Tạp chí Tin học và Điều khiển học, T.20, S.2 (2004), 111–120

VỀ MỘT PHƯƠNG PHÁP DỰ BÁO DỮ LIỆU SỬ DỤNG MẠNG NƠRON

LÊ HẢI KHÔI, TRẦN ĐỨC MINH

Viện Công nghệ thông tin

Abstract. One of the most difficulty problem that every organization, enterprise, business must deal with today is how to forecast the demands of using, spending of the market with their products. In this paper, we describe a method to forecast based on the neural network approach. We also deal with the steps for applying neural network in solving the data forecasting problems.

Tóm tắt. Một trong những vấn đề nan giải mà các cơ quan, xí nghiệp, doanh nghiệp sản xuất, kinh doanh gặp phải hiện nay là làm sao dự báo được nhu cầu sử dụng, tiêu dùng của thị trường đối với các sản phẩm của mình. Trong bài này mô tả sơ lược phương pháp dự báo dữ liệu dựa trên cách tiếp cận sử dụng mạng nơron, các bước cần thực hiện khi ứng dụng mạng nơron để giải quyết bài toán dự báo dữ liệu.

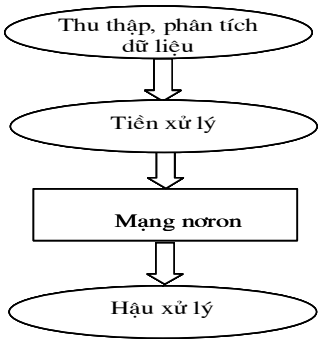
1. GIỚI THIỆU

Chúng ta biết rằng, mạng nơron, đôi khi được xem như là các mô hình liên kết (connectionist models), chúng là các mô hình phân bố song song (parallel-distributed models) có khả năng ánh xạ từ một tập dữ liệu vào đến một tập dữ liệu ra mà không yêu cầu các tập dữ liệu này phải đầy đủ. Trong quá trình phát triển, mạng nơron đã được ứng dụng thành công trong rất nhiều lĩnh vực như phân loại, giảm nhiễu, dự báo.

Mạng nơron đã được thực tiễn chứng minh là khá mạnh và hiệu quả trong các bài toán dự báo, phân tích dữ liệu. Chúng có khả năng biểu diễn các ánh xạ phi tuyến giữa đầu vào và đầu ra, và đôi khi được coi như là các "bộ xấp xỉ đa năng". Việc ứng dụng của mạng nơron đối với việc dự báo khuynh hướng thay đổi của các dữ liệu tác nghiệp trong các cơ quan, tổ chức kinh tế, xã hội... là một nhu cầu thực tế. Nếu có thể dự báo được khuynh hướng thay đổi của dữ liệu với một độ tin cậy nhất định, các nhà lãnh đạo có thể đưa ra được các quyết sách đúng đắn cho cơ quan, tổ chức của mình (xem chẳng hạn [2]).

Mặc dù vậy, dự báo dữ liệu là một bài toán rất phức tạp, cả về số lượng dữ liệu cần quan tâm cũng như độ chính xác của dữ liệu dự báo ([7]). Do vậy, để có thể chọn được mô hình phù hợp cho các bài toán dự báo dữ liệu là một việc rất khó khăn (chỉ có thể bằng phương pháp thử-sai). Tuy nhiên, việc sử dụng thuật toán huấn luyện lan truyền ngược đã được thực tế chứng tỏ là một công cụ tốt áp dụng cho các bài toán trong lĩnh vực dự báo dữ liệu.

Có thể coi bài toán dự báo dữ liệu sử dụng mạng nơron như là một dạng của bài toán xử lý dữ liệu nói chung. Việc xử lý dữ liệu bắt đầu bằng việc thu thập và phân tích dữ liệu, sau đó là bước tiền xử lý. Dữ liệu sau khi qua bước tiền xử lý được đưa vào mạng nơron. Cuối cùng, dữ liệu đầu ra của mạng nơron qua bước hậu xử lý, bước này sẽ thực hiện biến đổi kết quả trả về của mạng nơron sang dạng hiểu được theo yêu cầu của bài toán (hình 1).



Hình 1. Quá trình xử lý dữ liệu

2. CÁC BƯỚC THỰC HIỆN THIẾT KẾ MÔ HÌNH DỰ BÁO SỬ DỤNG MẠNG NƠRON

Cần nhấn mạnh rằng, sự thành công của mô hình dự báo sử dụng mạng nơron phụ thuộc rất nhiều vào các yếu tố như dữ liệu, thuật toán huấn luyện, sai số tuyệt đối sau khi huấn luyện... Chính vì vậy, việc xác định rõ các công việc chính cần thực hiện trước khi thiết kế mô hình là rất cần thiết. Do có rất nhiều việc cần thực hiện trong cả quá trình thiết kế, nên một trong các phương pháp hữu hiệu là phân chia quá trình này thành các bước riêng biệt ([3]).

Trong quá trình thiết kế mô hình dự báo sử dụng mạng nơron, các vấn đề sau cần được quan tâm:

Tiền xử lý dữ liệu: xác định tần xuất của dữ liệu: hàng ngày, hàng tuần, hàng tháng hay hàng quý; kiểu dữ liệu: các chỉ số kỹ thuật hay các chỉ số căn bản; cách thức chuẩn hóa dữ liệu: max/min hay trung bình/độ lệch chuẩn (mean/standard deviation).

Huấn luyện: xác định hệ số học, hệ số bước đà, hệ số thứ lỗi, số chu kỳ tối đa, hệ số học tối đa, thực hiện lấy ngẫu nhiên trọng số, xác định kích thước của các tập huấn luyện, kiểm tra, và kiểm định.

Cấu trúc mạng (topology): số đầu vào, số lớp ẩn, số nơron trong các lớp, số nơron đầu ra, hàm chuyển cho các nơron, hàm lỗi.

Dưới đây là các bước chính cần thực hiện khi thiết kế mô hình mạng nơron sử dụng cho bài toán dự báo:

(i) Chọn lựa các biến.

(ii) Thu thập dữ liệu.

(iii) Tiền xử lý dữ liệu.

(iv) Phân chia tập dữ liệu thành các tập: huấn luyện, kiểm tra, kiểm định.

(v) Xác định cấu trúc mạng: số lớp ẩn, số nơron trong các lớp ẩn, số nơron đầu ra, các hàm chuyển.

(vi) Xác định tiêu chuẩn đánh giá (hàm lỗi)

(vii) Huấn luyện mạng.

(viii) Thực thi trong thực tế.

Trong khi thực hiện, không nhất thiết phải theo thứ tự các bước mà có thể quay lại các bước trước đó, đặc biệt là ở bước huấn luyện và lựa chọn các biến. Lý do là trong quá trình thiết kế, nếu việc chọn lựa các biến đầu vào ban đầu cho kết quả không tốt thì cần thực hiện chọn lựa lại và kéo theo là cần huấn luyện lại ([1, 3, 5, 6, 8]).

Trong số các bước nêu trên, có thể coi bước *Chọn lựa các biến* là một bước trong quá trình *Thu thập dữ liệu*, bởi lẽ bài toán dự báo được giải, thông qua quá trình thử các khả năng kết hợp có thể của các biến ảnh hưởng. Tuy vậy, do tính chất quan trọng của việc chọn lựa các biến nên cần phải được đưa thành một bước riêng biệt. Bước *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. Các đầu vào, đầu ra của mạng nơron hiếm khi được đưa trực tiếp vào mạng. 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]).

Các phương pháp phổ biến như sau:

$$SV = ((0.9 - 0.1)/(MAX\_VAL - MIN\_VAL)) * (OV - MIN\_VAL),$$

hoặc đưa về khoảng giữa giá trị min và max:

$$SV = TF\min + ((TF\max - TF\min)/(MAX\_VAL - MIN\_VAL)) * (OV - MIN\_VAL),$$

trong đó:

SV: Scaled Value - Giá trị sau khi biến đổi,

MAX_VAL : Giá trị lớn nhất của dữ liệu,

MIN_VAL : Giá trị nhỏ nhất của dữ liệu,

$TF\max$: Giá trị lớn nhất của hàm chuyển,

$TF\min$: Giá trị nhỏ nhất của hàm chuyển,

OV : Original Value - Giá trị ban đầu.

Trước khi đưa tập dữ liệu mẫu đã qua tiền xử lý vào huấn luyện, người ta thường phân chia tập dữ liệu thành các tập: huấn luyện, kiểm tra và kiểm định. 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 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.

Cần nhấn mạnh rằng dữ liệu đóng vai trò quyết định đối với khả năng hoạt động của mạng. Mặc dù vậy, việc Xác định cấu trúc mạng cũng đóng một vai trò quan trọng khi thực thi.Việc xác định cấu trúc mạng nơron bao gồm việc xác định sự liên kết giữa các nơron, đồng thời xác định cấu trúc của mạng bao gồm số lớp ẩn, số nơron trong từng lớp. Các thực nghiệm cho thấy rằng, số lớp ẩn sử dụng trong mạng không nên vượt quá hai lớp và không có phương pháp đúng đắn nào có thể chọn được số tối ưu các nơron sử dụng trong lớp ẩn. Có một số phương pháp cho ta lựa chọn ban đầu mặc dù chúng không đảm bảo được rằng đó sẽ là lựa chọn tốt nhất cho bài toán ([4]).

Để có thể xác định được khả năng hoạt động của mạng trước và sau khi huấn luyện, người ta cần *Xác định tiêu chuẩn* đánh giá. Hàm được sử dụng để đánh giá mạng thường là hàm trung bình bình phương lỗi. Các hàm khác có thể là hàm độ lệch nhỏ nhất (least absolute deviation), hiệu phần trăm (percentage differences), bình phương nhỏ nhất bất đối

xứng (asymmetric least squares)... Tuy nhiên, các hàm này có thể không phải là hàm đánh giá chất lượng cuối cùng cho mạng. Phương pháp đánh giá các giá trị dự báo hay được sử dụng là giá trị trung bình tuyệt đối phần trăm lỗi (Mean Absolute Percentage Error - MAPE). Chẳng hạn trong các hệ thống bán hàng, các giá trị dự báo của mạng nơron sẽ được chuyển sang tín hiệu mua hoặc bán tùy thuộc vào một tiêu chuẩn xác định trước đó.

Một bước quan trọng khác trong quá trình thực thi giải pháp cho bài toán dự báo dữ liệu xử dụng mạng nơron đó là *Huấn luyện mạng*. Huấn luyện mạng học các 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.

Vấn đề đặt ra là khi nào thì ngừng huấn luyện. Có hai quan điểm trong vấn đề này. Quan điểm thứ nhất cho rằng chỉ nên ngừng huấn luyện chừng nào không có tiến triển nào của hàm lỗi nữa đối với dữ liệu dựa trên một số tập các tham số của mạng được chọn ngẫu nhiên. Nói cách khác là xác định được khả năng lớn nhất đạt đến được điểm cực tiểu toàn cục. Trường phái thứ hai cho rằng cần thực hiện xem xét thường xuyên khả năng tổng quát hóa của mạng bằng cách sau một số chu kỳ nào đó thực hiện kiểm tra và kiểm tra sự tổng quát hóa của mạng, sau đó lại tiếp tục quá trình huấn luyện.

Cả hai quan điểm này đều thống nhất rằng kết quả kiểm tra trên tập kiểm định là chính xác nhất bởi lẽ nó thể hiện trực tiếp kết quả trả lời của mạng sau khi được huấn luyện.

Việc thực hiện huấn luyện mạng còn cần phải xem xét khả năng của mạng nơron với một số nào đó lần thực hiện huấn luyện mạng trên các tập khởi tạo ban đầu của các tham số. Sau khi thực hiện huấn luyện trên tất cả các tham số này cần thực hiện đánh giá lại kết quả, từ đó đưa ra kết luận về số lần tối đa thực hiện huấn luyện cho mạng cho từng bài toán cụ thể của mình.

Một phương pháp khác là thực hiện vẽ đồ thị để có thể theo dõi trạng thái lỗi của mạng, từ đó có thể quan sát được các vùng mà mạng có trạng thái không thay đổi đối với dữ liệu vào. Thông thường, số lần tối đa thực hiện huấn luyện cho mạng thường có khoảng biến thiên khá lớn: từ vài nghìn cho đến vài chục, vài trăm nghìn chu kỳ, việc theo dõi được trạng thái của mạng đối với tập huấn luyện và khả năng tổng quát hóa để có thể ngừng khi cần là khá quan trọng. Có thể thực hiện cập nhật đồ thị sau một số chu kỳ định trước để có thể theo dõi được các tham số này.

Một bước khác cũng rất quan trọng đó là *Thực thi*. Bước này thực ra cần được xem xét trước cả bước thu thập dữ liệu. Bởi lẽ, việc xác định khả năng sẵn có của dữ liệu, xác định hàm lỗi sử dụng và thời gian huấn luyện đều là những đặc trưng của môi trường mà mạng sẽ được triển khai. Tuy vậy, đây lại là bước được thực hiện cuối cùng sau khi đã xác định được các yếu tố liên quan đến cấu trúc mạng và dữ liệu được chọn lựa. Người ta thấy rằng, do mạng nơron có đặc trưng tính toán song song, do vậy mạng nơron tốt nhất nên được thực hiện cài đặt trên các vi mạch điện tử. Tuy nhiên, môi trường máy tính cá nhân lại phù hợp hơn trong quá trình huấn luyện, dễ cài đặt, đồng thời có khả năng linh hoạt đáp ứng được nhiều bài toán hơn.

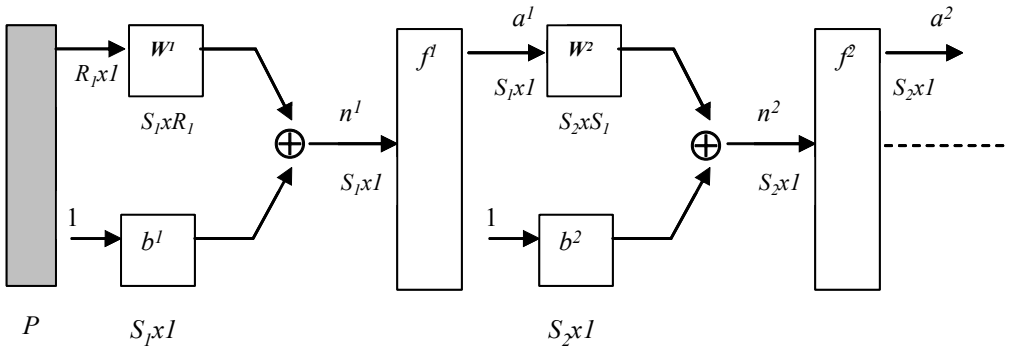
Sau khi cài đặt, triển khai, khả năng hoạt động đúng đắn của mạng nơron sẽ giảm đi theo thời gian nếu như không có bước thực hiện huấn luyện lại, bởi lẽ không thể đảm bảo được rằng các tham biến được lựa chọn sẽ luôn đóng vai trò quyết định đối với các kết quả mà ta mong muốn theo thời gian. Tần số thực hiện huấn luyện lại mạng cần hợp lý sao cho mạng có thể đạt được trạng thái hoạt động tốt nhất.

3. XÁC ĐỊNH THUẬT TOÁN HUẤN LUYỆN VÀ CẤU TRÚC MẠNG

Một phần rất quan trọng có tính chất quyết định là sử dụng cấu trúc mạng và thuật toán huấn luyện nào. Hiện nay, các nghiên cứu cho thấy rằng cấu trúc mạng nơron phù hợp nhất cho các bài toán dự báo là các mạng nơron truyền thẳng nhiều lớp (multi-layer feed-forward neural networks). Thuật toán huấn luyện cho loại mạng này là thuật toán huấn luyện lan truyền ngược có áp dụng các kỹ thuật làm tăng tốc độ hội tụ cho thuật toán (chẳng hạn: sử dụng tham số bước đà, sử dụng hệ số học biến đổi... [2, 3]).

3.1. Mạng nơron truyền thẳng nhiều lớp

Về cơ bản, cấu trúc của một mạng nơron truyền thẳng nhiều lớp có dạng như sau:



Hình 2. Mạng nơron truyền thẳng nhiều lớp

trong đó, P là vectơ đầu vào (vectơ cột),

W^i : Ma trận trọng số của các nơron lớp thứ i ($S^i \times R^i$: S^i hàng (nơron), R^i cột (số đầu vào)),

b^i : Vectơ độ lệch (*bias*) của lớp thứ i ($S^i \times l \to S^i \times 1$: cho S nơron),

n^i : Net input ($S^i \times 1$),

f^i : Hàm chuyển (hàm kích hoạt),

a^i : Net output ($S^i \times 1$),

$\oplus$: Hàm tổng thông thường.

Mỗi liên kết gắn với một trọng số, trọng số này được thêm vào trong quá trình tín hiệu đi qua liên kết đó. Các trọng số có thể dương-thể hiện trạng thái kích thích, hay âm - thể hiện trạng thái kiềm chế. Mỗi nơron tính toán mức kích hoạt của chúng bằng cách cộng tổng các đầu vào và đưa ra hàm chuyển. Một khi đầu ra của tất cả các nơron trong một lớp mạng cụ thể đã thực hiện xong tính toán thì lớp kế tiếp có thể bắt đầu thực hiện tính toán của mình, bởi vì đầu ra của lớp hiện tại tạo ra đầu vào của lớp kế tiếp. Khi tất cả các nơron đã thực hiện tính toán thì các nơron đầu ra trả lại kết quả tính toán được. Tuy nhiên, có thể là chưa đúng yêu cầu, khi đó cần áp dụng thuật toán huấn luyện để điều chỉnh các tham số của mạng.

3.2. Thuật toán huấn luyện lan truyền ngược (back-propagation)

Đây là một thuật toán khá phổ biến sử dụng trong việc huấn luyện cho mạng nơron. Tư

tưởng của thuật toán này dựa theo phương pháp học có thầy (supervised learning), nghĩa là đầu vào thuật toán là các cặp (p_i, t_i), ở đó thuật toán sẽ thực hiện tính toán các đầu ra của mạng đối với các đầu vào p_i, sau đó thực hiện cập nhật các trọng số của mạng nơron dựa trên xác định lỗi giữa đầu ra thực tế và đầu ra mong muốn t_i ([2]).

Đây cũng là một thuật toán theo kiểu giảm theo hướng (gradient descent). Thuật toán này được mô tả như sau:

Thuật toán huấn luyện lan truyền ngược (back-propagation)

Bước 1: Lan truyền xuôi đầu vào qua mạng:

$$a^0 = p,$$

$$a^{m+1} = f^{m+1}(W^{m+1}a^m + b^{m+1}), \quad \text{với } m = 0, 1, ..., M-1,$$

$$a = a^M.$$

Bước 2: Lan truyền độ nhậy cảm (lỗi) ngược lại qua mạng:

$$s^M = -2F^M(n^M)(t-a), \text{ với } m = M-1, ..., 2, 1.$$

Bước 3: Cuối cùng, các trọng số và độ lệch được cập nhật bởi công thức sau:

$$W^m(k+1) = W^m(k) - \alpha s^m(a^{m-1})^T,$$

$$b^m(k+1) = b^m(k) - \alpha s^m.$$

Tuy nhiên, mạng nơron sử dụng thuật toán huấn luyện này tồn tại nhược điểm: thuật toán có thể rơi vào điểm cực tiểu địa phương đối với mạng nơron truyền thẳng nhiều lớp sử dụng các hàm chuyển phi tuyến. Hơn nữa, khi thực hiện luyện mạng bằng cách đưa từng mẫu vào, sau đó thực hiện cập nhật tham số, sẽ làm ảnh hưởng đến quá trình học các mẫu khác. Do đó, một cách để tăng tốc độ hội tụ là sử dụng phương pháp học cả gói (batch training), nghĩa là tất cả các mẫu được đưa vào mạng, sau đó mới thực hiện cập nhật các tham số. Có một số biến thể của thuật toán lan truyền ngược sử dụng phương pháp học cả gói nhằm vượt qua các nhược điểm này. Dưới đây sẽ trình bày một vài biến thể đó.

3.2.1. Sử dụng tham số bước đà (Momentum)

Đây là phương pháp heuristic dựa trên quan sát kết quả luyện mạng nhằm làm tăng tốc độ hội tụ của thuật toán lan truyền ngược. Thuật toán lan truyền ngược cập nhật các tham số của mạng bằng cách cộng thêm vào một lượng thay đổi là:

$$\Delta W^m(k) = -\alpha s^m(a^{m-1})^T, \ \Delta b^m(k) = -\alpha s^m.$$

Khi áp dụng thuật toán lan truyền ngược có sử dụng bước đà, phương trình trên thay đổi như sau:

$$W^m(k) = \gamma \Delta W^m(k-1) - (1-\gamma)\alpha s^m(a^{m-1})^T,$$

$$\Delta b^m(k) = \gamma \Delta b^m(k-1) - (1-\gamma)\alpha s^m.$$

Người đa đã chứng tỏ rằng ([1, 2, 6]) khi sử dụng tham số bước đà thì hệ số học có thể lớn hơn rất nhiều so với thuật toán lan truyền ngược không sử dụng tham số bước đà trong khi vẫn giữ được độ tin cậy của thuật toán. Một điểm khác nữa là khi sử dụng tham số bước

đà thì sự hội tụ của thuật toán sẽ được tăng tốc nếu như hàm hiệu năng chỉ đi xuống trong một khoảng dài.

3.2.2. Sử dụng hệ số học biến đổi

Trong thực tế, các hàm hiệu năng có dạng biểu diễn hình học không đồng đều, có lúc có dạng phẳng (hàm không thay đổi giá trị hoặc thay đổi rất ít) hoặc có dạng phễu (giá trị của hàm thay đổi rất nhanh khi thay đổi tham số đầu vào). Nếu ta chỉ sử dụng hệ số học cố định thì có thể sẽ tốn thời gian tại các vùng phẳng. Vì vậy, tư tưởng của thuật toán lan truyền ngược sử dụng hệ số học biến đổi là khi gặp vùng phẳng thì tăng hệ số học lên và ngược lại, khi gặp vùng dạng phễu thì giảm hệ số học đi.

Người ta đã đưa ra rất nhiều phương pháp để thực hiện điều trên, ở đây chỉ nêu ra một cách biến đổi hệ số học dựa trên hiệu năng của mạng ([2]).

Bước 1: Nếu bình phương lỗi trên toàn bộ tập huấn luyện tăng một số phần trăm cho trước ξ (thông thường từ 1% cho đến 5%) sau một lần cập nhật trọng số thì bỏ qua việc cập nhật này, hệ số học được nhân với một số hạng ρ nào đó (với $0 < \rho < 1$) và tham số bước đà (nếu có sử dụng) được đặt bằng 0.

Bước 2: Nếu bình phương lỗi giảm sau một lần cập nhật trọng số, thì cập nhật đó là chấp nhận được và hệ số học được nhân với một số hạng nào đó lớn hơn 1, nếu tham số bước đà đã bị đặt bằng 0 thì đặt lại giá trị lúc đầu.

Bước 3: Nếu bình phương lỗi tăng một lượng bé hơn ξ thì cập nhật trọng số là chấp nhận được, nhưng hệ số học không thay đổi và nếu tham số bước đà đã bị đặt bằng 0 thì đặt lại giá trị lúc đầu.

Các thuật toán heuristic luôn cho sự hội tụ nhanh hơn trong một số bài toán, tuy nhiên chúng có hai nhược điểm chính sau đây:

Thứ nhất, việc sửa đổi thuật toán lan truyền ngược cần có thêm một số tham số, trong khi trong thuật toán lan truyền ngược chuẩn chỉ yêu cầu có một tham số là hệ số học. Một số thuật toán sửa đổi cần đến năm hoặc sáu tham số, trong khi hiệu năng của thuật toán khá nhạy cảm đối với những thay đổi của các tham số này. Hơn nữa việc chọn lựa các tham số lại độc lập với bài toán đặt ra.

Thứ hai, các thuật toán sửa đổi có thể không hội tụ trong một số bài toán mà thuật toán lan truyền ngược chuẩn có thể hội tụ được.

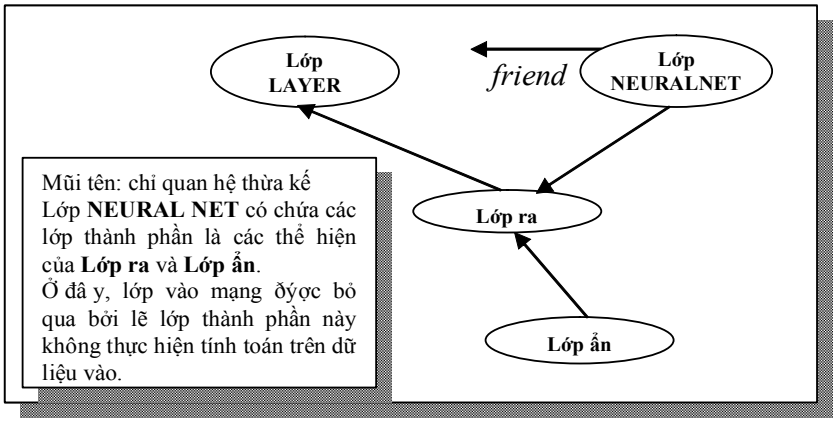
Thực tế cho thấy cả hai nhược điểm nêu trên thường xảy ra khi sử dụng các thuật toán sửa đổi phức tạp hơn (yêu cầu nhiều tham số hơn).

4. ỨNG DỤNG

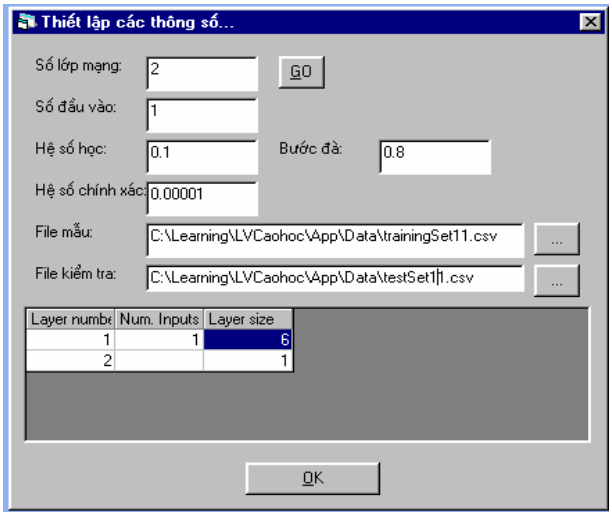
Dựa trên các cơ sở nêu trên, chúng tôi đã xây dựng một phần mềm thử nghiệm. Để có thể áp dụng được cho một lớp các bài toán dự báo dữ liệu, phần mềm được xây dựng theo cách tiếp cận hướng đối tượng, trong đó, mạng nơron áp dụng cho một bài toán cụ thể được xây dựng dựa trên các lớp thành phần ([7]).

Khi đó, chính bản thân mạng nơron là thể hiện của một lớp, lớp NEURAL NET.

Các tham số cho mạng do người sử dụng tự xác định. Các tham số này bao gồm: số lớp mạng, số đầu vào, hệ số học, hệ số bước đà, hệ số chính xác khi huấn luyện, tệp chứa các mẫu huấn luyện và tệp chứa các mẫu kiểm tra.

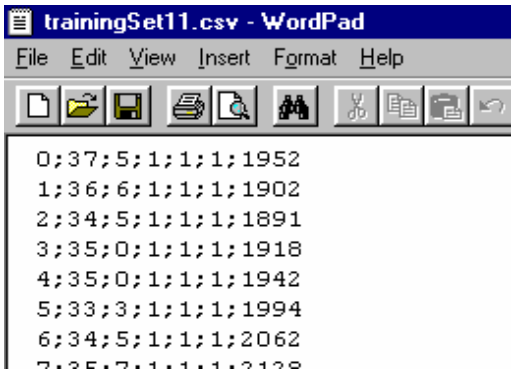


Màn hình nhập các tham số này như sau:



Các tệp dữ liệu nói trên là các tệp có cấu trúc, trong đó các trường dữ liệu được phân cách bởi dấu ";", trường dữ liệu dự báo là trường cuối cùng, sau trường dữ liệu dự báo không cần phải có dấu ";" và tệp dữ liệu không được có các khoảng trống ở phía cuối. Nếu có thì cần phải loại bỏ.

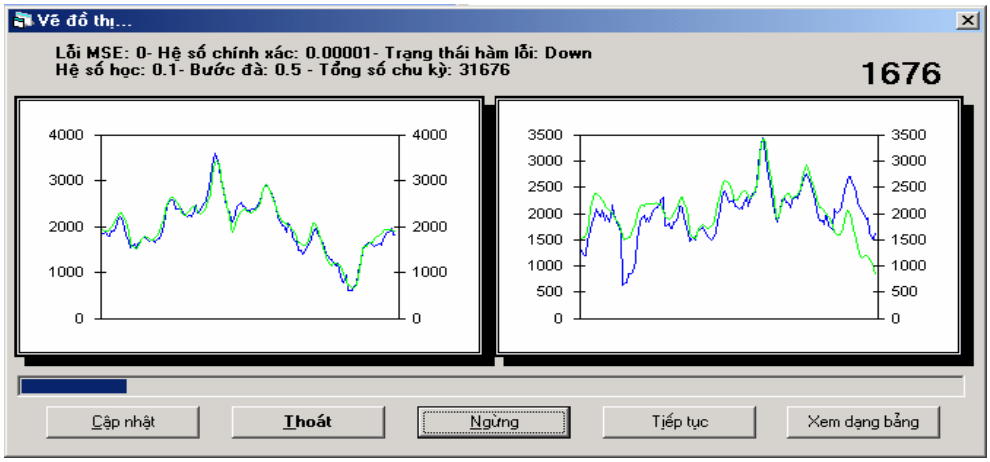
Ví dụ, tệp dữ liệu có dạng như sau:



Các dữ liệu sau khi được đọc vào sẽ được chuẩn hóa về khoảng [0,1] theo phương pháp:

$$SV = ((0,9-0,1)/(MAX\_OF\_EXP - MIN\_OF\_EXP)) * (OV - MIN\_OF\_EXP),$$

Sau khi qua bước thiết lập các thông số cho mạng, có thể bắt đầu huấn luyện mạng. Màn hình thể hiện trạng thái của việc huấn luyện có dạng sau:



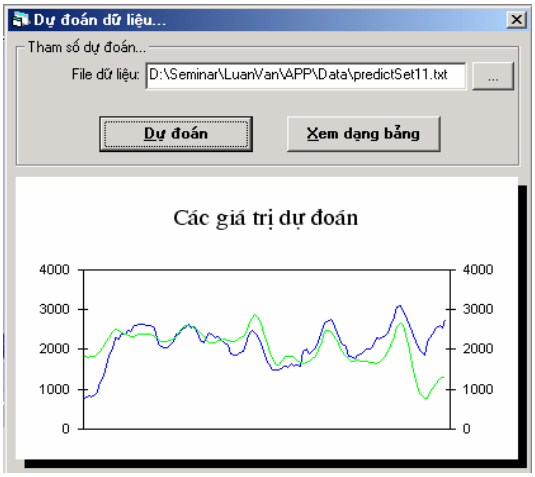
Chú thích:

Đồ thị bên trái thể hiện kết quả huấn luyện mạng trên tập mẫu đưa vào. Đồ thị bên phải thể hiện trả lời của mạng đối với các mẫu kiểm tra, các mẫu chưa đưa vào mạng.

Các đường nhạt là các đầu ra mong muốn đối với tập dữ liệu. Các đường sẫm là trả lời của mạng đối với các dữ liệu đầu vào đưa vào nó.

Lỗi MSE (Mean Square Error) được giảm sau một thời gian huấn luyện, đồng thời khả năng tổng quát hóa của mạng đối với các dữ liệu chưa được "biết" cũng sẽ tốt lên.

Sau khi mạng đã được huấn luyện, có thể sử dụng để dự báo dữ liệu. Chỉ cần xác định tệp chứa dữ liệu và thực hiện dự báo. Màn hình như sau:



Nếu như các tập dữ liệu được chuẩn bị tốt thì kết quả dự báo có thể chính xác đến hơn 90%.

5. KẾT LUẬN

Trên đây là các vấn đề cần quan tâm trong quá trình thiết kế mô hình dự báo dữ liệu sử dụng mạng nơron. Trong số đó, việc xác định các công việc cần thực hiện trước khi thiết kế mô hình mang ý nghĩa thiết thực, có thể giúp những người phát triển hệ thống có được cái nhìn cụ thể hơn trong quá trình thực hiện các giải pháp sử dụng mạng nơron.

Trong thực tế, việc ứng dụng thành công mạng nơron phụ thuộc vào 3 yếu tố chủ yếu sau: thứ nhất, thời gian để có thể chọn lựa các đầu vào cho mạng từ một số khổng lồ các dữ liệu, cũng như thực hiện tiền xử lý các dữ liệu này; thứ hai, phần mềm được xây dựng cần cung cấp các chức năng như kiểm tra mức độ tổng quát hóa, tìm ra số nơron tối ưu cho lớp ẩn và kiểm tra đối với các tập dữ liệu vào khác nhau; thứ ba, người phát triển cần xem xét các khả năng có thể trong mỗi lần thực hiện kiểm tra sự hoạt động của mạng đối với mỗi tập dữ liệu vào, cũng như mỗi cấu trúc mạng đã được chọn lựa.

TÀI LIỆU THAM KHẢO

[1] Dipti Srinivasan, A.C. Liew, S. John, P. Chen, Short term forecasting using neural network approach, *IEEE 91TH0374-9/91/0000-0012* (1991) 12–16.

[2] M.T. Hagan, H.B. Demuth, M. Beale, *Neural networks design,* PWS Publishing Company, Boston, Ma, 1996.

[3] I. Kaastra, M. Boyd, Designing a neural network for forecasting financial and economic time series, *Neurocomputing* **10** (1996) 215–236.

[4] S. Lawrence, C.L. Giles, a.C. Tsoj, What size neural network gives optimal generalization? Convergence properties of backpropagation, *Technical Report*, Institute for Advanced Computer Studies - University of Maryland College Park, June 1996.

[5] Y. Morioka, K. Sakurai, A. Yokoyama, Y. Sekine, Next day peak load forecasting using a multilayer neural network with an additional learning, *IEEE*, 0-7803-1217-1/93, (1993).

[6] H.L. Poh, J.T. Yao, T. Jaic, Neural networks for the analysis and forecasting of advertising and promotion impact, *International Journal of Intelligent Systems in Accounting, Finance & Management* **7** (1998) 253–268.

[7] Rao, B. Valluru, Rao, V. Hayagriva, *C^{++} Neural Networks and Fuzzy Logic*, MIS Press, 1993.

[8] O. Takashi, Next day's peak load forecasting using an artificial neural network, *IEEE 0-7803-1217-1/93* (1993) 284–289.

Nhận bài ngày 19 - 8 - 2003