

CONTRIBUTIONS REGARDING THE DESIGN OF SOME NEURO-FUZZY NEURAL NETWORKS APPLICABLE IN THE STRATEGIC MANAGEMENT FOR ORGANIZATIONS SPECIALISED IN NONCONVENTIONAL TECHNOLOGIES

Marinescu Simona-Ioana¹, Curteanu Silvia²

¹ Ph.D. Student, "Lucian Blaga" University of Sibiu, "Hermann Oberth" Engineering Faculty, Strada Emil Cioran, nr.4, Sibiu, Romania, simona.marinescu19@gmail.com

² Professor, Dr. Eng., "Gheorghe Asachi" Technical University Iași, Department of Chemical Engineering, Bd. Dimitrie Mangeron, No. 71A, 700050, Iași, Romania, silvia_curteanu@yahoo.com

ABSTRACT: This paper presents, in particular, some results obtained from modeling neural networks of neuro-fuzzy type and the similarity with the results obtained using neural feed-forward type networks. The application is developed to the level of organizational management regarding the establishing of a future strategy of a commercial organization. It was designed a neuro-fuzzy network with five input variables of type technical and economical indicators and an exit strategy type. Solving the problem was made using Matlab system.

KEYWORDS: input variables, output variables, neural network, neuro-fuzzy network, feed-forward network, Matlab system.

1. INTRODUCTION

The basic idea based on neural network modeling consists in applying an accessible methodology that will lead to very simple networks, but that will provide very precise predictions[1]. For this purpose, the attempts of modeling took into account the following aspects:

-various softwares were used: Matlab and NeuroSolutions, comparing the results both in terms of their accuracy, and procedure;

-different types of neural networks were tested, mainly from the category of feed-forward neural networks (multilayer perceptron, generalized feedforward, modulation) and neuro-fuzzy;

-five input variables were taken into consideration: risk factor, x_1 , compensation potential in terms of financial results, x_2 , potential for technological innovation, x_3 , competence in market analysis, x_4 , competence to sell the products, x_5 . The proper ranges of these variables were: $x_1 \in [0.1...0.4]$, $x_2 \in [0.1...0.4]$, $x_3 \in [0.1...0.3]$, $x_4 \in (0...0.2]$, $x_5 \in (0...0.2]$, these statements corresponding to "low", "medium" and "high";

-the output variable was the strategy type that has been defined by combining the values that the input parameters take, respectively: offensive strategy, defensive strategy, absorbing strategy, interstitial strategy and incorrect strategy; the outputs of the five variables were associated with the numerical values 1, 2, 3, 4 and 5;

-also, other issues have been formulated, with fewer inputs, respectively the following three cases, with

three input variables each. Case 1 has inputs x_1 , x_3 and x_4 , case 2 - x_2 , x_4 and x_5 and case 3 - x_3 , x_4 , x_5 .

Solving the neuro-fuzzy problem was done using Matlab.

MATLAB, using specialized toolboxes, Fuzzy Logic Toolbox- FLT creates the possibility of implementing techniques based on fuzzy logic, using FUZZY and FUZZYDEMOS. The FUZZY subdirectory contains function type files, grouped into categories of functions and operations [2]:

- functions for graphical user interface (GUI);
- editing functions for fuzzy inference system (FIS), membership functions, the rules used, the diagrams and the associated control surfaces; functions for generating FIS (by); functions for the implementation of other routines (FIS Sugeno type, type C-means clusters, etc.);
- operations that relate to the difference between two membership functions with different forms (sigmoid, Gaussian, trapezoidal, triangular, etc.), to the concatenation of matrices, to the mesh of the FIS's, to the evaluation of the multiple membership functions etc.

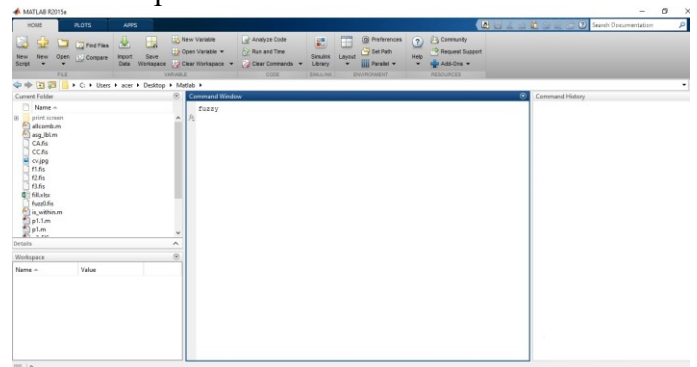


Figure 1 Command window

Fuzzy toolbox is accessed by typing "fuzzy" in the command window (Figure 1).

The system displays the FIS type editor, which processes the corresponding information of the systems based on fuzzy inference. At the top, the diagram of the system to be created is displayed; the entry and the exit are marked (Figure 2).

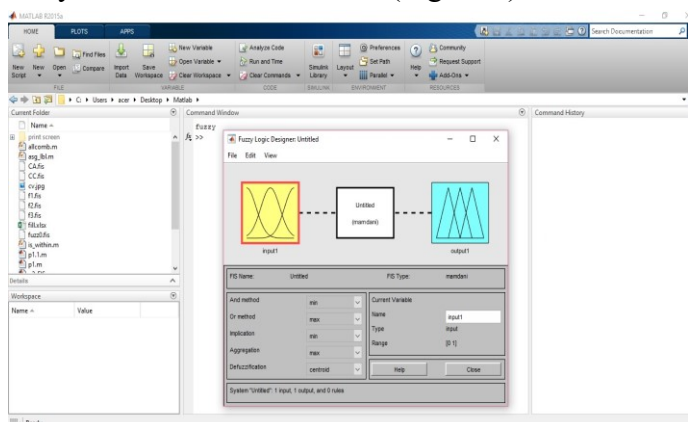


Figure 2 „FIS” Editor

It should be mentioned that the user can define multiple input and output variables.

Entering input and output variables is realized in the following way: in the Edit menu select Add Variable - Input (for adding input variables) or Output (for adding output variables) (Figure 3).

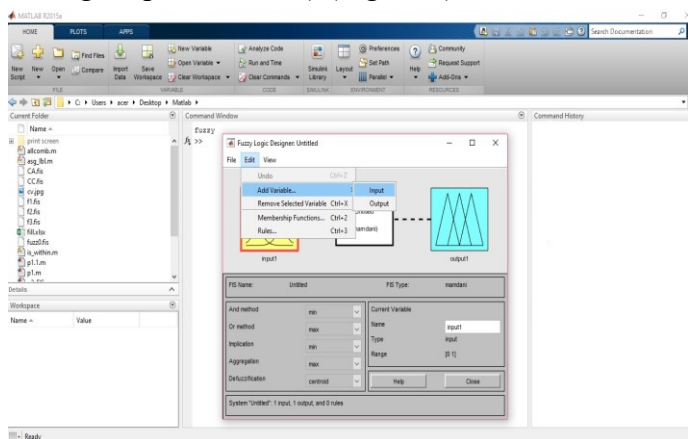


Figure 3 Selecting inputs/outputs

To delete these variables you select the variable you want to delete and from the Edit menu and you select Remove Selected Variable.

After entering the variables, their names are established: the variable's box is being selected and its name is being entered in the Name box.

After defining the name of the input and output variables, you define the membership functions and the universe of discourse for each variable. After that, you select the variable to be configured (from the Edit menu select Membership Functions) - Figure 4.

The editor of the membership functions (Membership Function Editor) (Figure 5) is used to

create, cancel or modify the membership functions of the fuzzy system.

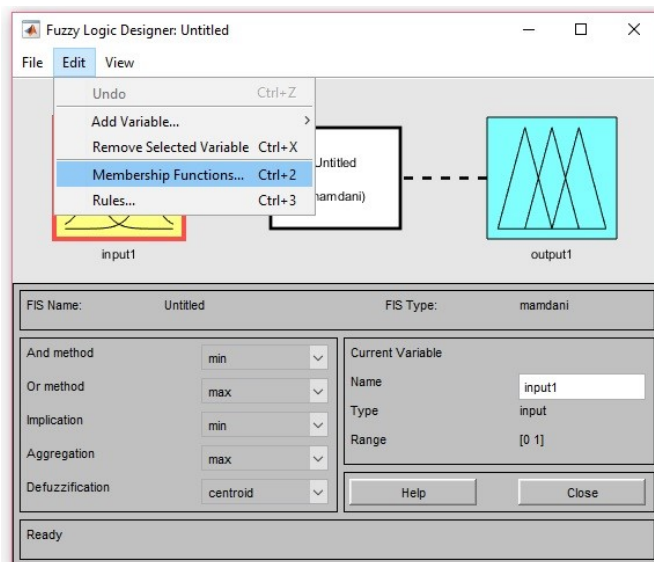


Figure 4 Selecting membership functions

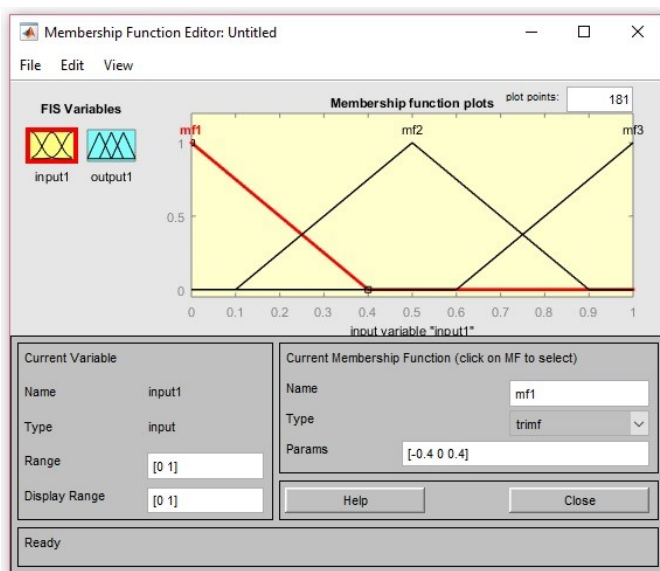


Figure 5 „Membership Function” Editor

Synthetic, defining the membership function involves the following steps:

- from the Edit menu of the Membership Function Editor graphic interface you select Add MFS;
- in the Membership Functions dialog box you select the number of membership functions that the variable can have;
- in the Membership Functions box you determine the overall shape of the membership functions (triangular - triumph, trapezoidal – trapmf, bell - gbellmf, Gaussian - gaussmf etc.);
- the universe of discourse is defined in the Range box;
- in the Display Range box, the user can choose to display the entire universe of discourse (in which case the same numbers as in Range box are entered), or to display a single sequence within the universe of discourse (in which case the numbers within the range displayed in the Range box are entered);

- parameters that define the geometry of the membership function are set in the Params box-figure 6.

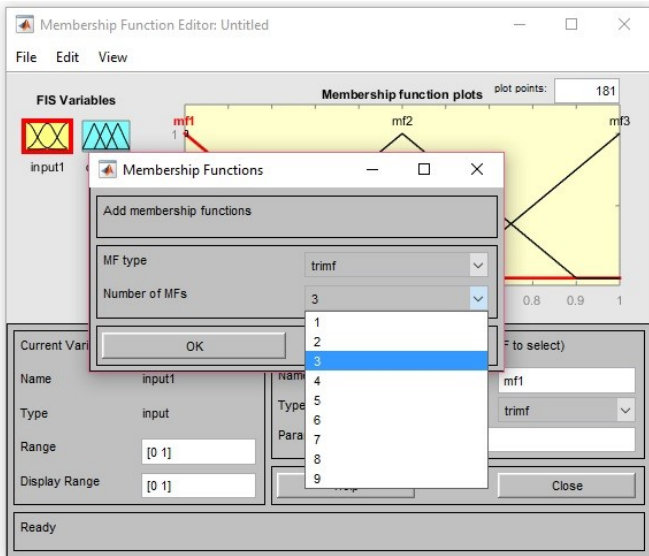


Figure 6 Number of membership functions

- for defining the functions, you select them one at a time and assign them proper names.
- To redefine the shape of each function, you select one function at a time, after which its geometric profile is selected from the "Type" down list belonging to the Membership Function Editor (Figure 7); to clear a function, you select it and from the Edit menu you choose Remove Selected MF.

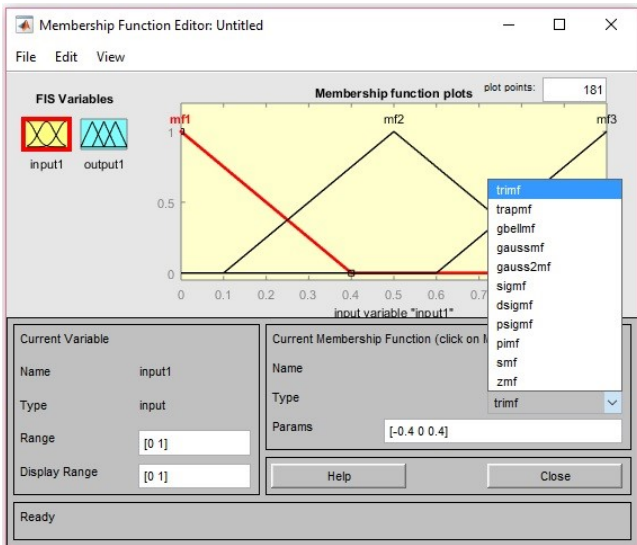


Figure 7 Changing the shape of a membership function

After defining the input and / or output variables, follows editing the rules for the Fuzzy system:

- you open one of the FIS Editor or Memberships Function Editor or Editor windows;
- from the Edit menu you select "Rules"; the program will display the Rule Editor edit window (Figure 8). As the rules are written, they will be displayed.

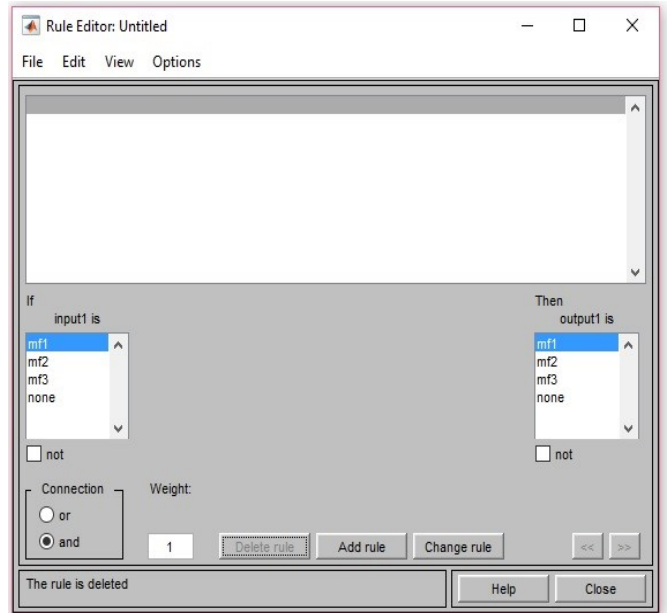


Figure 8 "Rule" Editor

To define the rules, you follow the next steps:

- you select the appropriate function to be edited from the entry list;
- you click on the Add rule button and the rule will be automatically edited in the upper window of the rules editor (Figure 9).

To delete / modify one rule, you select it and then, subsequent, one of the "Delete rule" (for deletion) or "Change rule" (for change) buttons will be operated.

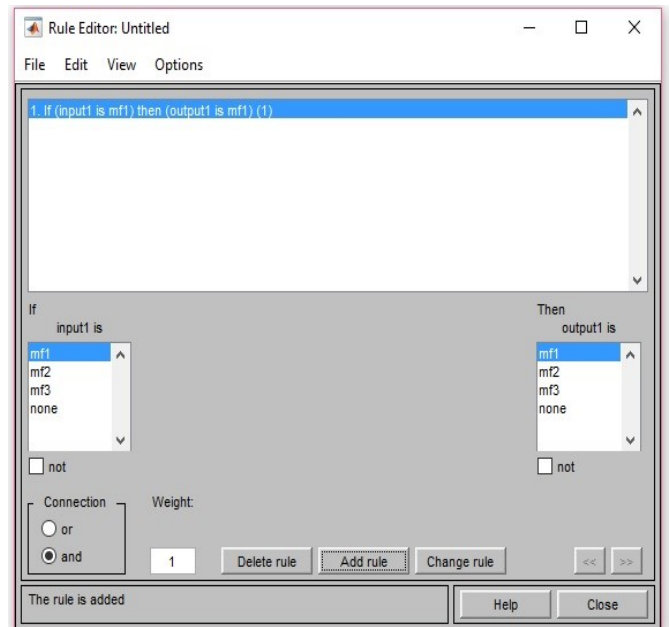


Figure 9 Define rules

To view the rules or surfaces, you select from the View menu one of buttons "Rules" or "Surface" (Figure 10).

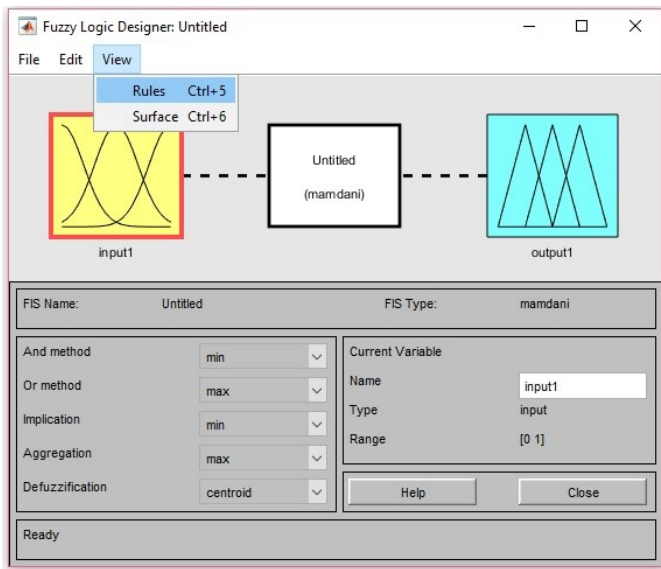


Figure 10 Selecting viewing rules or surfaces

2. MODELING WITH NEURAL NETWORKS, NEURO-FUZZY TYPE

To model the relationship between indicators and strategy, a draft version of the model was to develop a neuro-fuzzy model [3]. Defining the membership functions and the corresponding values are the most important steps in defining the model. After training,

the structure of the rules of the model is obtained. When the user changes the input values, the system automatically generates the output value.

A surface graph showing the relationship between inputs (indicators) and exit (strategy) is shown in Figure 1 which illustrates the locations of the fuzzy inferences, obtained for each variable fuzzy output (strategies) expressed by the first 2 entries (risk factor and compensation potential regarding the financial representative). The fuzzy inference spaces visually express the dependence of the fuzzy output variables, towards the fuzzy input variables on all the support area of the membership functions [4].

Using the neuro-fuzzy model several runs were made, the aim being to determine the strategy is case in which different values of the input parameters are being considered. Table 1 shows the obtained results. From the statistical point of view, for the 18 considered cases, different strategies have been obtained: offensive (1), defensive (4) interstitial (8) and incorrect (5).

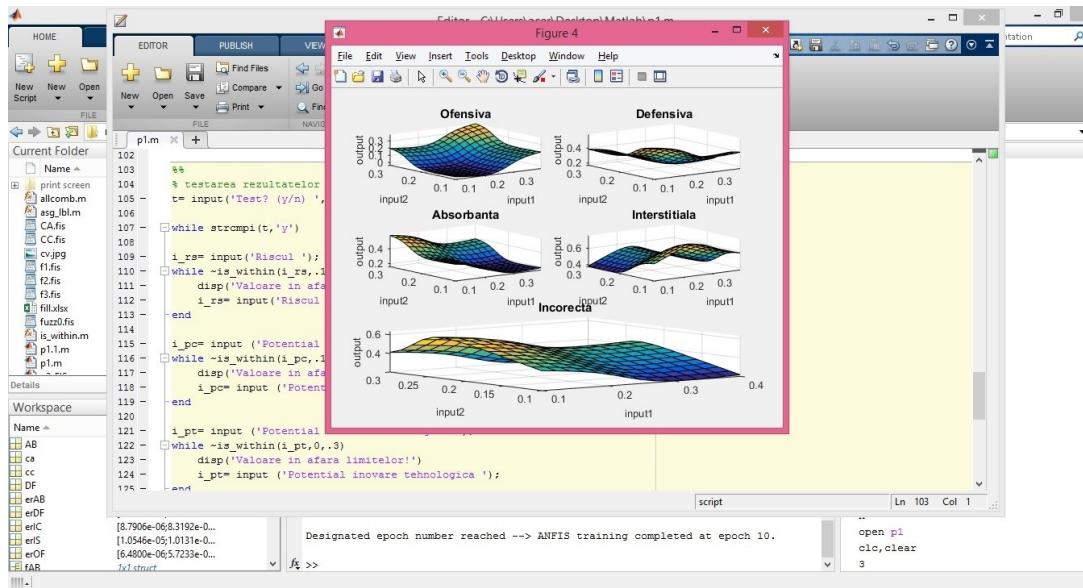


Figure 11 Fuzzy inference systems view

Table 1 Results of neuro-fuzzy model

Case	Parameters					Resulted strategy
	x1	x2	x3	x4	x5	
1	0,15	0,15	0,15	0,15	0,15	Defensive
2	0,35	0,15	0,15	0,05	0,05	Incorrect
3	0,15	0,15	0,25	0,15	0,05	Defensive
4	0,32	0,10	0,12	0,15	0,05	Interstitial
5	0,3	0,2	0,25	0,01	0,1	Incorrect
6	0,4	0,1	0,12	0,1	0,1	Interstitial
7	0,38	0,15	0,1	0,2	0,05	Incorrect

8	0,4	0,3	0,3	0,2	0,2	Offensive
9	0,3	0,1	0,01	0,01	0,01	Interstitial
10	0,15	0,15	0,15	0,2	0,1	Defensive
11	0,25	0,25	0,15	0,2	0,2	Defensive
12	0,3	0,12	0,1	0,1	0,1	Interstitial
13	0,35	0,2	0,05	0,15	0,05	Incorrect
14	0,3	0,1	0,25	0,15	0,1	Interstitial
15	0,3	0,1	0,1	0,15	0,2	Interstitial
16	0,32	0,25	0,15	0,05	0,05	Incorrect
17	0,36	0,28	0,08	0,12	0,06	Interstitial
18	0,35	0,15	0,15	0,1	0,1	Interstitial

Given the fact that two modeling techniques have been applied, obtaining neural models with feed-forward type networks and neuro-fuzzy models, a comparison between them was necessary, both in terms of accuracy of results and applied methodology. Table 2 shows some example of predictions, pointing out that both models give the same results.

In the modeling with feed-forward neural networks, a work algorithm was established, which takes into account, gradually, the possibilities to improve the models' performance. Such attempts were: • testing different types of neural networks; • designing various topologies (number of hidden layers and

number of neurons); • considering different sets of input data as number of entries (5 or 3); • different coding of outputs, depending on the chosen variant of modeling (regression or classification); • using different databases, expanded or collapsed (number of values); • dividing in different percentages in training and testing data; • using a different number of driving epochs.

Under these circumstances, the performances for training and testing for different models were recorded, of which the following have been selected for illustration: MLP (5: 40: 20: 1), MLP (5: 30: 15: 1), MLP (5: 12: 4: 1), MLP (5: 5: 1).

Table 2 Comparison of the predictions made by the neural network MLP (5: 5: 1) and the neuro-fuzzy model

Experiment no.	Parameters					Experimental strategy	NN Strategy	Neuro-fuzzy strategy
	x1	x2	x3	x4	x5			
1	0.15	0.15	0.15	0.15	0.15	2	2	2
2	0.35	0.15	0.15	0.05	0.05	5	5	5
3	0.15	0.15	0.25	0.15	0.05	2	2	2
4	0.32	0.1	0.12	0.15	0.05	4	4	4
5	0.3	0.2	0.25	0.01	0.1	5	5	5
6	0.4	0.1	0.12	0.1	0.1	4	4	4
7	0.38	0.15	0.1	0.2	0.05	5	5	5
8	0.4	0.3	0.3	0.2	0.2	1	1	1
9	0.3	0.1	0.01	0.01	0.01	4	4	4
10	0.15	0.15	0.15	0.2	0.1	2	2	2
11	0.25	0.25	0.15	0.2	0.2	2	2	2
12	0.3	0.12	0.1	0.1	0.1	4	4	4
13	0.35	0.2	0.05	0.15	0.05	5	5	5
14	0.3	0.1	0.25	0.15	0.1	4	4	4
15	0.3	0.1	0.1	0.15	0.2	4	4	4
16	0.32	0.25	0.15	0.05	0.05	5	5	5
17	0.36	0.28	0.08	0.12	0.06	4	4	4
18	0.35	0.15	0.15	0.1	0.1	4	4	4

The best model was MLP (5: 5: 1), trained at 10,000 epochs, which provided 100% correct answers to the test. The results were also verified through formulating a classification problem.

Also, in this case, the MLP model (5: 5: 5), with binary coded outputs had the best results, 100% correct answers.

In the considering cases of three input variables (in different variations) instead of five, the results are not too good, the percentage of correct answers was 59%, 73% and 94%.

Weaker results have been obtained, especially the smaller percentages from the previous listing can be attributed to the removal of significant variables for some strategies, that, as a result, have been wrongly classified.

Once again, this modeling is an argument for the complete case with five entries, respectively, for the fact that the 5 initial entries determine the considered strategies (100% correct answers).

3. CONCLUSIONS

Neuro-fuzzy models were designed using the Matlab system. Under this strategy, the most important steps are represented by the definition of the membership functions and the corresponding values. Subsequently, after the completion of the training phase, the model's structure of the rules is being obtained, and then the model can be used to make predictions for different input data sets.

A comparison between the neuro-fuzzy model and the MLP neural model (5:5:1) highlights the fact that the two types of patterns generate identical results. In these circumstances, the tools that have generated them are left for comparison, respectively Matlab and NeuroSolutions. Both are equipped with specialized software graphical user interface, choosing between the two methods is up to the user, depending on the preference and his ability.

4. BIBLIOGRAPHY

- [1] HECHT-NIELSEN, R., "Neurocomputing", Addison/Wesley, 1989
- [2] Curteanu, S., "Inițiere în Matlab", Editura Polirom, Iași, 2008
- [3] Marinescu S.I.- Research on Applications of Neural Networks in Organizational Management. In ACTA Universitatis Cibiniensis, Vol. 65, Issue 1, pg.64-68. 2014, ISBN (online) 1583-7149, DOI: 10.1515/ancts-2015-0011, 2015, de Gruyter.
- [4] Marinescu, S.I., Titu, M.- Aspects Regarding the Possibility to Use "Neural Networks" in the

Selection of the "R&D" Strategy in the "Nonconventional Technologies" Field. In ACTA Universitatis Cibiniensis, Vol67, Issue 1, Pg 179-184, 2015, ISSN(online) 1583-7149, DOI: 10.1515/ancts-2015-0086, 2015, de Gruyter

- [5] Simon Haykin - Neural Networks: A Comprehensive Foundation. Macmillan/IEEE Press, 1994.
- [6] Neagu, C., Ioniță, C.- Neural Networks. Theory and Applications in Modelling and Simulation of Manufacturing Processes and Systems. Academy Press Publisher, 2010.
- [7] Toderan, H.N., Chelaru M., Gâldea, D., Nistor, S., Tofan, I.-Fuzzy Systems and Applications. Politechnic Institute, Iași, 1989.
- [8] Sharma A., Chopra A., Artificial neural networks: „Applications in Management”, IOSR Journal of Business and Management, 12(3), 32-40, 2013.
- [9] Hakimpoor H., Arshad K., et. al."Artificial Neural Networks' Applications in management", World Applied Sciences Journal 14(7), 1008-1019, 2011.
- [10] Dubois, D., Prade, H., "Fuzzy sets and Systems: Theory and Applications." Academic Press, New York, 1980
- [11] Oprean, C., Titu, M., Bucur, V.- Global Management of an Organization Based on Knowledge, Agir Publisher, Bucharest, 2011.
- [12] Dzitac, I.- Artificial Intelligence, Aurel Vlaicu University Publisher, Arad, 2008
- [13] Keropyan, A., Gil-Lafuente, A.M., "A fuzzy-based decision model application on strategic management", African Journal of Business Management, Vol.5(15), pp.6586-6590. 4August, 2011, <https://pdfs.semanticscholar.org/d168/27709db03d916cdabaa031ece8e00cfd1c93.pdf>
- [14] Widrow, B., Lehr, M.A.R., "Adaptive Neural Networks and their Applications," *International Journal of Intelligent Systems*, 8(4):453-507, April 1993.
- [15] Brown, C., O'Leary, D. - Introduction to artificial intelligence and expert systems. International Journal of Intelligent Systems, 1995.