ACTA UNIVERSITATIS AGRICULTURAE ET SILVICULTURAE MENDELIANAE BRUNENSIS

Volume LXI 43 Number 2, 2013

http://dx.doi.org/10.11118/actaun201361020377

DATA ADJUSTMENT FOR THE PURPOSES OF SELF-TEACHING OF THE NEURAL NETWORK, AND ITS APPLICATION FOR THE MODEL-REDUCTION OF CLASSIFICATION OF PATIENTS SUFFERING FROM THE ISCHEMIC HEART DISEASE

Vladimír Konečný, Oldřich Trenz, Milan Sepši

Received: January 14, 2013

Abstract

KONEČNÝ VLADIMÍR, TRENZ OLDŘICH, SEPŠI MILAN: *Data adjustment for the purposes of self-teaching of the neural network, and its application for the model-reduction of classification of patients suffering from the ischemic heart disease.* Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis, 2013, LXI, No. 2, pp. 377–384

Neural networks present a modern, very effective and practical instrument designated for decision-making support. To make use of them, we not only need to select the neural network type and structure, but also a corresponding data adjustment. One consequence of unsuitable data use can be an inexact or absolutely mistaken function of the model.
The need for a certain adjustment of input data comes from the features of the chosen neural network type, from the use of various metrics systems of object attributes, but also from the weight, i.e., the importance of individual attributes, but also from establishing representatives of classifying sets and learning about their characteristics.
For the purposes of the classification itself, we can suffice with a model in which the number of output neurons equals the number of classifying sets. Nonetheless, the model with a greater number of neurons assembled into a matrix can testify more about the problem, and provides clearer visual information.

classification, neural network, self-teaching, set representative, data standardization, data reduction

Object classification is the basis for resolving many tasks from the field of directing and investigating systems of a differing nature. It provides information about what is good or what is bad, about the consequences of a decision made earlier on, or it helps decide about further development of the system. Aside from conventional mathematical methods, with the development of the application of artificial intelligence methods, the artificial neural networks are often being used for classification. Not solely for their simplicity, but also for the possibility of result visualization with a comparable and often higher precision.

A multi-level neural network, taught (or rather: trained) by a teacher – according to patterns – need classifying patterns, according to which the network parameters will be set in the course of the teaching. To put it simply, this neural network models the function set by the learning patterns. Unknown objects are then classified in a calculation regime according to a modeled function, or rather by several modeled functions.

If there are, for instance, classifying patterns of products according to n parameters, it is simple to create a classification model with a multi-layer neural network; nevertheless, a frequent problem is the existence of N objects with n parameters, while the classification is unknown, and so is often the method, or the method is so complex, that is it practically unusable. In such a case, we may use another neural network with self-teaching. During the visualization of classifying sets, e.g., in the form of Kohonen's map, we may then gather information not only to which subset the investigated object pertains, but also with what certainty, established by the distance from the set representative, a given set is the closest and which objects are on the border.

MATERIAL AND METHODS

Self-teaching

A self-taught neural network is used in situations when, in the case of known input values of the modeled system, the output values are unknown. Many applications using this type of neural networks solve the problem of Xi object classification defined by certain measurable x1 markers, which form an input vector.

$$Xi = (x_{1i}, x_{2i}, x_{3i}, \dots, x_{ni}). \tag{1}$$

Further, we will propose a classification model with output neurons assembled into a grid in the dimensions ($r \times s$). For the classification itself, it is sufficient to arrange the output neurons in the form ($1 \times s$) or rather ($s\times 1$). The membership of the X_i object to the M_k set is signaled by the activation of the k-th neuron of the output neuron with the R_k vector.

$$R_k = (w_{1k}, w_{2k}, w_{3k}, \dots, w_{nk}) \tag{2}$$

(see Fig. 1), for which it holds that

$$M_k = \{X_i; |R_k - X_i| \le |R_j - X_i|, X_i \in E, j = 1, 2, \dots, s, j \ne k\}, \tag{3}$$

where E is the set of vectors of the input X_i objects. Generally speaking, ($r \times s$) of outputs of a neural network enables the classification into $r \times s$ sets, i.e., every output represents one set of objects. The semantics of each set is given by the semantics of the objects which it contains.

In the process of self-teaching, the X_i vectors are being entered into the input of the neural network in random order, and by the same, the correction is being made of those vectors of the R_k output, for which it holds that:

$$|R_k - X_i| < |R_j - X_i|, j = 1, 2, \dots, rs. \tag{4}$$

The correction of the R_k vector is done according to the relationship

$$R_k^* = R_k + \alpha(T)(X_i - R_k), \tag{5}$$

where $\alpha(T)$ is the learning coefficient, which decreases with the increasing phase of T- learning. Aside from the vectors of the so-called victorious R_k neurons, the correction of vectors neighboring with the victorious neuron is also made. The correction coefficient of these neurons $\beta(T, D, R(T))$ is dependent on the coefficient α, the distance of the neighboring neuron from the victorious neuron D and the radius of the neighborhood of R, which decreases with the increase of the number of teaching phases. The teaching algorithm is described in detail in publications (Konečný, Trenz, 2009), (Kohonen, 2001), etc., and its use in solving problems, for instance in publications (Konečný *et al.*, 2012), (Weinlichová, 2010).

Representatives of sets

By performing the vector correction of neurons neighboring with the victorious neuron, the algorithm implements the rule of proximity, which causes that near objects in real space will be near each other also in graphic representation. For *Ri* vectors of output neurons in linear arrangement – see Fig. 1, we can, after the teaching of the network, apply the relationship:

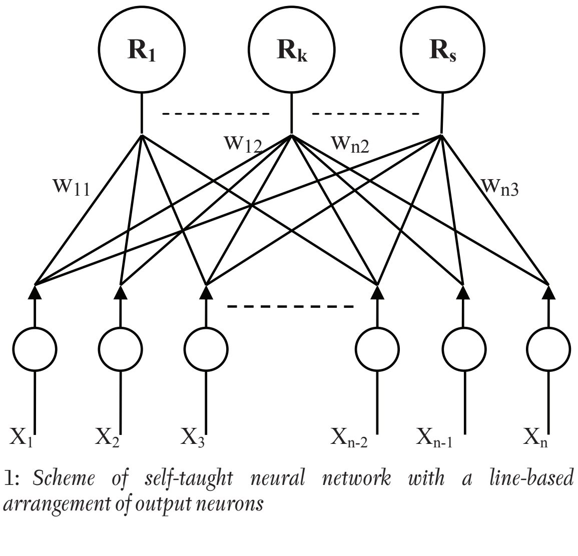
$$|R_i - R_{i-1}|\ RE\ |R_i - R^{i+1}|, \tag{6}$$

where *RE* is the relation „<“ or „>“ depending on whether the distance $|R_i - R_{i+1}|$ forms a growing or decreasing sequence.

In the case that the number of input objects is greater than the number of output neurons, every output neuron represents several objects, or rather, the entire set, and with the proximity rule in mind, we are dealing with those objects whose vectors X are closest to the output vector R_k. This vector is the position vector of the centre of gravity of the represented set, for which the following holds:

$$R_k = \frac{1}{N_k}\sum_{i}^{N_k} X_i\, , \tag{7}$$

where N_k is the number of elements of the represented set.



1: *Scheme of self-taught neural network with a line-based arrangement of output neurons*

The representatives of the classifying sets may be also established in different ways. One of the simple and often used methods is the K-means method. A comparison of these methods is shown in the publication (Fejfar *et al.*, 2010). From a practical point of view, it may be very useful to perform a partial expert classification, and a subsequent specification by the K-means method. It is necessary to note that the classifications done via the neural network and the K-means method do not have to be equal (Araújo, 2012).

Although the model where the output quantity equals the number of classification sets is sufficient for the purposes of classification, the model with a planar arrangement of the output neurons provides, as it has been said in the introduction, more information. In this model also, after the self-teaching process has been terminated, the output neuron whose vector is closest to the input object's vector, corresponds to every input object. In the case that, in the set of input objects there are very near objects, these will evidently be represented by a common output neuron.

Given a suitable distribution of objects in the input space, and the equality of number of input objects and output neurons, every input object will be represented by an independent output neuron and the classifying sets will then form coherent areas (Fig. 2b). Especially while displaying objects for instance from 10- and more-dimensional space (Konečný *et al.*, 2012), it may be difficult to attain a coherent graphic representation (Fig. 2a). The question then arises, whether it is possible to perform such an adjustment of the teaching file, so as to attain a coherent display of classification sets, and so as not to – or only to a minimum degree – disrupt the classification ability of the neural model.

Input data

With the exception of mistakes created, e.g., when recording or measuring and causing extreme deviations from common values, the erroneous function of the model can be caused by the metrics used. Random mistakes, as well as the used metrics, influence the size of the coordinates of the object vectors, and this way also their position in external space, but also in graphic representation.

The problem of a large distance $d(X, R_k)$of objects X from the R_k set representative can be solved by eliminating objects X, if it holds that:

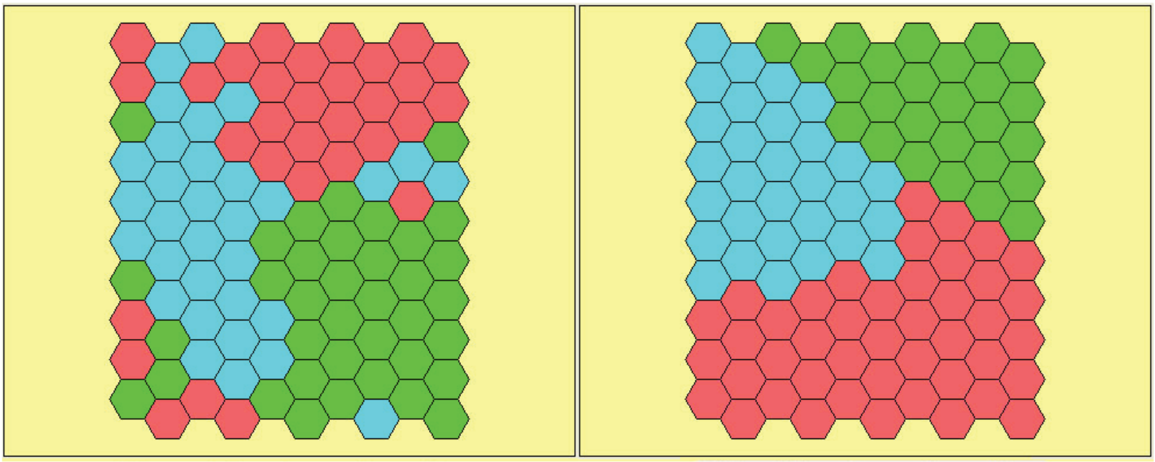
$$d(X,R_k) > D_p = [\overline{d(X,R_k)} + S(R_k)]K_{dil}, \qquad (8)$$

where D_p is the acceptable distance, $\overline{d(X,R_k)}$ the mid-value of distances, and $S(R_k)$ – the standard deviation of distances between objects X and the representative of the M_k set. It is possible to amplify or narrow down the acceptable distance – depending on the character of the objects – by changing the coefficient K_{dil}.

Different metrics can cause substantial deviations in the values of vector coordinates. In such a case, the object's position is dependent mostly on the coordinates with large values, which then practically eliminate the influence of coordinates with small values. The situation can be resolved by standardizing the vectors' coordinates and thus to equalize the weight of all coordinates on the same level, or by setting the standard deviations of coordinates of the object vectors according to the entered transformation weights

$$x^*_{ji} = \frac{(x_{ji} - \overline{X}_j)}{S_j} \frac{w_j}{\sum_i w_i} n \quad , \qquad (9)$$

$\overline{X}_j$ – is the mid-value of the *j-th* coordinate, w_j – weight of the coordinate j and n – the number of vector coordinates, and X a S_j standard deviation.



a) Incoherently mapping

b) Continuous mapping

2: *Graphic representation of classification sets by assembling neurons into a grid 10 x 10*

$$S_j = \sqrt{\frac{1}{N_x}\sum_i (x_{ji} - \overline{X}_j)^2} \ .$$

N_x – is the total number of X vectors. Given the same weights w_j, the standard deviation of all the coordinates will equal one. If the standard deviation should equal one, the coordinate value is constant and thus has no influence on the objects' variance. A larger standard deviation then means a greater weight of the given vector coordinate. The issue of the input data adjustment for the purposes of statistical data processing is dealt with in greater detail in the publication (Meloun, 1996).

A consequence of the rule of proximity implemented by the self-teaching algorithm has a decisive influence on the object classification their distance from the set representatives. Problems with classification and display are caused by those objects *Xi, Xj* which have the same or nearly the same distance from the representative of the set

$$d(X_i, R_k) \cong d(X_j, R_k), \qquad (10)$$

but a large mutual distance

$$|X_i - X_j| >> 0. \qquad (11)$$

Such objects are, bearing the set representatives (centres of gravity) in mind, placed on the surface of an *n-dimensional sphere*. Substituting these objects by their centre of gravity, we can optimize the self-taught neural network model with the output into a planar grid and we can create a coherent display of classificatory sets. A reduction thus made causes the compression of these sets, because the new object represented by the centre of gravity will always be closer to the representative of the classifying set than the objects of the *n-group*. The substituted *n-group* objects must be recorded into the testing file, and used for the evaluation of the neural network model.

Process of input data adjustment

On the basis of the abovementioned analysis of the self-teaching process and of the problems with the input objects data, we may, for the purpose of creating a classificatory model with output onto a planar grid, proceed as follows:

1. Perform standardization with the purpose of equalizing the weights to the same standard data deviation of all the input object vectors' coordinates, or standardization with standard deviations corresponding to the coordinates' weights according to the relation (9).
2. Establish the number of set representatives by means of the self-taught neural networks, with the number of outputs equal to the number of classifying groups.
3. Elimination of objects having a greater distance from the group representatives than is the acceptable distance and perhaps establishing new representatives of classifying sets.
4. Substituting *n-groups* of input objects by the centre of gravity and classifying the removed objects in the tested file. For the primary data reduction of the teaching file we may select Δ < 0, 1, and if the model's function is insufficient, we must select Δ in such a way that the mistake function decreases and the classification mistake of the tested objects is minimal.
5. Perform the self-teaching with the required dimensions of the planar representation.

RESULT AND DISCUSSION

Randomly generated data

For the purposes of the presentation of the process, an example has been selected of randomly generated data of one hundred vectors of objects with ten attributes (coordinates), which were standardized in such a way that the mid-value of all the coordinates be equal to zero and the standard deviation equal to one. For classification into three sub-groups, the self-teaching of the neural network with three outputs has set the vectors of the representatives of classification groups and, according to the rule of proximity (3), their elements were established as follows:

Requested classification:

$M_1(R_1) = [X_3, X_{10}, X_{11}, X_{12}, X_{17}, X_{19}, X_{20}, X_{21}, X_{22}, X_{23},$
$X_{25}, X_{27}, X_{32}, X_{33}, X_{40}, X_{44}, X_{48}, X_{55}, X_{56}, X_{57},$
$X_{60}, X_{61}, X_{63}, X_{64}, X_{66}, X_{68}, X_{71}, X_{72}, X_{73}, X_{75},$
$X_{76}, X_{77}, X_{80}, X_{83}, X_{88}, X_{89}, X_{93}, X_{99}],$

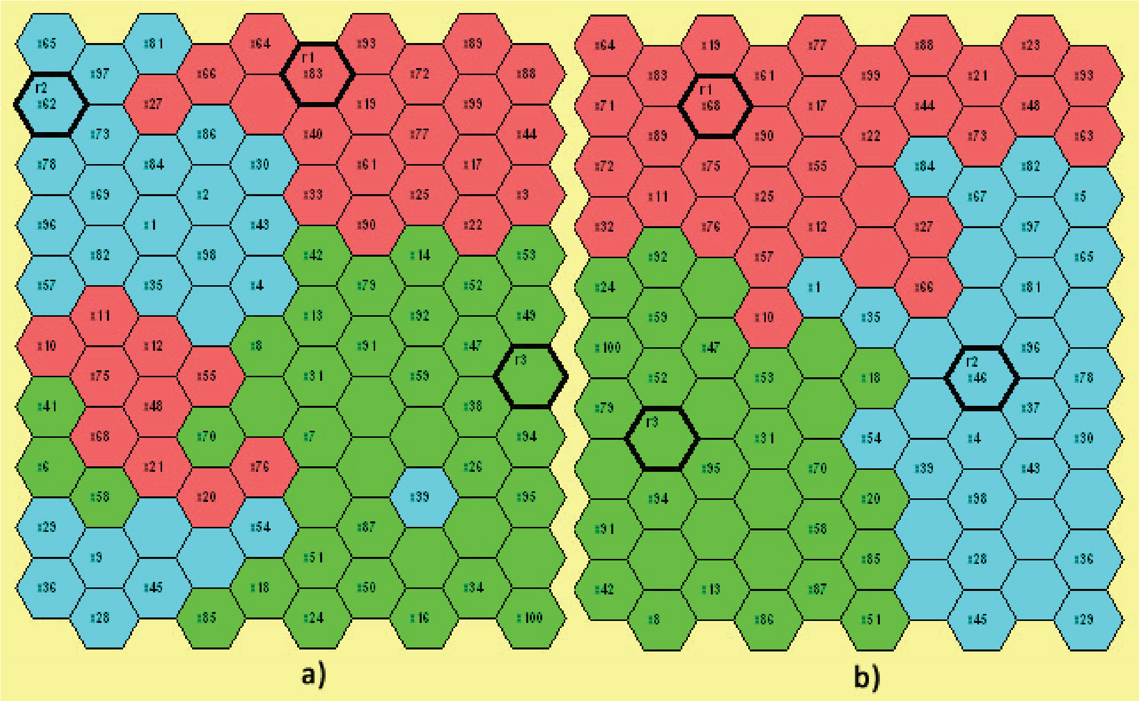
$M_2(R_2) = [X_1, X_2, X_5, X_9, X_{15}, X_{28}, X_{29}, X_{30}, X_{35}, X_{36},$
$X_{37}, X_{43}, X_{45}, X_{46}, X_{54}, X_{62}, X_{65}, X_{67}, X_{69}, X_{74},$
$X_{78}, X_{81}, X_{82}, X_{84}, X_{96}, X_{97}, X_{98}],$

$M_3(R_3) = [X_7, X_8, X_{13}, X_{14}, X_{16}, X_{18}, X_{24}, X_{26}, X_{31}, X_{34},$
$X_{38}, X_{41}, X_{42}, X_{47}, X_{49}, X_{50}, X_{51}, X_{52}, X_{53}, X_{58},$
$X_{59}, X_{70}, X_{79}, X_{85}, X_{86}, X_{87}, X_{91}, X_{92}, X_{94}, X_{95}].$

The set contents established this way are considered correct, i.e., requested, and it is desirable that they be achieved also using the model which assembles the output neurons into a grid 10 × 10, which reserves one output neuron for every object.

Assigning input objects after the teaching of the neural network is shown on Fig. 3a. The picture does not correspond entirely to the requisites of the proximity rule, because the function of the self-teaching algorithm is disrupted by an unsuitable structure of input data. This is also shown by the value of the error function, which is not equal to zero in various beginning conditions of the self-teaching process.

Nonetheless, the classification differs only in classifying objects X_{27} and X_{63}. Both objects are classified in the set $M_2(R_2)$ instead of $M_1(R_1)$.



3: *Display of input objects by the outputs of the neural network in a 10 × 10 neuron grid*

According to the third point of the mentioned proceedings, it will be suitable to eliminate, from the teaching file, those objects whose distance from the representative is greater than the acceptable distance.

I: *Distances of objects from representatives*

R_1		R_2		R3	
Object	**Distance**	**Object**	**Distance**	**Object**	**Distance**
X_{83}	1.745	X_{82}	2.136	X_{92}	1.785
X_{99}	2.014	X_{62}	2.143	X_{59}	1.894
…	…	…	…	…	…
X_{23}	3.491	X_{36}	3.273	X_{53}	3.276
X_{64}	3.645	X_{29}	3.397	X_{58}	3.367
X_{55}	3.729	X_{39}	3.653	X_{6}	3.696
X_{90}	3.986	X_{4}	3.919	X_{100}	3.746

After establishing the representatives of the classifying groups, their distances from the representatives have been set, the mid-values of the distance from the representatives have been calculated, standard deviations and – based on the relation (8) – the acceptable distances $D_{max}(R_i)$ have been amplified by 12 %. The distances of the nearest and furthest objects are listed in Tab. I.

From Tab. I it is evident that the acceptable distances

$$d_{\max}(X,R_1) = [\overline{d(X,R_1)} + S(R_1)] \times 1{,}12 = [2{,}886+0{,}514] \times \ 1{,}12=3{,}808\,,$$

$$d_{\max}(X,R_2) = [\overline{d(X,R_2)} + S(R_2)] \times 1{,}12 = [2{,}734+0{,}445] \times \ 1{,}12=3{,}808\,,$$

$$d_{\max}(X,R_3) = [\overline{d(X,R_3)} + S(R_3)] \times 1{,}12 = [2{,}724+0{,}498] \times \ 1{,}12=3{,}609\,,$$

surpass objects X_{100}, X_6, X_{39}, X_4 and X_{90}. These objects are removed from the teaching file, and inserted into the file of the testing objects.

In the following step, the *n-groups* of near objects will be removed from the teaching file and inserted into the file of the testing objects. Objects considered identical are those whose greatest distance difference does not surpass the value 0,018. We are speaking of the following pairs and triplets of objects of the classifying groups M_1, M_2 and M_3:

$$M_1 \Rightarrow [[X_{60}, X_{68}, X_{25}]_, [X_{44}, X_{73}], [X_{57}, X_{89}, X_3, X_{20}]],$$

$$M_2 \Rightarrow [[X_{82}, X_{62}]], [X_{37}, X_2, X_{46}, X_{15}], [X_9, X_{45}], [X_{81}, X_{96}], [X_{35}, X_{84}],$$

$$M_3 \Rightarrow [[X_{14}, X_{26}]], [X_{86}, X_{50}],[X_{34}, X_{94}], [X_{79}, X_{16}, X_{38}], [X_{85}, X_{41}], [X_{49}, X_{87}], [X_{31}, X_{53}].$$

Also objects substituted by the centre of gravity are re-arranged, from the teaching file, into the testing file, and used for checking the functionality of the taught neural network. Their classification must be the same, as if they had remained in the teaching file. The graphic representation of the classifying groups on the outputs of the neural network, after the process of self-teaching with an altered teaching file, is shown on Fig. 3b.

The classification abilities of the neural network have not, having reduced the input data, decreased. The real classification of all the input objects differs from the requested classification only in one case. Object X_{20} is classified in the M_3 group and is supposed to be in the M_1 group.

II: *Results of the tested classifying models*

No.	N_{obj}	N_x	$R_o \times S_o$	N_k	K_{dil}	Δ	N_{xpr}	N_t	N_{ch}	Chkl %	Indication used
1	36	8	6 × 6	3	1.05	0.02	28	10	1	10	N_{obj} – *number of input objects*
2	36	8	6 × 6	2	1.1	0.035	25	18	1	5.50	N_x – *number of object attributes*
3	64	12	8 × 8	3	1.0	0.02	48	29	1	3.45	$R_o \times S_o$ – *output grid*
4	64	12	8 × 8	2	1.08	0.02	45	36	1	2.77	N_k – *number of sub-groups*
5	100	15	10 × 10	3	1.1	0.06	47	85	4	4.7	K_{dil} – *dilation coefficient*
6	100	15	10 × 10	2	1.1	0.06	37	91	2	2.20	Δ – *reduction*
7	225	20	15 × 15	3	1.1	0.055	68	214	8	3.74	N_{xpr} – *number of objects after the reduction*
8	225	20	15 × 15	2	1.1	0.05	58	213	9	4.22	Chkl% – *classification errors %*
9	225	30	15 × 15	3	1.1	0.07	68	186	7	3.76	N_t – *number of objects tested*
10	225	30	15 × 15	2	1.1	0.05	59	196	5	2.55	N_{ch} – *number of classification mistakes*

The teaching error in all cases < 0.001

The testing results of the mentioned proceeding of random-generated input data reduction and classifying-model creation are summed up in Tab. II. The vector coordinate values of input objects were generated out of the interval (−3, 3) and subsequently standardized so that in all the coordinates the mid-value equals zero, and the standard deviation equals one. The two most common input-object classification modes – into two and three sub-groups – have been tested. The neural network teaching error is, in all cases, less than 0.001.

The perform experiments have shown that the teaching file is greatly reduced by substituting the *n-groups* of the *X*-objects, whose distance from R_k differs to a maximum by the value of Δ, with their centre of gravity, and by elimination of those objects that are further from the R_k set representative more than is the acceptable N_{pr} distance.

All the eliminated objects then create the testing file, and, as it is apparent from the table, most models have shown a smaller than 5.0% error occurrence in the classification of its elements. A greater classification error is shown by the first model, in which a single error represents 10% in the classification into three sub-groups, and 5.5% in the two-subgroup classification.

Classification of patients

The suggested input-data adjustment method has been used for the verification of a new risk-assessment classification model of patients with ischemic heart disease. This issue has already been listed in the article (Konečný *et al.*, 2012) with conclusions stating that the examined *complete model*

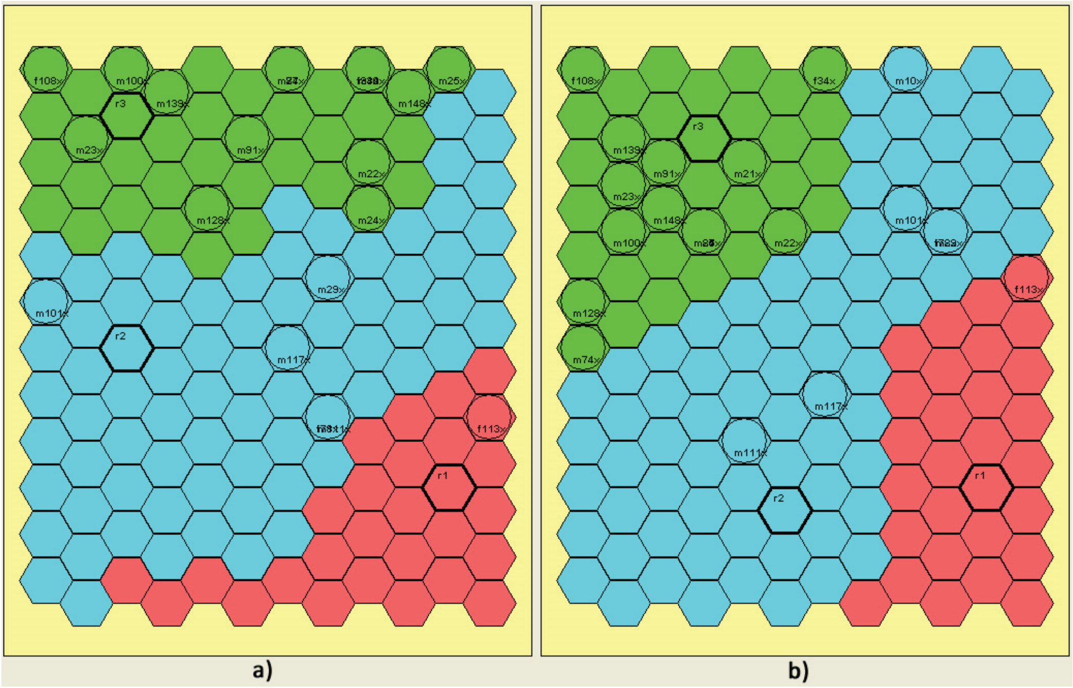
III: *List of parameters for the risk-assessment of patients with ischemic heart disease*

No.	Parameter	Mid-value	Standard deviation	No.	Parameter	Mid-value	Standard deviation
1	Sex	0.752	0.432	14	QRS width MI	1.426	0.852
2	Age	67.780	12.674	15	HR	75.468	18.954
3	ASA drugs	0.972	0.166	16	DC	4.759	3.053
4	BB drugs	0.844	0.363	17	AC	−5.340	3.423
5	ACEI drugs	0.858	0.349	18	Holter VPB	1.319	0.645
6	STATIN drugs	0.809	0.393	19	HRV VLFreg	25.044	9.487
7	DIURETICS drugs	0.383	0.486	20	HRV LFreq	14.800	6.500
8	HLP	1.482	0.500	21	HRV HFreq	9.563	4.014
9	HT	1.447	0.497	22	HRV Mean NN	873.794	139.367
10	DM	1.738	0.440	23	HRV SDNN	120.241	39.808
11	Revascularization	1.695	0.789	24	HRV SDANN	107.206	36.686
12	Vessel Disease	1.794	0.879	25	HRV ASDNN	44.993	15.741
13	LVEF stratification	45.709	12.053	26	HRV RMSSD	25.411	10.238

Abbreviations:
ASA: acetylsalicylic acid, BB: betablockers, ACEI: angiotensin-converting-enzyme inhibitor, HLP: hyperlipoproteinemia in anamnesis, HT hypertension in anamnesis, DM: diabetes mellitus in anamnesis, LVEF: left ventricular ejection fraction, MI: myocardial infarction, HR: heart rate, DC: deceleration capacity, AC acceleration capacity, VPB: ventricular premature beats, HRV heart rate variability. VLFreq: very low frequencies, LFreq: low frequencies, HFreq: high frequencies, Mean NN: mean normal-to-normal interval, SDNN: mean of all the 5-minute standard deviations of NN, SDANN: standard deviation of the NN intervals, ASDNN: mean of standard deviations of NN intervals determined in 5-minute periods of 24-hour ECG, RMSSD: The square root of the mean squared differences of successive NN intervals

IV: *Model parameters*

Set	Complete model			Simplified model			Object number coinci- dence
	Number of patients	Died	Mortality	Number of patients	Died	Mortality	
$M(r_1)$	19	1	0.053	14	1	0.071	14
$M(r_2)$	61	7	0.115	60	5	0.083	52
$M(r_3)$	27	14	0.52	34	16	0.471	11



4: *Classification of patients with ischemic heart disease*

is impractical taking into consideration the difficult accessibility of some parameters. Having performed their analysis, a group of parameters shown in Tab. III has been proposed.

Classification of patients with ischemic heart disease is derived from the same data file, only four *HRV* attributes and the *EFV positive* attribute are exchanged for the *drugs* attributes. What medicaments the patient is supposed to be using can be found out easily, from a practical point of view there is, however, a problem with the difficulty of finding out whether (s)he is actually using them. An important result of the experiment will be an answer to the question how strongly the complete model's function is affected by the mentioned parameter exchange.

According to data listed in Tab. IV, the deceased patients' classifications are very alike. A greater difference, in terms of the number of equal elements, is visible only in $M(r_3)$ groups. Nevertheless, we may state that, with the performed ten-attribute exchange in mind, we may have expected greater differences.

The graphic representation of classification groups is shown in Fig. 4a. The highlighted frame shows the outputs which are closest to the representatives of classification sets. The circle shows the deceased patients. From the image we may gather that the most critical is the set with the "*r3*" representative, having the largest number of deceased patients.

In the next experiment, the model was further simplified – this by leaving out all the attributes pertaining to the drugs being used. Bearing in mind that the standard deviation of values of these attributes is in three cases smaller than 0.5 and in two cases smaller than one, these will not have an important influence on the distance of objects represented by position vectors of all patients' attributes, from other objects.

The achieved result has confirmed the supposition. The classification model without the "*drugs*" attribute coincides to a great extent with the model containing the "*drugs*" attributes. In the classification of objects of taught neural networks, four patients are classified in a different way, and – as part of the deceased patients test – only one is classified in a different way. In a further set of thirty-four tested objects, there were three objects that were classified in a different way. The results offer us the incentive to explore the influence of other parameters and, eventually, a change in their weights.

SUMMARY

The aim of the article is to analyze the issue of preparation (pre-preparation) of data for a further processing in the field of classifying tasks, and a suggestion of an adequate proceeding that would eliminate incomplete and imprecise input tasks on the level of input data. In the presented case we were dealing with adjusting data for the self-teaching neural network model, i.e., for cases when the known input data does not have known output value counterparts (teaching without the teacher). This is a frequently used classification approach, when the goal is to identify the group representatives, and – with the help of the proximity rule – to classify the data. The classification success precision of the chosen model can be best assessed on the basis of an expert analysis (expert assessment).

The model's input data may be burdened by a recording mistake, a measurement mistake, or by inadequately chosen metrics (dimension of the used unit of the given quantity). In the latter case, the input vectors' coordinates will have a substantial influence on the result. It is therefore necessary to eliminate these influences, even before we approach the classification itself. In this text, an approach is coined leading to diminishing a potentially undesirable influence of input values on the classification result. We are dealing especially with the introductory standardization on the same standard deviation. If it be necessary, we may establish a unified weight value. Further, we may eliminate objects that are more distant than is the acceptable group representative distance, eventually substitute more objects with a similar value by their joint centre of gravity. The goal of these adjustments is an improvement of the classifying abilities of the model.

The suggested input data optimization mode has been verified on random-generated data, where we have achieved, except for two occurrences, an error occurrence of less than 5%. The worst result has been obtained when the error occurrence reached 10%, and this mostly with objects which are located on the borders of two sets. Based on the performed experiments, we may evaluate the classification ability of neural-models as very good.

Apart from random-generated data, the approach has been verified on a part of the population of patients suffering from the ischemic heart disease, with an aim to classify their health risk. First, as a part of the optimizing, an exchange of parameters from a previously described complete model was performed, an exchange for more accessible (from a data-collection viewpoint) parameters. In the next model, we have achieved satisfactory classification ability.

REFERENCES

KONEČNÝ, V., TRENZ, O., 2009: Decision support with Artificial Intelligence. Folia Universitatis Agriculturae et Silviculturae Mendelianae, Folia II, 2009, 8, ISSN 1803-2109, ISBN 978-80-7375-344-3.

KOHONEN, T., 2001: Self-organizing Maps. 3rd edition, Springer Verlag Berlin. ISBN 3-540-67921-9.

KONEČNÝ, V., TRENZ, O., SVOBODOVÁ, E., 2010: Classification of companies with assistance of self-learning neural networks. Agricultural Economics – Zemědělská ekonomika. 2010, Vol. 56, č. 2, s. 51–58. ISSN 0139-570X.

FEJFAR, J., ŠŤASTNÝ, J., CEPL, M., 2012: Time series classification using k-Nearest Neighbours, Multilayer Perceptron and Learning Vector Quantization algorithms. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis. 2012. Vol. 60, No. 2, s. 69–72. ISSN 1211-8516.

ARAÚJO, D., NETO, A.D., MARTINS, A., 2012: Comparative study on information theoretic clustering and classical clustering algorithms. Proceedings of the 2012 5th International Joint Conference on Computational Sciences and Optimization, 2012, p. 181–184.

KONEČNÝ, V., SEPŠI, M., Trenz, O., 2012: Analysis of evaluation problems of the risk situation of patients suffering from ischemic heart disease. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis. 2012. Vol. 60, No. 2, s. 125–133. ISSN 1211-8516.

MELOUN, M., MILITKÝ J., 1996: Statistical processing of experimental data (in Czech). Univerzita Pardubice, Czech Republic; ISBN 80-7194-075-5.

WEINLICHOVÁ, J., FEJFAR, J., 2010: Usage of self-organizing neural networks in evaluation of consumer behaviour. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis. 2010. Vol. LVIII, No. 6, s. 625–632. ISSN 1211-8516.

ŠKORPIL V., ŠŤASTNÝ J., 2008: Comparison of Learning Algorithms. In: 24th Biennial Symposium on Communications. Kingston, Canada, pp. 231–234; ISBN 978-1-4244-1945-6.

Address

doc. Ing. Vladimír Konečný, CSc., Ing. Oldřich Trenz, Ph.D., Department of Informatics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic, MUDr. Milan Sepši, Ph.D., Department of Internal Cardiology Medicine – Institutions Shared with the Faculty Hospital in Brno – Institutions of Adult Age Medicine – Faculty of Medicine, Masaryk university in Brno, Jihlavská 20, 625 00 Brno, Czech Republic, e-mail: vladimir.konecny@mendelu.cz, oldrich.trenz@mendelu.cz, msepsi@fnbrno.cz