An Extreme Learning Machine Algorithm for Higher Order Neural Network Models

Dr Shuxiang Xu
School of Computing, University of Tasmania, Locked Bag 1359
Launceston, Tasmania 7250, Australia
Email: Shuxiang.Xu@utas.edu.au

ABSTRACT
Artificial Neural Networks (ANN) have been widely used as powerful information processing models and adopted in applications such as bankruptcy prediction, predicting costs, forecasting revenue, forecasting share prices and exchange rates, processing documents and many more. This paper uses Extreme Learning Machine (ELM) algorithm for Higher Order Neural Network (HONN) models and applies it in several significant business cases. HONNs are neural networks in which the net input to a computational neuron is a weighted sum of products of its inputs. ELM algorithms randomly choose hidden layer neurons and then only adjust the output weights which connect the hidden layer and the output layer. The experimental results demonstrate that HONN models with ELM algorithm offer significant advantages over standard HONN models as well as traditional ANN models, such as reduced network size, faster training, as well improved simulation and forecasting errors.

KEYWORDS: Higher Order Neural Network, Feedforward Neural Network, Extreme Learning Machine, Financial Forecasting.

1. INTRODUCTION

Business is a diversified field with several general areas of specialisation such as accounting or financial analysis. Artificial Neural networks (ANNs) provide significant benefits in business applications. They have been actively used for applications such as bankruptcy prediction, predicting costs, forecast revenue, processing documents and more (Kurbel et al, 1998 ; Atiya et al, 2001 ; Baesens et al, 2003). Almost any neural network model would fit into at least one business area or financial analysis. Traditional statistical methods have been used for business applications with many limitations (Azema-Barac et al, 1997 ; Blum et al, 1991 ; Park et al, 1993).

While conventional ANN models have been bringing huge profits to many financial institutions, they suffer from several drawbacks. First, conventional ANNs can not handle discontinuities in the input training data set (Zhang et al, 2002). Next, they do not perform well on complicated business data with high frequency components and high order nonlinearity, and finally, they are considered as 'black boxes' which can not explain their behaviour (Blum et al, 1991; Zhang et al, 2002 ; Burns, 1986).

To overcome these limitations some researchers have proposed the use of Higher Order Neural Networks (HONNs) (Redding et al, 1993 ; Zhang et al, 1999 ; Zhang et al, 2000). HONNs are able to provide some explanation for the simulation they produce and thus can be considered as 'open box' rather than 'black box'. HONNs can simulate high frequency and high order nonlinear business data, and can handle discontinuities in the input traning data set (Zhang et al, 2002).

HONNs (Higher Order Neural Networks) (Lee et al, 1986) are networks in which the net input to a computational neuron is a weighted sum of products of its inputs. Such neuron is called a Higher-order Processing Unit (HPU) (Lippman, 1989). It was known that HONN's can implement invariant pattern recognition (Psaltis et al, 1988 ; Reid et al, 1989 ; Wood et al, 1996). Giles in (Giles et al, 1987) showed that HONN's have impressive computational, storage and learning capabilities. In (Redding et al, 1993), HONN's were proved to be at least as powerful as any other FNN (feedforward Neural Network) architecture when the orders of the networks are the same. Kosmatopoulos et al (1995) studied the approximation and learning properties of one class of recurrent HONNs and applied these architectures to the identification of dynamical systems. Thimm et al (1997) proposed a suitable initialization method for HONN's and compared this method to weight initialization techniques for FNNs. A large number of experiments were performed which leaded to the proposal of a suitable initialization approach for HONNs.

Unlike traditional ANN learning algorithms, Extreme Learning Machine (ELM) randomly chooses hidden neurons and analytically determines the output weights (Huang et al 2005, 2006, 2008). Many types of hidden nodes including additive nodes, RBF (radial basis function) nodes, multiplicative nodes, and other non neural alike nodes can be used as long as they are piecewise nonlinear. ELM algorithm tends to generalize better at very fast learning speed: it can learn thousands of times faster than conventional popular learning algorithms (Huang et al 2006).

This paper is organized as follows. Section 2 proposes an ELM learning algorithm for HONN models. Section 3 presents several experiments with results to compare the performance of 4 ANN models: HONN with ELM, standard HONN, traditional MLP (multilayer perceptron) neural network, and RBF (radial basis function) neural network. Finally, Section 4 summarizes this chapter.

2. HONN MODELS WITH ELM ALGORITHM

HONNs (Lee et al, 1986) are networks in which the net input to a computational neuron is a weighted sum of products of its inputs. Such neuron is called a Higher-order Processing Unit (HPU) (Lippman, 1989). The network structure of an HONN is the same as that of a multi-layer FNN. That is, it consists of an input layer with some input units, an output layer with some output units, and at least one hidden layer consisting of intermediate processing units. Usually there is no activation function for neurons in the input layer and the output neurons are summing units, the activation function for hidden layer neurons can be any nonlinear piecewise continuous ones.

Based on a one-dimensional HONN defined in (Zhang et al, 2002), this paper proposes the following ELM algorithm for HONNs. The ELM algorithm was originally proposed by Huang (2006), for Single-Layer Feedforward Neural Networks (SLFN). The main idea of ELM lies in the random selection of hidden neurons with random initialization of the SLFN weights and biases. Then, the input weights and biases do not need to be adjusted during training, only the output weights are learned. The training of the SLFN can be achieved with a few steps and very low computational costs.

Consider a set of S distinct training samples (X_i, Y_i) with $X_i \in R^n$ and $Y_i \in R^m$, where n and m are positive integers. Then an SLFN with N hidden neurons can be mathematically represented by

$$\sum_{i=1}^{N} O_i \times f(w_i \times X_j + b_i), \quad 1 \le j \le S \qquad (2.1)$$

with f being the randomly selected neuron activation function, w_i the input weights, b_i the biases, and O_i the output weights.

In case of two-dimensional HONN with a single hidden layer, equation (2.1) becomes

$$\sum_{i=1}^{NP} O_i \times f\left(w_i \times \begin{bmatrix} X_j \\ H(X_j) \end{bmatrix} + b_i\right), \quad 1 \le j \le S \qquad (2.2)$$

where

$$NP = N + C_N^2 \qquad (2.3)$$

$$H(X_j) = X_j \otimes X_j^T \qquad (2.4)$$

Assume that the single layer HONN approximates the training samples perfectly, then the errors between the estimated outputs are the actual outputs are zero, which means

$$\sum_{i=1}^{NP} O_i \times f\left(w_i \times \begin{bmatrix} X_j \\ H(X_j) \end{bmatrix} + b_i\right) = \begin{bmatrix} Y_j \\ H(Y_j) \end{bmatrix}, \quad 1 \le j \le S \qquad (2.5)$$

where

$$H(Y_j) = Y_j \otimes Y_j^T \qquad (2.6)$$

Then the ELM algorithm, when applies to a HONN, states that with randomly initialized input weights and biases, and with the condition that the randomly selected neuron activation function is infinitely differentiable, then the output weights can be determined so that the single layer HONN provides an approximation of the sample values to any degree of accuracy.

3. HONN MODEL APPLICATIONS IN BUSINESS

In this section, the HONN model with ELM algorithm as defined in Section 2 has been used in several financial applications. The algorithm has been implemented in Java, based on the ANN implementation in Matlab version R2009b. The results are given and discussed.

3.1 Simulating and Forecasting Total Taxation Revenues of Australia

The HONN model has been used to simulate and forecast the Total Taxation Revenues of Australia as shown in Figure 3.1. The financial data were downloaded from the Australian Taxation Office (ATO) web site. For this experiment monthly data between Jan 1994 and Dec 1999 were used. The detailed comparison between the following 4 models has been performed: HONN with ELM, standard HONN, traditional MLP (multilayer perceptron) neural networks, and RBF neural networks. The results are illustrated in Table 3.1.

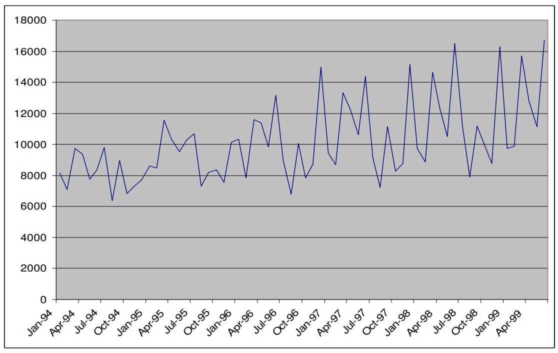

Figure 3.1. Total Taxation Revenues of Australia ($ million) (Jan 1994 To Dec 1999)

Neural Network	No. HL	HL Nodes	Epoch	RMS Error
HONN with ELM	1	4	3,000	0.085468
Standard HONN	1	4	7,000	0.165874
MLP ANN	1	10	12,000	0.856654
RBF ANN	1	10	12,000	0.987996

Table 3.1. HONN with ELM, Standard HONN, MLP, and RBF to Simulate Taxation Revenues (HL: Hidden Layer. RMS: Root-Mean-Square)

After the 4 ANN models have been well trained over the training data pairs, they were used to forecast the taxation revenues for each month of the year 2000. Then the forecasted revenues were compared against the real revenues for the period and the overall RMS errors reached (in order) were 4.55%, 7.65%, 11.23%, and 12.01%, respectively. It was worth noting that HONN with ELM was the fastest model in terms of convergence.

3.2 Simulating and Forecasting Reserve Bank Of Australia Assets

The HONN with ELM has also been used to simulate and forecast the Reserve Bank Of Australia Assets as shown in Figure 3.2. The financial data were obtained from the Reserve Bank Of Australia. For this experiment monthly data between Jan 1980 and Dec 2000 were used. The detailed comparison between the 4 ANN models is illustrated in Table 3.2.

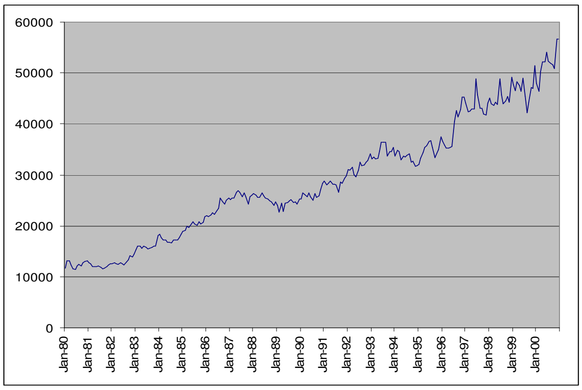

Figure 3.2. Reserve Bank Of Australia Assets ($ million) (Jan 1980 To Dec 2000)

Neural Network	No. HL	HL Nodes	Epoch	RMS Error
HONN with ELM	1	5	3,000	0.090231
Standard HONN	1	5	7,000	0.196753
MLP ANN	1	14	12,000	0.956454
RBF ANN	1	14	12,000	0.997216

Table 3.2. HONN with ELM, Standard HONN, MLP, and RBF to Simulate Reserve Bank Of Australia Assets ($ million) (HL: Hidden Layer. RMS: Root-Mean-Square)

After these 4 ANN models have been well trained over the training data pairs, they were used to forecast the Reserve Bank of Australia Assets for each month of the year 2001. Then the forecasted assets were compared against the real assets for the period and the overall RMS errors reached (in order) were 5.05%, 8.62%, 13.23%, and 11.81%, respectively. HONN with ELM was also the fastest model to converge.

3.3 Simulating and Forecasting Fuel Economy

In the next experiment a dataset containing information of different cars built in the US, Europe, and Japan was trained using the 4 ANN models to determine car fuel economy (MPG - Miles Per Gallon) for each vehicle. There were a total of 392 samples in this data set with 9 input variables and 1 output. The dataset was from UCI Machine Learning Repository (2007). The output was the fuel economy in MPG, and the input variables were:

- number of cylinders
- displacement
- horsepower
- weight
- acceleration
- model year
- Made in US? (0,1)
- Made in Europe? (0,1)
- Made in Japan? (0,1)

To compare the performance of the 4 ANN models the dataset was divided into a subset containing 300 samples for training, a subset containing 53 samples for testing, and the last subset of 39 samples for forecasting (or generalization). Each of the 4 ANN models was trained using the training subset, tested using the testing subset (for adjusting certain parameters), and finally applied on the forecasting subset. The experimental results for the forecasting subset are illustrated in the following Table 3.3.

Neural Network	No. HL	HL Nodes	Epoch	RMS Error
HONN with ELM	1	3	2,000	0.060768
Standard HONN	1	3	5,000	0.152359
MLP ANN	1	9	9,000	0.709845
RBF ANN	1	9	9,000	0.870946

Table 3.3. HONN with ELM, Standard HONN, MLP, and RBF to forecast fuel economy (HL: Hidden Layer. RMS: Root-Mean-Square)

4. CONCLUSIONS

In this paper an HONN with ELM model was introduced and applied in business applications such as simulating and forecasting government taxation revenues. Experiments demonstrated that such model offers significant advantages over standard HONN models and traditional ANN models such as reduced network size, faster training, as well as improved simulation and forecasting errors. It appears that HONN with ELM model works faster than the other ANN models due to the nature of the ELM algorithm. As part of the future research, some current cross-validation approaches may be improved and applied so that the forecasting errors could be reduced further down to a more satisfactory level.

Another direction for future research would be the use of an ensemble of HONN models for modelling and simulation.

REFERENCES

Arai, M., Kohon, R., Imai, H. (1991). Adaptive control of a neural network with a variable function of a unit and its application, *Transactions on Inst. Electronic Information Communication Engineering*, J74-A, 551-559.

Atiya, A.F., (2001). Bankruptcy prediction for credit risk using neural networks: Asurvey and new results, *IEEE Transactions on Neural Networks*, Volume: 12, Issue: 4, page(s): 929-935.

Azema-Barac, M., Refenes, A., (1997). Neural Networks for financial applications, in E. Fiesler & R. Beale (eds) *Handbook of Neural Computation*, Oxford University Press (G6:3:1-7).

Baesens, B., Setiono, R., Mues, C., Vanthienen, J., (2003). Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation, *Management Science*, Volume: 49, Issue: 3.

Barron, A. R., (1994), Approximation and Estimation Bounds for Artificial Neural Networks. *Machine Learning*, (14):115-133.

Blum, E., Li, K., (1991). "Approximation theory and feed-forward networks", *Neural Networks*, Vol.4, pp.511-515.

Burns, T., (1986). The interpretation and use of economic predictions, *Proc. Royal Society A*, pp.103-125.

Campolucci, P., Capparelli, F., Guarnieri, S., Piazza, F., & Uncini, A. (1996). Neural networks with adaptive spline activation function. *Proceedings of IEEE MELECON 96*, Bari, Italy, 1442-1445.

Chen, C.T., Chang, W.D. (1996). A feedforward neural network with function shape autotuning, *Neural Networks*, 9(4), 627-641

Dayhoff, J. E., (1990). *Neural network architectures : an introduction*. New York, N.Y. : Van Nostrand Reinhold.

Gallant, A. R., White, H. (1988) There exists a neural network that does not make avoidable mistakes. *IEEE Second International Conference on Neural Networks*, San Diego, SOS Printing, I: 657-664.

Giles, C.L., Maxwell, T. (1987) Learning, invariance, and generalization in higher order neural networks, *Applied Optics*, 26(23), 4972-4978.

Grossberg, S. (1986). Some nonlinear networks capable of learning a spatial pattern of arbitrary complexity. *Proc. National Academy of Sciences*. 59, 368-372.

Hammadi, NC, Ito, H. (1998). On the activation function and fault tolerance in feedforward neural networks. *IEICE Transactions on Information & Systems*, E81D(1): 66 – 72.

Hansen, J.V., Nelson, R.D. (1997). Neural networks and traditional time series methods: a synergistic combination in state economic forecasts, *IEEE Transactions on Neural Networks*, 8(4), 863-873.

Hinton, G. E. (1989). Connectionist learning procedure, *Artificial Intelligence*, 40, 251 – 257.

Holden, S.B., and Rayer, P.J.W. (1995). Generalisation and PAC learning: some new results for the class of generalised single-layer networks. *IEEE Transactions on Neural Networks*, 6(2), 368 – 380.

Haykin, S. S, (1994). *Neural networks : a comprehensive foundation*. New York : Macmillan.

Hu, Z., Shao, H. (1992). The study of neural network adaptive control systems, *Control and Decision*, 7, 361-366.

Huang, GB, Siew, CK, "Extreme Learning Machine with Randomly Assigned RBF Kernels," International Journal of Information Technology, vol. 11, no. 1, pp. 16—24, 2005.

Huang, GB, Zhu, QY, Siew, CK, "Extreme Learning Machine: Theory and Applications", Neurocomputing, vol. 70, pp. 489-501, 2006.

Huang, GB, Li, MB, Chen, L, and Siew, CK, "Incremental Extreme Learning Machine With Fully Complex Hidden Nodes," Neurocomputing, vol. 71, pp. 576-583, 2008.

Kawato, M., Uno, Y., Isobe, M., Suzuki, R. (1987) A hierarchical model for voluntary movement and its application to robotics, *Proc. IEEE Int. Conf. Network*, IV, 573-582.

Kay, A., (2006). Artificial Neural Networks, Computerworld website, Retrieved on 27 November 2006 from http://www.computerworld.com/softwaretopics/software/appdev/story/0,10801,57545,00.html

Kosmatopoulos, E.B., Polycarpou, M.M., Christodoulou, M.A., Ioannou, P.A. (1995). High-order neural network structures for identification of dynamical systems, *IEEE Transactions on Neural Networks*, 6(2), 422-431.

Kurbel, K., Singh, K., Teuteberg, F. (1998). Search And Classification Of "Interesting" Business Applications In The World Wide Web Using A Neural Network Approach. *Proceedings of the 1998 IACIS Conference*, Cancun, Mexico.

Lee, Y.C., Doolen, G., Chen, H., Sun, G., Maxwell, T., Lee, H., Giles, C.L. (1986) Machine learning using a higher order correlation network, *Physica D: Nonlinear Phenomena*, 22, 276-306.

Lippman, R.P. (1989) Pattern classification using neural networks, *IEEE Commun. Mag*., 27, 47-64.

Park, J., Sandberg, I.W., (1993). "Approximation and radial-basis-function networks", *Neural Computation*, Vol.5, pp.305-316.

Picton, P, (2000). *Neural Networks*. Basingstoke : Palgrave.

Psaltis, D., Park, C.H., Hong, J. (1988) Higher order associative memories and their optical implementations, *Neural Networks*, 1, 149-163.

Redding, N., Kowalczyk A. and Downs, T., (1993). “Constructive high-order network algorithm that is polynomial time”, *Neural Networks*, Vol.6, pp.997-1010.

Redding, N.J., Kowalczyk, A., Downs, T. (1993). Constructive higher-order network algorithm that is polynomial time, *Neural Networks*, 6, 997-1010.

Reid, M.B., Spirkovska, L., Ochoa, E. (1989). Simultaneous position, scale, rotation invariant pattern classification using third-order neural networks, *Int. J. Neural Networks*, 1, 154-159.

Rumelhart, D.E., McClelland, J.L. (1986). *Parallel distributed computing: exploration in the microstructure of cognition*, Cambridge, MA: MIT Press.

Thimm, G., Fiesler, E. (1997). High-order and multilayer perceptron initialization, *IEEE Transactions on Neural Networks*, 8(2), 349-359.

UCI Machine Learning Repository, (2007). ftp://ftp.ics.uci.edu/pub/machine-learning-databases/auto-mpg/auto-mpg.data, accessed in April 2007.

Vecci, L., Piazza, F., Uncini, A. (1998). Learning and approximation capabilities of adaptive spline activation function neural networks, *Neural Networks*, 11, 259-270.

Wood, J., Shawe-Taylor, J. (1996). A unifying framework for invariant pattern recognition, *Pattern Recognition Letters*, 17, 1415-1422.

Xu, S (2009), A Novel Higher Order Artificial Neural Networks, Proceedings of the Second International Symposium on Computational Mechanics (ISCM II), 30 Nov – 3 Dec 2009, Hong Kong – Macau, pp. 1507-1511. ISBN: 978-0-7354-0778-7.

Xu S, Chen L (2009), Adaptive Higher Order Neural Networks for Effective Data Mining, Proceedings of The Sixth International Symposium on Neural Networks (ISNN 2009), Wuhan, China, May 26-29, 2009, pp 165 – 173.

Yamada, T., Yabuta, T. (1992). Remarks on a neural network controller which uses an auto-tuning method for nonlinear functions, *IJCNN*, 2, 775-780.

Zhang, M., Xu, S., and Lu B., (1999). Neuron-adaptive higher order neural network group models, *Proc. Intl. Joint Conf. Neural Networks - IJCNN’99*, Washington, DC, USA, (Paper # 71).

Zhang, M., Xu, S., Fulcher, J., (2002). Neuron-Adaptive Higher Order Neural-Network Models for Automated Financial Data Modeling, *IEEE Transactions on Neural Networks*, Vol 13, No. 1.

Zhang, M., Zhang, J. and Fulcher, J., (2000). “Higher order neural network group models for financial simulation”. *Intl. J. Neural Systems*, vol 12, No. 2, pp 123 –142.