Abstract

Combining multiple neural networks appears to be a very promising approach in improving neural network generalisation since it is very difficult, if not impossible, to develop a perfect single neural network. In the building of an aggregated neural network model, a number of individual networks are developed from different data sets and/or different training algorithms. In this paper, individual networks are developed from bootstrap re-samples of the original training and testing data sets. Instead of combining all the developed networks, this paper proposes two selective combination techniques: forward selection and backward elimination. These two techniques essentially combine those individual networks that, when combined, can significantly improve model generalisation. In forward selection, individual networks are gradually added into the aggregated network until the aggregated network error on the original training and testing data sets cannot be further reduced. In backward elimination, all the individual networks are initially aggregated and some of the individual networks are then gradually eliminated until the aggregated network error on the original training and testing data sets cannot be further reduced. The proposed techniques are applied to dynamic nonlinear process modelling and classification of diabetes database. Application results demonstrate that the proposed techniques can significantly improve model generalisation and perform better than aggregating all the individual networks and the heuristic selective combination method where networks with better performance on the training and testing data are selected.
Keywords: Multiple Neural Networks, Selective Combination of Neural Networks, Generalisation, Forward Selection, Backward Elimination.

1. Introduction

Artificial neural networks have been increasingly used in developing nonlinear models in industry and model robustness is one of the main criteria that need to be considered when judging the performance of neural network models [10, 24]. Model robustness is primarily related to the learning or training methods and the amount and representativeness of training data [1]. Even though neural networks have a significant capability in representing nonlinear functions, inconsistency of accuracy still seems to be a problem where neural network models may not perform well when applied to unseen data. Furthermore, advanced process control and supervision of industrial processes require accurate process models promoting investigations on the robustness of neural network models [27]. Lack of robustness in neural network models is basically due to the over-fitting and poor generalisation of the models (e.g. [3]). Therefore, many researchers have been investigating on how over-fitting can be alleviated through improving network learning algorithms or through combining multiple imperfect neural networks (e.g. [8, 14, 21, 22, 23, 26]). In view of improving network learning algorithms, a number of techniques have been developed like regularisation and early stopping (e.g. [7, 17]). Ohbayashi et al. [18] implemented a universal learning rule with second order derivatives to increase the robustness in neural network models.

Among those approaches for improving neural network generalisation, the combination of multiple neural networks seems to be very effective. Figure 1 shows how multiple neural networks are combined. The individual networks in Figure 1 model the same relationship and are developed from different data sets and/or different training algorithms. They can also have different structures. Instead of choosing the single “best” neural network model, all the individual neural networks are combined. There are a number of methods in combining the networks like stacked neural network and bootstrap aggregated network where multiple networks are created on bootstrap re-samples of the original training data [2, 4, 6, 23, 25, 29].

The idea of multiple neural networks was actually developed based on stacked generalisation which is a technique for combining different representations to improve the overall prediction accuracy [24]. The hierarchical mixture of neural networks is also considered as one of the methods for combining neural networks [11, 12]. Most of the methods for combining networks
so far are based on linear combination where individual networks are linearly combined (e.g. [8, 9, 22, 23]). The main objective of this approach is to improve the generalisation capability of the neural network models in such a way that it will guard against the failure of individual component networks. This is because of the fact that some of the neural networks will fail to deliver the correct results or output predictions due to network training converged to undesirable local minima, over-fitting of noise in the data, or the limited training data set (e.g. [8, 16]). In another word, combining a set of imperfect models (networks) can be thought of as a way of managing the recognised limitation of the individual models, each is known to have errors, but they are combined in such a way as to minimise the effect of these errors [21]. Two example techniques in linear combination methods are averaging and weighted averaging. Averaging is a common and simple approach in the combination of neural networks. Weighted averaging includes the principal component regression (PCR) and multiple linear regression approaches. Zhang [26] uses the PCR approach to determine the combination weights and applies aggregated neural networks in the inferential estimation of polymer quality. Nonlinear combination methods are also sometimes used in the combination of neural networks like the Demspter-Shafers belief based method, combining using rank based information, voting, order statistic and Tumer and Ghosh methods [21]. In the stacked generalisation approach [24], the outputs of individual networks, referred to as a set of level 0 generalisers, are used as the input to level 1 generaliser, which is trained to produce the appropriate output.

In most of the reported works on aggregating multiple neural networks, all the developed individual networks are combined. However, some neural networks may not contribute to improving model prediction performance when combined with other networks. This could be due to several reasons, such as these networks severely over-fit the data or the information captured by these networks has already been represented by other networks included in the aggregated network. Excluding these networks could further improve the generalisation capability of the aggregated network. Perrone and Cooper [19] suggest a heuristics selection method whereby the trained networks are ordered in terms of increasing mean squared errors (MSE) and only those with lower MSE are included in combination. However, combining these networks with lower MSE may not significantly improve model generalisation since these networks can be severely correlated. Zhou et al. show that combining selected networks may be better than combining all individual networks and propose a genetic algorithm based approach for selecting individual networks in an ensemble [30]. In this paper, techniques
based on the forward selection (FS) and backward elimination (BE) methods in statistical regression [13] are proposed for selective combination of neural networks. In the first approach, individual networks are gradually added to the aggregated or ensemble network until the sum of squared errors (SSE) on the original training and testing data cannot be further reduced by adding more networks. In the second approach, all the individual networks are initially aggregated and then the individual networks are removed one at a time until the SSE on the training and testing data cannot be further reduced by removing more networks.

The paper is organised as follows. Section 2 presents the FS and BE selective combination methods for aggregating multiple neural networks. Section 3 presents three case studies to test the proposed techniques. Some results and discussions on the three case studies are given in Section 4. Finally, the last section concludes this paper.

2. Selective Combination of Multiple Neural Networks

Suppose that neural network models are to be developed from the data set \( \{X, Y\} \), where \( X \in \mathbb{R}^{N \times p} \) is the input data, \( Y \in \mathbb{R}^{N \times q} \) is the desired output data, \( N \) is the number of samples, \( p \) is the number of input variables, and \( q \) is the number of output variables. To develop an aggregated neural network model containing \( n \) individual networks, the original data set can be re-sampled using bootstrap re-sampling with replacement [5] to form \( n \) replications of the original data set [25]. The \( n \) replications can be denoted as \( \{X^{(1)}, Y^{(1)}\}, \{X^{(2)}, Y^{(2)}\}, \ldots, \{X^{(n)}, Y^{(n)}\} \), where \( X^{(i)} \in \mathbb{R}^{N \times p}, Y^{(i)} \in \mathbb{R}^{N \times q}, i=1, 2, \ldots, n \). A neural network model can be developed on each of these replications, which can be partitioned into a training data set and a testing data set if cross-validation is used in network training and network structure selection. If the predictions of these \( n \) networks on the original data set are denoted as \( \hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_n \), then the SSE of the \( i \)th network can be calculated as

\[
SSE_i = \text{trace}[(Y - \hat{Y}_i)(Y - \hat{Y}_i)^T]
\]

For the sake of simplicity in illustration, the simple average method is used in combining the selected networks. If all \( n \) networks are combined, then the aggregated network output is:

\[
\hat{Y} = \frac{1}{n} \sum_{i=1}^{n} \hat{Y}_i
\]

2.1 Forward selection
In this technique, the individual networks are added one at a time to the aggregated network. The network selected for inclusion at each step is the one that, when combined with previously added networks, produces the greatest decrease in model prediction SSE.

The process begins with an empty model and networks are added one at a time to form an aggregated neural network model. The first network to be added is the “best” individual network, i.e. the one with the smaller SSE on the original training and testing data. The second network added is the one, when combined with the first added network, produces the largest reduction in SSE on the original training and testing data. Note that the individual networks are developed from bootstrap re-samples of the original training and testing data, reductions in the SSE on the original training and testing data represent improved generalisation. This procedure is repeated until the SSE on the training and testing data cannot be further reduced by adding more networks.

The FS procedure can be summarised as follows:

**Step 1.** Generate \( n \) replications of the original data set using bootstrap re-sampling, \( \{X_{(1)}, Y_{(1)}\}, \{X_{(2)}, Y_{(2)}\}, \ldots, \{X_{(n)}, Y_{(n)}\} \), and develop a neural network on each replication. Denote the prediction of the \( i \)th network on the original data set as \( \hat{Y}_i \). Calculate the SSE of these networks on the original data using Eq(1).

**Step 2.** Set \( j = 1 \) and denote \( T \) as a set containing the indices of all networks and \( T = [1, 2, \ldots, n] \). Denote \( \hat{Y}_{a,j} \) and \( SSE(j) \) as, respectively, the predictions and SSE of the aggregated network at stage \( j \). Select the network with the smallest SSE on the original data as the first network to be added to the ensemble of networks:

\[
k = \arg \min_{i \in T} SSE_i
\]

\[
SSE(j) = SSE_k
\]

\[
\hat{Y}_{a,j} = \hat{Y}_k
\]

Denote \( I \) as a set containing the indices of the networks in the aggregated network and \( I = [k] \). Denote \( J \) a set containing the indices of the networks not currently included in the aggregated network and \( J = T - I \), i.e. \( J \) is obtained by deleting the elements in \( I \) from \( T \).
Step 3. If \( n-j=0 \), then go to Step 5;
else
\( j=j+1 \)
for \( i \in J \)
\[
\hat{Y}_{a,j}^{(i)} = [(j-1)\hat{Y}_{a,j-1} + \hat{Y}_i] / j
\]
end
\[
k = \arg \min_{i \in J} \text{trace}[(\hat{Y}_{a,j}^{(i)} - Y)^T (\hat{Y}_{a,j}^{(i)} - Y)]
\]
\[
SSE(j) = \text{trace}[(\hat{Y}_{a,j}^{(k)} - Y)^T (\hat{Y}_{a,j}^{(k)} - Y)]
\]

Step 4. If \( SSE(j) \geq SSE(j-1) \), then go to Step 5;
else
\( I=I, k \) (i.e. add \( k \) to set \( I \))
\( J=T-I \)
go to Step 3.

Step 5. Stop.

When the FS procedure terminates, the indices of the selected networks in the aggregated network are contained in set \( I \) and those not being selected are contained in set \( J \).  

2.2 Backward elimination

The BE approach begins with the aggregated neural network containing all the individual networks and removes one network at a time until the SSE on the training and testing data cannot be further reduced. The network deleted at each step is such selected that its deletion results in the largest reduction in the aggregated network SSE on the training and testing data.

The BE method is summarised as follows:

Step 1. Generate \( n \) replications of the original data set using bootstrap re-sampling, \( \{X_{(1)}, Y_{(1)}\}, \{X_{(2)}, Y_{(2)}\}, \ldots, \{X_{(n)}, Y_{(n)}\} \), and develop a neural network on each replication. Denote the prediction of the \( i \)th network on the original data set as \( \hat{Y}_i \). Calculate the SSE of these networks on the original data using Eq(1).
Step 2. Set $j=1$ and denote $I$ as a set containing the indices of the networks currently included in the aggregated network and $I=[1, 2, \ldots, n]$. Denote $J$ as a set containing the indices of the networks currently deleted from the aggregated network and $J=[]$, i.e. $J$ is initially empty. Denote $\hat{Y}_{a,j}$ and $\text{SSE}(j)$ as, respectively, the predictions and SSE of the aggregated network at stage $j$.

$$\text{SSE}(j) = \text{trace}\left(\frac{1}{n} \sum_{i \in I} \hat{Y}_i - Y\right)^T \left(\frac{1}{n} \sum_{i \in I} \hat{Y}_i - Y\right)$$

Step 3. If $n-j=0$, then go to Step 5;
else
\begin{align*}
j &= j+1 \\
\text{for } i \in I \\
\hat{Y}^{(i)}_{a,j} &= \frac{1}{n-j} \sum_{i \in I-1} \hat{Y}_i \\
\text{end}
\end{align*}
\begin{align*}
k &= \text{arg \: min } \text{trace}\left(\hat{Y}^{(i)}_{a,j} - Y\right)^T \left(\hat{Y}^{(i)}_{a,j} - Y\right) \\
\text{SSE}(j) &= \text{trace}\left(\hat{Y}^{(k)}_{a,j} - Y\right)^T \left(\hat{Y}^{(k)}_{a,j} - Y\right)
\end{align*}

Step 4. If $\text{SSE}(j) \geq \text{SSE}(j-1)$, then go to Step 5;
else
\begin{align*}
I &= I - k \quad \text{(i.e. remove } k \text{ from set } I) \\
J &= [J, k] \\
\text{go to Step } 3.
\end{align*}

Step 5. Stop.

Remark. The FS and BE selective combination approaches might lead to a suboptimal model. For example, in the FS approach, the best network on the training and testing data is guaranteed to be selected. However, it may not be required if the combination of other networks are considered. To overcome this problem, a combined FS and BE approach could be adopted. In the combined approach, BE is carried out after each step of FS to see if any earlier selected networks can be eliminated due to the later addition of other networks. It should be realised that, although the FS and BE approaches might not lead to an optimal
model, they are computationally much more efficient than an exhaustive search considering all possible combinations.

3. Modelling Case Studies

Two case studies, one on regression and one on classification, were used for testing the selective combination schemes presented in this paper. In each of the case studies, an aggregated neural network model was developed through combining networks selected from 20 individual neural networks. Two situations were considered: individual networks with fixed identical structure and individual networks with various structures. The individual networks were trained by the Levenberg-Marquardt optimisation algorithm with regularisation and “early stopping”. Weights and biases were initialised as small random values. The individual networks are single hidden layer feed forward neural networks. Hidden neurons use the sigmoid activation function whereas output layer neurons use the linear activation function for regression and sigmoid function for classification. To cope with different magnitudes in the input and output data, all the data were scale to zero mean and unit standard deviation. The data for neural network model building need to be divided into: 1). Training data (for network training); 2). Testing data (for cross-validation based network structure selection and early stopping); and 3). Unseen validation data (for evaluation of the final selected model). In networks with fixed structure, the network structures, i.e. the number of hidden neurons, were determined through cross validation. Single hidden layer neural networks with different numbers of hidden neurons were trained on the training data and tested on the testing data. The network with the lowest SSE on the testing data was considered as having the best network topology. In assessing the developed models, SSE on the unseen validation data is used as the performance criterion for regression and classification accuracy on the unseen validation data is used as the performance criterion for classification.

To test the performance of the proposed selective combination schemes, the combination schemes listed in Table 1 are investigated. In schemes 5 and 6, the 20 individual networks were sorted based on their SSE on the training and testing data. Then the 10 networks with low SSE on the training and testing data (the better half according to the performance on the training and testing data) were combined. This type of heuristic selective combination was suggested by Perrone and Cooper [19].
In order to demonstrate the capability of the proposed methods, Monte Carlo simulations of 20 runs with different initial network weights were carried out. These different initial weights were generated using different seeds in the MATLAB random number generator and applying different scaling factors to the generated random numbers.

### 3.1 Case study 1: modelling of pH in a neutralisation process

The neutralisation process takes place in a CSTR and there are two input streams to the CSTR. One is acetic acid of concentration $C_1$ at flow rate $F_1$ and the other is sodium hydroxide of concentration $C_2$ at flow rate $F_2$ [15].

The mathematical equations of the CSTR can be described as follows by assuming that the tank level is perfectly controlled:

\[
\frac{d\zeta}{dt} = F_1 C_1 - (F_1 + F_2)\zeta
\]

\[
\frac{d\varsigma}{dt} = F_2 C_2 - (F_1 + F_2)\varsigma
\]

\[
[H^+]^0 + (Ka + \zeta)[H^+] + [Ka(\varsigma - \zeta) - Kw[H^+] - KwKa = 0
\]

\[
pH = \log_{10} [H^+]
\]

where

\[
\zeta = [HAC] + [AC^-]
\]

\[
\varsigma = [Na^-]
\]

The meanings and nominal values of the variables in the above equations are given in Table 2. These equations show that the dynamic relationship between the titration flow and pH in the CSTR is very nonlinear. To generate training, testing and validation data, multi-level random perturbations were added to the flow rate of acetic acid while other inputs to the reactor were kept constant. The generated training, testing, and validation data sets each contains 200 samples.

The pH measurements were corrupted with normally distributed random noise with zero mean and a standard deviation of 0.2. The dynamic model representing the neutralisation process is of the form:

\[
y(t) = f[y(t-1), y(t-2), u(t-1), u(t-2)]
\]
where \( y(t) \) is the pH in the reactor at time \( t \) and \( u(t) \) is the acid flow rates at time \( t \).

3.2 Case study 2: Pima Indians Diabetes Database

This case study is on the pattern classification of diabetes data. The diagnostic, binary-valued variable, investigated is whether the patient shows signs of diabetes according to World Health Organization criteria (i.e., if the 2 hour post-load plasma glucose was at least 200 mg/dl at any survey examination or if found during routine medical care). The case study is taken form the Indian community or population lives near Phoenix, Arizona, USA [20], and the data is available from the UCI Machine Learning Repository (www.ics.uci.edu/~mlearn/MLRepository.html).

The data contain 768 samples and are divided into a training data set, a testing data set and an unseen validation data set each consisting of 256 samples. There are 9 variables:

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m) \(^2\))
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

A value of 1 in attribute number 9 is interpreted as “tested positive for diabetes” whereas a value of 0 means “tested negative for diabetes”. The task of pattern classification is to classify the pattern for positive and negative diabetes based on the 8 input variables (attribute number 1 to number 8). The model representing the classification is in the form:

\[
y = f[u_1, u_2, \ldots, u_8]
\]  

(10)

where \( y \) is the class attribution taking the binary value 0 or 1, and \( u_1 \) to \( u_8 \) are the 8 inputs. Since the output is a binary value, sigmoid function is used as the activation function in the output layer neuron. The weights and biases were randomly initialised in the range from –0.4 to 0.4. The performance criterion used in the FS and BE methods is based on the classification
accuracy. For classification, the minimisation of SSE in the FS and BE algorithms is replaced by the maximisation of classification accuracy.

4. Results and Discussions

4.1 Case study 1

It is well known that the dynamics of pH is highly nonlinear. In this case study 20 networks with fixed number of hidden neurons (5) and 20 networks with varying number of hidden neurons (between 1 and 10) were developed. In the fixed structure, the number of hidden neurons was determined through cross validation.

Figure 2 shows the mean SSEs with 95% confidence intervals for one-step-ahead predictions on the unseen validation data from the 20 runs. Figure 2 clearly shows that the proposed FS and BE approaches give better performance compare to the average and the median of all the single networks and the heuristic selective combination method suggested by Perrone and Cooper [19]. It can be seen from Figure 2 that the upper confidence bounds of the proposed FS and BE selective combination methods are lower than the lower confidence bounds of other combination schemes. This indicates that the proposed FS and BE combination approaches would always give better performance than other combination schemes (schemes 1 to 6). The results also indicate that combining networks with various structures gives more consistent performance (smaller confidence intervals) than combining networks with fixed structures.

Figure 3 shows the one-step-ahead prediction performance of individual neural networks with various structures in run 1. It can be seen from Figure 3 that among the networks with fixed structure, network 11 gives the best performance on the training and testing data with an SSE of 22.31, but its performance on the unseen validation data is not the best. The best individual network performance on unseen validation data is from network number 2 with an SSE of 7.86 but its SSE on the training and testing data is 26.56 which are quite high. Similar observations can be made for networks with various structures and this demonstrates the non-robust nature of individual networks.

Figure 4 shows the SSE of one-step-ahead predictions from aggregated neural networks with various structures in run 1. It is interesting to observe from Figure 4 that the aggregated
networks under either selective combination scheme give quite consistent performance on the training and testing data and on the unseen validation data. Similar observation was also made for the aggregated neural networks with fixed structure. As indicated by the top two panels of Figure 4, the FS algorithm stops after adding 3 networks and the BE algorithm stops after eliminating 17 networks.

4.2 Case study 2
In this case study, 20 networks with fixed number of hidden neurons (10) and 20 networks with various numbers of hidden neurons (between 5 and 15) were developed. Figure 5 shows the mean classification accuracies and the corresponding 95% confidence intervals from the different combination schemes on the unseen validation data for the 20 runs. It can be seen from Figure 5 that the mean classification accuracy of FS and BE is higher than those of the other combination schemes. The lower confidence bounds of schemes 8 and 9 are higher than the upper confidence bounds of schemes 1 to 6 indicating the schemes 8 and 9 would generally always give better performance than schemes 1 to 6. For schemes 7 and 10, their lower confidence bounds are higher than the upper confidence bounds of schemes 1 to 4 but not higher than the upper confidence bounds of schemes 5 and 6. This indicates that schemes 7 and 10 would generally always give better performance than schemes 1 to 4 but may not give better performance than schemes 5 and 6 in some runs. However, since the mean classification accuracies of schemes 7 and 10 are higher than those of schemes 5 and 6, schemes 7 and 10 would generally give better performance than schemes 5 and 6.

5. Conclusions
Forward selection and backward elimination methods for the selective combination of multiple neural networks are proposed in this paper in order to improve the model generalisation performance. Instead of combining all neural networks, selective combination aims to obtain the maximal generalisation capability by combining selected individual networks. In the FS method, the “best” individual network is selected as the first network in that aggregated networks and then other networks are gradually added until the aggregated network error on the original training and testing data cannot be further reduced. In the BE method, initially all individual networks are included in the aggregated network. Individual networks are then eliminated one at a time from the aggregated network until the aggregated
network error on the original training and testing data cannot be further reduced. Both selective combination methods have shown their superiority compared to the combination of all networks, the median of all networks, and the heuristic selective combination method where networks with better performance on the training and testing data are selected. The selective combination schemes provide models with better generalisation capability in that the performance on the training and testing data is quite consistent to that on the unseen validation data. Results from the two case studies indicate that the proposed methods work better for regression than for classification.

Acknowledgement: This work was supported by University Science Malaysia (USM) (for Z. Ahmad) and UK EPSRC through the Grant GR/R10875 (for J. Zhang).

References

Table 1. Combination schemes

1  Median of all the individual networks with fixed structure
2  Median of all the individual networks with various structures
3  Average of all the individual networks with fixed structure
4  Average of all the individual networks with various structures
5  Average of the 10 networks with fixed structure having low SSE on the training and testing data
6  Average of the 10 networks with various structures having low SSE on the training and testing data
7  Average of selected networks with fixed structure using the FS method
8  Average of selected networks with various structures using the FS method
9  Average of selected networks with fixed structure using the BE method
10 Average of selected networks with various structures using the BE method

Table 2. Physical parameters used in the simulation

<table>
<thead>
<tr>
<th>Variables</th>
<th>Meanings</th>
<th>Nominal values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V$</td>
<td>volume of the tank</td>
<td>1 L</td>
</tr>
<tr>
<td>$F_1$</td>
<td>flow rate of acid</td>
<td>0.081 L/min</td>
</tr>
<tr>
<td>$F_2$</td>
<td>flow rate of base</td>
<td>0.512 L/min</td>
</tr>
<tr>
<td>$C_1$</td>
<td>concentration of acids in $F_1$</td>
<td>0.32 mol/L</td>
</tr>
<tr>
<td>$C_2$</td>
<td>concentration of acids in $F_2$</td>
<td>0.05 mol/L</td>
</tr>
<tr>
<td>$K_a$</td>
<td>acid equilibrium constant</td>
<td>$1.8 \times 10^{-5}$</td>
</tr>
<tr>
<td>$K_w$</td>
<td>water equilibrium constant</td>
<td>$1.0 \times 10^{-14}$</td>
</tr>
</tbody>
</table>
Figure 1. An aggregated neural network

Figure 2. Mean SSEs and their 95% confidence intervals on the unseen validation data
Figure 3. SSE of one-step-ahead predictions from individual neural networks with various structures in run 1 of Case Study 1

Figure 4. SSE of one-step-ahead predictions from aggregated neural networks with various structures in run 1 of Case Study 1
Figure 5. Mean classification accuracies with their 95% confidence intervals on the unseen validation data