Jurnal Teknologi, 48(F) Jun 2008: 99-116

IMPROVING NONLINEAR PROCESS MODELLING THROUGH SELECTIVE COMBINATION OF MULTIPLE NEURAL NETWORKS USING COMBINED CORRELATION COEFFICIENT ANALYSIS

ZAINAL AHMAD[1*] & RABIATUL 'ADAWIAH MAT NOOR[2]

Abstract. This paper proposed a selective combination method based on combined correlation coefficient analysis to increase the robustness of the single neural network. The main objective of the proposed approach is to improve the generalisation capability of the neural network models by combining networks that are less correlated. The assumption that we made is that combining networks that are highly correlated might not improve the final prediction performance due to the fact that these networks present the same contribution to the final prediction. This might even deteriorate the robustness of the combined network. The result shows that combination multiple neural networks using the proposed approach improved the performance of the two nonlinear process modelling case studies in which there is a significant reduction of validation sum square error (SSE) of the networks was obtained.

Keywords: Multiple neural networks, selective combination neural networks, correlation coefficient, nonlinear process modelling

1.0 INTRODUCTION

Artificial neural networks have been used in developing non-linear models in industry for such a long time [1] and robustness of the model is one of the main criteria that need to be considered. Robustness of the model can be defined as one of the baselines to judge the performance of the neural network models and it is really related to the learning or training classes as what has been described in reference [2]:

> *' The importance of neural networks in this context is that they offer very special powerful and very general framework for representing nonlinear mappings from several input variables to several output variables, where the form of the mapping is governed by a number of adjustable parameters.'*

Even though a neural network is one of the powerful tools in nonlinear modelling; inconsistency of accuracy is still the main glitch in implementing neural network models. Robustness is still a main problem where neural networks model cannot

[1&2] School of Chemical Engineering Campus, Universiti Sains Malaysia (USM), Seri Ampangan, 14300 Nibong Tebal, Penang, Malaysia

* Corresponding author:Tel : +604-593 7788 ext 6462, Fax : +604 594 1013, Email: chzahmad@eng.usm.my

cope or perform well when it is applied to a new unseen data. This problem is basically due to the overfitting of the models [3]. Therefore, combination of neural networks is employed where researchers concentrate on how overfitting can be avoided by improving the learning algorithm or by combining the neural networks. Many techniques can improve the generalisation capability of the neural network model such as regularisation techniques (e.g. [3-5]), Bayesian Learning (e.g. [6, 7]) and also by using the parsimonious networks structure [8]. Among those approaches, the combination of multiple neural networks shows some encouraging results in terms of improving the robustness of neural networks. Figure 1 shows how multiple neural networks are combined (e.g. [9-17]).

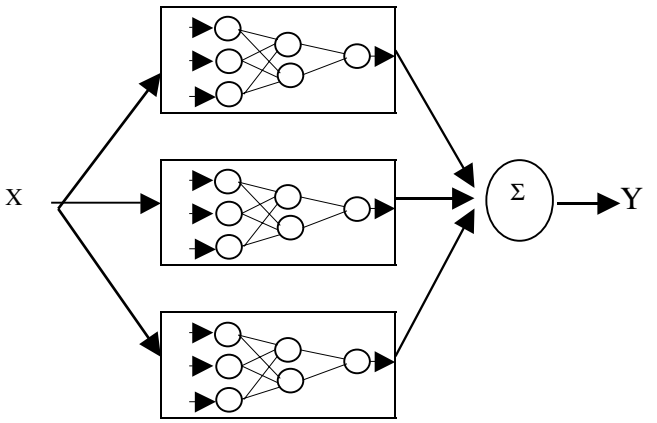

Figure 1 Multiple neural network

There are several methods in combining the individual networks like stacked neural network and bootstrap aggregated network or BAGNET where multiple networks are created on bootstrap re-samples of the original training data as shown in Figure 1 [14, 15, 17].

Sridhar *et al.* [14] also explained that the individual neural networks are trained using different training data set and/or from different initial weights. Then, the neural models are combined together to get better predictions of the outputs. Instead of choosing the best neural network model among the individual networks, all the individual neural networks are combined.

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 [1]. The hierarchical mixture of neural networks is also considered as one of the methods for combining neural networks [11,

18]. Most of the combinations of networks so far are based on linear combination [9, 10, 14, and 15].

The main objective of this approach is to improve the generalisation capability of single neural network model in such a way that it will guard against the failures of individual component networks when it is combined. This is because of the fact that some of the neural networks will or might fail to deliver the correct results or output predictions due to some problems such as limited training data set [3, 9].

This paper is structured as follows: Section 2 presents different methods for combining multiple neural networks and Section 3 for simulation and modelling overview. Two case studies are given in Section 4. Section 5 presents the results and discussions on these two case studies. Finally, some concluding remarks are drawn in Section 6.

2.0 DIFFERENT METHODS FOR COMBINING NEURAL NETWORKS

Method of combining multiple networks can be divided into two main approaches which are linear and nonlinear combination. Simple averaging and weighted averaging are the common linear combination approach while Demspter-Shafer method or using neural networks itself are the nonlinear combination methods. In this proposed case study, simple averaging and weighted averaging approaches are applied where the multiple neural outputs are linearly combined to create a single output as a final prediction. In weighted averaging method, weights for individual networks are calculated, for example, through multiple linear regression (MLR) or principal component regression (PCR).

2.1 Simple Averaging

This is the most common method in combining several model outputs with the weights fixed as shown below:

$$\hat{Y} = w_1\hat{y}_1 + w_2\hat{y}_2 + \ldots + w_n\hat{y}_n \tag{1}$$

where $\hat{y}_i$ is the network prediction from the i^{th} network, n is the number of networks to be combined, $\hat{Y}$ is the final prediction output, and $w_i = 1/n$ is the weight for combining the i^{th} network. The main disadvantage of this approach is that all the networks have the same contribution to the final prediction output even though some of the networks might have better predictions then others; consequently it might deteriorate the model.

2.2 Multiple Linear Regression (MLR)

MLR is also known as Ordinary Least Square (OLS) and it assumes that a vector of regression parameters, θ, can be used to determine the system output y, from the measured variables x, through a linear model of the following form;

$$y = x_1 \theta_1 + x_2 \theta_2 + ... + x_n \theta_n + e \tag{2}$$

By using MLR, the model parameter, θ, can be estimated through the following equation:

$$\hat{\theta} = (X^T X)^{-1} X^T Y \tag{3}$$

where X is the matrix of input variable data and Y is the matrix (or a vector in a single output case) output variable data. In this chapter, for MLR analysis, X will be the prediction output of the individual networks on the training and testing data and Y is the actual process output on the training and testing data. Once the model parameter is estimated, it can be applied to the unseen validation data as shown below:

$$\hat{Y} = \hat{y}_1 \hat{\theta}_1 + \hat{y}_2 \hat{\theta}_2 + ... + \hat{y}_n \hat{\theta}_n \tag{4}$$

where $\hat{Y}$ is the final prediction on unseen validation data while $\hat{y}$ is the prediction on unseen validation data and n is the number of neural networks for combination.

2.3 Principal Component Regressions (PCR)

PCR is one way to deal with the problem of ill-conditioned matrices or when the predictor variables are highly correlated. Instead of regressing the system properties on the original measured variables, the properties are regressed on the principle component scores of the measured variables (which are orthogonal and therefore well conditioned). The data matrix, $\hat{Y}_{(tr)}$ (which is the input based on the training data prediction output), will be decomposed into principal components as follows:

$$\hat{Y}_{(tr)} = t_1 p_1^T + t_2 p_2^T + ... + t_m p_m^T = TP^T \tag{5}$$

where t is the score vector (principal component) and p is the loading vector. The principal components are arranged in descending order of the data variations that they explain. PCR is a favourite tool for data compression and information

extraction. It finds combination of variables or factors that describe major trends in a data set. The predictor matrix, $\hat{Y}_{(tr)}$ is replaced by its scores matrix, T, which contains the major score vectors.

$$T = \hat{Y}_{(tr)} * P \tag{6}$$

Therefore

$$Y_{(tr)} = TB + E \tag{7}$$

where B is the vector for model parameter. The least square estimation of B is:

$$B = (T^T T)^{-1} T^T Y_{(tr)} \tag{8}$$

Therefore, the model parameter vector B can be changed into the model parameter vector, θ as follow:

$$\hat{\theta} = PB = P(T^T T)^{-1} T^T Y_{(tr)} \tag{9}$$

In this study, principal components (PC) are calculated using singular value decomposition (SVD) even though there are several methods that can be used to compute the PC. The main reason for using SVD is that it is easy to compute using Matlab™ and also easy to use. The SVD is presented below as Matlab™ command:

$$\begin{aligned} [U, S, V] &= \text{svd}\,(\hat{Y}_{(tr)}) \\ P &= V \\ T &= \hat{Y}_{(tr)} * V \end{aligned} \tag{10}$$

where $\hat{Y}_{(tr)}$ is the output prediction based on the training data, T is the principal component and P is the loading matrix.

Selections of the number of principal components used are determined through the cross validation technique. The PCR models are developed using the training data and tested on the testing data and the model giving the least model error on the testing data is selected. Evaluation of the PCR model will be based on the performance on the unseen validation data [17].

Another method is like stacked generalisation which is introduced by Wolpert [1]. They generalised the idea of combination by combining the networks with weights

that vary over the feature space. The outputs from a set of Level 0 generalisers are used as the inputs to Level 1 generaliser, which is trained to produce the appropriate output. Other method of combination is selective combination approach. The objective behind these selective combination approaches is to reduce the number of shared failure when combining the networks [13]. There are a number of methods on how to select a proper network for combination. Perrone and Cooper [19] suggested a heuristics selection method whereby the population of trained networks is ordered in terms of increasing mean squared errors and those with lower mean squared errors are combined.

3.0 METHODOLOGY OF PROPOSED COMBINATION

In this paper, the correlation coefficient analysis is used to select the individual network for combination as well as a heuristic analysis based on sum square error (SSE) on training and testing data. The rationale behind these method is to reduce the failure network by combining networks that are less correlated to each other. When combining neural networks, the individual networks should provide different information in order to enhance the overall performance of the combined network. If two networks are highly correlated, then they would usually represent the same information and combining them would not further improve model generalisation. Based on this consideration, a selective combination technique based on correlation analysis is proposed here.

Correlation coefficient is basically a numerical index of relationship between two variables. A positive (measure of direction) correlation or direct relationship indicates that a high score on one variable is associated with a high score on the second variable. A negative correlation or inverse relationship indicates that a high score on one variable is associated with a low score on the second variable. The magnitude of the correlation coefficient indicates the strength of the relationship between the two variables. This magnitude can vary from 0.00 to 1.00 or 0.00 to -1.00.

In this paper, the first networks selections were carried out by initially sorting the SSE of individual networks on testing and training data. Then, one network that gave the least SSE was selected. Another two networks were added to the group which was network that are least correlated with the first selected network. Then, these three selected networks were combined using simple averaging to get single new prediction output. Then, another correlation coefficient based on the first group combined prediction output with the rest of the network was calculated. Then, another selection was made by excluding any network that is highly correlated compared to the combination output of the first group (contain three networks). Then, based on the number of network that has been selected from the training and

testing data, the final prediction is implemented based on the validation data and SSE is calculated for evaluation and comparison.

The steps in this selective combination approach are shown as follows:

(i) Step 1 : Generate multiple neural networks based on bootstrap re-sampling of the original training and testing data.

(ii) Step 2 : Calculate the residuals for all networks generated in Step 1 on the testing and training data.

(iii) Step 3 : Calculate the SSE of the individual networks on the training and testing data and sort them in ascending order.

(iv) Step 4 : Chose the best predictor which is the network with the least SSE from Step 3.

(v) Step 5 : Calculate correlation coefficient of all networks then based on the correlation coefficient analysis select another 2 networks that less correlated to networks that has been selected in Step 4.

(vi) Step 6 : Combined all the networks that has been selected in Step 5 using simple averaging approach and calculated another correlation coefficient of the single output in Step 5 compared to the rest of 17 individual's network.

(vii) Step 7 : Select those networks from Step 5 that have less correlation coefficients based on the result in Step 6 for combination. Generally, a correlation coefficient of 0.98 to 0.99 is considered as being highly correlated but sometimes it varies depending on the case studies.

This paper also study the performance of combination using fixed and various structure of the networks. In networks with fixed structure, the network structures, i.e. the number of hidden neurons, was 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 sum of squared errors (SSE) on the testing data was considered as having the best network topology. The meaning of various structures in this study is that the numbers of hidden neurons may be different in different individual networks and they are randomly selected within a certain range. The rationale to choose 20 networks for all combination is based on the work of Zhang [17] which shows that constant SSE was observed after combining about 15 networks. Therefore, combining 20 neural networks would be reasonable. If the number of networks is too small, we might not get the optimum reduction of the SSE in the combination. The following networks/combination schemes are investigated:

(i) **M**ean SSE for all networks with fixed structures;

(ii) Mean SSE for all networks with various structures;

(iii) Simple average of all networks with fixed structures;

(iv) Simple average of all networks with various structures;

(v) Selective combination using simple averaging of networks with fixed structures;

(vi) Selective combination using simple averaging of networks with various structures;

(vii) PCR combination of networks with fixed structures;

(viii) PCR combination of networks with various structures;

(ix) PCR combination of selected networks with fixed structures;

(x) PCR combination of selected networks with various structures;

(xi) MLR combination of networks with fixed structures;

(xii) MLR combination of networks with various structures;

(xiii) MLR combination of selected networks with fixed structures;

(xiv) MLR combination of selected networks with various structures.

4.0 MODELLING AND SIMULATION OVERVIEW

Two case studies were used to compare different methods for combining multiple neural networks. In each of the case study, individual networks were trained by the Levenberg-Marquardt optimisation algorithm with regularisation and 'early stopping'. These individual networks, or part of them, were combined to improve model robustness. Regularisation is achieved by modifying the networks training objective to include a term to penalise unnecessarily large network weights as follows:

$$J = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}(i) - y(i))^2 + \rho \| W \|^2 \tag{11}$$

where N is the number of data points, $\hat{y}$ is the networks prediction, y is the target value, W is a vector of networks weights and ρ is the regularisation parameter, which is set to 0.001 in this study. The individual networks are single hidden layer networks where the hidden layer neurons use the sigmoidal activation function and the output layer neurons use a linear activation function.

All weights and biases were initialised randomly in the range (–0.1, 0.1). Bootstrap re-sampling approach was used to generate training and testing data for individual

networks in all case studies. Bootstrap application or bootstrap technique was first introduced in 1979 as a computer based method for estimating the standard error of empirical distribution. In neural networks, bootstrap basically relate or deals with the sampling to create random data sets for training and testing. By creating an equal number of bad and good data sampling, it actually improves the generalisation ability because it helps the identification of the characteristic of the scarce class.

In order to cope with different magnitudes in the input and output data, all the data were scaled to zero mean and unit standard deviation. The data for neural network model building need to be divided into: (i) Training data (for network training); (ii) Testing data (for cross-validation based network structure selection and early stopping); and (iii) Unseen validation data (for evaluation of the final selected model). In assessing the developed models, SSE on the unseen validation data is used as the performance criterion.

5.0 CASE STUDIES

5.1 Case Study 1: Neutralisation Prediction

The neutralisation process takes place in a CSTR as shown in Figure 2 and there are two input streams to the CSTR. One is acetic acid of concentration C1 at flow rates F1 and the other is sodium hydroxide of concentration C2 at flow rates F2. This case study is taken from reference [20].

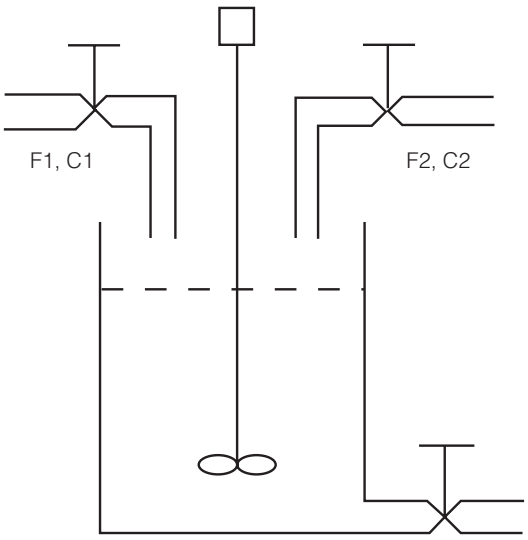

Figure 2 CSTR for neutralisation process

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

$$V\frac{d\zeta}{dt} = F_1C_1 - (F_1 + F_2)\zeta \tag{12}$$

$$V\frac{d\zeta}{dt} = F_2C_2 - (F_1 + F_2)\varsigma \tag{13}$$

$$\left[H^+\right]^3 + (Ka + \varsigma)\left[H^+\right]^2 + \left[Ka(\varsigma - \zeta) - Kw\right]\left[H^+\right] - KwKa = 0 \tag{14}$$

$$\text{pH} = \log_{10}[H^+] \tag{15}$$

where

$$\zeta = [HAC] + \left[AC^-\right] \tag{16}$$

$$\zeta = \left[Na^+\right] \tag{17}$$

All the nominal values of this simulation can be found in reference [20].These equations show that the neutralisation predictions are very non-linear between titration flow and pH prediction in the CSTR. To generate training, testing and validation data, multi level random perturbation were added to the flow rate of acetic acid while other input to the reactor were kept constant.

The pH measurement was corrupted with random noise with the distribution of (0, 0.2). Two single-hidden layer feed forwards neural networks was developed to model the neutralisation process. The dynamic model representing the neutralisation process is:

$$\hat{y}(t) = f[y(t-1), y(t-2), u(t-1), u(t-2)] \tag{18}$$

where $\hat{y}(t)$ is the pH prediction in the reactor at time *t*, *y(t)* is the actual pH value at time *t* and *u(t)* is the acid flow rates at time *t*. Determination of the number of hidden neurons are based on trial and error by varying the number of hidden neurons from 1 to 10 and calculating the SSE based on the testing data. In this analysis, the hidden layer neurons use the sigmoidal function as the activation function, while

the output layers neurons uses the linear activation function. For fixed structures, 4 hidden neurons are used for neural networks models.

5.2 Case Study 2: Water Tank Level Prediction

Figure 3 shows the model of water tank level apparatus [17]. There is an inlet stream to the tank and an outlet stream from the tank. The water tank level is regulated by manipulating the flowrates of the inlet water.

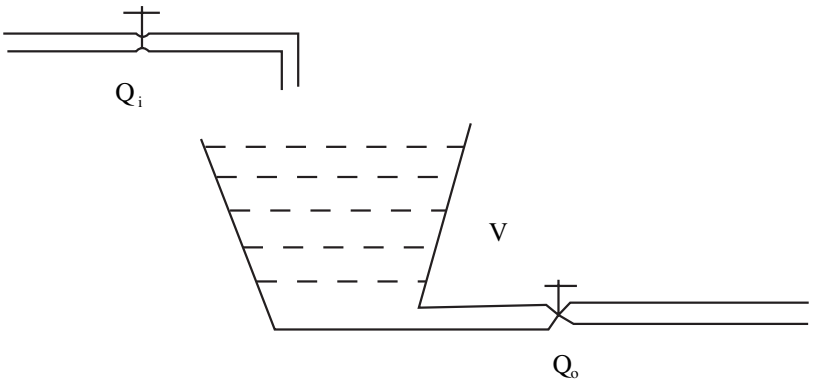

Figure 3 A conic water tank

Let V, Q_1 and Q_o be the volume of water in the tank and the inlet and outlet water flowrates rates respectively, then the material balance will be:

$$\frac{dV}{dt} = Q_1 - Q_o \tag{19}$$

The outlet water flowrates, Q_o, is related to the tank level, h, by the following equation:

$$Q_o = k\sqrt{h} \tag{20}$$

where k is constant for a fixed valve opening.

The volume of water in the tank is related to the tank level by the following equation:

$$V = \pi h\left[r^2 + \frac{h}{\tan\theta} + \frac{h^2}{3(\tan\theta)^2}\right] \tag{21}$$

where r is the tank bottom radius and θ is the angle between the tank boundary and the horizontal plane. Combining Equations (19), (20) and (21) gives the dynamics model for the tank level as shown below:

$$\frac{dh}{dt} = \frac{Q_1 - k\sqrt{h}}{n\left[r^2 + \frac{2rh}{\tan\theta} + \frac{h^2}{(\tan\theta)^2}\right]} \tag{22}$$

Based on the above model, a simulation programme is developed to simulate the process. The parameters used in the simulation are r = 10 cm, k = 34.77 cm $^{2.5}$/s and θ = 60. The sampling time is 10 seconds. The above equation indicates that the relationship between the inlet water flow rate and the water level in the tank is non-linear. The outlet valve characteristic determined that the static gain increased with tank level. Because the tank is of a conical shape, the time constant of the process increases with the tank level. Thus, both the static and dynamic characteristics of the process vary with the operating condition. All the data are generated from the simulation programme and noise with the distribution N (0, 0.7 cm) are added to the simulated tank. The dynamic model for tank level prediction is of the form:

$$\hat{y}(t) = f\left[y(t-1), u(t-1)\right] \tag{23}$$

where y represents the tank level and u represents the inlet flow rate. Hidden neuron determination is based on cross validation techniques based on the single neuron network and in this case 4 hidden neuron is the 'best'. Furthermore, in the various structure analyses, the hidden neurons were varied from 1 to 10.

6.0 RESULTS AND DISCUSSIONS

6.1 Case Study 1: Neutralisation Prediction

This is a second order system where there are 4 inputs to predict the single output. The same regularisation parameter was used and so was the training method. Figure 4 shows the variation of validation SSE in fixed and various structures of single neural network models. It clearly shows that the individual neural network performed differently even though the same method of training is applied. This is one of the non-robust nature of single neural network.

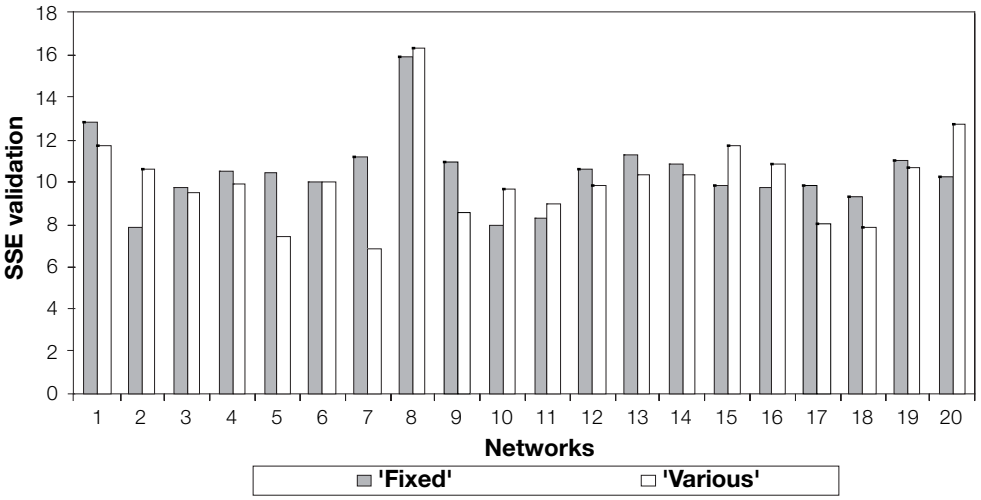

Figure 4 Validation SSE in single networks with fixed and varing structures for Case Study 1

Figure 4 also indicates that for networks with fixed structures, the best performance is achieved by a single network which is network number 2 with an SSE of 7.8595 while network number 8 gives the worst performance with SSE of 15.9080. In varying the structure, the best single neural network is achieved in network 7 with SSE of 6.8584 and the worst is in network 8 with SSE of 16.3254. The mean SSE for fixed and various structures are 10.4428 and 10.1160 respectively.

In selective combination method based on combination of correlation coefficient, 10 (7+3) and 11 (8+3) networks were used for combination for fixed and various structures respectively. Table 1 shows the validation SSE for combination schemes as shown in Section 2. A comparison to normal selection based on correlation coefficient was also made. It clearly shows that selective combination method performed well compared to other combination methods especially to mean validation SSE for both network structures.

The selective combination based on combined correlation coefficient showed significant improvement in reducing validation SSE. The best selective combination in this analysis is in schemes "N: MLR combination of selected networks with various structures" with validation SSE of 6.1286.

Table 1 Overall results for combination of multiple neural network for Case Study 1

Combination schemes	Validation SSE (Correlation coefficient)	Validation SSE (Combined correlation coefficient)
A		10.4428
B		10.1160
C		8.7766
D		7.8373
E	8.1714	8.8003
F	7.9212	7.7750
G		7.1564
H		6.2642
I	6.9420	6.8490
J	7.6253	6.2459
K		7.2520
L		6.3905
M	7.0640	6.8490
N	7.1835	6.1286

6.2 Case Study 2: Water Tank Level Prediction

In this case study, 20 networks with fixed number of hidden neurons (4) and 20 networks with varying number of hidden neurons (between 1 and 10) were developed. Each of the individual networks was trained on bootstrap re-samples of the original training data set.

Figure 5 shows the SSE on the unseen validation data of the individual networks for fixed and varying structure of network respectively. It can be seen that their performance varies significantly. This demonstrates the different generalisation

capabilities of the individual networks. It is shown in Figure 5 that the lowest SSE on the unseen validation data for the single neural networks with fixed and various numbers of hidden neurons are 3.3202 and 3.1628 respectively, and the majority of these networks have SSE over 3.5. Once these individual networks were developed, they were combined using the different combination schemes presented in Section 2.

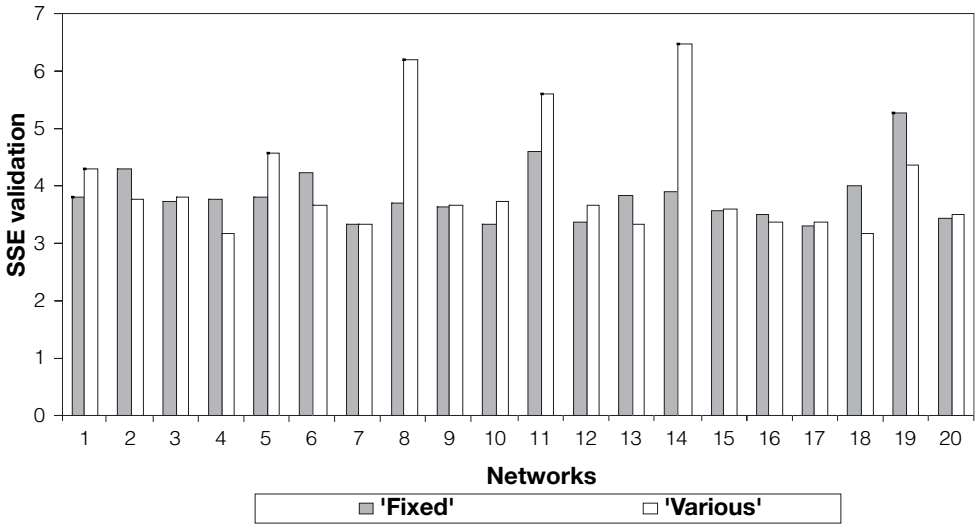

Figure 5 Validation SSE in single networks with fixed and varing structures for Case Study 2

Selective combination method using combined correlation coefficient is the main objective of this paper. Therefore, based on correlation coefficient techniques described in Section 2, simple averaging, PCR and MLR aggregation methods were implemented. Table 2 shows that most of the combination schemes significantly improve model performance on the unseen data.

For selective combination methods 10 (7+3) and 11 (8+3) networks are chosen for combination for fixed and varying structure models respectively. Again PCR and MLR aggregation methods in selective combination show better performance based on the least SSE in the validation data compared to combination of all networks. Significant improvement was also shown in MLR aggregation methods especially in combined correlation coefficient approach as shown in Table 2. MLR aggregation method is prone to the collinearity, therefore in this case, the combination are based on the predictor that is less correlated, thus enhance the MLR performance.

Table 2 Overall results for combination of multiple neural network for Case Study 2

Combination Schemes	Validation SSE (Correlation Coefficient)	Validation SSE (Combined Correlation Coefficient)
A	3.8122	
B	4.0324	
C	3.4378	
D	3.3890	
E	3.5586	3.4514
F	3.3329	4.9902
G	3.4727	
H	3.4299	
I	3.5592	3.4503
J	3.5481	3.3830
K	5.9368	
L	7.2672	
M	4.1633	3.8225
N	3.8595	3.4649

7.0 CONCLUSIONS

The objective of this paper is to improve the performance of the combined network by excluding the networks that are considered as 'failure' or highly correlated with the selected networks using combined correlation coefficient analysis. The result is convincing for both case studies where it is shown in MLR combination where the performance of the combined network model was significantly improved compared to combining all networks in reducing the validation SSE. The result confirms that combining some particular networks can improve the robustness of the combined network, however, a proper selection of networks need to be investigated.

The inconsistent result in some of the case studies might be related to the selection of the networks. The only problem in the correlation coefficient analysis is that there is no systematic methods to select the networks that should be combined. All selection is based on human judgement, based on correlation coefficients. For

instance, when we chosed the first three networks that we need to combine based on SSE and correlation coefficient, we just simply followed the heuristics analysis and based on human judgement for correlation coefficient like 0.99 or 0.95 is excluded.

Finally for the future study, we need to find a systematic technique or tool that can select the network properly or automatically for combination.

ACKNOWLEDGEMENTS

This work was supported by Universiti Sains Malaysia (USM) for Zainal Ahmad and Rabiatul Adawiah Mat Noor through short term grant 6035182.

NOTATIONS

C_1 acetic acid inlet concentration, mol/L

C_2 sodium hydroxide inlet concentration, mol/L

F_1 acetic acid input flow rate, L/min

F_2 sodium hydroxide input flow rate, L/min

K_a acid equilibrium constant

K_w water equilibrium constant

N numbers of samples

P loading matrix

Q_1 inlet water flow rate, cm/second

Q_o outlet water flow rate, cm/second

T principal component

V volume of water in the tank, cm^3
volume of the tank in CSTR, L

X input data vector

Y output data vector

$Y_{(tr)}$ output prediction based on the training data in PCR

a parameter constant for multiple linear regressions of the networks

h tank water level, cm

k constant for a fixed valve opening, $cm^{2.5}$/second

n total number of networks

pH pH in the reactor at time t

r tank bottom radius, cm

t sampling time Case Study 1, second
sampling time Case Study 2, min

REFERENCES

[1] Wolpert, D. H. 1992. Stacked Generalisation. *Neural Networks.* 5: 241-259.

[2] Bishop, C. 1995. *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press.

[3] Caruana, R., S. Lawrence, and C. Lee Giles. 2001. Overfitting in Neural Networks: Backpropagation, Conjugate Gradient and Early Stopping. *Advances in Neural Information Processing System.* 13: 402-408.

[4] Hagiwara, K. and K. Kuno. 2000. Regularisation Learning and Early Stopping in Linear Networks. International Joint Conference on Neural Networks (IJCNN 2000). 511 – 516.

[5] Mc Loone, S. and G. Irwin. 2001. Improving Neural Networks Training Solution Using Regularisation. *Neurocomputing.* 37: 71-90.

[6] Crucianu, M., R. Bone, and J. P. Asselin de Beauville. 2001. Bayesian Learning for Re-current Networks. *Neurocomputing.* 36: 235-242.

[7] Radford, M. N. 1997. Bayesian Learning for Neural Networks. *Statistics.* 118: 183.

[8] Jacobs, R. A., M. I. Jordan, S. J. Nowlan, and G. E. Hinton. 1991. Adaptive Mixture of Local Expert. *Neural Computation*. 3: 79-87.

[9] Hashem, S. 1997. Optimal Linear Combination. *Neural Networks.* 10(4): 599-614.

[10] Hashem, S. 1999. *Treating Harmful Collinearity in Neural Networks Ensembles*. *Combining Artificial Neural Nets Ensemble and Modular.* London: Springer Publication.

[11] Jordan, M. I. and R. A. Jacobs. 1994. Hierarchical Mixtures of Expert and the EM Algorithm. *Neural Computation.* 6: 181-214.

[12] Larsson, L., K. Krol, and K. Lagemann. 1996. NeNEB - An Application Adjustable Single Chips Neural Networks Processor for Mobile Real Time Image Processing. International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing. 154-163.

[13] Sharkey, A. J. C. 1999. *Multi Nets System*. *Combining Artificial Neural Nets Ensemble and Modular*. London: Springer Publication.

[14] Sridhar, D. V., E. B. Bartlett, and R. C. Seagrave. 1999. An Information Theoretic Approach for Combining Neural Network Process Models. *Neural Networks.* 12: 915-926.

[15] Sridhar, D. V., E. B. Bartlett, and R. C. Seagrave. 1996. Process Modelling Using Stacked Neural Networks. *AIChE Journal.* 42(9): 2529-2539.

[16] Zhang, J. 1999. Inferential Estimation of Polymer Quality Using Bootstrap Aggregated Neural Networks. *Neural Networks.* 12: 927-938.

[17] Zhang, J. 1999. Developing Robust Non-linear Models Through Bootstrap Aggregated Neural Networks. *Neurocomputing.* 25: 93-113.

[18] Jordan, M. I. and R. A. Jacobs. 1994. Hierarchical Mixtures of Expert and the EM Algorithm: *Neural Computation.* 6: 181-214.

[19] Perrone, M. P. and L. N. Cooper. 1993. When Networks Disagree: Ensembles Methods for Hybrid Neural Networks. *Artificial Neural Networks for Speech and Vision*. London: Chapman and Hall. 126-142.

[20] Mc Avoy, T. J., E. Hsu, and S. Lowenthal. 1972. Dynamics of pH in Controlled Stirred Tank Reactor. *Ind. Chem. Process Des. Development*. 11: 68-70.