

# Application of the Intuitionistic Fuzzy InterCriteria Analysis Method to a Neural Network Preprocessing Procedure

Sotir Sotirov<sup>1</sup> Vassia Atanassova<sup>2</sup> Evdokia Sotirova<sup>1</sup> Veselina Bureva<sup>1</sup> Deyan Mavrov<sup>1</sup>

<sup>1</sup> Laboratory of Intelligent Systems, University "Prof. Dr. Assen Zlatarov"  
1 "Prof. Yakimov" Blvd., Burgas 8010, Bulgaria

e-mails: ssotirov@btu.bg, esotirova@btu.bg, vbureva@btu.bg, dg@mavrov.eu

<sup>2</sup> Department of Bioinformatics and Mathematical Modelling  
Institute of Biophysics and Biomedical Engineering  
105 "Acad. G. Bonchev" Str., Sofia 1113, Bulgaria  
e-mail: vassia.atanassova@gmail.com

## Abstract

The artificial neural networks (ANN) are a tool that can be used for object recognition and identification. However, there are certain limits when we may use ANN, and the number of the neurons is one of the major parameters during the implementation of the ANN. On the other hand, the bigger number of neurons slows down the learning process. In our paper, we propose a method for removing the number of the neurons without reducing the error between the target value and the real value obtained on the output of the ANN's exit. The method uses the recently proposed approach of InterCriteria Analysis, based on index matrices and intuitionistic fuzzy sets, which aims to detect possible correlations between pairs of criteria.

**Keywords:** Neural network, InterCriteria Analysis, Intuitionistic fuzziness.

## 1. Introduction

The many difficulties in the neural network usage such as the large number of neurons in the perception of the individual values, proportional memory necessary for their training, computing power of training and therefore their dependence on time to train, have forced scientists to search for better methods for their training. BackPropagation is the main method by which neural networks are trained with uplink (Multi-Layer Perceptron). There are many other methods that accelerate the neural networks' training [11], or reduce memory usage, which in turn helps to shorten used computing power and hence training time.

In the stage of preprocessing, the data at the input of the neural network can be used as a constant threshold value, as it was done in [12] to distinguish static from dynamic activities, and thereby the amount of incidental values due on unforeseen circumstances is reduced.

Another approach is to use a wavelet-based neural-network classifier with which to reduce the power interference in the training of the neural network or randomly stumbled measurements [16]. Here the discrete

wavelet transform (DWT) technique is integrated with neural network to build a classifier.

Particle Swarm Optimization (PSO) is an established method for parameter optimization. It represents a population-based adaptive optimization technique that is influenced by several "strategy parameters". Choosing reasonable parameter values for the PSO is crucial for its convergence behavior, and depends on the optimization task. In [13] is presented a method for parameter meta-optimization based on PSO and its application to neural network training. The concept of the Optimized Particle Swarm Optimization (OPSO) is to optimize the free parameters of the PSO by having swarms within a swarm.

It is essential in the usage of neural networks that the amount of the neurons in the hidden layer to be reduced, thereby reducing the number of weight coefficients of the neural network as a whole. This leads to a smaller dimension of weight matrices, and hence the amount of used memory. As an additional property appears the less computing power which is used and the shortened time for training.

In this paper, we use the integration of intuitionistic fuzzy InterCriteria Analysis method for reducing the number of input parameters and Multi-Layer Perceptron. This will allow reduction of the weight matrices, implementation of the neural network in limited hardware, and will save time and resources in training.

The very important goal in testing the neural network after reducing some of the data (respectively the number of inputs) is to obtain an acceptable relation between the input and output values, as well as the average deviation (or match) of the result.

## 2. Presentation of the InterCriteria Analysis

The presented method, titled InterCriteria Analysis (ICA) [12] is based on two fundamental concepts: intuitionistic fuzzy sets and index matrices.

Intuitionistic fuzzy sets (IFSs) defined by Atanassov [3, 4, 5, 6] represent an extension of the concept of fuzzy sets, as defined by Zadeh [15], exhibiting function  $\mu_A(x)$  defining the membership of an element  $x$  to the set  $A$ , evaluated in the interval  $[0; 1]$ . The difference

between fuzzy sets and intuitionistic fuzzy sets (IFSs) is in the presence of a second function  $v_A(x)$  defining the non-membership of the element  $x$  to the set  $A$ , where  $\mu_A(x) \in [0; 1]$ ,  $v_A(x) \in [0; 1]$ , under the condition of  $(\mu_A(x) + v_A(x)) \in [0; 1]$ . The IFS itself is formally denoted by:

$$A = \{ \langle x, \mu_A(x), v_A(x) \rangle \mid x \in E \}.$$

Comparison between elements of any two IFSs, say  $A$  and  $B$ , involves pairwise comparisons between their respective elements' degrees of membership and non-membership to both sets.

The second concept on which the proposed method relies is the concept of index matrix, a matrix which features two index sets. The theory behind the index matrices is described in [1]. Here we will start with the index matrix  $M$  with index sets with  $m$  rows  $\{O_1, \dots, O_m\}$  and  $n$  columns  $\{C_1, \dots, C_n\}$ , where for every  $p, q$  ( $1 \leq p \leq m, 1 \leq q \leq n$ ),  $O_p$  is an evaluated object,  $C_q$  is an evaluation criterion, and  $e_{O_p, C_q}$  is the evaluation of the  $p$ -th object against the  $q$ -th criterion, defined as a real number or another object that is comparable according to relation  $R$  with all the rest elements of the index matrix  $M$ .

$$M = \begin{array}{c|cccccc} & C_1 & \dots & C_k & \dots & C_l & \dots & C_n \\ \hline O_1 & e_{O_1, C_1} & \dots & e_{O_1, C_k} & \dots & e_{O_1, C_l} & \dots & e_{O_1, C_n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ O_i & e_{O_i, C_1} & \dots & e_{O_i, C_k} & \dots & e_{O_i, C_l} & \dots & e_{O_i, C_n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ O_j & e_{O_j, C_1} & \dots & e_{O_j, C_k} & \dots & e_{O_j, C_l} & \dots & e_{O_j, C_n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ O_m & e_{O_m, C_1} & \dots & e_{O_m, C_j} & \dots & e_{O_m, C_l} & \dots & e_{O_m, C_n} \end{array}.$$

From the requirement for comparability above, it follows that for each  $i, j, k$  it holds the relation  $R(e_{O_i, C_k}, e_{O_j, C_k})$ . The relation  $R$  has dual relation  $\bar{R}$ , which is true in the cases when relation  $R$  is false, and vice versa.

For the needs of our decision making method, pairwise comparisons between every two different criteria are made along all evaluated objects. During the comparison, it is maintained one counter of the number of times when the relation  $R$  holds, and another counter for the dual relation.

Let  $S_{k,l}^\mu$  be the number of cases in which the relations  $R(e_{O_i, C_k}, e_{O_j, C_l})$  and  $R(e_{O_i, C_l}, e_{O_j, C_k})$  are simultaneously satisfied. Let also  $S_{k,l}^\nu$  be the number of cases in which the relations  $\bar{R}(e_{O_i, C_k}, e_{O_j, C_l})$  and its dual  $\bar{R}(e_{O_i, C_l}, e_{O_j, C_k})$  are simultaneously satisfied. As the total number of pairwise comparisons between the object is  $m(m-1)/2$ , it is seen that there hold the inequalities:

$$0 \leq S_{k,l}^\mu + S_{k,l}^\nu \leq \frac{m(m-1)}{2}.$$

For every  $k, l$ , such that  $1 \leq k \leq l \leq m$ , and for  $n \geq 2$  two numbers are defined:

$$\mu_{C_k, C_l} = 2 \frac{S_{k,l}^\mu}{m(m-1)}, \quad \nu_{C_k, C_l} = 2 \frac{S_{k,l}^\nu}{m(m-1)}.$$

The pair constructed from these two numbers plays the role of the intuitionistic fuzzy evaluation of the relations that can be established between any two criteria  $C_k$  and  $C_l$ . In this way the index matrix  $M$  that relates evaluated objects with evaluating criteria can be transformed to another index matrix  $M^*$  that gives the relations among the criteria:

$$M^* = \begin{array}{c|ccc} & C_1 & \dots & C_n \\ \hline C_1 & \langle \mu_{C_1, C_1}, \nu_{C_1, C_1} \rangle & \dots & \langle \mu_{C_1, C_n}, \nu_{C_1, C_n} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ C_n & \langle \mu_{C_n, C_1}, \nu_{C_n, C_1} \rangle & \dots & \langle \mu_{C_n, C_n}, \nu_{C_n, C_n} \rangle \end{array}.$$

From practical considerations, it has been more flexible to work with two index matrices  $M^\mu$  and  $M^\nu$ , rather than with the index matrix  $M^*$  of IF pairs.

The final step of the algorithm is to determine the degrees of correlation between the criteria, depending on the user's choice of  $\mu$  and  $\nu$ . We call these correlations between the criteria: 'positive consonance', 'negative consonance' or 'dissonance'. Let  $\alpha, \beta \in [0; 1]$  be the threshold values, against which we compare the values of  $\mu_{C_k, C_l}$  and  $\nu_{C_k, C_l}$ . We call that criteria  $C_k$  and  $C_l$  are in:

- $(\alpha, \beta)$ -positive consonance, if  $\mu_{C_k, C_l} > \alpha$  and  $\nu_{C_k, C_l} < \beta$ ;
- $(\alpha, \beta)$ -negative consonance, if  $\mu_{C_k, C_l} < \beta$  and  $\nu_{C_k, C_l} > \alpha$ ;
- $(\alpha, \beta)$ -dissonance, otherwise.

Obviously, the larger  $\alpha$  and/or the smaller  $\beta$ , the less number of criteria may be simultaneously connected with the relation of  $(\alpha, \beta)$ -positive consonance. For practical purposes, it carries the most information when either the positive or the negative consonance is as large as possible, while the cases of dissonance are less informative and are skipped.

### 3. Artificial Neural Networks

The artificial neural networks [9, 10] are one of the tools that can be used for object recognition and identification. In the first step it have to be learned and after that we can use for the recognitions and for predictions of the properties of the materials. Fig. 1 shows in abbreviated notation of a classic two-layered neural network.

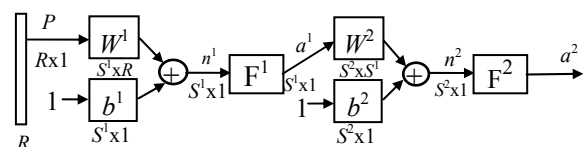


Figure 1: Abbreviated notation of a classical Multi-Layer Perceptron

In the two-layered neural networks, one layer's exits become entries for the next one. The equations describing this operation are:

$$a^2 = f^2(w^2 f^1(w^1 p + b^1) + b^2),$$

where:

- $a^m$  is the exit of the  $m$ -th layer of the neural network for  $m = 1, 2$ ;
- $w^m$  is a matrix of the weight coefficients of the each of the entries of the  $m$ -th layer;
- $b$  is neuron's entry bias;
- $f^1$  is the transfer function of the 1-st layer;
- $f^2$  is the transfer function of the 2-nd layer.

The neuron in the first layer receives outside entries  $p$ . The neurons' exits from the last layer determine the neural network's exits  $a$ .

Since it belongs to the learning with teacher methods, to the algorithm are submitted training set (an entry value and an achieving aim – on the network's exit)

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\},$$

$Q \in (1, \dots, n)$ ,  $n$  – numbers of learning couple, where  $p_Q$  is the entry value (on the network entry), and  $t_Q$  is the exit's value corresponding to the aim. Every network's entry is preliminary established and constant, and the exit have to corresponding to the aim. The difference between the entry values and the aim is the error  $e = t - a$ .

The "back propagation" algorithm [13] uses mean-quarter error:

$$\hat{F} = (t - a)^2 = e^2.$$

In learning the neural network, the algorithm recalculates network's parameters ( $W$  and  $b$ ) so to achieve mean-square error.

The "back propagation" algorithm for  $i$ -neuron, for  $k+1$  iteration use equations:

$$w_i^m(k+1) = w_i^m(k) - \alpha \frac{\partial \hat{F}}{\partial w_i^m};$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial \hat{F}}{\partial b_i^m},$$

where:

- $\alpha$  - learning rate for neural network;
- $\frac{\partial \hat{F}}{\partial w_i^m}$  - relation between changes of square error and changes of the weights;
- $\frac{\partial \hat{F}}{\partial b_i^m}$  - relation between changes of square error and changes of the biases.

The overfitting [8] appears in different situations, which effect over trained parameters and worsen the output results as shown in Fig. 2.

There are different methods that can reduce the overfitting – "Early Stopping" and "Regularization". Here we will use Early Stopping [8].

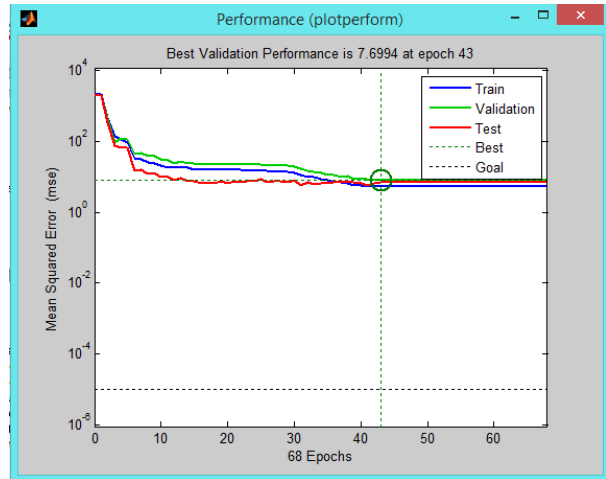


Figure 2: The learning process

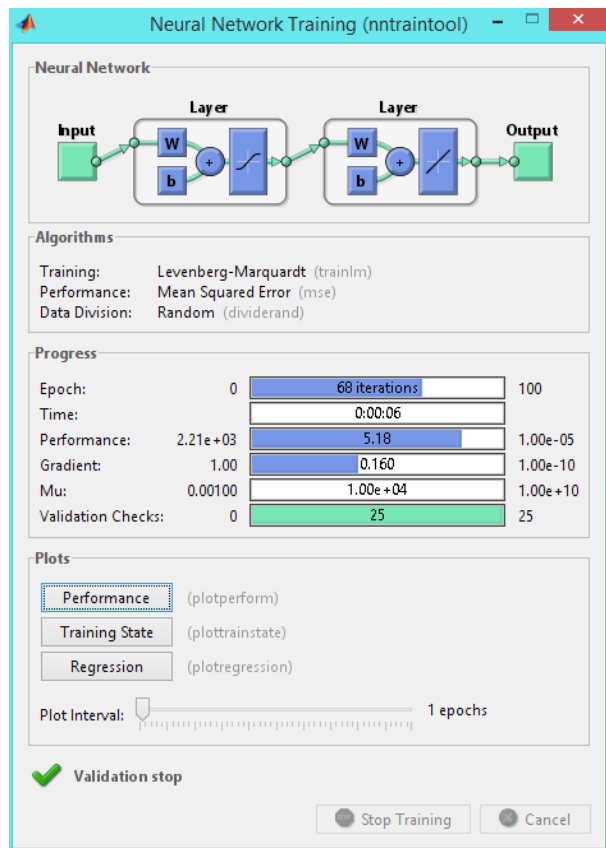


Figure 3: The neural network structure

When the multilayer neural network is trained, usually the available data has to be divided into three subsets. The first subset is named "Training set", is used for computing the gradient and updating the network weights and biases. The second subset is named "Validation set". The error of the validation set is monitored during the training process. The validation error normally decreases during the initial phase of training, as does the training set error. Sometimes, when the network begins to overfit the data, the error of the validation set typically begins to rise. When the validation error increases for a specified number of iterations, the training stops, and the weights and biases at the minimum of the validation error are returned [10]. The last

subset is named “test set”. The sum of these three sets has to be 100% of the learning couples.

When the validation error  $e_v$  increases (the amendment  $de_v$  have positive value) the neural network learning stops when:

$$de_v > 0.$$

The classic condition for the learned network is when

$$e^2 < E_{max},$$

where  $E_{max}$  is the maximum square error.

For the preparing we use MATLAB and neural network structure 8:45:1 (8 inputs, 45 neurons in hidden layer and one output (Fig. 3). The numbers of the weight coefficients are  $9 \times 45 = 405$ .

The proposed method is focused on removing part of the number of neurons (and weight coefficients) and thus not reducing the average deviation of the samples, used for the learning testing and validating the neural network.

#### 4. Testing

On the input of the neural network we put the experimental data for obtaining cetane number, based on certain correlations with the rest criteria of measurement of crude oil. We work with data for 140 crude oil probes, measured against 8 criteria:

- 1 - Density at 15°C  $g/cm^3$ ;
- 2 - 10% (v/v) ASTM D86 distillation, °C ;
- 3 - 50% (v/v) ASTM D86 distillation, °C;
- 4 - 90% (v/v) ASTM D86 distillation, °C;
- 5 - Refractive index at 20°C;
- 6 - H<sub>2</sub> content, % (m/m);
- 7 - Aniline point, °C;
- 8 - Molecular weight g/mol.

The same data we use as input data of the InterCriteria Analysis method, applied on to the whole 140×8 table, and a software applications that implements the ICA algorithm returns the results in the form of two index matrices in Tables 1 and 2, containing the membership and the non-membership parts of the IF correlations detected between each pair of criteria (28 pairs).

$\mu$	1	2	3	4	5	6	7	8
1	1	0.7	0.77	0.66	0.96	0.18	0.45	0.7
2	0.7	1	0.79	0.6	0.68	0.41	0.64	0.78
3	0.77	0.79	1	0.78	0.73	0.39	0.66	0.92
4	0.66	0.6	0.78	1	0.63	0.47	0.67	0.77
5	0.96	0.68	0.73	0.63	1	0.13	0.4	0.66
6	0.18	0.41	0.39	0.47	0.13	1	0.73	0.47
7	0.45	0.64	0.66	0.67	0.4	0.73	1	0.74
8	0.7	0.78	0.92	0.77	0.66	0.47	0.74	1

Table 1: Membership part of the IF pairs, giving the InterCriteria correlations

$\nu$	1	2	3	4	5	6	7	8
1	0	0.29	0.22	0.33	0.04	0.82	0.55	0.3
2	0.29	0	0.2	0.39	0.31	0.58	0.35	0.21
3	0.22	0.2	0	0.21	0.26	0.59	0.32	0.07
4	0.33	0.39	0.21	0	0.36	0.52	0.31	0.21
5	0.04	0.31	0.26	0.36	0	0.87	0.6	0.34
6	0.82	0.58	0.59	0.52	0.87	0	0.27	0.53
7	0.55	0.35	0.32	0.31	0.6	0.27	0	0.26
8	0.3	0.21	0.07	0.21	0.34	0.53	0.26	0

Table 2: Non-membership part of the IF pairs, giving the InterCriteria correlations

The objective of the preparation of the two matrices is to remove one or more columns of parameters such repetitive (with the corresponding index of the positive consonance). Testing is done as in the first step all the measurements of the 140 crude oil probes against the 8 criteria are analyzed in order to make a comparison of the obtained results thereafter. For this comparison to be possible, the predefined weight coefficients and offsets that are normally random values between -1 and 1, are now established and are the same in all studies of the various attempts.

For the learning process, we set the following parameters: Performance (MSE) = 0.00001; Validation check = 25. The input vector is divided into three different parts: Training (70/100); Validation (15/100) and Testing (15/100). For tagret we use the Cetane number ASTM D613.

At the first step of the testing process, we use all the 8 criteria listed above, in order to train the neural network. After the training process all input values are simulated by the neural network.

The average deviation of the all 140 samples is 1.98% (the matching coefficient is 98.02%). The coefficient  $R$  (regression  $R$  values measure the correlation between outputs and targets) obtained from the MATLAB program is 0.9781.

At the second step of the testing process, we make a fork and try independently to remove one of the columns, and experiment with data from the rest seven columns. We compare the results in the next section ‘Discussion’. First, we make a reduction of column 5 (with maximal intercriteria IF pair (0.956012; 0.04193)) and put the data on the input of the neural network.

After the training process all input values are simulated. The average deviation of all the 140 samples is 1.84% (the matching coefficient is 98.16%). The coefficient  $R$  is 0.9790.

At the third step, we alternatively experiment with the reduction of one different column, column 8 (with maximal intercriteria IF pair (0.92148; 0.06773)), and put the data on the input of the neural network.

After the training process all input values are simulated. The average deviation of the all 140 samples is 1.8391% (the matching coefficient is 98.1609%). The coefficient  $R$  is 0.9788.

Now, at the fourth step, we proceed with feeding the neural network with 6 inputs, with the reduction of both columns, 5 and 8, simultaneously, their maximal intercriteria IF pair given above. The average deviation of all the 140 samples is 1.80% (the matching coefficient is 98.2%). The coefficient  $R$  is 0.9795.

At the fifth step, we reduce the number of inputs with one more, i.e. we put on the input of the neural network experimental data from 5 inputs, with removed columns 5, 8, and 4, which maximal intercriteria IF pair is (0.77739, 0.21161). The average deviation of the all 140 samples is 1,83% and the matching coefficient is 98.17%. The coefficient  $R$  is 0.9789.

Finally, at the sixth step, we experiment with the reduction of a fourth column, feeding the neural network with only 4 inputs. After the reduced columns 5, 8 and 4, the fourth reduced column is column 3, which maximal intercriteria IF pair is (0.78674, 0.20442). The average deviation of the all 140 samples is 2.05 (the matching coefficient is 97.95%). The coefficient  $R$  obtained from the Matlab program is 0.9779.

## 5. Discussion

As we stated above, reducing the number of input parameters of a classical neural network leads to reduction of the weight matrices, resulting in implementation of the neural network in limited hardware and saving time and resources in training. For this aim, we use the intuitionistic fuzzy sets-based approach of InterCriteria Analysis (ICA), which gives dependencies between the criteria, and thus helps us reduce the number of input parameters, yet keeping high enough level of precision.

Table 3 below summarizes the most significant parameters of the process of testing the neural network with different numbers of inputs, gradually reducing the number in order to discover optimal results. These process parameters are the NN-specific parameters ‘Average deviation’, ‘Matching coefficient’, ‘Regression coefficient  $R$ ’, and ‘Number of the weight coefficients’, and the ICA-specific parameters: maximal value for  $\mu$  per column and respective value for  $\nu$ , [7].

Number of inputs	Maximal value for $\mu$ per column	Respective value for $\nu$ per column	Average deviation	Matching coefficient	$R$	Number of the weight coefficients
8 inputs	-	-	1.98 %.	98.02%	0.9781	405
7 inputs without column 5	0.95601	0.04193	1.84%	98.16%	0.9788	360
7 inputs without column 8	0.92148	0.06773	1.8391 %	98.1609%	0.9790	360
6 inputs without columns 5 and 8	0.95601 0.92148	0.04193 0.06773	1.80%	98.2%	0.9795	315
5 inputs without columns 4, 5 and 8	0.95601 0.92148 0.77739	0.04193 0.06773 0.21161	1.83 %	98.17%	0.9789	270
4 inputs without columns 3, 4, 5 and 8	0.95601 0.92148 0.77739 0.78674	0.04193 0.06773 0.21161 0.20442	2.05%	97.95%	0.9779	225

Table 3: Table of comparison

The matching coefficient in using 8 input vectors is 98.02% with number of weight coefficients 405. By reducing the number of the inputs the number of weight coefficients is also decreased which theoretically is supposed to reduce the matching coefficient. In this case the removal of column 5 (and therefore one input is removed) causes further increase of matching coefficient to 98.16%. With maximal membership of the intercriteria IF pair (0.956012; 0.04193) for column 5 the additional information used for training the neural network is very little, and the total MSE is less. The result is better compared to the formerly used attempt by training the neural network with 8 data columns.

The use of 7 columns (excluding column 8) leads to the result which is better than the previous one - 98.1609%. This shows that, while maintaining the number of weight coefficients and reducing the maximal membership in the intercriteria IF pair (0.92148; 0.06773), the neural network receives an additional small amount of information which it uses for further learning.

Best results (Matching coefficient = 98.2%) are obtained by removing the two columns with the greatest membership components of the respective IF pairs.

In this case, the effect of reducing the number of weight coefficients from 360 to 315 and the corresponding MSE is greater than the effect of the two columns.

The use of 5 columns (without columns 4, 5 and 8) leads a result which is less than the previous, i.e. 98.17%. This shows that reducing the number of weight coefficients (and the total MSE) and the information at the input of the neural network a small amount of information is lost with which the network is trained. As a result, the overall accuracy of the neural network is decreased.

The worst results (Matching coefficient = 97.95%) are obtained in the lowest number of columns – 4. In this case, columns 3, 4, 5 and 8 are removed. Although the number of weight coefficients here is the smallest, the information that is used for training the neural network is less informative.

## 6. Conclusion

The number of the neurons is one of the major parameters during the realization of the ANN. Here we use the integration of intuitionistic fuzzy InterCriteria Analysis method for reducing the number of input parameters of the classical neural network. This leads to a reduction of the weight matrices, and thus allow implementation of the neural network in limited hardware and saving time and resources in training.

Very important aspect of the testing of the neural network after reducing some of the data (respectively the number of inputs) is to obtain an acceptable correlation between the input and output values, as well as the average deviation (or match) of the result.

## Acknowledgments

The authors are thankful for the support provided by the Bulgarian National Science Fund under Grant Ref. No. DFNI-I-02-5 “InterCriteria Analysis: A New Approach to Decision Making”.

## References

- [1] K. Atanassov, *Generalized Nets*. World Scientific, Singapore, 1991.
- [2] K. Atanassov, D. Mavrov and V. Atanassova. InterCriteria decision making. A new approach for multicriteria decision making, based on index matrices and intuitionistic fuzzy sets. *Issues in IFS and GN*, 11:1–7, 2014.
- [3] K. Atanassov, Intuitionistic fuzzy sets, *Proc. of VII ITKR's Session*, Sofia, June 1983 (in Bulgarian).
- [4] K. Atanassov, Intuitionistic fuzzy sets. *Fuzzy Sets and Systems*. 20(1) 87–96, 1986.
- [5] K. Atanassov, *Intuitionistic Fuzzy Sets: Theory and Applications*. Physica-Verlag, Heidelberg, 1999.
- [6] K. Atanassov, *On Intuitionistic Fuzzy Sets Theory*. Springer, Berlin, 2012.
- [7] V. Atanassova, D. Mavrov, L. Doukovska and K. Atanassov, Discussion on the threshold values in the InterCriteria Decision Making approach. *Notes on Intuitionistic Fuzzy Sets*, 20(2):94–99, 2014.
- [8] S. Bellis, K. M. Razeeb, C. Saha, K. Delaney, C. O'Mathuna, A. Pounds-Cornish, G. de Souza, M. Colley, H. Hagrass, G. Clarke, V. Callaghan, C. Argyropoulos, C. Karistianos, and G. Nikiforidis. FPGA Implementation of Spiking Neural Networks - An Initial Step towards Building Tangible Collaborative Autonomous Agents, *Proc. of FPT'04, Int. Conf. on Field-Programmable Technology*, The University of Queensland, Brisbane, Australia, 6-8 December, 2004, pages 449–452.
- [9] M. Hagan, H. Demuth, and M. Beale, *Neural Network Design*, Boston, MA: PWS Publishing, 1996.
- [10] S. Haykin, *Neural Networks: A Comprehensive Foundation* NY: Macmillan, 1994.
- [11] S. Himavathi, D. Anitha, and A. Muthuramalingam, Feedforward Neural Network Implementation in FPGA Using Layer Multiplexing for Effective Resource Utilization, *IEEE Transactions on Neural Networks*, 18(3):880–888, 2007.
- [12] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, Implementation of a real-time human movement classifier using a tri-axial accelerometer for ambulatory monitoring, *IEEE Trans. Inform. Technol. Biomed.*, 10(1): 156–167, 2006.
- [13] M. Meissner, M. Schmuker, and G. Schneider. Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training. *BMC Bioinformatics* 7(1):125, 2006.
- [14] D. Rumelhart, G. Hinton, and R. Williams. Training representation by back-propagation errors, *Nature*, 323:533–536, 1986.
- [15] L. A. Zadeh, Fuzzy Sets. *Information and Control*, 8:333–353, 1965.
- [16] Zwe-Lee Gaing, Wavelet-based neural network for power disturbance recognition and classification, *IEEE Transactions on Power Delivery*, 19(4):1560–1568, 2004.