A Comparison of Different Methods for Combining Multiple Neural Networks Models

Zainal Ahmad and Jie Zhang
Centre for Process Analytics and Control Technology, Department of Chemical & Process Engineering,
University of Newcastle, Newcastle Upon Tyne NE1 7RU, U. K.
E-mail: Z.B.Ahmad@newcastle.ac.uk, jie.zhang@newcastle.ac.uk

Abstract – A single neural network model developed from a limited amount of data usually lacks robustness. Neural network model robustness can be enhanced by combining multiple neural networks. There are several approaches for combining neural networks. A comparison of these methods on three non-linear dynamic system modelling case studies is carried out in this paper. It is shown that selective combination and combining networks of various structures generally improve model performance. The principal component regression approaches generally give quite consistent good performance.

I. INTRODUCTION

Artificial neural networks have been increasingly used in developing non-linear 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 [17]. Model robustness is primarily related to the learning or training methods and the amount and representativeness of the training data [1]. Even though neural networks have a significant capability in representing non-linear functions, inconsistency of accuracy still seems to be a problem where a neural network model cannot cope or perform well when it is applied to new unseen data. Furthermore, advanced process control and supervision of industrial processes require accurate process models promoting investigations in the robustness of neural networks models. Lack of robustness in neural network models is basically due to the overfitting and poor generalisation of the models (e.g. [2]). Therefore, a lot of researchers have been interested and concentrated on how overfitting can be alleviated by improving the learning algorithms or by combining multiple neural networks (e.g. [4,14,15,16,21]). In view of improving the robustness of neural network models a lot of techniques have been developed like regularisation and the early stopping method (e.g. [3,8,11]). Reference [12] implemented the universal learning rule with second order derivatives to increase the robustness in neural network models.

Among those approaches, the combination of multiple neural networks shows some encouraging results in term of improving the robustness of neural networks. Fig. 1 shows how multiple neural networks are combined (e.g. [6, 7, 14, 15, 16, 20, 21, 22]). 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 [15,20,22]. Reference [22] 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.

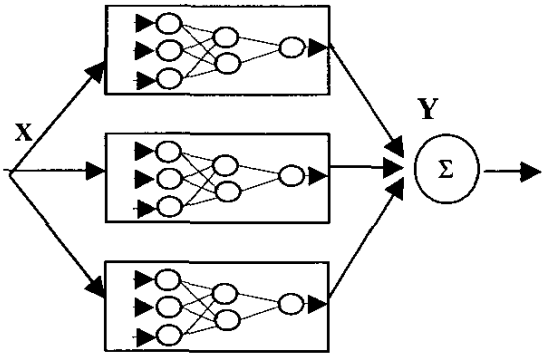

Fig. 1. Multiple Neural Networks

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 [17]. Reference [6] came up with the procedure of combining very different neural networks. The hierarchical mixture of neural networks is also considered as one of the methods for combining neural networks [7]. Most of the combinations of networks so far are based on linear combination (e.g. [4,5,15,16]). 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 failures 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 limited training data set (e.g. [4,10]). In another word, combining a set of imperfect estimators (nets) can be thought of as a way of managing the recognised limitation of the individual estimators, each component is known to make errors, but they are combined in such a way as to minimise the effect of these errors [14].

This paper presents a comparison of different methods for combining multiple networks using three non-linear dynamic modelling case studies. The paper is structured as follows. Section II presents different methods for combining multiple neural networks. Three case studies are given in Section III. Section IV presents the results and discussions on the three

0-7803-7278-6/02/$10.00 ©2002 IEEE

case studies. Finally, some concluding remarks are drawn in Section V.

II. DIFFERENT METHODS FOR COMBINING NEURAL NETWORKS

Methods for combining multiple networks can be divided into the following approaches:

1. Simple Averaging and Weighted Averaging.

This method is the most common approach in combining neural networks. The linear combination of multiple network outputs creates a single output as the final model prediction. In weighted averaging, weights for individual networks need to be calculated, for example, through multiple linear regression (MLR) or principal component regression (PCR). Reference [20] used the PCR approach to select the combination weights.

2. Non-linear Combining Methods

Non-linear combining methods include Demspter-Shafers belief based method, combining using rank based information, voting, order statistic and Tumer and Ghosh methods.

3. Supra Bayesian.

This approach basically contrasts to linear combination. The philosophy behind this approach is that the opinions of the experts are themselves a data set. Therefore the probability distribution of the experts can be combined with its own prior distribution.

4. Stacked Generalisation.

Reference [17] generalises this idea of combination by combining the networks with weights that vary over the feature space. The outputs from a set of level 0 generalisers used as the inputs to level 1 generaliser, which is trained to produce the appropriate output.

Other methods of combination are selective combination of networks. The objective behind selective combination is to reduce the number of shared failure when combining the networks [14]. There are a lot of methods on how to select proper networks for combination. Reference [13] suggests a heuristics selection method whereby the population of trained networks is ordered in terms of increasing mean squared errors, and in combination by including those with lower mean squared errors.

This paper compares several linear combination techniques with fixed individual network structures, variable individual network structures, and selective combination. The following combination schemes are investigated:

a). Simple average of networks with fixed structures;
b). Selective average of networks with fixed structures;
c). Simple average of networks with various structures;
d). Selective average of networks with various structures;
e). PCR combination of networks with fixed structures;
f). PCR combination of selected networks with fixed structures;
g). PCR combination of networks with various structures;
h). PCR combination of selected networks with various structures;
i). MLR combination of networks with fixed structures;
j). MLR combination of selected networks with fixed structures;
k). MLR combination of networks with various structures;
l). MLR combination of selected networks with various structures.

In selective combinations, networks with much larger training and testing errors are excluded from forming stacked networks.

III. CASE STUDIES

Three case studies were used to compare different methods for combining multiple neural networks. In each of the case studies, 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:

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

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 use to generate training and testing data for individual networks in all case studies. The sum of squared errors (SSE) on the unseen validation data is the performance criterion for all the comparisons. To cope with different magnitudes in the input and output data, all the data were scaled to zero mean and unit standard deviation.

1) Case Study 1: Water Tank Level Prediction: Fig. 2 shows the diagram of a conic water tank. There is an inlet stream to the tank and an outlet stream from the tank. The water tank level is regulated by manipulating the inlet water flow rates.

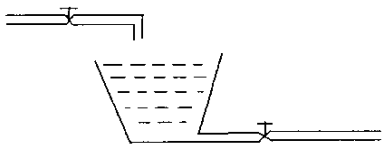

Fig. 2. A Conic Water Tank

This case study is taken from [18] and all the equations and mechanistic details of the process can be found in [18]. Due to the shape of the tank, the dynamic relationship between the inlet flow and tank level is quite non-linear. All the model building data were generated from the simulation programme and noises with the distribution N(0cm, 0.7cm) were added to the simulated tank level. The data were divided into three sections, which are training data, testing data and unseen validation data. Neural networks were trained on the training data set and tested on the testing data set. The testing data is for the selection of model structures and network structures, such as the number of hidden neurons. Performance of the final selected model is evaluated on the unseen validation data. The dynamic model structure selected for tank level prediction is of the form:

$$y(t) = f[y(t-1), u(t-1)] \qquad (2)$$

where y represents the tank level and u represents the inlet flow rate. Hidden neuron determination is based on the SSE on the testing data. For combining fixed structure networks, the number of hidden neurons was selected as 3. For combining variable structure networks, the number of hidden neurons of the individual networks randomly varied between 2 and 8.

2) Case Study 2: pH Prediction 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. This case study is taken from [9], where detailed mathematical equations about the neutralisation process can be found.

It is well known that the dynamic relationship between titration flow and pH in the CSTR is very non-linear. 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. Three sets of data were generated, one set was used as the training data, another set was used as the testing data, and the remaining set was used as the unseen validation data. The pH measurements were corrupted with random noise with the distribution N(0, 0.2). Single-hidden layer feed forward neural networks were developed to model the non-linear dynamic relationship between acid flow rate and the pH in the reactor. The dynamic model structure selected is:

$$y(t) = f[y(t-1), y(t-2), u(t-1), u(t-2)] \qquad (3)$$

where $y(t)$ is the pH in the reactor at time t and $u(t)$ is the acid flow rate at time t. For combining fixed structure networks, the number of hidden neurons was selected as 6. For combining variable structure networks, the number of hidden neurons of the individual networks randomly varied between 4 and 10.

3) Case Study 3: Sunspot Prediction: This is a non-linear time series where yearly sunspot blotches data are recorded from 1710 to 1974 consisting of 265 data points. This data set was divided into 3 sections, which are training, testing and unseen validation data. The time series models developed for this case study is of the form:

$$\hat{y}(t) = f[y(t-1), y(t-2), y(t-3)] \qquad (4)$$

For combining networks of fixed structure, the number of hidden neurons was selected as 5. For combining networks of variable structure, the number of hidden neurons of the individual networks randomly varied between 2 and 9.

IV. RESULTS AND DISCUSSIONS

A. Case Study 1: Water Tank Level Prediction

Due to the data are corrupted with noise, the neural network prediction might face some problems where the prediction output is driven by the noise. In this case study, 20 networks with fixed number of hidden neurons (3) and 20 networks with varying number of hidden neurons (between 2 and 8) were developed. Each of the individual networks was trained on bootstrap re-samples of the original training data set. Fig. 3 shows the SSE on the unseen validation data of the individual networks. It can be seen that their performance varies quite significantly. This demonstrates the different generalisation capabilities of the individual networks.

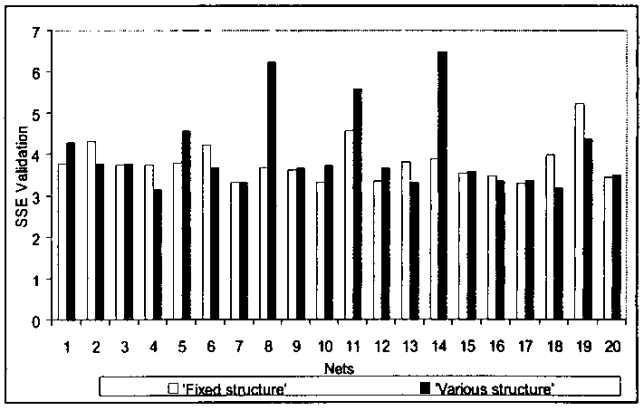

Fig. 3. Validation SSE in single networks with fixed and varying structures for Case Study 1

Once these individual networks were developed, they were combined using the different combination schemes presented in Section II. In selective combination of networks with fixed

structure, 16 networks with smaller SSE on the training and testing data were combined. In selective combination of networks with various structures, 14 networks with smaller SSE on the training and testing data were combined.

It is shown in Fig. 3 that the lowest SSE on the unseen validation data for the single neural networks with fixed number of hidden neurons is 3.3032 (the 17^{th} network) and the majority of these networks have SSE over 3.5. The mean SSE for fixed and various structures are 3.8122 and 4.0325 respectively. Table I shows that most of the combination schemes significantly improved model performance on the unseen data. The combination approaches involving simple average and PCR give good performance. The MLR based combination schemes do not give good performance in this case study. Selective combination and combining networks with various structures help to improve model performance in this case study.

TABLE I
OVERALL RESULT FOR COMBINATION OF MULTIPLE NEURAL NETWORKS FOR TANK LEVEL PREDICTION

Combination schemes	SSE (Validation)
a	3.4378
b	3.419
c	3.389
d	3.4101
e	3.4727
f	3.4234
g	3.4299
h	3.4251
i	5.9368
j	6.2736
k	7.2672
l	4.4218

B. Case Study 2: Neutralisation Prediction

Similar approaches were also applied in this case study. This system is quite complex and highly non-linear. In selective combination of networks with fixed structure, 14 networks with smaller SSE on the training and testing data were combined. In selective combination of networks with various structures, 10 networks with smaller SSE on the training and testing data were combined.

It can be seen from Fig. 4 that single networks give various performances based on the differences in SSE on the unseen validation data. Fig. 4 also indicates that, for networks with fixed structure, the best performance achieved by a single network is an SSE of 7.8595 from network 2 and the worst performance is an SSE of 15.9080 from network 8. This demonstrates the variation in individual network performance and the non-robust nature of single networks. The mean SSE for fixed and various structures are 10.4428 and 10.1160 respectively. Table II shows that most combination schemes can give better performance on the unseen data than the best single network. Overall the PCR based combination methods better performance than other approaches. The MLR combination approaches also achieved good performance. This could be due to that the individual networks are not significantly correlated in this case study. Combining networks with various structures improves model performance for most of the combination approaches in this case study. Selective combination works for simple average but not for PCR and MLR. Perhaps a different selection scheme should be looked at.

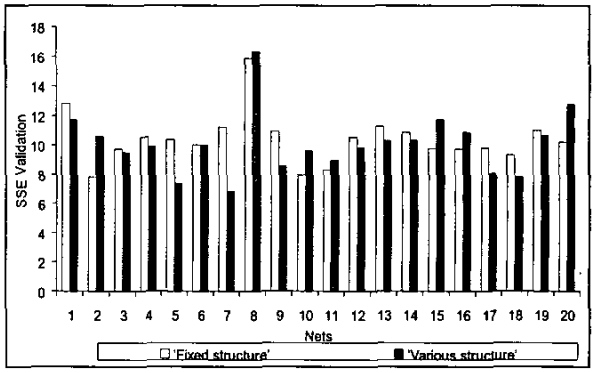

Fig. 4. Validation SSE in single networks with fixed and varying structures for Case Study 2

TABLE II
OVERALL RESULT FOR COMBINATION OF MULTIPLE NEURAL NETWORKS FOR NEUTRALISATION PREDICTION

Combination schemes	SSE (Validation)
a	8.7766
b	8.6151
c	7.8373
d	7.6242
e	7.1564
f	7.6899
g	6.2642
h	7.3518
i	7.252
j	7.9346
k	6.3905
l	7.6312

C. Case Study 3: Sunspot Prediction

This case study is a non-linear time series modelling problem, which is quite difficult to model. In selective combination of networks with fixed structure, 10 networks with smaller SSE on the training and testing data were combined. In selective combination of networks with various structures, 13 networks with smaller SSE on the training and testing data were combined.

Fig. 5 shows the variations of SSE on the unseen validation data of the 20 networks with fixed structure and the 20 networks with different structures. It indicates that the performance varies quite significantly among the individual networks with SSE ranging from about 17 to about 35. Most

of the networks have SSE on validation data higher than 20. This shows that single neural networks have poor generalisation capability in time series modelling. The individual networks that giving the best performance on the unseen data is network number 16 and number 7 for fixed and different structure respectively with SSE of 16.5286 and 16.4003 respectively. The mean SSE for fixed and various structures are 21.5462 and 23.4165 respectively.

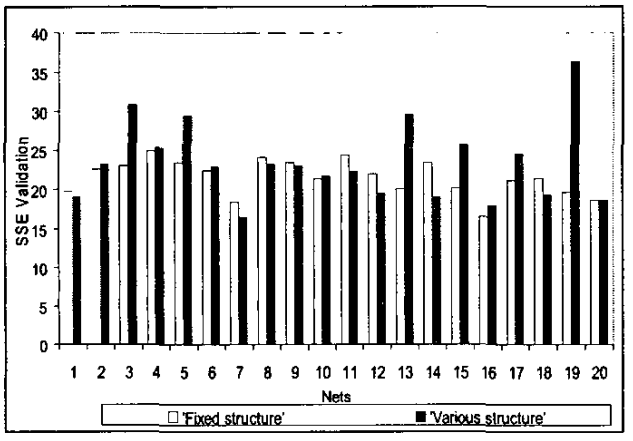

Fig. 5. Validation SSE in single networks with fixed and varying structures for Case Study 3

TABLE III
OVERALL RESULT FOR COMBINATION OF MULTIPLE NEURAL NETWORKS SUNSPOT PREDICTION

Combination schemes	SSE (Validation)
a	19.6802
b	19.0179
c	19.021
d	18.2332
e	19.5442
f	18.9251
g	21.9413
h	18.7761
i	19.5442
j	18.8678
k	21.9413
l	19.5064

Table III shows that all combination methods can improve model robustness and performance on unseen data. Once again, selective combination improves model performance. Combining networks with various structure improves the model performance in simple average and some of the PCR approach, but not in the MLR approaches. From Table III it also can be seen that the time series modelling system is quite difficult to predict because it totally depend on the quality of the previous data and also the volume of the data itself. Too little data can cause a lack of robustness of the model. However, based on this study it shows that time series modelling accuracy still can be improved by implementing the multiple combination of the neural networks.

V. CONCLUSIONS

In multiple NN, the generalisation capabilities of individual networks are not the same and, therefore, different networks generate different errors. Combining these networks can improve the robustness of the NN by sharing and averaging out these errors. In these case studies, combination using selective combination methods and combining networks with various structures seem quite improve the performance of the model. The PCR approaches also improve model performance.

Among those linear combinations that have been applied in this study, the PCR aggregation method has shown to be the most consistent approach. PCR performed quite well even though in limited amount of training data especially in time series modelling system that we know it will suffer when there are less training data. Therefore based on the result and discussion that has been made previously, PCR seems to be the promising aggregation method for combining NN, where it performed quite well in complex and non-linear dynamics systems, furthermore PCR is a quite simple and practical approach in many cases of applications.

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

VI. REFERENCES

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

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

[3]. Hagiwara, K. and K. Kuno, "Regularisation Learning and Early Stopping in Linear Networks", *International Joint Conference on Neural Networks (IJN 2000)*, pp 511 – 516, 2000.

[4]. Hashem, S, "Optimal Linear Combination", *Neural Networks*, Vol. 10: 4, pp. 599-614, 1997.

[5]. Hashem, S, "Treating Harmful Collinearity in Neural networks Ensembles", *Combining Artificial Neural Nets Ensemble and Modular*, A. J. C Sharkey (Ed), Springer Publication, London, 1999.

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

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

[8]. Heartz, J. A., A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA, 1991.

[9]. McAvoy, T. J., E. Hsu, S Lowenthal, "Dynamics of pH in Controlled Stirred Tank Reactor", *Ind. Chem. Process Des. Development*, Vol.11, pp. 68-70, 1972.

[10]. McLoone, S. and G. Irwin, "Improving Neural Networks Training Solution Using Regularisation", *Neurocomnputing*, Vol.37, pp. 71-90, 2001.

[11]. Morgan, N. and H. Bourlard, "Generalisation and Parameter Estimation in Feedforward Nets: Some Experiments", In Touretzkey, D. S. (Ed.), *Advances in Neural Information Processing System*, Vol.2, San Mateo CA, pp. 630-637, 1990.

[12]. Ohbayashi, M., K. Hirasawa, K. Toshimitsu, J. Murata, and J. Hu, "Robust Control for Non-linear System by Universal Learning Networks Considering Fuzzy Criterion and Second Order Derivatives", *IEEE World Congress on Computational Intelligence: IEEE International Conference Proceeding on Neural Networks*, Vol.2, pp. 968-973, 1998.

[13]. Perrone, M. P. and L. N. Cooper, "When Networks Disagree: Ensembles Methods for Hybrid Neural Networks", In R. J. Mammone (Ed), *Artificial Neural Networks for Speech and Vision*, pp.126-142. London Chapman and Hall, 1993.

[14]. Sharkey, A. J. C , "Multi Nets System", *Combining Artificial Neural Nets Ensemble and Modular*, A. J. C. Sharkey (Ed), Springer Publication, London, 1999.

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

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

[17]. Wolpert, D. H., "Stacked Generalisation", *Neural Networks*, Vol. 5, pp. 241-259, 1992.

[18]. Zhang, J, "Developing Robust Neural Network Models by Using Both Dynamic and Static Process Operating Data", *Ind. Eng. Chem. Res*, Vol. 40, pp. 234-241, 2001.

[19]. Zhang, J, "Inferential Estimation of Polymer Quality using Bootstrap Aggregated Neural Networks", *Neural Networks*, Vol.12, pp. 927-938, 1999.

[20]. Zhang, J, "Developing Robust Non-linear Models Through Bootstrap Aggregated Neural Networks", *Neurocomputing*, Vol. 25, pp. 93-113, 1999.

[21]. Zhang, J., E. B. Martin, A. J. Morris, C. Kiparissides, "Inferential Estimation of Polymer Quality Using Stacked Neural Networks", *Computers & Chemical Engineering*, Vol.21, pp. s1025-s1030, 1997.

[22]. Zhang, J., A. J. Morris, E. B. Martin and C. Kipaerissides, "Prediction of Polymer Quality in Batch Polymerisation Reactors Using Robust Neural Networks", *Chemical Engineering Journal*, Vol.69, pp. 135 – 143, 1998.