A Fuzzy Neural Tree Based on Likelihood

Özer Ciftcioglu, Senior Member, IEEE
Department of Architecture
Delft University of Technology | Maltepe University
Delft, The Netherlands | Maltepe - Istanbul, Turkey
o.ciftcioglu@tudelft.nl | ozerciftcioglu@maltepe.edu.tr

Abstract—A novel fuzzy-neural tree (FNT) is presented. Each tree node uses a Gaussian as a fuzzy membership function so that the approach uniquely is in align with both the probabilistic and possibilistic interpretations of fuzzy membership thereby presenting a novel type of network. The tree is structured by the domain knowledge and parameterized by likelihood. The FNT is described in detail pointing out its various potential utilizations demanding complex modeling and multi-objective optimization therein. One of such utilizations concerns design. This is exemplified and its effectiveness is demonstrated by computer experiments in the realm of Architectural design.

Keywords—Fuzzy logic; neural tree; knowledge modeling; evolutionary computation; likelihood; probability possibility

I. INTRODUCTION

Neural and neuro-fuzzy computation received much attention in literature for several decades, and their common features are well identified. However, in the existing works, the role of neural or neuro-fuzzy system is to form a model between input-output data set pairs that are available from sensor measurements. Therefore the utilization of neural networks and neuro-fuzzy systems has been mainly in the engineering domain for diverse applications. Due to this fact, utilization of neuro-fuzzy systems in soft sciences has relatively remained to be marginal. Because of the curse of dimensionality, fuzzy logic is restricted to modeling systems with low complexity. On the other hand, neural network can have possibility to deal with complex systems. Based on this view, we can understand that neuro-fuzzy systems are especially suitable for modeling systems somewhere between low and high of complexity. This interesting phenomenon occurs perhaps due to the transparency of fuzzy logic against the black-box character of neural-network modeling or computation. As to expert knowledge processing that happens in expert systems, one needs transparency. However one needs also capability to deal with the expert knowledge complexity. Emphasizing this dilemma as to knowledge modeling, one should note that fuzzy logic applications are praised as transparent, knowledge-driven solutions, yet they do not handle complexity. On the other hand neural network solutions are praised for their learning capability, although they operate with data-driven strategy as a black-box. Conventionally depending on their strength in applications such systems are called neuro-fuzzy systems or vice versa. Fuzzy neural computation is a matter of convenient utilization of neural computation, where fuzzy logic takes place as an antecedent in that computation. The consequent part generally is a neural network, which is subject to defuzzification interpretation [1]. Next to neural networks one other important method in neural computation is due to neural tree structure. A neural tree is quite similar to a feed-forward neural network in the sense that it has a feed-forward structure with nodes and weights and a single or multiple outputs. However, it is built not layer by layer but node by node so that it has more free dimensions compared to a strictly defined feed forward type neural network. In a neural node, the nonlinear processor works as a fuzzy logic operator. Thus the present work uses neural tree in fuzzy logic form distinguishing itself from the existing neural tree paradigm where weights and activation functions at the neurons are subject to identification without need for interpretability as to the model constituents. Consequently, they are nonparametric models at their best. The model error is the difference between given output values belonging to the input patterns and the values provided by the neural tree outputs. In such data-driven neural tree modeling, black-box approach is generally the case. Hence, the model constituents do not have an intelligible interpretation, as their role is restricted to being a mathematical object subject to model error minimization. The structure, weights and functions are optimized with combinations of different methods. Although in the literature there are many neural tree structures and relatively few fuzzy tree structures, there is no fuzzy neural tree structure working with fuzzy logic principles [2] reported in the literature, according to the best knowledge of the authors.

The neural system presented in this work has a particular structure known as neural tree. The novelty of the paper is in two folds. On one side it works according to the fuzzy logic principles and on the other side the fuzzy set of membership function is formed by a likelihood function thereby processing probabilistic/possibilistic knowledge in the tree in terms of likelihood. Based on this, the research explores new potentials of neural tree systems for real-life soft computing solutions in various disciplines and multidisciplinary areas, where transparency of a model is demanded [3]. This is the case when the problem domain is complex, expert knowledge is soft, and multi-faceted linguistic concepts are involved. Examples of such areas are design science disciplines, such as architectural design, industrial design, and urban design. For this exploration, the coordination of the fuzzy logic and neural network concepts in a compact neuro-fuzzy modelling framework with probabilistic/possibilistic interpretations is endeavored, introducing some novel peculiarities for solid interesting gains in interdisciplinary implementations. It is emphasized that the novel fuzzy neural tree has uniquely an interpretation of a fuzzy logic system as well. Further, next to representing a complex, non-linear relation between input and outputs of a data set, it satisfies the consistency condition of possibility. Due to this additional property, the fuzzy neural
tree emulates a human-like reasoning, and permits the direct integration of existing expert knowledge during the model formation. The new framework is introduced as fuzzy neural tree with Gaussian type fuzzy membership functions being reminiscent of functioning in radial basis functions (RBF) type networks. The paper aims to explain the fuzzy neural tree in more explicit form with application in Architectural design.

The organization of the paper is as follows. Section II describes the fuzzy neural tree concept starting with the neural tree and joining fuzzy logic to it as a tree structured fuzzy neural system development. Section III describes the probabilistic/possibilistic base underlying fuzzy-neural tree. Section IV describes probabilistic-possibilistic approach for membership function in a unified form. Section V gives computer experiments in the area of architectural design. This is followed by discussion and conclusions.

II. Fuzzy Neural Tree

Broadly, a neural tree can be considered as a feed-forward neural network organized not layer by layer but node by node. The nonlinear functions at the nodes are Gaussians as in RBF networks. In fuzzy neural networks, this nonlinear function is treated as a fuzzy logic element like membership function or possibility distribution. Therefore, fuzzy logic is integrated into a neural tree with the fuzzy information processing executed in the nodes of the tree. A generic description of a neural tree subject to analysis in this research is as follows. Neural tree networks are in the paradigm of neural networks with marked similarities in their structures. A neural tree is composed of terminal nodes that also termed as leaf nodes, non-terminal nodes that are also termed as internal or inner nodes, and weights of connection links between the pairs of 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 possibility distribution in the tree structure which is considered to be a fuzzy logic system as its outcome is based on fuzzy logic operations thereby providing associated reasoning. In a conventional neural network structure there is a hierarchical layer structure where each node at the lower level is connected to all nodes of the upper layer nodes. However, this is very restrictive to represent a general system. Therefore, a more relaxed network model is necessary and this is accomplished by a neural-tree, the properties of which are as defined above. An instance of a neural tree is shown in figure 1. Each terminal node, also called leaf, is labeled with an element from the terminal set \( T = \{x_i, x_2, \ldots, x_n\} \) where \( x_i \) is the \( i \)-th component of the external input \( x \) which is a vector. Each link \((i,j)\) represents a directed connection from node \( i \) to node \( j \). 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. A non-terminal node should have minimally multiple inputs. It may have single or multiple outputs. An internal node having a single input is a trivial case and is not defined as a node. This is because in this case output of the node approximately is equal to the input that it is to be considered equal. This is due to the consistency condition which is to be widely seen later in the text. The node outputs are computed in the same way as computed in a feed-forward neural network. In this way, neural trees can represent a broad class of feed-forward networks that have irregular connectivity and non-strictly layered structures. In conventional neural tree structures generally connectivity between the branches is avoided. They are used for pattern recognition, progressive decision making, or complex system modeling. In contrast with such works, in the present research connectivity between the branches is possible, and the fuzzy neural tree structure is in a fuzzy logic framework for knowledge modeling, where fuzzy probability/possibility as element of soft computing is central. Added to this, the fuzzy neural tree functionality is based on likelihood representing fuzzy probability/possibility. This is another important difference between the existing neural trees in literature and the one in this work. Although in literature a family of likelihood functions is used to define a possibility as the upper envelope of this family [4, 5], to the authors’ best knowledge there is no likelihood function approach in the context of neural tree.

![Structure of a neural tree](image)

In the fuzzy neural tree, the output of \( i \)-th terminal node is denoted \( y_i \) and it is introduced to a non-terminal node. The detailed view of node connection from terminal node \( i \) to internal node \( j \) is shown in figure 2a and from an internal node \( i \) to another internal node \( j \) is shown in figure 2b. The connection weight between the nodes is shown as \( w_{ij} \). In the neural network terminology, a node is a neuron and \( w_{ij} \) is the synaptic strength between the neurons. This means, it represents the strength of connection between the nodes involved. In the fuzzy neural tree it is between zero and unity. Figure 3 shows some membership functions for the terminal nodes.

III. Probabilistic Base Underlying Fuzzy Neural Tree

The premise of the motivation of this work is to implement soft computing methodology for complex system analysis and design, where transparency of the model is demanded. For this purpose a novel fuzzy neural-tree concept is developed. The
fuzzy neural tree is especially for both knowledge modeling and expert knowledge modeling, making use of fuzzy logic for transparency. Fuzzy logic operates with fuzzy sets, where membership function is a very important concept [2]. Afterwards, the related concepts and fuzzy logic is extensively treated in literature, e.g. a good book is due to Belohlavek [6]. To determine what the appropriate measurement of membership should be, it is important to consider the interpretation of membership that the investigator intends. Here, the different ways may be laid out that have been proposed in the past, though it should be noted that there might be others. Five different views of membership have been identified. These views were neatly exemplified by Bilgic and Turk [7].

The vague predicate “John (x) is tall (T)” is represented by a number t in the unit interval [0, 1]. There are several possible answers to the question “What does it mean to say $t=0.7$?”:
1. Likelihood view: 70% of a given population agreed with the statement that John is tall.
2. Random set view: when asked to provide an interval in height that corresponds to “tall,” 70% of a given population provided an interval that includes John’s height in centimeters.
3. Similarity view: John’s height is away from the prototypical object, which is truly “tall” to the degree 0.3 (a normalized distance).
4. Utility view: 0.7 is the utility of asserting that John is tall.
5. Measurement view: When compared to others, John is taller than some, and this fact can be encoded as 0.7 on some scale.

In this work we propose a sixth way of interpretation as possibility measure due to Zadeh [8, 9] and further works are e.g. by Dubois and Prade [10, 11], and Alola et al. [12]; namely,
6. Possibilistic view: 70% of a given population possibly agreed with the statement that John is tall.

All these six different views of membership identification fall into two essential categories. These categories can be seen as probabilistic and possibilistic. As it will be shortly seen, in this work both categories are integrated into fuzzy neural tree consistently in a unified manner. Membership functions and probability measures of fuzzy sets are extensively treated in literature [13].

To start with we refer to figure 2a. We assume the input to an input node, namely a terminal node, is a Gaussian random variable, which is instructive to start with. This is due to the random set view given above, and this view can be extended due to the well-known central limit theorem in probability. In the fuzzy neural tree introduced in this work, all the processors operating in the internal nodes are Gaussian. Since the inputs to neural tree are also Gaussian random variables, due to functions of random variable theorem [14] all the processes in the tree are to be considered Gaussian. In a neural tree for each terminal input we define a linear or Gaussian fuzzy membership function as seen in figure 3, whose associated membership provides a probabilistic/possibilistic value for that input. Referring to figure 2, let us consider two consecutive nodes as shown in figure 2c. In the neural tree, any fuzzy probabilistic/possibilistic input delivers an output at any non-terminal node. Due to Gaussian considerations given above, we can consider this probabilistic/possibilistic input value as a random variable $x$ which can be modelled as a Gaussian probability density around a mean $x_m$. The probability density is given by

$$f_x(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-x_m)^2} \quad (1)$$

where $x_m$ is the mean; $\sigma$ is the width of the Gaussian.

**Definition:** Assuming a statistical model parameterized by a fixed and unknown $\theta$ the likelihood $L(\theta)$ is the probability of the observed data $x$ considered as a function of $\theta$.

The likelihood function of the mean value $x_m$ is given by [15]

$$L(\theta) = e^{-\frac{1}{2\sigma^2}(x-\theta)^2} \quad (2)$$

where $\theta$ is the unknown mean value $x_m$. Likelihood function is considered to be as a fuzzy membership function or fuzzy probability, converting the probabilistic uncertainty to fuzzy logic terms. $\theta$ is a general independent variable of the likelihood function, and the likelihood is between $0$ and $1$. $L(\theta)$ plays the role of fuzzy membership function and the likelihood at the node output is given by

$$y_j = L_j(\theta) \quad (3)$$

Referring to figure 2c, we consider the input $x_j$ of node $j$ as a random variable given by

$$x_j = y_j w_{ij} \quad (4)$$

where $w_{ij}$ is the synaptic connection weight between the node $i$ and node $j$ seen in figure 2. In the same way as described above, the pdf of $x_j$ is given by

$$f_{x_j}(x_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{1}{2\sigma_j^2}(x_j-x_m)^2} \quad (5)$$

and the likelihood function of the mean value $x_mj$ with respect to the input $x_j$ is given by

$$L_j(\theta_j) = e^{-\frac{1}{2\sigma_j^2}(x_j-\theta_j)^2} = e^{-\frac{1}{2\sigma_j^2}(w_{ij}y_j-\theta_j)^2} \quad (6)$$

and using (3) in (6), we obtain

$$L_j(\theta_j) = e^{-\frac{1}{2\sigma_j^2}(y_j-\theta_j)^2} = e^{-\frac{1}{2\sigma_j^2}(w_{ij}L_i(\theta_i)-\theta_j)^2} \quad (7)$$

We consider the neural tree node status where the likelihood is maximum, namely $L_i(\theta_i)=1$. In (7) using $L_i(\theta_i)=1$ we obtain

$$\theta_j = w_{ij} \quad (8)$$

for the maximum likelihood $L_i(\theta_i)=1$ where $\theta=x_mj$ is the unknown mean value of $x_j$. Hence, from (7) and (8), we obtain

$$L_j(\theta_j) = e^{-\frac{1}{2\sigma_j^2}(x_j-\theta_j)^2} \quad (9)$$

(9) indicates that likelihood $L_j(\theta_j)$ is maximum for $L_i(\theta_i)=1$ as was designed. $L_i(\theta_i)$ is the likelihood of the preceding node.

Referring to (3), from (9) we can also write

$$L_j(\theta_j) = e^{-\frac{1}{2\sigma_j^2}(y_j-\theta_j)^2} \quad (10)$$
Referring to (9) two observations are due.

(a) The likelihood \( L_i(\theta) \) is the probability of observed data \( L_i(\theta) \) which is the likelihood of the preceding node. With other words, each likelihood output of a node is the probability of the outcome of the preceding node output, which is the observed data in this likelihood context.

(b) For \( L_i(\theta)=1 \) the likelihood \( L_i(\theta) \) is maximum being independent of \( \theta \). However for \( L_i(\theta)\neq1 \), the likelihood \( L_i(\theta) \) is dependent on \( \theta \) while the stipulation (8) is still valid. This is due to the function of random variable theorem [14].

In (9), we see the variation of \( L_i(\theta) \) with respect to \( \theta \) while \( L_i(\theta) \) is a parameter. For \( L_i(\theta) \) close to unity and at the same time \( \theta \) is close to zero the likelihood \( L_i(\theta) \) is virtually maximal. To improve both the sensitivity and effectiveness of \( L_i(\theta) \) in the fuzzy logic computation, the obvious choice is

\[
\theta_j = 1 - L_i(\theta) = 1 - y_i
\]

(11)

So that (9) becomes

\[
L_i(\theta) = e^{-\frac{1}{2\sigma^2}(1-L_i(\theta))^2(1-y_i)^2}
\]

(12)

or alternatively

\[
L_i(\theta) = e^{-\frac{1}{2\sigma^2}(1-y_i)^2(1-y_i)^2}
\]

(13)

which is strongly dependent on \( y_i=1 \). For \( y_i=0 \), \( L_i(\theta)=1 \), the likelihood \( L_i(\theta) \) is also equal to unity. Any deviation from the unity is a measure of deviation from the maximum likelihood and this deviation is requested to be minimal for the effectiveness of the model for an input composition sought and identified finding an optimal compromise among the inputs of the tree. It is to note that, if all tree inputs are unity then all the node outputs are also unity independent of the weights. Therefore in the model node inputs must be strongly conflicting and their compromise is subject to optimization as this will be seen in Section V.

The justification of (11) can also be seen by Shannon’s information theorem. As the node output \( y_i \) is probabilistic, according to the Shannon’s theorem, when the probability is zero, i.e. \( y_i=0 \) the information content is infinite and that is conveyed from the preceding node to the taking, the weight, i.e. \( \theta \), between the nodes maximum, i.e. \( \theta=1 \). Conversely, when the probability is unity, i.e. \( y_i=1 \), the information content is zero. Therefore the same weight is taken to be zero i.e. \( \theta=0 \), to avoid for any information conveyance. In (12) it is seen that if \( L_i(\theta)=1 \) then \( L_i(\theta)=1 \).

This is independent of \( \theta \), and that applies to any node. This means, if the probability of the likelihood \( L_i(\theta)=1 \) unity then no information is necessary because maximum likelihood is accomplished anyway as designed being independent of the weight \( w_i=\theta \) in this case. In other words, if the probability of the observed data i.e., the likelihood \( L_i(\theta) \) is unity, it propagates in the same way throughout the entire network. In the actual implementation, in place of weights given by (11) their normalized versions are used as \( w_i=\theta \). The explicit input node and inner node connections to the upper nodes are shown in figure 2b, where the node outputs are denoted by \( O \) as a generic symbol. Referring to (3) and figure 2b the likelihood function in (12) becomes

\[
L_i(\theta) = O_i = e^{-\frac{1}{2\sigma^2}(1-O_i)^2(1-O_i)^2}
\]

(14)

and from (8) and (11),

\[
w_i = 1 - O_i
\]

(15)

For a leaf node, i.e., an input node to the tree, we define a Gaussian or linear fuzzy membership function, which serves as a fuzzy likelihood function indicating the likelihood of that input relative to its ideal value, which is equal to unity. This input is shown as \( z_i \) in figure 2a, and the fuzzy likelihood is shown as \( y_i \) in the same figure.

IV. PROBABILISTIC-POSSIBILISTIC APPROACH FOR MEMBERSHIP FUNCTION IN A UNIFIED FORM

A. Fuzzy Neural Tree with Logical AND Operation

The likelihood function in its normalized form is considered to be as a fuzzy probability, being a membership function of a fuzzy set. To unify likelihood function concept of fuzzy membership function with the possibility interpretation, at this point we take a possibilistic view for the membership function departing from fuzzy probabilistic view. In this case, membership function can also be considered as a possibility distribution, so that the fuzzy membership function represents also a possibility distribution function [8, 9]. These are due to the axioms of possibility measure or a fuzzy possibility distribution function \( \pi(A) \) of a fuzzy event \( A \), given below

\[
\forall A, i \in I \quad \Pi(\bigcup_i A) = \max_{i \in I} (\Pi(A))
\]

(16)

\[
\forall A, i \in I \quad \Pi(\bigcap_i A) = \min_{i \in I} (\Pi(A))
\]

(17)

where \( \Pi \) represents the possibility, and \( \Pi(A) \) is the instantiation of \( \pi(A) \). In the probability theory, the probability of the intersection of independent events is given by

\[
\forall A, i \in I \quad P(\bigcap_i A) = P(\bigcap_i A) \quad \forall A, i = A
\]

(18)

In fuzzy logic the same intersection operation is given by

\[
\forall A, i \in I \quad \min_{i \in I} \bigcap_i A \quad \forall A = A
\]

(19)

From the possibility viewpoint, the independence is not defined. In place of that we define interaction. In possibility theory we say that two events are independent or non-interactive to express that they are not interdependent. Independence is a stronger condition than non-interaction, because in the latter case independence does not apply. However, still we can consider the equation given in (16). This is not standard from the standpoint of set theory. However, they could have been introduced by Zadeh in his quest for developing possibility-like measures for fuzzy sets [8, 9]. Based on these views, here we consider one additional possibility relation naturally derived from (16) as follows.

\[
\forall A, i \in I \quad \Pi(\bigcap_i A) = \Pi(\bigcup_i A) = (\Pi(A)) \quad \forall A = A
\]

(20)

If all the possibilities of events are equal, the outcomes of logical AND or OR operations are the same, and equal to the possibility of any event in those operations. This important result will be used later in this work, where it will be named as consistency condition. One can note that the possibilistic equations (14) and (15) are similar to those probabilistic fuzzy
logic operations applied to fuzzy sets, referring to the likelihood function interpretation of the fuzzy membership functions. Therefore, the approach used for fuzzy neural tree operation is named in this work as probability/possibility approach, where established conventional both, fuzzy probability and fuzzy possibility, operations are equally valid. However, in contrast to the conventional equivalence of fuzzy probabilistic and fuzzy possibilistic operations through the fuzzy membership functions, in this work a difference is recognized and implemented in such a way that fuzzy possibilistic computations are conservative estimate of fuzzy probabilistic computations and this will be exemplified later in the sequel.

We can summarize the work presented above as follows.

i. Node outputs always represent a likelihood function which at the same time represents a fuzzy probability/possibility function.

ii. \( L_i(\theta) = 1 \) corresponds to a fuzzy probability/possibility equal to unity and it propagates in the same way so that the following fuzzy likelihood \( L_i(\theta) \) component for the corresponding input \( O_i \) is also unity, as seen in (11).

iii. In the same way, if all the probabilistic/possibilistic inputs to the neural tree are unity, then all the node outputs of the neural tree nodes are also unity, providing a probabilistic/possibilistic integrity where the maximum likelihood prevails throughout the tree.

iv. Any deviation from unity at a leaf node output causes associated deviations from the maximum likelihood throughout the tree. Explicitly, any probabilistic/possibilistic deviation from unity at the neural tree input will propagate throughout the tree via the connected node outputs as estimated likelihood representing a probabilistic/possibilistic outcome in the model. Thus, any deviation from unity at a leaf node output is a degree of deviation from its ideally desired value. If this value is unity then it propagates as unity throughout the model.

v. Each inner node in the tree represents a fuzzy probabilistic/possibilistic rule. In a fuzzy modeling the shape and the position of a fuzzy set are essential questions. In the present neural tree approach all the locations are normalized to unity and the shape of the membership function is naturally formed as Gaussian based on the probabilistic considerations.

vi. Each input to a node is assumed to be independent of the others so that the fuzzy memberships of the inputs can be thought as forming a joint multidimensional fuzzy membership. The dependence among the inputs is theoretically possible but actually it is out of concern because each leaf node has its own stimuli or its own membership function, and they are not common to the others, in general. However joint membership concept is not central to the computations.

\( O_i \) propagates to the following node output \( O_j \) in a way determined by the likelihood function. If there is more than one input to a node, assuming that the inputs are independent, the output is given according the relation

\[
L_j(\theta_j) = L_i(\theta_i) L_{i-1}(\theta_{i-1}) \ldots L_1(\theta_1)
\]  

(21)

For a multiple input case of two node inputs, (21) becomes

\[
L_j(\theta_j) = L_i(\theta_i) L_{i-1}(\theta_{i-1}) = e^{-\frac{w_i^2}{\sigma_i^2}(O_i-1)^2} e^{-\frac{w_{i-1}^2}{\sigma_{i-1}^2}(O_{i-1}-1)^2} 
\]

(22)

For a case of \( n \) multiple inputs, in view of (15) and (21), we write

\[
L_j(\theta_j) = O_j = f(O) = \exp[-\frac{1}{2\sigma_j^2} \sum_{i=1}^{n} (1-O_i)^2 (O_i-1)]
\]

(23)

Logical AND

where \( n \) is the number of inputs to the node and \( \sigma_i \) is the common width of the Gaussians. As it is seen from (23), the previous node output \( O_i \) plays important role in the following node output and this role is weighted by the connection weight \( w_{ij} \) given by (15).

It is interesting to note that in fuzzy logic terms, the likelihood function (14) can be seen as a two-dimensional fuzzy membership function with respect to the weighted node outputs \( x_1 \) and \( x_2 \). In this case the neural tree node output can be seen as a fuzzy rule, which can be stated as

\[
IF \{ O_1 = X_1 \ AND O_2 = X_2 \} THEN \{ O_i \ given \ by \ (23) \}
\]

(24)

The output of an internal neural tree is determined by (21) as a product operation for the sake of computational accuracy, rather than min or max operation. However, in these computations the Gaussian width \( \sigma_j \) in (23) is assumed to be known, although it is not determined yet. To determine \( \sigma_j \) we impose the fuzzy probability/possibility measure for the cases all the inputs to a node are equal as probabilistic/possibilistic condition, as expressed in (20). In the same way, for the logical OR operation, to determine \( \sigma_j \) we impose the fuzzy possibility measure for the cases all the inputs to a node are equal as probabilistic/possibilistic condition. By these impositions there is no sacrifice of accuracy involved. We can determine the Gaussian width \( \sigma_j \) by learning the input and output association given in Table 1 for 6 inputs, as an example.

<table>
<thead>
<tr>
<th>In1</th>
<th>In2</th>
<th>In3</th>
<th>In4</th>
<th>In5</th>
<th>In6</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1</td>
<td>.1</td>
<td>.1</td>
<td>.1</td>
<td>.1</td>
<td>.1</td>
<td>.1</td>
</tr>
<tr>
<td>.2</td>
<td>.2</td>
<td>.2</td>
<td>.2</td>
<td>.2</td>
<td>.2</td>
<td>.2</td>
</tr>
<tr>
<td>.3</td>
<td>.3</td>
<td>.3</td>
<td>.3</td>
<td>.3</td>
<td>.3</td>
<td>.3</td>
</tr>
<tr>
<td>.4</td>
<td>.4</td>
<td>.4</td>
<td>.4</td>
<td>.4</td>
<td>.4</td>
<td>.4</td>
</tr>
<tr>
<td>.5</td>
<td>.5</td>
<td>.5</td>
<td>.5</td>
<td>.5</td>
<td>.5</td>
<td>.5</td>
</tr>
<tr>
<td>.6</td>
<td>.6</td>
<td>.6</td>
<td>.6</td>
<td>.6</td>
<td>.6</td>
<td>.6</td>
</tr>
<tr>
<td>.7</td>
<td>.7</td>
<td>.7</td>
<td>.7</td>
<td>.7</td>
<td>.7</td>
<td>.7</td>
</tr>
<tr>
<td>.8</td>
<td>.8</td>
<td>.8</td>
<td>.8</td>
<td>.8</td>
<td>.8</td>
<td>.8</td>
</tr>
<tr>
<td>.9</td>
<td>.9</td>
<td>.9</td>
<td>.9</td>
<td>.9</td>
<td>.9</td>
<td>.9</td>
</tr>
</tbody>
</table>

If all the inputs are 0.1 then output is 0.1; if all the inputs are 0.2 then output is 0.2; ... and so on. For all the inputs are unity, i.e. \( O_i = 1 \), then output is inherently unity irrespective to the weights of the system which means if the probability/possibility of all events at the input is unity, then probability/possibility of the output should be, and indeed is unity. If at the terminal nodes the inputs are fuzzy probabilities/possibilities, then this result remains the same matching the consistency condition given by (24) \[8, 9, 11, 16-18\].
Theoretically, if all the inputs are zero, i.e. \( P(A) \leq \Pi(A) \) \( \forall A \subseteq U \) (25) where \( P(A) \) is the fuzzy probability measure; \( U \) is the universe of discourse; \( \Pi(A) \) is the possibility measure. Incidentally, if \( \Pi(A) \) is equal to zero, then \( P(A) \) is also zero but the converse may not be true. The implementation of inequality (25) is due to conservative estimate of fuzzy possibility \( \Pi(A) \), which is equal to fuzzy probability \( P(A) \), during consistency conditions computations. The optimal sigma values are given in table 2.

It is interesting to note that, \( \sigma_i \) having been determined using Table 1, (21) can be written in the form

\[
L_i(\Theta) = O_i = \exp\left(-\frac{1}{2} \sum_{j=1}^{n} \left(O_j \sigma_j \right)^2 \right)
\]

which means, for each input there is an associated Gaussian width \( \sigma_i \) that is determined by the weight \( w_{ij} = J \cdot O_j \). If \( w_{ij} \) is zero, the respective \( \sigma_j \) is infinite so that the input via the weight \( w_j \) has no effect on the output, as the multiplication factor at the logic AND operation becomes unity. Theoretically, if all the inputs are zero, i.e. \( O_i = 0 \), then there is still a finite node output. This is due to the fact that the Gaussian does not vanish at the point where its independent variable vanishes. From the possibilistic viewpoint, this implies that even in the event probability or likelihood vanishes, the possibility remains finite. However the preceding node output never totally vanishes as far as \( O_i \) is concerned or it does not make sense to consider if terminal node output \( x_i \) vanishes. This is because a zero input becomes irrelevant throughout the model. The consistency condition refers to virtually multidimensional triangular fuzzy membership functions in a continuous form, where, in the case all inputs variables are equal, the multidimensional membership function value is equal to the same input number. In particular in the neural tree the membership function value is equal to the fuzzified node output. This is illustrated in figure 4a. Referring to this figure, it is clear that in a node multidimensional fuzzy membership function has a maximum at the point where all inputs are unity. Considering that the inputs are between zero and one, as a node only one half of multidimensional Gaussian fuzzy membership function enters into the computation. Its extension beyond unity is not to be considered in any computation. We are using Gaussian multidimensional fuzzy membership functions for likelihood computation; however, the consistency condition of possibility forces us to approximate the multi-dimensional Gaussian membership function to a virtually continuous triangular multi-dimensional membership function. Referring to this approximation, the exact triangular multi-dimensional fuzzy membership function is shown in figure 4b. The knowledge processing at a node is probabilistic, proportionally to the values of the inputs that they differ from each other. As the inputs all together tend to be the same, then the knowledge processing at the node proportionally tends to be possibilistic. This is the important property established in this work, making the probabilistic and possibilistic computations in a fuzzy-neural computation unified.

### Table II

<table>
<thead>
<tr>
<th>Nr. of inputs</th>
<th>( \sigma ) AND</th>
<th>( \sigma ) OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.299</td>
<td>0.295</td>
</tr>
<tr>
<td>3</td>
<td>0.244</td>
<td>0.241</td>
</tr>
<tr>
<td>4</td>
<td>0.212</td>
<td>0.208</td>
</tr>
<tr>
<td>5</td>
<td>0.189</td>
<td>0.186</td>
</tr>
<tr>
<td>6</td>
<td>0.173</td>
<td>0.170</td>
</tr>
<tr>
<td>7</td>
<td>0.160</td>
<td>0.158</td>
</tr>
<tr>
<td>8</td>
<td>0.150</td>
<td>0.147</td>
</tr>
<tr>
<td>9</td>
<td>0.141</td>
<td>0.139</td>
</tr>
<tr>
<td>10</td>
<td>0.134</td>
<td>0.132</td>
</tr>
</tbody>
</table>

**Fig. 4.** Description of the consistency condition for two-dimensional antecedent space (a); one-dimensional consequent space (singleton)

#### B. Fuzzy-Neural Tree with Logical OR Operation

A general \( n \)-weights case for a node, the knowledge model should be devised somewhat differently for logical OR operation, and referring to figure 2, this can be accomplished as follows. The logic OR operation is fulfilled by means of the de Morgan’s law which is given below.

\[
O_i \cup O_j = \overline{O_i \cap O_j} \quad \text{(27)}
\]

where the complement of \( O \) is given by

\[
\overline{O} = 1 - O \quad \text{(28)}
\]

Hence for OR operation corresponding to the AND operation in (23) becomes

\[
L_j(\Theta) = O_j = f(O_i) = 1 - \exp\left(-\frac{1}{2 \sigma_j} \sum_{i=1}^{n} w_{ij} O_i^2 \right)
\]

\[
= 1 - \exp\left(-\frac{1}{2 \sigma_j} \sum_{i=1}^{n} \frac{O_i^2}{w_{ij}} \right) \quad \text{(29)}
\]

**Logical OR**

where \( w_{ij} \) is given by (15), so that we have to consider \( \overline{W}_{ij} \) as given by (30).

\[
\overline{W}_{ij} = 1 - w_{ij} = L_j(\Theta) = O \quad \text{(30)}
\]

To obtain (29) for the node \( j \) we take the complement of the incoming node outputs \( O_i \) and carry out the logic AND operation, \( L_j(\Theta) \) is a multivariable fuzzy probability/possibility distribution, as seen from (26) the result gives the complement of the output of the node \( O_i \). After this operation complement of this outcome gives the desired final result (29) as \( O_j \). In words, first we take the complement of the \( O_i \), afterwards we execute multiplication and finally we take the complement of the multiplication. It is important to note that in this computation the Gaussian is likelihood representing a probabilistic/possibilistic entity. In this case the neural tree node output can be seen as a fuzzy rule which can be stated as

If \([O_i = X_i \text{ OR } O_j = X_j] \) Then

\[
[O_{ji} \text{ given by (21)}]
\]

If all the \( O_i \) inputs in (29) are zero, the output is also zero. However, if all the inputs are unity, i.e. \( O_i = 1 \), then the node output is apparently not exactly unity because the exponent in (29) given by

\[
\exp\left(-\frac{1}{2 \sigma_j} \sum_{i=1}^{n} \frac{O_i^2}{w_{ij}} \right)
\]

remains small but finite. From the probabilistic/possibilistic
view point, this implies that when the event fuzzy probability/possibility $O_i$ is $I$, the outcome possibility remains less than $I$, which apparently does not conform to (25).

However, this is circumvented by the consistency provision, namely, if all the inputs to a node are unity, the output of the node is also unity. Actually the preceding node output is never exactly unity as far as $O_i$ is concerned since such an output becomes irrelevant otherwise throughout the model. On the other hand, as the degree of association $w_i$ given by (15) goes to zero, the effect of $O_i$ on output $O_j$ in (29) becomes irrelevant, the result being consistent.

V. COMPUTER EXPERIMENTS

Computer experiments are carried out to demonstrate the validity of the fuzzy neural tree modeling in the context of a task involving soft objectives. The experiments address to an Architectural design problem, which is an actual design case, where the fuzzy neural tree plays role of objective functions. It concerns the design of an ensemble of residential housing units. The site is shown in figure 5 from a bird’s eye view. The lot divisions on the site are shown in the figures by means of black lines. The north direction is upward in the figure. In the figure, 20 houses are seen. Two of them are existing buildings, which are shown in white color, and these are not involved in the experiment. The other 18 houses are shown in blue color, and they are 12m long in east-west direction, and 8m long in north-south direction, except houses $H9-H18$ having a square shaped floor plan of 8m by 8m. The 18 houses are subject to optimal positioning, so that the ensemble has some desirable properties. Two objectives are involved in the present experiment. One refers to visual perception aspects of the houses and the other one to the size of their gardens. The privacies of every house are denoted by $O_i$, and they are 12m long in east-west direction, and 8m long in north-south direction, except houses $H9-H18$ having a square shaped floor plan of 8m by 8m. The 18 houses are subject to optimal positioning, so that the ensemble has some desirable properties. Two objectives are involved in the present experiment. One refers to visual perception aspects of the houses and the other one to the size of their gardens. The perceptual objective is all houses are wanted and desired to have a high visual privacy, meaning that the houses should be minimally exposed to visual perception from the other buildings around it. The second objective is all gardens are wanted and desired to be large. The gardens are represented in the figure as green surface patches. The two objectives are soft, due to their linguistic nature. More specifically, the attributes high and large are imprecise, and the privacy property involves both imprecision and uncertainty, due to its perceptual nature. Moreover, the statements all houses and all gardens are wanted and desired to have imply that the imprecise attributes of each house should be aggregated appropriately. This is accomplished by means of two logic AND operations, one at inner node $I_1$ and the other at $I_2$ of the Fuzzy Neural Tree seen in figure 6. Referring to the figure it is to note that the first objective is the output of node $I_1$ denoted by $F_1(f(x))=L_1(\theta)$. The second objective is the output of node $I_2$ denoted by $F_2(f(x))=L_2(\theta)$. Both objectives are likelihoods given by (23) and subject to maximization. The connection weights $w_{111}, w_{112},..., w_{136}$ in the model are assigned according to (15), so that every stimulus pattern has an associated distinct weight pattern in the tree model. In figure 6 the privacies of every house are denoted by $O_{T1},...,O_{T18}$. The privacy objective measurement is executed for the south facades of the buildings, as, due to direct sunlight considerations, the living rooms will be situated at the south façades of the buildings, and for these rooms visual privacy is a desirable property in general. Requirements for perceptual properties, such as visual privacy, are generally difficult to take into account, because every spatial situation contains abundant visual information that obscures the subtle differences among the situations in terms of their respective perceptual properties. To deal with this issue, a probabilistic perception model is used in to quantify the perception of a building’s façade from another building [19] and the perceptions of a facade are fused [20] yielding the joint perceptions $f_1(x),..., f_{18}(x)$ at the inputs of the terminal nodes of the neural tree. Each building is perceived from several other buildings, and the perception events are independent events. Therefore, for the privacy computation, the union of the perception events is obtained for each facade. The privacies are considered as fuzzy statements related to the union of perceptions via fuzzy membership function. Membership function used in the terminals $T_7-T_{10}$ is shown in figure 7 left, and the membership degrees are the visual privacies. In figure 6 the performances of the gardens are

![Fig. 6 Fuzzy neural tree model of the housing problem](Image)
denoted by $O_{TIP}O_{T37}$. The performance of a garden is considered as a fuzzy statement related to the size of the garden via fuzzy membership function, where the membership degree is the garden performance. The membership functions at the terminals $T_{IP}$-$T_{16}$ are exemplified for $T_{21}$ in figure 7 right. The likelihoods $F_{1}(f(x))=L_{1}(\theta)$ and $F_{2}(f(x))=L_{2}(\theta)$ in figure 6 are used to identify optimized housing layout configurations. This is accomplished by a Pareto-dominance based multi-objective evolutionary algorithm NSGA-II [21] with population size 300. The resulting Pareto front solutions are shown as black points in objective function space in figure 8. One of the Pareto optimal solutions is shown in figure 5 and marked by an arrow in figure 8.

![Actual implementation of membership functions in figure 3 for visual privacy at terminals $T_{7}$-$T_{10}$ (left), and garden size at $T_{21}$ (right) in figure 6](image)

**Fig. 7** Actual implementation of membership functions in figure 3 for visual privacy at terminals $T_{7}$-$T_{10}$ (left), and garden size at $T_{21}$ (right) in figure 6

![Pareto optimal front; one of the solutions is marked by an arrow and its instantiation is shown in figure 5](image)

**Fig. 8** Pareto optimal front; one of the solutions is marked by arrow and its instantiation is shown in figure 5.

VI. DISCUSSION AND CONCLUSION

A novel fuzzy-neural tree (FNT) is presented, where each tree node uses a Gaussian as a fuzzy membership function, so that the approach is shown to be uniquely compatible for both the probabilistic and possibilistic interpretations of fuzzy membership, thereby presenting a novel type of cooperation between fuzzy logic and neural structure. The processing of input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The processing of membership, thereby presenting a novel type of cooperation between fuzzy logic and neural structure. The processing of input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations. The tree is employed by multi-objective optimization for the input information is due to logical AND and OR operations.