17th Mediterranean Conference on Control & Automation
Makedonia Palace, Thessaloniki, Greece
June 24 - 26, 2009

An Efficient Structure Learning Algorithm For A Self-Organizing Neuro-Fuzzy Multilayered Classifier

Nikolaos E. Mitrakis and John B. Theocharis
Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
Thessaloniki 54124, Greece.
Email: nmitr@ee.auth.gr, theochar@eng.auth.gr

Abstract—**In authors' previous works, a novel self-organizing neuro-fuzzy multilayered classifier (SONeFMUC) was proposed. SONeFMUC is composed of small-scale interconnected fuzzy neuron classifiers (FNCs) arranged in layers. The structure of the classifier is revealed by means of the well known GMDH algorithm. In addition, the GMDH algorithm inherently implements feature selection, considering the most informative attributes as model inputs. However, previous simulation results indicate that the GMDH algorithm calculates a large number of FNCs with slightly higher or even the same classification capabilities than its parents. Hence, the computational cost of the GMDH is large without a direct impact to the classification accuracy. In this paper, a modified version of GMDH is proposed for an effective identification of the structure of SONeFMUC with reduced computational cost. To this end, a statistical measure of agreement of the generic FNCs in classifying the patterns of the problem is used. This measure is known as Proportion of Specific Agreement (Ps). Hence, only complementary FNCs are combined to construct a descendant FNC at the next layer and the total number of constructed FNCs is reduced. The proposed structure learning algorithm is tested on a well known classification problem of the literature, the forensic glass. Simulation results indicate the efficiency of the proposed algorithm.**

Keywords-structure learning; GMDH; neuro-fuzzy classifier; classifiers combination; decision fusion

I. INTRODUCTION

Neuro-fuzzy classifiers are well known among the researchers for their good generalization capabilities. The main feature of these techniques is the combination of the advantages of fuzzy logic and neural networks. Most of the neuro-fuzzy classifiers are represented by a multilayered structure. Among the most popular architectures are NEFCLASS [1], min–max and min–sum fuzzy inference networks [2], SANFIS [3] and FLEXNFIS [4].

A different approach is proposed in [5], [6] where the authors present a new type of self-organizing neuro-fuzzy multilayered classifier (SONeFMUC). SONeFMUC is a multilayered structure with small generic neuro-fuzzy classifiers interconnected in a layered structure similar to that of neural networks. In order to reveal the appropriate structure of the model, the Group Method of Data Handling (GMDH) [7] algorithm is used. This structure learning algorithm is capable of recognizing not only the number of layers and nodes but also tackling simultaneously the feature selection problem. The GMDH algorithm has been used widely in polynomial neural networks for solving modeling problems, [8]. However, apart from SONeFMUC, there are few classifiers which apply the GMDH algorithm for structure identification, [9], [10].

Previous simulation results, [5], [6], revealed the powerful capabilities of GMDH in structure identification and feature selection. However, the main drawback of the algorithm is that it demands a large number of candidate nodes (generic neuro-fuzzy classifiers) in order to form the best set of nodes in each layer. This demand results in an increase of the computational time of model's structure identification. In addition, simulation results indicate that many of the constructed generic neuro-fuzzy classifiers don't improve the classification capabilities of their parents and hence, the computational time is increased without a direct improvement in classification accuracy.

In this paper, a new modified version of the GMDH algorithm is proposed for efficient structure identification of a neuro-fuzzy multilayered classifier. Using a simple statistical measure, preliminary knowledge is gained concerning the result of a combination of generic neuro-fuzzy classifiers. The proposed structure learning algorithm reduces the computational time with a simultaneously improvement in final classification accuracy.

II. SONEFMUC ARCHITECTURE

The structure of the classifier is composed of a number of L layers, where the $\ell - th$ layer includes N_ℓ neurons. Neurons are considered as Fuzzy Neuron Classifiers (FNCs), denoted by $FNC_j^{(\ell)}$, $j = 0,...,N_\ell$, $\ell = 0,...,L$. The input layer $\ell = 0$ includes the feature components $x_1,\ldots,x_m$, while the output layer $\ell = L$ comprises the output node $FNC_1^{(L)}$. Fig. 1 shows an example of a three layered SONeFMUC architecture. Supervised learning is conducted using a training data set of N labeled pairs: $D_N = \{(\boldsymbol{x}[q], C[q]), q = 1,...,N\}$, and $C = \{C_1,...,C_M\}$ is the set of M classes.

Parent FNCs are combined to generate a descendant FNC at the next layer, with better classification capabilities. Each FNC provides two outputs: a decision support vector of length

Authorized licensed use limited to: UNIVERSITY OF WINDSOR. Downloaded on February 13,2010 at 21:07:06 EST from IEEE Xplore. Restrictions apply.

M (dotted thick line) and a transformed feature vector of length p (solid thick line). In this respect, the connective links can be regarded as data buses transferring information from a layer to the succeeding one. The ending $FNC_1^{(L)}$ produces only a decision vector $D(\boldsymbol{x}) = [d_1(\boldsymbol{x}),...,d_M(\boldsymbol{x})]^T$ representing the overall output of SONeFMUC, where $d_j(\boldsymbol{x}) \in [0,1]$, $j = 1,...,M$, denotes the certainty grade of $\boldsymbol{x}$ in class C_j .

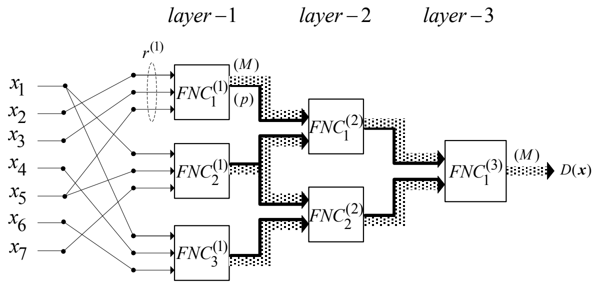

Figure 1. Example of a SONeFMUC structure with $m = 7$ and $L = 3$.

III. FNC ARCHITECTURE

Each FNC comprises four modules: $FNC_k^{(\ell)} = \{\mathcal{F}_k^{(\ell)}, DS_k^{(\ell)}, FPD_k^{(\ell)}, DMFU_k^{(\ell)}\}$, Fig. 2. The fuser $\mathcal{F}_k^{(\ell)}$ combines the decision outputs of the parent FNCs while the associated unit $DS_k^{(\ell)}$ performs a data splitting into well separated patterns and ambiguous ones. The pair of modules $\{FPD_k^{(\ell)}, DMFU_k^{(\ell)}\}$ implements an elementary neuro-fuzzy classifier within each $FNC_k^{(\ell)}$, which is used to improve the accuracy of ambiguous patterns. The $FPD_k^{(\ell)}$ part performs a feature transformation through supervised learning whereas $DMFU_k^{(\ell)}$ produces soft decision supports for the patterns.

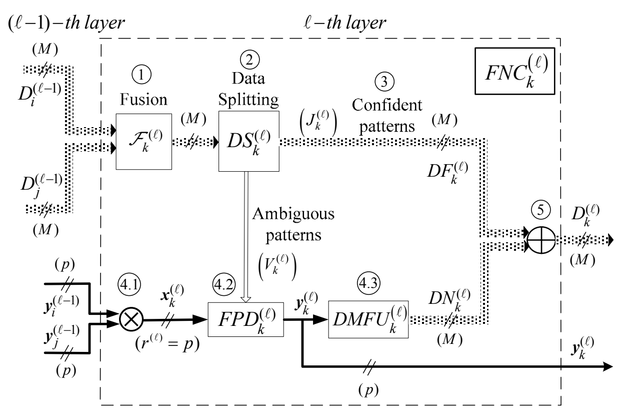

Figure 2. General structure of the $k-th$ FNC at layer ℓ .

A. Fuzzy Partial Description (FPD)

The $FPD_k^{(\ell)}$ units are represented by a TSK fuzzy system [11], with $r^{(\ell)}$ inputs, $\boldsymbol{x}_k^{(\ell)} \in \Re^{r^{(\ell)}}$, and p outputs, $\boldsymbol{y}_k^{(\ell)} \in \Re^p$, defined on the normalized range $[0,1]$. The input-output components form the input space $\mathcal{I}_k^{(\ell)} = [0,1]^{r^{(\ell)}}$ and output space $\mathcal{S}_k^{(\ell)} = [0,1]^p$ of the $FPD_k^{(\ell)}$, respectively. The input vector $\boldsymbol{x}_k^{(\ell)}$ is formed by aggregating the outputs of the parent $FNC_i^{(\ell-1)}$ and $FNC_j^{(\ell-1)}$ from the previous layer:

$$\boldsymbol{x}_k^{(\ell)} = \boldsymbol{y}_i^{(\ell-1)} \otimes \boldsymbol{y}_j^{(\ell-1)} = \frac{w_1}{w_1+w_2}\boldsymbol{y}_i^{(\ell-1)} + \frac{w_2}{w_1+w_2}\boldsymbol{y}_j^{(\ell-1)} \quad (1)$$

The weights represent the performance of the parent FNCs on a validation data set. The aggregation rule (1) confines the input size to $r^{(\ell)} = p$ for $\ell \geq 2$.

FNCs at layer 1 of the SONeFMUC operate on the original feature set, $\{x_1,...,x_m\}$ (Fig. 1). Instead of receiving the whole attribute set, the input vector $x_k^{(1)}$ of $FPD_k^{(1)}$, contains a small subset of $r^{(1)} = 2,3$ features taken from the original set. The particular features serving as inputs to each FPD are determined via the structure learning algorithm described in section IV. Further, since the FNCs in layer 1 obtain no previous decision evidence, fusion is not applied at this layer.

The output space $\mathcal{S}_k^{(\ell)}$ is a p-dimensional hyper-cube representing the class space, i.e., the space where the classes are defined. The value of p, and hence the number of FPD outputs, is determined as a function of the number of classes: $p = ceil(\log_2 M)$.

Definition of classes entails that for each class a target value is assigned for all output variables: $\boldsymbol{y}_d^{(C_j)} = \left[y_{d,1}^{(C_j)},...,y_{d,p}^{(C_j)}\right]$, $j = 1,..,M$. The class targets are located at the 2^p vertices of the hyper-cube $\mathcal{S}_k^{(\ell)}$.

Omitting for simplicity the layer index ℓ , let the input/output vectors of FPDs be denoted as $\boldsymbol{x} = [x_1,..,x_r]^T$ and $\boldsymbol{y} = [y_1,..,y_p]^T$. Each premise variable, x_i , $i = 1,..,r$, is described in terms of K_i fuzzy sets, $\{A_1^{(i)}, A_2^{(i)},..., A_{K_i}^{(i)}\}$. The fuzzy sets are represented by two-sided Gaussian MFs.

Considering a grid-type partition of the premise space, a total number of $R = \prod_{i=1}^{r} K_i$ fuzzy rectangular subspaces are formed, $\mathcal{A}^{(i)} = A_{i_1}^{(i)} \times \cdots \times A_{i_r}^{(i)}$. Defining a fuzzy rule for each $\mathcal{A}^{(i)}$, the FPDs are described by R TSK rules of the form:

Authorized licensed use limited to: UNIVERSITY OF WINDSOR. Downloaded on February 13,2010 at 21:07:06 EST from IEEE Xplore. Restrictions apply.

$$R_m^{(i)}: IF\ x_1\ is\ A_{i_1}^{(1)}\ AND\ \cdots\ AND\ x_r\ is\ A_{i_r}^{(r)}$$
$$THEN\quad y_1^{(i)} = g_1^{(i)}(\boldsymbol{x})\ AND \cdots AND\ y_p^{(i)} = g_p^{(i)}(\boldsymbol{x}) \quad (2)$$

The consequent functions of the TSK fuzzy model are described by linear polynomials of the input variables:

$$g_j^{(i)}(\boldsymbol{x}) = w_{0,j}^{(i)} + w_{1,j}^{(i)} x_1 + ... + w_{r,j}^{(i)} x_r, \quad j = 1,...,p \qquad (3)$$

A simplified rule form with crisp consequents can be obtained from (3) by retaining only the constant term: $g_j^{(i)}(\boldsymbol{x}) = w_{0,j}^{(i)}\ j = 1,...,p$.

The outputs of the FPD model are given as

$$y_j = \sum_{i=1}^{R} \bar{\mu}_i(\boldsymbol{x}) \cdot g_j^{(i)}(\boldsymbol{x}), \ j = 1,...,p \qquad (4)$$

where $\bar{\mu}_i(\boldsymbol{x})$ are the normalized firing strengths of the rules.

The structure and parameters of $FPD_k^{(\ell)}$ are determined using the following FPD learning procedure:

F.1 (*Rule base formulation*): Given K_i formulate R fuzzy rules of the form (2) with uniformly placed MFs.

F.2 (*Premise partition*): Apply the *K–means* clustering method [12] on the MF centers.

F.3 (*Rule base simplification*): Preserve the rules that cover (firing fulfilling $\mu_i(\boldsymbol{x}) \geq 0.5$) a number of patterns greater than a prescribed threshold (i.e., 5% of the training patterns) and discard the rest ones.

F.4 (*Consequent weights estimation)*: Since the FPD outputs are linear with respect to the polynomial coefficients (see (2)-(4)), calculate the optimal estimates of the consequent weights $w_{k,j}^{(i)}$ using the RLSE algorithm as in [5].

F.5 (*FPD outputs*): Compute the FPD outputs $\boldsymbol{y}_k^{(\ell)}$ using (4).

B. Decision Making Fuzzy Unit (DMFU)

The $DMFU_k^{(\ell)}$ is a fuzzy rule-based system operating on the space $\mathcal{S}_k^{(\ell)}$ of the transformed features. The outputs $\boldsymbol{y}_k^{(\ell)}$ of $FPD_k^{(\ell)}$ serve as inputs to $DMFU_k^{(\ell)}$ (Fig. 2). For the patterns $\boldsymbol{x}$ handled by $FPD_k^{(\ell)}$, $DMFU_k^{(\ell)}$ produces a soft decision vector including the degrees of support in each class: $DN_k^{(\ell)}(\boldsymbol{x}) = \left[dn_{k,1}^{(\ell)}(\boldsymbol{x}), dn_{k,2}^{(\ell)}(\boldsymbol{x}),....,dn_{k,M}^{(\ell)}(\boldsymbol{x})\right]^T$.

Each *DMFU* input y_j, $j = 1,...,p$ is partitioned into two fuzzy sets $\{B_1^{(j)}, B_2^{(j)}\}$ which are represented by trapezoidal membership function, centered at the target values of the classes. Regions with overlapping fuzzy sets are ambiguous regions between the classes, providing low degrees of certainty for the patterns. On the contrary, patterns belonging to the rectangular regions surrounding the class targets are well classified to a high degree of confidence.

Combining fuzzy sets along the inputs of DMFU, we form a partition comprising rectangular fuzzy spaces: $\mathcal{B}^{(i)} = B_{i_1}^{(1)} \times ... \times B_{i_p}^{(p)}, i = 1,...,2^p$. For each $\mathcal{B}^{(i)}$ a fuzzy classification rule is applied of the form:

$$R_c^{(i)}:\ IF\ y_1\ is\ B_{i_1}^{(1)}\ AND....AND\ y_p\ is\ B_{i_p}^{(p)}$$
$$THEN\ \left(y_1[q],...,y_p[q]\right)\ is\ C_i,\ C_i \in C \qquad (5)$$

The fuzzy inference of $DMFU_k^{(\ell)}$ proceeds as follows:

D.1 Calculate the degree of firing of the classes, $\beta_{C_i} = \mu_{i_1}^{(1)}\left(y_1(q)\right) \wedge \ldots \wedge \mu_{i_p}^{(2)}\left(y_p(q)\right),\ i = 1,...,M$

D.2. Calculate the soft decision output vector of DMFU:

$$DN = \left[dn_1, dn_2, ..., dn_M\right]^T = \left[\bar{\beta}_{C_1}, \bar{\beta}_{C_2}, ..., \bar{\beta}_{C_M}\right]^T \qquad (6)$$

where $\bar{\beta}_{C_i}$ denotes the normalized values of β_{C_i}.

The elements $dn_j \in [0,1]$ denotes the degree of support given by the FNC that pattern $\boldsymbol{y}[q]$, a transformed version of the pattern $\boldsymbol{x}[q]$, belongs to class C_j.

C. FNC Level Classification

A fusion unit is incorporated in each FNC which integrates the soft decision outputs of its parent FNCs at the previous layer. Based on the fuser outcomes a data splitting mechanism discriminates patterns into correctly classified and ambiguous ones. Data splitting allows focusing on those patterns for which adequate degree of confidence is not achieved yet, and improve their accuracy. This provides an effective data control, leading to computational savings for large data sets.

The data flow within each $FNC_k^{(\ell)}$ is described by the following stages (Fig. 2.):

Stage 1 (*Fusion*): The soft decision outputs derived from parent FNCs at layer $(\ell - 1)$ are fused using a fusion operator $\mathcal{F}_k^{(\ell)}$, such as min, fuzzy integral, decision templates or weighted average, [5]. The resulting decision vector is:

$$DF_k^{(\ell)}(\boldsymbol{x}) = \left[df_{k,1}^{(\ell)}(\boldsymbol{x}),\ df_{k,2}^{(\ell)}(\boldsymbol{x}), ..., df_{k,M}^{(\ell)}(\boldsymbol{x})\right]^T \qquad (7)$$

with $df_{k,r}^{(\ell)} \in [0,1]$, $r = 1,...,M$. The quality of parent classifiers is ascertained by comparing the $df_{k,r}^{(\ell)}$ values to a user defined threshold, $df_{k,r}^{(\ell)}(\boldsymbol{x}) \geq \vartheta$ (i.e. $\vartheta = 0.8$).

Stage 2 (*Data Splitting*): Based on the aforementioned inequality, data splitting ($DS_k^{(\ell)}$) is applied locally at each $FNC_k^{(\ell)}$, decomposing the data set D_N into two subsets: $J_k^{(\ell)}$,

Authorized licensed use limited to: UNIVERSITY OF WINDSOR. Downloaded on February 13,2010 at 21:07:06 EST from IEEE Xplore. Restrictions apply.

$V_k^{(\ell)}$ with $D_N = J_k^{(\ell)} \cup V_k^{(\ell)}$. $J_k^{(\ell)}$ includes patterns which are currently well classified with high grade of certainty, whereas $V_k^{(\ell)}$ contains patterns that are either misclassified or correctly classified with an inadequate level of confidence.

Stage 3 (*Confident patterns decisions*): Patterns $\boldsymbol{x} \in J_k^{(\ell)}$ are handled by the fuser $\mathcal{F}_k^{(\ell)}$ with their decision supports given by $DF_k^{(\ell)}(\boldsymbol{x})$.

Stage 4 (*Ambiguous patterns decisions*): Patterns $\boldsymbol{x} \in V_k^{(\ell)}$ are submitted to the local neuro-fuzzy classifier $\left\{FPD_k^{(\ell)}, DMFU_k^{(\ell)}\right\}$. Aggregate the parent FNC outputs through (1) to compute the input vector $\boldsymbol{x}_k^{(\ell)}$. Enter $\boldsymbol{x}_k^{(\ell)}$ to $FPD_k^{(\ell)}$ and perform steps **F1-F5** to obtain the transformed features $\boldsymbol{y}_k^{(\ell)} = FPD_k^{(\ell)}\{\boldsymbol{x}\}$. Input $\boldsymbol{y}_k^{(\ell)}$ to $DMFU_k^{(\ell)}$ and perform steps **D1-D2** to compute the decision output $DN_k^{(\ell)}(\boldsymbol{x})$.

Stage 5 (*Decision Aggregation)*: The overall decision output of $FNC_k^{(\ell)}$, denoted as $D_k^{(\ell)}(\boldsymbol{x}) = \left[D_{k,1}^{(\ell)}(\boldsymbol{x}),..,D_{k,M}^{(\ell)}(\boldsymbol{x})\right]^T$, is formulated by integrating the above two sources of evidence, $DF_k^{(\ell)}(\boldsymbol{x})$ and $DN_k^{(\ell)}(\boldsymbol{x})$.

The class each pattern belongs can be obtained using the maximum argument rule: $\max\left\{d_{k,j}^{(\ell)}(\boldsymbol{x})\right\}, j = 1,...,M$, where $d_{k,j}^{(\ell)}(\boldsymbol{x}) = df_{k,j}^{(\ell)}(\boldsymbol{x})$ if $\boldsymbol{x} \in J_k^{(\ell)}$ and $d_{k,j}^{(\ell)}(\boldsymbol{x}) = dn_{k,j}^{(\ell)}(\boldsymbol{x})$ if $\boldsymbol{x} \in V_k^{(\ell)}$.

IV. STRUCTURE LEARNING USING GMDHA

SONeFMUC model is generated in a self-organizing way by means of the GMDH algorithm [7]. For presentation purposes, we denote the original GMDH algorithm as GMDHa for the rest of the paper. Particularly, the structure of SONeFMUC is not fixed in advance. Starting from the original system inputs (features), new layers are sequentially developed, until a final topology is obtained, satisfying the performance requirements. Initially, we decompose data set into a training D_{tr}, a validation D_{vl}, and a testing D_{test} set, with $n_{tr} + n_{vl} + n_{test} = D_N$.

The GMDHa algorithm proceeds along the following steps:

G.1 *Construct the FNCs of the first layer*. Combining the original feature set $\boldsymbol{x} = \{x_1,...,x_m\}$ by $r^{(1)}$,

$$Q^{(1)} = \binom{m}{r^{(1)}} = m! / (m - r^{(1)})! r^{(1)}! \quad (8)$$

possible FNCs are created. A feature combination forms the input space of the corresponding FPD. Given a fuser type, construction of the FNCs involves previous outputs aggregation, FPD learning (**F1-F5**) and decision making through DMFU (**D1-D2**).

G.2 *Evaluate the performance of the FNCs at the current layer*. To assess the quality of the generated FNCs, each node is evaluated on the validation set, using an error measure:

$$E_{k,vl} = \frac{1}{n_{vl}} \sum_{j=1}^{p} \sum_{q=1}^{n_{vl}} \left\{ y_{d,j}^{C}[q] - y_{k,j}^{(\ell)} \right\}^2 + \sum_{q=1}^{n_{vl}} \left\{ C_d[q] \neq C_j[q] \right\} \quad (9)$$

G.3. *Formulate the best set at the current layer*. Sort the individual error measures in ascending order and retain a number of W FNCs with the best performance, whereas the rest are discarded. The design parameter W is set to 30 in our simulations. Outputs of these FNCs serve as inputs to the FPDs at the next layer.

G.4 *Construct the new FNCs*. Determine the structure of $FNC_k^{(\ell)}$, $k = 1,...,Q^{(\ell)}$ by combining its parent classifiers $FNC_i^{(\ell-1)}$ and $FNC_j^{(\ell-1)}$. FNCs at the previous layer are combined by two ($G = 2$), leading to a total number of $Q^{(\ell)} = \binom{W}{2} = W! / (W-2)! 2!$ nodes in the current layer. Combining parents FNCs means that we make use of both types of outputs being offered: the continuous outputs $\boldsymbol{y}_i^{(\ell-1)}$ and $\boldsymbol{y}_j^{(\ell-1)}$ (transformed feature values) and the decision vectors, $D_i^{(\ell-1)}$ and $D_j^{(\ell-1)}$. The former are first aggregated using (1) and fed as inputs to the $FPD_k^{(\ell)}$ while the latter ones are submitted to the fuser $F_k^{(\ell)}$.

G.5 The algorithm proceeds by repeating steps 2-4, until a termination criterion is fulfilled $E_*^{(\ell)} \geq E_*^{(\ell-1)}$, i.e., the best node performance attained at the current layer is inferior to the one at the previous layer, or when the number of layers created is larger than a predefined maximum number of layers, $L \geq L_{\max}$.

G.6 Once the stopping criteria are satisfied for some $L \leq L_{\max}$, the node classifier with the best performance $FNC_*^{(L)}$, is considered as the ending node of SONeFMUC, providing the decision outputs of the model. The remaining FNCs at the output layer are discarded. In the following, we perform a reverse flow tracing through the network's structure, moving from the output to the input layer. All nodes at the intermediate layers (the input layer included), having no contribution to the $FNC_*^{(L)}$ are removed from the network. As regards the model's inputs, a subset is retained from the original features including the most distinguishing features.

V. STRUCTURE LEARNING USING GMDHB

The effectiveness of the GMDHa in recognizing the structure of a SONeFMUC model has been proved through simulation results in our previous studies [5], [6]. However, according to steps 1-4, a large number of FNCs is needed to be created to formulate the best set in each layer. All the possible combinations of FNCs should be considered without

Authorized licensed use limited to: UNIVERSITY OF WINDSOR. Downloaded on February 13,2010 at 21:07:06 EST from IEEE Xplore. Restrictions apply.

incorporating any knowledge concerning the quality and complementary of the parent FNCs. If the number of W is large enough, then the number of produced FNCs is increased. The main drawback of using the GMDHa in constructing SONeFMUC models is that it doesn't take into account the classification capabilities of the combined FNCs of the previous layer in constructing the FNCs in the next layer.

The classification capabilities of the classifiers to be combined have a great impact to the outcome of their combination. In [13], the authors proved that in order the combination of two classifiers result in an increase in the classification accuracy, they should be different. Hence, the classifiers should produce different errors in classifying the patterns. If the classifiers produce the same errors, then the outcome will be the same regardless the fusion operator used in combining their decisions over the patterns. The authors propose the use of a simple statistical measure in order to identify the different classification capabilities of the classifiers to be combined.

Since the SONeFMUC structure is developed by combining generic neuro-fuzzy classifiers (FNCs) in order to produce another generic neuro-fuzzy classifier with enhanced classification capabilities in the next layer, it would be of great importance to gain prior knowledge concerning the outcome of the combination.

A. Proportion of Specific Agreement (Ps)

In order to asses the agreement on correct classification and agreement on error of two generic parent FNCs to be combined, we use Table I. According to Table I, we compare the outcome of the two FNCs to be combined given a set of labeled examples, and create a dichotomous table. In case the parent FNCs have few patterns in the off diagonal cells of the table, their combination will result in a new FNC with the same or slightly better classification accuracy than its parents. The reason is that these generic classifiers produce similar decision vectors over the patterns, and the fusion operator cannot discriminate between the two cases. In addition, the parent FNCs implement similar feature transformation and thus the combination of their output space in order to form the input space of the child FNC will not provide any new information regarding the distribution of the patterns. Hence, the outcome of the child FNC will be similar to that of its parents. In case the table contains a large number of patterns in the off diagonal cells, the combination of the output space of the parent FNCs will result in an input space for the child FNC with large diversity from the input space of its parents FNCs. Hence, the child FNC will have a greater potential of improving the classification accuracy.

TABLE I. 2X2 DICHOTOMOUS OUTCOME FOR TWO FNCS

Patterns correctly classified by both FNCs (a)	Patterns correctly classified by the first and incorrectly by the second (b).
Patterns incorrectly classified by the first and correctly by the second (c).	Patterns incorrectly classified by the both FNCs (d).

In order to measure the agreement on error classification we use the Proportion of Specific Agreement (Ps) [13], calculated as $Ps = 2d/(b+c+2d)$.

Using the values of Ps, we measure the potential of a combination of two parent FNCs. Ps can take values in the range of $[0,1]$. A value of Ps close to 1 indicates that the parent classifiers are committing the same errors, whereas a value close to zero indicates that the classifiers are strongly independent in classifying the patterns of the problem.

By incorporating the measure of Ps in order to gain previous knowledge for the classification capabilities of the child FNC, we can formulate a new structure learning algorithm, based on the principles of the GMDHa.

B. The Proposed Structure Learning Algorithm GMDHb

The basic idea behind the GMDHb algorithm is that it does not consider all the possible combinations of parent FNCs of the previous layer to create the child FNCs in the next layer. Instead, for each FNC in the previous layer, the algorithm searches for the most suitable to be combined among the other FNCs in the layer based on the value of Ps of their combination. Only this combination is considered while the others are being rejected. The same procedure is followed for all the FNCs in the layer until the population of the next layer is created.

The difference between GMDHa and the proposed GMDHb algorithm lies in the forth step of the algorithm where the FNCs of the next layer are created:

GB.4 *Construct the new FNCs*. Determine the structure of $FNC_k^{(\ell)}$, $k=1,...,Q'^{(\ell)}$ by combining its parent classifiers $FNC_i^{(\ell-1)}$ and $FNC_j^{(\ell-1)}$.

GB.4.1 For each $FNC_i^{(\ell-1)}$ calculate the value $Ps_{i,j}^{(\ell-1)}$ with each of the rest of (W-1) of the set $BS^{(\ell-1)}$ starting from $i=1$ using

$$Ps_{i.j}^{(\ell-1)} = (1-a)Ps_{trn,i,j}^{(\ell-1)} + aPs_{val,i,j}^{(\ell-1)}, \tag{10}$$

where $i=1,\dots,W-1$, $j=i,\dots,W$. The value $Ps_{trn,i,j}^{(\ell-1)}$ indicates the agreement on error between $FNC_i^{(\ell-1)}$ and $FNC_j^{(\ell-1)}$ on the training examples, whereas the value $Ps_{val,i,j}^{(\ell-1)}$ measures the agreement on error for the validation examples. Parameter a ($a = 0.5$ in this paper) controls the balance between the training and the generalization capabilities of the generic neuro-fuzzy classifier.

GB.4.2 For $i=1,\dots,W$, the combination with the smallest value of $Ps_{i,j}^{(\ell-1)}$ is selected as a combination of members of the population $P^{(\ell)}$, whereas the rest are rejected. In case the combination with the smallest value of $Ps_{i,j}^{(\ell-1)}$ has been already selected, the next smaller one is selected.

GB.4.3 Following steps **GB.4.1** and **GB.4.2** for each $FNC_i^{(\ell-1)}$ of $BS^{(\ell-1)}$, the $FNC_j^{(\ell-1)}$ with the higher potential their combination to result in a child FNC with greater classification capabilities is selected. Hence, the created population is of size $Q'^{(\ell)} = W$. For each selected

Authorized licensed use limited to: UNIVERSITY OF WINDSOR. Downloaded on February 13,2010 at 21:07:06 EST from IEEE Xplore. Restrictions apply.

combination, we construct the child $FNC_k^{(\ell)}$ by aggregating previous outputs, applying FPD learning (**F1-F5**) and decision making through DMFU (**D1-D2**).

Comparing the number of created FNCs in each layer, we can observe the large decrease in this number using GMDHb, since $W < \binom{W}{2} \Rightarrow Q'^{(\ell)} < Q^{(\ell)}$. For example, if we consider $W = 30$, then $Q^{(\ell)} = 435$ and $Q'^{(\ell)} = 30$. The decrease in the size of the population is obvious.

VI. Simulation Results

The forensic glass is a benchmark problem for comparing classification methods, [14]. It is a six-class problem with a feature space of nine dimensions and a total number of 214 available patterns. For cross-validation, we generated ten random splits of the data into a training (75 samples, 35%), a validation (32 samples, 15%), and a checking (107 samples, 50%) data set.

We proceed with the selection of the structural parameters such as the type of the consequent of the rules, the number of the fuzzy sets (2 or 3 in this paper, different for the first and the higher layers) and the type of the fuser (min, fuzzy integral, decision templates and weighted average). To this end, we follow the procedure described in more detail in [5], with the results shown in Table II. We end up using crisp consequent for the rules, two sets for the inputs of the FPD units and the min fusion operator.

TABLE II. Simulation Results Of SONeFMUC Using GMDHa And GMDHb For The Forensic Glass Problem

S. L. Alg.	Layers	C.T.	F.S.	F.T.	Time (sec)	Check. Perf. (%)
GMDHa	3.9	Crisp	(3,3)	MIN	9.40	62.06
GMDHb	4.8	Crisp	(2,2)	MIN	0.73	63.46

S.L.:Structure Learning, C.T.:Consequent Type, F.S.:Fuzzy Sets, F.T.:Fusion Type

In Table II, simulation results using the GMDHa taken from [5] are also shown. As we can observe, the GMDHb algorithm developed SONeFMUC models in only 0.73 sec on average comparing to the 9.40 sec of the original GMDHa. In addition, the classification accuracy on the checking patterns using GMDHb is 63.46% on average, slightly higher than that of the GMDHa algorithm (62.06%). Comparing the result of the proposed method with other methods in the literature taken from [15] (Table III), we conclude that the proposed structure method outperforms other methods used.

TABLE III. Comparison Of SONeFMUC Using GMDHa And GMDHb With Other Methods For The Forensic Glass Problem

Classifier	Check. Perf. (%)	Classifier	Check. Perf. (%)
GFC	61.96	NN	59.72
SVM	51.00	SONeFMUC - GMDHa	62.06
LDA	60.10	SONeFMUC - GMDHb	63.46

VII. Conclusions

In this paper, a novel efficient structure learning algorithm for a self-organizing multilayered classifier is proposed. The GMDHb algorithm is a modified version of the traditional GMDH algorithm, with the aim to reduce the computational load or the original algorithm. The GMDHb algorithm uses a simple statistical measure to gain prior knowledge concerning the classification capabilities of the child FNC to be generated. Only useful FNCs are created in each layer reducing thus the total computational time of the structure learning process. Simulation results on the forensic glass problem indicated that the proposed structure learning algorithm is capable of a great reduce in the computational time while increasing the classification accuracy of the SONeFMUC model.

References

[1] D. Nauck and R. Kruse, "A neuro–fuzzy method to learn fuzzy classification rules from data," *Fuzzy Sets and Systems*, vol. 89, no. 3, pp. 277–288, Aug. 1997.

[2] L. Y. Cai and H. K. Kwan, "Fuzzy classifications using fuzzy inference networks," *IEEE Trans. Systems, Man and Cybernetics, Part: B*, vol. 28, no. 3, pp. 334–347, June 1998.

[3] J.–S. Wang and C.S.G. Lee, "Self–adaptive neuro–fuzzy inference systems for classification applications," *IEEE Trans. on Fuzzy Systems*, vol. 10, no. 6, pp. 790–802, Dec. 2002.

[4] L. Rutkowski and K. Cpalka, "Flexible neuro–fuzzy systems," *IEEE Trans. Neural Networks*, vol. 14, no. 3, pp. 554–574, May 2003.

[5] N. E. Mitrakis, J. B. Theocharis and V. Petridis, "A multilayered neuro-fuzzy classifier with self-organizing properties," *Fuzzy Sets and Systems*, vol. 159, no. 23, pp. 3132-3159, Dec. 2008.

[6] N. E. Mitrakis, C. A. Topaloglou, T. K. Alexandridis, J. B. Theocharis and G. C. Zalidis, "Decision fusion of GA self-organizing neuro-fuzzy multilayered classifiers for land cover classification using textural and spectral features," *IEEE Trans. Geoscience and Remote Sensing*, vol. 46, no. 7, pp. 2137-2152, July 2008.

[7] A. G. Ivahnenko, "The group method of data handling: A rival of method of stochastic approximation," *Soviet Automatic Control*, vol. 13, no. 3, pp. 43–55, 1968.

[8] S.–K. Oh, W. Pedrycz, and B.–J. Park, "Polynomial neural networks architecture: Analysis and design," *Computers and Electrical Engineering*, vol. 29, no. 6, pp. 703–725, 2003.

[9] T. Kondo, "GMDH neural networks using the heuristics self–organization method and its application to the pattern identification problem," in *Proceedings of the 37th Annual Conference SICE*, Chiba, Japan, 29–31 July, 1998, pp. 1143–1148.

[10] E. Delivopoulos and J. B. Theocharis, "A modified PNN algorithm with optimal PD modeling using the orthogonal least squares method," *Information Sciences*, vol. 168, no. 1–4, Dec. 2004.

[11] T. Takagi and M. Sugeno, "Fuzzy identifications and its application to modeling and control," *IEEE Trans. Systems, Man and Cybernetics*, vol. 15, 116–132, Jan.-Feb.1985.

[12] H. M. Lee, C. M. Chen, J. M. Chen, and Y. L. Jou, "An efficient fuzzy classifier with feature selection based on fuzzy entropy," *IEEE Trans. Systems, Man and Cybernetics, Part: B*, vol. 31, no. 3, pp. 426–432, June 2001.

[13] M. Petrakos, J. A. Benediktsson, and I. Kanellopoulos, "The effect of classifier agreement on the accuracy of the combined classifier in decision level fusion," *IEEE Trans. Geoscience and Remote Sensing*, vol. 39, pp. 2539–2546, Nov. 2001.

[14] P.M. Murphy, D.W. Aha, UCI repository of machine learning databases, Univ. California, Dept. Info. Comput. Sci., Irvine, CA, http://www.ics.uci.edu/-mlearn/MLRepository.htm, 1998.

[15] F. Hoffmann, "Combining boosting and evolutionary algorithms for learning of fuzzy classification rules," *Fuzzy Sets and Systems*, vol. 141, vol. 1, pp. 47–58, Jan. 2004.

Authorized licensed use limited to: UNIVERSITY OF WINDSOR. Downloaded on February 13,2010 at 21:07:06 EST from IEEE Xplore. Restrictions apply.