An Extended ANFIS Architecture and its Learning Properties for Type-1 and Interval Type-2 Models

Chao Chen*, Robert John†, Jamie Twycross‡, Jonathan M. Garibaldi§
School of Computer Science
University of Nottingham
Email: {*psxcc4, †robert.john, ‡jamie.twycross, §jon.garibaldi}@nottingham.ac.uk

***Abstract*—In this paper, an extended ANFIS architecture is proposed. By incorporating an extra layer for the fuzzification process, the extended architecture is able to fit both type-1 and interval type-2 models. The learning properties of the proposed architecture based on the least-squares estimate method are studied on selected type-1 and interval type-2 ANFIS models. We show that the least-squares estimate method in general behaves differently for interval type-2 ANFIS models compared to type-1 ANFIS models, producing larger errors for interval type-2 ANFIS.**

I. Introduction

A common concern for constructing a fuzzy inference system (FIS) is that there are no standard methods for transforming human knowledge or experience into the rule base. Also, effective methods are required to tune the membership functions for optimising the performance of FISs. The architecture of *Adaptive-Network-based Fuzzy Inference System* (ANFIS) was proposed by Jang [1] to serve as a basis for constructing a set of fuzzy rules with appropriate MFs to generate the stipulated FIS. The original ANFIS was designed for type-1 (T1) Takagi-Sugeno-Kang (TSK) inference models. A hybrid learning algorithm based on the gradient method and the least-squares estimate (LSE) is applied to identify parameters.

An increasing interest in type-2 (T2) fuzzy systems has led to an increase in research on interval T2 (IT2) ANFIS. For example, early studies on T2 ANFIS can be found in [2, 3], who reported an approach that uses an adaptive network to learn a T2 fuzzy system based on linguistic inputs and numeric output. Other examples for research on T2 ANFIS, based on crisp inputs, can be found in [4, 5, 6, 7, 8, 9]. It has been proposed that least-squares based methods cannot be applied to a T2 fuzzy logic system (FLS) Mendel [10], since the *Fuzzy Basis Function* expansion for a T2 TSK FLS, which is the starting point for the LSE method, cannot be obtained without knowing the consequent parameters in advance. This issue was also concluded in [5] and addressed by giving initialised consequent parameters. In their study, it was proposed that convergence of the hybrid learning algorithms based on the recursive least-squares and the back-propagation (BP) methods can be obtained in practice. As an extension of the studies on ANFIS based on singleton fuzzification, studies on non-singleton fuzzification based IT2 ANFIS can be found in [7, 11, 12].

Though some studies has been made on IT2 ANFIS, a clear architecture for IT2 ANFIS has not been presented in literature compared to that of T1 ANFIS presented in [1]. In this paper, an extended architecture of ANFIS is proposed and clearly presented. By incorporating an extra layer for the fuzzification process, the extended architecture fits both T1 and IT2 models. The inappropriateness of the LSE algorithm for IT2 ANFIS, as mentioned by Mendel [10], is investigated in detail. The properties of the hybrid learning algorithms BP-LSE are studied on selected T1 and IT2 ANFIS models. Through both an analytical and practical exploration, we show that the LSE method does not behave in the same way for IT2 ANFIS compared to T1 ANFIS.

II. An extended architecture of ANFIS

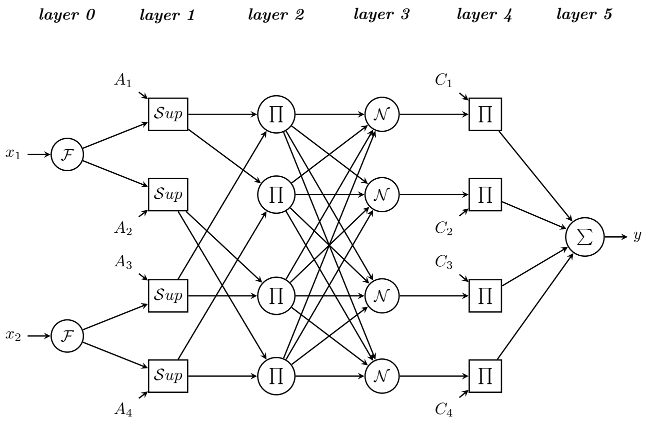

Fig. 1. An example of the extended ANFIS architecture, where the symbol $\mathcal{F}$ denotes the operation of fuzzification. $\mathcal{S}up$ in *layer 1* denotes the operation of getting the membership grades for given inputs and antecedents. The symbols $\prod$, $\mathcal{N}$ and $\sum$ denote production, normalisation and summation respectively.

In the original five-layer architecture of ANFIS presented in [1], only singleton fuzzification is considered and there is no illustration of the fuzzification step for crisp inputs. Also, the inputs of nodes in *layer 1* and *layer 4* are not clearly defined. Fig. 1 shows an extended six-layer architecture of ANFIS which makes both the fuzzification process and the node inputs more explicit. The first layer in the original architecture has been split into two layers (*layer 0* and *layer 1*) in the extended architecture. In the extended architecture, nodes in *layer 0* are fuzzifiers which are used to map crisp inputs into fuzzy sets

(FSs). Afterwards, the membership grades of the crisp inputs can be obtained in *layer 1*. The use of *layer 0* makes the fuzzification process clearer, especially when non-singleton fuzzification is applied. Also, the extended architecture of ANFIS is suitable for both T1 and IT2 ANFIS.

In the example of the extended ANFIS architecture, there are four fuzzy inferencing rules (TSK type) with two inputs (x_1, x_2) and one output (y). Specifically, the inferencing rules are:

$$\begin{aligned}
&\textbf{IF } x_1 \text{ is } A_1 \textbf{ AND } x_2 \text{ is } A_3, \textbf{ THEN } f_1 \text{ is } C_1\\
&\textbf{IF } x_1 \text{ is } A_1 \textbf{ AND } x_2 \text{ is } A_4, \textbf{ THEN } f_2 \text{ is } C_2\\
&\textbf{IF } x_1 \text{ is } A_2 \textbf{ AND } x_2 \text{ is } A_3, \textbf{ THEN } f_3 \text{ is } C_3\\
&\textbf{IF } x_1 \text{ is } A_2 \textbf{ AND } x_2 \text{ is } A_4, \textbf{ THEN } f_4 \text{ is } C_4
\end{aligned}$$

where A_1, A_2 and A_3, A_4 are the antecedent membership functions for input x_1 and x_2 respectively. C_i is the consequent linear function for the output f_i of the i^{th} rule, $i \in \{1, 2, 3, 4\}$. It should be noted that f_i is the output of each fuzzy rule and y is the output of the ANFIS. Specifically, y is the weighted average of all the f_i.

A. T1 ANFIS

For T1 ANFIS, all the antecedents are T1 FSs. For example, A_1 in the rule above can be represented as $\int_{x \in X_1} \mu_{A_1}(x)/x$ where X_1 is the universe of the primary variable x_1. The consequents can be either constants or linear functions. The most commonly used consequent representation is the first-order linear function which can be defined as:

$$C_i \equiv f_i = p_i + q_i x_1 + r_i x_2$$

where p_i, q_i, r_i are the coefficients of the linear functions, and they will be referred to as consequent parameters.

a) Layer 0: Every node i in this layer is a fuzzifier, for which the input is a crisp number and the output $O^i_{L_0}$ is a T1 FS F_i. The fuzzification method can be either singleton or T1 non-singleton. In the literature, most of the T1 ANFIS research and applications are based on singleton fuzzification, while T1 non-singleton is rarely used and can be seen in [13].

b) Layer 1: Every node i in this layer has two inputs, the fuzzy input F_j (the output from the corresponding node of *layer 0*) and the corresponding antecedent FS A_k. The output of node i is the maximum membership grade of the intersection of F_j and A_k. It can be represented as:

$$O^i_{L_1} = \sup F_j \cap A_k$$

For example, for the first node of layer 1 in Fig. 1, F_j is F_1 and A_k is A_1. Then,

$$O^1_{L_1} = \sup_{x \in X_1} \mu_{F_1}(x) \star \mu_{A_1}(x)$$

where $\star$ is the t-norm operation which can be either *min* or *prod*. It should be noted that the calculation in this layer is different from [1] where singleton fuzzification is used. Two examples are given in Figs. 2 and 3 to illustrate $F_j \cap A_k$ for singleton and non-singleton fuzzification respectively.

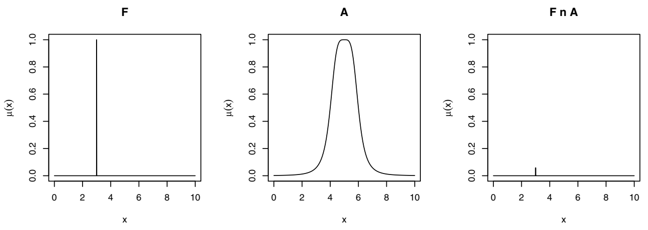

Fig. 2. An illustration of $F \cap A$ based on the *min* operator, where F denotes the fuzzy input generated by singleton fuzzification and A denotes the antecedent FS.

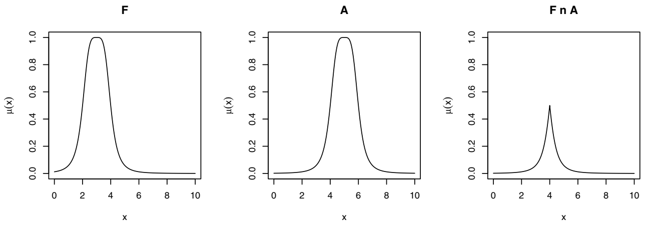

Fig. 3. An illustration of $F \cap A$ based on the *min* operator, where F denotes the fuzzy input generated by non-singleton fuzzification and A denotes the antecedent FS.

c) Layer 2 to 5: The calculations from layer 2 to 5 are the same as that defined in [1]. Thus, these layers are not explained here in detail. Only the formulas for the outputs of nodes in these layers are given below:

$$\begin{aligned}
O^i_{L_2} &\equiv w_i = O^j_{L_1} O^k_{L_1}\\
O^i_{L_3} &\equiv \overline{w}_i = \frac{O^i_{L_2}}{\sum_{j=1}^{4} O^j_{L_2}}\\
O^i_{L_4} &= O^i_{L_3} C_i = \overline{w}_i f_i\\
O_{L_5} &\equiv y = \sum_{i=1}^{4} O^i_{L_4}
\end{aligned}$$

B. IT2 ANFIS

According to the definition of T2 fuzzy logic systems by Karnik et al. [14], if there exists a IT2 FS in the antecedent or consequent MFs of an ANFIS, then such an ANFIS can be called as IT2 ANFIS. For simplicity, in this paper, all the antecedents are IT2 FSs. For example, A_1 in the rule above can be represented as two T1 MFs, which are lower MF $\underline{A}_1$ and upper MF $\bar{A}_1$. The consequent linear functions can also be defined as intervals such as $\left[\underline{C}_1, \bar{C}_1\right]$ for C_1 in Fig. 1. For example, as an equivalent representation in [12], $\left[\underline{C}_i, \bar{C}_i\right]$ for C_i can be defined as:

$$\begin{aligned}
\underline{C}_i \equiv \underline{f}_i &= p_i + q_i x_1 + r_i x_2 - s_i - t_i \left|x_1\right| - u_i \left|x_2\right|\\
\bar{C}_i \equiv \bar{f}_i &= p_i + q_i x_1 + r_i x_2 + s_i + t_i \left|x_1\right| + u_i \left|x_2\right|
\end{aligned}$$

It should be noted that the limitation for the above definition is that s_i, t_i and u_i should be no less than zero in order to guarantee that $\bar{f}_i$ is no less than $\underline{f}_i$. Alternatively, the consequent linear functions can be the same as that defined for T1 ANFIS above. In this case, $\bar{f}_i$ is equal to $\underline{f}_i$.

a) Layer 0: Similar to T1 ANFIS, every node i in this layer is a fuzzifier. However, the fuzzification method can be singleton [5], T1 non-singleton [7, 12], as well as IT2 non-singleton[11, 15]. Thus, the output $O^i_{L_0}$ can be either a T1 FS or a IT2 FS. For example, if IT2 non-singleton is used for node i, then $O^i_{L_0}$ is a IT2 FS F_i, which can be represented as a lower MF $\underline{F}_i$ and a upper MF $\bar{F}_i$. If singleton or T1 non-singleton fuzzification is used, the output $O^i_{L_0}$ is a T1 FS. However, to be simple, such a T1 FS can also be considered as a special IT2 FS for which $\underline{F}_i$ is the same as $\bar{F}_i$.

b) Layer 1: In this layer, the output $O^i_{L_1}$ of node i is a interval $\left[\underline{O}^i_{L_1}, \bar{O}^i_{L_1}\right]$, where:

$$\begin{aligned}\underline{O}^i_{L_1} &= \sup \underline{F}_j \cap \underline{A}_k \\ \bar{O}^i_{L_1} &= \sup \bar{F}_j \cap \bar{A}_k\end{aligned}$$

For example, for the first node of layer 1 in Fig. 1, F_j is F_1 and A_k is A_1. Then,

$$\begin{aligned}\underline{O}^1_{L_1} &= \sup_{x \in X_1} \mu_{\underline{F}_1}(x) \star \mu_{\underline{A}_1}(x) \\ \bar{O}^1_{L_1} &= \sup_{x \in X_1} \mu_{\bar{F}_1}(x) \star \mu_{\bar{A}_1}(x)\end{aligned}$$

c) Layer 2: The operation in this layer is similar to that for T1 ANFIS. The difference is that w_i here is an interval which can be represented as $[\underline{w}_i, \bar{w}_i]$. For instance, the output of the first node in this layer can be represented as:

$$\begin{aligned}\underline{O}^1_{L_2} \equiv \underline{w}_1 = \underline{O}^1_{L_1}\underline{O}^3_{L_1} \\ \bar{O}^1_{L_2} \equiv \bar{w}_1 = \bar{O}^1_{L_1}\bar{O}^3_{L_1}\end{aligned}$$

d) Layer 3: Before normalising the firing strengths obtained in *layer 2*, a single firing strength w'_i should be selected for each node i. Theoretically, there are infinite combinations of firing strengths and thus it is impossible to select and compute all of the possible combinations. In practice, two representative combinations are commonly used for approximations. In [9], all the $\underline{w}_i$ are used as one combination and all the $\bar{w}_i$ are used as another combination. The approach used by Mendez and Juarez [5] is based on the KM algorithms [16]. It will be explained later that the output of *layer 5* for IT2 ANFIS is constituted of $\underline{O}_{L5}$ and $\bar{O}_{L5}$. It is defined that $\underline{w}'_i$ is the selected in one combination for $\underline{O}_{L5}$, and $\bar{w}'_i$ is selected in another combination for $\underline{O}_{L5}$, where $\underline{w}'_i, \bar{w}'_i \in [\underline{w}_i, \bar{w}_i]$. Then, the output $O^i_{L_3}$ can be represented as $\underline{O}^i_{L_3}$ and $\bar{O}^i_{L_3}$, where:

$$\begin{aligned}\underline{O}^i_{L_3} &\equiv \overline{\underline{w}}_i = \frac{\underline{w}'_i}{\sum_{j=1}^4 \underline{w}'_j} \\ \bar{O}^i_{L_3} &\equiv \overline{\bar{w}}_i = \frac{\bar{w}'_i}{\sum_{j=1}^4 \bar{w}'_j}\end{aligned}$$

It should be noted that $\underline{w}'_i$ does not necessarily need to be smaller than $\bar{w}'_i$. Also, $\underline{O}^i_{L_3}$ $(\overline{\underline{w}}_i)$ does not necessarily need to be smaller than $\bar{O}^i_{L_3}$ $(\overline{\bar{w}}_i)$.

e) Layer 4: The output $O^i_{L_4}$ of the i^{th} node in this layer can be represented as $\underline{O}^i_{L_4}$ and $\bar{O}^i_{L_4}$, where:

$$\begin{aligned}\underline{O}^i_{L_4} = \overline{\underline{w}}_i\underline{f}_i = \underline{O}^i_{L_3}\underline{C}_i \\ \bar{O}^i_{L_4} = \overline{\bar{w}}_i\bar{f}_i = \bar{O}^i_{L_3}\bar{C}_i\end{aligned}$$

It should also be noted that $\underline{O}^i_{L_4}$ does not necessarily need to be smaller than $\bar{O}^i_{L_4}$.

f) Layer 5: O_{L_5}, which is the final output y, is constituted of $\underline{O}_{L5}$ and $\bar{O}_{L5}$. These can be represented as:

$$O_{L_5} \equiv y = q\underline{O}_{L5} + (1-q)\bar{O}_{L5}$$

where

$$\begin{aligned}\underline{O}_{L5} &= \sum_{i=1}^{4} \underline{O}^i_{L_4} = \sum_{i=1}^{4} \overline{\underline{w}}_i\underline{f}_i \\ \bar{O}_{L5} &= \sum_{i=1}^{4} \bar{O}^i_{L_4} = \sum_{i=1}^{4} \overline{\bar{w}}_i\bar{f}_i\end{aligned}$$

and as presented in [9], q is a constant, which is usually set to be 0.5 [12]. It should be noted that $\underline{O}_{L5}$ is guaranteed to be no larger than $\underline{O}_{L5}$ if the selection approach in *layer 3* is based on the KM algorithms. In contrast, this cannot be guaranteed in [9].

III. The Learning Algorithms of ANFIS

The learning algorithms of ANFIS are usually related to methods for updating the antecedent parameters and consequent parameters during the training process. Though not found in the literature, it is also possible to learn the parameters for fuzzification.

The back-propagation based on the gradient descent methods and the LSE algorithm are commonly used as the learning algorithms for updating the parameters [1]. Recursive square-root filters (REFIL) and orthogonal least-squares (OLS) algorithms are used in [11, 12] for learning the consequent parameters. Some other algorithms such as Particle Swarm Optimisation (PSO) and the Sliding Mode Control (SMC) approach are also reported as learning algorithms for IT2 ANFIS [17, 9]. The remains of this section are focused on the LSE algorithm, which has been proposed to be inappropriate for IT2 ANFIS [10].

A. Linear equation for T1 ANFIS

For a single entry of the training data, it is observed that the final output y of T1 ANFIS described in Section II-A can be represented as:

$$y = \vec{X}\vec{\theta}$$

where $\vec{X}$ and $\vec{\theta}$ are vectors which can be represented as:

$$\begin{aligned}\vec{X} &= \begin{pmatrix}\overline{w}_1 & \overline{w}_1x_1 & \overline{w}_1x_2 & ... & \overline{w}_4 & \overline{w}_4x_1 & \overline{w}_4x_2\end{pmatrix} \\ \vec{\theta}^T &= \begin{pmatrix}p_1 & q_1 & r_1 & ... & p_4 & q_4 & r_4\end{pmatrix}\end{aligned}$$

$$\vec{X} = \begin{pmatrix} \frac{\overline{\underline{w}}_1+\overline{\bar{w}}_1}{2} & \frac{\overline{\underline{w}}_1+\overline{\bar{w}}_1}{2}x_1 & \frac{\overline{\underline{w}}_1+\overline{\bar{w}}_1}{2}x_2 & \dots & \frac{\overline{\underline{w}}_4+\overline{\bar{w}}_4}{2} & \frac{\overline{\underline{w}}_4+\overline{\bar{w}}_4}{2}x_1 & \frac{\overline{\underline{w}}_4+\overline{\bar{w}}_4}{2}x_2 \end{pmatrix} \tag{1}$$

$$\vec{\theta}^T = \begin{pmatrix} p_1 & q_1 & r_1 & \dots & p_4 & q_4 & r_4 \end{pmatrix} \tag{2}$$

B. Linear equation for IT2 ANFIS

For simplicity, the consequent linear functions for IT2 ANFIS are defined to be the same as those for T1 ANFIS above. Then, for a single entry of the training data, the final output y of IT2 ANFIS described above can also be represented as:

$$y = \vec{X}\vec{\theta}$$

where $\vec{X}$ and $\vec{\theta}$ are vectors which can be represented as Equation 1 and 2 (supposing the coefficient q for O_{L_5} is set to be 0.5).

It should be noted that, if the KM algorithms are applied, $\vec{\theta}$ should firstly be initialised in order to obtain the matrix X by properly selecting the rule firing strengths from the output interval $[\underline{w}_i, \bar{w}_i]$ for each node i in *layer 2*.

The consequent linear functions can also be easily defined to be intervals as $\left[\underline{C}, \bar{C}\right]$. However, during the training process, it should be assured that $\underline{C}$ is no larger than $\bar{C}$. For example, as mentioned in Section II-B, b should be no less than zero for the defined consequent intervals.

C. LSE and its inappropriateness for IT2 ANFIS

For multiple entries of the training data, the output Y for both T1 and IT2 ANFIS can be represented as the following matrix equation:

$$\vec{Y} = X\vec{\theta}$$

Hence, given the input matrix X and the output $\vec{Y}$, the parameters $\vec{\theta}$ can be easily estimated by the LSE algorithm. For T1 ANFIS, the estimated $\vec{\theta}$ can fit the training data with an optimised training error, which is the same as the estimation error (residual) obtained by the LSE. However, it should be noted that the LSE algorithm cannot guarantee an optimised training error for IT2 ANFIS. This is because the estimation error for IT2 ANFIS obtained by the LSE algorithm will almost certainly be different from the final training error for the training data. To clearly illustrate this issue, the estimation error $\vec{e}'$ and the training error $\vec{e}''$ are defined as:

$$\vec{e}' = \vec{Y} - X'\vec{\theta}''$$
$$\vec{e}'' = \vec{Y} - X''\vec{\theta}''$$

where $\vec{\theta}''$ is estimated by the LSE algorithm with the input X' and the output $\vec{Y}$. It should be noted that when the same training data are used to validate the estimated $\vec{\theta}''$, the output $\vec{Y}$ does not change for calculating $\vec{e}''$. However, the input matrix could change and hence it is defined as X''.

Specifically, for T1 ANFIS, X'' is the same as X'. Thus, $\vec{e}''$ will be exactly the same as $\vec{e}'$. This means, for T1 ANFIS, the training error is the estimation error obtained by the LSE algorithm for the training data.

However, the situation for IT2 ANFIS is different from that for T1 ANFIS. For IT2 ANFIS, X' is determined by the initialised $\vec{\theta}'$ when the KM algorithms are applied for selecting the representative firing strengths in *layer 2*. After applying the LSE algorithm, the consequent parameters will be updated to $\vec{\theta}''$, which is optimised for the input matrix X'. It should be noted that $\vec{\theta}''$ will almost certainly be different from the initialised $\vec{\theta}'$. Thus, when the same training data are then used in the validation process, the input matrix X'', which is determined by the new parameters $\vec{\theta}''$, will almost certainly be different from the input matrix X' used by the LSE algorithm. For example, $\overline{\underline{w}}_1$ could be $\frac{\underline{w}_1}{\underline{w}_1+\bar{w}_2+\bar{w}_3+\bar{w}_4}$ for X', while it may be $\frac{\underline{w}_1}{\underline{w}_1+\underline{w}_2+\underline{w}_3+\underline{w}_4}$ for X''. Accordingly, $\vec{e}''$ will almost certainly be different from $\vec{e}'$ for IT2 ANFIS. This means, $\vec{\theta}''$ is optimised by the LSE algorithm for X' and $\vec{Y}$. However, it does not fit X'' as well as it fits X'. In other words, for IT2 ANFIS, the LSE method cannot be used to guarantee an optimised training error for the training data when the KM algorithms are applied for selecting the representative firing strengths in *layer 2*.

IV. Evaluation

In Section III, we discussed how the LSE method can be applied to T1 ANFIS to obtain an optimised error for the training data. However, such optimised error cannot be obtained by the LSE method in IT2 ANFIS in the same way as that in T1 ANFIS. In this section, we use the classification problem with the IRIS data [18] to further investigate this issue in practice.

In our testing models, five fuzzy rules are used where petal length (x_1) and petal width (x_2) are inputs, and species (C) are outputs. Each input has three MFs. Specifically, the five rules are:

IF x_1 is $A_{x_1,low}$ **AND** x_2 is $A_{x_2,low}$, **THEN** C_1
IF x_1 is $A_{x_1,mid}$ **AND** x_2 is $A_{x_2,mid}$, **THEN** C_2
IF x_1 is $A_{x_1,high}$ **AND** x_2 is $A_{x_2,high}$, **THEN** C_3
IF x_1 is $A_{x_1,mid}$ **AND** x_2 is $A_{x_2,high}$, **THEN** C_4
IF x_1 is $A_{x_1,high}$ **AND** x_2 is $A_{x_2,mid}$, **THEN** C_5

where the antecedents are gauss-bell shaped T1 and IT2 FSs for T1 and IT2 ANFIS models respectively. The consequent linear functions are defined as described in Section II-A for both T1 and IT2 ANFIS models. Thus, in this evaluation, T1 and IT2 ANFIS models have the same number of consequent parameters that are to be estimated by the LSE method.

A. Training with pure LSE

To clearly show how the LSE method works in T1 and IT2 ANFIS models, only the LSE method are used to tune the consequent parameters. The antecedent MFs are arbitrarily initialised and then are fixed during the training process. All the 150 instances of the IRIS data are used as the training data. The estimation error $\vec{e}'$ and the training error $\vec{e}''$ defined in Section III-C are examined during the training epochs with both batch and online training modes. With the batch mode, parameters are updated only after the whole training data set has been presented for each epoch. On the other hand, with the online mode, parameters are updated immediately after each input-output pair of training data within each epoch.

For the batch training mode, the results are shown in Fig. 4. It can be observed that the relevant estimation and training errors by the T1 model are the same. However, clear differences can be identified between the estimation and training errors by the IT2 model. Also, the errors obtained by the T1 ANFIS model do not change during the training epochs, while the errors by the IT2 ANFIS model fluctuate. No clear error convergence can be found in this figure. Some of the estimation errors by the IT2 ANFIS model are smaller than the relevant estimation errors by the T1 ANFIS model. However, almost all the training errors by the IT2 ANFIS model are larger than the training errors by the T1 ANFIS model.

The results of online mode are shown in Fig. 5. It can be noticed the results of T1 ANFIS with online mode are the same to those with the batch mode above. However, the IT2 model performs differently with online mode. Although the differences between estimation and training errors obtained by the IT2 ANFIS model still exist, they cannot be clearly identified especially after a couple of training epochs. Though not clear, error convergence can be found for the IT2 model. However, all the errors obtained by the IT2 ANFIS model are larger than those produced by the T1 ANFIS model.

B. Training with BP-LSE

More tests have been made with BP-LSE, as the hybrid learning algorithm, to further examine how the LSE method performs incorporating the gradient method for T1 and IT2 ANFIS models. Specifically, the parameters in Layer 2 are updated by the BP algorithm and the parameters in Layer 4 are estimated by the LSE method. The results are shown in Figs. 6 and 7 for batch and online training modes respectively.

In batch mode, the estimation and training errors for each epoch for T1 ANFIS are still the same. However, it should be noted that such consistency disappears when the T1 ANFIS model is trained in online mode. In contrast, clear differences between estimation and training errors for IT2 ANFIS can be observed in both batch and online modes.

With BP-LSE, fluctuations can be clearly observed from all the errors except that obtained by the T1 model in the batch mode. However, all the errors converge during the training epochs. It can be observed that, in this case, the T1 ANFIS model still gives smaller errors than the IT2 ANFIS model after a couple of training epochs.

V. Discussion

An IT2 ANFIS can be considered as a combination of all possible embedded T1 ANFIS. According to the definition in Section II-B, the output of IT2 ANFIS is in fact the output of its representative embedded T1 ANFIS, which is determined by the consequent parameters if the KM algorithms are applied for selecting the representative firing strengths. Thus, the optimisation of an IT2 ANFIS is in fact the optimisation of the representative T1 ANFIS.

As can be clearly observed from our results, the training errors by IT2 ANFIS are generally larger than the corresponding estimation errors. This is because the estimation errors are least square errors optimised by the LSE method for the selected representative embedded T1 ANFIS. However, as discussed in Section III-C, since the consequent parameters are updated after LSE optimisation, the representative embedded T1 ANFIS will also be changed. Hence, the updated consequent parameters do not fit the new representative embedded T1 ANFIS. As a result, the training errors will generally be larger than the estimation errors.

It should be noted that when the hybrid learning algorithm (BP-LSE) is applied in online mode, the estimation errors are also not equal to the training errors for T1 ANFIS (see Fig. 7). This is because the antecedent parameters are updated within each epoch for every entry of the training data. This makes X'' different from X' when errors are checked after all entries of the training data are learned for each learning epoch.

In Fig. 7, it is also interesting to note that the estimation errors gradually increase during the training epochs. This is a special property of the LSE method in online training mode. The BP-LSE algorithm in online mode does not guarantee an optimised fit for the training data. Therefore the estimation errors for T1 ANFIS in Fig. 7 are larger than that in Fig. 6.

In Figs. 4 and 6, errors obtained by IT2 ANFIS during the training epochs show more fluctuation than errors obtained by T1 ANFIS. The fluctuation of errors is in essence related to the change of the matrix X during the training epochs. For T1 ANFIS, the change of the matrix X is only related to the change of antecedent parameters. However, for IT2 ANFIS, it is related to the change of both the antecedent and the consequent parameters. Thus, the change of matrix X for IT2 ANFIS is potentially larger than that for T1 ANFIS. Thus, the larger change of the matrix X leads to the larger fluctuation. This is also the reason for the smaller error fluctuations seen in online mode compared to batch mode: the change of parameters in online mode is relatively smaller than that in batch mode.

In general, IT2 ANFIS produces worse RMSE values than T1 ANFIS. This is because IT2 ANFIS is in fact never well optimised by the LSE method. As discussed above, the LSE method can only be used to optimise the representative embedded T1 ANFIS. However, the optimised consequent parameters will leave a new unoptimised representative embedded T1 ANFIS. It is commonly acknowledged that IT2 FIS should be better in fitting the data than T1 FIS. Thus, it can be concluded

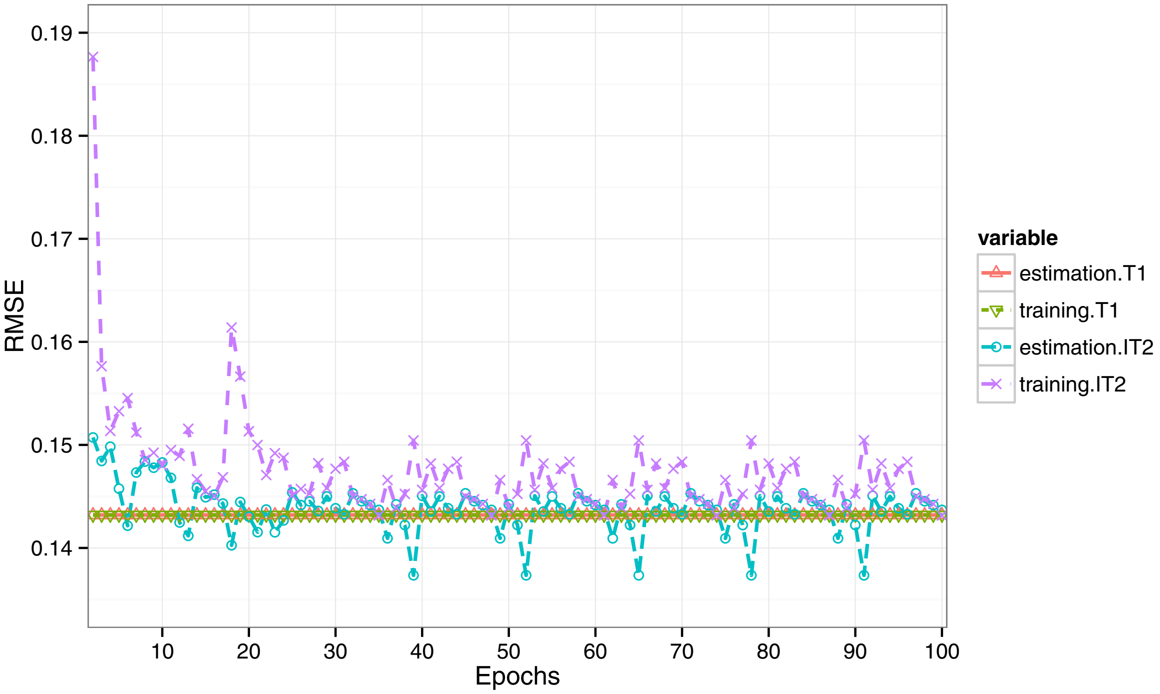

Fig. 4. The estimation and training error curves for the training data during the pure LSE training epochs with IRIS data (batch mode)

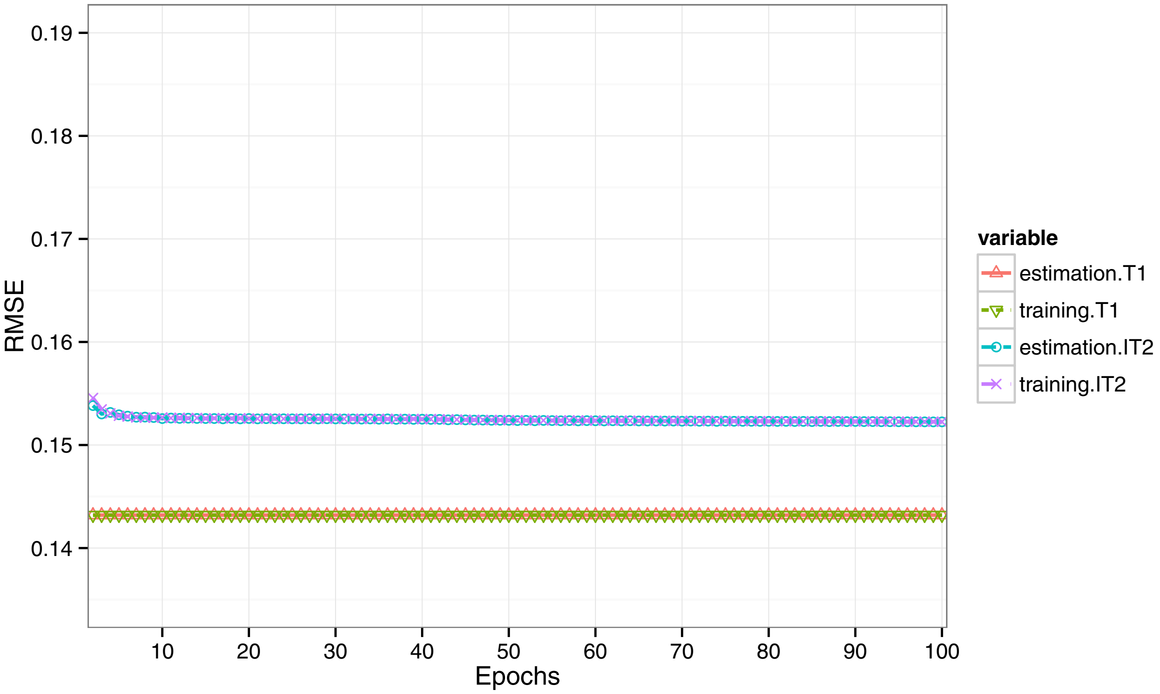

Fig. 5. The estimation and training error curves for the training data during the pure LSE training epochs with IRIS data (online mode)

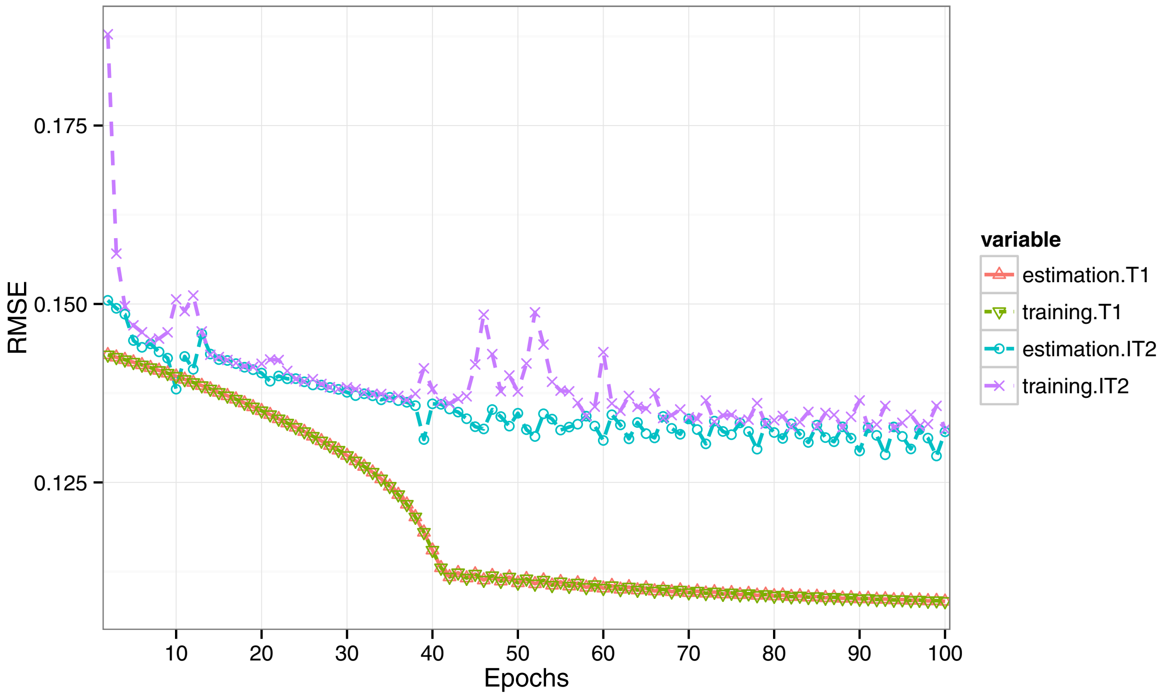

Fig. 6. The estimation and training error curves for the training data during the hybrid (BP-LSE) training epochs with IRIS data (batch mode)

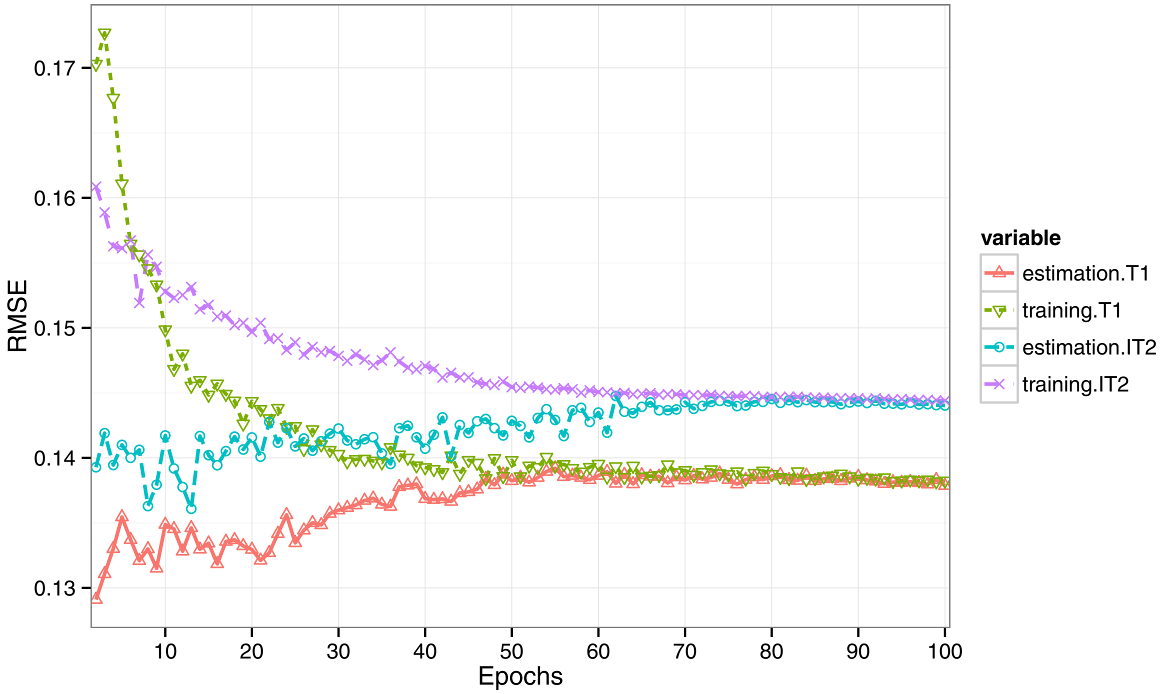

Fig. 7. The estimation and training error curves for the training data during the hybrid (BP-LSE) training epochs with IRIS data (online mode)

that the LSE method is not appropriate for IT2 ANFIS when the KM algorithms are applied for selecting the representative firing strengths in *layer 2*.

VI. Conclusion

We have presented an extended six-layer ANFIS architecture suited to both T1 and IT2 ANFIS. By incorporating an extra layer, the fuzzification process is explicitly addressed compared to the original five-layer architecture. Both singleton and non-singleton fuzzification methods are supported by our extended architecture. Through a detailed discussion, we have shown that the LSE method does not exhibit the same behaviour for KM-based IT2 ANFIS compared to T1 ANFIS. This is supported in practice through results presented on the IRIS classification problem. In our evaluation, T1 ANFIS models generally produce smaller errors than IT2 ANFIS models when they are trained by either the pure LSE or the hybrid BP-LSE algorithms. Thus, we conclude that the LSE method is not generally suitable as the training algorithm for IT2 ANFIS models when the KM algorithms are applied for selecting the representative firing strengths in *layer 2*. However, a simple data set is obviously not enough, the results need to be verified with more data sets in the future. On the other hand, we are currently exploring alternatives to the KM algorithms, such as the Nie-Tan method [19], which would allow the LSE method to be used for IT2 ANFIS models.

References

[1] J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[2] R. I. John and C. Czarnecki, "A type 2 adaptive fuzzy inferencing system," in *Proceedings IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, 1998, pp. 2068–2073.

[3] R. I. John and C. Czarnecki, "An adaptive type-2 fuzzy system for learning linguistic membership grades," in *Proceedings IEEE International Conference on Fuzzy Systems*, vol. 3, 1999, pp. 1552–1556.

[4] G. M. Mendez and O. Castillo, "Interval type-2 TSK fuzzy logic systems using hybrid learning algorithm," in *Proceedings IEEE International Conference on Fuzzy Systems*, 2005, pp. 230–235.

[5] G. Mendez and I. Juarez, "First-order interval type-2 TSK fuzzy logic systems using a hybrid learning algorithm," *WSEAS Transactions on Computers*, vol. 4, no. 4, pp. 378–384, 2005.

[6] G. Mendez and M. De Los Angeles Hernandez, "Interval Type-2 ANFIS," in *Innovations in Hybrid Intelligent Systems*, E. Corchado, J. Corchado, and A. Abraham, Eds. Springer Berlin Heidelberg, 2007, vol. 44, pp. 64–71.

[7] G. Mendez and M. Hernandez, "Interval type-1 non-singleton type-2 fuzzy logic systems are type-2 adaptive neuro-fuzzy inference systems." *International Journal of Reasoning-based Intelligent Systems*, vol. 2, no. 2, pp. 95–99, 2010.

[8] M. A. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak, "Extended Kalman Filter Based Learning Algorithm for Type-2 Fuzzy Logic Systems and Its Experimental Evaluation," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4443–4455, 2012.

[9] E. Kayacan, O. Cigdem, and O. Kaynak, "Sliding Mode Control Approach for Online Learning as Applied to Type-2 Fuzzy Neural Networks and Its Experimental Evaluation," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 9, pp. 3510–3520, 2012.

[10] J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall, 2001.

[11] G. Méndez and M. Hernández, "Hybrid learning mechanism for interval A2-C1 type-2 non-singleton type-2 TakagiSugenoKang fuzzy logic systems," *Information Sciences*, vol. 220, pp. 149–169, 2013.

[12] M. Hernandez, P. Melin, G. Méndez, O. Castillo, and I. López-Juarez, "A hybrid learning method composed by the orthogonal least-squares and the back-propagation learning algorithms for interval A2-C1 type-1 non-singleton type-2 TSK fuzzy logic systems," *Soft Computing*, vol. 19, no. 3, pp. 661–678, 2015.

[13] G. Méndez and M. Hernandez, "Hybrid learning for interval type-2 fuzzy logic systems based on orthogonal least-squares and back-propagation methods," *Information Sciences*, vol. 179, no. 13, pp. 2146–2157, 2009.

[14] N. Karnik, J. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643–658, 1999.

[15] H. MonirVaghefi, M. Rafiee Sandgani, and M. Aliyari Shoorehdeli, "Interval Type-2 Adaptive Network-based Fuzzy Inference System (ANFIS) with Type-2 non-singleton fuzzification," in *Proceedings Iranian Conference on Fuzzy Systems*, 2013, pp. 1–6.

[16] N. Karnik and J. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, no. 14, pp. 195–220, 2001.

[17] M. Khanesar, M. Teshnehlab, E. Kayacan, and O. Kaynak, "A novel type-2 fuzzy membership function: application to the prediction of noisy data," in *Proceedings IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2010, pp. 128–133.

[18] J. Bezdek, J. Keller, R. Krishnapuram, L. Kuncheva, and N. Pal, "Will the real iris data please stand up?" *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 3, pp. 368–369, 1999.

[19] M. Nie and W. W. Tan, "Towards an efficient type-reduction method for interval type-2 fuzzy logic systems," in *Proceedings IEEE International Conference on Fuzzy Systems*, 2008, pp. 1425–1432.