Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 144 (2018) 23–32

Procedia Computer Science

www.elsevier.com/locate/procedia

INNS Conference on Big Data and Deep Learning 2018

Wind Speed Intervals Prediction using Meta-cognitive Approach

Nguyen Anh[a], Mukesh Prasad[b], Narasimalu Srikanth[a], Suresh Sundaram[a,*]

[a]*Nanyang Technological University, Singapore*

[b]*Centre for Artificial Intelligence, School of Software, FEIT, University of Technology Sydney, Australia*

Abstract

In this paper, an interval type-2 neural fuzzy inference system and its meta-cognitive learning algorithm for wind speed prediction is proposed. Interval type-2 neuro-fuzzy system is capable of handling uncertainty associated with the data and meta-cognition employs self-regulation mechanism for learning. The proposed system realizes Takagi-Sugeno-Kang inference mechanism and adopts a fast data-driven interval-reduction method. Meta-cognitive learning enables the network structure to evolve automatically based on the knowledge in data. The parameters are updated based on an extended Kalman filter algorithm. In addition, the proposed network is able to construct prediction intervals to quantify uncertainty associated with forecasts. For performance evaluation, a real-world wind speed prediction problem is utilized. Using historical data, the model provides short-term prediction intervals of wind speed. The performance of proposed algorithm is compared with existing state-of-the art fuzzy inference system approaches and the results clearly indicate its advantages in forecasting problems.

Keywords: wind forecasting, fuzzy logic, interval type-2 fuzzy systems, meta-cognition

1. Introduction

Wind energy is growing as an critical source of renewable energy. As wind power becomes more common, forecasting that is integrated into energy management system is increasingly valuable to power system operators. Recently, artificial neural network, fuzzy rule based system and neuro-fuzzy inference system have improved forecasts drastically [15, 2, 12, 3, 25, 28]. It has been shown that these network are strongly capable of predicting and interpreting the approximation function. However, all of the mentioned algorithms lacks of ability to handle uncertainty in the data because these neuro-fuzzy inference systems employ type-1 fuzzy set, which has certain membership function. It has been shown in the literature that Type-2 fuzzy sets and Type-2 fuzzy logic systems employing these sets are able to deal with uncertainty [13, 19]. Due to the massive computation, Interval Type-2 Fuzzy sets are proposed to minimize the effort of handling uncertainty in data [17]. Based on these interval type-2 fuzzy sets, different fuzzy inference systems, known as Interval Type-2 fuzzy inference system or IT2FIS, and their learning algorithm have

* Corresponding author. Tel.: +65-6790-6185.
E-mail address: ssundaram@ntu.edu.sg

1877-0509
10.1016/j.procs.2018.10.501

been proposed in literature [30, 20, 16, 11, 7]. However, these algorithms cannot handle the temporally varying data in a practical problem like wind forecasting due to their fixed structure. In the literature, IT2FIS that have capability to handle varying data and evolve based on meta-cognition have been proposed [18, 8, 9, 10, 5]. Meta-cognition is known as the ability to make decision whether to learn a specific knowledge by using suitable learning strategies. Various studies on meta-cognitive algorithm with neuro-fuzzy inference system has clearly shown their generalization ability [27, 28, 10, 5, 6, 9, 4, 26, 21]. Although all the above mentioned algorithms provide good approximation by handling uncertainty in data and employing self-regulation, the forecasts are based on point forecasts, for which one predicted point is provided for one target value. Prediction intervals (PIs) [23, 22, 24], on the other hand, consists of a lower and an upper bound and the confidence level that the target value lies within the bounds. PIs are able to quantify the uncertainty associated with the prediction and give a qualitative forecasting information.

In this paper, a meta-cognitive sequential leaning algorithm for interval type-2 neuro fuzzy inference sysmem is proposed. The learning mechanism of the system is formulated on a five-layer network realizing Takagi-Sugeno-Kang fuzzy inference. The input layer passes data towards the membership layer to obtain Gaussian membership values. The firing layer calculates the strength of rules before employing interval reduction to approximate the output. The q factor referred in [1] together with upper and lower weights enable the construction of two bounds. In the output layer, recurrent neural network are employed by adding two memory neurons. The algorithm is able to construct prediction intervals and is indicated as recurrent neural network meta-cognitive interval type-2 fuzzy inference system, or RNN-McIT2FIS. Meta-cognitive learning controls the flow of RNN-McIT2FIS. The learning algorithm processes an input sample by deciding whether to delete it, learn it or reserve the sample based on its knowledge. These three decisions are $sample-delete-strategy$, $sample-learn-strategy$ and $sample-reserve-strategy$, respectively. The performance of proposed algorithm is measured based on a wind prediction problem. The results are compared with other algorithm such as support vector regression, simplified interval type-2 fuzzy neural network (SIT2FNN)[18], projection based learning meta-cognitive interval type-2 fuzzy inference (PBL-McIT2FIS)[6, 5] system and meta-cognitive interval type-2 fuzzy inference system gradient descent (McIT2FIS-GD)[8].

The rest of the paper is presented as follows. In section 2, the architecture of the interval type-2 fuzzy inference system is demonstrated. Section 3 is description of meta-cognitive learning mechanism for the IT2FIS. Performance of the proposed algorithm is evaluated on a wind forecasting problem in Section 4. Section 5 summarizes the key features of this study.

2. Interval type-2 fuzzy inference system structure

This section describes the structure of the proposed fuzzy neural network. The objective is to approximate functional relationship between input features x and output y. The structure is based on fuzzy rules. The premise part of a rule employs interval-type 2 fuzzy membership functions with uncertain means and fix standard deviations. The structure consists of five-layer network realizing Takagi-Sugeno-Kang fuzzy inference mechanism and is shown in Figure 1. Assume that the system learns m input features and has grown K rules. The detailed inference of each layer at a certain time t is presented as follows:

Layer 1- Input layer: This layer contains n nodes representing n number of input features. The input is passed directly to layer 2, Membership Function Layer. The output of *j-th* node is given:

$$u_j(t) = x_j(t);\ j = 1, 2, ..., n. \tag{1}$$

Layer 2- Fuzzification layer: This layer calculates the upper and the lower membership strength of each *j-th* feature in every *i-th* rule. The formulas are given by:

$$\mu_{ij}^{up}(x_j) = \begin{cases} \phi(m_{j1}^i, \sigma_j^i, x_j) & x_j < m_{j1}^i \\ 1 & m_{j1}^i \le x_j \le m_{j2}^i \\ \phi(m_{j2}^i, \sigma_j^i, x_j) & x_j > m_{j2}^i \end{cases} \tag{2}$$

$$\mu_{ij}^{lo}(x_j) = \begin{cases} \phi(m_{j2}^i, \sigma_j^i, x_j) & x_j \le \frac{m_{j1}^i + m_{j2}^i}{2} \\ \phi(m_{j1}^i, \sigma_j^i, x_j) & x_j > \frac{m_{j1}^i + m_{j2}^i}{2} \end{cases} \tag{3}$$

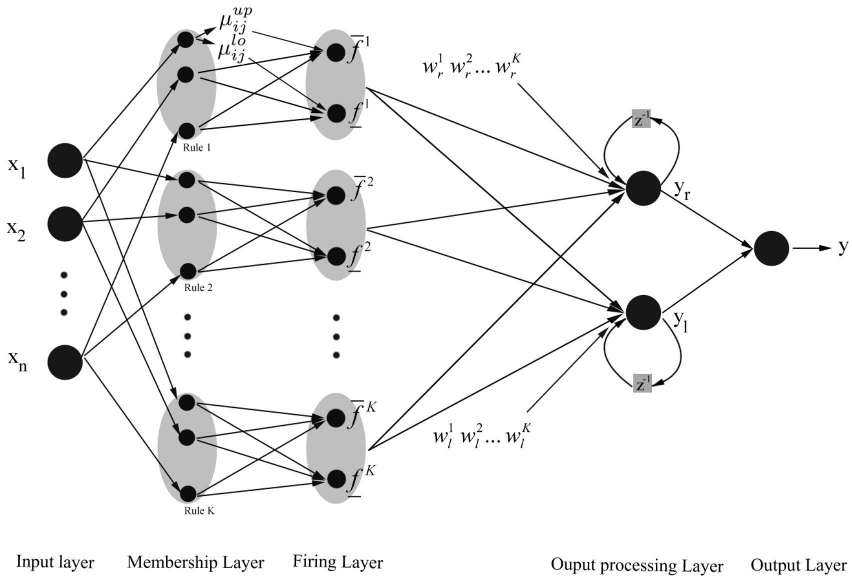

Fig. 1: Architecture of the RNN interval type-2 neural fuzzy system.

where,

$$\phi(m^i_j, \sigma^i_j, x_j) = exp(-\frac{(x_j - m^i_j)^2}{2(\sigma^i_j)^2}) \tag{4}$$

where, m_{j1}, m_{j2}, σ are center left, center right and width of the interval type-2 Gaussian membership function accordingly.

Layer 3- Firing layer: The firing strength of each rule is calculated in this layer based on the following formulas:

$$f^{lo}_i = \prod_{j=1}^{n} \mu^{lo}_{ij}; f^{up}_i = \prod_{j=1}^{n} \mu^{up}_{ij} \tag{5}$$

Layer 4- Output processing layer: This layer has K nodes, each of which represents a rule in the network. Instead of using Karnick-Mendel iterative procedure to find lower and upper end points, this study uses q factors to increase the learning speed and minimize the computational complexity. The two end points are given as:

$$y_l = \frac{(1 - q_l)\sum_{i=1}^{K} f^{up}_i w^i_l + q_l \sum_{i=1}^{K} f^{lo}_i w^i_l}{\sum_{i=1}^{K}(f^{lo}_i + f^{up}_i)} \tag{6}$$

$$y_r = \frac{(1 - q_r)\sum_{i=1}^{K} f^{lo}_i w^i_r + q_r \sum_{i=1}^{K} f^{up}_i w^i_r}{\sum_{i=1}^{K}(f^{lo}_i + f^{up}_i)} \tag{7}$$

Layer 5- Output layer: This layer employs memory neural network to process output bounds. Two recurrent neurons enable the algorithm to memorize the previous knowledge and generalize current output value. Upper and lower bounds at the time $t - th$ are computed as follows:

$$y_l(t) = \alpha y_l(t) + (1 - \alpha) y_l(t - 1) \tag{8}$$

$$y_r(t) = \alpha y_r(t) + (1 - \alpha) y_r(t - 1) \tag{9}$$

The crisp predicted output of the network is combination of y_l and y_r and is given as:

$$\hat{y} = \frac{1}{2}(y_l + y_r) \tag{10}$$

The above layers create a computationally fast inference and are able to evolve in the learning process. Next section describes the sequential learning algorithm for the proposed system.

3. Meta-cognitive learning for RNN-McIT2FIS

The leaning algorithm adapts the system by processing training samples only once. Each training sample is denoted as *(x, y)*, where $x = [x_1, x_2, ..., x_n] \in R^{1\times n}$ is *n-dimentional* input vector. The objective is to estimate the functional relationship *f[.]* between the input and the output ($x \rightarrow y$) so that the predicted output:

$$\hat{y} = f[x(t), \theta] \tag{11}$$

is an accurate approximation of the actual output. It is noted that vector θ represents the parameters of the rules. The error for *t-th* sample $e(t)$ is defined as the difference between the predicted output and the actual output.

$$e(t) = y(t) - \hat{y}(t) \tag{12}$$

In this study, meta-cogntive learning mechanism employs three indicators to train the network. The prediction error is the first indicator that monitors the difference between the predicted output and the actual output. Rule contribution are measured by spherical potential [28], which depends on the distance between the sample and existing rules in the network and indicates novelty of the current data. The third indicator is prediction interval. This indicator is calculated based on the relationship between two bounds and actual output. It describes how smooth the actual output falls into the interval. The formulas of three indicator are given as:

Prediction error *E(t)* for the sample *t-th* is:

$$E(t) = \sqrt{e^2(t)} \tag{13}$$

Spherical potential $\psi(t)$ is given as:

$$\psi(t) = \sum_{i=1}^{K} \frac{f_i^{lo}(t) + f_i^{up}(t)}{2K} \tag{14}$$

Prediction interval PI is calculated by:

$$PI = \frac{PI_{width}}{PI_{coverage}} \tag{15}$$

where, K is the total number of rules; PI_{width} is the width of the interval and is given as $PI_{width} = |y_l - y_r|$; $PI_{coverage}$ penalizes PI whose target does not lie within the constructed interval and is calculated as Gaussian function with the mean $\mu = (y_l + y_r)/2$ and variance $\sigma^2 = (y_l - y_r)/2$. The following subsections detail the learning algorithm of the system.

3.1. Sample delete strategy

The sample is deleted if its prediction error and prediction interval are smaller than the delete threshold, E_d and PI_d. E_d and PI_d are set close to zero. The criterion is given as:

$$E(t) < E_d \quad \text{AND} \quad PI < PI_d \tag{16}$$

3.2. Sample learn strategy

New rule is added into the system if knowledge in the new sample is novel to the contributing rules. In this case, a new rule is added into the system. The new sample could either be learnt to update parameters of existing rules. In this strategy, we define the rule growing criterion or rule updating criterion to check against the prediction error presented by the existing rules as well as the novelty of the sample.

3.2.1. The Rule Adding Criterion

$$E(t) > E_a \quad \text{AND} \quad PI > PI_a \quad \text{AND} \quad \psi(t) < E_S \tag{17}$$

where, E_a is the adding threshold and E_S is the novelty threshold. The two thresholds are self-adaptive and they are updated after a new rule is added into the network. γ_1 and γ_2 are the slopes at which E_a increases and E_S decreases in updating formulas below:

$$E_a := (1 - \gamma_1)E_a + \gamma_1 E(t) \tag{18}$$

$$E_S := (1 - \gamma_2)E_S - \gamma_2 \psi(t) \tag{19}$$

The new rule K+1 centers and the width are initialized as:

$$[m_{j1}^{K+1}, m_{j1}^{K+1},] = [x_j - 0.1, x_j + 0.1] \tag{20}$$

$$[\sigma^{K+1}] = min_{\forall j}[\| x(t) - m_{j1} \|, \| x(t) - m_{j2} \|] \tag{21}$$

It has been shown that the new output weight w^{K+1} should completely exploit the localization property of Gaussian membership function [29]. The new output weights $[w_l^i, w_r^i]$ are initialized so that the prediction error is minimal with the contribution of new rule. After adding a new rule, the network consisting of *K+1* rules gives the predicted output *$\hat{y}(t)$*, which should be equal to the actual output $y(t)$. The formula to initialize new weight is as follow:

$$w_{K+1} = \frac{y(t)(2K\psi(t) + 1 + f_{K+1}^{lo}) - \hat{y}(t)2K\psi(t)}{(1 - q_r + q_l)f_{K+1}^{lo} + 1 - q_l + q_r} \tag{22}$$

3.2.2. The Rule Updating Criterion

The rule update criterion is as follows:

$$E(t) > E_u \quad \text{AND} \quad PI > PI_u \quad \text{AND} \quad \psi(t) > E_S \tag{23}$$

When the update strategy is decided, the parameter update threshold E_u and novelty threshold E_S are self-adapted according to:

$$E_u := (1 - \gamma_1)E(t) + \gamma_1 E_u \tag{24}$$

$$E_S := (1 - \gamma_2)E_S + \gamma_2 \psi(t) \tag{25}$$

The parameter vector $\theta = [m_1, m_2, \sigma, w, q, \alpha]$ of the network is the *z-dimentional* vector of lower and upper rules center, rule width, output weights two q factors q_l qr. They are updated based on Extended Kalman Filter. The update formula is given as:

$$\theta = \theta + e(t)G^T. \tag{26}$$

where, *e(t)* is the error and G^T is the gain matrix.

The Kalman gain matrix is as follow:

$$G^T = P\partial\theta[R + \partial\theta^T P\partial\theta]^{-1} \tag{27}$$

where, P is the error covariance matrix of network parameter, $R = r_0 I$ is the variance of measurement noise and $\partial\theta$ is the gradient of predicted output with respect to network parameters. P is initialized as $P = p_0 I_{z\times z}$ and is updated as:

$$P = [I_{z\times z} - G\partial\theta^T]P + q_0 I_{z\times z} \tag{28}$$

where, p_0 is initialized error covariance and is set greater than 1; q_0 is step size control parameter and is set close to 0. The following equations are for calculating the gradient of predicted output with respect to network parameters.

The equations for updating θ are adopted from the study in [8] except for the updating equation of α. The equation is as follow:

$$\frac{\partial \hat{y}(t)}{\partial \alpha} = \frac{1}{2}(y_l(t) + y_r(t) - y_l(t-1) - y_r(t-1)) \tag{29}$$

3.3. Sample reserve strategy

The sample is reserved for training in the later stage if it does not satisfy all the conditions. As mentioned earlier in the sample update section, the samples are only learnt if the prediction error is above the update threshold and are discarded if it is smaller. The sample is reserved for later processing if its prediction error is not only small enough to be deleted but not big enough to update the rule in the system. These learning strategies are applied for every sample in the training process. The system is summarized in Algorithm 1.

4. Performance evaluation

In this section, the proposed algorithm is evaluated based on a real-world wind speed prediction problem. The performance of McIT2FIS is compared with other algorithm in the literature including Support vector regression (SVR), SIT2FNN [18], PBL-McIT2FIS [5] and McIT2FIS-GD [8]. The results except for RNN-McIT2FIS are adopted from the literature. To study the effectiveness of forecasting, prediction intervals for wind speed are then constructed. The performance of the algorithm is evaluated in Matlab R2013b environment on a Window system with Xeon CPU and 16GB RAM.

4.1. Performance measure:

Root mean square error (RMSE) is an accurate measure of the difference between the actual and the predicted output and formally popular in doing benchmark. RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{30}$$

The percentage of samples used in training process as an additional measure is given as:

$$PS = \frac{number\ of\ samples\ used\ for\ training}{total\ number\ of\ samples\ in\ training\ set} \tag{31}$$

4.2. Wind speed prediction problem

Wind is an important source of renewable energy. Due to the intermittent feature of wind in nature, the generated wind power rises and falls irregularly. This study first aims to to predict wind speed for better power management. The wind data is obtained from Iowa department of transportation (USA) over the period of February 1, 2011 to February 28, 2011. Wind speed and direction were sampled every 10 minutes. The data was then averaged every hour from which ten features are extracted. Training data consists of 500 samples and 100 samples are used for testing. The purpose is to predict one-step ahead wind speed $y(t+1)$ based on data from last five points of time. The root mean square error and number of rules for this problem are described in Table 1. It is observed that RNN-McIT2FIS is able to forecast the output with small error and reasonable number of rules. The percentage of samples used for training is 73%. Figure 2 and Figure 3 visualize the predicted and actual wind speed for training and testing data, respectively. It can be easily seen that RNN-McIT2FIS is able to well generalize the trend under the data.

4.3. Prediction intervals for wind speed forecasting

In this section, we shall construct prediction intervals for the above mentioned wind speed problem. In order to evaluate the performance of the prediction interval, coverage probability and average width are employed. Prediction intevals coverage probability (PICP), as used in [23] is defined as:

$$PICP = \frac{1}{N}\sum_{i}^{N} \epsilon_i \tag{32}$$

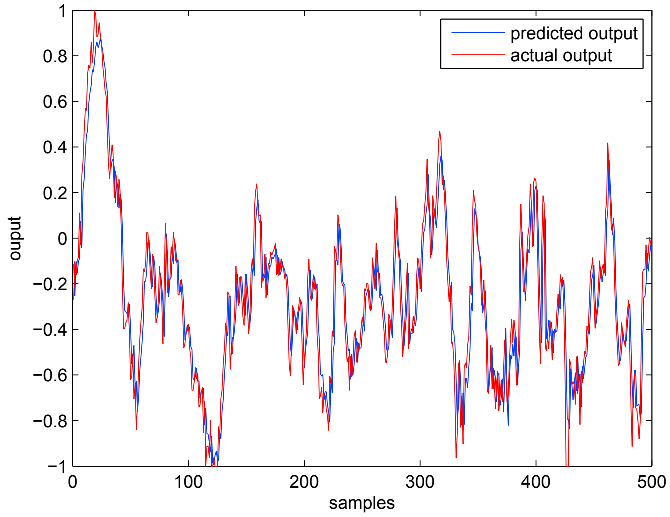

Fig. 2: Actual and predicted wind data for training set.

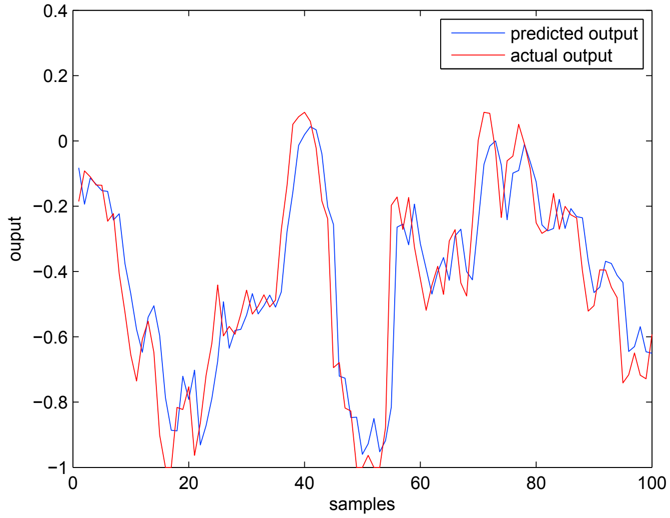

Fig. 3: Actual and predicted wind data for testing set.

where N is the number of samples and $\in_i$ show the coverage of PIs towards the $i-th$ sample. $\in_i$ is defined as:

$$\in_i = \begin{cases} 1, \ if \ y_i \in [L_i, \ U_i] \\ 0, \ if \ y_i \notin [L_i, \ U_i] \end{cases} \tag{33}$$

where L_i and U_i are lower bound and upper bound, respectively.
The average width of the intervals, PIAW [14] is given as:

$$PIAW = \frac{1}{N} \sum_{i=1}^{N} \frac{(U_i - L_i)}{R} \tag{34}$$

where R is the range of the underlying targets (maximum minus minimum). Figure 4 shows the constructed prediction intervals for testing data set. It is observed that majority of actual test data falls into the lower upper bound. For

Table 1: Performance comparison on wind speed prediction problem

Algorithm	Rules	Training RMSE	Testing RMSE
SVR	-	0.197	0.214
SIT2FNN	4	0.194	0.187
PBL-McIT2FIS	4	0.110	0.161
GD-McIT2FIS	4	0.136	0.132
RNN-McIT2FIS*	**4**	**0.1336**	**0.1381**

$^{*}PS = 73\%$

testing data set, the studied prediction intervals covers 92% of samples while average width is 34.5%. Table 2 shows PICP, PIAW for training and testing in this problem. From the table we can notice that RNN-McIT2FIS is able to give effective prediction intervals for wind speed prediction.

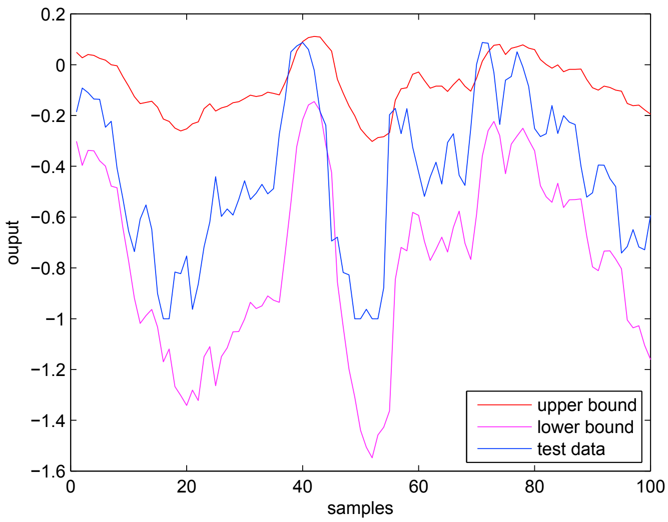

Fig. 4: Prediction intervals for wind speed.

Table 2: PI evaluation for training and testing

	Training	Testing
PICP	0.81	0.92
PIAW	0.24	0.35

$^{*}PS = 67.8\%$

5. Conclusion

This study proposes an interval type-2 fuzzy inference system with its meta-cognitive learning algorithm. The output layer of the system employs memory neurons and the learning mechanism considers prediction interval as a novel rule selection criterion. The algorithm utilizes a computationally fast interval-reduction approach and EKF-based parameter learning. Meta-cognitive decision on what-to-learn, how-to-learn and when-to learn is made by measuring knowledge in each sample. The sample is learnt if its knowledge is novel to the system and is deleted if it contains redundant knowledge. Learning strategies are based on the rule contribution, energy function and prediction interval criteria. Based on this tactics, the algorithm runs efficiently and is able to construct high quality prediction intervals. The performance is evaluated on real world wind speed prediction problem.

Acknowledgement

The present research was supported from EDB's Energy Innovation Research (EIRP) grant S14-1187-NRFEIPO-EIRP-IHL and the first author's PhD scholarship was generously supported by the IGS School of Nanyang Technological University.

References

[1] Abiyev, R.H., Kaynak, O., 2010. Type 2 fuzzy neural structure for identification and control of time-varying plants. IEEE Transactions on Industrial Electronics 57, 4147–4159.
[2] Abraham, A., 2005. Adaptation of Fuzzy Inference System Using Neural Learning. Springer Berlin Heidelberg, Berlin, Heidelberg.
[3] Angelov, P.P., Filev, D.P., 2004. An approach to online identification of takagi-sugeno fuzzy models. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34, 484–498.
[4] Babu, G.S., Suresh, S., 2012. Meta-cognitive neural network for classification problems in a sequential learning framework. Neurocomputing 81, 86 – 96.
[5] Babu, G.S., Suresh, S., 2013a. Meta-cognitive rbf network and its projection based learning algorithm for classification problems. Applied Soft Computing 13, 654 – 666.
[6] Babu, G.S., Suresh, S., 2013b. Meta-cognitive rbf network and its projection based learning algorithm for classification problems. Applied Soft Computing 13, 654 – 666.
[7] Castro, J.R., Castillo, O., Melin, P., Rodrguez-Daz, A., 2009. A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks. Information Sciences 179, 2175 – 2193. Special Section on High Order Fuzzy Sets.
[8] Das, A.K., Anh, N., Suresh, S., Srikanth, N., 2016. An interval type-2 fuzzy inference system and its meta-cognitive learning algorithm. Evolving Systems 7, 95–105.
[9] Das, A.K., Subramanian, K., Sundaram, S., 2015. An evolving interval type-2 neurofuzzy inference system and its metacognitive sequential learning algorithm. IEEE Transactions on Fuzzy Systems 23, 2080–2093.
[10] Das, A.K., Subramanian, K., Suresh, S., 2014. A computationally fast interval type-2 neuro-fuzzy inference system and its meta-cognitive projection based learning algorithm, in: 2014 International Joint Conference on Neural Networks (IJCNN), pp. 1510–1516.
[11] Gaxiola, F., Melin, P., Valdez, F., Castillo, O., 2014. Interval type-2 fuzzy weight adjustment for backpropagation neural networks with application in time series prediction. Information Sciences 260, 1 – 14.
[12] Jang, J.S.R., 1993. Anfis: adaptive-network-based fuzzy inference system. IEEEsransactions on Systems, Man, and Cybernetics 23, 665–685.
[13] Karnik, N.N., Mendel, J.M., Liang, Q., 1999. Type-2 fuzzy logic systems. IEEE Transactions on Fuzzy Systems 7, 643–658.
[14] Khosravi, A., Nahavandi, S., Creighton, D., 2011. Prediction interval construction and optimization for adaptive neurofuzzy inference systems. IEEE Transactions on Fuzzy Systems 19, 983–988.
[15] Kosko, B., 1992. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
[16] Lee, C.H., Lin, Y.C., Lai, W.Y., 2003. Systems identification using type-2 fuzzy neural network (type-2 fnn) systems, in: Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No.03EX694), pp. 1264–1269 vol.3.
[17] Liang, Q., Mendel, J.M., 2000. Interval type-2 fuzzy logic systems: theory and design. IEEE Transactions on Fuzzy Systems 8, 535–550.
[18] Lin, Y.Y., Liao, S.H., Chang, J.Y., Lin, C.T., 2014. Simplified interval type-2 fuzzy neural networks. IEEE Transactions on Neural Networks and Learning Systems 25, 959–969.
[19] Mendel, J.M., 2007. Type-2 fuzzy sets and systems: An overview [corrected reprint]. IEEE Computational Intelligence Magazine 2, 20–29.
[20] Park, K.J., Oh, S.K., Pedrycz, W., 2009. Design of interval type-2 fuzzy neural networks and their optimization using real-coded genetic algorithms, in: 2009 IEEE International Conference on Fuzzy Systems, pp. 2013–2018.
[21] Pratama, M., Lu, J., Anavatti, S., Lughofer, E., Lim, C.P., 2016. An incremental meta-cognitive-based scaffolding fuzzy neural network. Neurocomputing 171, 89 – 105.

[22] Quan, H., Srinivasan, D., Khosravi, A., 2012. Construction of neural network-based prediction intervals using particle swarm optimization, in: The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1–7.
[23] Quan, H., Srinivasan, D., Khosravi, A., 2014. Short-term load and wind power forecasting using neural network-based prediction intervals. IEEE Transactions on Neural Networks and Learning Systems 25, 303–315.
[24] Quan, H., Srinivasan, D., Khosravi, A., Nahavandi, S., Creighton, D., 2013. Construction of neural network-based prediction intervals for short-term electrical load forecasting, in: 2013 IEEE Computational Intelligence Applications in Smart Grid (CIASG), pp. 66–72.
[25] Rong, H.J., Sundararajan, N., Huang, G.B., Saratchandran, P., 2006. Sequential adaptive fuzzy inference system (safis) for nonlinear system identification and prediction. Fuzzy Sets and Systems 157, 1260 – 1275.
[26] Savitha, R., Suresh, S., Sundararajan, N., 2012. A meta-cognitive learning algorithm for a fully complex-valued relaxation network. Neural Networks 32, 209 – 218. Selected Papers from IJCNN 2011.
[27] Subramanian, K., Savitha, R., Suresh, S., 2014. A complex-valued neuro-fuzzy inference system and its learning mechanism. Neurocomputing 123, 110 – 120. Contains Special issue articles: Advances in Pattern Recognition Applications and Methods.
[28] Subramanian, K., Suresh, S., 2012. A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system. Applied Soft Computing 12, 3603 – 3614. doi:10.1016/j.asoc.2012.06.012.
[29] Subramanian, K., Suresh, S., Babu, R.V., 2012. Meta-cognitive neuro-fuzzy inference system for human emotion recognition, in: The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1–7.
[30] Wang, C.H., Cheng, C.S., Lee, T.T., 2004. Dynamical optimal training for interval type-2 fuzzy neural network (t2fnn). IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34, 1462–1477.