Neural Network for NN3 Time Series Prediction

Hassan Mohamed Rabie[1]

Introduction

Designing neural network (NN) to predict time series is not a trivial task. Some kind of science and art required to perform this task. One of the most important factors in building NN for time series prediction is the number of input nodes (Lags) [3]. Usually; the number of input nodes is chosen by the method "garbage in – garbage out" or trial and error method.

Problem Formulation

The inputs of the neural network are the past or lagged observations and the outputs are the predicted values (Figure1). Each input pattern is formed from a window of fixed length moved along the times series.
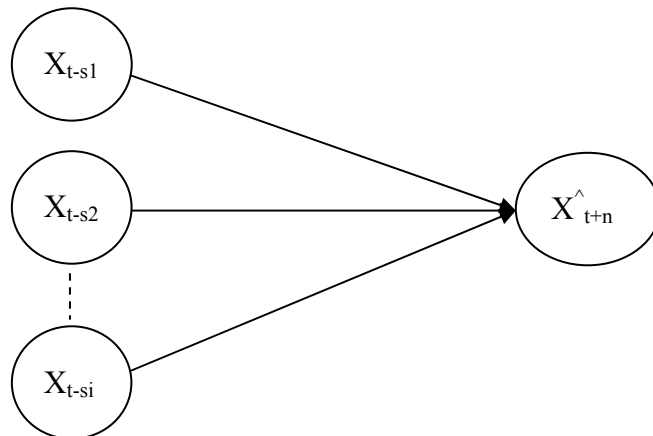

Figure 1: Neural Network for Time Series Prediction

The network represented by Figure 1 is a mapping function in the form:

$$\hat{X}_{t+n} = F(X_{t-s1}, X_{t-s2} \cdots, X_{t-si}) \qquad (1)$$

where $\hat{X}_{t+n}$ is the predicted output value of the neural network, and n prediction period ahead from the current period t; X_{t-s1}, X_{t-s2}, …, X_{t-si} are the actual inputs value

[1] M.Sc. Researcher, Decision Support Department, Faculty of Computers and Information, Cairo University. Email h_rabie@msn.com / hassan.rabie@gmail.com

of the neural network; S_i is the lag length from the current period t; F is a nonlinear function determined by the neural networks. The problem is to select the appropriate input variables $(X_{t-s1}, X_{t-s2}, \ldots, X_{t-si})$.

The Methodology

The correlation is a statistic indicator that measures the strength of the relationship between two variables; autocorrelation can serve the same purpose for a time series data, and can be calculated using next formula. [1]

$$r_k = \frac{\sum_{t=k+1}^{n}(X_t - \bar{X})(X_{t-k} - \bar{X})}{\sum_{t=1}^{n}(X_t - \bar{X})^2} \qquad (2)$$

Wei Huang [2] proposed an algorithm for input selection $\{X_{t-s1}, X_{t-s2}, \ldots, X_{t-si}\}$ depending on the idea; selecting the lagged variable which is more correlated to the predicted variable (output node) and less correlated to the already selected input variables. Figure 2 shows the steps of the algorithm.

The Implementation

1. Apply the algorithm in figure 2 on NN3 time series to obtain different time windows for different N (always N vary from 1 to 15 with step 2); i.e. for each time series, about 8 different time windows obtained. The reason for this large number is the care to scan all lags to understanding the time series.

2. For each time window obtained, text file created which contain the training and cross validation data.

3. For each time window (text file) build NN using NeuroSolutions Software (Demo Version 5.04) $\rightarrow$ NeuralExpert $\rightarrow$ Function Approximation[2].

4. The number of hidden nodes, learning rate and momentum constant obtained by trial and error.

---

[2] The data file is randomizing in NeuralExpert and divided into training file with size (80%) and cross validation cross file with size (20%)

5. Using Early Stopping Cross Validation as the stopping criteria for NN.

6. For each time series *select* time window (input selection) with minimum mean square error of cross validation.

7. Implement the selected model in step 6 to NN3 time series to obtain 18 predicted values required.

References

1 Spyros Makridakis, Steven C. Wheelwright, Rob J. Hyndman; "Forecasting methods and applications"; Third edition; John Wiley & Sons. Inc. 1998

2 Wei Huang, Shouyang Wang, Lean Yu, Yukun Bao, and Lin Wang; "A New Computational Method of Input Selection for Stock Market Forecasting with Neural Networks"; V.N. Alexandrov et al. (Eds.): ICCS 2006, Part IV, LNCS 3994, pp. 308 – 315, 2006; Springer-Verlag Berlin Heidelberg 2006

3 G. Peter Zhang, B. Eddy Patuwo, Michael Y. Hu: A simulation study of artificial neural networks for nonlinear time-series forecasting. Computers & OR 28(4): 381-396 (2001)

Set the initial parameters as follows:
  (1) Set the maximum lag period N.
  (2) Set the forecasting period ahead n.
  (3) Let $i=1$, $s_1 = 0$.

Search the next lag period s_{i+1} in the following way:

$$s_{i+1} = \arg \; si < j \le N \; \max \left(\frac{\left| r_{j+n} \right|}{\prod\limits_{K=1}^{i} \left| r_{j-s_k} \right|} \right)$$

Let $i=i+1$

Is s_i less than N-1?
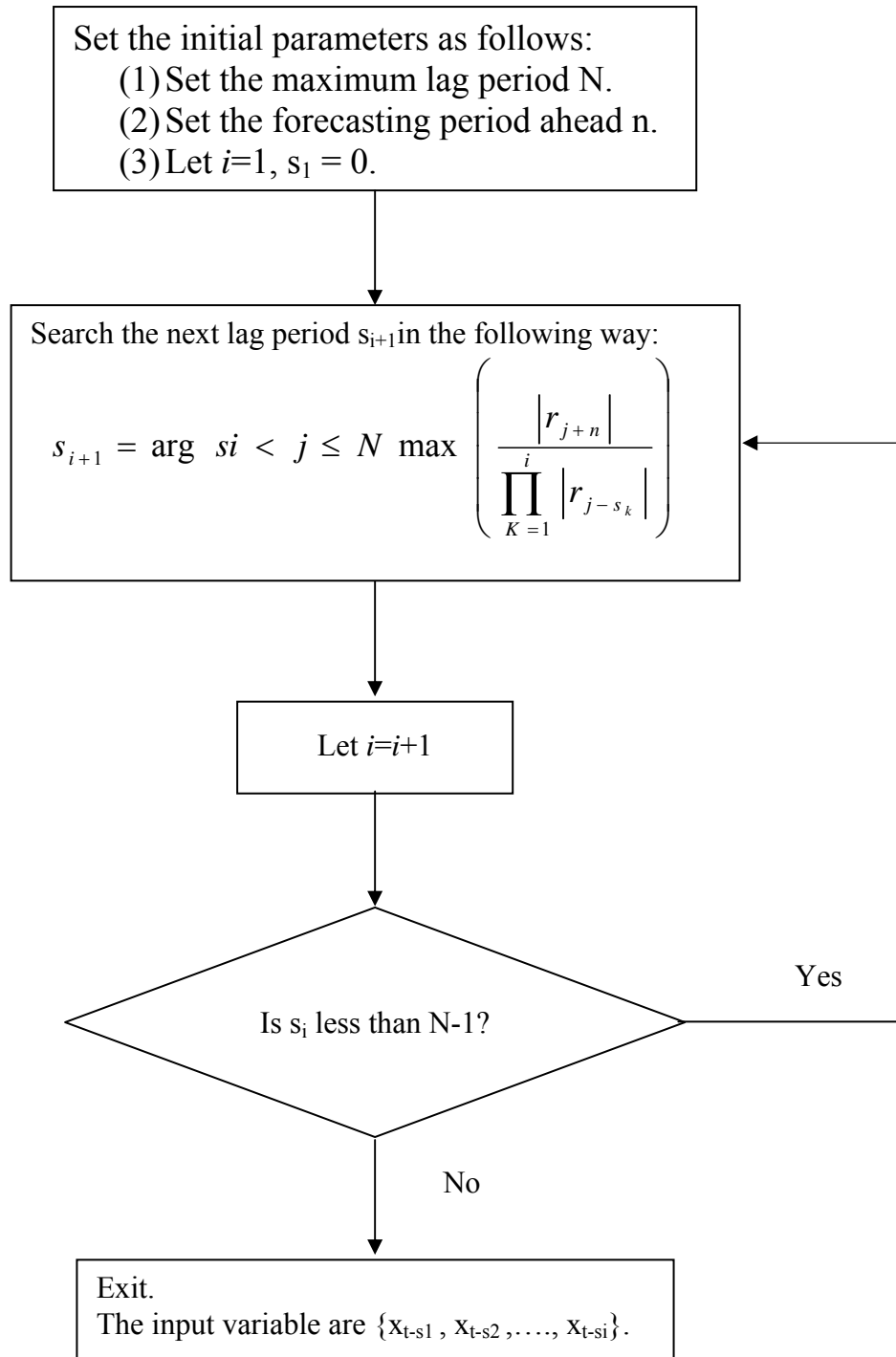
Yes

No

Exit.
The input variable are $\{x_{t-s1}, x_{t-s2}, \ldots, x_{t-si}\}$.

Figure 2: Input selection algorithm, [2]