Ana Maria Mihaela IORDACHE, PhD Candidate
Professor Liliana SPIRCU, PhD
Department of Economic Cybernetics
The Bucharest Academy of Economic Studies
E-mail: Iordache_ana_maria_mihaela@yahoo.com
E-mail: Liliana.Spircu@csie.ase.ro

USING NEURAL NETWORKS IN RATINGS

Abstract. *Analysis and determination of country risk was, is and will be a difficult problem to solve because of large amounts of data. The main results of model analysis and evaluation of country risk is the possibility of a large number of ranking countries according to their potential risk that characterize them. These rankings are an efficient tool for tracking the evolution of comparative risk for countries being the base for monitored and investment decisions. Using neural networks in classification of countries allows easily simulate situations for the issuance of some macroeconomic policies.*

Key words: *neural network, rating, classification, rprop, country*

JEL Classification: C15, C38, C45, C82, O11, O52.

Introduction

Risk is defined as a phenomenon that comes from circumstances for which the decision maker is able to identify the possible trends / events, and even their probability, without being able to specify exactly which of these events will actually happen.

In order of funding both investors (lenders) and recipients of funding decision are working with three possible situations: certain, uncertain and risky situations. Country rating given by different institutions is an important aggregate indicator when considering whether to invest or not in that area.

One of the most common risk rankings was conducted by Institutional Investor magazine, obtained as an average of estimates from 80-100 banking financial institutions. This ranking is now used as a reference point for international business environments.

Other classifications are made by rating agencies Standard & Poor's and Moody's. These two agencies rank countries in terms of ratings for various countries and loans for bonds issued by different states.

The problem of determining the country's rating is a difficult and multidimensional problem that cannot be approached from the classical point of view. Indicators that characterize a country were separated into three major groups:

economic, social and financial indicators. In terms of data analysis for each class of indicators we applied principal components analysis method, factor analysis and cluster analysis to identify the relevant indicators. Depending on the indicators that provide a maximum of information from each group we have classified countries using traditional methods of data analysis. Further, to obtain a better classification of countries we have assigned a neural network, using multilayer perceptron algorithm.

Neural network in classification

Model neural networks are networks of neurons connected through adjustable synapses. All neural network models are based on the interconnection of simple computing elements in a dense network of connections. Each processing unit (neuron) is able to run basic calculations, but the network as a whole may have outstanding problem solving skills (Figure 1).

Neural networks are built to solve problems that cannot be solved by using conventional algorithms. Such problems are optimization or classification. So neural networks are used in the areas such as classification patterns, regularity detection, image processing, voice analysis, optimization problems, control and guide robots, inaccurate or incomplete entry processing, quality assurance, forecast the stock market indices, simulations.

Although neural networks are able to find the solutions for difficult problems like the ones above, the results cannot be guaranteed to be perfect or at least some correct. The solutions offered are only approximations of the optimum solution, it is always present and must be analyzed and a specific error.

In the classification area, a neural network classification advantage is that it does not require the provision of templates or explicit rules of grouping. Furthermore, it learns to recognize the shapes of the examples used during training. A neural network can classify the incomplete data, often affected by noise and with a nonlinear distribution.

A neural network can adapt to training data. Once trained, it can generalize the knowledge. This will generate a reasonable output for an input that was not met during training (learning).

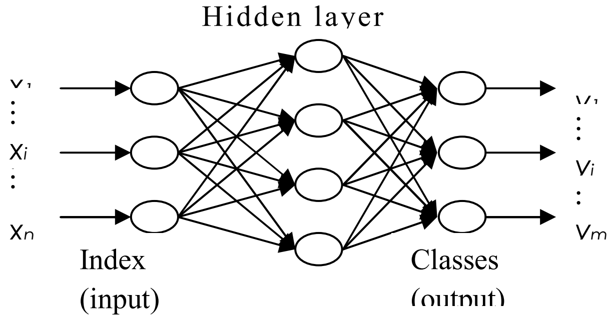

Figure 1. **General neural network**

Back propagation algorithm is usually considered as the most important and widely used algorithm for training neural networks.

Back propagation algorithm is a method for training multilayer neural networks with transmission before (unidirectional networks), which aims to minimize the mean square error by a gradient method. More specifically, the algorithm monitors the function to find the minimum error with respect to the weights of connections.

Back propagation algorithm operation is performing in two stages:

a. *m* is considered a model set of training data from which it extracts the required inputs - the vector $x^{(m)}$ - which applies to the entry system and, using current values of the weights, the information is spread before entry, real output is calculated from the network, $o^{(m)}$.

b. real output $o^{(m)}$ is compared with the desired value $d^{(m)}$ corresponding to the training set and the error thus calculated flows back to the network - from the output layer towards the input layer - to change the weights.

The training algorithm is used to determine precisely the approach of adjusting the network weights. In the case of Δ generalized form of the rule, the weights adjustment is made for the purposes of minimizing the deviation between the actual values of $o^{(m)}$ and the desired $d^{(m)}$ of the output network. If steps (a) and (b) above is repeated for the next model from the training set ($m \leftarrow m + 1$), their weights will apply a new correction in relation to output an $o^{(m+1)}$ and $d^{(m+1)}$. After exhausting all other models in the training set, it is said to have carried out a training cycle. It is expected that, as the consideration of new models from the training set, the deviations of $o^{(m)}$-$d^{(m)}$ to decrease. In most often, a single driving cycle is not approximated with sufficient accuracy the output of all values given in the training set. As a result, the algorithm repeats for a new cycle and the process continues to meet a certain criterion for stopping.

Form the generalized Δ rule proposed by Rumelhart is described by the following three propositions [3]:

a. For each model *m* of input - output from the training set, the correction of weights w_{ij} - noted $\Delta^{(m)}w_{ij}$ - the connection between neuron *j* and neuron *i* in the lower layer is proportional to an error term $\delta_j^{(m)}$associated with neuron *j*: $\Delta^{(m)}w_{ij} = \eta \cdot \delta_j^{(m)} \cdot o_i^{(m)}$, where $o_i^{(m)}$ is the output of the neuron *i* from the lower layer for model *m,* and η is factor called learning rate.

b. If the neuron *j* is in the output layer, the error term $\delta_j^{(m)}$ is calculated based on the deviation between the actual value $o_j^{(m)}$and the desired value $d_j^{(m)}$ and its derivative of the neuron activation function in relation to the entry *j* corresponding to the net of model *m*, noted $net_j^{(m)}$:

$$\delta_j^{(m)} = \left(d_j^{(m)} - o_j^{(m)}\right) \cdot f'\left(net_j^{(m)}\right).$$

c. If the neuron *j* from the hidden layer is being linked by synaptic connections with *k* neurons from the output layer, the error term $\delta_j^{(m)}$ is proportional to the sum of all error terms associated with *k* output neurons, being changed by the weights of those connections w_{jk} and derived activation function compared with net entry

$$\delta_j^{(m)} = \left(\sum_k \delta_k^{(m)} \cdot w_{jk} \right) \cdot f'\left(net_j^{(m)}\right).$$

Therefore, the weights of a particular neuron are adjusted in proportion of deviations between actual and desired sizes corresponding to the first of which neurons is connected.

The rules presented back propagation algorithm used for the application refers only to the weights, without mentioning anything by θ_j thresholds, associated with each neuron. This formality does not preclude using thresholds θ_j, which can be modeled as the weights w_{ij} of connections with a neuron i from the layer immediately below, whose output is always the unit value.

Define the classification problem

The main problem is to find some indicators that bring maximum of information and according of them to achieve a neural network witch associate a country to an already established class.

To build the application we used the following methodology:

1. We chose the countries on which analysis was carried out (European Union countries)
2. We chose the indicators used in the application and record their values,
3. Classification of countries according to each group of indicators (SAS Enterprise Guide software)
4. Extraction of indicators that bring maximum of information from each group
5. Setting number of groups of countries neural network,
6. Setting input data for future neural network (multilayer perceptron)
7. Implementation of neural network (SAS Enterprise Miner software)
8. Interpretation of results.

To assign a neural network to countries classification problem it is necessary to have a database entry. We took the indicator values from the websites of the European Institute of Statistics [14] and the International Monetary Fund [13]. To maintain comparability between countries we have taken the indicators as percentages of GDP for each country. If we chose all parameters to characterize a country then the neural network would be very complex and the time for training it would be very large. It is therefore necessary to identify, applying data mining techniques, several indicators that bring maximum of information. In this regard we have shared all the indicators in three categories: indicators of the real economy, social indicators and financial indicators. For each set of indicators we applied principal components analysis techniques and got the next results:

- The countries were grouped into three distinct classes, for a distance between clusters smaller than 1. Countries which are aggregated later, having the outlier role, we had associated them to the nearest cluster.
- The indicators from each set were grouped, at their turn also, into three classes. Depending on the scores from the correlation matrix, further it is kept the indicator that has the highest score. From the correlation matrix scores we kept further in the analysis that indicator that had the highest score. Thus, for each group of indicators would result three classes, from each class a single indicator would bring maximum of information, consequently it would be kept on in the analysis.

Association of a neural networks and its training using supervised learning techniques, requires knowing the output, or how countries are grouped into classes. Thus, we applied cluster analysis on all indicators database resulted from each set. We found that countries are divided into in three classes.

Therefore, input data in neural network consist of values for the following indicators: I2 (domestic demand), I4 (gross capital formation), I6 (exports), IS2 (share of employed population in total population), IS4 (share of 15-24 years old population in total population), IS10 (mean age of the population), IF2 (fiscal deficit), IF3 (public fixed investment), IF6 (foreign government loans), Class 1, Class 2, Class 3. As input data in each class we considered the value 1 if the country belongs to that class and 0 if it doesn't. (Table 1).

Table 1. Initial data from neural network problem

	I2	I4	I6	IF2	IF3	IF6	IS10	IS2	IS4	class 1	class 2	class 3
Austria	95.5	21	51	3.5	1.1	1.373	41.3	72	12	1	0	0
Belgium	97.3	20	73	6	1.9	-5.799	41.1	62	12	0	0	1
Bulgaria	108	26	48	4.7	4.9	0	40.9	63	13	1	0	0
Cyprus	105	17	40	6	4.1	-6.077	35.9	70	15	1	0	0
Czech Republic	94.4	22	69	5.8	5.2	-5.961	39.2	65	13	0	0	1
Denmark	96.2	17	47	2.7	2	-3.042	40.3	76	12	1	0	0
Estonia	92.5	19	65	1.7	5.1	0	39.3	64	15	0	0	1
Finland	98.3	18	37	2.5	2.8	-2.373	41.8	69	12	1	0	0
France	102	19	23	7.5	3.3	-7.874	39.5	64	13	0	1	0
Germany	95.1	17	41	3	1.6	-3.285	43.7	71	11	1	0	0
Greece	111	17	19	15	3.4	-12.87	41.4	61	11	0	1	0
Hungary	94.8	19	77	4.4	3.1	0	39.6	55	13	0	0	1
Ireland	84.2	14	91	14	4.7	-11.45	33.8	62	13	0	0	1
Italy	100	19	24	5.3	2.4	-5.313	42.8	58	10	0	1	0
Latvia	102	20	44	10	4.3	0	39.8	61	15	1	0	0
Lithuania	101	11	55	9.2	3.9	0	38.9	60	16	1	0	0
Malta	99.9	16	73	0.7	3.5	-1.11	39	55	12	0	0	1
Netherlands	92.8	16	69	3.8	2.2	-4.05	40.3	77	14	0	0	1
Poland	99.9	18	40	5.4	3.9	-4.914	37.5	59	12	1	0	0
Portugal	108	20	28	7.2	5.2	0	40.4	66	15	0	1	0
Romania	106	20	31	9.3	2.4	-9.334	38	59	11	0	1	0
Slovakia	100	25	71	8.6	5.3	0	36.5	60	14	0	0	1
Slovenia	98.7	20	58	7.9	2.3	-6.3	41.2	68	15	0	0	1
Spain	102	23	23	5.8	4.6	-6.127	39.5	60	12	0	1	0
Sweden	93.5	24	48	11	4.4	-11.45	40.7	72	11	1	0	0
Luxembourg	89.1	16	52	0.9	3.6	-2.213	38.7	65	13	1	0	0
United Kingdom	102	14	28	11	2.7	-10.89	36.9	70	13	0	1	0

Solving the countries classification problem

For each data set from the initial matrix SAS offers the possibility to calculate several descriptive indicators: minimum, maximum, average, standard deviation, skewness and kurtosis (Table 2).

Table 2. Descriptive indicators values for each data series

Name	Min	Max	Average	Standard deviation	Skewness	Kurtosis
I2	84.2	111	98.874	59.917	-0.21428	0.31054
I4	10.6	25.6	18.811	3.478	-0.01528	0.38222
I6	19	90.7	49.044	194.235	0.26955	-0.8051
IS2	54.9	77	64.515	59.221	0.41733	-0.5753
IS4	10.2	15.7	12.926	14.991	0.24417	-0.8466
IS10	33.8	43.7	39.556	21.525	-0.66067	0.92834
IF2	0.7	15.4	64.593	38.342	0.63119	0.0025
IF3	1.1	5.3	34.778	12.308	-0.13144	-10.972
IF6	-12.87	1.37	-4.409	41.888	-0.51241	-0.7851

Standard deviation and dispersion (skewness and kurtosis) are indicators which give a measure of dispersion values in a frequency distribution. Thus, skewness indicator shows obliquity and kurtosis indicates arching of the data series but both are indicators for showing the data sets distribution (Table 2). Asymmetry (skewness) may be positive (positive towards the long tail, right) or negative (long tail in the negative direction, left). Normal distribution has skewness 0, being symmetric. Kurtosis is based on increasing arms of a distribution. Distributions with relatively high queues are called leptokurtic, while those with small queues platykurtic. A normal distribution is mesokurtică (kurtotsis = 0). [15]

The SAS data were imported (in file work.mihaela1 in Figure 2), then they were divided into three data sets by using Partition Data module, in the network training data, model validation data and test data. Then it was constructed and trained neural network (Neural Network module) and finally we created a report with all the results (Reporter module). In addition to importing data we set in the model input and output variables and also the role played by each of the variable in the model. So for the future neural network input indicators are extracted from each database and output data are classes.

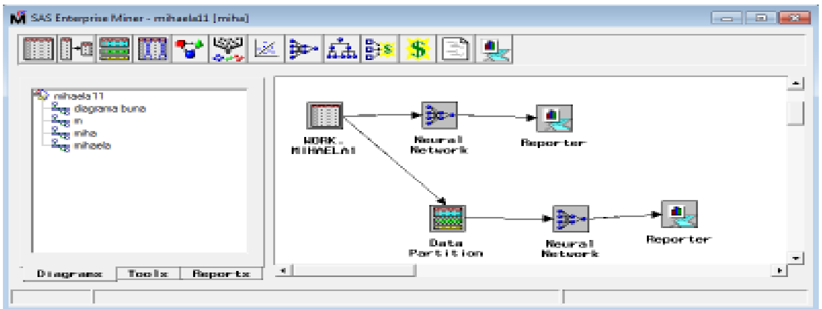

Figure 2. Working diagram from SAS Enterprise Miner

Through Partition Data module (Figure 2) SAS can share the data in: training data (80% in this case), test data and model validation data.
The analysis will effectively use the training data and neural network validation database. The test data can be applied by the user for verifying the neural network output.

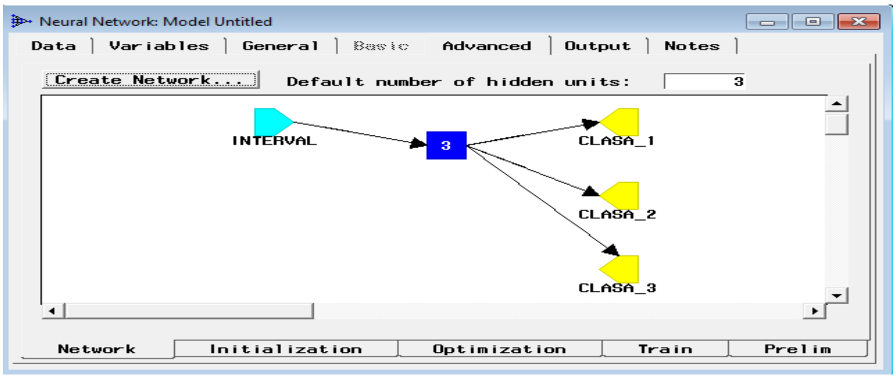

Figure 3. Neural Network Configuration in SAS Enterprise Miner

In the stage of neural network setup all indicators are grouped in the icon named Interval (all indicators are numeric input), three units are 'hidden' and it will result in three classes: clasa_1, clasa_2 and clasa_3 (Figure 3).

According to the objective function we chose Maximum likelihood (maximum likelihood a country will be in a class).

Before starting the effectively training of neural network the SAS calculation indicate neuronal specific indicators, namely:

a. the weights from input layer to intermediate layer neurons are:

-The weights from input layer to neuron 1 from interlayer are: -0.014, -0.033, -0.081, -0.074, 0.025, 0.166, -0.507, 0.264, 0.191;

-The weights from input layer to neuron 2 from intermediate layer: -0.56, -0.481, 0.15, -0.215, 0.57, 0.01, 0.321, 0.202, 0.1;

-The weights from input layer to neuron 3 from intermediate layer: 0.284, 0.243, 0.06, 0.21, -0.344, 0.164, 0.491, 0.614, -0.377;

b. the interlayer weights of the final layer neurons (consisting of classes). All these weights are initially equaled by 0.

c. the activation thresholds of neurons from the intermediate and final layers:

-The thresholds of interlayer are: 1556397, 1377611 and 1112926;

-The final layer thresholds are: -0.368, 1223775; 0.059.

d. the value of the objective function is 0.311.

To choose optimization technique, SAS has a number of techniques [8]:

- Standard backprop is the most used technique for training the neural networks. But this method is slowly, unreliable and requires adjustment of learning rate, a process that can be extremely bored;
- RPROP: The process of iteration is very fast, but it requires more than reversals of conjugate gradient technique. RPROP uses several learning rates for each weight and it is more stable than any other prop technical type, since the algorithm does not depend on the exact rate of learning at initial phase;
- Quickprop applies search technique such as linear and downward slope technique. This method is faster and more back-prop reliable than standard algorithms except for certain improvements.
- Incremental Backprop involves updating the weights after each record adjustments applied to learning error rate. But this technique is not a convergent one, and if the estimators are closed to a local minimum then you should select another method of learning to reach convergence.

Further into the application we used RPROP algorithm because it has the best features of training for constructed neural network. This algorithm is an adaptive one, the learning rate changes depend on the change of sign of partial derivatives in two successive iterations.

In application of RPROP algorithm, SAS made several hypotheses description of trustworthiness optimum, including [8]:

- Absolute convergence (ABSCONV) is based on absolute convergence criterion of the objective function. It is calculated as $|f(x^k)|\leq c$;
- Absolute function convergence (ABSFCONV). Instead look for the absolute value of the objective function, it is search for module objective functions between two successive iterations, and convergence occurs when this index value falls below the specified convergence criterion. The formula is: $|\ f(x^{k-1}) - f(x^k)|\leq c$;
- Absolute convergence gradient (ABSGCONV) implies choosing the maximum gradient of the objective function values and the convergence is considered achieved when the maximum is less than the convergence criteria. It is calculated with the following formula: $\max_j|g_j^{(k)}|\leq c$ for any function gradient g;
- Absolute convergence parameter (ABSXCONV) shows the difference between the estimates of parameters from two successive iterations, with choosing the maximum value. Convergence is achieved when the difference is less than the convergence criterion. It is calculated as: $|w^{k-1} - w^k|\leq c$,
- Relative Function Convergence (FCONV) is calculated as:

$$\frac{|f(x^{k-1}) - f(x^k)|}{\max(|f(x^{k-1})|, c)} \leq c$$

- Relative gradient convergence (GCONV) is calculated as:

$$\frac{(g^k)'[H^k]^{-1}(g^k)}{\max(|f(x^{k-1})|,c)} \le c$$

There are considered some conditions for RPROP algorithm before implementing the optimization function: ABSCONV= -2688534671; ABSFCONV=0; ABSGCONV=0.00001; ABSXCONV=0; FCONV=0.0001; GCONV=1.00E-08, acceleration=1.2; deceleration=0.5; maxlearn=150 and minlearn=0.00001.

After completion of training the network it is observed that there were 95 iterations (Figure 4) up to achieve the desired result (convergence criteria showed by the variable ABSGCONV). The function has been activated by 99 times, and the function gradient by 97 times. Last value of the objective function was 1.127246E-6, and the gradient function was 1.4795579E-6.

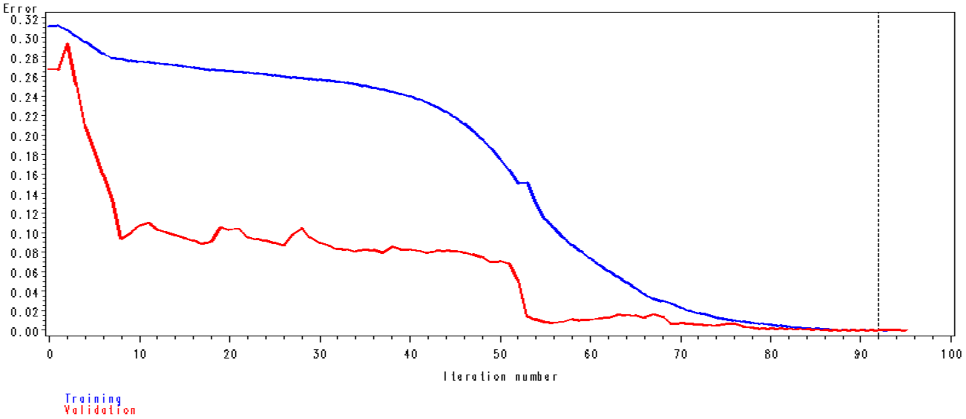

Figure 4. **Chart training of sets of errors (blue line) and validation of sets (red line) of the neural network**

According to the development of training and validation errors of the objective function in the figure 4 the two functions have evolutions from 0.32 and 0.27 and tend gradually by each iteration to zero. If at the first iteration the functions have oscillating trend, to the end it begin to stabilize showing that this function is set near the threshold of convergence and training of network is made successfully.

After the completion of the neural network training process the specific indicators are:

a. weights from input layer to the intermediate layer neurons:

-weights from input layer to the neuron 1 from the interlayer are: 0.395, -0.53, -1.88, -0.16, 0.17, 0.85, -0.75, -0.52, 0.11;

- weights from input layer to the neuron 2 from the interlayer are: -1.16, -0.12, 1.89, -1.18, 0.92, 0.29, -0.33, -0, 07, 0.34;

-weights from input layer to the neuron 3 from the interlayer are: 0.36, 0.35, -0.48, 0.52, 0.29, 1.12, 0.68, 1.89, -0.51;

b. the interlayer weights of the final layer neurons (consisting of classes) are: 17.78, 15.22, 18.73 (class 1), 2.71, -24.37, -4.6 (class 2), -15.1, 3.53, -12.34 (class 3).

c. activation thresholds of the neurons from intermediate and final layers:

-for interlayer the thresholds are: -1.11, 1.82, -1.4;

-the values of the final layer thresholds are: 2.9, -5.9, -16.03.

d. objective function value is 1.127246E-6.

After the optimization process closed the new value of the objective function is much smaller than the first (0.311) this shows that error of assignment a country to a class has decreased markedly, which it was desirable.

For the future use of neural network already trained SAS offers the user the transfer functions through the network, as follows:

a. The scores of hidden layer neurons, and transfer functions are:

- H11 = 0.396 * S_I2 + -0.522 * S_I4 + -1.752 * S_I6 + -0.163 * S_IF2 + 0.17 * S_IF3 + 0.874 * S_IF6 + -0.755 * S_IS10 + -0.521 * S_IS2 + 0.12 * S_IS4
- H12 = -1.17 * S_I2 + -0.0938 * S_I4 + 1.516 * S_I6 + -1.0861 * S_IF2 + 0.826 * S_IF3 + 0.274 * S_IF6 + -0.271 * S_IS10 + -0.038 * S_IS2 + 0.252 * S_IS4
- H13 = 0.359 * S_I2 + 0.352 * S_I4 + -0.393 * S_I6 + 0.49 * S_IF2 + 0.298 * S_IF3 + 1.101 * S_IF6 + 0.583 * S_IS10 + 1.807 * S_IS2 + -0.506 * S_IS4
- H11 = -1.11 + H11
- H12 = 1.82 + H12
- H13 = -1.4 + H13

b. To write the final nodes in the network (Class 1, Class 2 and Class 3) SAS Enterprise Miner uses the following functions, with corresponding scores:

- class 1: P_clasa_11 = H11 + 9.431 * 11.995 * 12.943 * H12 + H13
P_clasa_11 = 2.9 + P_clasa_11
- class 2: P_clasa_21 = -26.778 + 2.676 * H11 * H13 * H12 + -4571
P_clasa_21 = -5.9 + P_clasa_21
- class 3: P_clasa_31 = -12.556 + 3.549 * H11 * H13 * H12 + -12.278
P_clasa_31 = -14.8 + P_clasa_31.

After completing of training the neural network and checking of results some countries have changed their initial class. So, the new grouping of countries based on scores obtained from SAS Enterprise Miner is:

- Class 1: Austria, Denmark, Sweden, Finland, Germany, Cyprus, Latvia, Poland, Lithuania, Netherlands Spain, Portugal;
- Class 2: France, Italy, Romania, Greece, Britain, Bulgaria;
- Class 3: Belgium, Hungary, Malta, Slovakia, Czech Republic, Estonia, Slovenia, Luxembourg, Ireland.

After learning the neural network it is shown that there are countries which have changed the class. This is because the data set was divided before the network learning in three parts: training data (80% of all countries), data validation and test data. So, for training network we used data for all countries less than Netherlands Luxembourg, Spain, Portugal and Bulgaria. For each country we tested the network and on the basis of the maximum score we assigned it into an existing class.

Conclusion

Based on the already trained neural network and its results, for another country it can be forecast on the economic, social and financial. In this way, for Romania we elaborated several solutions / scenarios for improving the macroeconomic policies and to make possible its accession in the class of developed countries.

The calculation shows that Romania was classified in the Class 2, together with Greece, Bulgaria, Great Britain, etc. Class 1 is composed of developed countries, characterized by economic stability, social and financial. To move in the class of developed countries it should take measures for changing the economic, social or financial indicators. So, according to the analysis, among the solutions (scenarios) for Romania are presented in table 3.

Table 3. Scenarios for improving the general situation from Romania and its transition from Class 2 to Class 1

Scenario	Initial values	Simulated values
1.	IF2=9.3; IF3=2.4; IS2=58.6	IF2=0; IF3=3; IS2=60
2.	IF6=-9.33; I2=106.1	IF6=0; IS2=95
3.	I2=106.1; IS10=38	I2=95; IS10=30
4.	IS4=11.4; IF2=9.3; IF3=2.4	IS4=15; IF2=8; IF3=5

To achieve scenario 1 Romania should have a rather important role in the economy, so it should found ways for decreasing the fiscal deficit (IF2) from 9.3 to 0, increasing public investment in fixed assets (IF3) by 0.6 percentage points and the real economy should absorb more labor (IS2), 1.4 percentage points more.

For scenario 2 consumers should produce some of the goods so that the domestic demand for goods and services in the real economy falls from 106.1% to 95%, and the state should not make external loans.

Scenario 3 shows that if the average age of the population could decrease from 38 to 30 years together with the domestic demand (whether births were encouraged and domestic production of goods and services supported) then Romania would have economic, financial and social stabilities.

Scenario 4 shows that if IS4 indicators (share of population aged 0-14 years in the total population) could increase from 11.4 to 15, IF2 (the fiscal deficit) would decrease from 9.3 to 8 and IF3 (investment of public fixed assets) would increase from 2.4 to 5 then Romania could register economic growth and would qualify in the class 1 along with other developed countries.

According to the proposed model there are, theoretically, many scenarios (solutions) to change our country's position in the rankings (the transition from Class 2 to Class 1). But from economic point of view it should be choose those solutions that can be implemented without causing major macroeconomic disturbances and help the state to pass easily over the possible shocks arising from various worldwide causes (global economic crisis, wars, natural disasters, calamities, etc.).

Acknowledgement : ***This article is a result of the project POSDRU/6/1.5/S/11 „Doctoral Program and PhD Students in the education research and innovation triangle". This project is co funded by European Social Fund through The Sectorial Operational Programme for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies***."

REFERENCES

[1] **Bran Paul (1995),** ***Relatii financiare si monetare internationale: economia fenomenului mondo-financiar si monetar*** ; Economica Publishing, ISBN: 9739198201;
[2] **Diana A. W. (2008),** ***Neural Network Models Using SAS Enterprise Miner***. 22nd Annual SAS Malaysia Forum, Kuala Lumpur Convention Center;
[3] **Caleanu Catalin- Daniel, Tiponut Virgil (2002),** ***Retele Neuronale. Arhitecturi si algoritmi***, Politehnica Publishing, Timisoara, ISBN 973-9389-66-X;
[4] **Howard Plemmons (2011),** ***Working with a DBMS using SAS Enterprise Guide*****;** SAS Global Forum, Las Vegas, Nevada;
[4] **Lazarescu Sorin (2003),** ***Rating financiar*****,** ASE Publishing, ISBN: 973-594-296-8;
[5] **Meyers L., Gamst G., Guarino A.J. (2009),** ***Data Analysis Using SAS Enterprise Guide***; Cambridge University Press, ISBN: 9780511601842;
[6] **Miron Dumitru, Dima Alina Mihaela, Paun Cristian (2009),** ***A Model for Assessing Romania`s Real Convergence Based upon Distances and Clusters Methods ;*** Economic Computation and Economic Cybernetics Studies and Research, pg. 5-23, ISSN: 1842-3264;
[7] **Randall Matignon(2007),** ***Data Mining Using SAS Enterprise Miner*** ; Wiley – Interscience Publishing, ISBN: 9780470149010;
[8] **Randall Matignon (2005),** ***Neural Network Modeling Using SAS Enterprise Miner ;*** Author House Publishing, ISBN: 9781418423414;
[9] **Ruxanda Gheorghe (2010),** ***Learning Perceptron Neural Network with Backpropagation Algorithm*** **.**Economic Computation and Economic Cybernetics Studies and Research, pg. 37-55, ISSN: 1842-3264;
[10] SAS Institute Inc. 2003. Data Mining Using SAS Enterprise MinerTM: A Case Study Approach, Second Edition, USA, 2003, ISBN 1-59047-395-7;
[11] **Spircu Liliana (2005),** ***Analiza datelor. Aplicatii economice.*** ASE Publishing, ISBN: 9735947013;
[12] http://epp.eurostat.ec.europa.eu/portal/page/portal/microdata/introduction
[13] www.fmi.com
[14] http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home/
[15] http://www.mymed.ro/statistica-descriptiv.html
[16] http://www.sas.com/technologies/analytics/datamining/miner/