THE JOURNAL OF
COGNITIVE SYSTEMS
VOLUME 01 NUMBER 01 DECEMBER 2016

NOTE ON THE INFORMATION-THEORETIC ASPECT OF FUZZY NEURAL TREE

Özer Ciftcioglu[1], Michael S. Bittermann[2]

[1] Delft University of Technology, Delft, The Netherlands
[2] Maltepe University, Istanbul, Turkey

A note on the information theoretic aspect of the fuzzy neural tree (FNT) is presented. The detailed description of the FNT is given in an earlier work, where its information-theoretic aspect is heuristically mentioned but not elaborated because of some space limitation. The present note is to highlight this aspect of the tree as this is important in working of the tree with its knowledge-driven structure.

Index Terms — Fuzzy logic, neural tree, knowledge modeling, evolutionary computation, likelihood, probability possibility

I. INTRODUCTION

Fuzzy neural tree (FNT) structure is given in an earlier published work where its information-theoretic aspect is briefly mentioned [1]. The present note is to highlight this aspect of the tree as this is important in working of the tree with its knowledge-driven structure. The information-theoretic aspect of the FNT concerns the application of fuzzy concept to some concepts of Information Theory and using the result as knowledge in the tree structure. In this way, the FNT is driven by both assessments of soft issues as fuzzy memberships, and fuzzy membership of measurement data. The rest of the FNT structure is formed by the fuzzy information as knowledge source for the tree. The brief description of the FNT is intentionally presented here for the sake of the completeness of the note.

Neural tree concept and neuro-fuzzy computation is well established in the literature. In particular, neural tree concept is a kind of “free format” neural computation where layer–by-layer structure of neural network is relaxed as this will be shown shortly afterwards. In the realm of neuro-fuzzy paradigm, a neural network can be considered as fuzzy system in the sense of the non-linearity introduced at the neurons can be seen as fuzzy membership functions. Although such a view is appealing from the fuzzy system viewpoint, fuzzy interpretation of a neural network becomes formidably involved as the network is not a simple one. Therefore, a neural network is established generally by learning the input data without any recourse to fuzzy considerations. Then such structure is considered as non-parametric model. On the other hand a neural network can be established by some fuzzy considerations as a knowledge model and the same structure can be seen as a parametric model, depending on the input data in both cases. Even in this parametric model case some ranked structure of the neural network can be relaxed and the knowledge considered in this context can be the information provided by the inputs of the network. As it can duly be anticipated, in this fuzzy model the input data is represented in terms of information and this information is fuzzified being subject to fuzzy information processing.

The organization of the paper is as follows. Section II describes the structural and computational aspects of fuzzy neural tree, as well as its information-theoretic aspect. This is followed by conclusions.

Manuscript received September 30, 2016; accepted October 14, 2016.
Corresponding author: Özer Ciftcioglu (E-mail:: o.ciftcioglu@tudelft.nl).
Digital Object Identifier:

II. FUZZY NEURAL TREE

A. Structural and Computational Aspects

A neural tree can be considered as a feed-forward neural network that is organized not layer by layer but node by node. The nodes comprise nonlinear functions for processing the incoming information. 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 obvious similarities in their structures. A neural tree consists of terminal nodes that also referred to as leaf nodes, non-terminal nodes that are also referred to as internal or inner nodes, and weights associated with the connection links between the pairs of nodes. The non-terminal nodes are considered to be neural units, as 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, as well as 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 stipulation is very restrictive when a general system should be represented. 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, is labeled with an element from the terminal set $T=[x_1, x_2, \dots, x_n]$ where x_i is the *i*-th component of the external input $\boldsymbol{x}$ 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.

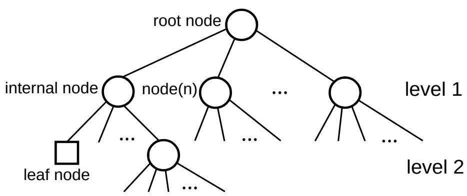

Fig. 1. Structure of a neural tree

A non-terminal node should have minimally multiple inputs to be meaningful, although a single input is also valid for operation. A node may have a single or multiple outputs;.

An internal node having a single input is considered to be a trivial case. This is because in this case output of the node is approximately equal to the input that it is to be considered equal. 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 [2, 3], to the authors' best knowledge there is no likelihood function approach in the context of neural tree. 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.

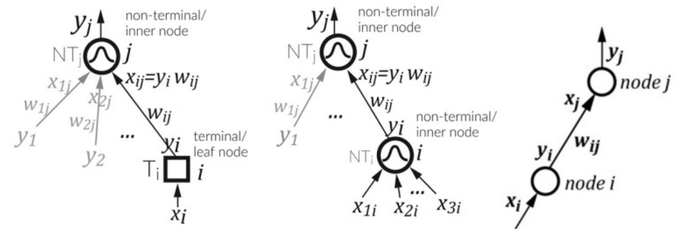

Fig. 2. The detailed structure of different type of node connections

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 sample membership functions for the terminal nodes.

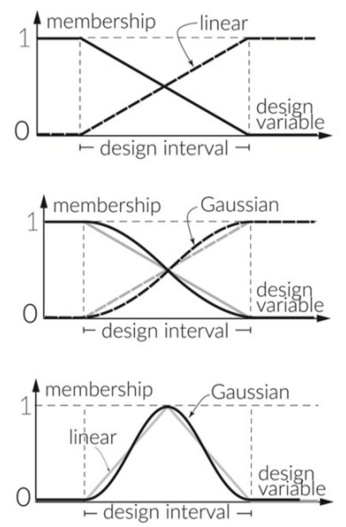

Fig. 3. Some sample membership functions at the terminal nodes

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. 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 [4] 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 of a node 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

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

where x_m is the mean; σ is the width of the Gaussian.

D e f i n i t i o n : Assuming a statistical model parameterized by a fixed and unknown θ the likelihood $L(\theta)$ is the probability of the observed data x considered as a function of θ.

The likelihood function of the mean value x_m is given by [5]

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

where θ 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. θ 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

$$y_j=L_j(\theta_j) \tag{3}$$

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

$$x_j=y_i w_{ij} \tag{4}$$

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

$$f_{x_j}(x_j)=\frac{1}{\sqrt{2\pi}\sigma_j}e^{-\frac{1}{2\sigma_j^2}(x_j-x_{mj})^2} \tag{5}$$

and the likelihood function of the mean value $\theta=x_{mj}$ with respect to the input x_j is given by

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

where θ is the likelihood parameter. Using (3) in (6), we obtain

$$L_j(\theta_j)=e^{-\frac{1}{2\sigma^2}(x_j-\theta_j)^2}=e^{-\frac{1}{2\sigma^2}(w_{ij}L_i(\theta_i)-\theta_j)^2} \tag{7}$$

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

$$\theta_j=w_{ij} \tag{8}$$

for the maximum likelihood $L_j(\theta_j)=1$. Hence, from (7) and (8), we obtain 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.

$$L_j(\theta_j)=e^{-\frac{1}{2\sigma_j^2}\theta_j^2(L_i(\theta_i)-1)^2} \tag{9}$$

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

$$L_j(\theta_j)=e^{-\frac{1}{2\sigma_j^2}\theta_j^2(y_j-1)^2}=e^{-\frac{1}{2\sigma_j^2}\theta_j^2(1-y_j)^2} \tag{10}$$

Referring to (9) the likelihood $L_j(\theta_j)$ is the probability of observed data as a function of θ via $L_i(\theta_i)$ which is the likelihood of the preceding node output. In other words, each likelihood output of a node is dependent on the probability of the outcome of the preceding node output, which is the observed data in this likelihood context.

B. Information-Theoretic Aspect

For $L_i(\theta_i)=1$ the likelihood $L_j(\theta_j)$ is maximum being independent of θ_j. However for $L_i(\theta_i)\neq 1$, the likelihood $L_j(\theta_j)$ is dependent on θ_j. In (9), we note the variation of $L_j(\theta_j)$ with respect to θ_j while $L_i(\theta_i)$ is a parameter. For $L_i(\theta_i)$ close to unity or θ_j is close to zero likelihood, then $L_j(\theta_j)$ is close to maximum. From the information theory viewpoint, likelihood is probability p and the information is given by

$$I=-\log p=-\log L(\theta) \tag{11}$$

The information content of likelihood is given by (11) since $L(\theta)$ is considered to be a fuzzy probability [6] in the form of a membership function. The fuzzification of this information is accomplished by means of the information fuzzy membership function

$$MF=1-\exp(-I) \tag{12}$$

as this is shown in figure 4 with respect to information.

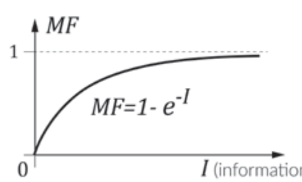

Fig. 4. Fuzzy membership function of information

The same information fuzzy membership function with respect to likelihood is shown in figure 5.

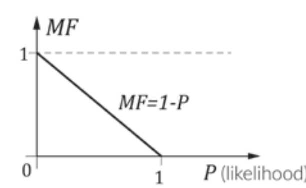

Fig. 5. Membership function of fuzzified information

The fuzzy membership function of information in figure 4 can take slightly different forms, taking the decay constant different than unity. In that case the membership function in figure 5 would read $MF = 1 - P^{\tau}$ where τ denotes the decay constant.

The membership function value of the fuzzified information is used as the connection weight in the fuzzy neural tree,

$$w_{ij} = 1 - p_{ij} = 1 - L_i(\theta_i) \quad (13)$$

as was explained above by (1) through (8). The fuzzified information is to consider as *fuzzy information* between zero and unity. In the FNT, the connection weights throughout the model are determined by the inputs of the FNT without recourse to any expert knowledge, in this knowledge model. It is interesting to note that if the inputs of the model are measurement data, then the measurements are fuzzified by means of appropriate membership function to a fuzzy probability as shown in figure 6.

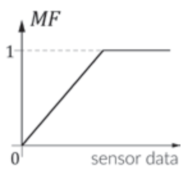

Fig. 6. Membership function as fuzzy probability

If the inputs of the model are soft inputs, then these inputs are considered to be directly fuzzified inputs between zero and unity and the fuzzfied information introduced above prevails throughout the model.

The heuristic explanation of (13) is as follows. θ_j refers to the connection of the node i to the node j. From the information theoretic viewpoint y_i is a probability and it contains no information when it is unity. In this case we do not have to convey any information from node i to node j, and therefore $\theta_j = 0$. From other side if y_i is zero, it contains information that it goes to infinity. Therefore, we connect the node i to node j with total connectivity, that means $\theta_j = 1$ in the case of single input. For a multiple input case, which is the non-trivial or actual situation, θ_j is selected in a normalized form for defuzzification in the rule-chaining process through from node to node process in the tree.

$$\theta_{ji} = \frac{1 - y_i}{\sum_{i=1}^{n}(1 - y_i)} \quad (14)$$

$$y_j = L_j(\theta_j) = e^{-\frac{1}{2\sigma_j^2}\theta_j^2 (y_i - 1)^2} = e^{-\frac{1}{2\sigma_j^2}\left[\frac{1-y_i}{\sum_{i=1}^{n}(1-y_i)}\right]^2 (y_i - 1)^2} \quad (15)$$

In (15) n is the index number of the number of inputs to the node j.

III. CONCLUSION

A note on a fuzzy neural tree is presented from the information-theoretic properties viewpoint involved in the tree. Information-theoretic viewpoint is essential for an automatic knowledge model formation directly from the inputs of the model. Heuristically we can consider that the information supplied to the model is from the inputs, and this information is used to form the model without any training process. This is an important property of the present neural tree structure since in the general neural tree concept in literature, the tree structure is determined in one way or other by learning and hence the model is non-parametric.

Fuzzy neural tree subject to study in this work, is an essential component of computational cognition, and its effectiveness is demonstrated in several applications reported in the literature [7, 8].

REFERENCES

[1] Ö. Ciftcioglu and M. S. Bittermann, "A Fuzzy Neural Tree Based on Likelihood," presented at the 2015 IEEE International Conference on Fuzzy Systems - FUZZ-IEEE 2015, Istanbul, Turkey, 2015.

[2] D. Dubois and H. Prade, "A semantics for possibility theory based on likelihoods," *J. of Mathematical Analysis and Applications* pp. 205, 359-380, 1997.

[3] D. Dubois and H. Prade, "A semantics for possibility theory based on likelihoods," in *Int. Joint Conf. of Fourth IEEE Int. Conf. on Fuzzy Systems and Second Int. Fuzzy Engineering Symposium - Fuzzy Systems 1995*, Yokohama, Japan, 1995, pp. 1597-1604.

[4] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 1965.

[5] Y. Pawitan, *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. New York: Clarendon Press Oxford, 2001.

[6] L. A. Zadeh, "Is there a need for fuzzy logic?," *Information Sciences,* vol. 178, pp. 2751-2779, 2008.

[7] Ö. Ciftcioglu and M. S. Bittermann, "Computational Cognitive Color Perception," presented at the IEEE World Congress on Computational Intelligence - WCCI 2016, Vancouver, Canada, 2016.

[8] O. Ciftcioglu and M. S. Bittermann, "Fuzzy Neural Tree in Evolutionary Computation for Architectural Design Cognition," in *IEEE Congress on Evolutionary Computation - CEC 2015*, Sendai, Japan, 2015, pp. 2319 - 2326.

Özer Ciftcioglu was born in Turkey. He received the PhD degree in Electrical Engineering from Istanbul Technical University (ITU) in 1976. He became associate professor in 1982 and full professor in 1988 at the same university.

Since 2009 he has been working at TU Delft in the Netherlands as Emeritus professor. He had been working at the same university from 1998 to 2009. From 1967 to 1998 he was affiliated with ITU as an academic member. His current research interests are Signal Processing, Computational Intelligence, and Computational Cognition.

Prof. Ciftcioglu is a senior member of IEEE since 2001.

Michael S. Bittermann was born in Germany in 1976. He graduated with the highest honors as MSc in Architecture from Delft University of Technology (TU Delft), The Netherlands, in 2003. He received his PhD degree in Design Informatics with the highest honors from the same university in 2009. Bittermann was distinguished as one of the five eminent researchers among all 2008-2009 PhD graduates of TU Delft in the form of the Young Researcher Fellowship of Delft University of Technology (80k€). He joined Maltepe University, Istanbul, Turkey in 2013, where he contributed to the research project titled 'Design by Cognition and Comprehension,' funded by the Scientific and Technological Research Council of Turkey. His research interest is computational cognition.