

Research Article

PiLiMoT: A Modified Combination of LoLiMoT and PLN Learning Algorithms for Local Linear Neurofuzzy Modeling

Atiye Sarabi-Jamab^{1,2} and Babak N. Araabi^{1,2}

¹ Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Tehran 1439957131, Iran

² School of Cognitive Sciences, Institute for Research in Fundamental Sciences (IPM), Tehran 1954851167, Iran

Correspondence should be addressed to Atiye Sarabi-Jamab, asarabi@ut.ac.ir

Received 15 September 2010; Accepted 16 January 2011

Academic Editor: G. Yin

Copyright © 2011 A. Sarabi-Jamab and B. N. Araabi. 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.

Locally linear model tree (LoLiMoT) and piecewise linear network (PLN) learning algorithms are two approaches in local linear neurofuzzy modeling. While both methods belong to the class of growing tree learning algorithms, they use different logics. PLN learning relies on training data, it needs rich training data set and no division test, so it is much faster than LoLiMoT, but it may create adjacent neurons that may lead to singularity in regression matrix. On the other hand, LoLiMoT almost always leads to acceptable output error, but it often needs more rules. In this paper, to exploit the complimentary performance of both algorithms piecewise linear model tree (PiLiMoT) learning algorithm is introduced. In essence, PiLiMoT is a combination of LoLiMoT and PLN learning. The initially proposed algorithm is improved by adding the ability to merge previously divided local linear models, and utilizing a simulated annealing stochastic decision process to select a local model for splitting. Comparing to LoLiMoT and PLN learning, our proposed improved learning algorithm shows the ability to construct models with less number of rules at comparable modeling errors. Algorithms are compared through a case study of nonlinear function approximation. Obtained results demonstrate the advantages of combined modified method.

1. Introduction

System modeling plays an important role in many areas such as control, expert systems, communication, and so forth. Most of the control structures used in industry is designed on the basis of the classical control theory, which is well suited for linear processes control whose exact model is known. However, the majority of physical systems contain complex nonlinear relations, which are difficult to model [1].

Consequently, a nonlinear system function is needed and nonlinear neurofuzzy networks may be used to do this job [2]. Neurofuzzy approach, in contrast to pure neural or fuzzy methods, possesses both of their advantages: it brings the low-level learning and computational power of neural networks into fuzzy systems and provides the high-level human-like thinking in reasoning of fuzzy systems into neural networks [3]. In general, this approach involves two major phases, structure identification and parameter estimation [4].

For parameter identification, most systems, including [1, 5], use backpropagation to refine parameter of the systems. However, backpropagation suffers from the problems of local minima and low convergence rate [5]. To alleviate these difficulties, different methods of least squares estimation [6] have been proposed. In [7], combinations of two well-known identification methods namely series-parallel and parallel configurations, are proposed to identify the learning parameters of neurofuzzy inference system. For structure identification, Latini et al. [8] proposed a method that by means of a learning algorithm, it is possible to define, within the input space, a set of regions of “vague” or “noncertain” classification, each one associated to a fuzzy rule. Jakubek and Keuth [9] proposed a method that the validity function of each local model is fitted to the available data using statistical criteria along with regularization and thus allowing an arbitrary orientation and extent in the input space. In [6], the region of validity of each local model is adaptively optimized using the Chi-squared distribution of

the estimated residual. The proposed method of Cakmakci [10] concerns the simultaneous optimization for structure and parameters of fuzzy inference systems that is based on hierarchical fair competition-based parallel genetic Algorithms (HFCGA) and information data granulation. HFCGA is used to optimize structure and parameters of ANFIS-based fuzzy model simultaneously. The granulation is realized with the aid of the C-means clustering.

The object of structure identification is to identify an optimal partition of the input space into fuzzy set. The important task in the structure identification of a neural fuzzy network is partitioning the input space, which influences the numbers of fuzzy rule generated. The most direct way is partitioning the input space into grid type or clustering the input training vectors in the input space [11–13]. LoLiMoT and PLN learning algorithms are two approaches in structure optimization based on local linear modeling, which use different algorithms in their training phase.

The locally linear model tree (LoLiMoT) algorithm is applied to train the extended radial bases function (RBF) network [14, 15]. Local loss function for all regions is computed and the worst region is chosen to be split into two new regions. Divisions in all axis orthogonal dimensions are tested and the best division is selected. The new center is found and the center of worst region is changed.

Piecewise linear network (PLN) learning algorithm is a general neural network with three layers based on MLP network [16]. It was designed for fast function approximation with a good generalization capability even in the case of very few data points.

Because of some drawbacks in LoLiMoT and PLN, in this paper, a new learning algorithm is introduced for nonlinear approximation. This method is a modified combination of these two main approaches in local linear modeling. It takes suitable error from LoLiMoT and suitable number of neurons from PLN that leads to efficient network which is applicable for function approximation. Some initial results of this research have been presented in [17, 18].

The rest of paper is organized as follows: LoLiMoT and PLN learning algorithms and simulated annealing are briefly reviewed in Section 2. In Section 3 a modified combination of LoLiMoT and PLN learning algorithms is suggested to improve their structure identification performance. In Section 4, performances of the original and modified algorithms are compared in a function approximation case study. The paper is concluded with final discussions and concluding remarks in Section 5.

2. Background

This section explains the main characteristics of locally linear model tree, piecewise linear network and simulated annealing algorithms.

2.1. Locally Linear Model Tree (LoLiMoT). The local linear model tree algorithm proposed by Nelles and Isermann [5, 19, 20], is based on the idea to approximate a nonlinear function with piecewise linear models [5].

The LoLiMoT algorithm partitions the input space in hyperrectangles by axis-orthogonal splits. In each iteration of the algorithm, a new rule or locally linear model is added to the model. Thus, LoLiMoT belongs to the class of incremental or growing algorithms. It implements a heuristic search for the rule premise structure and avoids a time-consuming nonlinear optimization. The validity functions which correspond to the actual partitioning of the input space are compared and the corresponding rule consequents are optimized by the local weighted least square technique [5, 19, 21].

The LoLiMoT algorithm is as follows.

Start with an Initial Model. Construct the validity functions for the initially given input space partitioning and estimates the LLM parameters by the local weighted LS algorithm.

Find Worst LLM. Calculates a local loss function for each of the LLMs and finds its maximum as the worst LLM.

Check All Divisions. The worst LLM is selected for further refinement. The hyper-rectangle of this LLM is split into two halves with an axis-orthogonal split. Division in all dimensions is tried. For each p , dimensions validity functions and LLM parameter are constructed and local function is computed for them.

Find Best Division. The best p alternates according to constructed in step 3 and the optimized LLMs are adapted for the model.

Test for Convergence. If the termination criterion is met, then stop, else go to step 2 [5, 19].

2.2. Piecewise Linear Network (PLN). A piecewise linear network (PLN) divides the input space into several linear regions. It is a general neural network with three layers that provides a link between feed forward networks (multilayer perceptrons) and classical system modeling. The hidden neurons of the PLN represent the regions. Every hidden neuron evaluates the distance from the input vector to a center point represented by the weights vector of the hidden neuron. The linear output neurons receive their inputs both from the hidden layer and the input layer [2, 22, 23]. An outline of the algorithm is as follows.

- (1) Start with an Initial Model. Construct the validity functions for the initially given input space partitioned by the normalized Euclidean distance from the input pattern to all of the region center patterns, and estimate the LLM parameters.
- (2) Find Maximum Error. Calculate squared error in each data point and find worst data point. Note that the error can be defined to be absolute or relative error.
- (3) Add New Neuron and Find LLM Parameters. Generate a new hidden neuron with its center at the sample

with maximum error and rearrange the partitioning of input space and make a least square fit for every region.

- (4) Test for Convergence. If the termination criterion is met then stop, else go to step 2 [22].

2.3. Simulated Annealing (SA). SA is a stochastic model for global optimization. A warm particle is simulated in a potential field. Generally, the particle moves down toward lower potential energy, but since it has a nonzero temperature, that is, kinetic energy, it moves around with some randomness and therefore occasionally jumps to higher potential energy. Thus, the particle is capable of escaping local minima and possibly finding a global one. In the context of optimization, the particle represents the parameter point in search space and the potential energy represents the loss function. The general form of a simulated annealing algorithm is discussed in [24].

In order to enable the escape from local minima, SA accepts the new point with a probability that has the following properties: (i) acceptance is more likely for better points than for worse ones and (ii) acceptance of worse points has a higher probability for larger temperatures than for smaller ones. The acceptance probability in standard SA is chosen as [24]

$$h(\Delta I_k, T_k) = \frac{\exp(-I_k/T_k)}{\exp(-I_k/T_k) + \exp(-I_{k-1}/T_k)} \quad (1)$$

$$= \frac{1}{1 + \exp(\Delta I_k/T_k)},$$

where ΔI_k represents the difference of the loss function value between the new and the old parameters.

3. Proposed Algorithm

In the sequel, we first provide some motivation on how the proposed algorithm improves the performance of existing algorithms. We then present PiLiMoT formally.

3.1. A Combination of LoLiMoT and PLN. Locally linear model tree (LoLiMoT) and piecewise linear network (PLN) learning algorithms are two approaches in local linear modeling for structure optimization. The main difference between them is their new region finding strategy. This difference causes many advantages and disadvantages. PLN learning is more dependent on training data than LoLiMoT is; therefore, PLN needs very suitable and fair training data sets to avoid over training phenomenon. In PLN learning, no division test is needed which results in a much faster learning algorithm than LoLiMoT. But, suggested validity function in PLN is not efficient when two or more neurons are close to each other. In this case, the new sample lies in its linear region with its own linear function. The same activity function leads to singularity in regression matrix and the algorithm will be failed. In LoLiMoT, because of the regular splitting of input space, this problem does not occur and always it

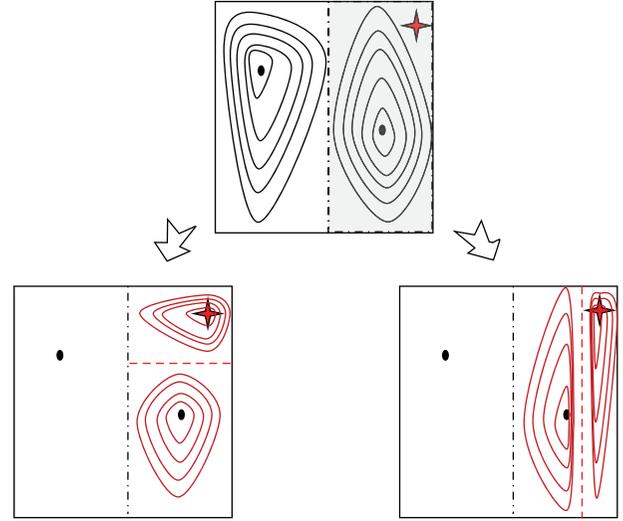


FIGURE 1: Finding the worst data point in the selected region as the new neuron center in PiLiMoT.

leads to acceptable output error, but needs large number of neurons.

In this paper, piecewise linear model tree (PiLiMoT) is proposed as a new modified combination of two main algorithms. This algorithm chooses the worst region in the sense of modified loss function and sets the worst data point in selected region as the new neuron center while the center of selected region does not change. All divisions in axis-orthogonal dimension are tried, and the best division is selected. Validity functions in this method are nonsymmetrical normalized Gaussian function with contour as combination of four quarters of distance ellipses. Figure 1 shows new region finding strategy in PiLiMoT algorithm.

In this new algorithm, selecting the worst region for partitioning and finding the best division is based on LoLiMoT algorithm. But generating a new hidden neuron with its center is based on PLN learning algorithm.

3.2. Self-Construction Ability in Combined Algorithm. PiLiMoT belongs to the class of heuristic construction algorithms because it adds a new rule in each iteration of the algorithm. Problem of the incremental, constructive learning algorithm for this new modified method is that generation of many LLMs may become superfluous. This problem arises because the algorithm is working locally and at the beginning it does not know whether further data for a special region is presented or not. After the training is finished, all is presented and then a more globally working procedure such as pruning may be applied. Pruning tries to reduce the neural network size to get better performance and better generalization [25]. In some cases, with only few training data, even a better approximation with smaller error may be achieved. If pruning is well done, small region with significantly big difference to neighbor regions may be removed [26].

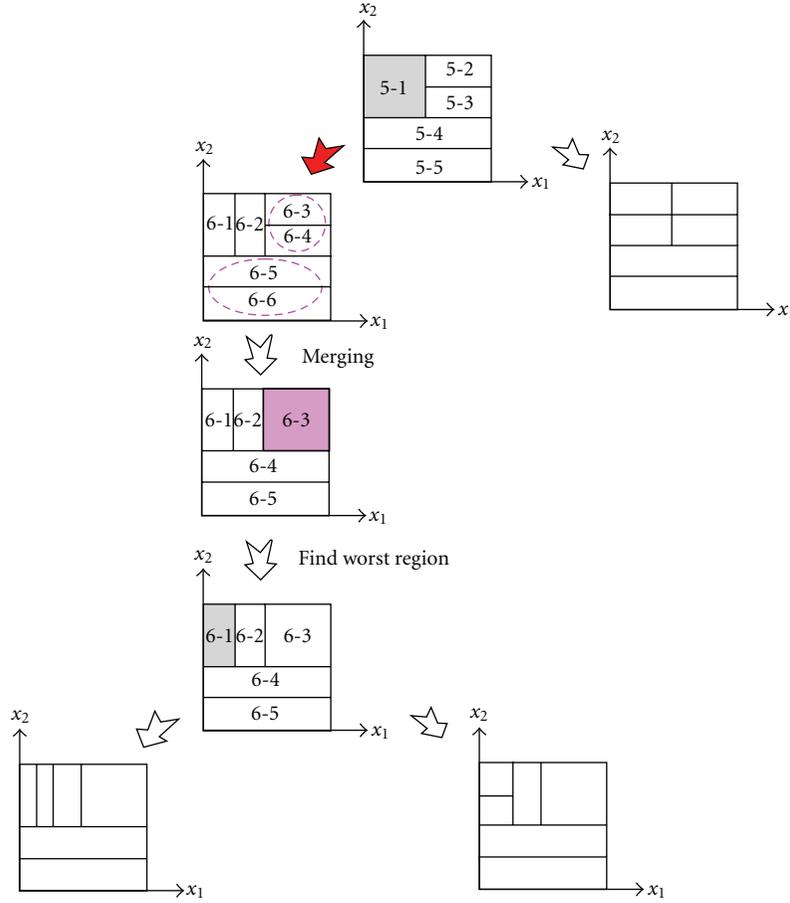


FIGURE 2: Self-construction ability in PiLiMoT algorithm with merging in 6th iteration.

By extending PiLiMoT with a pruning strategy which is able to merge formerly divided local linear models, this drawback can be remedied [3, 5]. The complicated model can be reconstructed to a simple model with the same or better accuracy. Consequently, the new algorithm is modified with self-construction through merging and splitting; Figure 2 shows this ability in 6th iteration in PiLiMoT algorithm.

3.3. *Piecewise Linear Model Tree (PiLiMoT)*. The PiLiMoT algorithm is formally presented as follows.

- (1) Start with an Initial Model. Construct the validity functions for the initially given input space partitioning and estimates the local linear model (LLM) parameters by the local weighted LS algorithm. If no input space partitioning is available a priori, then it starts with a single LLM ($\phi_1(\mathbf{u}) = 1$) as the validity function

$$\mathbf{w}_i = (\mathbf{X}^T \mathbf{Q}_i \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Q}_i \mathbf{y}, \quad (2)$$

where

$$\mathbf{X} = \begin{bmatrix} 1 & u_1 & (1) & \cdots & u_p & (1) \\ 1 & u_1 & (2) & \cdots & u_p & (2) \\ \vdots & \vdots & & \ddots & \vdots & \\ 1 & u_1 & (N) & \cdots & u_p & (N) \end{bmatrix}, \quad (3)$$

$$\mathbf{Q}_i = \text{diag}(\phi_i(\mathbf{u}(1)), \phi_i(\mathbf{u}(2)), \dots, \phi_i(\mathbf{u}(N))).$$

- (2) Find the Worst Region in the Sense of Modified Local Loss Function. Calculate a local loss function for each of the LLMs and find its maximum as the worst LLM. For calculating each local loss function all points are considered that belong to that region

$$J_i = \sum_{t=1}^N e^2(t) \quad \text{if } \mathbf{u}(t) \in \text{ith region}. \quad (4)$$

- (3) Find the Worst Data Point in the Selected Region in the Sense of Modified Output Error.

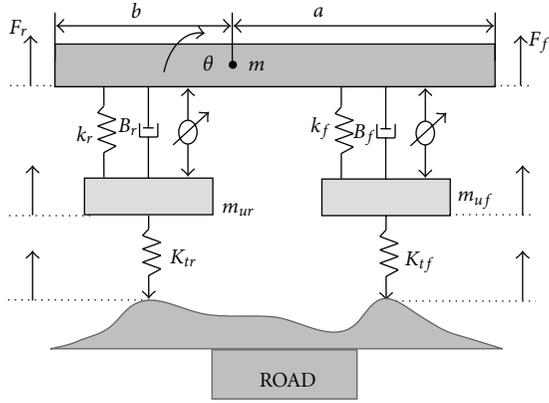


FIGURE 3: Half-car suspension model.

- (4) Check All Divisions. Divisions in all dimensions are tried. For each p dimension, validity functions of two recently generated regions and LLM parameters are constructed and local function is computed for them.
- (5) Find Best Division. The best of the p alternates and the optimized LLMs are adapted for the model.
- (6) Perform the Merging Algorithm by the Following Steps If It Yields an Improvement Comparing with the Same Number of Rules.
 - (i) Find all LLMs that can be merged.
 - (ii) Perform the following stage, for all these LLMs:
 - (a) construction of the multidimensional MSF in the merged hyper-rectangle,
 - (b) construction of all validity functions,
 - (c) local estimation of the rule consequent parameters for the merged LLM, and
 - (d) calculation of loss function for the current overall model.
 - (iii) Find the best merging possibility.
- (7) Test for Convergence. If the termination criterion is met then stop, else go to step 2.

4. Case Study:

Half-Car Active Suspension Model

A half-car active suspension system is considered here as a test bed to examine and compare the performance of proposed PiLiMoT algorithm with that of LoLiMoT and PLN algorithms. We start with a mathematical model (a set of equations) for half-car active suspension system. The model is used to generate I/O data. The I/O data is then utilized in training and evaluation of proposed and existing neurofuzzy models as system identification tools.

This model can be described as a nonlinear four degree-of-freedom system in Figure 3, including heave, pitch, and motion of the front and rear wheel. The mass of the vehicle

body is m and the unsprung mass on the front and rear tires are respectively m_{uf} and m_{ur} ; moment inertia of the sprung mass is $J_y \cdot z_{rf}$, and z_{rr} denote the road excitation on front and rear tires, respectively [27]. According to Figure 3, we can have the following half-car active suspension model

$$\dot{x}_1 = x_2,$$

$$\dot{x}_2 = \frac{1}{m_s} \left[-(B_f + B_r)x_2 + (L_f B_f - L_r B_r)x_4 \cos x_3 - k_f x_5 + B_f x_6 - k_r + B_r x_8 + (u_1 + u_2) \right] - g,$$

$$\dot{x}_3 = x_4,$$

$$\dot{x}_4 = \frac{1}{J_y} \left[(L_f B_f - L_r B_r)x_2 \cos x_3 - (L_f^2 B_f + L_r^2 B_r)x_4 \cos^2 x_3 + (L_f K_f x_5 - L_f B_f x_6 - L_r K_r x_7 + L_r B_r x_8) \times \cos x_3 + (-L_f u_1 + L_r u_2) \cos x_3 \right],$$

$$\dot{x}_5 = x_2 - L_f x_4 \cos x_3 - x_6,$$

$$\dot{x}_6 = \frac{1}{m_{uf}} \left[-K_{tf} x_1 + B_f x_2 + L_f K_{tf} \sin x_3 - L_f B_f x_4 \cos x_3 + (k_f + K_{tf})x_5 - B_f x_6 + K_{tf} z_{rf} - u_1 \right] - g,$$

$$\dot{x}_7 = x_2 + L_r x_4 \cos x_3 - x_8,$$

$$\dot{x}_8 = \frac{1}{m_{ur}} \left[-K_{tr} x_1 + B_r x_2 - L_r K_{tr} \sin x_3 + L_r B_r x_4 \cos x_3 + (k_r + K_{tr})x_7 - B_r x_8 + K_{tr} z_{rr} - u_2 \right] - g, \quad (5)$$

where k_f and k_r are the front and the rear spring coefficients, B_f and B_r are the front and the rear damping coefficients, and u_1 and u_2 are force inputs.

From these equations, 500 uniform distributed random training data pairs were obtained. We chose 450 training data and 50 checking data pairs. The mean of squared error (MSE) is used as a performance index. For the termination criterion, it is desired to reach to 0.025 for MSE. The convergence curves in Figure 4 compare three learning procedures for original LoLiMoT and PLN with new combined learning algorithm, (PiLiMoT). In new combined learning algorithm a comparable model error with 7 rules (LLM) can be achieved when standard LoLiMoT algorithm constructs a model with 9 rules and PLN algorithm is failed after four times. So, the performance in new combined learning algorithm improves significantly. Consequently, less output error and less neurons number are obtained by this new combined algorithm.

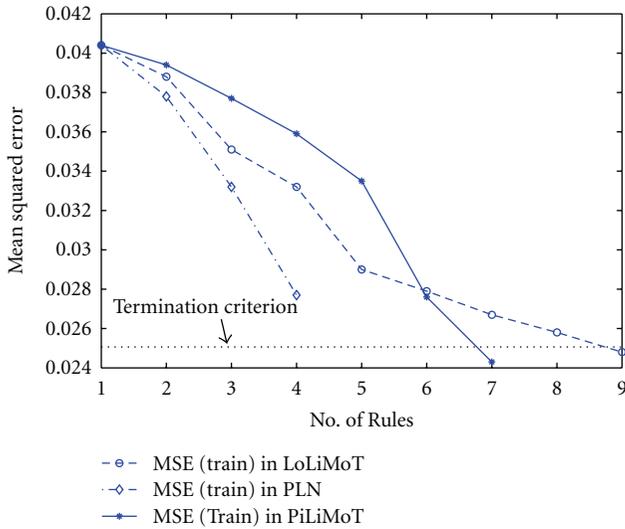


FIGURE 4: Convergence curves in comparing original LoLiMoT and original PLN with new combined learning algorithm in training data. In first steps, new combined learning algorithm has worse performance than original LoLiMoT because of the irregular choice of new center for new local linear model, but after some steps it has better performance.

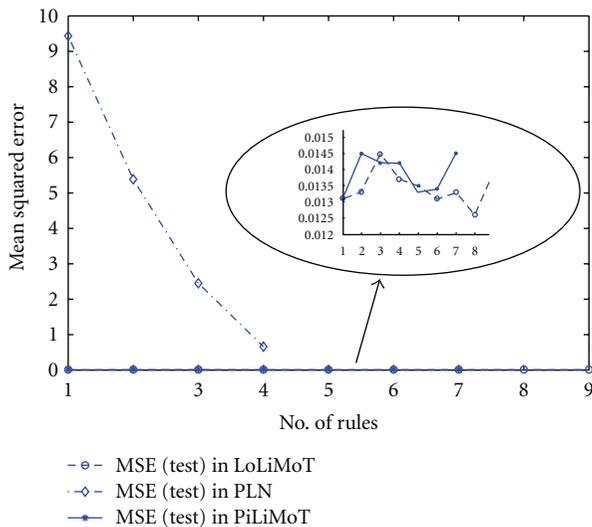


FIGURE 5: Convergence curves in comparing original LoLiMoT and original PLN with PiLiMoT learning algorithm in testing data.

Figure 5 compares three learning algorithm, original LoLiMoT, PLN, and new combined learning algorithm (PiLiMoT), in testing data, and Figure 6 shows the neuron centers in learning algorithms. This figure shows that original LoLiMoT in two regions (2, 8) contain no new algorithm neuron and centers change in other regions. As described before, suggested validity function in PLN is not efficient when two or more neurons are close to each other (center 4 lies near center 3). In this case, the new sample lies in its linear region with its own linear function and the same

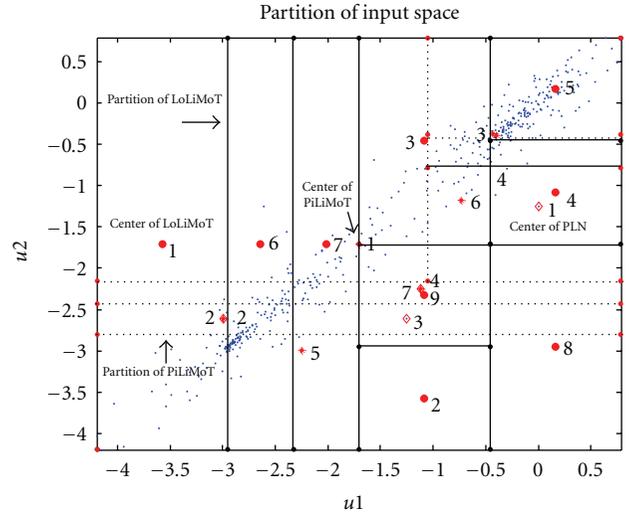


FIGURE 6: Neurons centers for original LoLiMoT, PLN, and new combined learning algorithm.

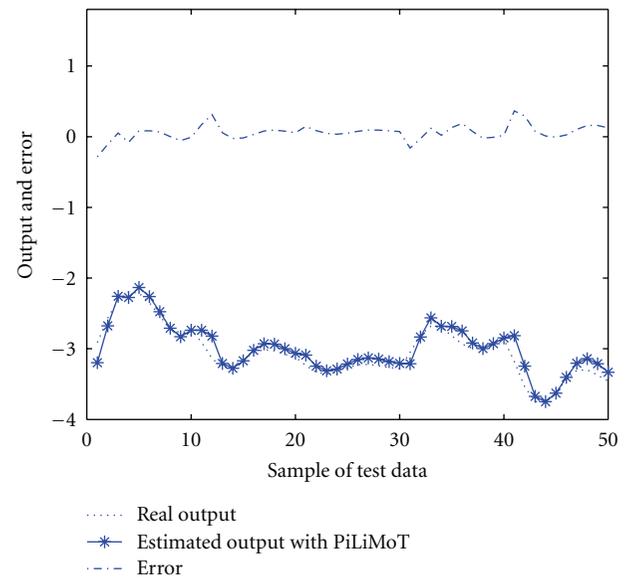


FIGURE 7: Real and estimated output in new combined algorithm.

activity function leads to singularity in regression matrix and the algorithm will be failed after four times.

Figure 7 compares real output and estimated output for new combined learning algorithms.

As described before, this new algorithm improved in two ways: (1) the ability to merge previously divided local linear models is added, and (2) a simulated annealing stochastic decision process is responsible to select a local model for splitting. Figures 8 and 9 compare the performance of them in training and testing data, respectively.

TABLE 1: The comparison of PiLiMoT with LoLiMoT and PLN (in same number of rules or same MSE).

Model	No. of rules (LLM)	MSE (train)	MSE (test)
LoLiMoT	9	0.0248	0.0140
PLN	4	0.0277	0.6612
PLN	5	failed	failed
PiLiMoT	7	0.0243	0.0145
PiLiMoT with Merge ability	6	0.0246	0.0145
PiLiMoT with Merge and Split by SA	5	0.0245	0.0144

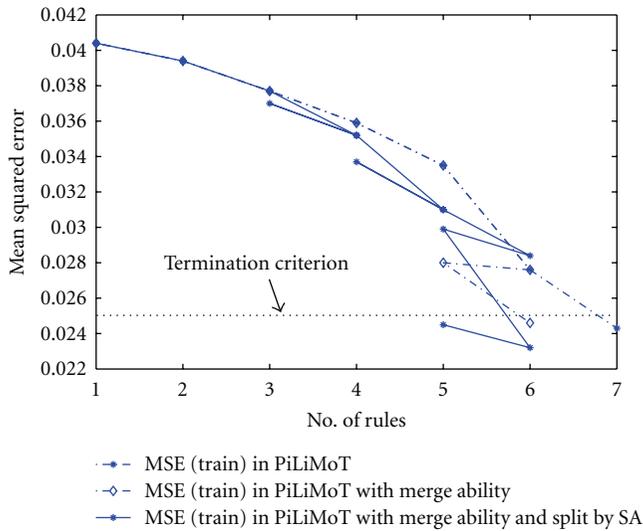


FIGURE 8: Convergence curves in comparing PiLiMoT, PiLiMoT with Merge ability, and PiLiMoT with Merge ability and modified split by SA in training data.

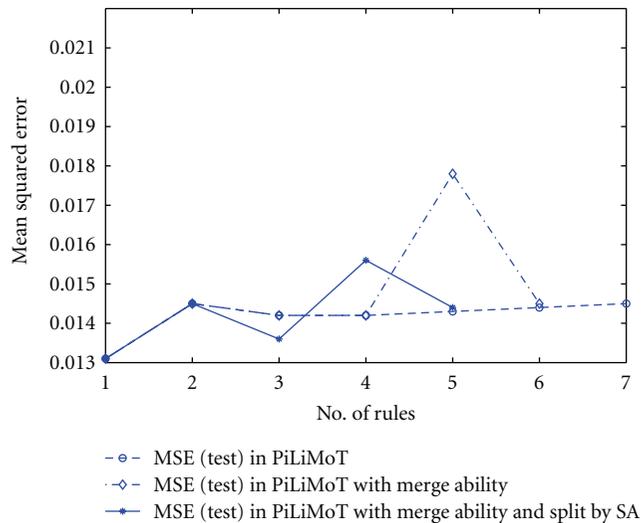


FIGURE 9: Convergence curves in comparing PiLiMoT, PiLiMoT with Merge ability, and PiLiMoT with Merge ability and modified split by SA in test data.

Note that in Figure 8 in PiLiMoT desired model error with 7 rules (LLM) can be achieved when modified PiLiMoT with merge ability algorithm constructs a model with 6 rules

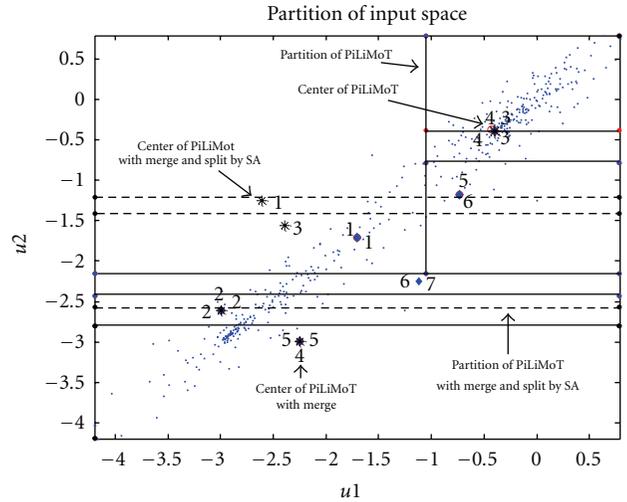


FIGURE 10: Neurons centers for PiLiMoT, PiLiMoT with Merge ability, and PiLiMoT with Merge ability, and modified split by SA.

and modified PiLiMoT with merge ability and split by SA algorithm construct a model with 5 rules.

Figure 10 shows the neuron centers that in PiLiMoT both regions 3 and 4 contain one new modified algorithm neuron and region 1 contains two new modified learning algorithm neurons.

The pruning strategy allows one to merge LLMs (3 and 4 regions in PiLiMoT) where they are not required, and thus more LLMs are available in the important region. In addition, the PiLiMoT with merge ability and modified split by SA is able to backtrack in case of suboptimal intermediate split decision.

In Table 1, our proposed methods has been compared with original LoLiMoT and original PLN identifying the same data. It can be seen from Table 1 that the performance of our modified learning algorithms is superior to original LoLiMoT and original PLN.

5. Conclusions

The LoLiMoT and PLN learning algorithm for structure identification in locally linear neurofuzzy models were combined to exploit the complimentary features of the two methods. The combined algorithm resulted in a low output error, which was inherited from LoLiMoT, and a small

number of neurons, which was inherited from PLN learning. The proposed algorithm was more improved by adding the ability to merge previously split local models using a pruning strategy. Pruning objective was not only aimed to reduce the number of neurons, but also intended to improve the generalization ability of the model. A simulated annealing decision process was responsible for selecting a local model for splitting. The final proposed algorithm, PiLiMoT, was able to alleviate some suboptimal intermediate split decisions as well. In PiLiMoT, it was possible to reverse a previously made decision, if it did not seem to be a good decision anymore. An extensive case study on half-car active suspension modeling demonstrated the superior performance of the proposed method as compared with that of its ancestors, where a comparable error can be achieved with less number of rules. The proposed algorithm in the current form can hardly be used in online modeling applications. Currently, we are working on an extension of PiLiMoT for online applications, which also relaxes the axis orthogonal decomposition of input space.

Acknowledgment

Authors would like to thank Mr. Aras Adhami Mirhosseini, who was part of the team at the early stages of this research.

References

- [1] C. F. Juang and C. T. Lin, "An on-line self-constructing neural fuzzy inference network and its applications," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 1, pp. 12–32, 1998.
- [2] W. Eppler and H. N. Beck, "Piecewise linear networks (PLN) for function approximation," in *Proceedings of International Joint Conference on Neural Networks (IJCNN '99)*, pp. 388–391, July 1999.
- [3] D. Kukolj and E. Levi, "Identification of complex systems based on neural and Takagi-Sugeno fuzzy model," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 34, no. 1, pp. 272–282, 2004.
- [4] S. J. Lee and C. S. Ouyang, "A neuro-fuzzy system modeling with self-constructing rule generation and hybrid SVD-based learning," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 3, pp. 341–353, 2003.
- [5] O. Nelles, *Nonlinear System Identification*, Springer, Berlin, Germany, 2000.
- [6] M. Kozek and S. Sinanovic, "Identification of wiener models using optimal local linear models," in *Proceedings of EUROSI*, September 2007.
- [7] A. Banakar and M. F. Azeem, "Parameter estimation of neuro-fuzzy model by parallel and series-parallel identification configurations," in *Proceedings of IEEE International Conference on Fuzzy Systems*, July 2007.
- [8] G. Latini, G. R. Cocci, G. Passerini, and S. Tascini, *A Neuro-Fuzzy Model for Ozone Forecasting*, Transaction of the Wessex Institute, 2007.
- [9] S. Jakubek and N. Keuth, "A local neuro-fuzzy network for high-dimensional models and optimization," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 6, pp. 705–717, 2006.
- [10] M. Cakmakci, "Adaptive neuro-fuzzy modelling of anaerobic digestion of primary sedimentation sludge," *Bioprocess and Biosystems Engineering*, vol. 30, no. 5, pp. 349–357, 2007.
- [11] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition—part I," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 29, no. 6, pp. 778–785, 1999.
- [12] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition—part II," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 29, no. 6, pp. 786–801, 1999.
- [13] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.
- [14] C. Halfmann, O. Nelles, and H. Holzmann, "Modeling and identification of the vehicle suspension characteristics using local linear model trees," in *Proceedings of IEEE Conference on Control Applications*, vol. 2, pp. 1484–1489, Coast-Island, Hawaii, USA, August 1999.
- [15] H. Holzmann, O. Nelles, C. Halfmann, and R. Isermann, "Vehicle dynamics simulation based on hybrid modeling," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '99)*, pp. 1014–1019, Atlanta, Ga, USA, September 1999.
- [16] J. W. Park, R. G. Harley, and G. K. Venayagamoorthy, "Comparison of MLP and RBF neural networks using deviation signals for on-line identification of a synchronous generator," in *Proceedings of IEEE Power Engineering Society Winter Meeting*, vol. 1, pp. 274–279, January 2002.
- [17] A. S. Jamab and B. N. Araabi, "A learning algorithm for local linear neuro-fuzzy models with self-construction through merge & split," in *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems (CIS&RAM '06)*, Bangkok, Thailand, June 2006.
- [18] A. S. Jamab and B. N. Araabi, "Piecewise linear model tree: a modified combination of two learning algorithms for neuro-fuzzy model," in *Proceedings of the IEEE International Conference on Control Applications (CCA '06)*, pp. 2155–2159, Munich, Germany, October 2006.
- [19] O. Nelles and R. Isermann, "Basis function networks for interpolation of local linear models," in *Proceedings of the 35th IEEE Conference on Decision and Control (CDC '96)*, Kobe, Japan, December 1996.
- [20] O. Nelles, S. Sineel, and R. Isermann, "Local basis function networks for identification of a turbocharger," in *Proceedings of UKACC International Conference on Control (CONTROL '96)*, September 1996.
- [21] M. Hafner, M. Schueler, and O. Nelles, "Dynamical identification and control of combustion engine exhaust," in *Proceedings of the American Control Conference*, vol. 1, pp. 222–226, San Diego, Calif, USA, June 1999.
- [22] W. Eppler, H. N. Beck, and H. Gemmeke, "Piecewise linear network (PLN) for approximation tasks," in *Proceedings of International Workshop on Soft Computing in Industry*, Muroran, Japan, June 1999.
- [23] W. Eppler and T. Böttger, "Optimization of piecewise linear network (PLN) by pruning," in *Proceedings of the 9th IFSA World Congress and 20th NAFIPS International Conference*, Vancouver, Canada, July 2005.
- [24] L. Ingber, "Simulated annealing: practice versus theory," *Mathematical and Computer Modelling*, vol. 18, no. 11, pp. 29–57, 1993.

- [25] R. Reed, "Pruning algorithms—a survey," *IEEE Transactions on Neural Networks*, vol. 4, no. 5, pp. 740–747, 1993.
- [26] F. Hoffmann and O. Nelles, "Genetic programming for model selection of TSK-fuzzy systems," *Information Sciences*, vol. 136, no. 1–4, pp. 7–28, 2001.
- [27] S. J. Wu, C. T. Wu, and T. T. Lee, "Neural-network-based optimal fuzzy control design for half-car active suspension systems," in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 376–381, June 2005.



Hindawi

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

