Two More Modifications of SOMs to Handle Signals with Special Properties

Stefan Schünemann, Udo Seiffert, Bernd Michaelis

Otto-von-Guericke-University of Magdeburg

Institute for Measurement Technology and Electronics

P.O. Box 4120, D-39016 Magdeburg, Germany

stefan.schuenemann@zenit.uni-magdeburg.de

Abstract

This paper presents two variations of the Self-Organizing Map in order to adapt it for special input signal attributes. The first one improves the properties concerning outliers and clusters with low feature density within the provided training data set by changing the neighbourhood function while training the net. In the second introduced modification the originally two-dimensional feature map has been extended to an ordered two-dimensional structure of two-dimensional feature maps. The Quasi-Four-Dimensional Neuroncube (QFDN) is introduced. To each variation a short description of a prospective application has been attached.

1 Introduction

Artificial neural networks provide an efficient method for cluster analysis and classification. The theory of Self-Organizing Maps (SOM) facilitates a reduction and cluster analysis of high-dimensional feature spaces onto two-dimensional arrays of representative weight vectors [3]. The unsupervised learning, the preservation of the topographies and the smoothing of the feature vectors [5] are further properties of these artificial neural networks. The introduced modifications are based on the original SOM algorithm presented by Kohonen in [3]. A summary of some new aspects also concerning the contents of this paper can be found in [4].

The smoothing property of the SOM leads to a relative insensitivity to outliers and clusters of low feature density. The problem is that the weight vectors of the representing neurons cannot be considered as reference vectors for low feature density clusters. That's why the approximation error increases. In the case that these clusters represent significant qualities of the investigated distribution, the smoothing property is highly undesirable. The first modification encourages a training of the neuron weights into the direction of clusters with low feature density and makes a projection onto separate neurons on the SOM possible.

The second described modification is an extension of the ancient two-dimensional feature map to a quasi four-dimensional cube of neurons. This supplies a local independent classification of shifted patterns. As a special feature this map can be connected with a two-dimensional input layer to present matrix-shaped input data and is able to grow. In most cases this can speed up the training and has advantages building the classes [10].

2 Analysing Clusters with Differing Feature Density Using 2D-SOM

During classification clusters of different and in some cases low feature density may be encountered. Due to the averaging property of the SOM, these will not necessarily project onto separate clusters on the SOM. The weight vectors of the representing neurons cannot be considered as reference vectors for

clusters of low feature density. This can be attributed to the strongly dissimilar probabilities during random selection of a feature vector from the individual clusters of differing feature density [5].
But, in some classification tasks in science and medicine clusters of low feature density represent significant qualities of the feature distribution. The smoothing property is highly undesirable.
The main idea of the first modification is to improve the learning ability of individual neurons on the SOM in direction of regions with low feature density. This can be achieved with a greater weighting of regions with low feature density in contrast to high density clusters during the training phase. For the realisation a definition of partial spaces (low and high feature density) in the feature space is used. This idea leads to a modified neighbourhood function in Eq.(1), which in contrast to the basic algorithm more strongly differentiates the degree of coupling of the neighbouring neurons with the winner neuron.

$$\hat{h}_{ci}(t) = \begin{cases} \alpha(t) & , \quad \| \boldsymbol{x} - \boldsymbol{m}_c \| = \min\limits_i \{ \| \boldsymbol{x} - \boldsymbol{m}_i \| \} \\ \dfrac{\alpha(t) \cdot \beta(t)}{2 \cdot D_{ci}(t)} & , \quad r(t) \geq D_{ci}(t) \\ 0 & , \quad r(t) < D_{ci}(t) \end{cases} \tag{1}$$

The neighbourhood distance $D_{ci}(t)$ is defined as the vector distance between the winner and neighbourhood neurons and the radius $r(t)$ characterises the effective training region on the SOM.
If the neuron i is the winner neuron c then the neighbourhood function is equal to the learning-rate factor $\alpha(t)$. The neighbourhood function for the neurons with the radius $r(t)$ greater than the neighbourhood distance $D_{ci}(t)$ is local-dependent from its position i on the SOM to the position of the winner neuron c. So the neighbourhood neurons have the relation $\alpha(t)\cdot\{2\cdot D_{ci}(t)\}^{-1}$. In this case they are also local-dependent from the position in the feature space through the term $\beta(t)$. This term controls the update in the direction of clusters of low feature density in dependence on partial spaces. An idea concerning the initialisation of the SOM will be used for the description of the partial spaces of the N-dimensional feature space which utilises the principal components analysis and defines a hyper-ellipsoid γ with the eigenvalues λ_k of the feature distribution as its half-axes:

$$\gamma = \frac{x_1'^2}{\lambda_1} + \frac{x_2'^2}{\lambda_2} + \cdots + \frac{x_N'^2}{\lambda_N} \tag{2}$$

where $\boldsymbol{x}'$ are the feature vectors transformed using the eigenvectors. Inside the ellipsoid $\gamma \leq 1$ the modified neighbourhood function is only dependent from the position on the SOM. The partial space with low feature density is located outside the hyper-ellipsoid $\gamma > 1$, where the term $\beta(t)$ should lead to the convergence of the neuron weights to the clusters with low feature density (Figure 1). In the orientation and convergence stages of the training phase the empirically found term $\beta(t)$ is outside the ellipsoid a decreasing function, which converges to zero with increasing distance from the ellipsoid borderline.
The complete representation of the modified training algorithm including the term $\beta(t)$ and its effects in the different training stages is described in [6], [7].

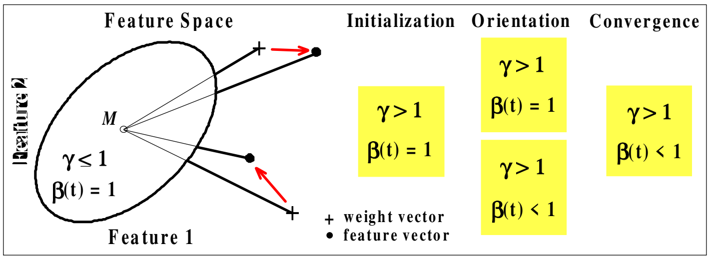

Figure 1: Definition of the term $\beta(t)$.

Figure 2 shows a result of the training in a three-dimensional feature space for the classification of an example distribution, side-by-side for both the basic method and the modified algorithm. The spots correspond to the feature vectors and the crosses to the weight vectors.
From the figure it is evident that using the basic method, the weight vectors have been pulled into the direction of the ellipsoid, i.e. into the main cluster I. Cluster III is represented by a separate neuron (upper gray box in Figure 2 left side). Because of the high Euclidean distances between the neuron weight and the feature vectors of this cluster the weight vector of the neuron is not a reference vector. Cluster II is projected onto one neuron together with some feature vectors of the main cluster I (lower grey box in Figure 2 left side). Consequently, cluster II is suppressed.
In contrast to the basic method the modified algorithm projects the clusters II and III onto separate neurons and the weight vectors represent reference vectors of these clusters (Figure 2 on the right side).

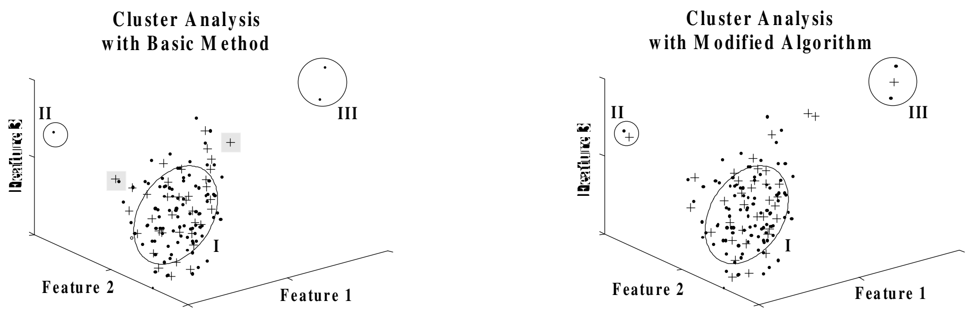

Figure 2: Feature (·) and weight (+) vectors for the cluster analysis of an example distribution.

After the recall phase the averaged approximation error $\overline{E}_i$:

$$\overline{E}_i = \frac{1}{q \cdot N} \sum_{j=1}^{q} \left\| \boldsymbol{x}_j - \boldsymbol{m}_i \right\| \quad , \quad \overline{E}_i \in [0,1] \qquad (3)$$

of each neuron referring to the learning set makes an objective judgement about the projection of the N-dimensional feature space onto the array of neuron weights. Using the standardisation term $1/N$ a comparison independent of the dimension of the feature space is possible. The term q indicates the number of states of excitations for each neuron and $\boldsymbol{x}_j$ with $j = 1 \ldots q$ describes the feature vectors projected onto it. For the visualisation of the calculated error function in Figure 3 a logarithmic form of expression is chosen.

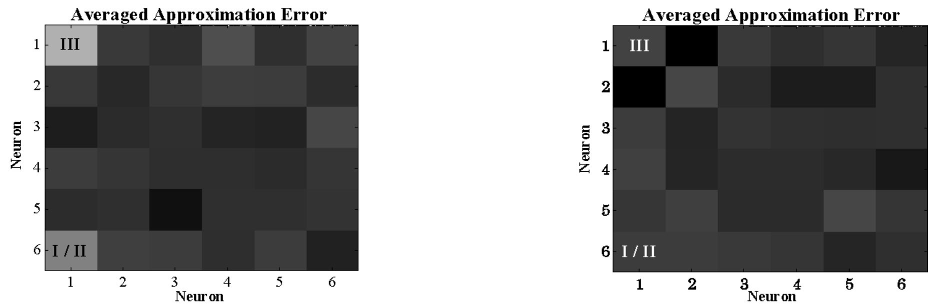

Figure 3: Approximation error for the basic method (left) and the modified algorithm (right).

If the training algorithm converges during training, this should yield a smooth regular function for the grey values in Figure 3. But, the error function of the basic algorithm shows clearly visible grey-level differences for the neurons n_{11} and n_{16}. The increased approximation error for these neurons results from the discussed difficulty with the Euclidean distances between the feature vectors from cluster II and III to the weights of their reference vectors on the SOM. The error function of the modified algorithm shows a relatively regular course. The neurons n_{12} and n_{21} are never activated during the recall phase. These neurons separate the neuron n_{11}, which represents the cluster III. Another investigation of the neuron weights, for example calculations of the Euclidean distances, for the detection of the cluster borders for the cluster III is not necessary.

3 Quasi-Four-Dimensional Neuroncube (QFDN-Network)

An important problem using SOMs is the local dependence of the representation of the input data. Often it is the task of a SOM to sort out items into appropriate categories or classes of similar objects. If two input vectors contain the same pattern, but at different positions, they are treated as two different classes (see Figure 4).

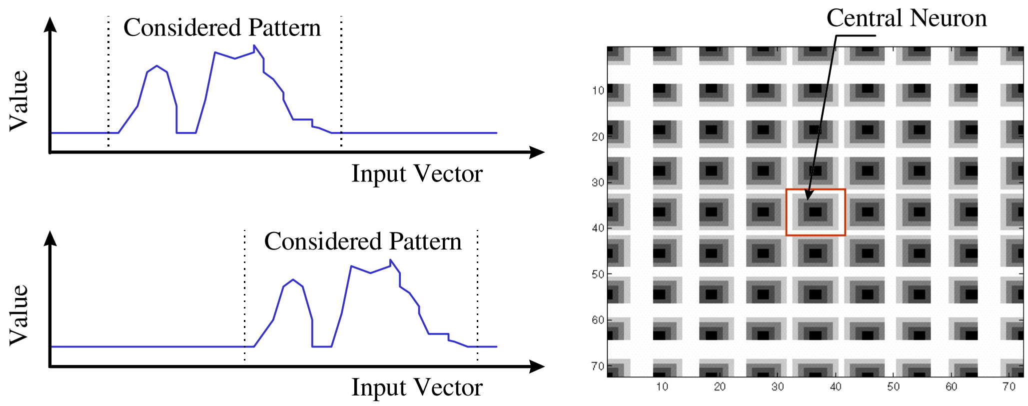

Figure 4: Pattern at different positions within the input layer in one-dimensional case (left). Weights of a complete submap (9x9 neurons) with shifted basic pattern (right).

This can be avoided by a suitable pre-processing procedure (transformation, normalisation). However, this may not be possible in every case. Additionally, if a complete neural solution, because of its training abilities, is preferred, it is helpful to implement a local independent classification system based on the SOM algorithm. If the input data is matrix shaped instead of the classically vector shaped inputs, the local dependence is two-dimensional, too. Matrix shaped data can be found for instance in image processing.

In continuation of the suggested three-dimensional feature map [9] and its growing variant [10] the Quasi-Four-Dimensional Neuroncube (QFDN-Network) is presented. The net consists of a two-dimensional structure of again two-dimensional submaps. Thus it is two-plus-two-dimensional or quasi-four-dimensional, because it was derived from the three-dimensional feature map [9].

The network has optionally a two-dimensional input layer to present matrix shaped input data $\mathbf{V}$ and in this case also two-dimensional weight matrices $\mathbf{W}$. If the two-dimensional input matrix is reshaped into a vector by a fixed order, the data can not be observed by humans anymore. The so formed vectors are not comparable by a simple look at it. This affects a comparison between the weights themselves and of course between the weights and the originally matrix shaped input data.

Each submap contains one basic pattern. The central neuron represents this pattern in central position. All other neurons represent this pattern in a shifted position corresponding to their own relative position within the submap (Figure 4, right). All submaps are building a two-dimensional structure, similar to the original feature map and perform a classification. The weights of all other neurons within this submap are derived from the kernel weights by shifting these according to the relative position (n,m) of the corresponding neuron. For the single weight element w_{kl} can be written:

$$w_{kl}^{(i-n,j-m)} = w_{k+n,l+m}^{(i,j)} \tag{4}$$

Thus all neurons of one submap contain the same pattern but at different positions. When the net reads a

basic pattern at any position, the submap is indicated by the winning neuron. Thus the net acts like a two-dimensional map with the advantage of a local independent response to moved input data. The derivation of the changed learning algorithm for a three-dimensional map can be found in [9].
The general learning rules are not affected. The training runs in two major phases.

3.1 Growing of the Net

In the first phase the net grows similar to [2] from an initially small size. In contrast to [2] the single neurons are submaps consisting of predefined neurons. The weight matrix of the central neuron is randomly initialised and is building the kernel weights.
An error value Δe derived from the computation of the Euclidean distance D at the current (t_c) and the previous time step (t_{c-1})

$$\Delta e = \sum_{\forall i} D(\mathbf{V}, \mathbf{W}_i)\Big|_{t=t_{c-1}} - \sum_{\forall i} D(\mathbf{V}, \mathbf{W}_i)\Big|_{t=t_c} \tag{5}$$

is used to decide when a row or column of new neurons has to be inserted. The Euclidean distance between the most frequently winning neuron and those neurons at the same relative position in all neighbouring submaps (outer neighbours) is used to decide where the new row or column of neurons has to be inserted. This is the area of the highest data density. New neurons supplying more weights (storing space for more information) are needed just there. Figure 5 demonstrates the insertion process.

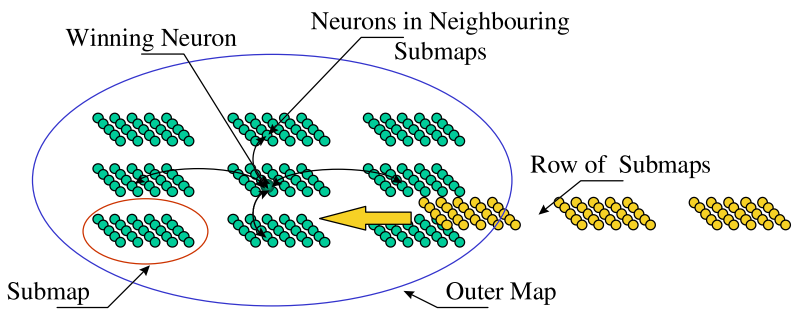

Figure 5: Quasi-Four-Dimensional Neuroncube with 3x3 submaps, 5x5 neurons each.

The weights of the new neurons are interpolated from the two outer neighbours not just being inserted. Thus the area of the highest local data density is stretched out. This procedure may be less exactly than other possible strategies, but it has a sufficient accuracy and makes a very fast implementation possible.
This growing process stops if one of the stopping criteria is met. The primary criterion is

$$\frac{1}{\mu} \sum_{\mu} D(\mathbf{V}_{\mu}, \mathbf{W}^{(i_c, j_c)}) \leq E_{stop} \tag{6}$$

where μ is the number of training samples $\mathbf{V}$ and $\mathbf{W}^{(i_c, j_c)}$ is the weight matrix of the winning neuron for the current input μ . E_{stop} is a user-defined threshold. However, if this is never met, a maximal number of neurons (submaps) acts as an emergency stop and prevents the net from growing indefinitely.

3.2 Adaptive Fine-Tuning

If the desired network size is achieved, a fine-tuning process will be performed. The final weights will be completed by an adaptive learning phase, with decreasing learning rate and different neighbourhood functions. Special attention has to be paid to these neighbourhood functions. Two independent areas are defined. Each of them is two-dimensional.
The first area is spread out around the winning neuron inside the corresponding submap. This neighbourhood is responsible for the ordering of the moved basic patterns inside the submap. Because the data set contains possibly not all patterns at all positions a 'movement' of some neurons within their submap will happen. In the long run the match between the weights and the actual input data is much

better. The second neighbourhood area is defined in the outer map. It contains the outer neighbours and serves the real classification.
If this process has been finished, the QFDN net has learnt several basic patterns. Each pattern is represented in one submap at moved positions. The coordinate of the submap containing the winning neuron indicates the class of the current input, independent of the real position of the basic pattern within the input matrix.

4 Applications of the Modified SOMs

The performance of the described modified SOM for the classification of clusters with different and in same cases low feature density is to be tested with an investigation of the self-organization during the growth of crystals of semiconductor nanostructures [1] and the detection of highly complex combination patterns of cell surfaces under normal and pathological conditions in the human immune system [8]. Both projects are characterized by clusters with low feature density which yield important consequences in physics or medicine.
The QFDN has been utilised as a tool in a motion detection system to classify image blocks and two-dimensional correlation functions between image blocks [10]. These data consists of basic patterns (smaller than the input matrix), two-dimensionally moved to several positions within the input matrices. These basic patterns have to be classified independent of their actual position without any pre-processing. Because the complexity and thus the number of classes is not known a-priori, also the growing feature of the net is very advantageous to get an optimal topology and distribution of the classes.

This work was supported by the grants BMBF/DFG INK 15/A1 and LSA 1441A/0083.

References

[1] S. Dressel, et al. Structural Analysis in the Atomic Force Microscopy of Self-Organized Semiconductor Nanostructures. *Proc. DPG*, Reihe VI(31), Regensburg 1996.

[2] B. Fritzke. Growing Grid - A Self-Organizing Network with Constant Neighbourhood Range and Adaption Strength. *Neural Processing Letters*, Vol. 1, no. 5, pp. 1-5, 1995.

[3] T. Kohonen. *Self-Organizing Maps*. Springer Series in Information Sciences, Springer, New York 1995.

[4] T. Kohonen. New Developments and Applications of Self-Organizing Maps. *Proc. International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing* (NICROSP), Venice, pp. 164-172, IEEE Press, Los Alamitos, 1996.

[5] K. Obermayer, et al. Statistical-Mechanical Analysis of Self-Organization and Pattern Formation During the Development of Visual Maps. *Physical Review A*, Vol. 45(10), pp. 7568-7589, 1992.

[6] St. Schünemann, et al. A Self-Organizing Map for Analysis of High-Dimensional Feature Spaces with Clusters of Highly Differing Feature Density. *Proc. 4th European Symposium on Artificial Neural Networks* (ESANN), pp. 79-84, Bruges 1996.

[7] St. Schünemann, et al. Analysis of Multi-Fluorescence Signals Using a Modified Self-Organizing Feature Map. *Proc. 6th International Conference on Artificial Neural Networks* (ICANN), In: Lecture Notes in Computer Science. Vol. 1112, Springer, pp. 575-580, Berlin 1996.

[8] St. Schünemann, et al. Analysis of Coupled Multi-Image Information in Microscopy. *Proc. 4th International Conference on Visualization in Biomedical Computing* (VBC), In: Lecture Notes in Computer Science. Vol. 1131, Springer, pp. 147-152, Berlin 1996.

[9] U. Seiffert, et al. Three-Dimensional Self-Organizing Maps for Classification of Image Properties. *Proc. The Second New Zealand International Two Stream Conference on Artificial Neural Networks and Expert Systems* (ANNES), Dunedin, pp. 310-313, IEEE Press, Los Alamitos, 1995.

[10] U. Seiffert, et al. Growing 3D-SOM's with 2D-Input Layer as a Classification Tool in a Motion Detection System. In: Bulsari, A.B. et al. (eds.): Solving Engineering Problems with Neural Networks. *Proc. International Conference on Engineering Applications of Neural Networks* (EANN), London, pp. 351-354, Abo Akademis Tryckeri, Turku, 1996.