ICICIC, September 5-7, 2007, Kumamoto, Japan

Fuzzy Neural Tree for Knowledge Driven Design

Ö. Ciftcioglu, M. S. Bittermann and I. S. Sariyildiz
Faculty of Architecture, Delft University of Technology, The Netherlands
o.ciftcioglu@tudelft.nl

Abstract

A neural tree structure is considered with nodes of neuronal type, which is a Gaussian function playing the role of membership function. The total tree structure effectively works as a fuzzy logic model with inputs and outputs. In this model the locations of the fuzzy membership functions are normalized to unity so that the system has several desirable features and it represents a fuzzy model maintaining the transparency and effectiveness while dealing with complexity. The research is described in detail and its outstanding merits are pointed out in a framework having transparent fuzzy modelling properties and addressing complexity issues at the same time. A demonstrative application exercise of the model is presented and the favourable performance is demonstrated.

1. Introduction

The potentials of neural tree for structuring information is combined with the reasoning process of fuzzy logic to obtain a special type of neural tree which is transparent as well as able to deal with complexity. The limitations of a fuzzy logic system in a complex environment are substantially circumvented by integrating the domain knowledge into the tree structure and determining the fuzzy membership functions accordingly.

2. Neural tree models

A neural tree [1-3], is composed of terminal nodes, non-terminal nodes, and weights of connection links between two nodes. The non-terminal nodes represent neural units and the neuron type is an element introducing a non-linearity simulating a neuronal activity. In the present case, this element is a Gaussian function which has several desirable features for the goals of the present study; namely, it is a radial basis function ensuring a solution and the smoothness. At the same time it plays the role of membership function in the tree structure which is considered to be a fuzzy logic system as its outcome is based on fuzzy logic operations and thereby associated reasoning. An instance of a neural tree is shown in figure 1.

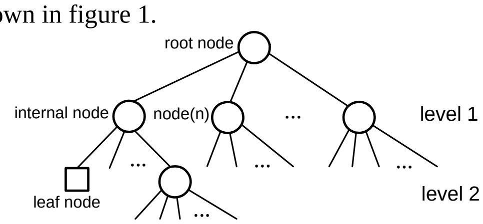

Fig. 1. The structure of a neural tree

Each terminal node, also called *leaf*, is labelled with an element from the terminal set $T=\{x_1, x_2, \ldots, x_n\}$, where x_i is the *i*-th component of the external input $\mathbf{x}$ which is a vector**.** Each link *(j,i)* represents a directed connection from node *j* to node *i*. A value w_{ij} is associated with each link. In a neural tree, the root node is an output unit and the terminal nodes are input units. The node outputs are computed in the same way as computed in a feed-forward neural network. In particular, in the present work the nodes are similar to those used in a radial basis functions network with the Gaussian basis functions. In this way, neural trees can represent a broad class of feed-forward networks that have irregular connectivity and non-strictly layered structures.

3. Neural tree as underlying structure domain knowledge

In the neural tree considered in this work each of the non-terminal nodes consists of a Gaussian radial basis function. The output for this node is given by

$$f(x) = w_j \phi(\| x - c_j \|) \qquad (1)$$

where $\mathbf{x}$ *is input vector;* $\phi(.)$ is the Gaussian basis function, c_j is the centre of the basis function at the j-th node. The Gaussian is of particular interest and used in this research due to its relevance to fuzzy-logic.

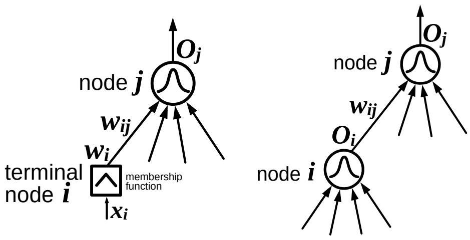

Fig. 2. The detailed structure of a neural tree with respect to different type of node connections.

Referring to figure 2, the centres of the basis functions are the same as the input weights w_{ij} to that node. For a node connected to a preceding terminal node, we can write

$$O_j = \exp(-\frac{1}{2}\sum_i^n \left[\frac{(w_i - 1)}{\sigma_j / w_{ij}}\right]^2) \qquad (2)$$

and for a node connected to a preceding non-terminal node, we can write

$$O_j = \exp(-\frac{1}{2}\sum_i^n \left[\frac{(O_i - 1)}{\sigma_j / w_{ij}}\right]^2) \qquad (3)$$

Above, σ_j is the width of the basis function and it is used to measure the uncertainty associated with the node inputs designated as external input; w_{ij} is the weight connecting a node to another node forward; j is the layer number $(j=1,2, ..)$; i denotes the i-th input to that node. Note that, in (2) and (3), the centre of the basis function is a vector $\{1, 1, 1,\ldots,1\}$, that is $c_i=1$ in (1). This implies that the Gaussian which plays the role of fuzzy membership function has its maximum value at the point w_i=1 or O_i=1, indicating that if the inputs x_i at the non-terminal nodes are transformed to between zero and unity, represented by w_i, than the universe of discourse of Gaussian fuzzy membership functions extend also to unity; this is illustrated in figure 3.

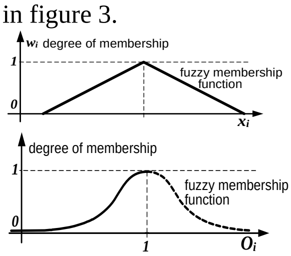

Fig. 3. Fuzzy membership functions at terminal and non-terminal nodes

By means of the above described approach the following limitations encountered in a general fuzzy logic modelling are eliminated.

- The location of the membership functions of a non-terminal node is always located at the point $c_i=1$.

- Although the type of the fuzzy membership function is determined in advance as Gaussian, its shape, i.e., the width, is determined by learning using the domain knowledge rather than choosing some arbitrary width.

- The number of fuzzy membership functions for a node is the same as the number of input weights w_i to that node. In other words, the number of fuzzy membership functions for a node is only one, and it is a multidimensional membership function. A multidimensional membership function can be decomposed into single-dimensional membership functions the number of which is equal to the number of inputs to that node.

- The curse of dimensionality is circumvented since the radial basis function centre of each node is determined in the form of multidimensional membership function separately without recourse to other nodes, with respect to their centres.

- With the increasing value of the inputs at the terminal nodes, the output at the root node increases as well. In the fuzzy logic terminology, approaching to the maximum of the fuzzy membership function at the input is reflected to the output of the model, following the same trend. Based on this, the widths of the Gaussians are determined by means of learning making use of this tacit knowledge embedded in the domain knowledge. This is a consistency or boundary condition peculiar to the application. In general, we should consider such consistency or boundary condition which may be specific to that application. In the formulation of the modelling the domain knowledge, the system determinants selected should be carefully verified in advance to identify such intrinsic consistency requirements to be fulfilled in the model.

4. Implementation of the Model

For the implementation of the novel neural tree structure presented by this research, a knowledge-based fuzzy model is developed and implemented where the examples being reported here are selected from architecture and building technology. In the architectural exercise the design performance of a scene is considered. The domain knowledge possessed by architect directed the tree structure seen in figure 4.

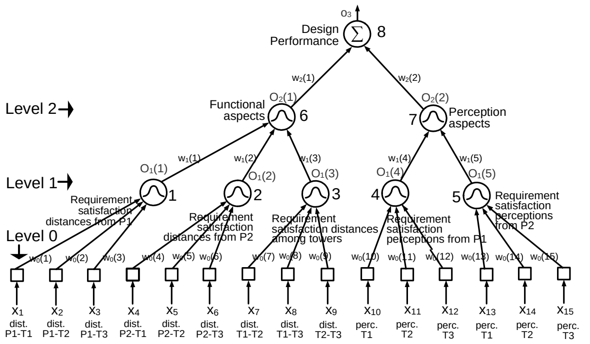

Fig. 4. Neural tree structure for design performance assessment.

The design performance is determined by two sub-domains, namely its *functional aspects* and *perception aspects* at one level further below from the root node. At one level further below we identify five sub-domains, namely, *1-Requirement satisfaction of distances from the perceiver P1; 2-Requirement satisfaction of distances from the perceiver P2; 3-Requirement satisfaction of distances among the towers; 4-Requirement satisfaction of perceptions by the perceiver P1*; *5-Requirement satisfaction of perceptions by the perceiver P2.* At the terminal level, the determinants of the design performance take place which are given in Table 2. For the structure established above, at each level the weights assessed are given in Table 3. These are the connection weights assessed to assign to the neural tree model for a scene shown in figure 5. Each input x_i at the *leaf level* indicates its graded assessment with respect to design performance. They are given in Table 4. With these inputs, the assessment of the design performance of the scene is given in Table 5.

Table 2. Determinants of the design performance.

Demand satisfaction distances from P1	***Demand satisfaction distances from P2***	***Demand satisfaction distances among towers***	***Demand satisfaction perception from P1***	***Demand satisfaction perception from P2***
Distance between *P1* and *T1*	Distance between *P2* and *T1*	Distance between *T1* and *T2*	Perception of *T1* from *P1*	Perception of *T1* from *P2*
Distance between *P1* and *T2*	Distance between *P2* and *T2*	Distance between *T1* and *T3*	Perception of *T2* from *P1*	Perception of *T2* from *P2*
Distance between *P1* and *T3*	Distance between *P2* and *T3*	Distance between *T2* and *T3*	Perception of *T3* from *P1*	Perception of *T3* from *P2*

Table 3. Levelwise weights of the neural tree

L1	.40	.60													
L2	.30	.20	.50	.60	.40										
L3	.25	.30	.45	.40	.30	.30	.70	.20	.10	.50	.25	.25	.20	.30	.50

Table 4. Input composition; initial

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
.86	.90	.99	.27	.40	.97	.98	.85	.58	.34	.37	.86	.73	.38	.99

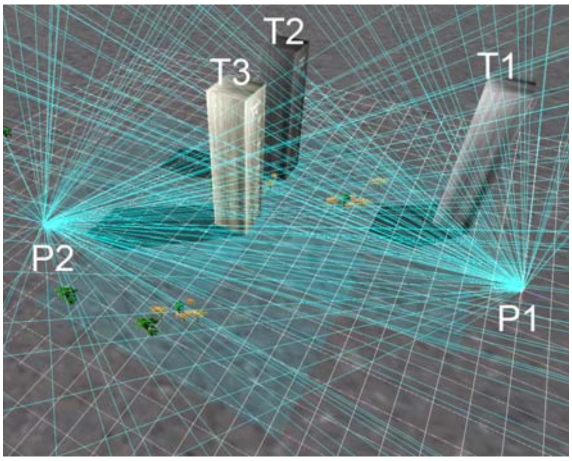

Fig. 5. Design based on performance; initial object locations

Table 5. Design performance of initial design configuration

	1	2	3	4	5
Design Performance	.36				
Functional and perception aspects (columns 1 and 2)	.20	.48			
Requirement satisfaction distances: from P1, from P2, and among towers; requirements satisfaction perceptions: from P1 and P2 (columns 4 and 5)	.99	.51	.90	.49	.85

The initial design performance results seen in Table 5 are subject to examination by the architect constructing the scene, while it is subject to improvement by changing the input composition in appropriate way to maximize the design performance. This is accomplished by genetic search. The output at the root node, which quantifies the design performance, is used as the representation of the fitness of the respective chromosome. In this way the genetic algorithm makes use of the knowledge embedded in the neural tree during its search for obtaining maximal performance. The design obtained after genetic search is shown in Fig. 6. The performance improvement scores from 0.36 given in Table 5 to 0.94 given in Table 7, with the corresponding input compositions are given in Tables 4 and 6, respectively.

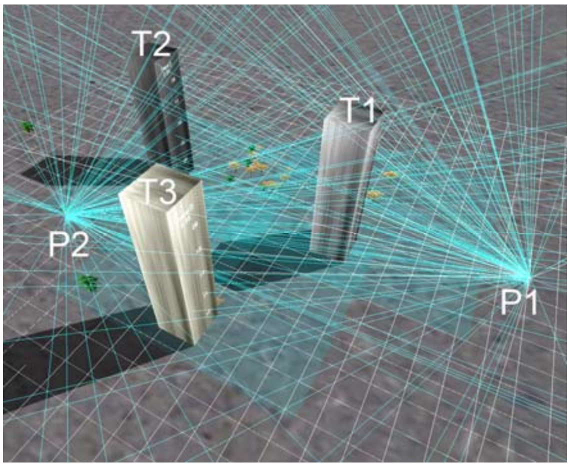

Fig. 6. Design based on performance; final object locations

Evolutionary search has unique favourable merits and performance for this search mission; namely the search is carried out deliberately in a discrete space not to be trapped in local optima during the search while it is highly constrained by design performance demands. By means of the genetic algorithm (GA), next to performance-related demands the input space can be searched to obtain a suitable input composition satisfying certain conditionality imposed on the inner aspects that belong to the inner nodes. For the input given in Table 6, the design performance results from neural tree are shown in Table 7.

Table 6. Input composition; final

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
.97	.97	.99	.89	.98	1.0	.98	.83	.27	.84	.96	.80	.70	.93	.98

Table 7. Design performance of final design configuration

	1	2	3	4	5
Design Performance	.94				
Functional and perception aspects (columns 1 and 2)	.87	.99			
Requirement satisfaction distances: from P1, from P2, and among towers; requirements satisfaction perceptions: from P1 and P2 (columns 4 and 5)	1.0	.99	.93	.96	.98

5. Discussion and conclusions

The research describes knowledge driven fuzzy modelling where the model has neural tree structure. It has another novelty feature as it may integrate analytical hierarchy process [4] for knowledge transfer in the modelling process thereby providing efficiency in knowledge representation in a complex modelling task. The model is finally determined by the integration of the consistency of the knowledge into it by stipulating the consistency onto the widths of the Gaussians through learning. It is noteworthy to mention, that the Gaussian nodes of the neural tree correspond to fuzzy logic rules so that the outcome of the model is result of a number of logic operations and finally de-fuzzification at the root node. The equivalence between neural networks and fuzzy logic for Gaussian fuzzy membership functions is known in the literature. The neural tree with fuzzy logic presented in this research forms a fuzzy model especially as described by Murray [5], where some strict conditions stipulated on the equivalency are relaxed. A demonstrative architectural design application exercise is reported indicating the suitability of the work for a wide range of similar applications of technological, industrial and practical interest. The work is described in detail with design illustrations and its outstanding merits are pointed out in a framework having transparent fuzzy modelling properties and addressing complexity issues at the same time.

6. References

[1] A. Sankar and R. J. Mammone, "Neural tree networks," in *Neural Networks:Theory and Applications,* R.J. Mammone and Y. Zeevi, Eds, New York: Academic, 1991, pp.281 –302.

[2] J. A. Sirat and J. P. Nadal, "Neural tree networks," *Network*, Vol.1, pp.423-438, 1990.

[3] H. Guo and S.B. Gelfand, "Classification Trees with neural Network.feature extraction," *IEEE Trans. Neural Networks*, vol.3, no 6, pp. 923-933, 1992

[4] T. L. Saaty, *The Analytic Hierarchy Process*, McGraw-Hill, New York, 1980.

[5] K. J. Hunt, R. Haas and R. Murray-Smith, "Extending the functional equivalence of radial basis function networks and fuzzy inference systems", *IEEE Trans. Neural Networks*, vol. 7, no. 3, May 1996.