A Bayesian Approach for Constructing Ensemble Neural Network

Sai Hung Cheung, Yun Zhang and Zhiye Zhao
School of Civil and Environmental Engineering, Nanyang Technological University, Nanyang Avenue, Singapore, Singapore

Keywords: Neural Network, Bayesian Approach.

Abstract: Ensemble neural networks (ENNs) are commonly used in many engineering applications due to its better generalization properties compared with a single neural network (NN). As the NN architecture has a significant influence on the generalization ability of an NN, it is crucial to develop a proper algorithm to design the NN architecture. In this paper, an ENN which combines the component networks by using the Bayesian approach and stochastic modelling is proposed. The cross validation data set is used not only to stop the network training, but also to determine the weights of the component networks. The proposed ENN searches the best structure of each component network first and employs the Bayesian approach as an automating design tool to determine the best combining weights of the ENN. Peak function is used to assess the accuracy of the proposed ensemble approach. The results show that the proposed ENN outperforms ENN obtained by simple averaging and the single NN.

1 INTRODUCTION

The artificial neural network (NN) is a mathematical or computational model for information processing based on the biological neural networks (McCulloch and Pitts, 1943). The ensemble neural network (ENN) can be significantly improved through ensembling a number of NNs (Hansen and Salamon, 1990). Since this approach behaves remarkably well, nowadays it has been widely applied in many engineering areas.

In Bayesian data analysis, all uncertain quantities are quantified by probability distributions, and inference is performed by constructing the posterior conditional probabilities for the unobserved variables of interest, given the observed data sample and prior assumptions (Lampinen and Vehtari, 2001). The application of Bayesian theory to NNs was started by Buntine and Weigend (1991). Marwala (2007) proposed a Bayesian neural network trained using Markov Chain Monte Carlo (MCMC) and genetic programming (GP) in binary space. Wang et al. (2010) proposed a sequential Bayesian learning for ENNs. This paper proposes a method based on a Bayesian approach and stochastic modelling. One simulated example is used to illustrate the performance of the proposed method.

2 PROPOSED BAYESIAN APPROACH FOR DESIGNING ENN

An ENN is a collection of a finite number of NNs that are trained for the same task. Usually the networks in the ensemble are trained independently and then their predictions are combined (Sollich and Krogh, 1996). The architecture of the ENN is shown in Figure 1. The two main steps to construct an ENN are: Step 1 - creating component networks; Step 2 - combining these component networks in ENN.

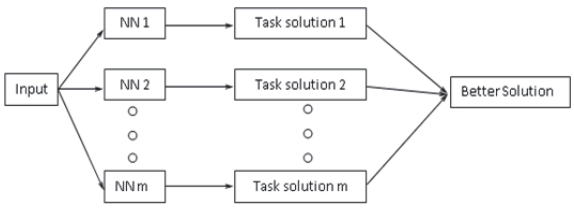

Figure 1: The architecture of the ENN.

In Step 1, Creation of the component network can also be divided into two steps. The first step is to create the training data, the test data and the cross validation data sets, and the second step is to create the component networks. For creating the training data, the test data and the cross validation data sets,

Hung Cheung S., Zhang Y. and Zhao Z..
A Bayesian Approach for Constructing Ensemble Neural Network.
DOI: 10.5220/0004147103740377
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2012), pages 374-377
ISBN: 978-989-8565-29-7
Copyright © 2012 SCITEPRESS (Science and Technology Publications, Lda.)

some common ratio of them will be used in the analyses. All training data are used for each component network. To avoid the overfitting of the component network, the cross validation data set is used to stop the training network. The test data set is used to verify the performance of the network and will not be used in network training. For creating component networks, each component network is created several times, but the best structure will be used in the ENN. The procedure to define the number of hidden nodes in each component network is similar to the method presented in Zhao et al. (2008). The best number of hidden nodes for a single NN is chosen to achieve the smallest training mean squared error (MSE) for sufficient training and smallest test MSE to avoid network overfitting.

After a set of component networks has been created, the method to combine these networks has to be considered. The most widely used method is to use the equal combination weights to combine the members of an ensemble (Hashen, 1993). This set of outputs combined by a uniform weighting is referred to as the simple ensemble (or simple averaging method).

The stochastic system based framework for Bayesian model updating presented in Beck and Katafygiotis (1998) and part of the methodology presented in Cheung and Beck (2010, 2012) are used as a basis for the proposed method presented here. For the proposed ENN, the weight of each best component network will be calculated using the cross validation data set or by the training data set for comparison purpose. Without loss of generality, for illustration, only the case where the output variable is a scalar is considered here. The output $y(\underline{x})$ of the ENN is modelled as a stochastic process in continuous input variables $\underline{x}$ given as follows:

$$y(\underline{x}) = \underline{f}(\underline{x})^T \underline{w} + \varepsilon(\underline{x}) \tag{1}$$

where $\underline{f}(\underline{x})$ is a vector with components given by the output of the component networks corresponding to the input variables $\underline{x}$; the error term $\varepsilon(\underline{x})$ is modelled as a stochastic process in $\underline{x}$ which is chosen to be Gaussian here with mean zero and covariance function $\mathrm{cov}(\varepsilon(\underline{x}^{(i)}), \varepsilon(\underline{x}^{(j)}); \underline{\sigma}, \underline{l})$ which is a function of $\underline{x}^{(i)}$ and $\underline{x}^{(j)}$ with parameters $\underline{\sigma}$ and $\underline{l}$. The weight of the component network $\underline{w}$ together with $\underline{\sigma}$ and $\underline{l}$ are treated as uncertain parameters. Given the measured input $X=[\underline{x}^{(1)} \ldots \underline{x}^{(N)}]^T$ and output data $\underline{y}=[y_1 \ldots y_N]^T$, the probabilistic information about these parameters is encapsulated in the posterior probability density distribution (PDF) given as follows by Bayes' Theorem:

$$p(\underline{w},\underline{\sigma},\underline{l} \mid X, \underline{y}) = \frac{\exp[-\frac{1}{2}(\underline{y}-F\underline{w})^T \sum^{-1}(\underline{\sigma},\underline{l})(\underline{y}-F\underline{w})]}{(2\pi)^{N/2} \mid \sum(\underline{\sigma},\underline{l}) \mid^{1/2}} p(\underline{w},\underline{\sigma},\underline{l}) \tag{2}$$

where $p(\underline{w},\underline{\sigma},\underline{l})$ is the prior PDF taken as uniform here; $F=[\underline{f}(\underline{x}^{(1)}) \ldots \underline{f}(\underline{x}^{(N)})]^T$ is a matrix with entries given by the output or the predictor of the component networks corresponding to the measured inputs given in X; the (i,j) element of the covariance matrix $\sum(\underline{\sigma},\underline{l}) = \mathrm{cov}(\varepsilon(\underline{x}^{(i)}), \varepsilon(\underline{x}^{(j)}); \underline{\sigma}, \underline{l})$. In the globally identifiable case (Beck and Katafygiotis, 1998) where there is only one optimal solution $\boldsymbol{\theta}^*$ (called the most probable solution) maximizing the posterior PDF of the uncertain parameters $\boldsymbol{\theta}$, it can be shown that given a sufficient amount of data, the posterior PDF can be well approximated by a Gaussian distribution with mean equal to $\boldsymbol{\theta}^*$ and covariance matrix given by the inverse of the Hessian matrix of the negative natural logarithm of the posterior PDF evaluated at $\boldsymbol{\theta} = \boldsymbol{\theta}^*$.

For the important special case where $\mathrm{cov}(\varepsilon(\underline{x}^{(i)}), \varepsilon(\underline{x}^{(j)}); \underline{\sigma}, \underline{l}) = \sigma^2 g(\underline{x}^{(i)}, \underline{x}^{(j)}; \underline{l})$, $\sum(\underline{\sigma},\underline{l}) = \sigma^2 R(\underline{l})$ where the (i,j) element of the matrix $R(\underline{l}) = g(\underline{x}^{(i)}, \underline{x}^{(j)}; \underline{l})$ and $g(\underline{x}^{(i)}, \underline{x}^{(j)}; \underline{l})$ takes the form such that $R(\underline{l})$ approaches an identity matrix if $\underline{l}$ approaches a zero vector. For this case, $\boldsymbol{\theta}^* = [\underline{w}^{*T}\ \sigma^{2*}\ \underline{l}^{*T}]^T$ can be determined by using the proposed iterative algorithm as shown in Figure 2. The objective function $\tilde{J}$ of the sub-optimization problem as shown in the figure is given by the negative natural logarithm of $p(\underline{w},\underline{\sigma},\underline{l} \mid X, \underline{y})$ as follows:

$$\tilde{J}(\underline{w},\sigma^2,\underline{l}) = \frac{N}{2}\ln(2\pi\sigma^2) + \frac{1}{2}\ln \mid R(\underline{l}) \mid + \frac{1}{2\sigma^2}(\underline{y}-F\underline{w})^T R^{-1}(\underline{l})(\underline{y}-F\underline{w}) - \ln p(\underline{w},\underline{\sigma},\underline{l}) \tag{3}$$

The solution of this optimization problem can be obtained using Newton's method because analytical expression for the gradient and Hessian matrix of the objective function can be derived. The algorithm can be modified easily to tackle other forms of covariance function. It is worth noting that when $\underline{l}$ approaches a zero vector, $\varepsilon(\underline{x})$ becomes a Gaussian white noise implying there is no probabilistic dependence between the outputs corresponding to different inputs for given $\underline{w}$ and $\underline{\sigma}$.

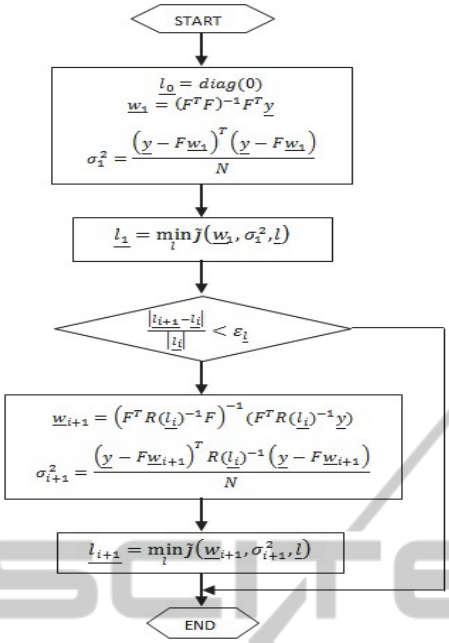

Figure 2: Iterative algorithm for determining the most probable uncertain parameters.

Thus using the proposed method, the optimal combined weights of component networks, $\underline{\sigma}$ and $\underline{l}$ are given by the most probable solution and the uncertainty in these parameters is quantified by the corresponding posterior PDF.

3 COMPUTATIONAL EXPERIMENTS

To verify the performance of the Bayesian based ENN proposed in this paper, peak function is carried out by an ENN program written in MATLAB. The peak function, which is shown in Figure 3, is a function of two variables and obtained by translating and scaling Gaussian distributions. It is a typical complex two-dimensional function as follows:

$$Z = 3(1-x)^2 e^{(-x^2-(y+1)^2)} - 10(x/5 - x^3 - y^5) e^{(-x^2-y^2)} - e^{(-(x+1)^2-y^2)}/3 \quad (4)$$

The peak function contaminated by additive Gaussian white noise with mean 0 and variance 0.05 is used to generate the training data, the cross validation data and the test data. First, 11×11 evenly distributed data along both the x-axis and the y-axis are selected from the domain [-3, 3] as the training data for the simulation. Two other 10×10 evenly distributed points from the same domain are used as the cross validation data and the test data.

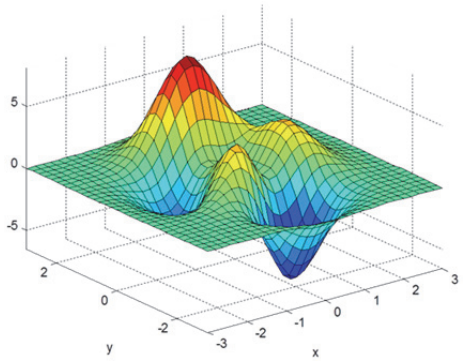

Figure 3: Peak function.

5 cases of ENNs will be investigated later. All ENNs have the same input and output layers: the number of the input nodes is 2 and the number of the output nodes is 1. There are 3 component networks, and the numbers of hidden nodes in the component networks are 11, 14 and 17, respectively. Each component network is trained 3 times randomly to find the best weight configuration within the network. The ENNs will then combine the component networks with the best weight configuration within each component network. For the simple averaging ENN, the output of the ENNs is combined with the simple averaging method (denoted by Ave-ENN). The proposed Bayesian based ENN constructed using the training data and the error covariance function $\mathrm{cov}(\varepsilon(\underline{x}^{(i)}), \varepsilon(\underline{x}^{(j)}); \underline{\sigma}, \underline{l}) = \sigma^2 \delta(\underline{x}^{(i)} - \underline{x}^{(j)})$ is denoted by Btr-ENN and the one using the cross validation data with the same error covariance function is denoted by Bcv-ENN. Btrdp_ENN and Bcvdp-ENN are the same as Btr-ENN and Bcv-ENN, respectively except that the error covariance function $\mathrm{cov}(\varepsilon(\underline{x}^{(i)}), \varepsilon(\underline{x}^{(j)}); \underline{\sigma}, \underline{l}) = \sigma^2 \exp[-(\underline{x}^{(i)} - \underline{x}^{(j)})^T (\underline{x}^{(i)} - \underline{x}^{(j)})/l^2]$. For a fair comparison, the results using single NNs which are used as the component networks in the ENN are also obtained.

The statistical results on the test data set for 20 runs are shown in Table 1, in which Single 11, 14 and 17 denote the single NNs with 11, 14 and 17 hidden nodes, respectively. It can be observed that ENNs have better accuracy than the single NNs. For the single networks, the network with higher number of the hidden nodes has the better performance. When these 3 component networks combined, the performance of ENNs becomes better than any of the single one. Among the ENNs, Bcvdp-ENN has the smallest mean and standard deviation (S.D.) of MSEs for the test data, indicating the best generalization capability and the most stable performance. From the mean and S.D. of MSEs for

the test data, it can be seen that the proposed Bayesian ENN outperforms both the single NNs and the simple averaging ENN.

Table 1: Test MSE of twenty runs on peak function with three component networks.

MSE	Minimum	Mean	S.D.
single 11	0.3418	0.5617	0.1361
single 14	0.2485	0.3683	0.1089
single 17	0.1989	0.3013	0.1049
Ave-ENN	0.2114	0.2726	0.0419
Btr-ENN	0.1756	0.2400	0.0484
Btrdp-ENN	0.1756	0.2397	0.0484
Bcv-ENN	0.1768	0.2331	0.0431
Bcvdp-ENN	0.1766	0.2323	0.0415

4 CONCLUSIONS

This paper improves the existing ENN by the following ways: 1) instead of using component NN directly, a preliminary selecting process is used to get the best component NN; 2) the stochastic system based Bayesian is adopted to construct a methodology to determine the weights of the component networks by using the cross validation data set in the ENN with error term being modelled as a stochastic process in network input variables.

Peak function is used to verify the performance of the proposed ENN. The results show that the proposed Bayesian based ENN outperforms the single NNs and the simple averaging ENN. These results also show the potential of the proposed ENN can be applied to other kinds of problems. Moreover, comparison with other ensemble methodologies is currently under investigation and experiments with additional data sets will be carried out. Further improvements to the proposed method by considering the dependence of measured output with predicted output, multiple optimal models, improving the stochastic modelling, using advanced stochastic simulation algorithms and coupling the construction and combination of component networks for prediction improvement are currently under investigation.

REFERENCES

Beck, J. L., Katafygiotis, L. S., 1998. Updating models and their uncertainties. I: Bayesian statistical framework. *Journal of Engineering Mechanics* 124(4), 455-461.

Buntine, W. L., Weigend, A.S.1991. Bayesian Back-propagation. *Complex Systems* 5, 603-643.

Cheung, S. H., Beck, J. L., 2010. Calculation of posterior probabilities for Bayesian model class assessment and averaging from posterior samples based on dynamic system data. *Computer-Aided Civil and Infrastructure Engineering* 25, 304-321.

Cheung, S. H. and Beck, J. L., 2012. New Bayesian updating methodology for model validation and robust predictions of a target system based on hierarchical subsystem tests. *Computer Methods in Applied Mechanics and Engineering*, accepted for publication.

Friedman, J. H., 1991. Multivariate adaptive regression splines. *Ann Statist* 19(1): 1-82.

Hansen, L. K., Salamon, P., 1990. Neural network ensembles. *IEEE Transactions on Pattern Analysis Machine Intelligence* 12 (10), 993-1001.

Hashem, S., 1993. Optimal Linear Combinations of Neural Networks. PhD thesis, School of Industrial Engineering, Purdue University.

Hippert, H. S., Taylor, J. W., 2010. An evalution of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting, *Neural Networks* 23, 386-395.

Lampinen, J., Vehtari, A., 2001. Bayesian approach for neural networks—review and case studies. *Neural Networks* 14, 257-274.

Marwala, T., 2007. Bayesian training of neural networks using genetic programming. *Pattern Recognition Letters* 28, 1452-1458.

McCulloch, W. S., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 1943; 5, 115-133. Reprinted in Anderson & Rosenfeld 1988, 18-28.

Posada, D., Buckley, T. P., 2004. Model selection and model averaging in phylogenetics: advantages of Akaike information criterion and Bayesian approaches over likelihood ratio tests. *Systematic Biology* 53, 793-808.

Sollich, P., Krogh, A., 1996. Learning with ensembles: How over-fitting can be useful, in: Touretzky, D. S., Mozer, M. C., Hasselmo, M. E. (Eds.), *Advances in Neural Information Processing Systems* 8, Denver, CO, MIT press, Cambridge, MA, pp. 190-196.

Wang, P., Xu, L., Zhou, S., Fan, Z., Li, Y., Feng, S., 2010. A novel Bayesian learning method for information aggregation in modular neural networks. *Expert Systems with Applications* 37, 1071-1074.

Zhao, Z. Y., Zhang, Y., Liao, H. J., 2008. Design of ensemble neural network using the Akaike information criterion. *Engineering Applications of Artificial Intelligence* 21, 1182-1188.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS