QUANTITATIVE METHODS

QUANTITATIVE METHODS provide a forum for communication between statisticians and mathematicians with the users of applied statistical and mathematical techniques among a wide range of areas such as operations research, computing, actuary, engineering, physics, biology, medicine, agriculture, environment, management, business, economic, politics, sociology, and education. The journal will emphasize on both the relevance of numerical techniques and quantitative ideas in applied areas and theoretical development, which clearly demonstrate significant, applied potential.

Editor in Chief:
Karan P. Singh
Department of Biostatistics,
North Texas University, USA

Editorial Assistant:
Mustafa Usman
Department of Sciences
International Islamic University, Malaysia

Editors:
George A. Milliken
Department of Statistics
Kansas State University, USA

Warsono
Department of Mathematics
University of Lampung, Indonesia

Imad H. Khamis
Department of Mathematics
Southeast Missouri State University, USA

Feng, S.K.
Department of Statistics
Kansas State University, USA

Andreas Dress
Max Planck Institute of Math. in the Sciences
Leipzig, Germany

Pietro N. Juho
Department of Mathematics,
Natal University, South Africa

Subanar
Department of Mathematics
Gajah Mada University, Indonesia

Samsy Mohamad
Department of Mathematics
University Brunei Darussalam,
Brunei Darussalam

Jaetsung Choi
Department of Mathematics
Keimyung University, Taegu, S.Korea

Noor Akma Ibrahim
Department of Mathematics
University Putra Malaysia, Malaysia

Maher Taka
Faculty of Planning and Management
Al-Balqa Applied University, Al-Salt, Jordan

Ghulam Hussain
Department of Statistics
The University of Karachi, Pakistan

Konstantin Khanin
Department of Mathematics
University of Toronto, Canada

B. Wiwatpanapataphee
Department of Mathematics
Mahidol University, Thailand

Edi Soewono
Department of Mathematics
Bandung Institute of Technology, Indonesia

Tahir Ahmad
Department of Mathematics
University Technology Malaysia,
Malaysia
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Title</th>
<th>Authors</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step-stress Test using the Non-homogeneous Poisson Process for Repairable System and its Applications in Agriculture</td>
<td>Xiao Yang and Imad Khamis</td>
<td>1</td>
</tr>
<tr>
<td>Model Selection in Neural Networks by Using Inference of $R^2$ Incremental, PCA and SIC Criteria for Time Series Forecasting</td>
<td>Suhartono, Subanar and S. Gurtiño</td>
<td>9</td>
</tr>
<tr>
<td>Quality Control Using Survival Model Based on Reliability Application</td>
<td>Faiz A.M. Elfaki, Bin Daud. I, N.A. Ibrahim, M.Y. Abdullah</td>
<td>21</td>
</tr>
<tr>
<td>A Mathematical Model for Election Timing</td>
<td>Dharma Lesmono</td>
<td>38</td>
</tr>
<tr>
<td>Comparison of Some Methods of Testing Hypotheses in Unbalanced Split Plot Design</td>
<td>Mustofa Usman, Faiz A.M. Elfaki, Jamal I. Daoud, Agung E. Hadi and Rudi Ruswandi</td>
<td>50</td>
</tr>
<tr>
<td>Analysis of Missing Data Based on Estimable Parameter Function</td>
<td>Jamal I. Daoud, Mustofa Usman, Raihan Othman and Ahmad Sidik</td>
<td>62</td>
</tr>
<tr>
<td>Curve and Surface Constructions Using Non-Uniform B-Spline</td>
<td>Mohm. Naim Hj. Awang</td>
<td>75</td>
</tr>
</tbody>
</table>
MODEL SELECTION IN NEURAL NETWORKS BY USING INFEERENCE OF R² INCREMENTAL, PCA AND SIC CRITERION FOR TIME SERIES FORECASTING

Suhartono¹, Subanar² and S. Gurnian²
¹Department of Statistics, Institut Teknologi Sepuluh Nopember, Indonesia
PhD Candidate, Department of Mathematics, Gadjah Mada University
suhartono@statistika.its.ac.id; har_arena@yahoo.com
²Department of Mathematics, Gadjah Mada University, Indonesia
subanar@yahoo.com

Abstract. The aim of this paper is to discuss and propose a procedure for model selection in neural networks for time series forecasting. We focus on the model selection strategies based on statistical concepts, particularly on the inference of R² incremental, Principal Component Analysis (PCA) of the residual model and SIC criterion. In this paper, we employ this new procedure in two main approaches for model selection in neural networks, those are bottom-up or forward approach which starts with a large neural networks and top-down or backward approach which begins with an empty model. We use simulation as a case study. The results show that statistical inference of R² incremental combined with SIC criterion is an effective procedure for model selection in neural networks for time series forecasting.

Key words: Neural networks, model selection, statistical inference, time series forecasting

1. Introduction

In recent years, an impressive array of publications has appeared claiming considerable successes of neural networks (NN) in data analysis and engineering applications. NN model is a prominent example of such a flexible functional form. The use of the NN model in applied work is generally motivated by a mathematical result stating that under mild regularity conditions, a relatively simple NN model is capable for approximating any Borel-measurable function to any given degree of accuracy (see e.g. [5, 6, 15]).

In the application of NN, it contains limited number of parameters (weights). How to find the best NN model, i.e. how to find an accurate combination between number of input variables and unit nodes in hidden layer, is a central topic on the some NN literatures that discussed on many articles and books (see e.g. [1, 4, 11]).

In general, there are two procedures usually used to find the best NN model (the optimal architecture), those are “general-to-specific” or “top-down” and “specific-to-general” or “bottom-up” procedures. “Top-down” procedure is started from complex model and then applies an algorithm to reduce number of parameters by using some stopping criteria, whereas “bottom-up” procedure works from a simple model. The first procedure in some literatures is also known as “pruning” (see [10]), or “backward” method in statistical modeling. The second procedure is also known as “constructive learning” and one of the most popular is “cascade correlation” (see e.g. [2, 9]), and it can be seen as “forward” method in statistical modeling.
The aim at this paper is to discuss and compare the backward procedure proposed by Kaskoek and Van Dijk [7] and the new forward procedure by combining the inference of $R^2$ increment and SIC (Schwarz Information Criteria) criterion. We emphasize on the used of NN for time series forecasting.

2. Feedforward Neural Network

Neural networks (NN) are a class of flexible nonlinear models that can discover patterns adaptively from the data. Theoretically, it has been shown that given an appropriate number of nonlinear processing units, NN can learn from experience and estimate any complex functional relationship with high accuracy. Empirically, numerous successful applications have established their role for pattern recognition and time series forecasting.

Feedforward Neural Networks (FFNN) is the most popular NN models for time series forecasting applications. Figure 1 shows a typical three-layer FFNN used for forecasting purposes. The input nodes are the previous lagged observations, while the output provides the forecast for the future values. Hidden nodes with appropriate nonlinear transfer functions are used to process the information received by the input nodes.

![Figure 1. Architecture of neural network model with single hidden layer](image)

The model of FFNN in figure 1 can be written as

$$y_t = \beta_0 + \sum_{j=1}^{q} \beta_j f \left( \sum_{i=1}^{p} y_{ij} y_{t-i} + y_{oj} \right) + \epsilon_t,$$

where $p$ is the number of input nodes, $q$ is the number of hidden nodes, $f$ is a sigmoid transfer function such as the logistic.
\[ f(x) = \frac{1}{1 + e^{-x}}, \quad (2) \]

\{\beta_j, j = 0, 1, \ldots, q\} is a vector of weights from the hidden to output nodes and \{\gamma_i, i = 0, 1, \ldots, p; j = 1, 2, \ldots, q\} are weights from the input to hidden nodes. Note that equation (2) indicates a linear transfer function is employed in the output node.

Functionally, the FFNN expressed in equation (2) is equivalent to a nonlinear AR model. This simple structure of the network model has been shown to be capable of approximating arbitrary function (see e.g. [5, 6, 15]). However, few practical guidelines exist for building a FFNN for a time series, particularly the specification of FFNN architecture in terms of the number of input and hidden nodes is not an easy task.

Kaashoek and Van Dijk [7] introduced a “pruning” procedure by implementing three kinds of methods to find the best FFNN model; those are incremental contribution \(R^2\) incremental), principal component analysis, and graphical analysis. Whereas, Swanson and White [13, 14] applied a criterion of model selection, SIC, on “bottom-up” procedure to increase number of unit nodes in hidden layer and input variables until finding the best FFNN model.

2.1. Incremental Contribution through \(R^2\)

Kaashoek and Van Dijk [7] stated that a natural candidate for quantification of the network performance is the square of the correlation coefficient of \(y\) and \(\hat{y}\)

\[ R^2 = \frac{(\hat{y}'y)^2}{(y'y)(\hat{y}'\hat{y})}, \quad (3) \]

where \(\hat{y}\) is the vector of network output points. The network performance with only one unit hidden cell deleted can be measured in a similar way. For instance, if the contribution of hidden cell \(h\) is put to zero \((\beta_h = 0)\), then the network will produce an output \(\hat{y}_{-h}\) with errors

\[ e_{-h} = y - \hat{y}_{-h}, \quad (4) \]

This reduced network can be measured by the square of the correlation coefficient \(R^2_{-h}\) between \(y\) and \(\hat{y}_{-h}\), i.e.

\[ R^2_{-h} = \frac{(\hat{y}'_h y)^2}{(y'y)(\hat{y}_h'\hat{y}_{-h})}, \quad (5) \]

Now the \(R^2\) incremental contribution of unit hidden cell \(h\) is given by

\[ R^2_{(h)} = R^2 - R^2_{-h}. \quad (6) \]

The same procedure can be applied to reduce the number of input layer cells. In this case, \(\{\hat{y}_{1,i}(t)\}\) is network output, given network parameter estimates, without input cell \(i\). The contribution of unit input cell \(i\) is put to zero \((\gamma_{ih} = 0, \quad (7)\)
where $i = 1, 2, \ldots, p$; $h = 1, 2, \ldots, q$, then the reduced network can be quantified by the square of the correlation coefficient $R^2_i$ between $y$ and $\hat{y}_{-i}$ with

$$R^2_i = \frac{(\hat{y}_i - y)^2}{(y'(y)(\hat{y}_i - \hat{y})^2}.$$  \hfill (7)

The $R^2$ incremental contribution of input cell $i$ is measured as

$$R^2_{(i)} = R^2 - R^2_{-i}. \hfill (8)$$

The relative value of $R^2$ incremental contribution can be used in evaluating whether an input or unit hidden cell can be omitted or not.

2.2. Incremental contribution through Principal Component Analysis of network residuals

Kaashoek and Van Dijk [7] wrote that for the hidden layer cells we can define a matrix

$$E_{-H} = (e_{-1}, e_{-2}, \ldots, e_{-h}) \hfill (9)$$

where $e_{-h}$, $h = 1, 2, \ldots, q$ defined in equation (4). A principal component analysis on the matrix $E_{-H}$, i.e. the calculation of the orthonormal eigenvectors and eigenvalues of the symmetric matrix $E_{-H}'E_{-H}$, will give the principal components of $E_{-H}$. The first principal component, corresponding to the maximal eigenvalue, will have maximal variance since the amount of variance of each principal component is proportional to the corresponding eigenvalue.

Similarly, for the input layer cells, we define the matrix

$$E_{-I} = (e_{-1}, e_{-2}, \ldots, e_{-p}) \hfill (10)$$

where $e_{-i}$, $i = 1, 2, \ldots, p$ are defined as

$$e_{-i} = y - \hat{y}_{-i}. \hfill (11)$$

Kaashoek and Van Dijk [7] stated that cells (hidden and input) are candidates for exclusion if they have a minimal incremental contribution in $R^2$, measured by equation (3), and/or are the smallest component (in absolute sense) of the first principal component. Both quantities measure the contribution of one cell only. This procedure is descriptive and meant for exploratory data analysis.

2.3. Statistically inference of $R^2$ incremental contribution

In this paper we propose a new forward procedure based on the statistical inference of $R^2$ incremental contribution. This approach involves three basic steps, which we now describe in turn.

(i). Simple or Reduced model

We begin with the simple model considered to be appropriate for the data, which in this context is called the reduced or restricted model. In this case, we firstly evaluate the contribution of unit hidden cells. For the simple case, the reduced model is a linear model or NN model without hidden layer, i.e.
We fit this reduced model and obtain the error sum of squares, denoted by $SSE(R)$. (ii). Complex or Full model

Next, we consider the complex or full model, i.e. NN model as in equation (1). We start fitting NN model with single unit hidden cell or $q=1$. The error sum of squares of this full model denoted by $SSE(F)$. Here, we have:

$$SSE(F) = \sum (y - \hat{y}_F)^2.$$  \hspace{1cm} Full model (13)

(iii). Test Statistic

Kutner, Nachstein and Neter [8] stated when a large-sample test concerning several parameters (i.e. $\beta_j$ and $\gamma_j$ in equation (1)) model simultaneously is desired, we can use the same approach as for the general linear test. First, we fit the reduced model and obtaining $SSE(R)$, then fit the full model and obtaining $SSE(F)$, and finally calculate the test statistic:

$$F^* = \frac{SSE(R) - SSE(F)}{df_R - df_F} \cdot \frac{SSE(F)}{df_F}.$$  \hspace{1cm} (14)

For large sample size, $n$, this test statistic is distributed approximately as an $F(v1 = df_R - df_F, v2 = df_F)$ when $H_0$ holds, i.e. additional parameters in full model all equal to 0.

Gujarati [3] showed that equation (14) can be written in $R^2$ incremental contribution as

$$F^* = \frac{R^2_F - R^2_R}{df_F} \cdot \frac{(1 - R^2_R)}{df_F}.$$  \hspace{1cm} (15.a)

or

$$F^* = \frac{R^2_{(Incremental)}}{df_F} \cdot \frac{(1 - R^2_F)}{df_F}.$$  \hspace{1cm} (15.b)

We continue step 1 to 3 until the optimal unit hidden cells are found. Then, the forward procedure continues to find the optimal unit input cells. We start with the input which has the largest $R^2$. In this paper, we combine this test statistic with SIC criteria for determining the optimal cells.

3. Research Methodology

In this paper, these two procedures are compared by using a simulated data. The simulation experiment is carried out to show how the proposed NN modeling procedures work. Simulated data are generated as ESTAR (Exponential Smoothing Transition Autoregressive) model, i.e.

$$y_t = 6.5y_{t-1}.\exp(-0.25y_{t-1}^2) + u_t,$$  \hspace{1cm} (16)

where $u_t \sim \text{nid}(0,0.5^2)$. 
Time series and the lags plots of this simulated data can be seen in Figure 2. We can observe clearly that data follow nonlinear autoregressive pattern at lag 1.

![Simulated Data](image)

Figure 2. Time series and lags \( y_{t-1} \) and \( y_{t-2} \) plots of simulated data

4. Empirical Results

In this section the empirical results for both backward procedure introduced by Kaashoek and Van Dijk [7] and the proposed forward procedure are presented and discussed.

4.1. Results of backward procedure

We apply the backward procedure starting with a FFNN with six variable inputs \((y_{t-1}, y_{t-2}, \ldots, y_{t-6})\), one constant input and six unit hidden layer cells. The results of an optimization are reported in Table 1 and 2.
The results of the optimal unit hidden cells determination by implementing backward procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Unit Hidden</th>
<th>AIC</th>
<th>SBC</th>
<th>$R^2$</th>
<th>$R^2_{\text{iter}}$</th>
<th>PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6 units (6 inputs)</td>
<td>-176.403</td>
<td>-62.1157</td>
<td>0.98668</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>without h1</td>
<td></td>
<td></td>
<td>0.79924</td>
<td>0.18744</td>
<td>0.127</td>
</tr>
<tr>
<td></td>
<td>without h2</td>
<td></td>
<td></td>
<td>0.07076</td>
<td>0.91592</td>
<td>0.843</td>
</tr>
<tr>
<td></td>
<td>without h3</td>
<td></td>
<td></td>
<td>0.95844</td>
<td>0.02824</td>
<td>-0.043</td>
</tr>
<tr>
<td></td>
<td>without h4</td>
<td></td>
<td></td>
<td>0.44489</td>
<td>0.54179</td>
<td>-0.371</td>
</tr>
<tr>
<td></td>
<td>without h5</td>
<td></td>
<td></td>
<td>0.66586</td>
<td>0.32082</td>
<td>-0.365</td>
</tr>
<tr>
<td></td>
<td>without h6</td>
<td></td>
<td></td>
<td>0.98010</td>
<td>0.00658</td>
<td>0.017</td>
</tr>
<tr>
<td>2</td>
<td>4 units (1 input)</td>
<td>-159.071</td>
<td>-122.833</td>
<td>0.97545</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>without h1</td>
<td></td>
<td></td>
<td>0.95258</td>
<td>0.02288</td>
<td>-0.014</td>
</tr>
<tr>
<td></td>
<td>without h2</td>
<td></td>
<td></td>
<td>0.22184</td>
<td>0.75361</td>
<td>0.737</td>
</tr>
<tr>
<td></td>
<td>without h3</td>
<td></td>
<td></td>
<td>0.40196</td>
<td>0.57350</td>
<td>-0.675</td>
</tr>
<tr>
<td></td>
<td>without h4</td>
<td></td>
<td></td>
<td>0.88360</td>
<td>0.09185</td>
<td>0.041</td>
</tr>
<tr>
<td>3</td>
<td>2 units (1 input)</td>
<td>-157.277</td>
<td>-137.764</td>
<td>0.97246</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>without h1</td>
<td></td>
<td></td>
<td>0.36845</td>
<td>0.60588</td>
<td>0.762</td>
</tr>
<tr>
<td></td>
<td>without h2</td>
<td></td>
<td></td>
<td>0.24701</td>
<td>0.72732</td>
<td>-0.610</td>
</tr>
</tbody>
</table>

At the first step, with respect to hidden layer cells, comparing incremental contributions and the eigenvectors of $E_{-H}E_{-H}$, hidden layer cells 3 and 6 may be excluded. From Table 2, it is obvious that input cells 2, 3, ..., 6 or $(y_{t-2}, y_{t-3}, ..., y_{t-6})$ can be excluded. This gives a network with one input and four hidden layer cells, a FFNN(1,4). The graphs of network output with exclusion of one input are presented in Figure 2.

The results of the optimal unit inputs determination by implementing backward procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Input lags</th>
<th>AIC</th>
<th>SBC</th>
<th>$R^2$</th>
<th>$R^2_{\text{iter}}$</th>
<th>PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6 (lag 1-6)</td>
<td>-176.403</td>
<td>-62.1157</td>
<td>0.98668</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>without lag 1</td>
<td></td>
<td></td>
<td>0.00476</td>
<td>0.98192</td>
<td>0.997</td>
</tr>
<tr>
<td></td>
<td>without lag 2</td>
<td></td>
<td></td>
<td>0.96040</td>
<td>0.02628</td>
<td>0.042</td>
</tr>
<tr>
<td></td>
<td>without lag 3</td>
<td></td>
<td></td>
<td>0.97614</td>
<td>0.01053</td>
<td>0.052</td>
</tr>
<tr>
<td></td>
<td>without lag 4</td>
<td></td>
<td></td>
<td>0.97417</td>
<td>0.01251</td>
<td>-0.014</td>
</tr>
<tr>
<td></td>
<td>without lag 5</td>
<td></td>
<td></td>
<td>0.96432</td>
<td>0.02325</td>
<td>-0.029</td>
</tr>
<tr>
<td></td>
<td>without lag 6</td>
<td></td>
<td></td>
<td>0.97417</td>
<td>0.01251</td>
<td>-0.045</td>
</tr>
</tbody>
</table>
In Table 1 the results of a further optimization run (step 2 and 3) are reported. Step 2 shows that hidden layer cells 1 and 4 can be excluded; see e.g. the remarkable pattern in the $R^2$ incremental and the eigenvectors of $E_{-H}E_{-H}$. Hence, the optimal size of the network is FFNN with one input cell, $y_{t-1}$, and two hidden layer cells.

4.2. Results of forward procedure

First, we apply the proposed forward procedure starting with a FFNN with six variable inputs ($y_{t-1}, y_{t-2}, \ldots, y_{t-6}$) and one constant input to find the optimal unit hidden layer cells. The result of an optimization steps are reported in Table 3.

**Table 3.** The results of the optimal unit hidden cells determination by implementing forward procedure

<table>
<thead>
<tr>
<th>Number of hidden cell</th>
<th>AIC</th>
<th>SBC</th>
<th>$R^2$</th>
<th>$R^2_{\text{incremental}}$</th>
<th>F test</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>258.65</td>
<td>278.17</td>
<td>0.1185</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>0</td>
<td>0.5354</td>
<td>0.16487</td>
<td>49.801</td>
<td>0.00000</td>
</tr>
<tr>
<td>2</td>
<td>185.79</td>
<td>210.88</td>
<td>0.9741</td>
<td>0.438765</td>
<td>218.78</td>
<td>0.00000</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>7</td>
<td>0.9776</td>
<td>0.003511</td>
<td>5</td>
<td>0.07391</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>0.9817</td>
<td>0.004034</td>
<td>1.869</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>144.99</td>
<td>97.610</td>
<td>0.9833</td>
<td>0.001657</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>146.53</td>
<td>76.849</td>
<td>0.9874</td>
<td>0.004073</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>154.46</td>
<td>62.481</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>149.87</td>
<td>35.587</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>167.61</td>
<td>31.025</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Figure 3. FFNN(6,6) network output without one input layer cell \((y_t-1, y_{t-2}, \ldots, y_{t-6})\) compared with actual data

Table 3 shows that two unit hidden layer cells are the optimal result and further optimization runs are not needed. The graphs of network output by adding one unit hidden cell are presented at Figure 3. Then, we continue an optimization to find the optimal input cells. The results are presented in Table 4. It shows that unit input 1, i.e. \(y_{t-1}\), is the optimal input cell of the network. Hence, this forward procedure yields the optimal network is FFNN with one input cell and two hidden unit layer cells or FFNN(1,2).
Figure 4. The network output by adding one unit hidden layer cell compared with actual data

Table 4. The results of the optimal unit inputs determination by implementing forward procedure

<table>
<thead>
<tr>
<th>Input</th>
<th>AIC</th>
<th>SIC</th>
<th>K</th>
<th>Inaccurate</th>
<th>Leers</th>
<th>Previous</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-157.277</td>
<td>-137.764</td>
<td>0.972463</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>215.720</td>
<td>235.233</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>252.966</td>
<td>272.478</td>
<td>0.383648</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>265.067</td>
<td>284.580</td>
<td>0.159330</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>266.389</td>
<td>285.902</td>
<td>0.070129</td>
<td>8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>259.082</td>
<td>278.594</td>
<td>0.059832</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1−2</td>
<td>-</td>
<td>-130.9003</td>
<td>0.973078</td>
<td>0.0006150</td>
<td>1.239546</td>
<td>0.293492</td>
</tr>
<tr>
<td>1−3</td>
<td>155.9878</td>
<td>-129.0907</td>
<td>0.972669</td>
<td>0.0002061</td>
<td>0.415399</td>
<td>0.661099</td>
</tr>
<tr>
<td>1−4</td>
<td>-</td>
<td>-129.1086</td>
<td>0.972669</td>
<td>0.0002100</td>
<td>0.423461</td>
<td>0.655830</td>
</tr>
<tr>
<td>1−5</td>
<td>154.1381</td>
<td>-128.5544</td>
<td>0.972673</td>
<td>0.0000835</td>
<td>0.168296</td>
<td>0.845319</td>
</tr>
<tr>
<td>1−6</td>
<td>154.1960</td>
<td>-130.3262</td>
<td>0.972547</td>
<td>0.0004859</td>
<td>0.979342</td>
<td>0.378779</td>
</tr>
<tr>
<td>1−7</td>
<td>-</td>
<td>-</td>
<td>0.972949</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1−8</td>
<td>-</td>
<td>-</td>
<td>0.972949</td>
<td>6</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
4.3. The Comparison Results

There are two main evaluations for the comparison results between backward and forward procedures, i.e. the final result of architecture FFNN and the number of running steps. In general, the results of this simulation study show that the optimal FFNN architecture yielded by both two procedures is the same, i.e. FFNN(1,2). This result is similar to the paper of Suhartono and Hadiyat [12].

The comparison on the number of running steps shows that forward procedure yields less running steps than backward procedure proposed by Kaashoek and Van Dijk [7]. The results in Table 1 and 2 show that backward procedure need 22 running steps, i.e. 15 running for determining the optimal unit hidden layer cells and 7 running for input layer cells. Whereas, we need 14 running steps to find the final results of forward procedure, i.e. 3 running for determining the optimal unit hidden layer cells and 11 running for input layer cells (see Table 3 and 4).

5. Conclusions

Based on the results at the previous sections, we can make two main conclusions, i.e.

(i). The forward procedure and backward procedure proposed by Kaashoek and Van Dijk [7] yield the same final FFNN architecture model.

(ii). Forward procedure gives less running steps particularly on the determination of the optimal unit hidden layer cells, whereas backward procedure yield less running steps on determining the optimal input layer cells.

In general, the results also show that both two procedures give an advantage for FFNN modeling, i.e. the building process of FFNN model is not a black box. Additionally, we can do further research by combining forward procedure (with implementing $R^2$ incremental inference) for determining the optimal unit hidden layer cells and backward procedure proposed by Kaashoek and Van Dijk [7] for determining the optimal input layer cells, particularly on the real time series data.

References


