

## Research Article

# Applying Neural Networks to Prices Prediction of Crude Oil Futures

John Wei-Shan Hu,<sup>1,2</sup> Yi-Chung Hu,<sup>2</sup> and Ricky Ray-Wen Lin<sup>3</sup>

<sup>1</sup> Department of Finance, Chung Yuan Christian University, Chung-Li 32023, Taiwan

<sup>2</sup> Department of Business Administration, Chung Yuan Christian University, Chung-Li 32023, Taiwan

<sup>3</sup> Tsann Kuen Trans-Nation Group, Taiwan

Correspondence should be addressed to Yi-Chung Hu, ychu@cycu.edu.tw

Received 2 March 2012; Accepted 12 June 2012

Academic Editor: Jung-Fa Tsai

Copyright © 2012 John Wei-Shan Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The global economy experienced turbulent uneasiness for the past five years owing to large increases in oil prices and terrorist's attacks. While accurate prediction of oil price is important but extremely difficult, this study attempts to accurately forecast prices of crude oil futures by adopting three popular neural networks methods including the multilayer perceptron, the Elman recurrent neural network (ERNN), and recurrent fuzzy neural network (RFNN). Experimental results indicate that the use of neural networks to forecast the crude oil futures prices is appropriate and consistent learning is achieved by employing different training times. Our results further demonstrate that, in most situations, learning performance can be improved by increasing the training time. Moreover, the RFNN has the best predictive power and the MLP has the worst one among the three underlying neural networks. This finding shows that, under ERNNs and RFNNs, the predictive power improves when increasing the training time. The exceptional case involved BPNs, suggesting that the predictive power improves when reducing the training time. To sum up, we conclude that the RFNN outperformed the other two neural networks in forecasting crude oil futures prices.

## 1. Introduction

During the past three years, the global economy has experienced dramatic turbulence owing to uneasiness because of terrorists' attacks and rapidly rising oil prices. For example, the US light crude oil futures price rapidly climbed to the all-time peak about US\$80 recently. Simultaneously, the US Federal Reserve continuously increased its benchmark short-term interest rates by seventeen times to prevent inflation till August 2006. Consequently, many governments and corporate managers attempted to seek a method of accurately forecasting the crude oil prices.

Accurate prediction of crude oil price is important yet extremely complicated and difficult. For example, Kumar [1] found that the traditional model-based forecasts had larger errors compared to forecast crude oil prices using futures price. However, Pham and Liu [2] and Refenes et al. [3] showed that neural networks had significant performance of forecasting. According to Chen and Pham [4], numerous real-world application problems cannot be fully described and handled via classical set theory. Meanwhile, fuzzy set theory can deal with partial membership. Although Omlin et al. [5] argued that fuzzy neural networks (FNNs) combine the advantages of both fuzzy systems and neural networks, whereas Omlin et al. [5] proposed that most of the FNNs could only process state input-output relationships, FNNs were unable to process temporal input sequences with arbitrary length. Since recurrent neural networks (RNNs) are dynamic systems involving temporal state representation, RNNs are computationally powerful. Although the Elman recurrent neural network (ERNN) is a special case of RNN and is less efficient than standard RNN, Pham and Liu [2] posited that ERNN can model a very large class of linear and nonlinear dynamic systems. Additionally, Lee and Teng [6] argued that the RFNNs have the same advantages as RNNs and extended the application domain of the FNNs to temporal problems. These findings motivate us to apply three neural network models (namely, traditional backpropagation neural networks (BPNs), Elman recurrent neural networks (ERNNs), and recurrent fuzzy neural networks (RFNNs)) to forecast crude oil futures prices.

The focus of this paper is to apply neural networks for predicting crude oil futures prices. This work has the following objectives: forecast crude oil futures prices using BPNs, ERNNs, and RFNNs; compare the learning and predictive performance among BPNs, ERNNs, and RFNNs, and explore how training time affects prediction accuracy.

This study classifies the previous literature into three main groups: (1) the studies that compared artificial neural networks (ANNs) with other methods to forecast futures prices, (2) the works that combined fuzzy systems with recurrent neural networks, and (3) the researches that examined the evolution or forecasting accuracy of energy futures prices.

The following studies have applied various ANNs to predict futures prices. Refenes et al. [3], Castillo and Melin [7], Giles et al. [8], Donaldson and Kamstra [9], and Sharma et al. [10] all demonstrated that neural networks outperformed classical statistical techniques in forecasting ability. Although Kamstra and Boyd [11] also found that ANNs outperformed the naive model for most commodities in forecasting ability, yet Kamstra and Boyd found that ANNs have less predictive power than linear model for barley and rye.

The following works combined fuzzy system with recurrent neural networks. Omlin et al. [5], Juang and Lin [12], Nürnberger et al. [13, 14], Zhang and Morris [15], Giles et al. [8], Lee and Teng [6], Mastorocostas and Theocharis [16, 17], Juang [18], Yu and Ferreyra [19], Hong and Lee [20], and Lin and Chen [21] all designed to combine recurrent neural networks (RNNs) with fuzzy system for identification and prediction.

The following researches examined the evolution or forecasting accuracy of energy prices. Hirshfeld [22], Ma [23], Serletis [24], Kumar [1], Pindyck [25], Adrangi et al. [26], and Krehbiel and Adkins [27] reviewed or examined the energy futures prices and the price risk. Among these literatures, Adrangi et al. [26] found strong evidence of nonlinear dependencies in crude oil futures prices, but the evidence is not consistent with chaos.

The paper is organized as follows. Three kinds of artificial neural networks are described in Section 2. The performance of the proposed learning algorithms is examined by the computer simulations are described in Section 4. Conclusion is presented in Section 5. The performance valuation method is presented in Section 3.

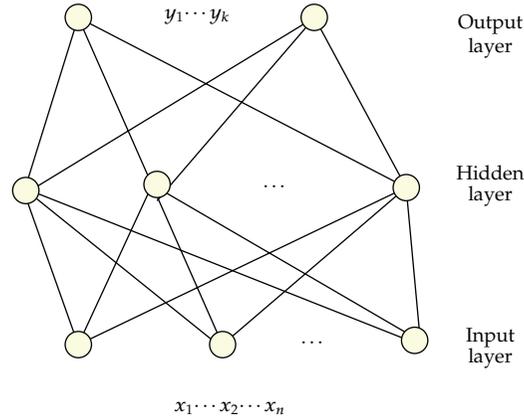


Figure 1: Structure of multilayer perceptron.

## 2. Neural Networks for Crude Oil Forecasting

### 2.1. Multilayer Perceptron (MLP)

As a promising generation of information processing system that expresses the ability to learn, recall, and generalize based on training patterns or data, artificial neural networks (ANNs) are interconnected assembly of simple processing nodes, whose functionality is similar to human neurons. ANNs have become popular during the last two decades for diverse applications, ranging from financial prediction to machine vision. According to Refenes et al. [3], the main potentials with respect to ANNs include the following: (1) handling complex nonlinear function; (2) learning from training data; (3) adapting to changing environments; (4) handling incomplete, noisy, and fuzzy information; (5) performing high-speed information processing. The most popular and widespread method used to train the multilayer perceptron (MLP) is the back propagation algorithm. MLP can be interpreted as a universal approximators and is used to estimate the parameter values via a gradient descent algorithm in problems involving nonlinear regression. The popularity of MLP is based on the simplicity and power of the underlying algorithm. Figure 1 shows the structure of MLP.

BPN involves two steps. The first step generates a forward flow of activation from the input layer to the output layer via the hidden layer.

The sigmoid function is usually served as

$$y = \frac{1}{(1 + e^{-\alpha \sum x_i})}, \quad (2.1)$$

where  $\alpha \in R$ . An activation function can be differentiated since the steepest descent method is employed to derive the weight updating rule. The response of the hidden layer is the input of the output layer. In the second step, an overall error,  $E_T$ , which is the difference between the actual and the desired output, is minimized employing a supervised learning task performed by MLP:

$$E_T = \sum_p E_p = \frac{1}{2} \sum_p \sum_k (d_{pk} - y_{pk})^2, \quad (2.2)$$

where  $E_T$  denotes the total error for a neural network across the entire training set,  $E_p$  represents the network error for the  $p$ th pattern,  $d_{pk}$  denotes the desired output of the  $k$ th unit in the output layer for pattern  $p$ , and  $y_{pk}$  is the actual output of the  $k$ th unit in the output layer for the  $p$ th pattern.

Then the gradient method is applied to optimize the weight vector of  $E_T$  to minimize the summed square error between the actual and the desired network outputs throughout the training period. The network weight is adjusted whenever a training data is inputted. The size of the adjustment is positively related to the sensitivity of the error function to weight connections. The general weight updating rule for the connection weight between the  $i$ th input node and the  $j$ th output node is as follows:

$$\Delta w_{ji} = -\eta \frac{\partial E_T}{\partial w_{ji}}, \quad (2.3)$$

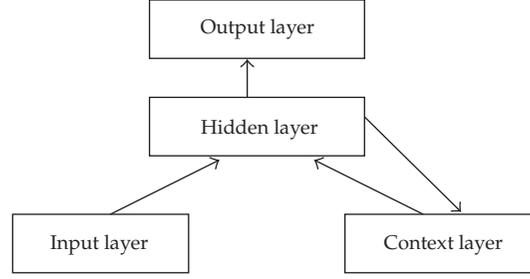
where  $\eta$  is the learning rate.

BPNs can be widely applied to sample identification, pattern matching, compression, classification, diagnosis, credit rating, stock price trend forecasting, adaptive control, functional link, optimization, and data clustering. They can also be trained via a supervised learning task to reduce the difference between the desired and the actual outputs and have high learning accuracy. Yet BPNs have the following weaknesses: (1) slow learning speed, (2) long executing time, (3) very slow convergence; (4) falling into a local minimum of error functions, (5) lack of systematic methods in the network dynamics, (6) inability to use past experience to forecast its future behavior. This study further uses two dynamic neural networks to predict the crude oil futures prices.

## 2.2. Recurrent Neural Networks

Recurrent neural networks (RNNs) were first developed by Hopfield [28] in a single-layer form and later were developed using multilayer perceptrons comprising concatenated input-output, processing, and output layers. An RNN is a dynamic neural network that permits self-loops and backward connections so that the neurons have recurrent actions and provide local memory functions. The feedback within RNN can be achieved either locally or globally. The ability of feedback with delay provides memory to the network and is appropriate for prediction. According to Haykin [29], RNN can summarize all required information on past system behavior to forecast future behavior. The ability of RNN to dynamically incorporate past experience based on internal recurrence makes it more powerful than BPN. The structure of RNN generally comprises the following: (a) recurring information from the output or hidden layer to the input layer, and (b) mutual connection of neurons within the same layer. The advantages of RNNs are as follows: (1) fast learning speed, (2) short executing time, and (3) fast converging to a stable state.

Refenes et al. [3] argued that RNNs exhibited full connection between each node and all other nodes in the network, whereas partial recurrent networks contain a number of specific feedback loops. They quoted Hertz et al. [30] who assigned the name Elman architecture when the feedback to the network input is from one or more hidden layers. This name originated from Elman [31] who designed a neural network with recurrent links for providing networks with dynamic memory. The Elman recurrent neural network (ERNN) is a temporal and simple recurrent network with a hidden layer, assuming that the neural network operates



**Figure 2:** Structure of the simple recurrent neural network.

using discrete time steps. The activations of the hidden units at time  $t$  are fed backwards and serve as inputs to “context layer” at time  $t+1$  and thus represent a form of short-term memory that enables limited recurrence. Moreover, the feedback links run from the hidden layer to the context layer and produce both temporal and spatial patterns. As depicted in Figure 2, this network is a two-layer network involving feedback in the first layer [32].

### 2.3. Elman Recurrent Neural Network

According to Hammer and Nørskov [33], ERNN is a special case of RNN, differing mainly in that the learning algorithm is simply a truncated gradient descent method and training is less efficient than the standard method employed for RNN. However, Pham and Liu [2] argued that ERNN can be used to model a very large class of linear and nonlinear dynamic systems. Li et al. [34] designed a new nonlinear neuron-activation function into the framework of an Elman network, thus enabling the neural network to generate chaos. Li et al. [34] described a context layer  $x^c(\kappa)$  in the Elman network as follows:

$$x^c(\kappa) = x(\kappa - 1). \quad (2.4)$$

The descriptive equations of ERNN can be considered as a nonlinear state-space model in which all weight values are constant following initialization:

$$x_i(\kappa) = f\left(\sum_{j=1}^N w_{ij}^x x_j(\kappa - 1) + w_i^u u(\kappa - 1)\right), \quad (2.5)$$

$$y(\kappa) = \sum_{i=1}^n w_i^y x_i(\kappa), \quad (2.6)$$

where  $w_{i,j}^x$  demonstrates weight linking the  $i$ th hidden-layer neuron and the  $j$ th context-layer neuron,  $w_i^u$  indicates weight linking the input neuron  $u(\kappa - 1)$  and the  $i$ th hidden-layer neuron,  $w_i^y$  refers to weight linking the output neuron  $y(\kappa)$  and the  $i$ th hidden-layer neuron,  $f(\cdot)$  expresses nonlinear activation function in the hidden-layer node, and  $n$  is the number of hidden-layer nodes. Since it is difficult to interpret the network functions of RNNs, this study further incorporates the fuzzy logic into RNNs.

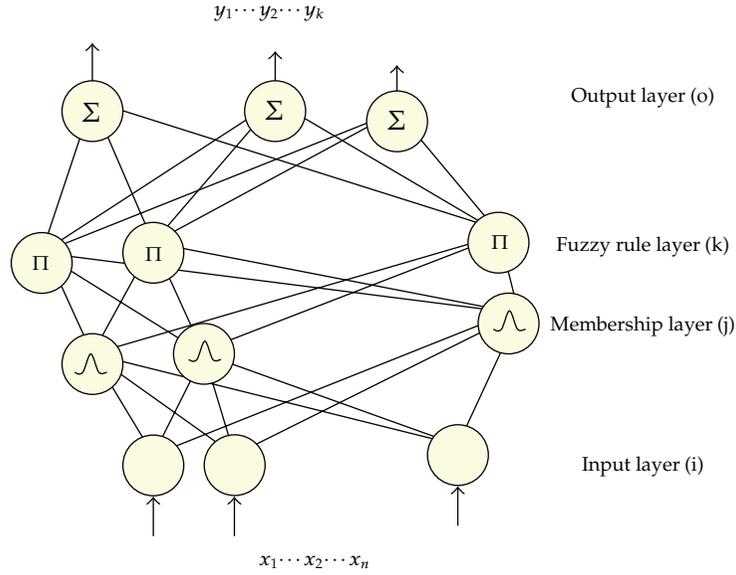


Figure 3: A structure of the recurrent fuzzy neural networks.

#### 2.4. Recurrent Fuzzy Neural Networks

Fuzzy sets theory has first been introduced by Zadeh [35, 36]. Zimmermann [37] argued that fuzzy set theory can be adapted to different circumstances and contexts. Buckley and Hayashi [38] stated that fuzzy neural network (FNN) is a layered, feedforward, neural net that has fuzzy set signals and weights. Neural networks may utilize the data bank to train and learn, while the solution obtained by fuzzy logic may be verified by empirical study and optimization. Omlin et al. [5] noted that FNN comprises both clear physical meanings and good training ability. However, FNN only applies to static problems.

Besides the fact that RNN's underlying theory is complicated and RNN is difficult to interpret, Hu and Chang [39] also found that there are limitations to forecast the accurate valuation for the long-term period by both BPN and RNN. This study thus uses a recurrent fuzzy neuron network (RFNN) model in addition to BPN and ERNN. According to Lee and Teng [6], RFNN has several key aspects: dynamic mapping capability, temporal information storage, universal approximation, and the fuzzy inference system. Li et al. [34] also argued that RFNN has the same dynamic and robust advantage as RNN. In addition, the network function can be interpreted using fuzzy inference mechanism. Therefore, long-term prediction fuzzy models can be easily implemented using RFNNs. The network output in RFNN is fed back to the network input using one or more time delay units. As depicted in Figure 3, the general microstructure of RFNN consists of four layers in general: an input layer, a membership layer, a fuzzy rule layer, and an output layer.

The information transmission process and basic functions of each layer are as follows.

- (1) Input layer: the input nodes in this layer represent input variables. The input layer only transmits the input value to the next layer directly and no computation is

conducted in this layer. From (2.5), the connection weight at the input layer ( $w_i^1$ ) is unity:

$$g_i^{(1)} = x_i^{(1)}. \quad (2.7)$$

- (2) Membership layer: the membership layer is also known as a fuzzification layer and contains several different types of neurons, each neuron performs membership function. The membership nodes in this layer correspond to the linguistic label of the input variables in the input layer and serve as a unit of memory. Each of these variables is transformed into several fuzzy sets in the membership layer where each neuron corresponds to a particular fuzzy set, with the actual membership function being provided by the neuron output. Each neuron in this layer represents characteristics of each membership function, and Gaussian function serves as the membership function. The  $j$ th neuron in this layer has the following input and output:

$$g_{i(t)}^{(2)} = \exp \left[ -\frac{[x_{ij}^{(2)} - m_{ij}]^2}{\sigma_{ij}^2} \right], \quad (2.8)$$

where

$$x_{ij}^{(2)}(t) = g_i^{(1)}(t) + g_{ij}^{(2)}(t-1)\theta_{ij}, \quad (2.9)$$

$m_{ij}$  denotes the mean value of a Gaussian membership function of the  $j$ th term with respect to the  $i$ th input variable,  $\sigma_{ij}$  represents the standard derivation of the Gaussian type membership function of the  $j$ th term with respect to the  $i$ th input,  $x_{ij}^{(2)}(t)$  is the input of this layer at the discrete time  $t$ ,  $g_{ij}^{(2)}(t-1)$  denotes the feedback unit of memory which stores the past network Information and represents the main difference between FNN and RFNN, and  $\theta_{ij}$  indicates the connection weight of the feedback unit.

Each node in the membership layer possesses three adjustable parameters:  $m_{ij}$ ,  $\sigma_{ij}$ , and  $\theta_{ij}$ .

- (3) Fuzzy rule layer: The fuzzy rule layer comprises numerous nodes, each node corresponds to a fuzzy operating region of the process being modeled. This layer constructs the entire fuzzy rule data set. The nodes in this layer equal the number of fuzzy sets corresponding to each external linguistic input variable and receive the one-dimensional membership degree of the associated rule from the nodes of a set in the membership layer. The output of each neuron in the fuzzy rule layer is obtained by using a multiplication operation. The input and output for the  $k$ th neuron in the fuzzy rule layer are as follows:

$$g_i^{(3)} = \Pi_i x_i^{(3)} = \exp \left\{ -D_i(x_i^{(2)} - m_i)^T [D_i(x_i^{(2)} - m_i)] \right\}, \quad (2.10)$$

where  $D_i = \text{diag}[1/\sigma_{1i}, 1/\sigma_{2i}, \dots, 1/\sigma_{ni}]$ ,  $X_i = [x_{1i}, x_{2i}, \dots, x_{ni}]^T$ ,  $m_i = [m_{1i}, m_{2i}, \dots, m_{ni}]$ ,  $x_j^3$  is the  $i$ th input value that inputs to the neuron of the fuzzy rule layer, and  $g_i^{(3)}$  denotes the output of a fuzzy rule node representing the “firing strength” of its corresponding rule. Links before and fuzzy rule layer indicate the preconditions of the rules, and links after and fuzzy rule layer demonstrate the consequences of the fuzzy rule nodes.

- (4) Output layer: the output layer performs the defuzzification operation. Nodes in this layer are called output linguistic nodes, where each node is for an individual output of the system. The links between the fuzzy rule layer and the output layer are connected by the weighting values  $w_{jk}$ .

For the  $k$ th neuron in the output layer,

$$y_k = g_k^{(4)} = \sum_i^m w_{jk} g_j^{(3)}, \quad (2.11)$$

where  $w_{jk}$  is the output action strength of the  $k$ th output associated with the  $j$ th fuzzy rule and serves as the tuning factor of this layer,  $g_k^{(4)}$  is the final inferred result, and  $y_k$  represents the  $k$ th output of the FRNN.

### 3. Valuation Performance

This work uses the mean square error (MSE) method to assess the performance of three neural networks. The MSE is calculated as the average of the sum of the square of the error, which is given by the difference between the actual and the designed output. MSE thus is computed as

$$\text{MSE} = \frac{1}{T - (T_1 - 1)} \sum_{t=T_1}^T (r_t - \hat{r}_t)^2, \quad (3.1)$$

where  $T$  indicates the total number of samples,  $T_1$  refers to the number of estimated samples,  $r_t$  represents the actual output, and  $\hat{r}_t$  denotes the desired output.

## 4. Computer Simulations

### 4.1. Data Description

This study focuses on energy futures for the near-month. Daily oil prices for Brent, WTI, DUBAI, and IPE are used in this investigation. The data sources are obtained from the Energy Bureau of USA and International Petroleum Exchange (IPT) of the Great Britain. This work explores the influence of training times on prediction performance so that it classifies the training period from January 1, 1990 to April 30, 2005 into three five-year sections. Table 1 shows the different training periods.

**Table 1:** The periods of the training patterns.

Data set	Training period	Training times (sec.)
I	2000.01.01–2004.12.31.	1244
II	1995.01.01–2004.12.31.	2499
III	1990.01.01–2004.12.31.	3758

**Table 2:** The training times of three parts of various ANNs.

Data set	Backpropagation	Elman RNN	RNN with fuzzy	Times for each ANN
I	BPN1	ERNN1	RFNN1	1244
II	BPN2	ERNN2	RFNN2	2499
III	BPN3	ERNN3	RFNN3	3758

## 4.2. Comparison among Various Neural Networks

This work divides the training period into three parts and uses Matlab software to perform training and testing. In order to compare the predictive power of these three artificial neural network (ANNs), the training function, namely, Levenberg Marquardt method, is employed, and the number of iterations over the data set is arbitrarily set to 1000 in order to train individual neural networks.

Table 2 shows the symbolization and training times of multi-layer perceptron (MLP), Elman recurrent neural networks (ERNNs) and recurrent fuzzy neural networks (RFNNs).

### (1) The Comparison of Learning Performance

Following 1000 training times, Table 3 illustrates the following ranking for the learning performance of the three ANNs: RFNN ranks first, followed by ERNN, and finally BPN. Table 3 shows that the learning performance of the ANNs improves with increasing training time of ANNs. One exceptional case is the MSE at part 2 under RFNN, which is less than that obtained from part 3.

### (2) The Comparison of Predictive Power

The empirical results indicate that the predictive power of the three ANNs is ranked as follows: RFNN ranks first, followed by ERNN, and finally MLP. Table 4 shows that, under ERNNs and RFNNs, the predictive power of the ANNs improves with increasing training time. However, the predictive power of MLP differs from those of ERNNs and RFNNs as the predictive power of the MLP retrogresses with increasing training time. One possible explanation for this phenomenon is that a large difference exists between the forecasting value and the actual value from March 20, 2005 to March 28, 2005. The MLP is not a dynamic network and it cannot be applied by the past experience to the behavioral forecasting.

## 5. Conclusion

This study uses multi-layer perceptron (MLP), Elman recurrent neural networks (ERNNs) and recurrent fuzzy neural networks (RFNNs) to forecast the crude oil prices and compare

**Table 3:** The comparison of the learning ability (MSE) for the three neural networks.

ANNs	Data set			MSE
	I	II	III	
BPN	$6.84589E - 05$	$4.13072E - 05$	$3.75245E - 05$	$4.90969E - 05$
ERNN	$6.53719E - 05$	$3.71397E - 05$	$3.42490E - 05$	$4.355769E - 05$
RFNN	$2.43727E - 05$	$1.66990E - 05$	$1.78329E - 05$	$1.96349E - 05$

**Table 4:** The comparison of the predictive power of the three neural networks.

ANNs	Part 1	Part 2	Part 3	MSE average value
MLP	$2.6652E - 03$	$2.6653E - 03$	$2.6767E - 03$	$2.6691E - 03$
ERNN	$2.6730E - 03$	$2.6558E - 03$	$2.6449E - 03$	$2.6579E - 03$
RFNN	$2.6085E - 03$	$2.6074E - 03$	$2.6045E - 03$	$2.6068E - 03$

the predictive power of the above three neural network models. Results of this work are summarized as follows.

All of the MSE values obtained under different training times through MLP, ERNNs, and RFNNs are below 0.0026768, suggesting that the use of the neural networks to forecast the crude oil futures prices is appropriate, and consistent learning ability can be obtained by using different training times. This investigation confirms that, under most circumstances, the more training times the neural networks take, the more the learning performance of the neural networks improves. The only exceptional case occurs at part 2 under the RFNN model, where MSE is slightly less than that obtained from part 3.

Regarding the predictive power of the three neural networks, this study finds that RFNN has the best predictive power and MLP has the least predictive power among the three neural networks. This work also finds that, under ERNNs and RFNNs, the predictive power improves when increasing the training time. However, the results are different from those obtained under MLP, indicating that the predictive power improves when decreasing the training time. Possible explanation for this phenomenon is the existence of a large difference between the predictive value and the actual value during a 9-day period. To summarize, this study concludes that the recurrent fuzzy neural network is the best among the three neural networks.

## Acknowledgments

The authors would like to thank Dr. Oleg Smirnov for his insightful comments at the 81st WEA Annual Conference on July 1, 2006. Also, the authors would like to thank the anonymous referees for their valuable comments. This research is partially supported by the National Science Council of Taiwan under grant NSC 99-2410-H-033-026-MY3.

## References

- [1] M. S. Kumar, "The forecasting accuracy of crude oil futures prices," *IMF Staff Papers*, vol. 39, no. 2, pp. 432–461, 1992.
- [2] D. T. Pham and X. Liu, *Neural Networks for Identification Prediction and Control*, Springer, London, UK, 1995.

- [3] A. Nicholas Refenes, A. Zapranis, and G. Francis, "Stock performance modeling using neural networks: a comparative study with regression models," *Neural Networks*, vol. 7, no. 2, pp. 375–388, 1994.
- [4] Chen and Pham, "Time series prediction combining dynamic system, theory and statistical methods," in *Proceedings of the IEEE/IAFE 1995 Computational Intelligence for Financial Engineering*, pp. 151–155, 2001.
- [5] C. W. Omlin, K. K. Thornber, and C. L. Giles, "Fuzzy finite-state automata can be deterministically encoded into recurrent neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 1, pp. 76–89, 1998.
- [6] C.-H. Lee and C.-C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 4, pp. 349–366, 2000.
- [7] O. Castillo and P. Melin, "Intelligent system for financial time series prediction combining dynamical systems theory, fractal theory, and statistical methods," in *Proceedings of the IEEE/IAFE Computational Intelligence for Financial Engineering (CIFEr '95)*, pp. 151–155, April 1995.
- [8] C. L. Giles, D. Chen, G. Z. Sun, H. H. Chen, Y. C. Lee, and M. W. Goudreau, "Constructive learning of recurrent neural networks: limitations of recurrent cascade correlation and a simple solution," *IEEE Transactions on Neural Networks*, vol. 6, no. 4, pp. 829–836, 1995.
- [9] R. G. Donaldson and M. Kamstra, "Forecast combining with neural networks," *Journal of Forecasting*, vol. 15, no. 1, pp. 49–61, 1996.
- [10] J. Sharma, R. Kamath, and S. Tuluca, "Determinants of corporate borrowing: an application of a neural network approach," *American Business Review*, vol. 21, no. 2, pp. 63–73, 2003.
- [11] I. Kamstra and M. S. Boyd, "Forecasting futures trading volume using neural networks," *The Journal of Futures Markets*, vol. 15, no. 8, pp. 935–970, 1995.
- [12] C. F. Juang and C. T. Lin, "A recurrent self-organizing neural fuzzy inference network," *IEEE Transactions on Neural Networks*, vol. 10, no. 4, pp. 828–845, 1999.
- [13] A. Nurnberger, A. Radetzky, and R. Kruse, "Determination of elastodynamic model parameters using a recurrent neuro-fuzzy system," in *Proceedings of the 7th European Congress on Intelligent Techniques and Soft Computing*, Verlag Mainz, Aachen, Germany, 1999.
- [14] A. Nürnbergger, A. Radetzky, and R. Kruse, "Using recurrent neuro-fuzzy techniques for the identification and simulation of dynamic systems," *Neurocomputing*, vol. 36, pp. 123–147, 2001.
- [15] J. Zhang and A. J. Morris, "Recurrent neuro-fuzzy networks for nonlinear process modeling," *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 313–326, 1999.
- [16] P. Mastorocostas and J. Theocharis, "FUNCOM: a constrained learning algorithm for fuzzy neural networks," *Fuzzy Sets and Systems*, vol. 112, no. 1, pp. 1–26, 2000.
- [17] P. A. Mastorocostas and J. B. Theocharis, "A recurrent fuzzy-neural model for dynamic system identification," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 32, no. 2, pp. 176–190, 2002.
- [18] C. F. Juang, "A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 155–170, 2002.
- [19] W. Yu and A. Ferreyra, "System identification with state-space recurrent fuzzy neural networks," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, December 2004.
- [20] Y.-Y. Hong and C.-F. Lee, "A neuro-fuzzy price forecasting approach in deregulated electricity markets," *Electric Power Systems Research*, vol. 73, no. 2, pp. 151–157, 2005.
- [21] C. J. Lin and C. H. Chen, "Identification and prediction using recurrent compensatory neuro-fuzzy systems," *Fuzzy Sets and Systems*, vol. 150, no. 2, pp. 307–330, 2005.
- [22] D. J. Hirshfeld, "A fundamental overview of the energy futures market," *The Journal of Futures Market*, vol. 3, no. 1, pp. 75–100, 1983.
- [23] C. W. Ma, "Forecasting efficiency of energy futures prices," *The Journal of Futures Markets*, vol. 9, pp. 373–419, 1989.
- [24] A. Serletis, "Rational expectations, risk and efficiency in energy futures markets," *Energy Economics*, vol. 13, no. 2, pp. 111–115, 1991.
- [25] R. S. Pindyck, "Long-run evolution of energy prices," *Energy Journal*, vol. 20, no. 2, pp. 1–24, 1999.
- [26] B. Adrangi, A. Chatrath, K. K. Dhanda, and K. Raffiee, "Chaos in oil prices? Evidence from futures markets," *Energy Economics*, vol. 23, no. 4, pp. 405–425, 2001.
- [27] T. Krehbiel and L. C. Adkins, "Price risk in the NYMEX energy complex: an extreme value approach," *Journal of Futures Markets*, vol. 25, no. 4, pp. 309–337, 2005.
- [28] J. J. Hopefield, "Networks and physical systems with emergent collective," *Proceedings of National Academy Science*, vol. 79, pp. 2554–2558, 1982.

- [29] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 1999.
- [30] J. A. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley Longman, Boston, Mass, USA, 1991.
- [31] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.
- [32] C.-T. Lin and C. S. G. Lee, *Neural Fuzzy System: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice Hall, Upper Saddle River, NJ, USA, 1996.
- [33] B. Hammer and J. K. Nørskov, "Theoretical surface science and catalysis-calculations and concepts," *Advances in Catalysis*, vol. 45, pp. 71–129, 2000.
- [34] X. Li, Z. Chen, Z. Yuan, and G. Chen, "Generating chaos by an Elman network," *IEEE Transactions on Circuits and Systems I*, vol. 48, no. 9, pp. 1126–1131, 2001.
- [35] L. Zadeh, "Fuzzy Sets Yulia," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [36] L. Zadeh, "Fuzzy sets and systems," in *Proceedings of the Symposium on Systems Theory*, Polytechnic Institute of Brooklyn, April 1965.
- [37] H.-J. Zimmermann, *Fuzzy Set Theory and Its Applications*, Kluwer Academic Publishers, Boston, Mass, USA, 1985.
- [38] J. J. Buckley and Y. Hayashi, "Fuzzy neural networks: a survey," *Fuzzy Sets and Systems*, vol. 66, no. 1, pp. 1–13, 1994.
- [39] J. W.-S. Hu and C.-H. Chang, "Euro/USD currency option valuation combining two artificial neural network models and various volatilities," in *Proceedings of the 80th Western Economic Association Annual Conference (WEA '05)*, San Francisco, Calif, USA, July 2005.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

