TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics and Mechanics
Vol. 54, Issue II, 2011

A NEURAL FUZZY NETWORK FOR CUSTOMER REQUIREMENTS

Lavinia-Gabriela SOCACIU, Ioan BLEBEA and Maria-Magdolna HARANGOZO

Abstract: *When developing new products it is very important for design teams to understand customer requirements, because the success of products is dependent of customer satisfaction level. The chance of new product's success in a marketplace is higher if the customer is satisfied with it. Usually, the customer requirements are confused, contradictory or incomplete, because they are expressed using linguistic terms. In this paper we make a study, using the fuzzy analytic hierarchy process to determine the weights for customer requirements, and use this information as input in a neural fuzzy network for obtain alternatives the most appropriate to customer requirements.*

Key words: *fuzzy network, customer level*

1. INTRODUCTION

Understanding customer requirements has been recognized as a pressing challenge for companies. Poor understanding of customer requirements and inaccurate assumptions made during the elicitation and analysis of requirement information have a significant negative implication on design and manufacturing of the product in terms of quality, time and cost[3].

In this paper, we construct a fuzzy mathematical model to better match the requirements engineering problems encountered in decision-making. If after a tender process hierarchical fuzzy numbers cut a set of requirements, applying the fuzzy AHP algorithm to determine the set of weights corresponding set of requirements, which then put them in a neural fuzzy network. This way we obtain a set of offers (alternatives) appropriate to requirements. Also in this paper, and the trend shows the evolution process is taking place between the set of demands and offers. Theoretic algorithms AHP and QFD can say that the mathematical model is proposed to optimize the bids according to customer requirements.

2. FUZZY LOGIC: KEY ASPECTS

Boolean logic considers the true value of a sentence in terms of *true* or *false*, there was no possibility of intermediate values. Fuzzy logic was introduced in 1960 by Lotfi Zadeh of UC / Berkeley to model uncertainty of natural language. Regarding this, he proposed an extension of the concept of truth in terms of value between *completely false* and *completely true* [5,6].

Generally speaking, fuzzy logic is a form of artificial intelligence that works with non-quantifiable concepts [2]. It allows programmers to develop software that can simulate the uncertainty and imprecision of terms such as: *small* or *young* [1,6].

Fuzzy Sets can be characterized by a function that shows the membership degree of an item from the sets. If the value of membership function on an element x is 0, then x certainly does not belong to that set. If the value of membership function on x is 1, then x belongs certainly to that set. Otherwise, the membership function values, lying between 0 and 1 shows the status element *x* in relation to the set that might belong [5]. If A is an abstract and non-empty set, called universe of fuzzy elements, membership function μ_A is defined as:

$$\mu_A : A \to [0,1] \tag{1}$$

Consequently, μ_A will be called a membership function (member is its definition field) and its graph will be noted:

$$G_A(\mu_A) = AxI_m(\mu_A) \tag{2}$$

and being composed of fuzzy set pairs $(x, \mu_A(x))$ will be generated by the universe A and its membership function, which must fulfill two minimum conditions:

(a) $\mu_A : A \to [0,1], \exists x_0 \in A$ with: $\mu_A(x_0) = 1$

(b) $\forall x, y \in A$ and $\forall x < z < y$ to have: $\mu_A(z) \geq \min\{\mu_A(x), \mu_A(y)\}$.

If $A \not\subset \Re$, an embedding of *A* in $\Re$ it is made first, i.e. an encoding of *A* elements by real numbers, discrete or continuous type. The embedding of set E in other set F is a mathematical concept which assumes a continuous and injective function $f : E \to F$.

A fuzzy subset $N_A \subset G_A$ is called "fuzzy number", if it satisfies the following conditions:

(i) N_A is convex, of cut $[a_1, a_2]$.

(ii) $\exists x_0 \in [a_1, a_2]$ unique, so $\hat{\mu}_A(x_0) = 1$, i.e. N_A is unique normal.

Since the cut of a fuzzy number N_A is unique, i.e. the range $A_\alpha = [a_1, a_2]$ is unique and has the property $\forall x \in A_\alpha$ that follows $\mu_A(x) \geq \alpha$. And vice versa $\forall(x, \hat{\mu}_A(x)) \in \hat{G}_A$, results that $x \in [a_1, a_2]$. By this embedding it can be supposed that fuzzy number N_A will be confused with his cut, i.e. the following will be considered fuzzy number N_A is given by:

$$N_A = [a_1, a_2] \tag{3}$$

Figure 1 shows the geometric image of a fuzzy number. The particularity of fuzzy membership function, which become fuzzy number is that this fuzzy membership function must be concave, or if possible, to obtain the fuzzy number from a fuzzy cut. In Figure 2 are presented fuzzy numbers of triangular cut type.

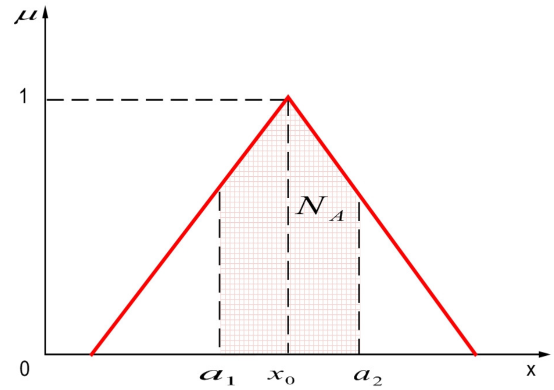

Figure 1. Geometric image of a fuzzy number

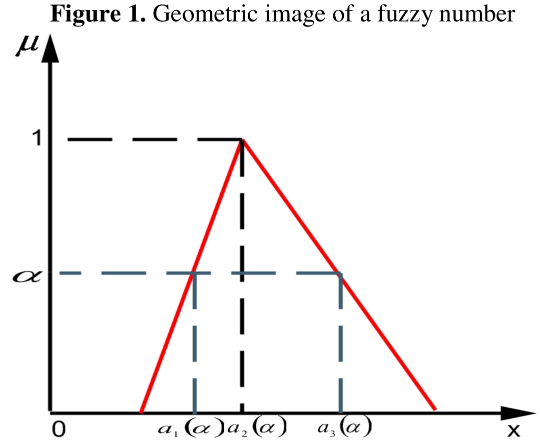

Figure 2. Triangular fuzzy numbers generated by the α cuts

The introduction of fuzzy logic fuzzy numbers on a structure can be achieved in two ways. The first way is to do operations directly with fuzzy numbers, and the second is derived by doing operations on embedding fuzzy numbers in $\Re$. An example of embedding is the convex combinations, i.e. if N_F is the fuzzy number in Figure 2, then it`s embedding in $\Re$ is done by:

$$N_F(\beta) = a_1(\alpha) + \beta(a_3(\alpha) - a_1(\alpha)), \beta \in [0,1] \tag{4}$$

Note that for $\beta = 0, N_F(0) = a_1$, respectively $\beta = 1, N_F(1) = a_3$.

Another way of embedding is to transform itself into the range that is projected. Using expression (4) for a fuzzy number of cuts has the advantage that the defuzzification operation it is not necessary in the end. In addition, we can study economic phenomenon depending on the parameter $\beta \in [0,1]$, which in turn generates a family of fuzzy numbers by (4). In conclusion, on the family of fuzzy numbers it is created a natural fuzzy logic structure [4].

3. FUZZY AHP FOR CUSTOMER REQUIREMENTS

Requirements engineering inputs are represented by the requirements itself, which often are expressed using linguistic terms [3]. When fuzzy theory it is use in applications, the first step you need to go through, is to prepare input data. Whether primary requirements vector:

$$C_R = \begin{bmatrix} C_{R_1} \\ C_{R_2} \\ \vdots \\ C_{R_n} \end{bmatrix} \tag{5}$$

which is associated with fuzzy numbers of cut vector:

$$N_F = \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{bmatrix} \tag{6}$$

Previously, requirements have been ranked according to their importance, then using the average method fuzzy numbers were ordered ascending according (6). Embedding is currently done by:

$$f\left(C_{R_i}\right) = N_i, i = 1 \div n \tag{7}$$

In this way we can pass to the hierarchy of requirements in classes, drawing the ranking matrix and determining values and eigenvectors for the ranking matrix passing through (7) and embedding relations (4).

Ranking matrix has the following form:

$$R\tilde{M} = \begin{bmatrix} 1 & \tilde{a}_{12} & \dots & \tilde{a}_{1n} \\ \tilde{a}_{21} & 1 & \dots & \tilde{a}_{2n} \\ \vdots & \vdots & \dots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \dots & 1 \end{bmatrix} \tag{8}$$

Loading ranking matrix has the following formula:

$$\tilde{a}_{ij}^{\alpha} = \begin{cases} \max\left\{C\tilde{R}_i, C\tilde{R}_j\right\}, i < j \\ \left(\tilde{a}_{ij}^{\alpha}\right)^{-1}, j < i \\ 1, i = j \end{cases} \tag{9}$$

Finally we obtain the weights vector $[w]$, given by:

$$[w] = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \tag{10}$$

$[w]$-The weights of the partial and overall requirements are then used as input data in neural fuzzy network.

4. NEURAL FUZZY NETWORK FOR CUSTOMER REQUIREMENTS

A traditional neural network has the general scheme given in Figure 3. If input data are fuzzy numbers or their embeddings, then in Figure 3 is a neural fuzzy network configuration, as is our case. Typically, neural fuzzy networks uses as models linguistic terms, fuzzy numbers or fuzzy intervals [1]. For nodes configuration function was used membership function Bell.

To become output, the neural fuzzy system inputs suffer several transformations. Figure 4 is the data flow of afferent neural fuzzy system. First, the fuzzification process converts each entry in the system through a membership function.

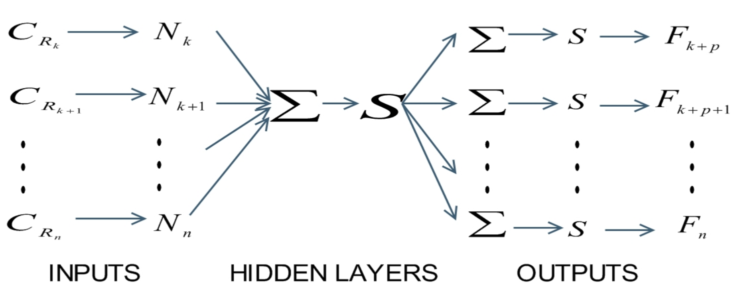

Figure 3. Neural network configuration

Fuzzification consists of calculating (assign) values to represent a factor of membership in qualitative groups named fuzzy sets. Membership factor is determined by the membership function that is defined on the basis of intuition or experience researcher. This process allows connection with the membership functions of linguistic terms.

Membership functions can be chosen from a variety of forms. Membership function doesn't have a significant influence over the system response. Fuzzy algorithms are flexible in terms of choices over fuzzy levels associated to linguistic values. Membership functions will be defined in a general flexible form that allows customizing them according to context. System behavior can be modified through natural language, and allows concise descriptions of complex tasks.

The rules included in database are evaluated by awareness of belonging factors used to form the weights of output. These rules have the form of instructions like IF, THEN. IF statement contains one or more conditions, named antecedent, while THEN contains one or more actions named consequences. History rules are linked directly to membership factors, calculated in the fuzzification phase. Fuzzy algorithm activates fuzzy rules. Each output of fuzzy rules is fuzzy values resulting from the basic operations on fuzzy logic. Each fuzzy rule is a logical operation built using a conjunction operator, after this at the exit point is obtained at punctual minimum of membership functions on the whole definition the main of the output variables. Because only significant rules are being retained, many of the fuzzy rules are not used.

Application of neural fuzzy systems is especially suitable if the system uses a mathematic model difficult to deduce, for example when input values are vague, imprecise or difficult to deduce. Using neural fuzzy networks eases the decision making process when using estimated values for inaccurate information (if the adopted decision is incorrect, it can be changed later when we have more information). Fuzzy models allow the representation of descriptive or qualitative expressions, which are then transformed into symbolic instructions.

Neural fuzzy network is similar to standard multilayer network it with the addition, have direct connections between input and output nodes. Activation of these nodes is discrete and can take the values 1, 0 or -1.

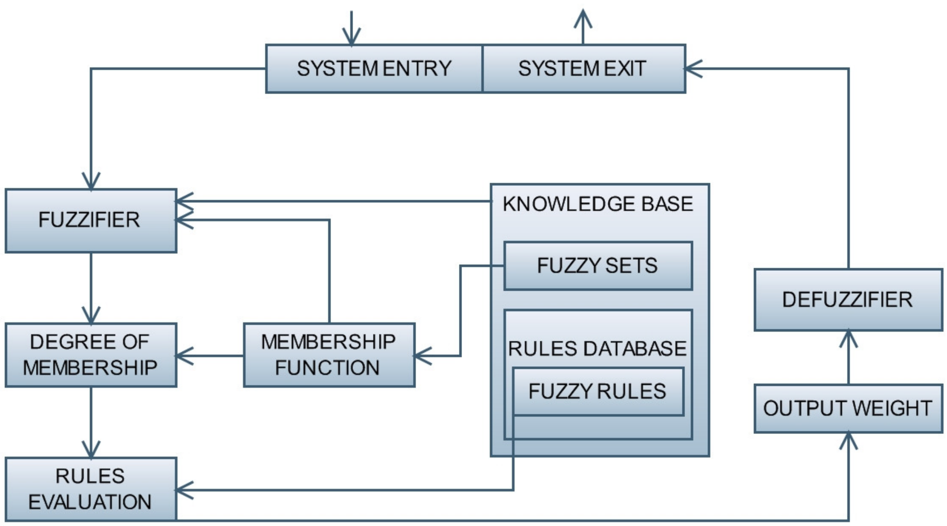

Figure 4. The data flow of neural fuzzy system

Neuronal evolution H_i is:

$$H_i = \begin{cases} +1 or TRUE(SH > 0 \\ 0 or UNKNOWN(SH = 0) \\ -1 or FALSE(SH < 0) \end{cases} \quad (11)$$

Enabling the hidden neurons H_i can be done using the formula:

$$SH_j = \sum_{i=0}^{p} w_{ij} I_j \quad (12)$$

where I_i, $i = 1 \div p$ is the activations of input nodes.

Output neuron state is calculated using the formula:

$$O_k = \begin{cases} +1 or TRUE(SO_K > 0 \\ 0 or UNKNOWN(SO_K = 0) \\ -1 or FALSE(SO_K < 0) \end{cases} \quad (13)$$

Enabling output neuron:

$$SO_K = \sum_{i=0}^{p} u_{k_i} I_j + \sum_{j=0}^{q} v_{k_j} H_j \quad (14)$$

In neural fuzzy network is included the average cost of the group i of requirements, calculated by:

$$C_{m_i} = \frac{\sum_{j=i+1}^{n_i} w_j \cdot c_j}{\sum_{j=i+1}^{n_i} w_j} \quad (15)$$

Where c_j is the cost in fuzzy numbers or embedding them in $\Re$ requirements j, w_j represents the weights of requirements, k and i show us that the number of the first k, respectively i requirements are not to be taken into account, since they have a negligible influence.

5. CONCLUSIONS

Using fuzzy logic leads us to identify alternatives to a problem and allows us to draw conclusions based on vague information, ambiguous, imprecise. The fuzzy techniques adopt a human similar reasoning, allowing us to build robust and feasible systems. Neural fuzzy systems allow us to change results in a smooth and continuous way, regardless input type.

This model known as AHP algorithm coupled with QFD is one of the most important methods to optimize the offers proposed by the developer for offering customers requirements related to a product. In this process the developer can occur, using his experience he can quit the first *k*, respectively *i*, requirements that does not influence the final cost.

6. ACKNOWLEDGEMENTS

Paper prepared under the project with the title: "Proiect de dezvoltare a studiilor de doctorat in tehnologii avansate-PRODOC" contract POSDRU 6/1.5/S/5.

7. REFERENCES

1. S., Abe, *Neural Networks and Fuzzy Systems*, Kluwer Academic Publishers, (1997).
2. G., Bojadziev, *Fuzzy Logic for Business, Finance and Management,* World Scientific Publishing Company, (2007).
3. N., Dzamashvili, *Understanding and supporting requirements engineering decision in market-driven software product development*, Blekinge Institute of Technology, (2010).
4. S., Tzafestas, *Fuzzy Logic and neural network handbook*, Ed C.H,. (2000).
5. L., Zadeh., *Fuzzy sets*, Information and control, (1965).
6. L.,Zadeh, *Fuzzy logic*, IEEE Computer vol 21, (1988).

REȚEA NEURO-FUZZY PENTRU CERINȚELE CLIENTULUI

Rezumat: Atunci când dezvoltăm produse noi este foarte important pentru echipa de proiectare să înțeleagă cerințele clientului, deoarece succesul unui produs este dependent de nivelul de satisfacție al clientului. Şansa ca un produs nou să aibă succes pe piață este mai mare, atunci când clientul este satisfăcut de acesta. De obicei, cerințele clientului sunt confuze, contradictorii sau incomplete, deoarece ele sunt exprimate cu ajutorul termenilor lingvistici. În această lucrare, am realizat un studiu utilizând procesul de analiză ierarhic fuzzy pentru a determina ponderile de importanță pentru cerințele clientului, şi am folosit aceste informații ca şi date de intrare în rețeaua neuro-fuzzy pentru a obține alternativele cele mai apropiate de cerințele clientului.

Socaciu Lavinia-Gabriela, Technical University of Cluj-Napoca, lavinia.socaciu@muri.utcluj.ro
Blebea Ioan , Technical University of Cluj-Napoca, Romania, blebea@muri.utcluj.ro
Harangozo Maria-Magdolna Technical University of Cluj-Napoca, maria.harangozo@muri.utcluj.ro