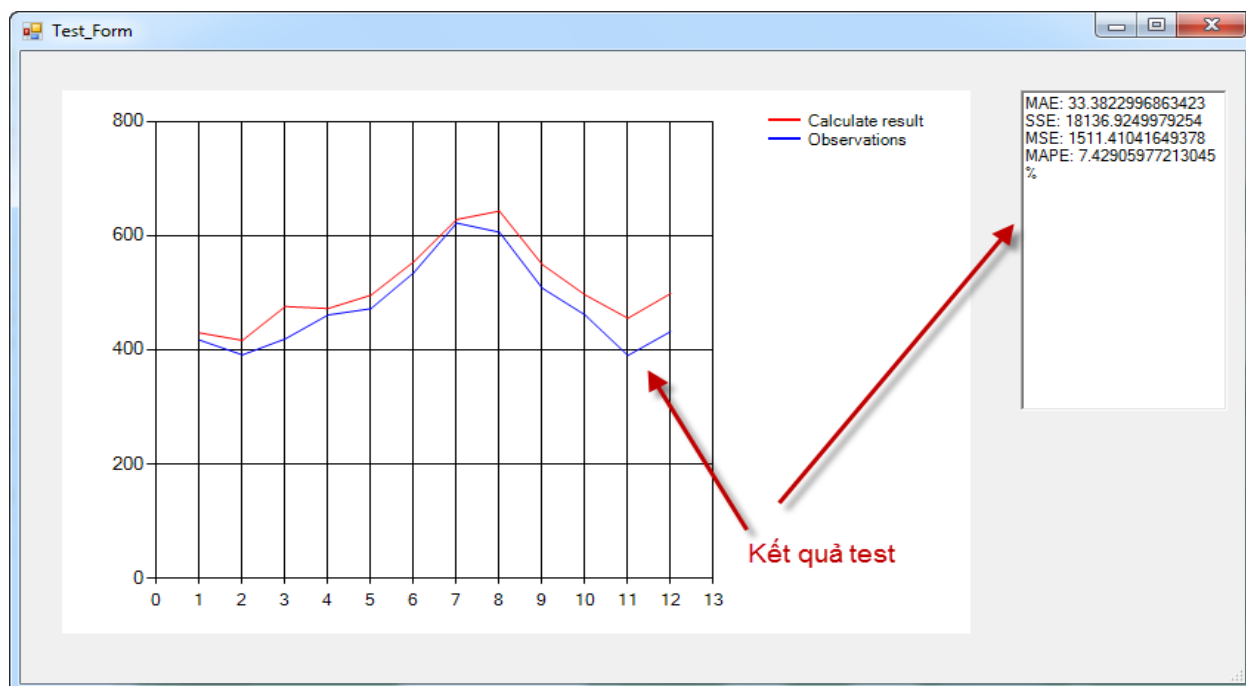
Cài đặt thông số kiểm tra và dự báo

Ứng dụng mạng Neuron nhân tạo dự báo dữ liệu chuỗi thời gian có tính xu hướng và tính mùa



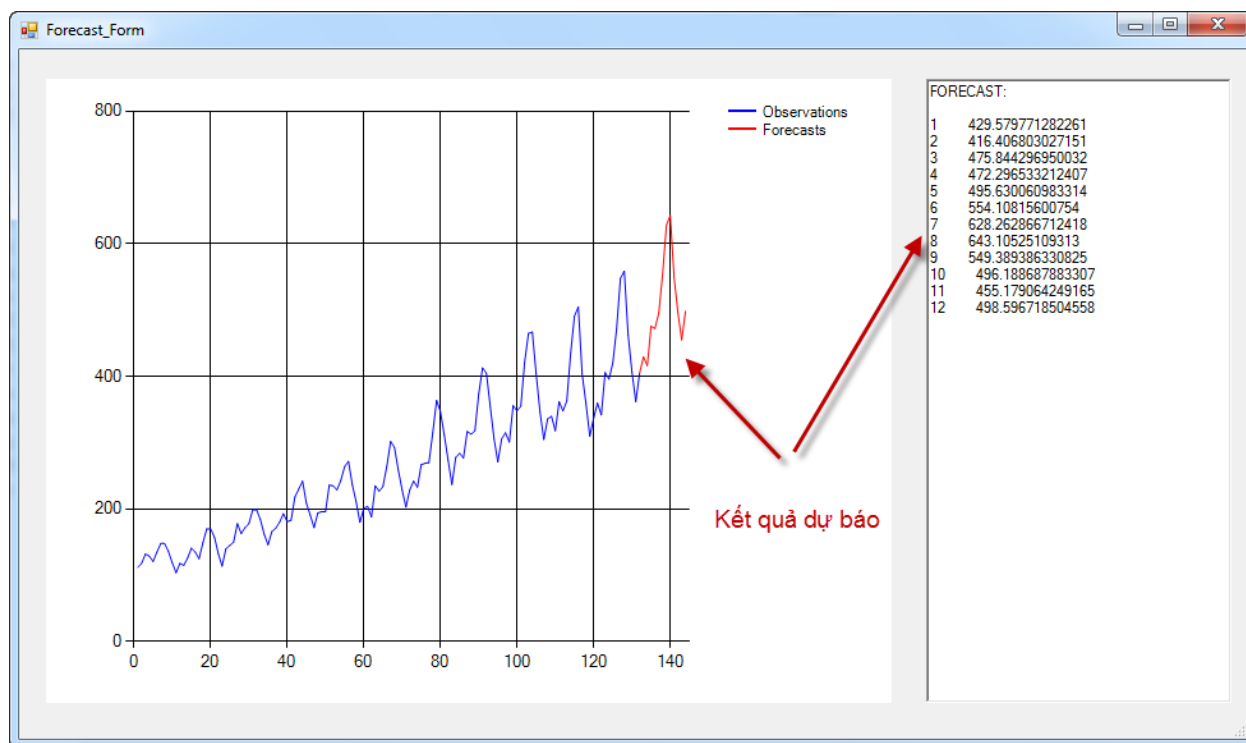
Hình B.23: Giao diện cài đặt test và dự báo

Kết quả kiểm tra như sau:



Hình B.24: Giao diện kết quả test mô hình

Kết quả dự báo như sau:



Hình B.25: Giao diện kết quả dự báo