Learning Recurrent ANFIS Using Stochastic Pattern Search Method

Hang Yu† and Zheng Tang†,

University of Toyama, Toyama-shi, 930-8555 Japan

Summary
Pattern search learning is known for simplicity and faster convergence. However, one of the downfalls of this learning is the premature convergence problem. In this paper, we show how we can avoid the possibility of being trapped in a local pit by the introduction of stochastic value. This improved pattern search is then applied on a recurrent type neuro-fuzzy network (ANFIS) to solve time series prediction. Comparison with other method shows the effectiveness of the proposed method for this problem.
Key words:
stochastic pattern search method, ANFIS, , time series prediction.

1. Introduction

In recent years, the study of time series such as approximation, modulation, prediction and others constitutes a useful task for many fields of research. A reliable system is a model that is able to forecast with minimal error to yield good preparation for the future and serves as a good decision-making. For these goals, different methods have been applied: linear methods such as ARX, ARMA, etc. [1], and nonlinear ones such as artificial neural networks [2]. In general, these methods try to build a model of the process where the last value of the series is used to predict the future values. The common difficulty of the conventional time series modeling is the determination of sufficient and necessary information for an accurate prediction.
On the other hand, neural fuzzy networks [3, 4] have become a popular research topic [3-5]. They are widely applied in fields such as time series prediction [6], control problem [7], and pattern recognition [8]. The integration of neural network and fuzzy logic knowledge combines the semantic transparency of rule-based fuzzy systems with the learning capability of neural networks. However, a major disadvantage of existing neuro-fuzzy systems is that their application is limited to static problems as a result of their internal feed forward network structure. Therefore, without the aid of tapped delays, it cannot represent a dynamic mapping such as in the case of recurrent networks [9-11].
Taking this into consideration, we choose ANFIS which is a pioneering result of the early years of neuro fuzzy. In fact, it is also regarded to be one of the best in function approximation among the several neuro-fuzzy models [12]. As mentioned earlier, since ANFIS is based on a feed forward structure, it unable to handle time series patterns successfully because it does not have any dynamics features as in the case of recurrent network. Despite of the research that has already been done in the area of neuro-fuzzy systems the recurrent variants of this architecture are still rarely studied, although the most likely first approach was presented already several years ago [13]. Thus, in this paper we perform a time series prediction on a modified structure of ANFIS with self feedbacks. By doing so, we can forego the necessity of preprocessing the time series data to map the dynamic structure of the network.
As reported by Y.Bengio [14] in his paper, gradient-based optimization is not suitable to train recurrent type networks. Simi1ar results were also obtained by Mozer [15], where it was found that that back-propagation was not sufficiently powerful to discover contingencies spanning long temporal intervals. Learning based on gradient descent learning algorithms includes real-time recurrent learning (RTRL) [16], ordered derivative learning [17] and so on [18]. Disadvantages of these methods include the complexity of learning algorithms and local minimum problems. When there are multiple peaks in a search space, search results are usually stuck in a local solution by the gradient descent learning algorithm. To avoid these disadvantages, parameter design by genetic algorithms (GAs) seems to be a good choice. However, the learning speed of GA is not satisfactory and sometimes difficult to find convergence.
Most of these algorithms suffer from local optimal problem due to the fact that the error function is the superposition of nonlinear activation that may have minima at different points which often results in non convex error surface. Motivated by this understanding, we introduced a random operator in order to provide the mechanism required to escape the local pit and at the same time reduce the possibility of premature convergence.
The concept of pattern search is to find a better candidate nearby the current one before move to the next stage of search [19]. By introducing a random operator, we temporary increase the error when it comes across a local pit but by doing so, we are also creating a boast much required to escape the pit.
In this paper we proposed improved pattern search learning for a time series prediction for a neuro-fuzzy model that was designed to learn and optimize a hierarchical fuzzy rule base with feedback connections. The recurrent nature of the ANFIS networks allows us to

Manuscript received June 5, 2010
Manuscript revised June 20, 2010

store information of prior system states internally, which may also lead to a reduced complexity, since no additional input variables providing information of prior system states have to be used. By using an improved pattern search, we can often times avoid local minima problems and thus have a higher probability of success for attaining the global maxima. Moreover, the advantages of the proposed learning are that since it is based on pattern search, it is simple and can be implemented with short computation time. Furthermore, the mathematical calculation is also easier to understand as compared to the tedious derivative calculations found in back propagation or the complicated process in genetic learning.

2. Recurrent type neuro-fuzzy network

The network consists of five layers as proposed by Jang in his paper [20]. The details are introduced in the following.

Layer 1: Fuzzification Layer

$$O_{1,i} = \mu_{Ai}(x) \quad (1)$$

Layer 1 is the input layer and it specifies the degree to which the given x satisfies the quantifier *Ai*. A bell-shaped membership function is chosen with minimum and maximum set to 0 and 1 while the parameters of a_i, b_i, c_i is the parameter set, where b is a positive value and c locates the center of the curve. As the values of these parameters change, the bell-shaped functions varies accordingly, thus exhibiting various forms of membership functions on linguistic label *Ai*

$$\mu_{Ai}(x) = \frac{1}{1+\left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}} \quad (2)$$

Layer 2: Rule layer where by the incoming signals are multiplied by the AND operation and sends out the product as the firing strength of a rule.

$$O_{2,i} = w_i = \mu_{Ai}(x) * \mu_{Bi}(y) \quad (3)$$

Layer 3: Normalization of firing strengths

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1 + w_2} \text{ for } i = 1, 2\ldots.. \quad (4)$$

This layer calculates the normalized firing strengths by calculating the ratio of the *i*-th rule's firing strength to the sum of all rule's firing strengths.

Layer 4: Defuzzification layer

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) \ \ldots\ldots\ldots \quad (5)$$

where $\overline{w_i}$ is the output from layer 3 and *pi, q, ri* is the parameter set.

Layer 5: Summation

$$O_{5,i} = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (6)$$

It computes the overall output as the summation of all the incoming signals.

As mentioned earlier in the introduction, the conventional ANFIS network is adapted with 2 feedback connections so that the result of the time series of the previous input can be fed back from the time sequence to the current input.

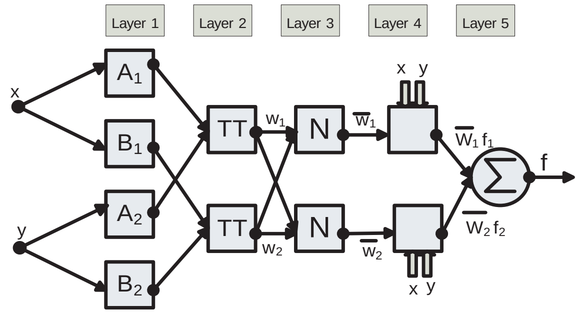

Fig1: The structure of the recurrent type neuro-fuzzy network

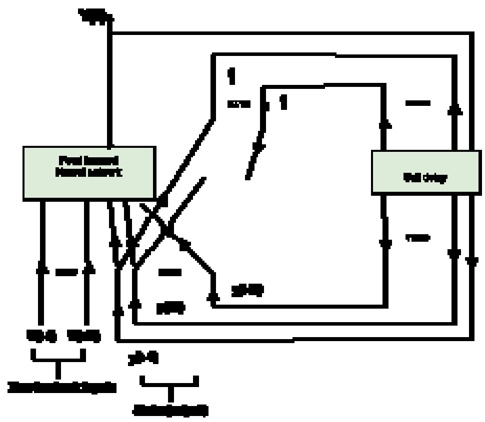

Fig 2: Output Self-feedback

As shown in Fig 2 and Fig 3, the output of the feed forward network is fed-back to the input of the system. The output at time *t* is defined as *y* (*t*), *f* represents a feed forward ANFIS network and external variable *U* represent the network input at a defined time. Therefore it can be represented as such:

$$y(t) = f[y(t-1), \ldots.. y(t-N), U(t-1), \ldots\ldots\ldots U(t-M)] \quad (7)$$

By doing so, we are able to train the canonical ANFIS as a recurrent of order *N* by providing the time delay element at the output.

For the error self-feedback, it is similar to the NARMAX (Non-linear Auto Regressive Moving Average model with eXogenous variables) approach. It predicts a time series *y* at time *t* using as regressors the last *p* values of the series itself and the network input *U* of the last *P* values. Also included is the last *P* value of the prediction error, which forms a self feedback layer. The non-linear function *f* represents a feed forward ANFIS network and its weights. The feed-back connection from the output is feedback to the input node by the equation stated below:

$y(t) = f[Yp(t-1)\ldots\ldots\ldots Yp(t-N), U(t-1); e(t-1)\ldots\ldots\ldots\ldots\ldots e(t-P)]$ (8)

where the error, $e(t) = Yp(t) - Y(t)$

The error feedback describes the difference relating the current output to combinations of inputs and past outputs. The proposed model is well suited to identify with the temporal behavior of FF networks by passing the error through a time-delay back to the network inputs.

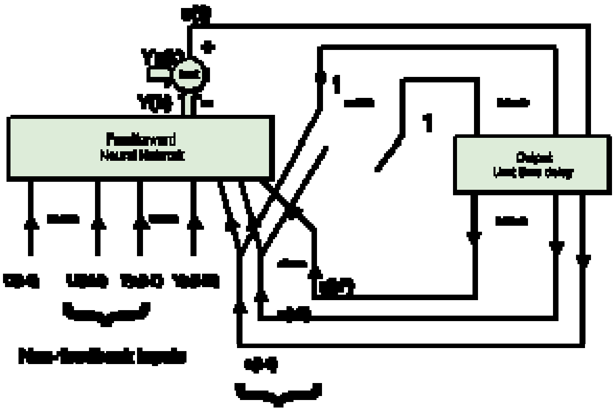

Fig 3: Error Self-Feedback

3. Traditional Pattern Search Method

Pattern search has become a widely accepted technique for the solution of hard combinatorial optimization problems. Some of the earlier works of pattern search have already been described in the late fifties and the early sixties but it is only in the last ten to fifteen years that pattern search algorithms have become very popular and successfully applied to many problems. The renewed interest in pattern search algorithms has several reasons. An important aspect is that pattern search algorithms are intuitively understandable, flexible, generally easier to implement than exact algorithms, and in practice have shown to be very valuable when trying to solve large instances. The solution of large instances has been made feasible by the development of more sophisticated data structures, for example, to search more efficiently the neighborhood of solutions and the enormous increase in computer speed and memory availability.

Pattern search technique is based on iterative exploration of neighborhoods of solutions trying to improve the current solution by local changes. The search starts from an initial solution α and continues to replace α with a better solution in the neighborhood $N(\alpha)$ until no better solution is found. $N(\alpha)$ is a set of solutions obtained from α with a slight perturbation. Based on our multi-layered feed forward network, the neighborhood consists of the weights between all layers threshold of the neurons.

$$N = [w_{11}, w_{12,}\ldots\ldots w_{ij}]^T \quad (9)$$

By interactively adjusting N through the search, we are able to minimize the error. The error for this system can be defined as the difference between the actual system output and the expected output. The search begins at some initial feasible solution, N_0 and uses a subroutine improve to search for a better solution in the N neighborhood as defined previously. The direction it takes can be either one of the many n directions which represents the number of elements in the neighborhood, N. The first direction vector e_k^1 at k iteration can be defined as such:

$$e_k^1 = (0\ldots\ldots 0, \underset{1}{1}, 0,\ldots\ldots\ldots 0)^T \quad (10)$$

$$E(N_k + \Delta_k e_k^1) < E(V_k) \text{ or} \quad (11)$$

$$E(N_k - \Delta_k e_k^1) < E(V_k) \quad (12)$$

In which case 1 is initiated as 1,2,….n when is a positive step size parameter. When such a point is found, then the iteration is declared successful, subsequent is

$$N_{k+1} = N_k + \Delta_k e_k^1 \text{ or} \quad (13)$$

$$N_{k+1} = N_k + \Delta_k e_k^1 \quad (14)$$

The iteration can be termed as unsuccessful if no such point is found. Instead a pattern search option would be taken and the next iteration would be the same as the current point, $N_{k+1} = N_k$. The new step size would be reduced to $\eta\Delta_k$, where $0 < \eta < 1$. A constant for all the iterations provided with $N = 0$ and 0. As for the batch mode the weights of the network are updated only after the entire pattern search training set has been applied to the network. The gradients calculated at each training example are added together to determine the change in the weights.

4. Stochastic Pattern Search Method

During a pattern search, “bumpy" landscapes are serious concern because it can result in a minimum within the neighborhood which might not be a global minimum. Although some problems have been alleviated by increasing the step size, this has often times lead to inaccurate solutions. Furthermore, different step sizes cause the search along different paths, effecting final outcomes. As pointed out by Magoulus [21], by providing a mechanism to escape the local minimum, we are able to overcome this problem. The improved approach of pattern search introduces randomness into a function estimation procedure to improve the search performance. It also modifies the evaluation function of a penalty weight of a random function. This random penalty provides an escape route from a local pit by making a move from the current candidate solution's to a higher function values of some of its neighbors. By doing so, we temporary experience an error increase but nevertheless move away from the stagnation.

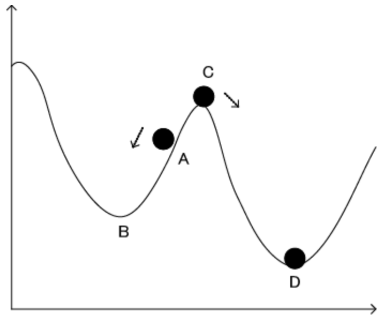

Fig 4: Conceptual graph of the stochastic method

The basic idea is explained in the following. Fig.4 is a conceptual graph of the error landscape with a local minimum and global minimum. The X-coordinate denotes the state of the network and the Y-coordinate denotes the value of error function. For example, if the network is initialized onto point A. Because of the mechanism of the local search method, the state of network moves towards decrease direction and reaches the local minimum (Point B). If we change the dynamics of the MVL at point A to increase the value of error temporarily, point A can become a new point C. From point C, the network returns to move towards decrease direction and reaches the global minimum point D.

The random penalty for the above algorithm is given by

$$\mu(t) = \lfloor random(h(t), 1) \rfloor \quad (15)$$

where the function of random(a,b) returns a value between *a* and *b*. The function $\lfloor x \rfloor$ removes the fractional part of *x* and returns an integer value.

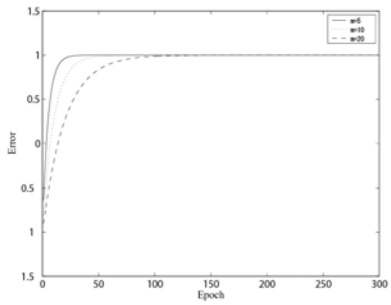

Fig 5: The characteristics graph of h(x)

If *x* is negative, it returns the first negative integer less than or equal to *x*. Parameter *t* denotes the epoch. As for $h(t) = 1 - 2e^{-t/m}$, whose characteristics graph is shown in Fig.5 , with varying the value of *m*, we noticed that climb to attain the saturated level becomes slower. At the beginning, $\mu(t)$ appears randomly as 1, 0 or -1 but as time goes on it eventually becomes 1. The three possible conditions with $\mu(t)$ can be summarized as following:

Algorithm scheme for Improved Pattern search

procedure step HLS(α,N)
input problem instance α, candidate solution N
output candidate solution N^*
1. Begin
2. α := chosen element of Assing(N)
3. For try:=1, ..., maxTries do
4. if ($\Delta E^+ < 0$ and $\Delta E^- < \Delta E^+$) then
5. $N_{k+1} = N_k + \mu(t)\Delta_k e_k^1$
6. Else if ($\Delta E^- < 0$ and $\Delta E^- < \Delta E^+$) then
7. $N_{k+1} = N_k - \mu(t)\Delta_k e_k^1\ k$
8. Else
9. $N_{k+1} = N_k$
10. N^* = localsearch(α, N_{k+1} , N)
11. EndIf
12. EndFor
13. End

Condition 1: $\mu(t)$ = -1. The sequence of iteration is selected contrary to the original pattern search direction and so the value of error increases temporarily.
Condition 2: $\mu(t)$ = 0. No move is taken, remains at the current value. No change in error.
Condition 3: $\mu(t)$ = 1. The algorithm is similar to the pattern search.

5. Simulation Results

In the following section, we discusses the simulation that has been carried out with the proposed method on 4 types of time series data set such as Mackey Glass, sunspot data, laser series and Box-Jenkins. Comparison with other proven methods is carried out in order to show the effectiveness of the proposed model.

5.1 Mackey-Glass data

The method has also been applied to the well-known Mackey–Glass chaotic series given by the following equation.

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t) \quad (16)$$

Table 1: Comparison of Mackey Glass test results for various methods

Models[ref #]	RMSE
ANFIS and fuzzy system [20]	0.007
PG-RBF network [22]	0.0028
Neural tree [23]	0.0069
Radial basis function network [24]	0.0015
Local linear wavelet with hybrid learning [25]	0.0036
Evolving radial basis function with input selection[26]	0.00081
Proposed Improved Pattern search	0.00050

The objective is to compare our results with those obtained by other authors on the same data. The parameters of the series are the following: τ =17, the sampling rate is Δ= 6, the training and the test sets are set to 500 respectively. Table 1 contains the results obtained with our method (last line), and those obtained with other methods. Our method is shown to be more efficient since it increases the prediction accuracy on the test set.

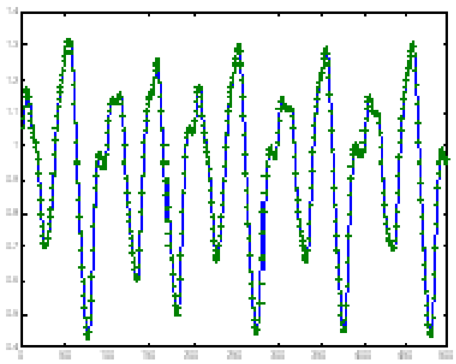

Figure 6 (a) Mackey's Prediction Vs Actual

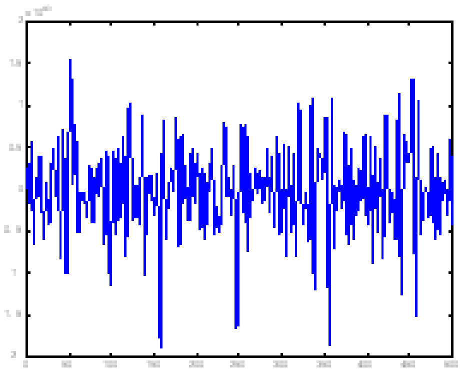

Figure 6 (b) Mackey's Prediction Error

Based on Table 1, it is obvious that with our proposed learning, we are able to attain much smaller RMSE as compared to the rest of the methods especially the more recently reported work by H.Du's ERB with genetic learning and Chen's local linear wavelet method.

5.2 Box-Jenkins Data

The gas furnace data (series J) of Box and Jenkins (1970) is well known and frequently used as a benchmark example for testing identification and prediction algorithms. The data set consists of 296 pairs of input-output measurements. The input u(*t*) is the gas flow into the furnace and the output *y*(*t*) is the CO_2 concentration in outlet gas. The sampling interval is 9s. For this simulation, 4 inputs variables are used for constructing a proposed model. Following previous researchers in order to make a meaningful comparison, the inputs of the prediction model are selected as u (t-4) and y (t-1) and the output is y (t-P).

Table 2: Comparison of Box- Jenkins test results for various methods

Method	RMSE
ARMA	0.843
Tong's model	0.685
Pedryc's model	0.566
Xu's model	0.573
Sugeno's model	0.596
Surmann's model	0.400
Lee's model [17]	0.638
Lin's model [18]	0.511
Nie's model [20]	0.412
ANFIS model [10]	0.085
FuNN model [11]	0.071
HyFIS model [16]	0.042
Neural tree model [4]	0.026
Chen's LWNN	0.01095
Proposed Improved Pattern search	0.0065

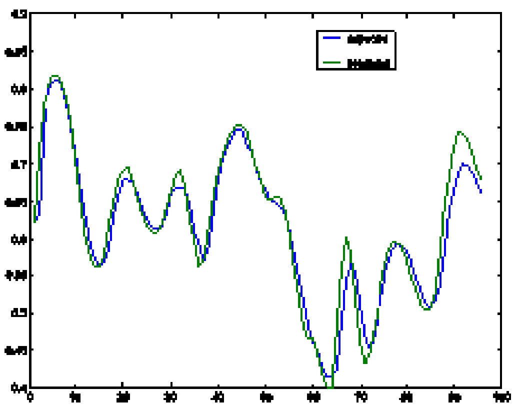

Figure 7 (a) Box-Jenkins Prediction Vs Actual

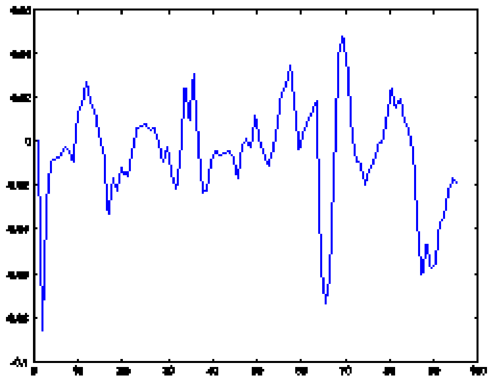
Figure 7(b) Box-Jenkins Prediction Error

For box Jenkins case, we try to predict y(t) based on the Nie's approach as cited Table 4. Compared with the recent result presented as published [26], we can see the developed improved ANFIS with proposed learning method can achieve higher prediction accuracy than the rest of the cited works. The predicted time series and the desired time series are plotted as well as the prediction error.

As seen in the simulation, all the 4 sets of data series shows the lowest predicted error when compared to some other methods. Results obtained by using improved pattern search shows good predictability ability and are suitable to be used in time series prediction.

5.3 Laser data

Laser data was utilized in the 1992 Santa Fe time series competition. The laser generated data consists of intensity measurements made on an 81.5 micron 14NH3 cw (FIR) laser. The data are a cross-cut through periodic to chaotic intensity pulsations of the laser. The chaotic pulsations follow the theoretical Lorenz model of a two level system. The data series were scaled between [0, 1]. The calculated NMSE is compared to the result of other works.

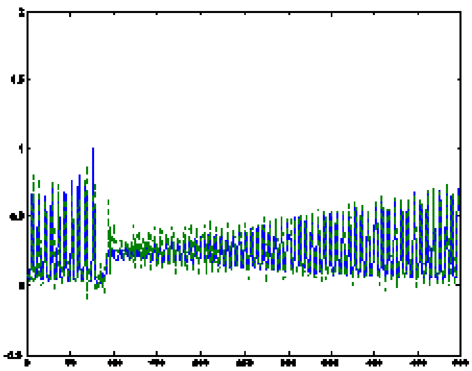
Figure 8(a) Laser Data Prediction Vs Actual

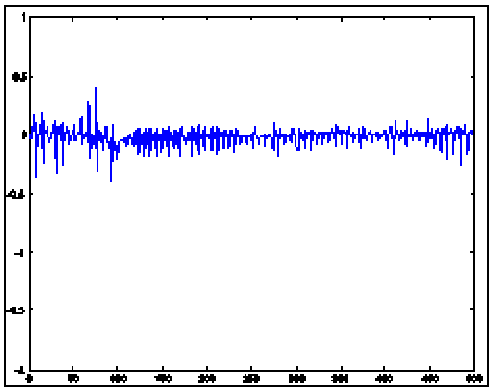
Figure 8(b) Laser Data Prediction Error

Table 3: Comparison of Laser test results for various methods

Models[ref number]	*NMSE*
FIR network [31]	0.00044
Multiscale ANN [32]	0.00074
Proposed Improved Local .Search	0.000025

6. Conclusion

In this paper, a method for predicting time series was presented by using a recurrent based ANFIS network. The improvement made to the conventional ANFIS network is to further strengthen the capability of handling temporal data series data. The success of any neuro-fuzzy model not only depends on the structural layout but also on the learning algorithm since the appropriate tuning of membership function and the rules plays an important role in improving the prediction accuracy. Based on this assumption, we improved the canonical local by introducing a stochastic parameter in order to provide the required escape needed to avoid a local pit. By doing so, we not only retain the advantages of the conventional pattern search but also improving the disadvantages of local optimum problem. The various data series simulations clearly show the effective and the superiority of the proposed algorithm in a time series application.

References

[1] L. Ljung, *System Identification Theory for User*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[2] A. Weigend, N. Gershenfeld, *Times Series Prediction: Forecasting the Future and Understanding the Past,* Addison-Wesley, Reading, MA, 1994.

[3] J.S.R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," IEEE Trans. Systems Man Cybern. Vol. 23, pp. 665–685, 1993.

[4] C.T. Lin, C.S.G. Lee, *Neural Fuzzy Systems: A Neural-fuzzy Synergism to Intelligent Systems,* Prentice-Hall, Englewood Cliffs, NJ, May 1996.

[5] G. Castellano, A.M. Fanelli, C. Mencar, "A neuro-fuzzy network to generate human-understandable knowledge from

data," Cognitive Systems Res. J. Vol. 3 No. 2, pp. 125–144, 2002.

[6] N.K. Kasabov, Q. Song, "DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction," IEEE Trans. Fuzzy Systems Vol. 10, No. 2, pp. 144–154, 2002.

[7] C.J. Lin, C.H. Chen, "Nonlinear system control using compensatory neuro-fuzzy networks," IEICE Trans. Fundamentals, Vol. E86-A No.9, pp.2309–2316, 2003.

[8] J.H. Chiang, P.Y. Hao, "A new kernel-based fuzzy clustering approach: support vector clustering with cell growing," IEEE Trans. Fuzzy Systems Vol.11 No 4, pp. 518–527, 2003.

[9] C.F. Juang, "ATSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms," IEEE Trans. Fuzzy Systems Vol. 10 No. 2, pp. 155–170, 2002.

[10] C.H. Lee, C.C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks, IEEE Trans. Fuzzy Systems," Vol. 8, No. 4, pp. 349–366, 2000.

[11] P.A. Mastorocostas, J.B. Theocharis, "A recurrent fuzzy-neural model for dynamic system identification," IEEE Trans. Systems Man Cybernet. Vol. 32, No.2, pp. 176–190, 2002.

[12] A. Abraham and B. Nath, "Optimal design of neural nets using hybrid algorithms," PRICAI 2000, Melbourne, VIC (2000).

[13] V. Gorrini, H. Bersini, "Recurrent fuzzy systems, in: Proceedings of the 3rd Conference on Fuzzy Systems (FUZZ-IEEE 94)," IEEE, Orlando, 1994.

[14] Y.Bengiot, P.Frasconit and P.Simardt, "The Problem of Learning Long-Term Dependencies in Recurrent Networks, Neural Networks, Vol. 3, pp. 1183 – 1188, 1993.

[15] M.C. Mozer, *Induction of multiscale temporal structure,* Advances in Neural Information Processing Systems 4**,** (eds. Moody, Hanson, Lippman), Morgan Kauffmann, pp. 275-282, 1992.

[16] R.J. Williams, D. Zipser, "A learning algorithm for continually running recurrent neural networks," Neural Comput. Vol. 1, No. 2, pp. 270–280, 1989.

[17] S. Satini, A.D. Bimbo, R. Jain, "Block-structured recurrent neural networks," Neural Networks Vol. 8, No.1, pp. 135–147, 1995.

[18] B.A. Pearlmutter, "Gradient calculations for dynamic recurrent neural networks: a survey," IEEE Trans. Neural Networks Vol. 6, No. 5, pp. 1212–1228, 1995.

[19] A. Kolen, E. Pesch, "Genetic pattern search in combinatorial optimization," Discr. Appl. Math. Combin. Oper. Res. Comput. Sci. Vol. 48, pp. 273–284, 1994..

[20] J.R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," IEEE Trans. Syst Man Cybern Vol.23, No.3, pp.665-685, 1993.

[21] G.D.Magoulas, M.N. Vrahatis and G.S.Androulakis,"' On the alleviation of the problem of local minima in back-propagation"', NonLinear Analysis,Theory, Methods & Applications, Vol. 30, No. 7, pp.4545-4550,1997.

[22] I. Rojas, et al., "Time series analysis using normalized PG-RBF network with regression weights," Neurocomputing Vol. 42, pp. 167-285, 2002.

[23] Y.H. Chen, et al., "Nonlinear system modelling via optimal design of neural trees," Int. J. Neural Sys. Vol. 14, No. 2, pp. 125-137, 2004.

[24] C. Harpham, C.W. Dawson, "The effect of different basis functions on a radial basis function network for time series prediction:a comparative study," Neurocomputing Vol. 69, No. 16, pp. 2161-2170, 2006.

[25] Y.H. Chen, et al., "Time series prediction using a local linear wavelet neural network," Neurocomputing Vol. 69, pp. 449-465, 2006.

[26] H. Du, N.Zhang, "Time series prediction using evolving radial basis function networks with new encoding scheme," Neurocomputing (2007) article in press.

[27] C.J. Lin, Y.J. Xu, "A self-adaptive neural fuzzy network with group-based symbiotic evolution and its prediction applications," Fuzzy Sets and Systems Vol. 157, pp. 1036 - 1056, 2006.

[28] K. Chellapilla, S. Rao, "Optimization of bilinear time series models using fast evolutionary programming," IEEE Transaction on Signal Processing Letters Vol. 5, No. 2, pp. 39-42, 1998.

[29] A.S.Weigend, B.A.Huberman, D.E. Rumelhart, "Predicting the Future:A connectionist approach,Int'l. J. of Neural systems," Vol. 1, No. 3, pp. 209, 1990.

[30] E.Wan, "Combining fossils and sunspots: Committee predictions," In IEEE Int'l Conf. on Neural Network, Vol. 4, pp. 2176-2180, Houstan, USA, 1997.

[31] E.Wan, "Finite Impulse Response Neural Networks with Applications in Time Series Prediction," PhD thesis (Standford University, 1993).

[32] A.B. Geva, "ScaleNet- multiscale neural-network architecture for time series prediction," IEEE Trans. on Neural Networks, Vol. 9, No. 5, pp. 1471-1482, 1998.

[33] M.S. Kim, C.H. Kim, J.J. Lee, "Evolving compact and interpretable Takagi-Sugeno fuzzy models with a new encoding scheme," IEEE Trans. Syst. Man Cybern. Vol. 36, No. 5, pp.1006–1023, 2006.

[34] M. Sugeno, T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," IEEE Trans. Fuzzy Syst. Vol. 1, No.1, pp. 7–31, 1993.

[35] Y. Lin, G.A. Cunningham, "A new approach to fuzzy-neural system modeling," IEEE Trans. Fuzzy Syst. Vol. 3, No. 2, pp. 190–197, 1995.

[36] J. Nie, "Constructing fuzzy model by self-organizing counterpropagation network," IEEE Trans. Syst. Man Cybernet. Vol. 25, No. 6, pp. 963–970, 1995.

Yu Hang received the B.S. degree from Jiangxi University of Finance and Economics, Jiangxi, China in 2006 and an M.S. degree from University of Toyama, Toyama, Japan in 2009. Now he is working towards the D.E. degree at University of Toyama, Toyama, Japan. His main research interests are intelligence computing and neural networks.