UDC 004.032.26

Ye. Bodyanskiy, O.Vynokurova, P. Mulesa, O. Slipchenko
Kharkiv National University of Radio Electronics,
Control Systems Research Laboratory

SELF-ORGANIZING MAP AND ITS LEARNING IN THE FUZZY CLUSTERING-CLASSIFICATION TASKS

In the paper, combined self-learning and learning method of self-organizing map (SOM-LVQ) is proposed. Such method allows to increase quality of information processing under condition of overlapping classes due to rational choice of learning rate parameter and introducing special procedure of fuzzy reasoning in the clustering-classification process, which occurs both with external learning signal ("supervised"), and without one ("unsupervised"). As similarity measure of neighborhood function or membership one, cosine structures are used, which allow to provide a high flexibility due to self-learning-learning process and to provide some new useful properties.

Key words: combined self-learning and learning method of self-organizing map, clustering, classification, overlapping classes.

Introduction

Self-organizing maps (SOM) and neural networks of learning vector quantization (LVQ) have seen extensive use for solving different problems in Data Mining domain (clustering, classification, fault detection and compression of information to name a few). This type of neural networks was proposed by T. Kohonen [1, 2] and represents, in fact, a single-layer feedforward architecture, which provides an operator for mapping of input space into the output space. Operation-wise SOM and LVQ are quite similar: each neuron is fed input signal (sample) producing output, which is used during competition stage to determine winning neuron – usually the one with maximum output signal value. Vector of synaptic weights for winning neuron is the one closest to the input sample in terms of the metric chosen (which is Euclidian metric in most cases). Next is neurons adjustment phase. Synaptic weights of the winning neuron gets moved closer to input sample. Alternatively, a subset of neurons (rather than a single one) can be adjusted – those determined to be "reasonably close" to the input sample are updated. Resulting network output, however, is determined exclusively by winning neuron (this principle is usually referred to as "Winner-Takes-All" (WTA)). It is this principle (WTA) which negatively affects accuracy in case when there are overlapping clusters in underlying data.

Taking into account the above mentioned properties of SOM and LVQ networks, it makes sense to introduce fuzzy classification capabilities on top of them, while preserving online operation. In [8, 9] fuzzy self-organizing map was proposed, in which conventional neurons are replaced by fuzzy rules. This neural network shows enough high efficiency, but its learning properties was significantly lost especially in on-line mode. In [5, 10, 11] fuzzy clustering Kohonen network and fuzzy linear vector quantization network are described. In fact such networks are neural representation of fuzzy c-means (FCM) [3], which is far enough from SOM and LVQ mathematical tool and designed for operation in batch mode.

Problem statement

Let us consider single-layer neural network with lateral connections containing receptors and neurons in the Kohonen layer with each neuron being characterized by a vector of its synaptic weights.

During learning stage input vector is fed to the inputs of all neurons (usually adaptive linear associators) (here - either the number of observation in a table “object-properties”, or current discrete time for on-line processing mode) and neurons produce the scalar signals on their outputs

$$y_j(k) = w_j^T(k)x(k),\ \ j = 1, 2, ..., m\,. \tag{1}$$

Note that neuron’s output depends on current values of synaptic weights vector, assuming iterative learning algorithm.

Each input vector can activate either single neuron (w_j) or a set of neighboring neurons – this also depends on learning algorithm chosen.

Self-organization procedure is based on the competitive learning approach (self-learning), and begins with the initialization of synaptic weights. Selecting initial values for weights from a uniform random distribution over input space has become a common practice. In addition, weights are usually normalized to reside on unit hypersphere:

$$\left\| w_j(k) \right\|^2 = w_j^T(k)w_j(k) = 1. \tag{2}$$

Self-organizing is usually divided into three stages [2]: competition, cooperation and synaptic adaptation. This paper introduces additional stage to this process – fuzzy inference, which allows an online learning algorithm to classify data samples belonging to several classes simultaneously.

Learning algorithm for SOM

The competition process is started to analysis of current pattern $x(k)$, which is fed to all neurons of Kohonen’s layer from receptive (zero) layer. For each neuron the distance between input sample and a vector of synaptic weights is computed:

$$D(w_j(k), x(k)) = \left\| x(k) - w_j(k) \right\|. \tag{3}$$

In case when all inputs and synaptic weights are normalized according to (2), i.e.

$$\left\| w_j(k) \right\| = \left\| x(k) \right\| = 1, \tag{4}$$

and Euclidian metric is used, the proximity measure between the vectors $w_j(k)$ and $x(k)$ can be in the following way:

$$w_j^T(k)x(k) = y_j(k) = \cos(w_j(k), x(k)) = \cos\theta_j(k)\,. \tag{5}$$

So the expression (3) takes the form

$$D(w_j(k), x(k)) = \sqrt{2(1 - y_j(k))}\,, \tag{6}$$

and type of dependence $D(w_j(k), x(k))$ from output signal value $y_j(k)$ is shown on fig. 1.

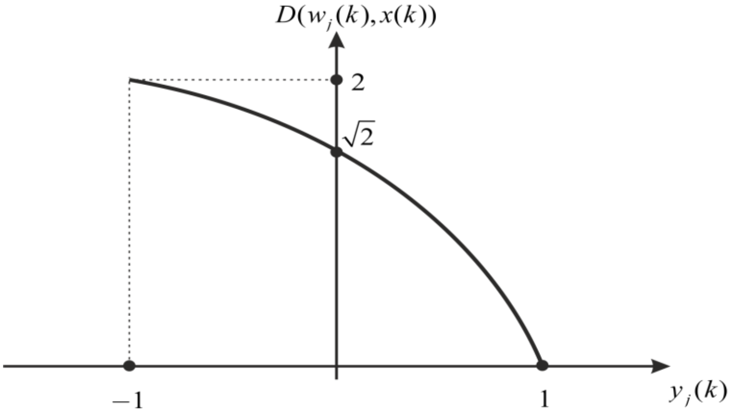

Fig. 1 – Dependence of distance $D(w_j(k),x(k))$ from output signal level for *j-th* neuron

Using relation (5), we can replace metric (3) with a simpler construction, usually referred to as a measure of similarity [12]

$$\psi(w_j(k),x(k)) = \max\left\{0, \cos(w_j(k),x(k)\right\} = \max\left\{0, \cos\theta_j(k)\right\}, \tag{7}$$

which satisfies softer conditions compared to classic measure requirements:

$$\begin{aligned} &\psi(w_j(k),x(k)) \geq 0, \\ &\psi(w_j(k),w_j(k)) = 1, \\ &\psi(w_j(k),x(k)) = \psi\left\{x(k),w_j(k)\right\} \end{aligned} \tag{8}$$

and which has a form shown on fig 2.

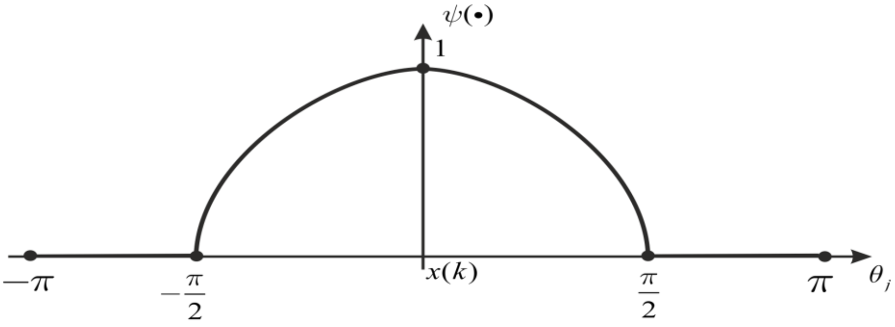

Fig.2 – The measure of similarity (7)

Next we look for a winning neuron that has a biggest value of similarity to the input vector in the sense (6) or (7), i. e.

$$D(w_j^*(k),x(k)) = \min_l D(w_l(k),x(k)) \tag{9}$$

or

$$\psi(w_j^*(k),x(k)) = \max_l \psi(w_l(k),x(k)). \tag{10}$$

After that we tune values of synaptic weights using WTA self-learning rule is realized in form

$$w_j(k+1) = \begin{cases} w_j^*(k) + \eta(k)(x(k) - w_j^*(k)), \text{ if } j - \text{th neuron won in the competition,} \\ w_j(k) \ \text{ otherwise.} \end{cases} \tag{11}$$

In a nutshell, expression (11) means that synaptic weights vector of winner is moved closer to the input pattern $x(k)$ by a value which is proportional to learning rate $0 < \eta(k) < 1$.

Choice of learning rate value $\eta(k)$ defines the convergence rate of self-learning process, and is usually based on empirical reasons, at that the parameter must monotonically decrease in on time.

It is easy to see, that first relation of the rule (11) minimizes the following criterion

$$E_j^k = \sum_{\tau=1}^{k_j} \left\| x(\tau) - w_j \right\|^2 \tag{12}$$

(here k_j is a number of data in the sampling with dimensions k, when j-th neuron was winner), notably in practice as synaptic weights estimation conventional arithmetic mean is used:

$$w_j(k) = \frac{1}{k_j} \sum_{\tau=1}^{k_j} x(\tau). \tag{13}$$

Therefore, in fact self-learning rule (3) is stochastic approximation procedure [13], and learning rate parameter $\eta(k)$ must be selected according to the condition of A. Dvoretzky [14]. It is know that using the following value

$$\eta(k) = \frac{1}{k_j} \tag{14}$$

leads to slowdown of the learning process.

Requirement of monotonic decreasing of the parameter $\eta(k)$ leads to expression in form

$$\eta(k) = r^{-1}(k),\ r(k) = \alpha r(k-1) + \left\| x(k) \right\|^2,\ 0 \le \alpha \le 1. \tag{15}$$

When taking into account normalization to unit hypersphere (4) we have:

$$r(k) = \alpha r(k-1) + 1, \tag{16}$$

Which with α =1 gives a well-known expression, that was introduced in [15].

Changing the forgetting parameter α provides enough variance for the learning rate to both satisfy Dvoretzky condition and adjust for characteristics of different data sets:

$$\frac{1}{k_j} \le \eta(k) \le 1. \tag{17}$$

Note that adjusting synaptic weights with (13) in genera breaks normalization (4). In order to maintain it, we should slightly modify our learning algorithm:

$$w_j(k+1)=\begin{cases}\dfrac{w_j^*(k)+\eta(k)(x(k)-w_j^*(k))}{\left\|w_j^*(k)+\eta(k)(x(k)-w_j^*(k))\right\|},\ \text{if } j-th \text{ neuron won in the competition,}\\ \eta(k)=r^{-1}(k),\ \ r(k)=\alpha r(k-1)+1,\ \ 0\le\alpha\le 1,\\ \\ w_j(k),\ otherwise.\end{cases}\tag{18}$$

In order to better adjust to input data, a so called "cooperation stage" is frequently introduced to SOM learning process. During this stage a winning neuron defines a local domain of topological neighborhood, where weights are adjusted for a set of neurons rather than only for a winning one. Those neurons closer to the winner receive a bigger adjustment.

This topological domain is defined by neighborhood function $\varphi(j,p)$, which depends on distance $D(w_j^*(k),w_p(k))$ between winner $w_j^*(k)$ and any of Kohonen's layer neurons $w_p(k)$, $p=1,2,\ldots,m$ and some parameter σ, which sets its width.

Usually $\varphi(j,p)$ is the bell-shaped function, symmetrical with respect to the maximum in point $w_j^*(k)$ ($\varphi(j,j)=1$) and decreasing when distance $D(w_j^*(k),w_p(k))$ increases. Gaussian function is commonly used here:

$$\varphi(j,p)=\exp\left(-\frac{\left\|w_j^*(k)-w_p(k)\right\|^2}{2\sigma^2}\right).\tag{19}$$

Adding neighborhood function results in the following learning algorithm:

$$w_p(k+1)=w_p(k)+\eta(k)\varphi(j,p)(x(k)-w_p(k))\ ,\ \ p=1,2,\ldots,m,\tag{20}$$

which minimizes criterion

$$E_p^k=\sum_{\tau=1}^{k}\varphi(j,p)\left\|x(\tau)-w_p\right\|^2\tag{21}$$

This criterion is commonly referred to as "Winner Takes More" (WTM).

The necessity to maintain normalization to unit hypersphere (4) leads to the expression in form

$$\begin{cases}w_p(k+1)=\dfrac{w_p(k)+\eta(k)\varphi(j,p)(x(p)-w_p(k))}{\left\|w_p(k)+\eta(k)\varphi(j,p)(x(p)-w_p(k))\right\|},\ p=1,2,\ldots,m,\\ \eta(k)=r^{-1}(k),\ \ r(k)=\alpha r(k-1)+1,\ 0\le\alpha\le 1.\end{cases}\tag{22}$$

It is possible to skip the entire competition by tuning synaptic weights of network depending on their similarity with the current vector-pattern $x(k)$. In this case instead of conventional neighborhood function $\varphi(j,p)$ depending on winner $w_j^*(k)$, we can use similarity measure (7), that depends on $x(k)$.

In this case instead of (20) we can use the rule in form:

$$
\begin{aligned}
& w_p(k+1) = w_p(k) + \eta(k)\psi(w_p(k), x(k))(x(k) - w_p(k)) = \\
& = w_p(k) + \eta(k)\max\left\{0, \cos(w_p(k), x(k))\right\}(x(k) - w_p(k)) = \\
& = w_p(k) + \eta(k)\max\left\{0, w_p^T(k)x(k)\right\}(x(k) - w_p(k)) = \\
& = w_p(k) + \eta(k)\max\left\{0, \cos\theta_p(k)\right\}(x(k) - w_p(k)) = \\
& = w_p(k) + \eta(k)\max\left\{0, y_p(k)\right\}(x(k) - w_p(k)),
\end{aligned}
\tag{23}
$$

which reminds «INSTAR» learning rule of S. Grossberg [16].

For maintaining (4) the rule (23) has to be rewritten in the form

$$
\begin{cases}
w_p(k+1) = \dfrac{w_p(k) + \eta(k)\psi(w_p(k), x(k))(x(k) - w_p(k))}{\left\| w_p(k) + \eta(k)\psi(w_p(k), x(k))(x(k) - w_p(k)) \right\|}, \\
\eta(k) = r^{-1}(k),\ r(k) = \alpha r(k-1) + 1,\ 0 \le \alpha \le 1.
\end{cases}
\tag{24}
$$

In many real world problems clusters can overlap as shown on fig 3, where * denotes patterns, belonging to j-th clusters, and o – p-th ones. In this case vector $x(k)$ belongs *j-th* cluster with proportional membership level $\cos\theta_j(k)$, and with proportional level $\cos\theta_p(k)$ - to *p*-th one. Synaptic weights concentrated in crosshatched region, don't have relationship to the pattern $x(k)$ according to (7).

Using similarity measure, that is shown on fig.2, we can introduce the membership estimate of pattern $x(k)$ to *j*-th class in form:

$$
0 \le \mu_{w_j(k)}(x(k)) = \frac{\psi(w_j(k), x(k))}{\sum_{l=1}^{m} \psi(w_l(k), x(k))} \le 1\,.
\tag{25}
$$

Learning algorithm for LVQ

Learning vector quantization neural networks in contrast to self-learning SOM adjust their parameters based on external learning (reference) signal, which fixes the membership of each pattern $x(k)$ to a particular class.

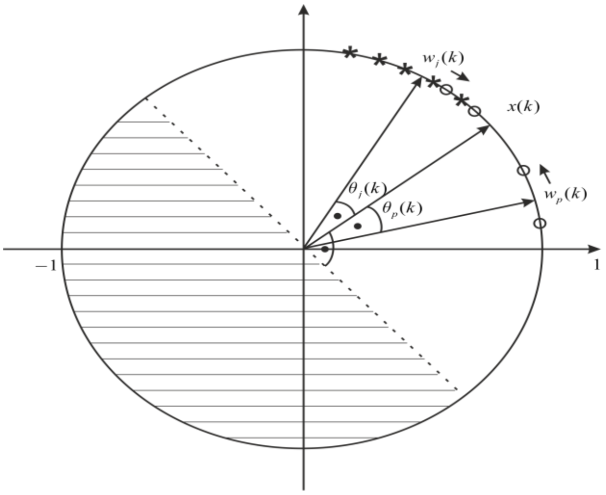

Fig. 3 – The overlapping clusters

The main idea of LVQ neural network is to build a compact representation of large data set based on a restricted set of prototype samples (centroids) $w_j(k)$, $j=1,2,\ldots,m$, that provide sufficiently accurate approximation of the original space X.

For each input vector $x(k)$ which is normalized according to (4), we look for a winning neuron $w_j^*(k)$ that corresponds to a centroid of a certain class. In other words, winner is defined as a neuron with minimal distance to the input vector (9) or, which is the same, with maximal similarity measure (10).

Since the learning is supervised, membership of the vector $x(k)$ to specific domain X_j of the space X is known, which allows considering two typical situation occurring in the trained linear vector quantization:

- the input vector $x(k)$ and neuron-winner $w_j^*(k)$ belong to the same cell of Voronoy [2];
- the input vector $x(k)$ and neuron-winner $w_j^*(k)$ belong to the different cells of Voronoy.

Than corresponding learning LVQ-rule can be written in form:

$$w_j(k+1)=\begin{cases} w_j^*(k)+\eta(k)(x(k)-w_j^*(k)), \text{ if } x(k) \text{ and } w_j^*(k) \text{ belong to the same cell,} \\ w_j^*(k)-\eta(k)(x(k)-w_j^*(k)), \text{ if } x(k) \text{ and } w_j^*(k) \text{ belong to the different cells,} \\ w_j(k) \text{ for neurons, which aren't won in instant } k. \end{cases} \tag{26}$$

The rule (26) has a clear physical interpretation: if the winning neuron and presented sample belong to the same class, than prototype $w_j^*(k)$ is moved to $x(k)$; otherwise prototype $w_j^*(k)$ is pushed away from $x(k)$, effectively increasing the distance $D(w_j^*(k),x(k))$, i.e. solving the maximization task based on criterion (12).

As for the choice of learning rate parameter $\eta(k)$, that common recommendations are the same that for SOM, i.e. the learning rate parameter must monotonically decrease in process controlled learning.

In [17] it was proved that LVQ tuning algorithm converges in case of learning rate $\eta(k)$ satisries Dvoretzky conditions. This, in turn, allows choosing $\eta(k)$ according to Goodwin-Ramadge-Caines algorithm [13], or in the previously defined form (16) with $\alpha=1$, which is essentially the same. When $\alpha<1$, the algorithm gets tracking properties and handles the case when class centroids are drifting.

For providing to fulfillment of the condition (4) it possible to introduce modification of rule (26) in the form (18):

$$w_j(k+1)=\begin{cases} \dfrac{w_j^*(k)+\eta(k)(x(k)-w_j^*(k))}{\left\|w_j^*(k)+\eta(k)(x(k)-w_j^*(k))\right\|}, \text{if } x(k) \text{ and } w_j^*(k) \text{ belong to the same cell,} \\ \dfrac{w_j^*(k)-\eta(k)(x(k)-w_j^*(k))}{\left\|w_j^*(k)-\eta(k)(x(k)-w_j^*(k))\right\|}, if\ x(k) \text{ and } w_j^*(k) \text{ belong to the different cells,} \\ w_j(k) \text{ for neurons, which are not won in instant } k. \end{cases} \tag{27}$$

First and third expressions in formula (27) are completely identical to WTA – self-learning algorithm, which the second one represents a “push-back” scenario.

Let’s consider a situation, shown on fig. 4, when neuron $w_j^*(k)$ won the competition. At the same time current sample $x(k)$ belongs to a class with different centroid $w_p(k)$. Now we need to «push» vector $w_j^*(k)$ away so, that $x(k)$ was equidistant from $w_j^*(k)$ and from $w_p(k)$.

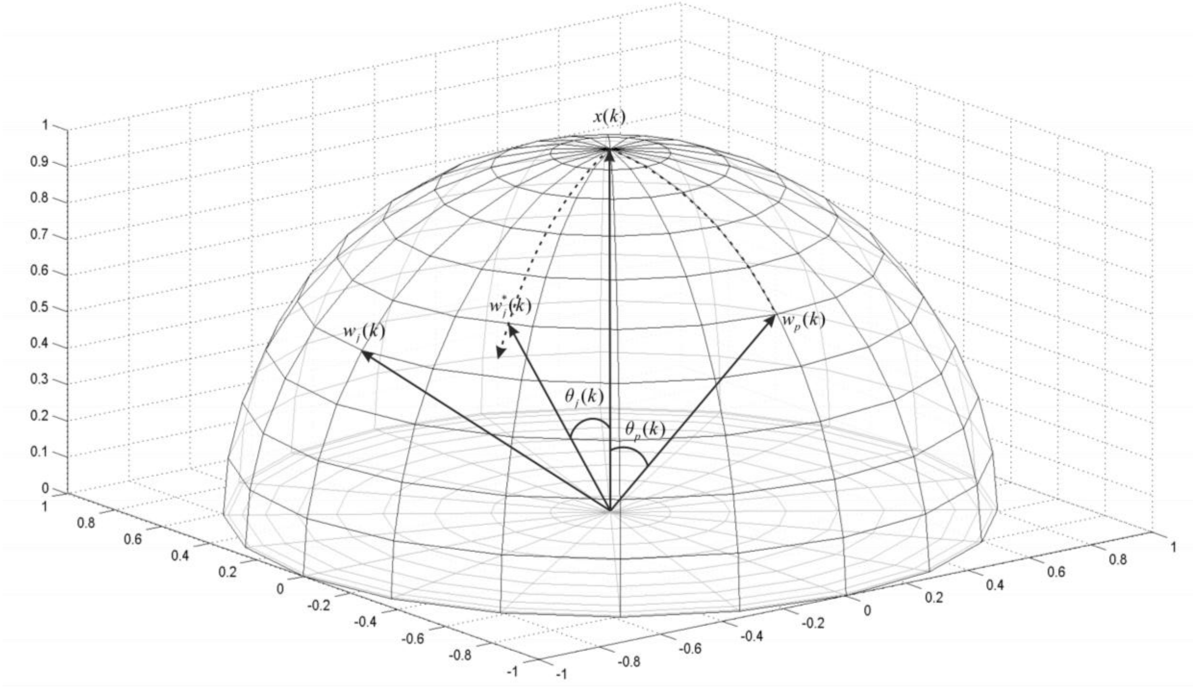

Fig. 4 – Learning of LVQ

Applying elementary transformations, we get the following formulas:

$$w_j(k+1) = w_j^*(k) - \eta(k)(x(k) - w_j^*(k)), \tag{28}$$

$$x^T(k)w_j(k+1) = x^T(k)w_j^*(k) - \eta(k)\|x(k)\|^2 - \eta(k)x^T(k)w_j^*(k), \tag{29}$$

$$\cos(w_j(k+1), x(k)) = \cos(w_j^*(k), x(k)) - \eta(k)(1 + \cos(w_j^*(k), x(k))) \tag{30}$$

In order to satisfy

$$\cos(w_j(k+1), x(k)) = \cos(w_p(k), x(k)) \tag{31}$$

We need to set $\eta(k)$ in form

$$\begin{aligned}\eta(k) &= \frac{\cos(w_j^*(k), x(k)) - \cos(w_p(k), x(k))}{\cos(w_j^*(k), x(k)) + 1} = \frac{\cos(w_j^*(k), x(k)) - \cos(w_p(k), x(k))}{\cos(w_j^*(k), x(k)) + \cos(x(k), x(k))} = \\ &= \frac{\cos\theta_j(k) - \cos\theta_p(k)}{\cos\theta_j(k) + 1} = \frac{x^T(k)w_j^*(k) - x^T(k)w_p(k)}{x^T(k)w_j^*(k) - x^T(k)x(k)}.\end{aligned} \tag{32}$$

After one step of the weights tuning the pattern $x(k)$ will belong equally to both $w_j(k+1)$ and $w_p(k) = w_p(k+1)$, i.e.

$$\mu_{w_j(k+1)}(x(k)) = \mu_{w_p(k+1)}(x(k)) = 0{,}5\,. \tag{33}$$

In case when several classes are overlapping, we can use estimate (25) for computing membership levels.

Joining SOM and LVQ learning algorithms

A promising application of Kohonen neural network is adaptive pattern recognition, which implemented by the systems. The implementation consists of sequentially connected SOM and LVQ [2] layers. First network operates in self-learning mode, while the second one adds supervised component to the process. From an input vector $x(k)$ which is fed to the input system, SOM network extracts a feature sample $y(k)$ in a space with sufficiently reduced dimensionality. This simplifies a problem in hand without significant loss of information. On the second stage LVQ is trained to classify of incoming pattern $y(k)$ using supervised learning. A distinctive feature of proposed system is the connection of two identical architectures, that are tuned by different learning algorithms.

For many data mining problems, especially in healthcare domain, substantial shares of input samples lack clear class association. Indeed, a diagnosis might be either missing, be ambiguous or yet unknown.

In this case it is possible to tune the synaptic weights with unified architecture with lateral connections using different learning methods. Each learning method is initialized according to the level of apriori information about $x(k)$ available.

A resulting learning algorithm for combined (SOM+LVQ) neural network can be presented in the following form:

$$w_j(k+1) = \begin{cases} \dfrac{w_j^*(k) + \eta(k)(x(k) - w_j^*(k))}{\left\| w_j^*(k) + \eta(k)(x(k) - w_j^*(k)) \right\|}, \\ \delta_L(k) \dfrac{w_j^*(k) - \eta(k)(x(k) - w_j^*(k))}{\left\| w_j^*(k) - \eta(k)(x(k) - w_j^*(k)) \right\|}, \\ \delta_L = \begin{cases} 1, \text{ if the network works in supervised mode} \\ \quad \text{and sample x(k) does not belong to class j,} \\ 0, \text{ otherwise,} \end{cases} \\ w_j(k) \text{ for neurons, which did not get activated by sample } x(k). \end{cases} \tag{34}$$

where $w_j^*(k)$ is winning neuron.

It is clear, that for $\delta_L = 0$, i.e. in self-learning mode, the first expression of formula (34) can be replaced by expression (24).

Conclusion

The combined self-learning procedure for Kohonen neural network is proposed. Such method allows data processing under the overlapping classes condition, when memberships of training data to specific classes can be unknown at all, and have both crisp and fuzzy nature. This method is based on using similarity measure, recurrent optimization and fuzzy inference and differs with high speed, possibility of operating in on-line mode and simplicity of computational realization.

References

1. Kohonen, T. Self-Organizing Maps. – Berlin: Springer-Verlag, 1995 – 362 p. 2. Haykin, S. Neural Networks: A Comprehensive Foundation. - Upper Saddle River, N.Y.: Prentice Hall, 1999. – 842 p. 3. Bezdek, J.C. Pattern Recognition with Fuzzy Objective Function Algorithms. - New York: Plenum Press, 1981. – 272 p. 4. Hoeppner, F., Klawon, T., Kruse, R. Fuzzy Clusteranalyse: Verfahren fuer die Belderkennung, Klassification und Datenanalyse. – Braunschweig: Vieweg, Reihe Computational Intelligence, 1996. – 280 S. 5. Bezdek, J.C., Keller, J., Krisnapuram, R., Pal, N. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Springer, 1999 – 777 p. 6. Hoeppner F., Klawonn F., Kruse R., Runkler T. Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition. - Chichester: John Wiley & Sons, 1999. – 300 p. 7. Sato-Ilic M., Jain L. Innovations in Fuzzy Clustering: Theory and Applications. – Berlin: Springer, 2006. – 152 p. 8. Vuorimaa, P. Use of the fuzzy self-organizing map in pattern recognition // Proc. 3-rd IEEE Int. Conf. Fuzzy Systems "FUZZ-IEEE'94". - Orlando, USA, 1994 – P.798-801. 9. Vuorimaa, P. Fuzzy self-organizing map // Fuzzy Sets and Systems. - 1994. - 66. - P. 223-231. 10. Tsao E.C.-K., Bezdek J.C., Pal N.R. Fuzzy Kohonen clustering networks // Pattern recognition. – 1994. – 27. – P. 757-764. 11. Pascual-Marqui R.D., Pascual-Montano A.D., Kochi K., Caraso J.M. Smoothly distributed fuzzy C-means: a new self-organizing map // Pattern Recognition. – 2001. – 34. – P. 2395-2402. 12. Sepkovski, J.J. Quantified coefficients of association and measurement of simularity // J. Int. Ass. Math. - 1974. – 6 (2). – P. 135-152. 13. Wasan M.T. Stochastic Approximation. – Cambridge: The University Press, 2004. – 216 p. 14. Dvoretzky, A., On stochastic approximation // Proc. 3-rd Berkley Symp. Math. Statistics and Probability. – 1956. – 1. – P. 39-55. 15. Goodwin, G.C., Ramadge, P.J., Caines, P.E. A globally convergent adaptive predictor // Automatica. – 1981 – 17 (1). – P.135-140. 16. Grossberg S. Classical and instrumental learning by neural networks // In "Progress in Theoretical Biology". – N.Y.: Academic Press, 1974. – 3 – P. 51-141. 17. Baras J.C., La Vigna A. Convergence of Kohonen's learning vector quantization // Proc. Int. Joint Conf. on Neural Networks. – San Diego, CA, 1990. – 3. – P.17-20.