

# A Novel Approach for Determining the Optimal Number of Hidden Layer Neurons for FNN's and Its Application in Data Mining

Shuxiang Xu and Ling Chen

**Abstract—** Optimizing the number of hidden layer neurons for an FNN (feedforward neural network) to solve a practical problem remains one of the unsolved tasks in this research area. In this paper we review several mechanisms in the neural networks literature which have been used for determining an optimal number of hidden layer neuron (given an application), propose our new approach based on some mathematical evidence, and apply it in financial data mining. Compared with the existing methods, our new approach is proven (with mathematical justification), and can be easily handled by users from all application fields.

**Index Terms –** neural network, data mining, number of hidden layer neurons.

## I. INTRODUCTION

Feedforward Neural Networks (FNN's) have been extensively applied in many different fields, however, given a specific application, optimizing the number of hidden layer neurons for establishing an FNN to solve the problem remains one of the unsolved tasks in this research area. Setting too few hidden units causes high training errors and high generalization errors due to under-fitting, while too many hidden units results in low training errors but still high generalization errors due to over-fitting. Several researchers have proposed some rules of thumb for determining an optimal number of hidden units for any application. Here are some examples: "A rule of thumb is for the size of this hidden layer to be somewhere between the input layer size and the output layer size ..." [6], "How large should the hidden layer be? One rule of thumb is that it should never be more than twice as large as the input layer..." [5], and "Typically, we specify as many hidden nodes as dimensions needed to capture 70-90% of the variance of the input data set..." [7]

Dr Shuxiang Xu is a lecturer in the School of Computing and Information Systems, University of Tasmania, Launceston, Tasmania 7250, Australia. Email: Shuxiang.Xu@utas.edu.au

Ling Chen is currently working as an IT Officer, Information Services, Department of Health and Human Services, Hobart, Tasmania 7000, Australia. Email: [ling.chen@dhhs.tas.gov.au](mailto:ling.chen@dhhs.tas.gov.au)

ICITA2008 ISBN: 978-0-9803267-2-7

However, most of those rules are not applicable to most circumstances as they do not consider the training set size (number of training pairs), the complexity of the data set to be learnt, etc. It is argued that the best number of hidden units depends in a complex way on: the numbers of input and output units, the number of training cases, the amount of noise in the targets, the complexity of the function or classification to be learned, the architecture, the type of hidden unit activation function, the training algorithm, etc [15].

A dynamic node creation algorithm for FNN's is proposed by Ash in [2], which is different from some deterministic process. In this algorithm, a critical value is chosen arbitrarily first. The final structure is built up through the iteration that a new node is created in the hidden layer when the training error is below the critical value. On the other hand, Hirose et al in [10] propose an approach which is similar to Ash [2] but removes nodes when small error values are reached.

In [13], a model selection procedure for neural networks based on least squares estimation and statistical tests is developed. The procedure is performed systematically and automatically in two phases. In the bottom-up phase, the parameters of candidate neural models with an increasing number of hidden neurons are estimated, until they can not be approved anymore (i.e. until the neural models become ill-conditioned). In the top-down phase, a selection among approved candidate models using statistical Fisher tests is performed. The series of tests start from an appropriate full model chosen with the help of computationally inexpensive estimates of the performance of the candidates, and end with the smallest candidate whose hidden neurons have a statistically significant contribution to the estimation of the regression. Large scale simulation experiments illustrate the efficiency and the parsimony of the proposed procedure, and allow a comparison to other approaches.

The Bayesian Ying-Yang learning criteria [17 – 20] put forward an approach for selecting the best number of hidden units. Their experimental studies show that the approach is able to determine the best number of hidden units with minimized generalization error, and that it outperforms Cross Validation approach in selecting the appropriate hidden unit numbers for both clustering and function approximation.

In [8] an algorithm is developed to optimize the number of hidden nodes by minimizing the mean-squared errors over noisy training data. The algorithm combines training sessions with statistical analysis and experimental design to generate new sessions. Simulations show that the developed algorithm requires fewer sessions establishing the optimal number of hidden nodes, compared with using a straightforward way of eliminating nodes successively one by one.

Three researchers in [11] propose a hybrid optimization algorithm based on the relationship between the sample approximation error and the number of hidden units in an FNN, for simultaneously determining the number of hidden units and the connection weights between neurons. They mathematically prove the strictly decreasing relationship between the sample approximation error and the number of hidden units. They further justify that the global nonlinear optimization of weight coefficients from the input layer to the hidden layer is the core issue in determining the number of hidden units. The synthesis of evolutionary programming and gradient-based algorithm is adopted to find the global nonlinear optimization. This approach is also a deterministic process rather than creating or removing nodes as described before.

In this paper, we propose a novel approach for determining an optimal number of hidden layer neurons for FNN's, and investigate its application in financial data mining. In the following section, mathematical evidence is given which offers theoretical support to the novel algorithm. Experiments are then conducted to justify our new method, which is followed by a summary of this report in the final section.

## II. MATHEMATICAL BACKGROUND

Barron in [4] reports that, using artificial neural networks for function approximation, the rooted mean squared (RMS) error between the well-trained neural network and a target function  $f$  is shown to be bounded by

$$O\left(\frac{C_f^2}{n}\right) + O\left(\frac{nd}{N} \log N\right) \quad (2.1)$$

where  $n$  is the number of hidden nodes,  $d$  is the input dimension of the target function  $f$ ,  $N$  is the number of training pairs, and  $C_f$  is the first absolute moment of the Fourier magnitude distribution of the target function  $f$ .

According to [4], the two important points of the above contribution are the approximation error and the estimation error between the well-trained neural network and the target function. For this research we are interested in the approximation error which refers to the distance between the target function and the closest neural network function of a given architecture (which represents the simulated function). To this point, [4] mathematically proves that, with  $n \sim C_f$

$(N/(d \log N))^{1/2}$  nodes, the order of the bound on the RMS error is optimized to be  $O(C_f((d/N) \log N)^{1/2})$ .

Based on the above result, we can conclude that if the target function  $f$  is known then the best number of hidden layer nodes (which leads to a minimum RMS error) is

$$n = C_f(N/(d \log N))^{1/2} \quad (2.2)$$

Note that the above equation is based on a known target function  $f$ .

However, in most practical cases the target function  $f$  is unknown, instead, we are usually given a series of training input-output pairs. In these cases, [4] suggests that the number of hidden nodes may be optimized from the observed data (training pairs) by the use of a complexity regularization or minimum description length criterion. This analysis involves Fourier techniques for the approximation error, metric entropy considerations for the estimation error, and a calculation of the index of resolvability of minimum complexity estimation of the family of networks. Complexity regularization is closely related to Vapnik's method of structural risk minimization [16] and Rissanen's minimum description-length criterion [12, 3]. It is a criterion which reflects the trade-off between residual error and model complexity and determines the most probable model (in this research, the neural network with the best number of hidden nodes).

## III. OUR NOVEL APPROACH

So when  $f$  is unknown we use a complexity regularization approach to determine the constant  $C$  in the following

$$n = C(N/(d \log N))^{1/2} \quad (3.1)$$

The approach is to try an increasing sequence of  $C$  to obtain different number of hidden nodes, train an FNN for each number of hidden nodes, and then observe the  $n$  which generates the smallest RMS error (and note the value of the  $C$ ). The maximum of  $n$  has been proved to be  $N/d$ . Please note the difference between the equation (3.1) and the equation (2.2): in (2.2),  $C_f$  depends on a known target function  $f$ , which is usually unknown (so (2.2) is only a theoretical approach), whereas in our approach as shown in (3.1),  $C$  is a constant which does not depend on any function.

Based on our experiments conducted so far we have found that for a small or medium-sized dataset (with less than 5000 training pairs), when  $N/d$  is less than or close to 30, the optimal  $n$  most frequently occurs on its maximum, however, when  $N/d$  is greater than 30, the optimal  $n$  is close to the value of  $(N/(d \log N))^{1/2}$ .

## IV. APPLICATION IN DATA MINING

Data Mining is an analytic process designed to explore data (usually large amounts of data - typically business or

market related) in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. The ultimate goal of data mining is prediction - and predictive data mining is the most common type of data mining and one that has the most direct business applications. The process of data mining usually consists of three stages: (1) the initial exploration, (2) model building or pattern identification with validation/verification, and (3) deployment. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. Data mining tools can answer business questions that traditionally are too time-consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations. One of the most commonly used techniques in data mining, Artificial Neural Network (ANN) technology offers highly accurate predictive models that can be applied across a large number of different types of financial problems [1, 9, 14].

For our experiments we use our new approach for determining the best number of hidden layer neurons to establish a standard FNN to simulate and then forecast the Total Taxation Revenues of Australia. Figure 4.1 shows the financial data downloaded from the Australian Taxation Office (ATO) web site. For this experiment monthly data between Sep 1969 and June 1999 are used (358 data points). Based on our new approach, the optimal number of hidden layer neurons for this experiment is  $n=5$ . It's easy to verify whether this is the optimal number simply by setting a different number of hidden layer neurons and then compare the simulation and forecasting errors. The learning algorithm used is an improved back-propagation algorithm from [6].

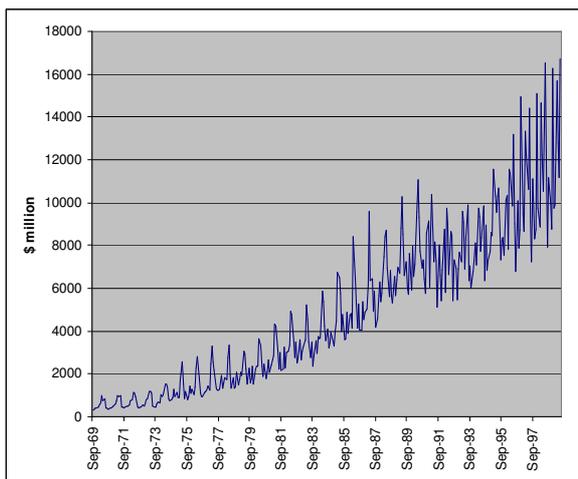


Figure 4.1. Total Taxation Revenues of Australia (\$ million) (Sep 1969 to June 1999)

After the FNN (with 5 hidden layer units) has been well trained over the training data pairs, it is used to forecast the taxation revenues for each month of the period July 1999 –

June 2000. Then the forecasted revenues are compared with the real revenues for the period, and the overall RMS error reaches 5.53%. To verify that for this example the optimal number of hidden layer neuron is 5, we try to apply the same procedure by setting the numbers of hidden layer neurons to 3, 7, 11, and 19, which result in overall RMS errors of 7.71%, 8.09%, 9.78%, and 11.23%, respectively.

Some cross-validation method is used for this experiment: the training data set is divided into a training set made of 70% of the original data set and a validation set made of 30% of the original set. The training (training time and number of epochs) is optimized based on evaluation over the validation set.

## V. SUMMARY AND DISCUSSION

In this paper we review several mechanisms in the neural networks literature which have been used for determining an optimal number of hidden layer neurons, propose our new approach based on some mathematical evidence, and apply it in financial data mining. Our experiment described in section IV and many other experiments not described in this report show that our new approach is in an advantageous position to be applied in practical applications which involve learning small to medium-sized data sets. However, this paper does not address the local minima problems.

It would be a good idea to extend the research to involve large applications which contain training datasets of over 5000 input-out pairs in the future. With large datasets the mechanisms that can be used to determine an optimal number of hidden neurons would be improved based on the current approach.

For the current study we have only considered the input dimension ( $d$ ) and the number of training pairs ( $N$ ) in the data set. A good direction for future research would be to also consider other factors which can affect the determination of an optimal number of hidden layer neurons, such as the amount of noise in the targets, the complexity of the function or classification to be learned, the architecture, the type of hidden unit activation function, and the training algorithm used.

## REFERENCES

- [1] Adriaans, P., Zantinge, D., 1996, *Data Mining*, Addison-Wesley.
- [2] Ash T., 1989, Dynamic node creation in backpropagation networks, *Connection Science*, Volume 1, Issue 4, pp 365 – 375.
- [3] Barron, A. R., Cover, T. M. (1991). Minimum complexity density estimation. *IEEE Transactions on Information Theory*, 37, 1034-1054.
- [4] Barron, A. R., (1994), Approximation and Estimation Bounds for Artificial Neural Networks, *Machine Learning*, (14): 115-133, 1994.
- [5] Berry, M.J.A., and Linoff, G. 1997, *Data Mining Techniques*, NY: John Wiley & Sons.
- [6] Blum, A., 1992, *Neural Networks in C++*, NY: Wiley.

- [7] Boger, Z., and Guterman, H., 1997, "Knowledge extraction from artificial neural network models," *IEEE Systems, Man, and Cybernetics Conference*, Orlando, FL, USA.
- [8] Fletcher, L. Katkovnik, V., Steffens, F.E., Engelbrecht, A.P., 1998, Optimizing the number of hidden nodes of a feedforward artificial neural network, *Proc. of the IEEE International Joint Conference on Neural Networks*, Volume: 2, pp. 1608-1612.
- [9] Han, J., Kamber, M., 2001, *Data Mining : Concepts and Techniques*, Morgan Kaufmann Publishers.
- [10] Hirose Y. Yamashita I.C., Hijjiya S., 1991, Back-propagation algorithm which varies the number of hidden units, *Neural Networks*, Vo1.4. 1991.
- [11] Peng, K., Shuzhi S.G., Wen, C., 2000, An algorithm to determine neural network hidden layer size and weight coefficients, *15th IEEE International Symposium on Intelligent Control*, Rio Patras, Greece, pp. 261-266.
- [12] Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11, 416-431.
- [13] Rivals, I., Personnaz, L., 2000, A statistical procedure for determining the optimal number of hidden neurons of a neural model, *Second International Symposium on Neural Computation (NC'2000)*, Berlin, May 23-26 2000.
- [14] Sarker, R. A., Abbass, H. A., Newton, C. S., 2002, *Data Mining : A Heuristic Approach*, Idea Group Pub./Information Science Publishing.
- [15] Sarle, W. S. 2002, *Neural Network FAQ*, <ftp://ftp.sas.com/pub/neural/FAQ.html>, accessed on 5 Dec 2007.
- [16] Vapnik, V. (1982). *Estimation of Dependences Based on Empirical Data*, New York: Springer-Verlag.
- [17] Xu, L., 1995. Ying-Yang Machine: A Bayesian- Kullback scheme for unified learnings and new results on vector quantization. Keynote talk, *Proceedings of International Conference on Neural Information Processing (ICONIP95)*, Oct. 30 - NOV. 3, 977 – 988.
- [18] Xu, L., 1997. Bayesian Ying-Yang System and Theory as A Unified Statistical Learning Approach: (I) Unsupervised and Semi-supervised Learning. An invited book chapter, S. Amari and N. Kassabov eds., *Brain-like Computing and Intelligent Information Systems*, 1997, New Zealand, Springer-Verlag, pp241-274.
- [19] Xu, L., 1997 . Bayesian Ying-Yang System and Theory as A Unified Statistical Learning Approach: (II) From Unsupervised Learning to Supervised Learning and Temporal Modelling. Invited paper, *Lecture Notes in Computer Science: Proc. of International Workshop on Theoretical Aspects of Neural Computation*, May 26-28, 1997, Hong Kong, Springer-Verlag, pp25-42.
- [20] Xu, L., 1997. Bayesian Ying-Yang System and Theory as A Unified Statistical Learning Approach: (III) Models and Algorithms for Dependence Reduction, Data Dimension Reduction, ICA and Supervised Learning. *Lecture Notes in Computer Science: Proc. of International Workshop on Theoretical Aspects of Neural Computation*, May 26-28, 1997, Hong Kong, Springer-Verlag, pp. 43-60.