DESIGN OF KOHONEN SELF-ORGANIZING MAP WITH REDUCED STRUCTURE

S. Kajan, M. Lajtman

Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Slovak Republic
slavomir.kajan@stuba.sk, lajtman@gmail.com

Abstract

This paper deals with design of optimal structure of Kohonen Self-organizing maps for cluster analysis applications. The cluster analysis represents a group of methods whose aim is to classify the objects into clusters. There have been many new algorithms solving cluster analysis applications, which used neural networks. This paper deals with the use of advanced methods of neural networks represented by Kohonen self-organizing maps for cluster analysis. For attainment of good results of cluster analysis is necessary to optimize the Kohonen network structure and algorithm parameters. There has been presented an example of a case study in Matlab software, where cluster analysis with optimization of Kohonen network structure and algorithm parameters is used.

1 Introduction

The cluster analysis represents a group of methods whose aim is to classify the objects into clusters. This paper deals with the use of an advanced use of neural network represented by Kohonen self-organizing map for cluster analysis. The basic principle of their function is cluster analysis, i.e. ability of algorithm, network, to find certain properties and dependencies just in the offered training data without presence of any external information. The idea of just network structure self-organizing has been formed for the first time at the beginning of seventies by von der Malsburg and later has been followed by Willshaw [8], [11]. All of their works from that time are characteristic by orientation to biological knowledge, mainly from the field of neuron and cerebral cortex research [6], [7], [10]. For attainment of good results of cluster analysis is possible to optimize the Kohonen network's structure and algorithm parameters.

2 Kohonen neural network

The basic idea of Kohonen self-organizing map emanates from knowledge, that brain uses inner space data representation for storing an information [6]. At first, data received from the environment are transformed to vectors which are encoded to the neural network. The extension of competitive learning rests on the principle that there are several winners permitted. The output from such neural network is geometrically organized to some arrangement, e.g. abreast, or to the rectangle and thus there is a possibility of neighbour identification. This layer is called Kohonen layer. Number of inputs entering to the network is equal to the input space dimension. In the Fig. 1, the topology of Kohonen self-organizing map with 2 inputs is depicted.

The neuron structure in Kohonen network is different from the neuron structure in perceptron network. Number of inputs entering to the neuron is equal to the number of inputs entering to the Kohonen network. The weights of these inputs serve for the encoding of patterns, which represent submitted patterns as well as in the case of a perceptron. These neurons don't have actual transfer function. The only operation, which neuron executes is calculation of distance (error) d of submitted pattern from pattern encoded in the weights of given neuron according to the equation:

$$d = \sum_{i=1}^{N} \left[x_i(t) - w_i(t) \right]^2 \qquad (1)$$

where N is a number of submitted patterns, $x_i(t)$ are individual elements of input pattern and $w_i(t)$ are appropriate weights of neuron which represent the encoded patterns [6], [7], [10].

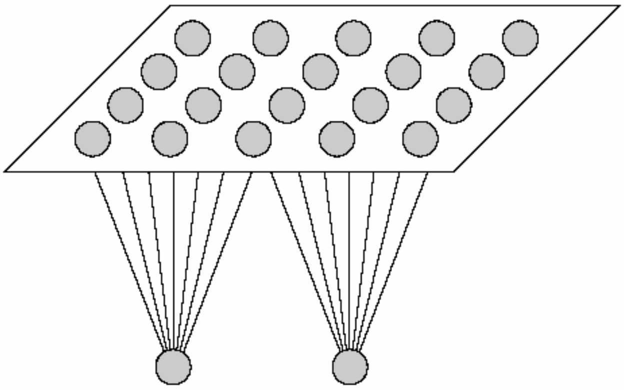

Fig. 1: Topology of Kohonen self-organizing map with two inputs

The learning algorithm tries to arrange the neurons in the grid to the certain areas so that they are able to classify the submitted input data. This organization of neurons can be imagined as unwrapping of originally wrinkled paper on the plane so that it covers the whole input space. The process of learning is autonomous, i.e. without the presence of external additional information and it is performed iteratively, i.e. weights are adapted in the each learning step. This adaptation is based on comparison of input patterns and vectors embedded in each neuron (1). Whenever the vector, which fits the input pattern the best is found, it's adapted as well as all the vectors of neurons situated near this neuron. Entire grid is gradually optimized to fit input space of the training data the best.

So-called surroundings of a neuron take a great role in the learning. The surroundings are defined for each neuron particularly. In initialization phase, the surroundings are usually chosen to cover all the neurons in the grid, i.e. radius of the surroundings is equal to the number of neurons on the one side of the grid. The surroundings are gradually reduced and also a parameter of learning is reduced similarly. The surroundings of neuron are calculated for given so-called winning neuron, whose vector of weights fits the input pattern the best. For this neuron and their surroundings, the weights are adapted. The neighbourhood function can be described by the equation:

$$\lambda\left(j^*, j\right) = h(t).\exp\left(\frac{d_E^2\left(j^*, j\right)}{r^2(t)}\right) \tag{2}$$

where $d_E(j^*,j)$ represents Euclidean distance of winner neuron j^* and another compared neuron, $r(t)$ is a radius of neighbourhood function and $h(t)$ is a height of neighbourhood function which decreases to the zero during the time whereby it provides decreasing the surroundings during the learning [6], [7], [10].

In the process of learning, the best results are achieved when the size of surroundings r is discreetly decreasing during the time and also it's useful to reduce the size of learning parameter $\eta(t)$ too. The parameter of learning is used to control the learning rate, whose value is from the interval $0 \le \eta(t) \le 1$. At the beginning it is efficient if the parameter is the maximum so that the network of neurons fans out as fast as possible and it covers the largest area of definition range of input vectors. This parameter is gradually decreasing and it allows the network the finer setting of output neuron weights. Adaptation of weights w_{ij} of neurons according to following equation:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t).\lambda\left(j^*, j\right)\left[x_i(t) - w_{ij}(t)\right] \tag{3}$$

After the learning phase follows the phase of execution, in which the network responds to the submitted pattern with classifying the pattern to the appropriate class [2, 4, 6, 7].

3 Cluster analysis using Kohonen self-organizing map

Principle of cluster analysis is ability of the algorithm to set respective neurons to the clusters of submitted patterns and thus to distribute submitted patterns into the clusters. In the Fig. 2, the common scheme of Kohonen self-organizing map allowing classification of inputs $X=(x_1,x_2,\ldots,x_n)$ into the classes is depicted. For debugging of the algorithm, program Matlab has been used on the example of classification of points with coordinates x_1, x_2 into the clusters. For testing, different numbers of neurons there have been applied. As an example, a Kohonen self-organizing map with 2 inputs and with 9 neurons in the grid 3x3 has been used [1-4], [9].

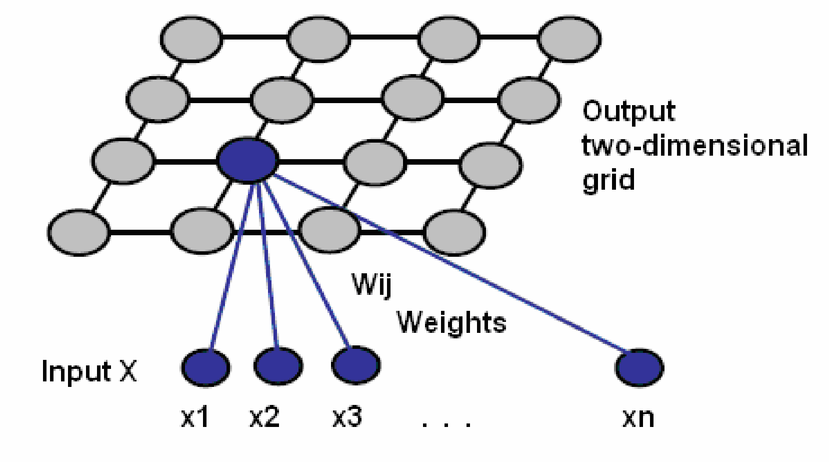

Fig. 2: Kohonen self-organizing map for cluster analysis

The aim of experiments was to set the initial parameters: learning step η_0, size of neighbourhood surroundings r_0, height of neighbourhood function h_0, number of learning cycles *num_cycles*, so that the learning algorithm described in the previous chapter performs the best. For decreasing of learning step η, radius of neighbourhood function r and height of neighbourhood function h in dependence on number of learning cycle, the following exponential dependencies have been used:

$$\eta(t) = \exp(\eta_0 / cycle) - 1 \tag{4}$$

$$r(t) = r_0 . \exp(-cycle) \tag{5}$$

$$h(t) = \exp((1 - h_0)/cycle) \tag{6}$$

Optimal setting of parameters of learning algorithm is noted in the Table 1. There are shown courses of decreasing of learning step η, size of neighbourhood surroundings r and height of neighbourhood function h in the Fig. 3 [4], [5]

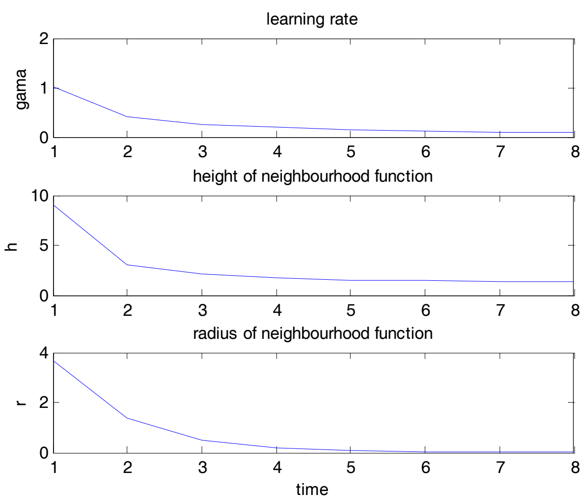

Fig. 3: Charts of algorithm parameters functions

TABLE 1: OPTIMAL INITIAL PARAMETERS OF LEARNING ALGORITHM

Number of cycles	Learning rate η_0	Height NF h_0	Radius NF r_0
8	0.7	1.2	10

For testing of optimal structure a Kohonen self-organizing map with 3 inputs has been used. Input data X witch coordinates (x_1, x_2, x_3) was randomly generated in twelve groups with random variance. In learning algorithm for Kohonen network with 3 inputs and 12 neurons, parameters shown in the Table 1 have been used. Cluster analysis results using Kohonen self-organizing map with 12 neurons is shown in Fig. 4.

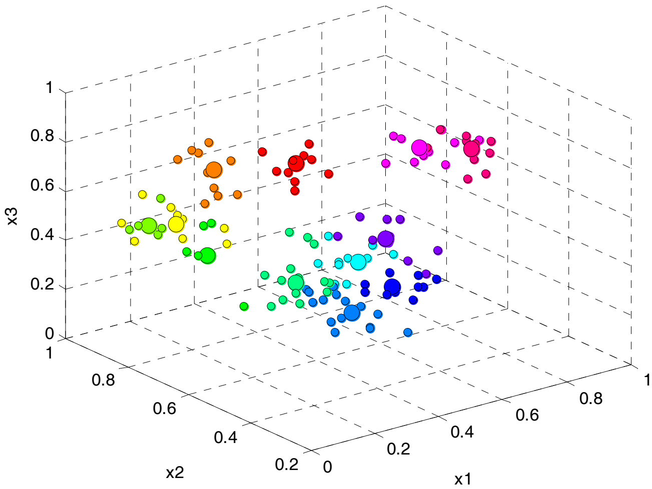

Fig. 4: Cluster analysis result using Kohonen self-organizing map

4 Structure reduction of Kohonen self-organizing map for cluster analysis

For the evaluation of achieved cluster analysis quality, we used criterion function (7), which was obtained from equation (1). The criterion function expresses the sum of distances of points to their nearest neurons.

$$J = \sum_{j=1}^{M} \min(d_j) = \sum_{j=1}^{M} \min\left(\sum_{i=1}^{N} [x_i(t) - w_{ij}(t)]^2 \right) \tag{7}$$

In the view of the classified groups (neurons), the criterion function J may be defined as the sum of distances between the neurons and their associated points,

$$J = \sum_{j=1}^{M} \sum_{k=1}^{P_j} \sum_{i=1}^{N} [x_{ik}(t) - w_{ij}(t)]^2 \tag{8}$$

where k is index of point in group and P_j are counts of points in groups.

The method for structure reduction of Kohonen self-organizing map is based on elimination of nearby neurons and neurons with small point count. In this method, the network learns only in the beginning with the maximum number of neurons. Subsequently the nearby neurons are connected and the neurons with small point count are eliminated. This method is faster, but more complicated to set conditions for elimination of neurons.

For connection of near neurons is necessary to calculate the distance *dn* between neurons.

$$dn_{lj} = \sum_{i=1}^{N} \left[w_{il}(t) - w_{ij}(t) \right]^2 \tag{9}$$

Connection of neurons occurs if the distance is less than 35% of the average distances between all neurons. After connection of *j-th* and *l-th* neuron, *j-th* neuron takes over the points *l-th* neuron. The weights of *j-th* neuron are adjusted according to the following equation:

$$w_{ij}(t) = \frac{\sum_{k=1}^{P_j + P_l} x_{ik}(t)}{P_j + P_l} \tag{10}$$

For elimination of neurons with small point count it is necessary to define the evaluative index, which will be calculated by following equation:

$$I_j = \frac{P_j^2}{\sum_{k=1}^{P_j} d_{jk}} \tag{11}$$

where P_j is count of points in *j-th* group and d_{jk} is distance between *j-th* neuron and *k-th* point from its group.

Elimination of neuron occurs, if the neuron index *I* is less than 80% of the average indexes of all neurons and simultaneously is a nearby neuron (50% of average neuron-to-neuron distances), which takes over all its points.

4.1 Example of Kohonen self-organizing map optimization for random generated data

The algorithm for optimization of Kohonen network structure was implemented in Matlab. The algorithm works as follows: At first start Kohonen network trains with maximal number of neurons *Mmax*, where *Mmax* equals 12. Consequently neurons without any points are eliminated. Next, the neurons are reduced by connecting them according to their distance (9) and (10). Finally, they are reduced by evaluative index (11). Result of the cluster analysis using Kohonen self-organizing map is shown in Fig. 5. In the Figure you can see, that algorithm reduced the network structure to 8 neurons. Criterion values function is depicted in Fig. 6.

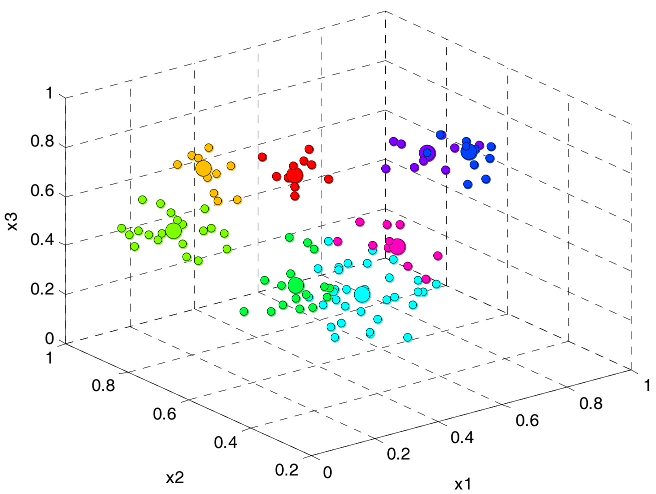

Fig. 5: Cluster analysis result using optimized Kohonen self-organizing map

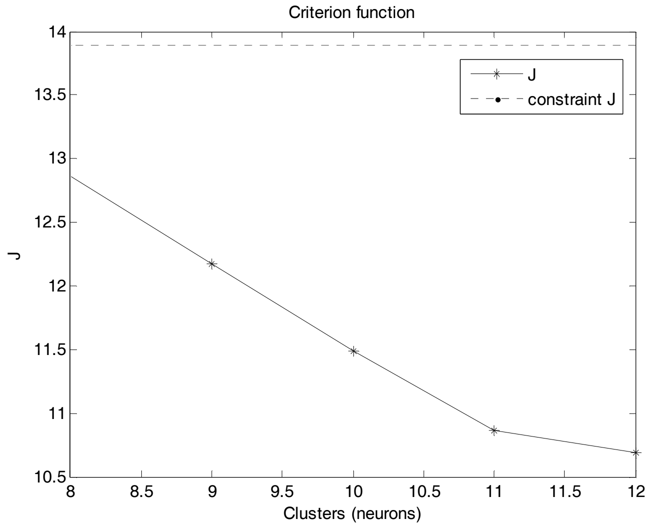

Fig. 6: Criterion values functions

4.2 Example of Kohonen self-organizing map optimization for data of house colours

The algorithm for optimization of Kohonen network structure was tested in 1800 samples of house colours [12]. The main aim of experiment is to find characteristic colours of houses in tested data. At first start Kohonen network trains with maximal number of neurons *Mmax*, where *Mmax* equals 20. Cluster analysis result using optimized Kohonen self-organizing map is shown in Fig. 7. In the Figure you can see, that algorithm reduced the network structure to 11 neurons, i.e. algorithm found 11 characteristic colours of houses. Criterion values function is depicted in Fig. 8.

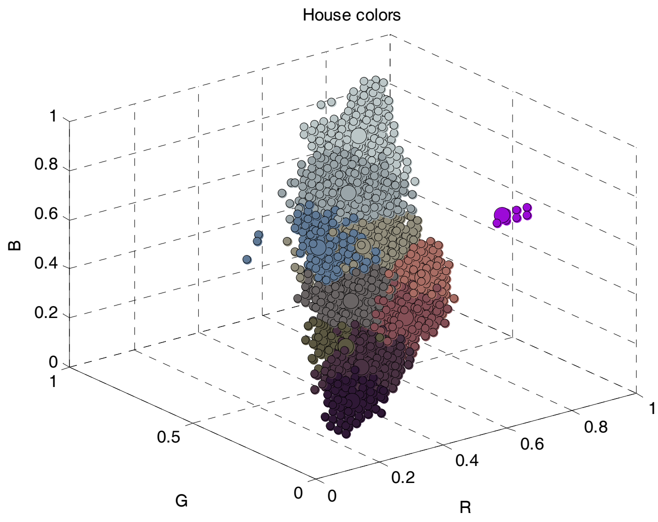

Fig. 11: Cluster analysis result using optimized Kohonen self-organizing map for house colours

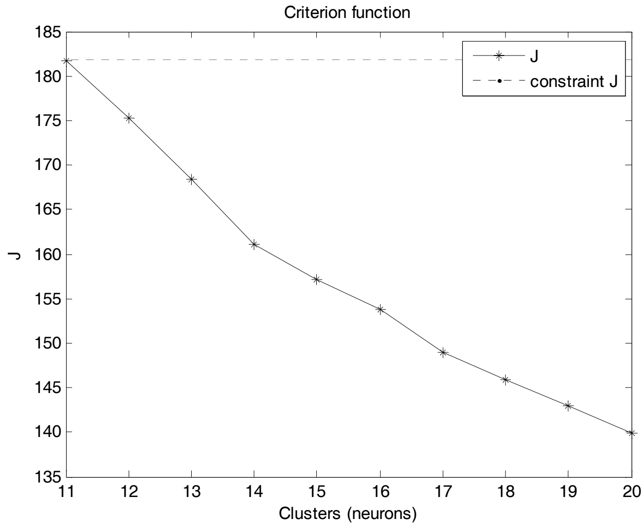

Fig. 8: Criterion values function

5 Conclusion

The main objective of this article was to demonstrate very good properties of Kohonen self-organizing map for cluster analysis problems. The created support programs for cluster analysis using Kohonen self-organizing map has been used for setting of initial parameters of learning algorithm. Due to good properties, the Kohonen self-organizing map can solve arbitrary classification problems very effectively. The realized algorithm intended to design an optimal structure of Kohonen self-organizing map is very convenient for complex classification practical problems.

Acknowledgement

The work has been supported by the grant agency VEGA no. 1/1241/12.

References

[1] H .Demuth, M. Beale: *Neural Network Toolbox, For use with Matlab*, User’s guide, 2003
[2] P. Dostál, P. Pokorný: *Cluster analysis and neural network.* In: Technical Computing Prague 2009: 17th Annual Conference Proceedings. Prague, Czech Republic, 2008.
[3] F. Gorgonio, J. Costa: *Privacy-Preserving Clustering on Distributed Databases: A Review and Some Contributions*, In: book Self Organizing Maps - Applications and Novel Algorithm Design, Published by InTech, Rijeka, Croatia, january 2011, p. 33, ISBN 978-953-307-546-4
[4] S. Kajan, I. Sekaj, M. Lajtman: *Cluster Analysis Application in Matlab Using Kohonen Network*, Technical Computing Prague, 19th Annual Conference Proceedings. Prague, (2011)
[5] S. Kajan, M. Lajtman, Z. Dideková: *Cluster Analysis Using Self-Organizing Maps*, ATP Journal plus. - No. 1 : Systems of automatic control, pp. 22-25, Bratilsva (2011)
[6] T. Kohonen: *Self-Organizing Map*, Proc. of the IEEE, Vol. 78, No 9, pp. 1464-1480, (1990)
[7] V. Kvasnička a kol.: *Úvod do teórie neurónových sietí.* Bratislava: IRIS, 1997. 142 s. ISBN 80-88778-30-1
[8] C. von der Malsburg: *Self-organization of orientation sensitive cells in the striate cortex*, Kybernetik (1973)
[9] L. Meza, L. Neto: *Modelling with Self-Organising Maps and Data Envelopment Analysis: A Case Study in Educational Evaluation*, In: book Self Organizing Maps - Applications and Novel Algorithm Design, Published by InTech, Rijeka, Croatia, january 2011, p. 33, ISBN 978-953-307-546-4

[10] P. Sinčák, G. Andrejková: *Neurónové siete Inžiniersky prístup (1. diel).* Košice: ELFA, 1996. ISBN 80-88786-38-X

[11] D.J. Willshaw, C. von der Malsburg: *How patterned neural connections can be set up by self-organization*, Proc R Soc Lond B Biol Sci., (1976)

[12] http://cs.joensuu.fi/sipu/datasets/, *Clustering datasets*, Speech and Image Processing Unit School of Computing University of Eastern Finland

---

Ing. Slavomír Kajan, PhD, E-mail: slavomir.kajan@stuba.sk
Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava

Ing. Miloš Lajtman, E-mail: lajtman@gmail.com
Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava