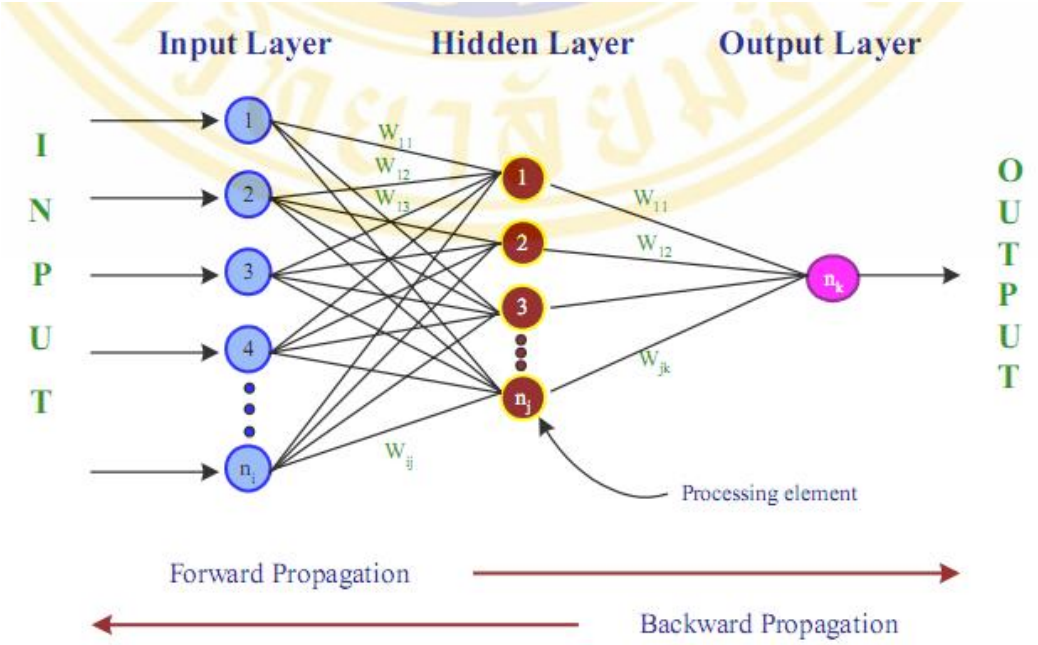
MỤC LỤC

Chương 1: Hướng tiếp cận mô hình dự đoán giá

1.1 Phương pháp Training:



Luồng dữ liệu khi áp dụng vào mô hình ANN sẽ qua các bước xử lý sau:

- Cho vector input I = (i_1, i_2, …, i_n) được truyền vào lớp input của mạng. Sau quá trình tiền xử lý (khử nhiễu, chuẩn hóa), mỗi node ở lớp input sẽ tạo ra vector $O^{(i)}$ = (o_1, o_2, …, o_n) truyền đến lớp ẩn.
- Giá trị đầu vào $i_j^{(h)}$ của neuron thứ j của lớp ẩn (h) sẽ được tính như sau:

$$i_j^{(h)} = b + \sum_{i=1}^{N} w_{ij}^{(h)} o_i^{(i)}$$

Giải thích thêm:Ở đây ta sẽ có một số công thức khác cho cách tính giá trị đầu vào cho neuron ở lớp ẩn như sau:

- Phép hợp nhất tuyến tính (Công thức đang được dùng)

$$z_i(t) = \sum_{j=1}^{m} w_{ij}(t) x_j(t) - \theta_i(t)$$

- Phép hợp nhất bậc hai

$$z_i(t) = \sum_{j=1}^{m} w_{ij}(t) x_j^2(t) - \theta_i(t)$$

- Phép hợp nhất cầu

$$z_i(t) = \rho^{-2} \sum_{j=1}^{m} (x_j(t) - w_{ij}(t))^2 - \theta_i(t)$$

- Phép hợp nhất đa mức

$$z_i(t) = \sum_{j=1}^{m}\sum_{k=1}^{m} w_{ijk}(t)x_j(t)x_k(t) + x_j^{\alpha_j} + x_k^{\alpha_k} - \theta_i(t)$$

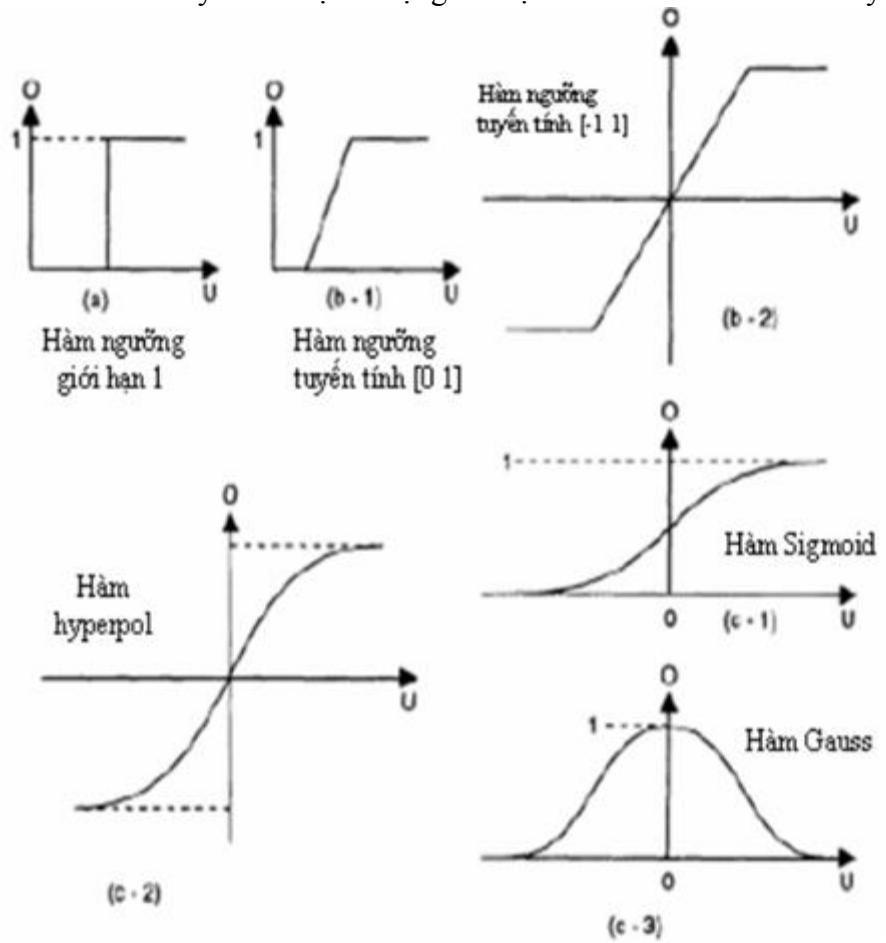
- Đây được gọi là phép toán hợp nhất, mục đích của nó là đo lường (hay thể hiện) mối quan hệ giữa vector input với vector trọng số. Nó giống như một sự tổng hợp thông tin trong não con người, từ mỗi luồng input sẽ có độ mạnh yếu khác nhau mà ở đây nó thể hiện qua công thức nhân giá trị input với giá trị trọng số tương ứng.
- Đa phần trong nhiều bài báo **ta thấy thường dùng nhất là phép hợp nhất tuyến tính**.

▪ Giá trị output của neuron thứ j trong lớp ẩn là $o_j^{(h)}$ được tính theo công thức:

$$o_j^{(h)} = f_j(net)$$

Ở đây net chính là $i_j^{(h)}$ *và* $f_j(net)$ được gọi là hàm truyền phi tuyến.

Giải thích thêm: Ở đây ta có một số dạng đồ thị biểu diễn cho các hàm truyền như sau:



- Công dụng: Hàm truyền phi tuyến có chức năng xếp hạng độ đo sao cho đảm bảo tính mềm dẻo và chặt chẽ của ánh xạ neuron và trả về độ đo tại đầu ra của neuron. Theo cách hiểu đơn giản hơn, nó cho biết tín hiệu sau tổng hợp này là mạnh hay yếu trong một ngưỡng nào đó (thường là ngưỡng [0,1] vì ngưỡng này giống như chỉ độ mạnh 0 – 100%).

- Ở đây do giá đóng cửa chứng khoán không âm nên ta sẽ chọn ngưỡng [0,1] và để biểu đạt sự chuyển biến mịn, mềm dẻo trong ngưỡng [0,1] thì ta chọn hàm Sigmoid.

$$f_j(net) = \frac{1}{1 + e^{-net}}$$

- Giá trị đầu vào $i_k^{(o)}$ của neuron thứ k trong lớp output (o) được tính theo công thức:

$$i_k^{(o)} = c + \sum_{j=1}^{M} w_{jk}^{(o)} o_j^{(h)}$$

- Giá trị đầu ra của neuron thứ k trong lớp output là $o_k^{(o)}$được tính theo công thức:

$$o_k^{(k)} = g_k(net)$$

Ở đây net chính là $i_k^{(o)}$và $g_k(net)$ là hàm truyền phi tuyến cho lớp output.

$$g_k(net) = net$$

- Tiếp theo khi đã có kết quả ở lớp output, ta cần tính độ lỗi $\delta_k^{(o)}$ giữa giá trị dự đoán với giá trị thực để lan truyền ngược lại cập nhật trọng số theo công thức:

$$\delta_k^{(o)} = \left(d_k - o_k^{(o)}\right).g_k'(net)\ (****)$$

Ở đây d_k là giá trị thực và $g_k'(net)$là đạo hàm của hàm truyền phi tuyến ở lớp xuất.

$$g_k'(net) = 1$$

- Ta tính tiếp độ lỗi ở các neuron lớp ẩn $\delta_j^{(h)}$theo công thức:

$$\delta_j^{(h)} = f_j'(net).\sum_{k=1}^{P} \delta_k^{(o)} w_{jk}^{(o)}$$

Ở đây $f_j'(net)$ là đạo hàm của hàm truyền phi tuyến ở lớp ẩn.

$$f_j'(net) = f_j(net).[1 - f_j(net)]$$

- Độ thay đổi trọng số ở lớp output $\Delta w_{jk}^{(o)}$ và ở lớp ẩn $\Delta w_{ij}^{(h)}$ ở vòng lặp t được tính như sau:

$$\Delta w_{jk}^{(o)}(t) = \eta \delta_k^{(o)} o_j^{(h)} + \alpha[\Delta w_{jk}^{(o)}(t-1)]$$
$$\Delta w_{ij}^{(h)}(t) = \eta \delta_j^{(h)} o_i^{(i)} + \alpha[\Delta w_{ij}^{(h)}(t-1)]$$

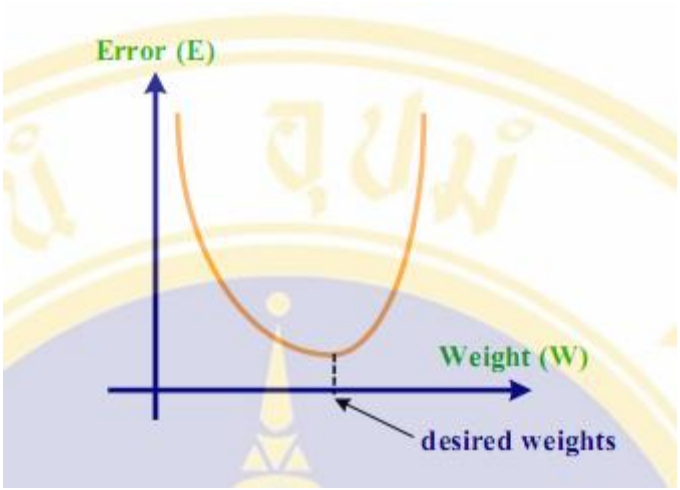
Ở đây η là hệ số học và α là nhân tố hướng để tránh bị rơi vào bẫy tối ưu cục bộ.

- Trọng số mới được cập nhật theo công thức sau:

$$w_{jk}^{(o,new)} = w_{jk}^{(o)} + \Delta w_{jk}^{(o)}(t)$$
$$w_{ij}^{(h,new)} = w_{ij}^{(h)} + \Delta w_{ij}^{(h)}(t)$$

1.2 Tối thiểu hóa độ lỗi:

- Để tối thiểu hóa độ lỗi của mô hình, ta sẽ tính độ lỗi ở mỗi lần train (hay test) mẫu, người ta đưa ra mối tương quan giữa độ lỗi và trọng số qua đồ thị sau:



- Mục tiêu của chúng ta là cần tìm ra một trọng số sao cho ở đấy độ lỗi đạt cực tiểu.
- Mỗi mẫu p, chúng ta gán tương ứng độ lỗi E_p. Ta có một trong những công thức tính độ lỗi đơn giản như sau:

$$E_p = \frac{1}{2}(d - o)^2$$

- Độ lỗi cho cả mô hình chính là tổng độ lỗi khi train (hay test) một mẫu

$$E = \sum_p E_p$$

- Như vậy để mô hình dự đoán chính xác, ta cần tìm ra một bộ trọng sao cho ở đấy độ lỗi đạt cực tiểu.
- Mean of square error (MSE) được dùng như là một điều kiện đề ngừng training mạng.

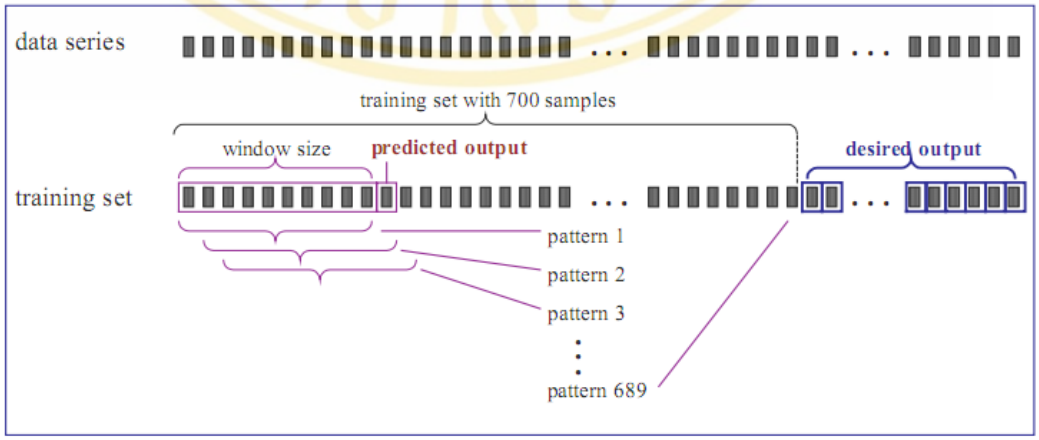
$$MSE = \frac{1}{2N}\sum_{i=1}^{N}(d_i - o_i)^2$$

  Ở đây N là tổng số mẫu.
- Như vậy khi MSE đạt đến một ngưỡng chấp nhận được (do ta qui định), thì mạng coi như training thành công.

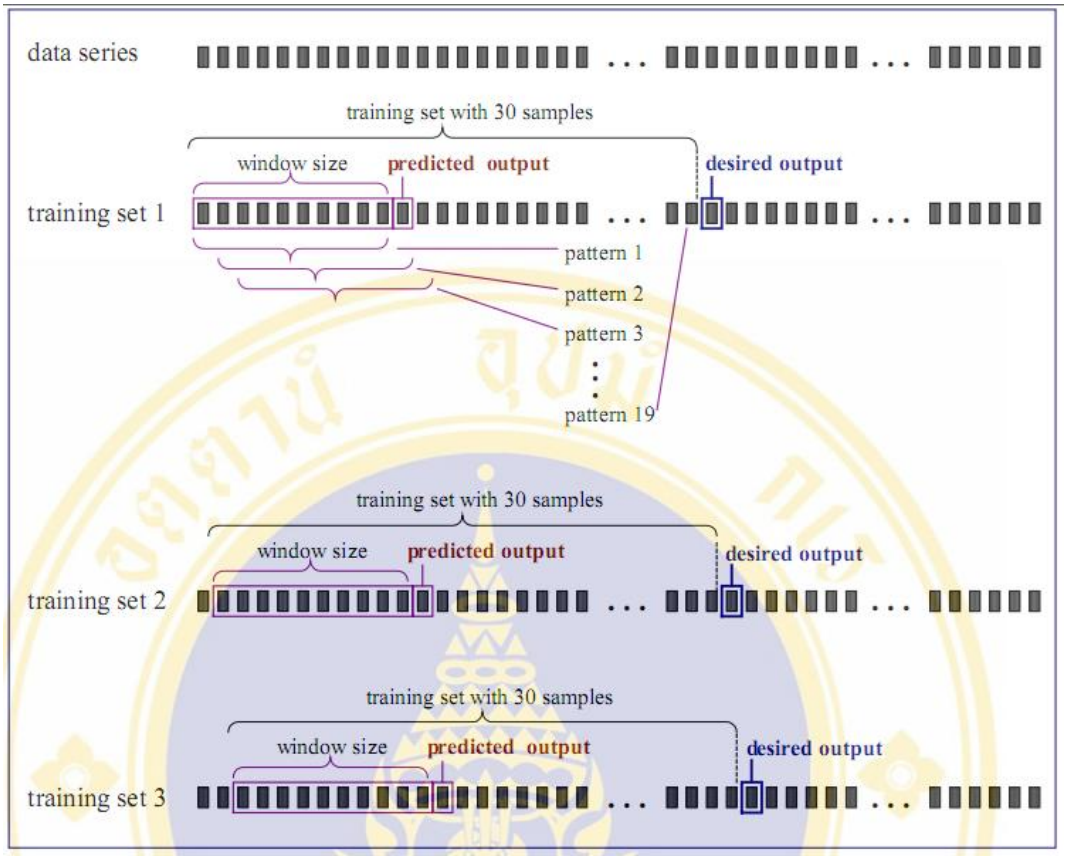
1.3 Các chiến lược dự đoán:

1.3.1 Cách tổ chức dữ liệu đưa vào mô hình:

- Cách truyền thống:



- Cách cải tiến:

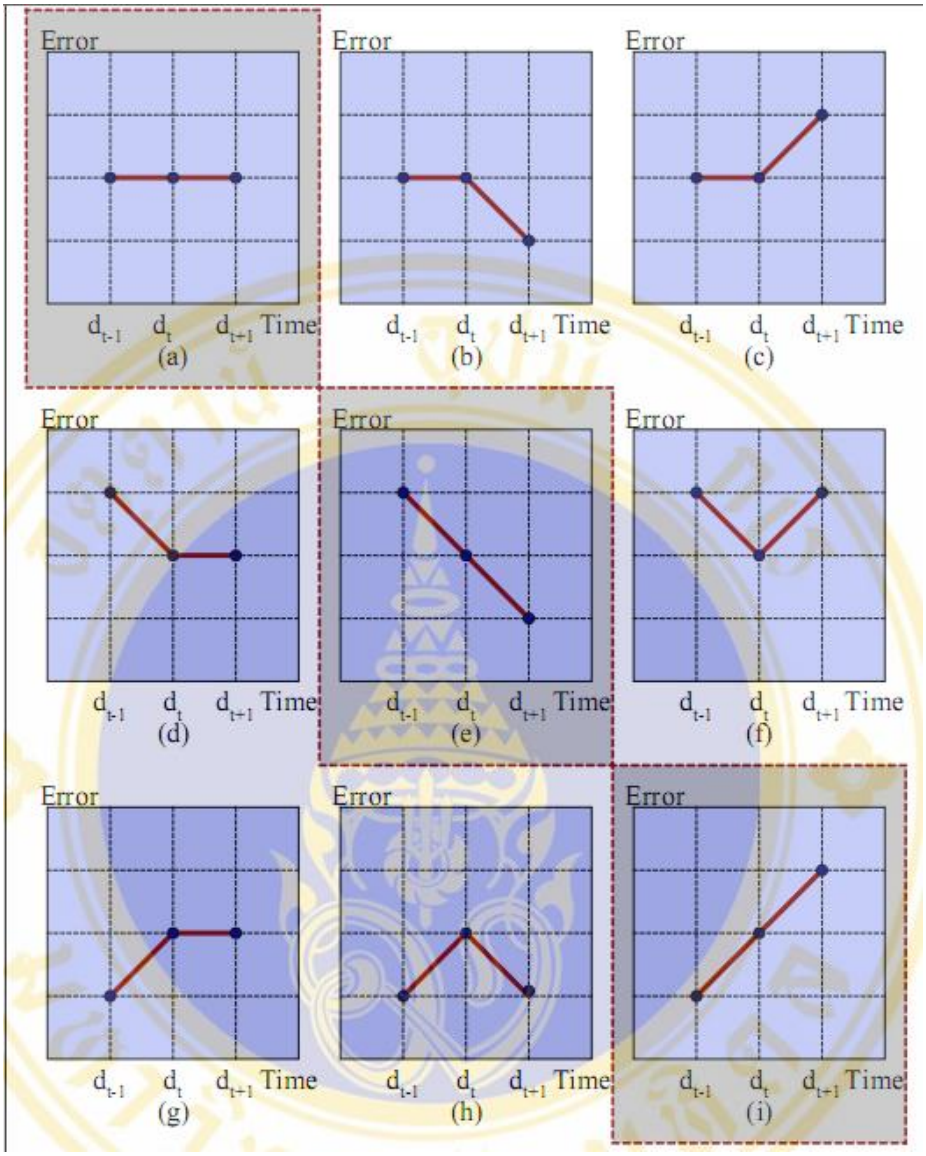


Nhận xét:

- Như đã nói, một trong những nhược điểm lớn của ANN là cần một lượng lớn dữ liệu mẫu cho quá trình training, đó là một trong những nguyên nhân làm quá trình training chậm đi, và hơn nữa chúng ta tiến hành xử lý dữ liệu chuỗi thời gian nên sẽ có một số mẫu sau một thời gian nó không còn mang tính hợp lệ (như giai đoạn đầu mới mở thì nó không theo một quy luật nào, nhưng dần dần tuy vẫn còn nhiễu nhưng nó sẽ tiến đến một xu hướng chung của quốc tế). Đó cũng là nhược điểm ở cách truyền thống.
- Ở cách cải tiến ta sẽ phân đoạn bộ training ra thành những bộ dữ liệu nhỏ hơn để đạt được những kết quả tốt hơn bởi vì mạng neuron bây giờ sẽ được train dần dần theo thời gian.

1.3.2 Tiếp cận theo hướng dùng hàm TSCFD

- Trong dự đoán dữ liệu chuỗi thời gian, độ chính xác không chỉ phụ thuộc vào việc mô hình cho ra kết quả dự đoán gần với giá trị thực, mà nó còn bị chi phối bởi yếu tố xu hướng biến động của giá trị dự đoán.
- Do đó hàm chi phí đáp ứng vấn đề này được đưa ra có tên Tow-Step Continuous Fluctuation Direction-based (TSCFD cost function)
- Hình bên dưới đây sẽ mô tả các trường hợp có thể xảy của giá trị thực:



- Ở đây d_t chính là giá trị thực của ngày dự đoán, d_{t+1} là giá trị của ngày tiếp theo.
- Chúng ta chỉ quan tâm đến trường hợp (a), (e), (i) vì nó biểu lộ xu hướng rõ ràng tăng, giảm hoặc giữ nguyên.
- Hàm này sẽ được cải tiến ở hàm (****) ở mục 1.1

$$\delta_k^{(o)} = \left(x - o_k^{(o)}\right).g_k'(net)$$

Giá trị x ở đây đã thay thế cho d_k ở hàm (****)

- Và x sẽ được tính như sau:

$$x = \begin{cases} d_{k,t} \text{ nếu } (d_{k,t} - d_{k,t-1}) = 0 \text{ và } (d_{k,t+1} - d_{k,t}) = 0 \\ d_{k,t+1} \text{ nếu } (d_{k,t} - d_{k,t-1}) * (d_{k,t+1} - d_{k,t}) > 0 \\ d_{k,t} \text{ trường hợp còn lại} \end{cases}$$

1.3.3 Một số độ đo lỗi đánh giá mô hình:

- MSE, MAE, MAPE: để đo sự khác biệt giữa giá trị đoán và giá trị thực.

$$MSE = \frac{1}{2N}\sum_{i=1}^{N}(d_i - o_i)^2$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|P_i - A_i|$$

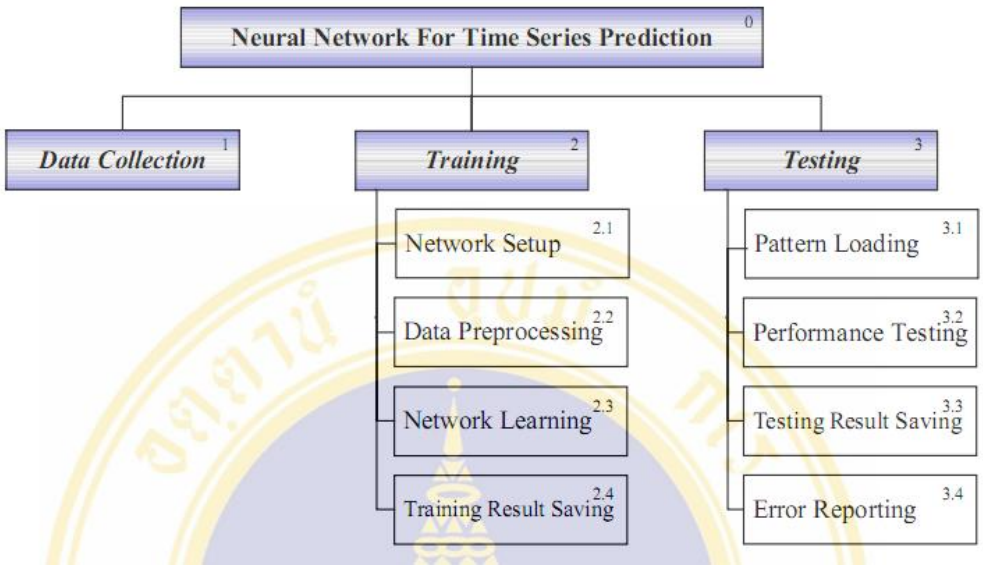
$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{P_i - A_i}{A_i}\right| * 100$$

- Tolerance n%: Đo độ lỗi giữa giá trị dự đoán và giá trị thực trong một phạm vi sai số cho trước.

$$Tolerance\ n\% = \begin{cases} 1 \text{ nếu } (d_k - \frac{d_k * n}{100}) \le x_k \le (d_k + \frac{d_k * n}{100}) \\ 0 \text{ còn lại} \end{cases}$$

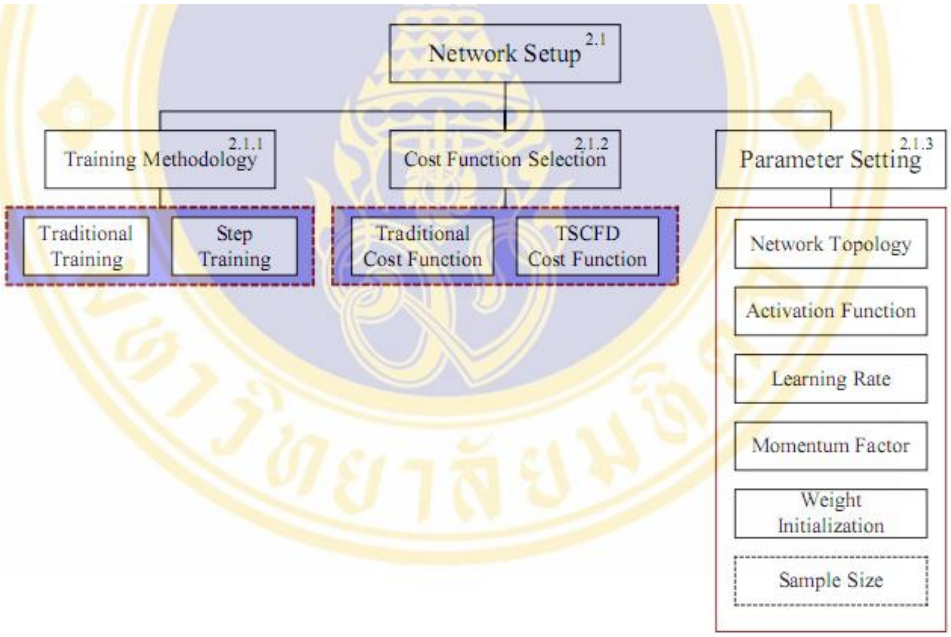
Chương 2 Thiết kế và cài đặt mô hình dự đoán giá

2.1 Sơ đồ tổng quan các chức năng trong mô hình:



2.2 Training:

2.2.1Cài đặt Mạng:



2.2.1.1 Tổ chức dữ liệu training:

- Như đã trình bày ở trên, ta sẽ có 2 cách là cách truyền thống và cải tiến (step-training)

2.2.1.2 Lựa chọn hàm chi phí:

- Ta cũng sẽ có 2 hàm là truyền thống và hàm TSCFD.

2.2.1.3 Các tham số:

- Network Topology: Ở đây ta chọn mạng truyền thẳng 3 lớp với thuật toán học lan truyền ngược.
  - Trong đó lớp output sẽ có 1 node xuất ra giá dự đoán.
  - Số node ở lớp input và lớp ẩn là không biết trước, sẽ phải qua thực nghiệm để quyết định. (Bởi vì không có một giả thiết nào có trước đưa ra số node gần đúng cho mô hình.)
  - Cấu trúc mạng mà một số bài báo đã áp dụng:
    - Input (5,6,10,15,20) – Hiden ($\sqrt{input * output}$)
      5-2-1, 5-4-1,6-3-1,10-3-1,15-3-1,**20-2-1**
    - 64-8-1, 64-16-1, 64-32-1, **64-64-1**, 64-100-1
    - 28-60-1 (Cố định số node ẩn = số node input = n, khảo sát n, rồi sao đó khảo sát nó node ẩn)
    - ?-(9-14)-1
- Hàm truyền phi tuyến: Nhìn chung đều là dùng hàm sigmoid, nhưng có một số dạng như sau
  - $f(net) = \frac{1}{1+e^{-net}}$ $\quad f'(net) = f(net) * [1 - f(net)]$
  - $f(net) = \frac{2}{1+e^{-net}} - 1$ $\quad f'(net) = \frac{1}{2}[1 + f(net)][1 - f(net)]$

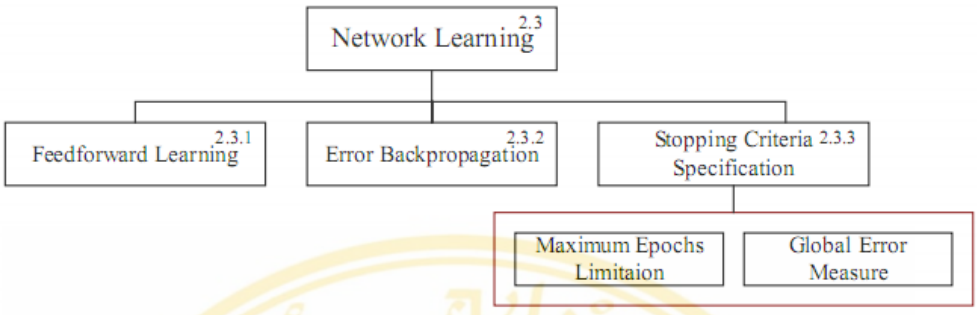
- Hệ số học η: Hệ số lớn sẽ giúp đẩy nhanh tốc độ học, nhưng nếu quá cao sẽ làm cho kết quả không thể hội tụ như mong muốn (step over).Miền giá trị một số bài báo đưa ra là 0.1, 0.01-0.05
- Momentum Factor α: Giúp cho việc tránh rơi vào bẫy tối ưu cục bộ. Cần qua thực nghiệm để lựa chọn.
- Khởi tạo bộ trọng số w ngẫu nhiên trong đoạn [0,1].

2.2.2 Tiền xử lý dữ liệu:

- Đây được đánh giá là một bước khá quan trọng, vì nó sẽ giúp cho mô hình có thể biết được mối quan hệ tưởng tiềm ẩn trong dữ liệu. Trong đó gồm 2 vấn đề chính là khử nhiễu và chuẩn hóa giá trị.
- Về khử nhiễu hiện tại có một số phương pháp: dùng giá trị trung bình trượt (mô hình ARIMA), rút đặc trưng thành phần chính (ICA, PCA, KICA, KPCA), dùng Kohonen.
  Ví dụ: Một công thức ARIMA:
  - $\mu = \frac{x_N - x_1}{N-1}$
  - $\sigma^2 = \frac{1}{N-1}(\sum_{i=1}^{N-1}(x_{i+1} - x_i - \mu)^2)$

- Về chuẩn hóa giá trị, ta sẽ chuẩn hóa về [0,1] cho phù hợp với miền giá trị của hàm truyền phi tuyến sigmoid trong mô hình. Hiện tại có 2 công thức:
  - $z_i = \frac{1}{1+e^{-\frac{x_{i+1}-x_i-\mu}{\sigma}}}$
  - $z_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$

2.2.3 Quá trình training:

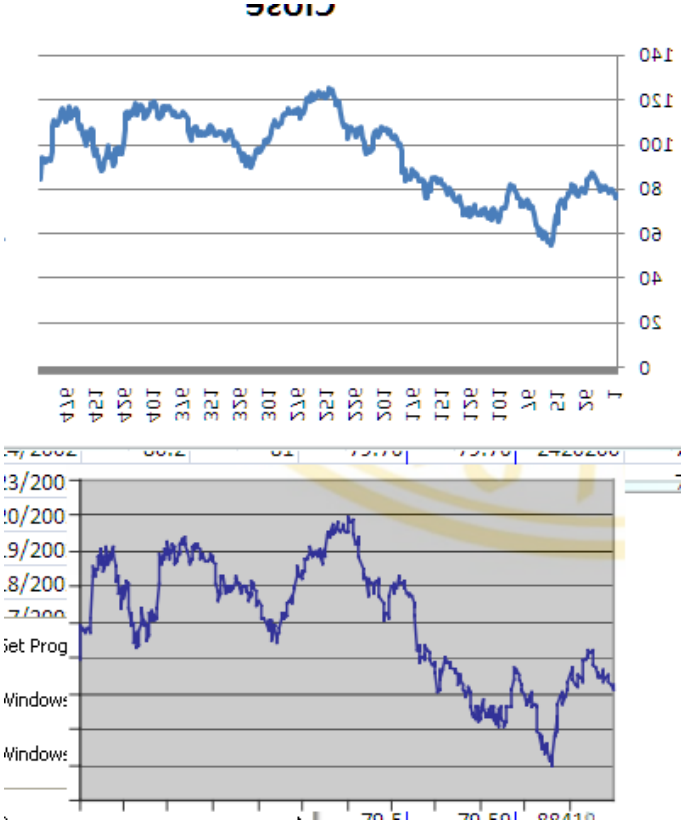


- Số vòng lặp tối đa trong quá trình training sẽ qua thực nghiệm. (Một Bài báo đưa ra 50000)
- Global Error Measure: Ta sẽ dùng MSE, MAE, MAPE, RMSE, NMSE, Tolerance và so sánh kết quả xem cái nào tốt hơn.

2.3 Testing:

2.3.1 Kiểm tra tính đúng đắn trong cài đặt:

- Mô hình sẽ được cài đặt và so sánh với mô hình trong bài báo A Stock Price Prediction Model By The Neural Network Approach của tác giả Prapaphan Pan-O theo độ đo mà bài báo dùng là MSE. Bộ dữ liệu là giá đóng cửa từ tháng 1-2001 đến tháng 1-2003 của IBM.
- Bộ dữ liệu đã tìm được, về cơ bản thì giống nhưng chi tiết số liệu bên trong có sai lệch chút ít. Trong bài báo ghi giá đóng cửa thấp nhất là 54.86 và cao nhất là 124.85, nhưng trong bộ dữ liệu down được thì giá thấp nhất là 54.01 và cao nhất 125.2.



2.3.2 Kịch bản kiểm thử:

- Bộ dữ liệu sẽ chia làm 2 phần train (2/3) và test (1/3).
- Dùng các độ đo đã ghi ở 2.2.3
- Ta sẽ đo độ lỗi lúc train (E_{train}) và đo độ lỗi lúc test (E_{test}).
- Tính hiệu quả của mô hình $H = E_{train} - E_{test}$
- Nếu $H < 0$ cũng có nghĩa là độ lỗi lúc test cao hơn train, mô hình chưa hiệu quả
- Nếu $H > 0$ và H nhỏ hơn ngưỡng chấp nhận được thì mô hình hoạt động tốt.

Chương 3 Mô hình ANN Cho dự đoán xu hướng

Các vấn đề tóm gọn:

3.1 Thông số độ đo

Các độ đo của mục 1.1.3 đây là các độ đo tương đối thông dụng và được sử dụng trong mô hình này.
Và độ đo cải tiến của MSE:
a)

$$E_{DP} = \frac{1}{2}\sum_{p=1}^{N} f_{DP}(p)(t_p - o_p)^2$$

Với:

$$f_{DP}(p) = \begin{cases} a_1 \text{ if } \Delta t_p \Delta o_p > 0 \text{ and } |\Delta t_p| \le \sigma \\ a_2 \text{ if } \Delta t_p \Delta o_p > 0 \text{ and } |\Delta t_p| > \sigma \\ a_3 \text{ if } \Delta t_p \Delta o_p < 0 \text{ and } |\Delta t_p| \le \sigma \\ a_4 \text{ if } \Delta t_p \Delta o_p < 0 \text{ and } |\Delta t_p| > \sigma \end{cases}$$

b)

$$E_{DLS} = \frac{1}{2}\sum_{p=1}^{N} w(p)(t_p - o_p)^2$$

Với:

$$w(p) = \frac{1}{1 + e^{\left(a - \frac{2ap}{N}\right)}}$$

2 công thức cải tiến này giúp hỗ trợ trong việc đánh giá độ lỗi tốt hơn, phục vụ trong việc lan truyền ngược để hiệu chỉnh bộ trọng số(tham khảo từ bài báo *Stock Price Forecasting using Back Propagation Neural Networks*).

3.2 Dữ liệu đầu vào

Dữ liệu đầu vào có 2 lựa chọn:

Dữ liệu dạng cơ bản: chỉ bao gồm giá đóng cửa của các ngày giao dịch (số đầu vào tùy thuộc vào việc chọn Period trong việc train).

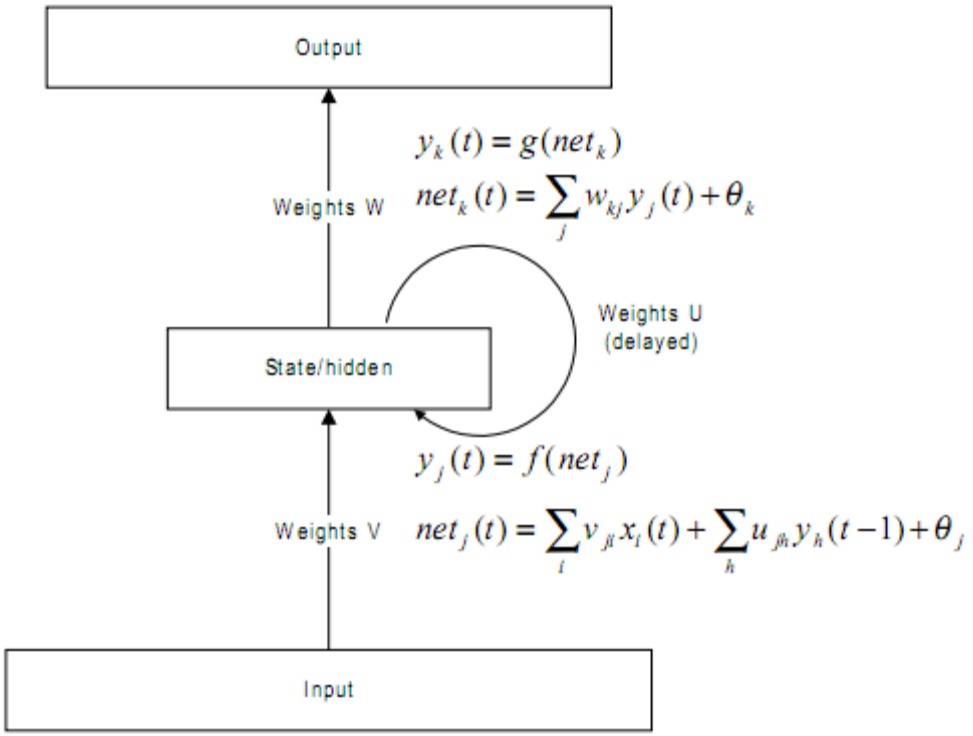
Dữ liệu đạng phức hợp: gồm dữ liệu cơ bản, một vài các chỉ số kỹ thuật thông dụng MFI, RSI, MACD…, các chỉ số cơ bản (nếu có) và một chỉ số đặc biệt đó là chỉ số niềm tin của khách hàng vào cổ phiếu đó(cái này nghe có vẻ lạ tuy nhiên nếu theo chủ quan của người sử dụng cảm nhận mã chứng khoán đó có khả năng đi lên hoặc của chuyên gia thì ta nên gi nhận là 1 input).

Các giá trị đầu vào nên đưa về một dạng chuẩn trong khoản [0,1], có thể dùng công thức ở mục 2.2.2

Và đầu bên dự đoán xu hướng SVM:
"*Đối với mô hình dự đoán xu hướng, giá đóng cửa phải được tinh chỉnh về dạng return. Đây sẽ là dữ liệu đầu vào thực tế:*
$y_t = 100 \times (logI_{t+1} - logI_t)$ *với I là giá*"

3.3 Về mô hình

Trong mô hình ANN áp dụng mô hình ANN hồi quy + lan truyền ngược. Tức là có sử dụng lại yếu tố kết quả của chu kỳ trước trong lúc train, việc này giúp việc cập nhật lại trọng số mang tính kế thừa đồng thời thực hiện lan truyền ngược để hiệu chỉnh bộ trọng tốt nhất.

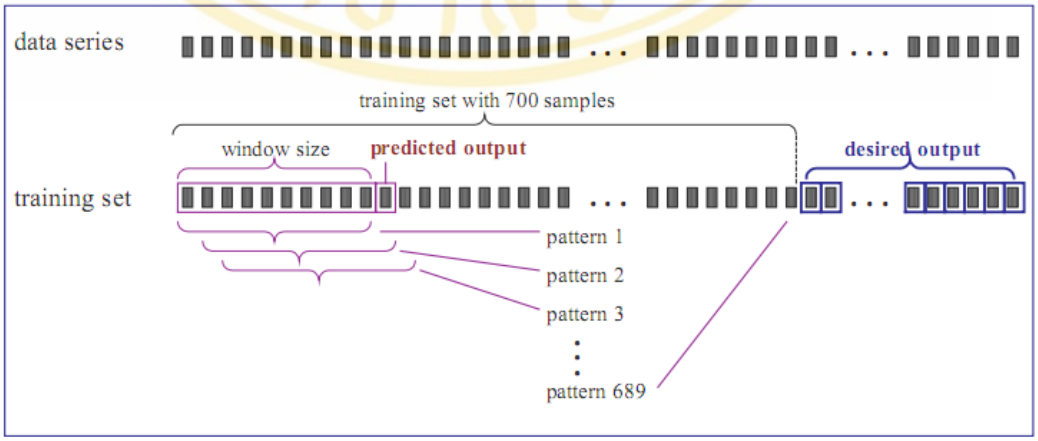


3.4 Về các tham số

Các vấn đề về việc chọn tham số cho ANN phụ thuộc vào quá trình thực nghiệm để tìm ra bộ tham số phù hợp, có thể bộ tham sô này tương đối tốt với dữ liệu của thị trường nước ngoài nhưng với thị trường VN thì không phải là lựa chọn tốt, vì thế cần có quá trình thực nghiệm nhiều để tìm.

3.5 Quá trình huấn luyện và kiểm thử

Với dữ liệu của từng mã chứng khoán, thực hiện huấn luyện theo từng Period cụ thể, dữ liệu được lấy từ Period ngày trước đó để dự đó cho ngày tiếp theo. Như hình minh họa


data series
training set with 700 samples
window size
predicted output
desired output
training set
pattern 1
pattern 2
pattern 3
pattern 689