Decreasing the Feature Space Dimension by Kohonen Self-Organizing Maps

Igor Mokriš, Radoslav Forgáč
Institute of Informatics Slovak Academy of Sciences, Dúbravská cesta 9, 846 07 Bratislava 45,Slovak Republik
e-mail: mokris@savbb.sk, forgac@savbb.sk

Abstract

The paper deals with Kohonen Self – Organizing Maps for generation of features and decreasing the dimension of classifying space in the image recognition process. In the conclusions there are introduced the results of comparison for feature generation effectivity by Kohonen Self - Organizing Maps and chosen methods for reduction of classifying feature space dimension.

Key words: Kohonen self – organizing maps, reduction of feature space dimension

1 Introduction

Kohonen Self – Organizing Maps [3, 10, 13, 14, 15] are suitable for realization of mapping of topology preservation and express this way the characteristic features for classifying of input images. Based on this typical property the neurons in Kohonen Self – Organizing Maps are ordered in two – or one – dimensional space. This mapping that preserve the topology of neural networks in the learning process has one important property [14], i.e. for similar patterns in the input image space are responding the neurons in Kohonen Self – Organizing Map which are in the output layer physically near each other. All neurons are shared into representation of features in Kohonen Self – Organizing Maps and feature value is defined by winner neuron position, i.e. the extracted features are expressed topologically. The extracted features may be without physical interpretation.

The structure of Kohonen Self – Organizing Map incoming from Willshaw and von der Malsburg model [19], in which the neurons of output self – organizing layer mutually affect themselves by means of the laterally feedback connections which are in time invariable. On account of that substitution of laterally interactions by neighbor function in the output layer are Kohonen learning algorithms more simple in relation to the different learning algorithms of self – organizing neural networks because the laterally feedback connection is time consuming from point of computation [14].

Kohonen Self – Organizing Map are suitable not only for reduction of the feature space of images and patterns in the feature generation process, but also for the classification of input images and patterns into corresponding classes. For classification purpose the quantization learning vector algorithms LVQ can be utilized, which is based on the statistical approaches of classification.

2 The structure of Kohonen Self – Organizing Map

Kohonen Self – Organizing Map belongs into group of feed forward self – organizing full connected neural networks. It is two layer neural network (Fig. 1), which consists of input and output layer, i.e. Kohonen Self – Organizing Map. Input layer copies the vector of input features in *D*-dimensional space. From point of visualization of generated features Kohonen layer can be two- or one - dimensional.

As above mentioned, important contribution of Kohonen Self – Organizing Maps is substitution of laterally interactions between neurons in output layer by neighbor function $h(i^*, i)$, where i^* is index of winner neuron and i is index of neurons in neighbourhood of the winner neuron.

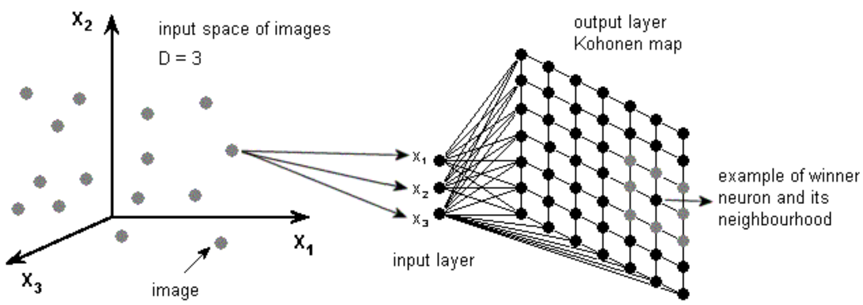

Fig.1 Structure of Kohonen Self – Organizing Map

Each neuron has defined itself neighbourhood consists of the surrounding neurons. The neighbor function defines the region of cooperation between neighbor neurons. I.e., that in the process of learning were adapted only weight vectors of winner neuron and neurons from its neighbourhood. The adaptation of weight vectors is influenced by type of utilized neighbor function $h(i^*, i)$. Simplest neighbor function $h(i^*, i)$ is „bubble“ function [5, 8], which is defined by relation

$$h(i^*, i) = \begin{cases} \alpha(t), \text{ if } r_M(i^*, i) \le \sigma(t) \\ \\ 0, \quad \text{in other cases} \end{cases} \tag{1}$$

where $\alpha(t) \in (0, 1)$ is the learning coefficient, $\sigma(t)$ is the radius of defined neighbourhood, $r_M(i^*, i)$ is distance between neurons i^* and i of „Manhattan" type. Function $\alpha(t)$ decreases to 0 in time, i.e., the neural network is not able to learn.

In practice it is problem to set up the optimal initialized value of learning coefficient α, because for small values α it is true that the neural network is quickly learning, but account of new stimulus in input layer also quickly forget. Kohonen suggested, that best results are achieved by monotonous decreasing of function $\sigma(t)$ [5].

Second very frequently used neighbor function is function of Gaussian type, which is defined by

$$h(i^*, i) = \alpha(t).\exp\left(-\frac{r_E^2(i^*, i)}{2\sigma^2(t)}\right) \tag{2}$$

where r_E is Euclidean distance between neurons i^* and i in Kohonen layer, for which the holds true

$$r_E(i^*, i) = \left|\mathbf{r}_{\mathbf{i}^*} - \mathbf{r}_{\mathbf{i}}\right| = \left|\mathbf{w}_{\mathbf{i}^*} - \mathbf{w}_{\mathbf{i}}\right| \tag{3}$$

where $\mathbf{r_i}$ is a coordinate vector of i - neuron and $\mathbf{w_i}$ is weight vector directed into i-neuron.

Main disadvantage of Kohonen Self – Organizing Maps is necessity to defining the structure of neural networks and number of neurons in Kohonen layer, a priori. Possible existence of so-called died neurons can be eliminated by substitution of concurrency learning „winner takes all" type by „winner takes most" type.

3 Learning of Kohonen Self – Organizing Map

In this chapter will presented two algorithms for learning of Kohonen Self – Organizing Maps, i.e. Kohonen algorithms of concurrency learning and algorithms LVQ.

Kohonen algorithms of concurrency learning is based on assumption, that space of input feature vector $\mathbf{x}$ is identical to the space of weight vectors $\mathbf{w_i}$, tend to corresponding winner neurons. In case of utilization of normalized input feature vector **x** and weight vectors $\mathbf{w_i}$ the relation holds true

$$|\mathbf{x}| = |\mathbf{w_i}| = 1 \qquad (4)$$

i.e. input feature vector **x** actives neurons in Kohonen layer by full connection of neurons in individual layers. The activity of output neuron is defined by relation

$$y_i(t) = \sum_{j}^{D} w_{ij}(t).x_j(t) \qquad (5)$$

where D is number of neurons in output layer.

Self – organizing neural networks use competitive principle and because of it is necessary to determine the winner neuron, which is the most sensitive on input stimulus **x**. There exist two methods for learning of winner neuron:

- Method of finding the most activity of output neuron (so-called Dot Product Method) [9], where finding the winner neuron i^* with maximal activity is realized by rule

$$i^* = \arg\max |\mathbf{w}_i.\mathbf{x}| \qquad (6)$$

- Method of finding the minimal distance between vectors **x** and $\mathbf{w_i}$ [10]. As the winner neuron is considered this neuron, which of the weight vector is by Euclidean metric nearest to actual input vector **x** by rule

$$i^* = \arg\min |\mathbf{x} - \mathbf{w}_i| \qquad (7)$$

Both methods are mutually equivalent each other because holds

$$|\mathbf{x} - \mathbf{w_i}|^2 = |\mathbf{x}|^2 - 2\mathbf{w_i^T}.\mathbf{x} + |\mathbf{w_i}|^2 \qquad (8)$$

From equation (8) due to utilized normalization (4) influences, that finding of minimal distance (7) is analogy to finding of maximal scalar product of vectors **x** by rule (6) [14]. Kohonen algorithm based on Euclidean metric don't require normalization and because of is its utilization more effective than the algorithm of finding the most activity of output neuron [16]. Adaptation process of weights is based on minimization of error function, which is defined by

$$J(t) = \frac{1}{2}\sum_{j=1}^{D}\left(w_{ij}(t) - x_j(t)\right)^2 \qquad (9)$$

The goal of competitive learning is projection of input vector **x** into weight vectors $\mathbf{w_i}$ for winner neuron and its neighbourhood. Adaptation of weights in learning process is defined by relation for standard gradient method

$$\Delta w_{ij}(t) = -\alpha \frac{\partial J(t)}{\partial w_{ij}(t)} \tag{10}$$

where α is learning coefficient. Then the gradient of error function can be determined by relation

$$\frac{\partial J(t)}{\partial w_{ij}(t)} = \begin{cases} w_{ij}(t) - x_j(t), & \text{if it is the winner neuron} \\ 0, & \text{in other cases} \end{cases} \tag{11}$$

By substitution of relation (11) into (10) can be obtained relation for computation of weight vector for winner neuron in time *t+1*

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t)\left(x_j(t) - w_{ij}(t)\right) \tag{12}$$

Kohonen algorithm of competitive learning „winner takes most" type is better as algorithm of competitive learning „winner takes all" type. In the algorithm „winner takes most" there are adapted also weight vectors of neurons from neighborhood of winner neuron, which is defined by neighbor function $h(i^*, i)$ by relations (1) and (2). Therefore, relation (12) is modified as

$$w_{ij}(t+1) = w_{ij}(t) + h(i^*, i)\left(x_j(t) - w_{ij}(t)\right) \tag{13}$$

More detail description of Kohonen algorithm „winner takes most" type is described as follows:

Number of iterations is variable in interval approx. 10^4-10^5. Kohonen

procedure Kohonen
  Define the Kohonen Self – Organizing Map topology.
  Initialize of all weight vectors by random generator in interval <0, 1>.
  t = 0
  Set parameter α(t) and σ(t) for t = 0.
  repeat
    for k=1 **to** N
      Get input vector $\mathbf{x_k}$ from training set N.
      Find winner i^* by rule i^* = arg min|**x**-$\mathbf{w_i}$|.
      Adapt weights of winner neuron and its topological neighbors by (13).
      Increment t.
    end;
    Decrement parameter α(t) and σ(t).
  until (algorithm converge) **OR (**number iteration is overflow**)**
end;

recommended empirically verified number of iterations, which can be minimally 500 times more than number of neurons in network.

Advantage of Kohonen algorithm „winner takes most“ type in comparison with algorithm „winner takes all“ type is elimination of undesirable effect of existence so – called died neurons. There are neurons, which were eliminated in learning process and because of their weight vectors aren’t adapted.

Algorithm LVQ [4, 6, 7, 11, 12] enables to set the weight vectors of Kohonen network from point of minimization of number of bad classifications, which arises from reason of overlaying the probability density functions for each classified class. The classification error is minimized in case, when the boundary between classified classes is identical to Bayesian boundary [14]. The role of algorithm LVQ is approximation of Bayesian boundary between classified classes without knowledge of statistical description of input vectors **x**.

There are three versions of algorithm LVQ, i.e. LVQ1, LVQ2 and LVQ3. Because the principle in all cases is similar, in next part will described more detail only algorithm LVQ1 [12].

Adaptation of weight vectors, unlike of Kohonen competitive learning is realized by supervised learning. For each class of input images in Kohonen layer is allocated a label, which represents membership of input vector to specified class. From point of difference of competitive learning is not important, which neuron belonging of specified label is winner, but more important is fact, which of neuron represents correct class.

Before realization of LVQ1 algorithm is necessary to initialize the weight vectors and allocate labels. Weight vectors are initialized by Kohonen competitive learning. Labels can be determined by “most” rule, i.e. evaluate number of winnings for each neuron and its corresponding weight vector. Adaptation of weight vectors is performed by relations

$$\mathbf{w}_{\mathbf{i}^*}(t+1) = \mathbf{w}_{\mathbf{i}^*}(t) + \alpha(t)\big(\mathbf{x}(t) - \mathbf{w}_{\mathbf{i}^*}(t)\big) \quad \text{ak } C(\mathbf{x}) = C(\mathbf{w}_{\mathbf{i}^*}) \tag{14}$$

$$\mathbf{w}_{\mathbf{i}^*}(t+1) = \mathbf{w}_{\mathbf{i}^*}(t) - \alpha(t)\big(\mathbf{x}(t) - \mathbf{w}_{\mathbf{i}^*}(t)\big) \quad \text{ak } C(\mathbf{x}) \neq C(\mathbf{w}_{\mathbf{i}^*}) \tag{15}$$

where *C(***x***)* specifies class, whom belong an image represented by input vector **x**. Algorithms LVQ1 is more detail described in next part:

procedure LVQ1
    Initialize all weight vectors by Kohonen competitive learning.
    Initialize parameter α(t) and t=0.
    repeat
        for k=1 **to** N
            Get input vector $\mathbf{x_k}$ from training set N.
            Find winner neuron i^*.
            if class_label = desired_class_label **then**
                Adapt weights of winner neuron by (14).
            else
                    Adapt weights of winner neuron by (15).

Increment t.
end;
Decrement $\alpha(t)$.
until (algorithm converge) **OR (**number iteration is overflow**)**
end;

By optimization of LVQ1 was developed algorithm LVQ2 [7] and LVQ3 [12]. Algorithm LVQ2 unlike of origin algorithm LVQ1 adapts not only the weight vectors of winner neuron, but also of second nearest neurons of input vector **x**. Algorithm LVQ3 removes disadvantages of LVQ2, mainly deteriorate results during longer learning, i.e. approx. 10^5 iterations. Choice of suitable algorithm LVQ depends on applications. From point of stability learning is recommended utilization of algorithm LVQ1 or LVQ3 [6].

4 Comparison of Kohonen Self - Organizing Maps with chosen methods for reduction of feature space dimension

In this chapter are presented results of comparison of Kohonen Self – Organizing Map (in Fig. 3 and 4 signed by SOM – Self Organizing Map) with chosen methods for reduction of feature space dimension [1]:

1. linear method of principal component analysis (PCA),
2. nonlinear auto-associative neural network based on principal component analysis method (AFN),
3. Sammon neural network based on multidimensional scaling method (SAM),
4. stochastic method of multidimensional scaling (MDS).

All introduced methods except for linear method of principal component analysis are based on the principle of minimizing the error function.

Comparison of chosen methods is performed by two groups of textured images. Both groups of images were pre–processing by wavelet transform on account of remove variance under rotation of images [17, 18]. First group of images is created by 245 gray images, which are divided into 5 classes and input feature space has dimension $D=39$. Second group of images is created by 1024 color images divided into 16 classes and input feature space has dimension $D=48$.

Initialized configuration of weight coefficients for Kohonen and auto-associative neural network have been setting randomly. The structure of AFN was defined as:

- D-30-d-30-D for first training set of images,
- D-40-d-40-D for second training set of images,

where D is number of neurons in input and output layer and represents dimension of feature space of input images, d is number of neurons in hidden layer and represents number of generated features for $d < D$. Initialized configuration for both methods based on multidimensional scaling methods was set up by the results of linear method of principal components analysis. Number of iterations was determined as 100.

Quality of generated features for all methods of feature dimension reduction was evaluated by classifier kNN (k=5, k nearest neighbors). Effectivity of compared methods was evaluated by classification correctness in % in dependency on number of features in feature vectors, which were entered into classifier. The results were analyzed from point of influence for dimension of input feature space in relation to the classification correctness and number of classified classes. Results of experiments for first group of images are represented by graph in Fig. 2 and for second group of images are represented by graph in Fig.3.

Learning algorithms for AFN converges very slowly, number of iteration is approx. 10^3*N (where N is number of images of training set). Algorithms SOM needs for convergence approx. 10 times less time as in the case of AFN, i.e. approx. 10^2*N iterations. In this case is more advantage the utilization of statistical multidimensional scaling method, or Sammon neural network based on multidimensional scaling method. Linear method of principal component analysis got worse results in comparison with nonlinear methods of projection under reduction for one or two features. By increasing of number of generated features the classification correctness of linear method of principal component analysis is comparative to nonlinear methods. On the other hand advantage of linear method of principal component analysis is implementation by neural networks with simple structure and shorter learning time.

From experiments resulted, that Kohonen neural networks due to nonlinear projection are able to more precisely describe classified images in the case of small number of generated features for $d \leq 2$. On the other hand from point of time consuming of computation in the learning process (Tab. 1) Kohonen neural networks are not suitable for reduction of input feature space for large dimension.

From point of number of classified classes can be mentioned, that Kohonen and auto-associative neural networks are suitable for reduction of input feature space with lower dimension and higher number of classified classes.

	1. group of images				2. group of images			
d	SOM	AFN	SAM	MDS	SOM	AFN	SAM	MDS
1	13	76	0,1	0,5	67	611	1,3	7,8
2	10	100	0,5	0,7	33	790	7,5	11,0
3	9	120	0,7	0,9	31	840	15,0	13,0
4	6	130	1,0	1,1	18	893	27,0	21,0

Tab.1 Time consuming of learning process for nonlinear PCA by [1]

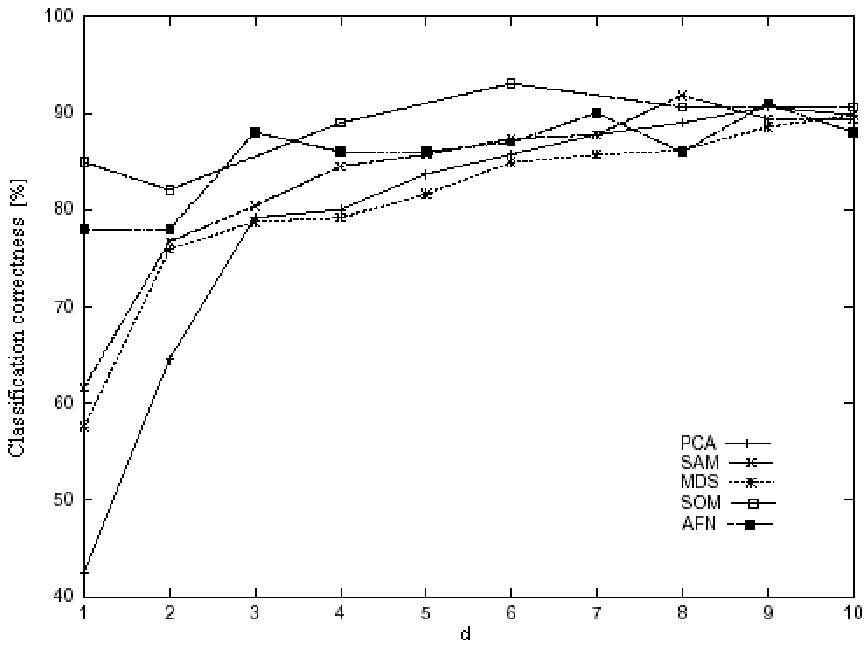

Fig.2 Classification correctness for first group of images

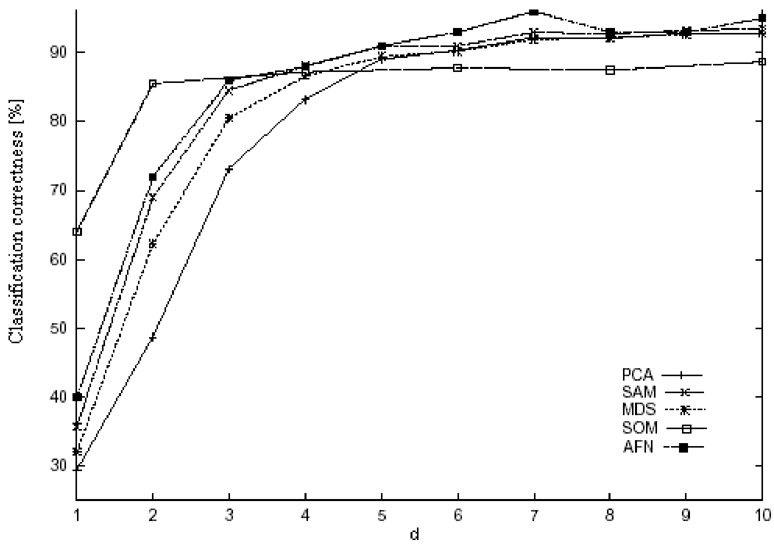

Fig.3 Classification correctness for second group of images

5 Conclusion

Significant property of Kohonen Self-Organizing Maps is data projection ability with topology preservation. After successful learning arbitrary two images, which are near each other in input space, they activate in output Kohonen layer topologically near neurons, too. With the assistance of projection of input images seems to be dramatic dimension reduction of output space. Therefore Kohonen Self-Organizing Maps in dependence on used learning algorithm are suitable for dimension reduction of features and also for classification tasks, too.

Kohonen Self-Organizing Maps with unsupervised learning were applied in dimension reduction of features. They are suitable for reduction of input space with lower dimension and higher number of classification classes. In the case of reduction input space of features with higher dimension is suitable to combine Kohonen Self-Organizing Maps with neural networks, which are good for feature generation [2].

Kohonen neural networks based on Learning Vector Quantization belong to neural networks with supervised learning. These neural networks are suitable for classification tasks.

References

[1] DE BACKER, S.: Unsupervised Pattern Recognition - Dimensionality Reduction and Classification. [PhD Thesis]. University of Antwerp, 2002.

[2] FORGÁČ, R. – MOKRIŠ, I.: Artificial Neural Networks for Reduction of Dimension of Feature Space and Classification. ISBN 80-8055-743-8, Matej Bel University Banská Bystrica, 2002, (in Slovak).

[3] HRISTEV, R. M.: The ANN Book. GNU Public License, 1998.

[4] KANGAS, J. A. - KOHONEN, T. K. - LAAKSONEN, J. T.: Variants of Self-Organizing Maps. IEEE Transactions on Neural Networks, Vol. 1, 1990, pp. 93-99.

[5] KOHONEN, T. - HYNNINEN, J. - KANGAS, J. - LAAKSONEN, J.: SOM PAK: The Self-Organizing Map Program Package. Helsinki University of Technology, 1996.

[6] KOHONEN, T. et al: LVQ_PAK: The Learning Vector Quantization Program Package. [Report A30]. Helsinki University of Technology, ISBN 951-22-2948-X, 1996.

[7] KOHONEN, T.: Improved Versions of Learning Vector Quantization. Proc. of the International Joint Conference on Neural Networks, Vol. 1, San Diego, 1990, pp. 545-550.

[8] KOHONEN, T.: Self-Organization and Associative Memory. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 3. Edition, 1989.

[9] KOHONEN, T.: Self-Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics, Vol. 43, No. 1, 1982, pp. 56-69.

[10] KOHONEN, T.: Self-Organizing Maps. Springer-Verlag, ISBN 3-540-

58600-8, 1995.
[11] KOHONEN, T.: Statistical Pattern Recognition Revisited. Advanced Neural Computers, 1990, pp. 137-144.
[12] KOHONEN, T.: The Self-Organizing Map. Proc. of the IEEE, Vol. 78, No. 9, 1990, pp. 1464-1480.
[13] KROSE, B. - VAN DER SMAGT, P.: An Introduction to Neural Networks. University of Amsterdam, 1996.
[14] KVASNIČKA, V. et al: An Introduction to the Neural Networks Theory. IRIS, ISBN 80-88778-30-1, 1997, (in Slovak).
[15] LIPPMAN, R. P.: An Introduction to Computing with Neural Nets. IEEE on ASSP Magazine, 1987, pp. 4-22.
[16] SINČÁK, P. - ANDREJKOVÁ, G.: Neural Networks - Part 1, ELFA, ISBN 80-88786-38-X, Košice, 1996, (in Slovak).
[17] VAN DE WOUWER, G. et al: Wavelet Correlation Signatures for Color Texture Classification. Pattern Recognition, Vol. 32, No. 3, 1999, pp. 443-451.
[18] VAUTROT, P. et al: Rotation-Invariant Texture Segmentation using Continuous Wavelets. Proc. of 2nd IEEE Symposium on Applications of Time-Frequency and Time-Scale Methods, Coventry, 1997.
[19] WILLSHAW, D.J. - VON DER MALSBURG, C.: How Patterned Neural Connections can be Set up by Self-Organization. Proc. of the Royal Society of London B, Vol. 194, 1976, pp. 431-445.