Online Nonlinear Mapping Using the Neural Gas Network

Pablo A. Estévez and Cristián J. Figueroa
Department of Electrical Engineering
University of Chile
Casilla 412-3, Santiago, Chile
{**pestevez, cfiguero**}**@ing.uchile.cl**

Abstract - *An efficient distance preserving output representation is provided to the neural gas network. The nonlinear mapping is concurrently determined along with the codebook vectors. The adaptation rule for codebook positions in the projection space minimizes a cost function that favors the preservation of the local topology. The results on several data sets show that the proposed strategy outperforms alternative methods such as DIPOL-SOM, TOPNG, SOM/NLM and NG/NLM in terms of the topology preservation measure* q_m.

Key words - neural gas, data projection, data visualization, nonlinear mapping, topology preservation.

1 Introduction

The self-organizing feature map (SOM) [9] has been widely used for vector quantization (VQ) and for data projection and visualization. The VQ techniques encode a manifold of data by using a finite set of reference or "codebook " vectors. The SOM performs VQ under the constraint of a predefined neighborhood between neurons in a discrete output grid. In this way the SOM yields a topographic mapping from the manifold to the output grid. In the conventional SOM the distances between codebook vectors are not directly represented in the map. Post-processing techniques such as Cluster Connections [16], P-matrix [20] or U-matrix [21] allow to incorporate the distance information in the conventional output display by using coloring schemes. Other extensions to the SOM are Adaptive Coordinates [16] and Double SOM [18], which allow to visualize the original structure of the data in a low-dimensional output space without using post-processing techniques. Both models use a heuristic updating rule to move and group the output nodes in a continuous output space, but they do not preserve the intra-cluster and inter-cluster distances as well as the Sammon's mapping [17]. Another SOM extension is DIPOL-SOM [10], which computes a distance preserving projection. The DIPOL-SOM moves nodes in an additional projection layer by using a heuristic online adaptation rule that depends on the conventional SOM neighborhood function, as well as on the sign of the distance preservation error. The quality of the mapping is measured by using the topology preservation measure defined in [11].

A visualization induced SOM (ViSOM) has been proposed to preserve the interneuron distances in the map [23][24]. The ViSOM constrains the lateral contraction force between neurons in the SOM, allowing to preserve the interpoint distances on the input data on the map, along with the topology. However the ViSOM uses the same fixed grid structure of neurons as SOM, and imposes an uniform distribution of the codebook vectors in the input space. As a consequence, ViSOM requires a large number of codebooks to get an adequate quantization error, which entails a heavy computational load. The resolution and the computational cost can be enhanced by interpolating a trained map or incorporating local linear projections [22].

Another enhancement to the SOM is the curvilinear component analysis (CCA) [1]. Firstly, CCA performs VQ of the data manifold in input space using SOM. Secondly, CCA makes a nonlinear projection of the quantizing vectors. The projection part of CCA is similar to multidimensional scaling (MDS) [19] or Sammon's mapping (NLM) [17], since it minimizes a cost function based on the interpoint distances. However the computational complexity of CCA is $O(N)$, while MDS and NLM are $O(N^2)$. Another difference is that the cost function of CCA allows to unfold strongly nonlinear or closed structures. In CCA the output is not a fixed grid but a continuous space that is able to take the shape of the data manifold. The codebook vectors are projected as codebook positions in output space, which are updated by a special adaptation rule. An enhanced version of CCA incorporates curvilinear distances instead of Euclidean distances in the input space [12][13].

The neural gas (NG) is another well-known self-organizing neural network [15]. The main difference with SOM is that NG does not define an output space. The SOM is able to obtain good VQ results only if the topology of the output grid matches the topology of the data manifold, which is usually unknown. As a consequence, NG can outperform SOM when quantizing topologically arbitrary structured manifolds. Instead of a neighborhood function in the output grid, the NG utilizes a neighborhood ranking of the codebook vectors within the input space. In addition, the NG adaptation rule for codebook vectors obeys a stochastic gradient descent of a global cost function, while no cost function exists for the SOM adaptation rule [14].

The lack of an output space has limited the application of NG to data projection and visualization. After learning, the codebook vectors can be projected onto a two-dimensional space using MDS or Sammon's mapping. Alternatively, the authors proposed in [8] an online visualization method called TOPNG, which updates simultaneously the codebook vectors in input space and their corresponding vector positions in a continuous output (projection) space. The TOPNG adds to NG a simple heuristic adaptation rule for codebook positions. When TOPNG is combined with particle swarm optimization the minimum Sammon error is obtained [7]. In a different approach, a cross-entropy embedding of high-dimensional data using the neural gas model has been proposed [2][3]. This method allows to project simultaneously the input data and the codebook vectors onto a projection plane, preserving the relationship defined by the NG neighborhood ranking function.

In this paper we propose a distance preserving nonlinear mapping of the quantizing vectors obtained with the NG algorithm. The codebook positions are adjusted in a continuous output space by using an adaptation rule that minimizes a cost function that favors the local distance preservation. The proposed strategy is online, i.e. the codebook positions are updated simultaneously with the codebook vectors. The performance of the proposed mapping method is compared with DIPOL-SOM, TOPNG, NG/NLM and SOM/NLM, in terms of the topology preserving measure q_m.

2 The Online Visualization Algorithm

2.1 Problem Setting

Let $\{x_i : 1 \leq i \leq M\}$ and $\{w_j : 1 \leq j \leq N\}$ be D-dimensional input and codebook vectors, respectively. For a given set of input vectors, our problem is to adjust simultaneously the codebook vectors in the input space and their respective codebook positions z_j ($j = 1, \ldots, N$) in a A-dimensional continuous output space, with $A << D$.

In order to obtain a distance preserving mapping a cost function is defined, which depends on the difference between the interpoint distances in both the input and the output spaces. Let $d_{j,k}$ and $D_{j,k}$ be the Euclidean distances defined in the input and the output spaces, respectively:

$$d_{j,k} = \|\mathbf{w}_j - \mathbf{w}_k\| = \sqrt{\sum_{d=1}^{D}(w_{j,d} - w_{k,d})^2}, \tag{1}$$

$$D_{j,k} = \|\mathbf{z}_j - \mathbf{z}_k\| = \sqrt{\sum_{a=1}^{A}(z_{j,a} - z_{k,a})^2}. \tag{2}$$

In the Neural Gas model, the neighborhood ranking function of the codebook vectors $\mathbf{w}_j$, for $j = 1, \cdots, N$, with respect to a given input vector x_i, is defined as follows:

$$h_\lambda(x_i, w_j) = e^{-\dfrac{r(x_i, w_j)}{\lambda(t)}}, \tag{3}$$

where $r(x_i, w_j) \in \{0, 1, \ldots, N-1\}$ denotes the rank of the j^{th} codebook vector, and the parameter λ controls the width of the neighborhood function. Likewise, for mapping purposes it is convenient to introduce a ranking of the codebook positions $\mathbf{z}_j, j = 1, \cdots, N$, with respect to a given output vector y_i. The term $s(y_i, z_j) \in \{0, 1, \ldots, N-1\}$ denotes the rank of the j^{th} codebook position. Here $r = 0$ and $s = 0$ are associated with the nearest codebook vector and the nearest codebook position, respectively.

2.2 New Position Adaptation Rule

A global cost function similar to the one used by CCA [1] is introduced:

$$E = \frac{1}{2}\sum_{j=1}^{N}\sum_{k\neq j}(D_{j,k} - d_{j,k})^2 F(s_{j,k}) = \frac{1}{2}\sum_{j=1}^{N}\sum_{k\neq j} E_{j,k}, \tag{4}$$

where the weighting function $F(s_{j,k})$, is a bounded and monotonically decreasing function, in order to favor local topology preservation, since a perfect matching is not always possible when making a low-dimensional embedding of high-dimensional data. The eq. (4) could be minimized with respect to z_j, by using stochastic gradient descent. In that case a codebook position would be adjusted according to the sum of every other codebook influences. The complexity of such an algorithm is $O(N^2)$. Instead in our method, the codebook position associated to the winner unit, $z_j^*(t)$, is fixed, and all the other positions are moved towards the winner's position, disregarding any interaction among them, thus reducing the complexity to $O(N)$. Using this method the cost function will decrease on average as shown in [1]. This approach is similar to CCA, but CCA is an offline method since it performs the nonlinear mapping after the codebook vectors have been learned by SOM. Our approach is online and it uses NG as a vector quantizer instead of SOM. Another difference is that in our approach the weighting function F in (4) is an exponential function of the ranking of position vectors in the output space, while in CCA the weighting function F is a step function of the interpoint Euclidean distances in output space. In order to minimize (4), the derivative of the local objective function $E_{j,k}$ with respect to position vector z_j is calculated by utilizing the chain rule as follows:

$$\frac{\partial E_{j,k}}{\partial z_j} = F(s_{j,k})\frac{(D_{j,k} - d_{j,k})}{D_{j,k}}(z_j - z_k), \tag{5}$$

where the quantized function F is considered constant with respect to the variation of the codebook positions $z_j(t)$, independent of the choice of F and the distance measurement in output space. Consequently, the updating rule for the codebook positions is as follows:

$$z_j(t+1) = z_j(t) - \alpha(t)F(s_{j,k})\frac{(D_{j,k} - d_{j,k})}{D_{j,k}}(z_j(t) - z_k(t)), \tag{6}$$

where $\alpha(t)$ is the learning rate, which typically decreases with the number of iterations t.

2.3 Learning Algorithm

The initial topology of the network is a set of N neurons. The j^{th} neuron has associated a D-dimensional codebook vector, w_j, and a two-dimensional codebook position, z_j ($j = 1, \ldots, N$).

1. Initialize the codebook vectors, w_j, and the codebook positions, z_j, randomly.

2. Present an input vector, $x_i(t)$ to the network ($i = 1, \ldots, M$) at iteration t.

3. Find the best matching unit (BMU), j^* using:

$$j^* = \text{argmin}_{j=1 \ldots N} \|x_i(t) - w_j(t)\|, \tag{7}$$

   and generate the ranking $r(w_j^*(t), w_j(t)) \in \{0, 1, \ldots, N-1\}$ for each codebook vector $w_j(t)$ with respect to the winner $w_j^*(t)$.

4. Update the codebook vectors:

$$w_j(t+1) \quad = \quad w_j(t) + \epsilon(t)h_\lambda(t)(x_i(t) - w_j(t)), \tag{8}$$

   where $h_\lambda(t)$ = $h_\lambda(\mathbf{x}_i, \mathbf{w}_j)$ is the neighborhood function defined in (3), and $\epsilon(t)$ is the learning rate. In our experiments $\epsilon(t)$ was decreased linearly from 0.3 to 0.0001 in $3000 \times M$ iterations.

5. Generate the ranking $s(z_j^*(t), z_j(t)) \in \{0, 1, \ldots, N-1\}$ for each codebook position $z_j(t)$ with respect to the codebook position associated to the winner unit $z_j^*(t)$.

6. Update the codebook positions:

$$z_j(t+1) = z_j(t) + \alpha(t)F(s(z_j^*(t), z_j(t))\frac{(D_{j,j^*} - d_{j,j^*})}{D_{j,j^*}}(z_{j^*}(t) - z_j(t)). \tag{9}$$

   Alternatively $F(s(z_j^*, z_j))$ could be changed to $F(r(w_j^*, w_j))$ in (9). In both cases $F(f)$ is defined as

$$F(f) = e^{-\dfrac{f}{\lambda_f}}, \tag{10}$$

   where λ_f is a user-defined constant. Here $\alpha(t)$ was set the same way than $\epsilon(t)$.

7. If $t < t_{max}$ go back to step 2).

2.4 Mapping Quality

The topology preservation measure q_m defined in [11] is considered as a performance measure. It is based on an assessment of rank order in the input and output spaces. The n nearest codebook vectors NN_{ji^w} ($i \in [1, n]$) of each codebook vector j, ($j \in [1, N]$) and the n nearest codebook positions NN_{ji^z} of each codebook position j are computed. The global q_m measure is defined as:

$$q_m = \frac{1}{3nN} \sum_{j=1}^{N} \sum_{i=1}^{n} q_{m_{ji}}, \tag{11}$$

where

$$q_{m_{ji}} = \begin{cases} 3, & \text{if } NN_{ji^w} = NN_{ji^z} \\ 2, & \text{if } NN_{ji^w} = NN_{jl^z}, l \in [1, n], i \neq l \\ 1, & \text{if } NN_{ji^w} = NN_{jt^z}, t \in [n, k], n < k \\ 0, & \text{else.} \end{cases} \tag{12}$$

Typical parameter settings are $n = 4$ and $k = 10$. The range of the q_m measure is between 0 and 1, where $q_m = 0$ indicates a poor neighborhood preservation between the input and output spaces, and $q_m = 1$ indicates a perfect neighborhood preservation.

3 Simulation Results

In this section, the simulation results obtained with the proposed model, called OVI-NG (Online VIsualization Neural Gas), are presented. For comparison purposes, the results obtained with the online strategies DIPOL-SOM and TOPNG, and the offline strategies DIPOL-SOM offline, SOM/NLM and NG/NLM are also presented. A variant of our model is considered, called OVI-NG-2, where the weighting function $F(r)$ is used in (9) instead of $F(s)$.

Four different data sets were considered: Iris, Sleep, Fraud, and Wood. The parameter λ_f in (10) was set to 12.5, 50, 75, and 100, for each data set, respectively. The number of codebooks was set to 70, 200, 300, and 400, for each data set, respectively. The Iris data set contains 150 samples, with 4 attributes, distributed in three types of Iris flowers. The Sleep data set contains features extracted from polysomnographic recordings [4]. The Sleep data set contains 6463 samples, with 6 attributes and 6 classes. The Fraud data set has information about telephone subscribers [5], and contains 10624 samples, with 26 attributes and 3 classes. The Wood data set contains features extracted from 16800 color images of wood surface defects [6], with 64 attributes and 11 classes.

Fig. 1 shows the codebook positions obtained for the Iris data set with DIPOL-SOM, NG/NLM, OVI-NG-2, and OVI-NG. Figs. 2a and 2b show the codebook positions obtained for the Sleep data set with OVI-NG-2 and OVI-NG, respectively. Figs. 2c and 2d show the codebook positions obtained for the Fraud data set with OVI-NG-2 and OVI-NG, respectively. The position vectors were labelled with the majority class of the input vectors present in the Voronoi cell of the corresponding codebook vector. If the percentage of input vectors of the majority class is greater than a given threshold, the node is marked with the number of that class. Otherwise, the node is represented by a black ball with an exclamation sign inside. If the node wins no input vectors, it is marked with a number 0.

Fig. 3a shows the topology preservation measure q_m as a function of λ_f for both the OVI-NG and the OVI-NG-2 algorithms, with the Iris data set. The maximum q_m value is obtained for $\lambda_f = 12.5$. Fig. 3b shows the topology preservation measure q_m as a function of the number of codebook vectors for both the OVI-NG and the OVI-NG-2 algorithms, with the Iris data set.

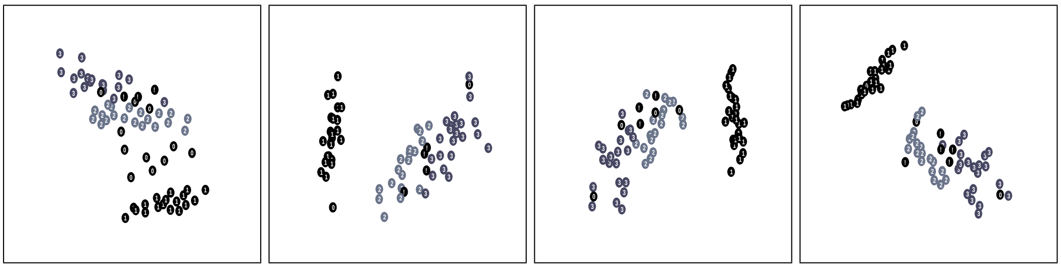

(a) (b) (c) (d)

Figure 1: Iris data set (150x4) projection results using 70 neurons (position vectors): (a) DIPOL-SOM online, (b) NG/NLM, c) OVI-NG-2, d) OVI-NG.

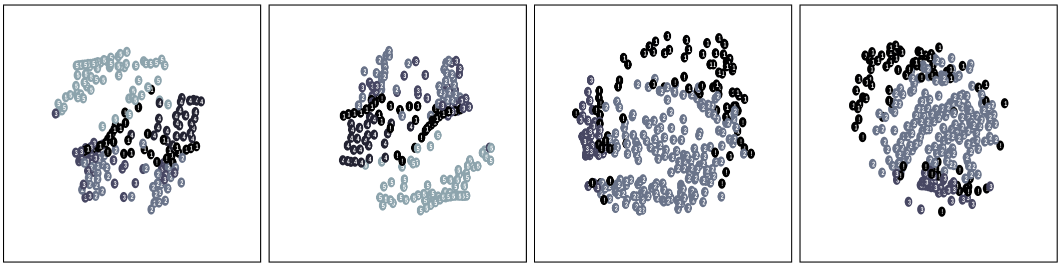
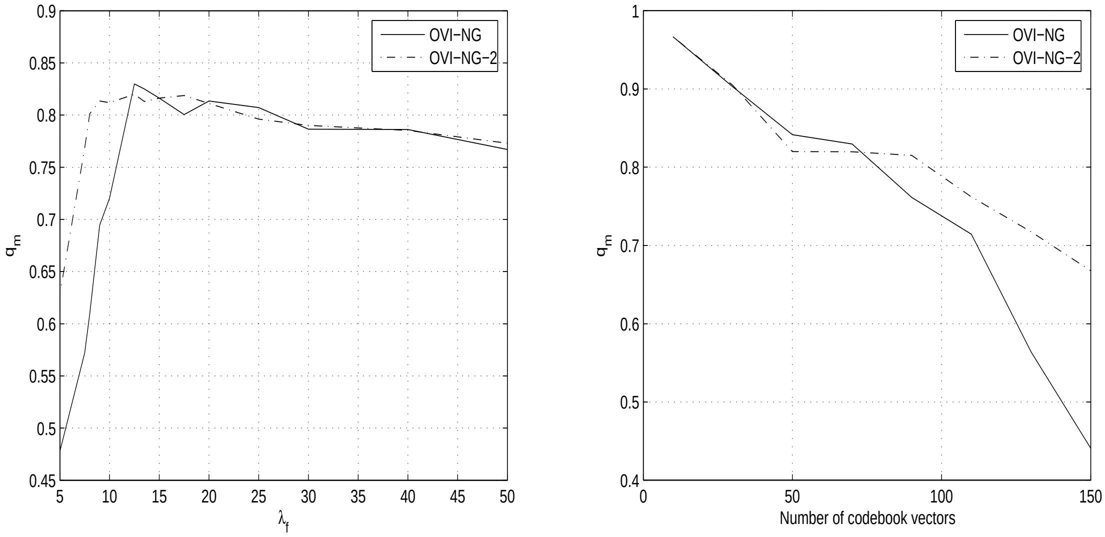

(a) (b) (c) (d)

Figure 2: Projection results using 200 neurons (position vectors) for the Sleep data set (6463x6) and 300 neurons for the Fraud data set (10624x26): (a) OVI-NG-2 (sleep), b) OVI-NG (sleep), c) OVI-NG-2 (fraud), and d) OVI-NG (fraud).



(a) (b)

Figure 3: Topology preservation measure q_m as a function of a) λ_f and b) the number of codebook vectors for $\lambda_f = 12.5$, for both the OVI-NG and OVI-NG-2 algorithms applied to the Iris data set.

Table 1: Topology preservation measure q_m for the different algorithms and data sets considered.

Algorithm	Iris	Sleep	Fraud	Wood
OVI-NG	**0.8298 ± 0.0120**	**0.6029 ± 0.0015**	**0.4817 ± 0.0182**	**0.3467 ± 0.0114**
OVI-NG-2	0.8198 ± 0.0138	0.5979 ± 0.0188	0.4622 ± 0.0111	0.2977 ± 0.0019
Dipol-SOM online	0.7405 ± 0.0235	0.5214 ± 0.0288	0.3068 ± 0.0236	0.2219 ± 0.0187
Dipol-SOM offline	0.7669 ± 0.0063	0.5107 ± 0.0395	0.3123 ± 0.0318	0.2179 ± 0.0227
SOM/NLM	0.7712 ± 0.0114	0.5746 ± 0.0143	0.4461 ± 0.0301	0.3107 ± 0.0219
NG/NLM	0.6988 ± 0.0154	0.5545 ± 0.0065	0.3614 ± 0.0080	0.2054 ± 0.0267
TOPNG	0.6812 ± 0.0064	0.5219 ± 0.0051	0.3912 ± 0.0057	0.2483 ± 0.0051

Table 1 shows the q_m value averaged over 5 simulation runs, and the standard deviation, for the 7 algorithms considered. For all data sets, the proposed OVI-NG method obtained the highest q_m value, yielding the best topology preservation.

4 Conclusions

The proposed OVI-NG method provides an output representation to the neural gas model that is useful for data projection and visualization. The OVI-NG is an online method that concurrently adjusts the codebook vectors in input space and the codebook positions in a continuous output space. The method is computationally efficient with $O(N)$ complexity. The position adaptation rule minimizes a cost function similar to CCA, which favors the local topology. In this sense, the OVI-NG method could be regarded as online CCA but other differences include the use of a neighborhood ranking instead of Euclidean distances in both the output and input spaces, and the use of NG instead of SOM as vector quantizer. For all data sets, the OVI-NG method obtained the best results in terms of the topology preservation measure q_m. In future research, the OVI-NG could be extended to map all data points in an efficient way, using the codebook vectors and their positions as a reference. Another possible extension is to incorporate curvilinear or graph distances, in order to unfold strongly nonlinear maps.

5 Acknowledgement

This research was supported by Conicyt-Chile, under grant Fondecyt 1050751. The authors want to thank Research Assistant Andrés M. Chong for his help with the experiments.

References

[1] P. Demartines & J. Hérault (1997), Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets, *IEEE Trans. on Neural Networks*, **vol.** 8, no. 1, pp. 148–154.

[2] P. A. Estévez, C. J. Figueroa, & K. Saito (2005). Cross–entropy embedding of high dimensional data using the neural gas model. *Neural Networks*, Special Issue on IJCNN2005 (in press).

[3] P. A. Estévez, P. A., C. J. Figueroa, & K. Saito (2005). Cross–entropy approach to data visualization based on the neural gas network. *Proceedings of the International Joint Conference on Neural Networks*, Montreal, Canada (in press).

[4] P. A. Estévez, C. M. Held, C. A. Holzmann, C. A. Perez, J. P. Pérez, J. Heiss, M. Garrido & P. Peirano (2002), Polysomnographic pattern recognition for automated classification of sleep-waking states in infants, *Medical & Biomedical Engineering & Computing*, **vol.** 40, pp. 105–113.

[5] P. A. Estévez, C. M. Held, & C. A. Perez (2005), Prevention of subscription fraud in telecommunications using fuzzy rules and neural networks (unpublished).

[6] P. A. Estévez, C. Perez, & E. Goles (2003), Genetic input selection to a neural classifier for defect classification of radiata pine boards, *Forest Products Journal*, **vol.** 53, no. 7/8, pp. 87–94, July/August.

[7] C. J. Figueroa, P. A. Estévez & R. Hernández (2005), Non-linear mappings based on particle swarm optimization, *Proceedings of the International Joint Conference on Neural Networks*, Montreal, Canada (in press).

[8] C. J. Figueroa & P. A. Estévez (2004), A new visualization scheme for self-organizing neural networks, *Proceedings of the International Joint Conference on Neural Networks*, pp. 757–762, Budapest, Hungary.

[9] T. Kohonen (1995), *Self–Organizing Maps*, Berlin, Germany, Springer-Verlag.

[10] A. König & T. Michel (2003), DIPOL-SOM - A distance preserving enhancement of the self-organizing map for dimensionality reduction and multivariate data visualization, *Proceedings of the Workshop on Self-Organizing Maps (WSOM'03)*, pp. 219–224.

[11] A. König (2000), Interactive visualization and analysis of hierarchical neural projections for data mining, *IEEE Trans. on Neural Networks*, **vol.** 11, no. 3, pp. 615–624.

[12] J. A. Lee, A. Lendasse, & M. Verleysen (2004), Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis, *Neurocomputing*, **vol.** 57, pp. 49–76.

[13] J. A. Lee, A. Lendasse, N. Donckers, & M. Verleysen (2000), A robust nonlinear projection method, *Proceedings of the European Symposium on Artificial Neural Networks (ESSAN'2000)*, pp. 13–20.

[14] T. M. Martinetz, S. G. Berkovich, & K. J. Schulten (1993). "Neural gas" network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, **vol.** 4, pp. 558–569.

[15] T. M. Martinetz, & K. J. Schulten (1991). A neural gas network learns topologies. *Artificial Neural Networks*, 397–402, Elsevier.

[16] D. Merkl, & A. Rauber (1997), Alternative ways for cluster visualization in self-organizing maps, *Proceedings of the Workshop on Self-Organizing Maps (WSOM'97)*, pp. 106–111.

[17] J. W. Sammon (1969), A nonlinear mapping for data structure analysis, *IEEE Trans. on Computers*, **vol.** C-18, pp. 401–409.

[18] M. C. Su, & H. T. Chang (2001), A new model of self-organizing neural networks and its application in data projection, *IEEE Trans. on Neural Networks*, **vol.** 12, pp. 153–158.

[19] W. S. Togerson (1958), *Theory and methods of scaling*, New York, Wiley.

[20] A. Ultsch (2003), Maps for the visualization of high-dimensional data spaces, *Proceedings of the Workshop on Self-Organizing Maps (WSOM'03)*, pp. 225–230.

[21] A. Ultsch, & H. P. Siemon (1990), Kohonen's self-organizing feature maps for exploratory data analysis, *Proceedings of the International Neural Network Conference, INNC'90*, pp. 305–308.

[22] H. Yin (2003), Resolution enhancement for the ViSOM, *Proceedings of the Workshop on Self-Organizing Maps (WSOM'03)*, pp. 208–212.

[23] H. Yin (2002), Data visualization and manifold mapping using the VISOM, *Neural Networks*, **vol.** 15, pp. 1005–1016.

[24] H. Yin (2002), ViSOM- A novel method for multivariate data projection and structure visualization, *IEEE Trans. on Neural Networks*, **vol.** 13, pp. 237–243.