Fifth IFSA World Congress (1993), 985-988

An alternative architecture for application-driven fuzzy systems

W. Pedrycz
Dept. of Electrical & Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada, R3T2N2
email: pedrycz@eeserv.ee.umanitoba.ca

J. Valente de Oliveira
INESC - Control of Dynamic Systems Group
Apartado 10105, 1017 Lisboa Codex, Portugal.
email: jlvo@eniac.inesc.pt

Abstract: **An alternative approach to the design of application-driven fuzzy systems is proposed. A broad class of fuzzy systems applications requires a certain fuzzy partition of the input space while it demands for simple numerical quantities. For this class, a dedicated fuzzy system archictecture is presented and a design strategy is proposed. Both the single-input/single-output and multi-input/multi-output cases are considered. Numerical analysis are complete illustrating several aspects of the proposed framework.**

1 Introduction

A broad class of applications driven problems expressed with the aid of fuzzy sets formalisms lends itself into approximation tasks. These tasks require a certain fuzzy partion of the involved (input/output) spaces. Some other tasks require only the fuzzy partion of the input space (space of inputs) while simple numerical quantities are enough at the outputs of the system.

Considering, for instance, fuzzy controllers on can look at the inputs (say error and changes in error) as being transformed via their fuzzy partitions and utilized to infer (nonfuzzy) numerical values of control. Bearing this in mind one could conclude that the use of fuzzy partitions for the input spaces is fully legitime (since they do reflect the way how humans perceive the state of the object under control) whereas the control actions (even realized by a human being) are purely numerical. As a consequence of this fact, it becomes difficult to imagine the way in which the fuzzy partition of the output space (space of control actions) can be worked out as a useful concept.

This argument should be reflected in the overall system architecture. This idea is exploited here starting with a single-input/single-output systems and then generalizing the involved concepts for the multi-input/multi-output cases.

2 An architecture of single-input/single-output systems

The overall topology of the single-input/single-output fuzzy system that reflects its functions is visualized in Fig. 1. It consists of a numerical/linguistic (N/L) interface and a linguistic system. The N/L interface translates numerical data into linguistic data (fuzzy sets). The linguistic system is constructed with the use of k AND neurons (logic units) followed by a single averaging functional element. The AND neurons are logic-oriented processing elements combining fuzzy input signals via an extended AND operation [5]. The averaging functional element combines these activation levels with the prototypes (centroids) distributed in the output space to produce an output value. The following subsections will elaborate in more detail on several main aspects of this architecture.

2.1 The N/L interface

The function of the N/L interface is to "code" any value of the input variable of the system in sense of a finite family of linguistic labels (fuzzy sets) being accepted *a priori* as useful in handling the available data by the considered system. This coding can be achieved in many different ways. The use of possibility and necessity measures might be of a particular interest. Note also that for a numerical datum these two measures coincide.

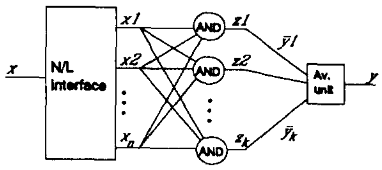

Figure 1: The proposed single-input/single-output architecture.

Simultaneously, the interface can be seen as a pre-processing block of the proposed fuzzy system. From information-processing point of view one can put forward several postulates such as: (i) maintaining the semantic integrity of the linguistic terms; (ii) contributing for improving the overall system capabilities, and (iii) keeping as low as possible its computational requirements.

For the construction of a N/L interface one can assume r reference fuzzy sets (cf. [2]). Here, the i-th reference set ($i = 1, \ldots, r$) is characterized by a normal and convex membership function with two parameters representing its center and aperture.

In this work, two types of membership functions will be investigated: (piecewise) linear and nonlinear membership functions. The input interface can be optimized by considering equivalences of the input information processed by a series of the N/L and L/N interface. One can assume that the Numerical/Linguistic interface is *directly* connected to an *hypothetic* Linguistic/Numerical interface (produced by the center of gravity method). The parameters of the membership functions are then determined to achieve a (nearly) zero conversion error subject to some integrity constraints. The resulting N/L interface is called an optimal interface [7].

Using linear membership functions, namely triangular membership functions, an optimal interface is achieved if consecutive membership functions are overlapping at a level of 0.5. On one hand, this kind of interface does not require any computational effort. On the other, it might not reflect well the nature (distribution) of the input numerical data. The contribution of this interface towards enhancements of the overall system capabilities could be also fairly limited.

As alternative to the triangular membership functions one can think on nonlinear membership functions such as gaussian or exponential ones. For those functions one has to apply numerical optimization techniques in order to find an optimal interface. The PAFIO algorithm has proposed to come up with the optimal interface [7].

2.2 The linguistic system

The linguistic system can be viewed as a network whose first layer consists of k AND neurons. Consider $X = [x_1 \ldots x_i \ldots x_r]$ as the input vector of this layer, x_i being the possibility measure of the numerical input of the overall system taken with respect to the i-th reference fuzzy set. We will also admit complemented values of the inputs considering vector X embracing both x_i as well as their complements $\bar{x}_i$ ($\bar{x}_i = 1 - x_i$). The expression describing the j-th neuron is equal to:

$$z_j = \mathrm{AND}(X, W_j); \qquad j = 1, \ldots, k \tag{1}$$

where W_j summarizes the connections of the neuron. The AND neuron forms a global AND type aggregation of its input signals. The implementation involving triangular operators yields

$$z_j = \mathrm{T}_{i=1}^{r}(x_i \text{ s } w_{ij}) \tag{2}$$

The symbols "T" and "s" represent triangular norm and triangular co-norm, respectively (cf. [2]). The individual connections modulate the impact that the individual inputs (i.e. $x_1 \ldots x_i \ldots x_r$) have on the output of the neuron. Observe that the k neurons put together realize a fuzzy relation equation under t-s composition.

Subsequently the values z_j are viewed as activation levels of the prototypes (centroids) distributed in the output space and coded as the connections between the hidden and the output layer. More specifically, for the output node one has:

$$y = \frac{z_1\bar{y}_1 + z_2\bar{y}_2 + \cdots + z_k\bar{y}_k}{z_1 + z_2 + \cdots + z_k} \tag{3}$$

2.3 Learning in the network

The learning activity is of parametric character and includes modifications in all the values of the connections between the layers.

Let the performance index be given as a standard sum of squared errors, say

$$Q = \frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2$$

where t_i stands for the i-th target value at the output while y_i denotes the actual output of the network. N is the number of samples of an available learning set. In general, the updates of each connection (say $w_{(.)}$) are proportional to the corresponding gradient,

$$\Delta w_{(.)} = -\eta_{(.)}\frac{\partial Q}{\partial w_{(.)}}$$

with $\eta_{(.)}$ being the learning rate parameter associated with a specific connection. Considering independent learning rate parameters allows us to introduce a very simple, yet efficient mechanism aiming at improving the convergence rate of the learning procedure. This mechanisms uses the signs of the successive derivatives of the performance index to adjust the learning speed of each $w_{(.)}$.

2.4 Illustrative example

Let us discuss the task of modelling the static nonlinearity: $t(x) = \frac{1}{1+e^{-x}}$. Furthermore, consider that a non-uniformly distributed data set is available for learning purpose. The data set consists of $N = 50$ pairs, $\{(x_i, t_i)|i = 1, \ldots, 50\}$, and is presented in Fig. 2. At the level of the N/L interface triangular and gaussian membership functions are tested, resulting in

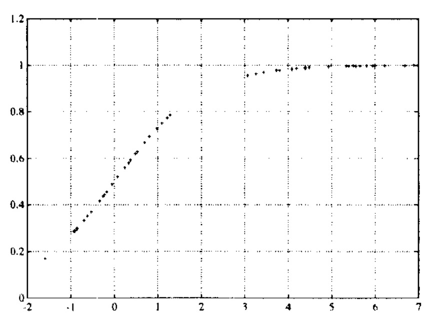

Figure 2: The 50 pairs of the data set.

T and G system, respectively. Three linguistic terms were assumed for both systems. For the realization of the AND neurons, the t-norm has been defined as the product and the s-norm has been taken as the probabilistic sum. For assessing the performance of the trained systems, a sequence of testing inputs has been applyed to both of them. The results of this cross-validation are presented in Fig. 3. This Figure shows a good performance for system G while a relatively week performance can be observed for the system T. The increased number of the AND neurons will not be able to improve this performance. Similar comparative performance have been visualized when the number of reference fuzzy sets has been increased. Fig. 4 illustrate this scenario in which we consider now five linguistic terms at the interface level.

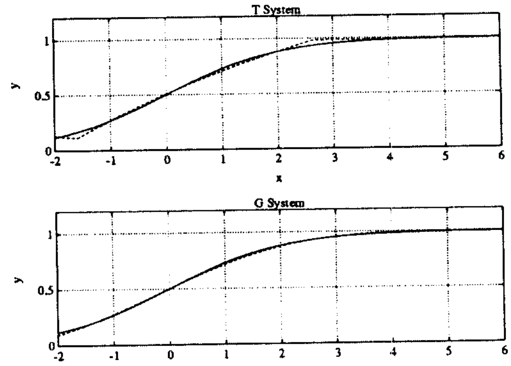

Figure 3: Cross validation for systems with 3 reference fuzzy sets.

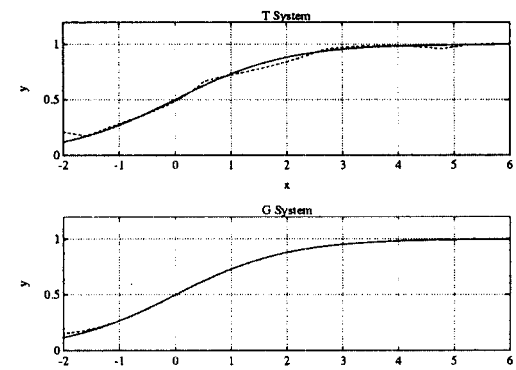

Figure 4: Cross validation for systems with 5 reference fuzzy sets.

3 A general multidimensional architecture

In this section a generalization of the presented fuzzy system is provided, that is, the multi-input/multi-output case is adressed. The layout of this general architecture is visualized in Fig. 5.

In what concerns the linguistic system one can aggregate the n linguistic inputs into a single one, say $X = [X_1|X_2|\ldots|X_n]$ and still use the linguistic system proposed for the single input case. The network will learn just like conventional neural networks do, however the linguistic meaning of the structure it will be lost. Thus there is a semantic need to generalize the above mentioned AND neurons.

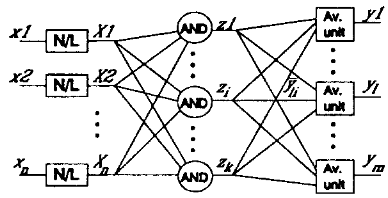

Figure 5: The layout of a general architecture.

The n inputs of the (extended) AND neurons are $X_1, \ldots, X_n$, where as before $X_i \in [0,1]^{r_i}$. Again, each X_i can be augmented by considering the complements of its elements. The expression describing the j-th neuron is equal to:

$$z_j = \text{AND}(X_1, X_2, \ldots, X_n, W_j); \qquad j = 1 \ldots k \tag{4}$$

and the implementation involving t-operators yielding:

$$z_j = \mathrm{T}^{r_1}_{j_1} \cdots \mathrm{T}^{r_n}_{j_n}(x_{1j_1} \cdots \text{ s } x_{nj_n} \text{ s } w_{j_1,\ldots,j_n,j}) \tag{5}$$

With this generalization the semantic meaning of the AND neuron is preserved. Namely, the individual connection, $w_{j_1,\ldots,j_n,j}$, still modulate the impact that inputs, $X_1, \ldots, X_i, \ldots, X_n$, have on the output of the neuron, and the layer consisting of the k extended AND neurons can be understood as an extended version of the fuzzy relation equation (cf. [6]). The importance of such structures is augmented even further since there is no analytical solution for fuzzy relation equations under t-s composition [1, 3].

The parametric learning process is now driven by the performance index involving m outputs:

$$Q = \frac{1}{mN} \sum_{j=1}^{m} \sum_{k=1}^{N} (t_{kj} - y_{kj})^2$$

4 Concluding remarks

An alternative architecture for application-driven fuzzy systems was presented.

The problem of the construction of Numerical/Linguistic interfaces was addressed. The effect that the shape of membership functions representing the reference fuzzy sets has on the overall system performance has been emphasized. It is argued and then tested throught experimental considerations that nonlinear membership functions can enhance the system performance.

AND neurons followed by averaging functional units were used as the linguistic system. An appropriate generalization of the AND neurons was provided to accomodate the multi-input case. The layer of the AND neurons can be visualized as a fuzzy relation equation (or an extented version of the fuzzy relation equation) under s-t composition. This is a important aspect since there is no analytical solution for such equations.

References

[1] A. Di Nola, S. Sessa, and W. Pedrycz, "Fuzzy relation equations theory as a basis of fuzzy modelling: An overview", *Fuzzy Sets and Systems*, **40**, 1991, pp. 415-429.

[2] W. Pedrycz, *Fuzzy Control and Fuzzy Systems*, (Research Studies Press/J. Wiley & Sons, Chichester), 1989.

[3] W. Pedrycz, "Processing in relational structures: Fuzzy relational equations", *Fuzzy Sets and Systems*, **40**, 1991, pp. 77-106.

[4] W. Pedrycz, "Fuzzy neural networks with reference neurons as pattern classifiers", *IEEE Trans. Neural Networks*, 1992.

[5] W. Pedrycz and J. Valente de Oliveira, "Optimization of fuzzy relational models", *Proc. of 5-th IFSA World Congress*, Korea, 1993.

[6] J. Valente de Oliveira, "Neuron inspired learning rules for fuzzy relational structures" *Fuzzy Sets and Systems* (in press).

[7] J. Valente de Oliveira, "On optimal fuzzy systems I/O interfaces", *Proc. of the Second IEEE International Conference on Fuzzy Systems*, San Francisco, 1993.