Classification of companies with theassistance of self-learning neural networks

Klasifikace podniků za pomoci samoučících se neuronových sítí

Vladimír KONEČNÝ[1], Oldřich TRENZ[1], Eliška SVOBODOVÁ[2]

[1]Department of Computer Science, Faculty of Business and Economy, Mendel University in Brno, Brno, Czech Republic
[2]Department of Regional and Business Economics, Faculty of Regional Development and International Studies, Mendel University in Brno, Brno, Czech Republic

Abstract: The article is focused on rating classification of financial situation of enterprises using self-learning artificial neural networks. This is such a situation where the sets of objects of the particular classes are not well-known. Otherwise, it would be possible to use a multi-layer neural network with learning according to models. The advantage of a self-learning network is particularly the fact that its classification is not burdened by a subjective view. With reference to complexity, this sorting into groups may be very difficult even for experienced experts. The article also comprises the examples which confirm the described method functionality and the neural network model used. A major attention is focused on the classification of agricultural companies. For this purpose, financial indicators of eighty one agricultural companies were used.

Key words: artificial intelligence, neural network, Kohonen network, learning, classification

Abstrakt: Příspěvek se zabývá klasifikací ohodnocení finanční situace podniků využitím samoučící se umělé neuronové sítě. Jedná se o situaci, kdy nejsou známé množiny objektů jednotlivých klasifikačních tříd. V opačném případě by bylo možné využít vícevrstvou neuronovou síť s učením podle vzorů. Výhodou samoučení sítě je zejména fakt, že její klasifikace není zatížena subjektivním názorem. S ohledem na složitost může být třídění do skupin velmi obtížné i pro zkušené experty. V příspěvku jsou uvedeny příklady potvrzující funkčnost metody i použitého modelu neuronové sítě. Hlavní pozornost je orientována na klasifikaci zemědělských podniků. K tomuto účelu bylo využito 15 finančních ukazatelů u 81 zemědělských podniků.

Klíčová slova: umělá inteligence, neuronová síť, Kohonenova síť, učení, klasifikace

Rating companies on the basis of the attained values of economic indicators is an interesting partial problem not only from the point of view of the decision-making processes of managers and the corporate strategy as well (Svoboda 2007), but also from the position of decision-making in the customer-supplier relationship management. An example can be the evaluation of credibility of buyers who are provided with products and services. Similarly, this approach can be used for modelling customer behaviour, with regard to the provision of tailor-made services (Trenz 2006). Of course, each sub-task requires a specific selection of attributes of the examined objects depending on the problem.

For many classification tasks, it is possible to select objects, the so-called class representatives, which characterize the sub-sets. However, it is sometimes not possible to describe the pertinence by an arithmetic formula. For such tasks, multilayer neural networks able to learn the classification in accordance with models, respectively with a teacher, can be utilized very effectively (Konečný et al. 2005).

For solutions of the classification tasks in the circumstances where the representatives of each class

Supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. MSM 6215648904).

and eventually the exact number of classes are not known, the artificial intelligence methods can be used – specifically the self-learning neural networks, eventually including the relevant graphic display – the Kohonen maps (Deboeck, Kohonen 1998). Self-learning neural network (learning mode without a teacher) is principally a simple tool which is described in the literature (Kohonen 2001) with wide application possibilities.

The initial situation of a classification problem is given by the existence of a set of objects (defined as n-dimensional vectors), that is to be divided into k subsets. The number of subsets can be specified for some tasks (e.g. the classification of products into *good* and *bad*) but for some, it is an objective part of the classification task solution. The specification of the object attributes (vector components) done by an expert in the field is desirable. The aim of this work is to devise a model of a Kohonen neural network for classifying sets of objects along with displaying the objects into a plane and to verify the classification procedure of a set of the given companies described by the selected financial indicators.

MATERIAL AND METHODS

The base of the model according to the objective defined is a neural network (Figure 1a), in which each point $X = (x_1, x_2, x_3, ..., x_n)$ in n-dimensional vector space corresponds to an output neuron Y^{ij} with weight-vector $(w_1^{ij}, w_2^{ij}, w_3^{ij}, \ldots, w_n^{ij})$. In their arrangement in a two-dimensional grid, the neural network will display the set of points X from n-dimensional space in two-dimensional space represented by neurons Y^{ij}. It is a form of certain simplification allowing the qualitative assessment of the selection.

The number of the neural network inputs is given by the number of the coordinates of vectors X defining the individual objects. The number of the neural network outputs is given by the number of the classified objects of the learning file, because it presupposes the existence of one output for each input sub-vector/object X.

This condition may not be taken dogmatically; the output map can be greater than the input file robustness. The mentioned extension is also applied to the mentioned practical example. This modification will make it easier and clearer to deploy inputs into the output map. Each sub-vector, in this case, describes the specific financial situation in a company at a given time. Thus the vector X represents one point in the n-dimensional vector space.

x_k inputs are connected to every output neurons by weights $w_k^{i,j}$ (Figure 1b) representing the vectors of output neurons and amplifying the self-learning process. The initial values of vectors $W^{i,j}$ are the values randomly generated in the interval <−1.1>. This configuration allows achieving the solutions stability with regard to the initial conditions.

The core of the self-learning algorithm is the principle of proximity, i.e.

- for any input vector X, there is chosen such output Y, the weighting vector of which is the closest to the vector X;
- near vectors X_i and X_j must have near corresponding output images and near vectors W as well

The learning algorithm is described in many publications in detail (Kohonen 2001) and therefore here are given just the formulas for the calculation of the learning coefficient of winning neurons – $\alpha(E)$, neighbouring neurons (in consideration of the winner) – $\beta(E, d)$ and the formula for the calculation of the actual size of the winner's surround.

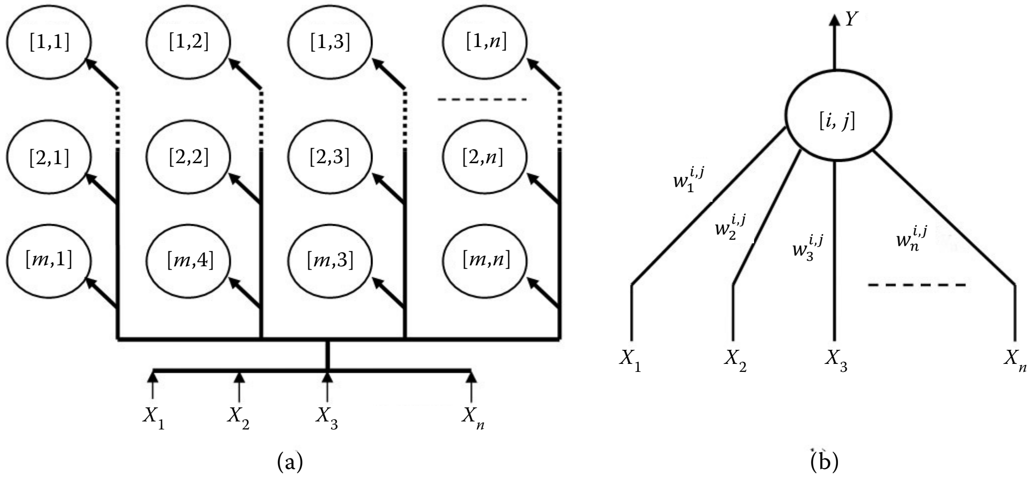

Figure 1. Model of the self-learning neural network

To calculate some constant conclusion, both learning coefficients and the size radius of the surround (depending on the number of proceeded periods) must with $E \to E_{max}$ converge to zero.

In the applied model of the self-learning neural network, the learning coefficient calculation of the winning neurons is implemented as follows:

$$\alpha(E) = \frac{\alpha_0}{1 + \left(\frac{E}{\sqrt{3}K_\alpha E_{max}}\right)^2} \qquad (1)$$

where α_0 is the initial value of learning coefficient, E_{max} is the specified maximum number of periods of learning and K_α is the relative position of the inflection point, $\alpha(E)$ is expressed in relation to E_{max}.

The learning coefficient of surrounding (neighbouring) neurons is calculated as follows:

$$\beta(E,d) = \alpha(E)\exp\left(-\left(\frac{d - d_0}{\sqrt{2}(R_0 K_\beta - d_0)}\right)^2\right) \qquad (2)$$

where R_0 is the initial radius of the surround of the winning neurons, d_0 is the radius of the surround with a coefficient of learning, $\beta(E, d) = \alpha(E)$ and K_β is relative distance of function's inflection point $\beta(E, d)$ with respect to the current radius of the surround $R(E)$.

$$R(E) = R_0 \exp\left(-\frac{E \ln(R_0)}{K_R E_{max}}\right) \qquad (3)$$

R_0 – initial and E – the current period of the neural network learning, K_R – a relative (in relation to E_{max}) value for the number of periods, after which the $R(E) < 1$.

The carried out experiments indicate that a well-functioning model can be obtained when setting coefficients mentioned as follows: $K_\alpha \approx 0.3$; $K_\beta \approx 40$; $K_R = 0.5$.

To avoid an undesirable looping of algorithm the selection of input vectors, X for each period of learning is carried out randomly.

Correction of the output neurons $W^{i,j}$ is performed in each period by each vector X. If the condition

$$| X - W^{i,j} | \leq | X - W^{k,l} | \qquad \text{for } \forall k,l \qquad (4)$$

is positive, the vector correction of the winning neuron $W^{i,j}$ is implemented as follows:

$$W^{i,j}_{nový} = W^{i,j} + \alpha(X - W^{i,j}) \qquad (5)$$

And with the neighbouring neurons as follows:

$$W^{i,j}_{nový} = W^{i,j} + \alpha(X - W^{i,j})$$

The neighbouring neuron's distance d with the vector $W^{r,s}$ from the winning neuron with the vector $W^{i,j}$ needed for the calculation of β is calculated in the learning algorithm as follows:

$$d = \sqrt{(i - r)^2 + (j - s)^2} \qquad (6)$$

Making corrections gets the coordinates of the winning neurons and neighbouring neurons closer to the coordinates of the input vector X. If the number of the output neurons is equal to the number of input vectors, then with an increasing number of steps of the algorithm's learning and with a small learning coefficient, the coordinate vectors values of the winning neurons will converge to respective values of the input vectors.

The application used for experiments was created by the authors of the article.

RESULTS AND DISCUSSION

Classification of products

A simpler example with the possibility of graphic display of the set of input vectors is chosen for the presentation of the process. The aim is to discover the distribution of 22 products defined by two quantifiable attributes into three subsets. Let there be *the first, second* and *third* class of goods. The list is given in Table 1.

1. It would be sufficient to define a neural network with two inputs and three outputs for a simple classification. Setting the configuration, entering the input vectors and using the self-learning neural network, three vectors of representatives of three requested subsets of products can be obtained. The representatives are characterized by the minimal sum of distances to all sub-elements. In other words, the distance of any element of a subset to its representative is less than the distance to the representatives of other sets.

Table 1. The products

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
X	–1	–6	–5	–3	–4	–6	–2	–1	–2	0	0	1	1	2	2	3	4	–7	–6	–4	–3	–3
Y	4	3	2	2	1	0	0	–1	–2	–3	2	3	1	0	–2	2	0	–4	–2	–3	–4	–1

According to the representatives´ attributes, an expert for the quality control must determine what classes they represent. Such a model would be sufficient for any other products, but when it is necessary to obtain further information on the classification accuracy and distribution of products with respect to the representatives, it is appropriate to establish the so-called Kohonen map.

2. Displaying will be done in a two-dimensional grid with the number of neurons equal to or greater than the number of products.

It is evident that products with the same or close characteristics must belong to the same class, and this is respected by the self-learning neural network algorithm. With regard to the defined set of objects, the configuration of the neural network can be determined.

The number of entries is given uniquely by the number of the coordinates of vectors. The number of outputs (or vectors of output neurons) is selected depending on the required form of the output map, thus on the fact how many outputs the entry of input objects will display. Usually, it is required to display the grid in the shape of a square or a related rectangle. The number of output neurons is chosen close to or equal to the number of the input objects. To display the objects of the task, there is chosen the format 6 × 4. In case the number of outputs N_{out} is lower than the number of inputs $N_{in,}$ some outputs will be common for two or more input objects.

The displaying of objects itself is not a solution of the given problem because its solution requires the allocation of objects into three groups. For this purpose, we can use the self-learning neural network with three outputs. A self-learning algorithm causes that each output will be the winner just for that input vector, for which it is relevant (4). Because each input vector must have an output, it is clear that every vector of the output neurons achieved $N_{out} < N_{in}$ can be corrected by several closest input vectors.

The vectors of outputs in such cases will be considered as the representatives of sets

$$M_j = \{X_i; \forall k |X_i - R_j| < |X_i - R_k|\} \qquad (7)$$

It can be proven easily that the vector R_j is equal to the gravity centre of the represented set of points with the unit weight, i.e. the vector

$$R = \frac{\sum_k X^k}{N} \qquad (8)$$

where N is the number of vectors X. This formula (8) can be used for the control (or refinement) of the representatives' coordinates of various sets established by the neural network or during the correction of classification sets.

As the learning process result with a set of input data according to Table 1 and three outputs, the following vectors of representatives were obtained:

$$\begin{aligned}
&R1 = [1.3722, 1.2760] \\
&\quad \text{with a set of elements} \\
&M1 = [1, 11, 12, 13, 14, 15, 16, 17] \\
&R2 = [-2.0455, -2.0699] \\
&\quad \text{with a set of elements} \\
&M2 = [7, 8, 9, 10, 20, 21, 22] \\
&R3 = [-5.2384, 0.4777] \\
&\quad \text{with a set of elements} \\
&M3 = [2, 3, 4, 5, 6, 18, 19]
\end{aligned} \qquad (9)$$

At this stage, the classification of products derived from this training task can be considered as solved, because only a human – an expert in the field – can give semantic importance to the particular sets.

Figure 2 shows the possible solution displayed as a Kohonen map. The particular vectors are shown here (for highlighting, with the prefix "*x*") as well as the representatives of each group (prefix "*r*").

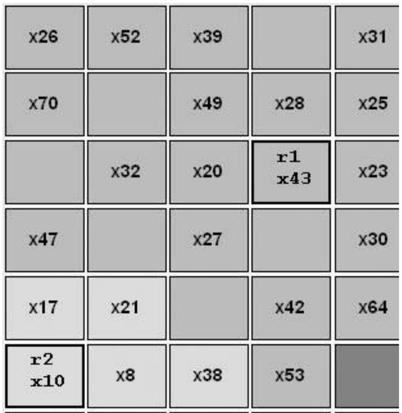

Figure 2. Resulting classification of products in three sets

Classification of agricultural companies

The approach described above (a model) will be used in evaluating the financial situation of agricultural companies. Let the input value be a simplification of 15 indicators which are used even by an expert for his/her assessment. These indicators are usually divided into groups relating to various areas of management, and to the indicators of profitability, activity, debt and liquidity. This expert simplification will allow reducing the complexity of the problem in the evaluation with the aid of fractional sub-groups and the subsequent determination of the resulting

Table 2. Input set of real data of agricultural companies

Input	Inventory turn over (days)	Average collection period (days)	Creditors payment period (days)	Current ratio (–)	Quick ratio (–)	Cash ratio (–)	ROA (%)	ROE (%)	Added value per person employee (th. CZK/m.)	Average month pay (th. CZK/m.)	Current debt ratio (%)	Long-term debt ratio (%)	Debt equity ratio (–)	Over-capitalized (–)	Total assets turn-over (years)
1	86.000	182.000	82.000	4.910	3.980	2.100	16.240	13.780	8.000	12.000	8.286	6.843	0.017	1.561	0.325
2	140.000	92.000	17.000	6.070	3.760	2.270	18.450	5.980	17.000	20.000	10.754	56.847	0.243	2.584	0.639
3	111.000	49.000	82.000	1.250	0.720	0.050	54.970	8.420	11.000	16.000	36.797	47.893	0.428	1.168	0.632
20	133.000	60.000	69.000	3.870	2.120	1.160	4.790	2.730	20.000	20.000	10.119	32.935	0.056	1.468	0.478
21	68.000	84.000	27.000	3.840	2.260	0.330	3.920	2.180	24.000	19.000	12.151	32.090	0.038	1.647	1.013
22	174.000	77.000	78.000	3.440	1.490	0.520	0.540	0.410	13.000	16.000	12.733	12.542	0.029	1.553	0.515
30	150.000	65.000	33.000	3.870	1.290	0.170	1.560	1.180	20.000	21.000	8.511	15.830	0.012	1.364	0.527
31	228.000	110.000	69.000	5.870	2.570	0.890	2.260	1.810	20.000	21.000	5.374	14.314	0.011	1.382	0.280
32	144.000	108.000	30.000	4.990	2.480	0.540	5.430	3.240	24.000	20.000	6.159	34.179	0.044	1.354	0.386
41	191.000	46.000	73.000	3.530	1.190	0.600	5.490	3.320	13.000	18.000	11.797	27.612	0.045	1.512	0.521
42	110.000	29.000	28.000	3.330	1.150	0.550	4.950	3.290	12.000	13.000	8.270	25.270	0.039	1.270	0.590
50	153.000	130.000	0.000	1.230	0.730	0.050	1.770	0.230	7.000	0.000	51.049	35.822	0.904	1.401	0.601
51	135.000	71.000	0.000	4.670	2.110	0.360	7.060	3.750	39.000	0.000	6.693	40.185	0.051	1.359	0.457
52	217.000	122.000	11.000	6.850	2.740	0.410	6.480	5.030	42.000	24.000	7.104	15.376	0.017	1.817	0.484
60	136.000	55.000	26.000	5.690	3.060	2.080	0.200	0.110	17.000	18.000	9.292	38.913	0.152	1.940	0.648
61	155.000	75.000	257.000	1.430	0.580	0.160	0.950	0.500	25.000	27.000	27.974	19.049	0.052	1.199	0.552
62	139.000	143.000	56.000	3.920	1.700	0.360	2.940	2.600	13.000	12.000	8.787	2.791	0.020	1.606	0.503
70	115.000	135.000	98.000	3.870	2.210	0.740	−18.850	−3.150	23.000	19.000	13.166	70.138	0.247	2.059	0.684
71	148.000	142.000	97.000	2.750	1.420	0.130	−6.730	−2.580	7.000	12.000	24.465	37.288	1.005	2.306	0.791
72	175.000	102.000	19.000	3.780	1.590	0.320	−3.830	−3.300	11.000	19.000	12.487	1.264	0.020	1.657	0.560
79	178.000	30.000	127.000	1.240	0.220	0.050	−27.310	−4.980	8.000	10.000	36.847	44.925	0.559	1.159	0.756
80	164.000	32.000	136.000	1.080	0.220	0.050	−32.200	−6.980	9.000	11.000	40.716	37.619	0.421	1.059	0.769
81	180.000	45.000	81.000	2.690	0.630	0.120	−17.090	−6.810	10.000	11.000	17.883	42.295	0.148	1.584	0.735

financial situation of a particular company. Using the self-learning network, this approach may not be considered and it is possible to adapt all the considered inputs in the process of learning. For brevity reasons, only some companies of 81 total are shown in Table 2.

Standardization of data

In order to facilitate processing, the absolute value is not interesting, but rather the temper of these items of data. The values of the particular coordinates of the input vector can be in different units, but this may cause that some of them can act in a dominant way at the expense of others. Sometimes it is appropriate to adjust the input data so that the individual components are equivalent. One of the approaches how to achieve this is to standardize the data.

1. Let us have a matrix $Z = (z_{ij})$ $n \times p$, the rows of which there are p-dimensional vectors of numbers characterizing n objects. Data standardization is accomplished in two steps:
   Calculate a mean value $\bar{z}_j$ character z_j with index j and standard deviation s_j for the $j = 1, 2, ..., p$ according to formulas

$$\bar{z}_j = \frac{1}{n}\sum_{i=1}^{n} z_{ij} \qquad s_j = \left[\frac{1}{n}\sum_{i=1}^{n}\left(z_{ij}-\bar{z}_j\right)^2\right]^{1/2} \tag{10}$$

2. The initial values of z_{ij} character with index j of object with index i are transformed to so-called standardized values

Now these standardized values of characters have the mean value equal to 0 and the variance equal to 1.

Let these classified companies represent three qualitative groups. It is necessary to classify each input vector into one group, respectively one qualitative class, although there are no classification rules.

Similarly, as in case with the model example of the products classification, the first step in solving will be the usage of self-learning of the neural network to find the representatives of the sets. The requirement of three sets implies the use of a neural network with three outputs. In the self-learning process, inputs (companies) are assigned to outputs (representatives) according to the rule of proximity of vectors and the proper subsets listed in Table 3 are created.

In the second step of the task solving, the entries (companies) are displayed into the map. As a consequence of the self-learning with the proximity rule, all inputs of each set will be deployed in the vicinity of its representative. The required classification of companies into three subsets is shown in Figure 3a. The subsets representatives have their frame borders highlighted and all sub-sets are varied with grayscale levels. In the output map, those inputs sharing a common output with one of the displayed inputs are not visible (Figure 3b). According to an estimate of an expert economist, M1 companies belong to the group of companies with good health, M2 represents average companies and M3 are the worst.

Groups can be semantically assessed only by an expert. The division criterion into groups may not be always clear and it also may not correspond to the classical approach; however, in particular, the correct classification ability of the network used is primarily important.

In this case, it means:

Classifying financial situation of agricultural companies, four basic groups of ratio indicators were used: the indicators of profitability, activity, debt and liquidity. As for the final evaluation, the indicators were evaluated separately and also in context, taking into account the nature of the commercial segment. Aggregate financial distress prediction models were used to illustrate the results. On the basis of the calculated characteristics, the financial situation for each company was evaluated.

Quite naturally there arises a question why the classification was performed in three subsets and not in two or more than three subsets. This and the consequential problem solving the semantic subsets site is very closely linked to the problem, which is often called *knowledge mining*. This means that the requirement of the classification depends on what an expert in the field solves or expects. The usual procedure includes first a binary, eventually a ternary classification and according to the results of the classification of higher degrees, eventually the decision-tree classification.

Table 3. Classification of enterprises into three classes

M1	9, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 37, 39, 40, 41, 42, 43, 44, 47, 48, 49, 51, 52, 53, 54, 59, 60, 62, 63, 64, 65, 70, 72, 75
M2	3, 8, 10, 11, 12, 14, 15, 16, 17, 21, 38, 46
M3	45, 50, 55, 56, 57, 58, 61, 66, 67, 68, 69, 71, 73, 74, 76, 77, 78, 79, 80, 81

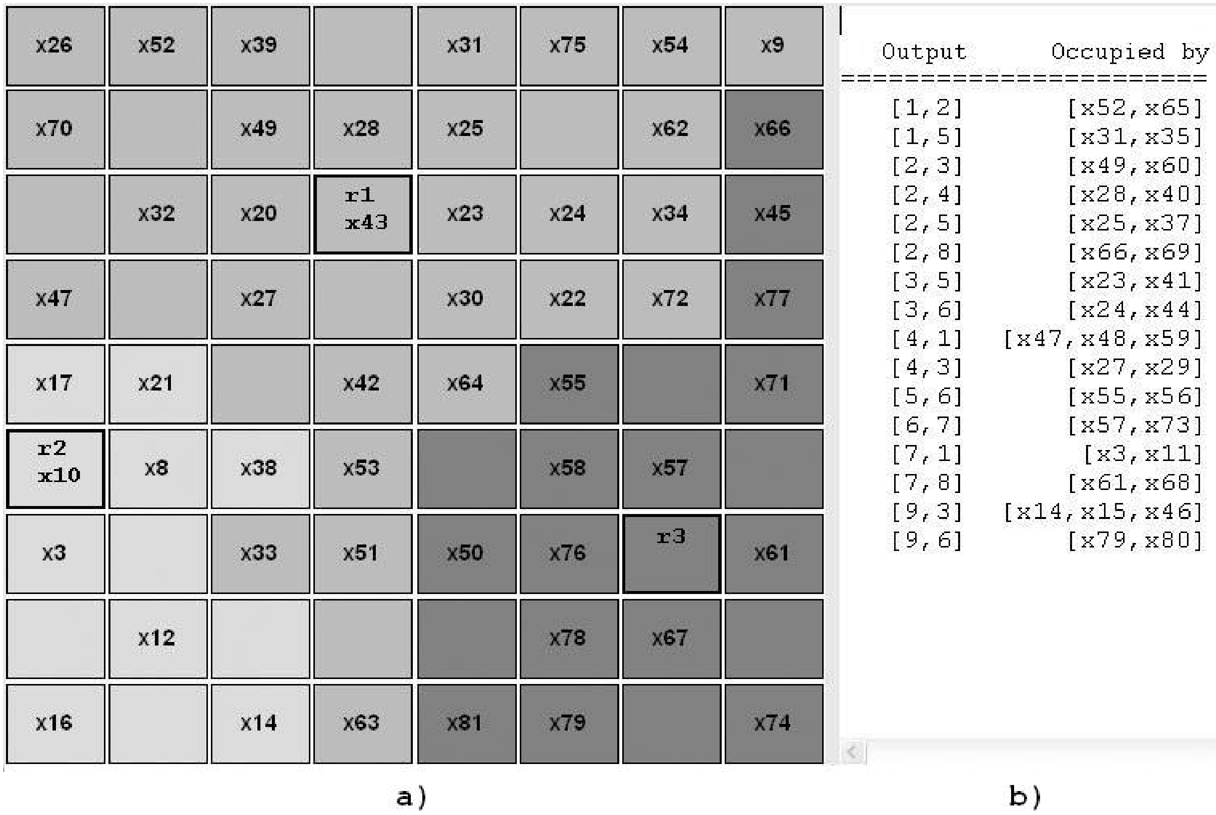

Figure 3. Classification of firms into three sets

CONCLUSION

The paper is focused on the classification of agricultural organizations into the specified number of classes. Using the Kohonen's map, that besides the information about classification offers also information about the result's trueness, it makes our method more favourable in comparison with other methods. This kind of information is important e.g. for the selection of customers, the evaluation of partners for a shared investment, the enterprise evaluation for obtaining a bank credit or for the enterprise strategy development.

Like the multi-neural networks, the self-learning network can be used for prediction, therefore, to determine the status of the followings. In that case, there would be an outline of the state in which the firm will be found in a certain time. This, however, needs historical data of the company in order to make a prognosis of its future state. This is an approach which may, especially at the beginning of an expert's decision-making process, facilitate gaining the first outline of the situation in a fairly quick way.

The performed classification described in this article with the aid of a neural network model was applied to the standardized data of 81 food manufacturing companies characterized by 15 financial indicators. The source of these items of data was the database Credit info.

Based on the presented data, the classification ability of the introduced model of Kohonen's network was demonstrated (the data was related to agricultural companies). Using this approach for analysis of a company can help to quickly and easily decide whether the company is healthy or in recession with respect to the historical data. This is often a crucial request for a successful business in the given field.

The results of the performed classification can be followed with the classification by the means of a multi-layer neural network with learning according to models, (Trenz, Konečný 2008) and it can refine the classification model as needed. It does not mean, however, that the self-learning network cannot be corrected. If desired, the set of elements can be changed, then according to the formula (7), the representatives can be calculated and by self-learning, a new output map will be created.

REFERENCES

Deboeck G., Kohonen T. (1998): Visual Explorations in Finance with Self-Organizing Maps. Springer-Verlag Limited, London; ISBN 3-540-76266-3.

Kohonen T. (2001): Self-Organizing Maps. 3rd ed. Springer-Verlag, Berlin; ISBN 3-540-67921-9.

Konečný V., Matiášová A., Rábová I. (2005): Learning of N-layers neural network. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis, LIII: 75–84.

Svoboda E. (2007): Knowledge-management in managerial work of business management. Agricultural Economics – Czech, *53*: 298–303.

Trenz O., Konečný V. (2008): Using Cohonen Neural Network for Multi-Layer Classification. SPU, Nitra; ISBN 978-80-552-0061-3.

Trenz O. (2006): Economic Decision Processes Support by Means of Artificial Intelligence Technologies. Alfa Publishing, Brno; ISBN 80-86851-62-1.

Arrived on 1st June 2009

Contact address:

Oldřich Trenz, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic
e-mail: xtrenz@node.mendelu.cz