Nonlinear ASSOM
Constituted of Autoassociative Neural Modules

K. Tokunaga **T. Furukawa**
Department of Brain Science and Engineering, Kyushu Institute of Technology
Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan
tokunaga-kazuhiro@edu.brain.kyutech.ac.jp
furukawa@brain.kyutech.ac.jp

Abstract - *We propose a method for realizing the nonlinear adaptive subspace SOM (nonlinear ASSOM). The proposed method is a variation of the modular network SOM (mnSOM) constituted by autoassociative neural networks (ANN-mnSOM). The mapping objects of ANN-mnSOM are classes made up of data vectors. Thus a map generated by an ANN-mnSOM represents the similarities or differences between the class distributions. Here a learning algorithm derived from the EM algorithm is introduced, and some simulation results are presented.*

Key words - ASSOM, nonlinear subspace, mnSOM, autoassociative neural network

1 Introduction

One of the directions for a generalizing SOM is to extend the mapping objects from vectorized data to other data types. One example of such attempts is the Operator Map in which each nodal unit of a conventional SOM is replaced by an operator, such as an autoregressive (AR) estimator[1]. Thus, the Operator Map is an extension of an SOM, the mapping objects of which are linear systems. Another example is the adaptive subspace SOM (ASSOM) that deals with a set of data distributions[2]. In an ASSOM, each nodal unit represents a linear subspace formed by a set of basis vectors. The best matching unit of the ASSOM is defined as the unit that approximates the data distribution best for each class. Thus an ASSOM can be regarded as a generalization of an SOM in which the reference points are extended to the reference planes. The ideas of both the Operator Map and the ASSOM can be naturally extended to nonlinear cases. However, adequate adaptation algorithms dealing with nonlinear tasks are left undetermined.

More recently the authors have proposed the idea of adopting the modular network scheme to an SOM called the modular network SOM (mnSOM)[3, 4]. Our mnSOM consists of an array of function modules, e.g. multilayer perceptrons (MLPs). As the result, the mnSOM has acquired the ability of dealing with nonlinear functions, and nonlinear dynamical systems, etc.[3, 4, 5]. Therefore an mnSOM would be a solution for a nonlinear operator map. Furthermore, if one were to employ nonlinear PCA neural networks as the function modules, then the mnSOM is expected to be a nonlinear ASSOM.

There are several methods for neural implementation of PCA modules. Amongst them, one of the most popular and convenient ways is to use autoassociative neural networks (ANNs). An ANN is a variation of an MLP, its task being to reproduce input vectors as output. It is known that a 3-layer

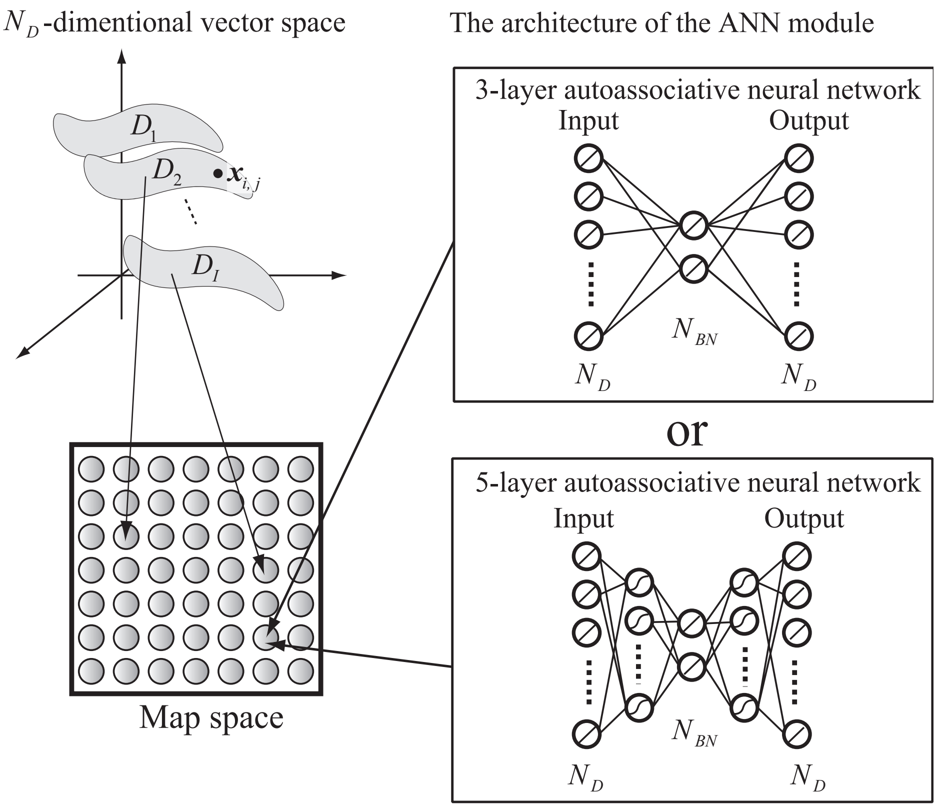


Figure 1: The architecture and the scheme of the ANN-mnSOM

ANN (3L-ANN) approximates the data distribution of a linear subspace, whereas a 5-layer ANN (5L-ANN) represents the distribution of a curved subspace, i.e. a manifold. Thus, 3L-ANN and 5L-ANN are equivalent to linear- and nonlinear-PCA, respectively. Therefore an mnSOM with 5L-ANN modules (5L-ANN-mnSOM) would be the solution for a nonlinear ASSOM.

In this paper, the architecture and algorithm of an ANN-mnSOM are introduced, and then some simulation results are reported.

2 Theoretical Framework

An autoassociative neural network (ANN) is a variation of an MLP, which has a mirror symmetrical structure. The numbers of the input and output units are the same number as the dimension of the data vector N_{D}, whereas the unit number of the middle layer (the bottleneck layer) N_{BN} is set to be smaller than N_{D}. The task of an ANN is to reproduce input as output. Because the N_{BN} is less than N_{D}, an ANN can at most only reproduce the vectors which belong to an N_{BN}-dimensional manifold. The manifold is determined by the weight vector of the ANN, which is trained to minimize the mean square error between the input vectors and the corresponding outputs, i.e. the reproduced input vectors. As a result, the manifold becomes the approximation of the distribution of the training data vectors. The 3L-ANN can represent only the N_{BN}-dimensional linear subspace that is equivalent to the space formed by the N_{BN}th principle components, whereas the 5L-ANN does not have such a limitation.

If we suppose that there are I classes $\{C_1, \ldots, C_I\}$ each of which has J sample vectors, then the sample dataset of C_i is represented by $D_i = \{\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,J}\}$. Suppose further that D_i is produced by the probability density function (pdf) $p_i(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{\theta}_i)$. Here $\boldsymbol{\theta}_i$ is a hidden parameter that determines $p_i(\boldsymbol{x})$. In addition, the pdf $p(\boldsymbol{x}|\boldsymbol{\theta})$ is assumed to vary continuously when the hidden parameter $\boldsymbol{\theta}$ changes.

The architecture and the scheme of an ANN-mnSOM are shown in Fig. 1. An ANN-mnSOM is an assembly of the function modules of a 3L- or 5L-ANN. These ANN modules are arrayed on a lattice that represents the coordinates in the map space. Thus, each module has a fixed position in the map space. Suppose that an mnSOM has K ANN modules $\{M^1, \ldots, M^K\}$, the coordinates of which are $\{\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^K\}$ in the map space. The weight vector $\boldsymbol{w}^k$ determines the manifold U^k represented by M^k. (Here the subscripts mean the index of the data, while the superscripts mean the index of the mnSOM.)

An ANN-mnSOM aims to approximate the set of pdfs $\{p_i(\boldsymbol{x})\}$ by the ANN modules and simultaneously to generate a map of $\{C_i\}$. The map is expected to represent the degree of similarities or differences between the pdfs $\{p_i(\boldsymbol{x})\}$. More precisely, the tasks for an ANN-mnSOM are (i) to estimate $\{p_i(\boldsymbol{x})\}$ by the set of manifolds $\{U^k\}$, and (ii) to undertake homeomorphic mapping from the hidden parameter space to the map space $\Psi : \boldsymbol{\theta} \to \boldsymbol{\xi}$. As a result, the pdfs $\{p(\boldsymbol{x}|\boldsymbol{\theta}_i)\}$ can be replaced by $\{p(\boldsymbol{x}|\boldsymbol{\xi}_i)\}$. Therefore the task of the ANN-mnSOM is regarded as being to estimate $p(\boldsymbol{x}|\boldsymbol{\xi})$ and $\{\boldsymbol{\xi}_i\}$, which are the alternative expressions of $p(\boldsymbol{x}|\boldsymbol{\theta})$ and $\{\boldsymbol{\theta}_i\}$.

The learning algorithm of an ANN-mnSOM is same as the case of the MLP-module-mnSOM which has been described elsewhere[4]. Briefly, the *best matching module* (the *BMM*) is determined for each class, then the BMM and its neighbors achieve a larger learning rate than other modules for the class. This algorithm is a straightforward extension of a conventional SOM. However, here we take another approach, described by an extended EM algorithm.

Now let $\hat{\boldsymbol{x}}^k$ be the output of the kth ANN module, i.e. the output of M^k, when a input vector $\boldsymbol{x}$ is presented. Since $\hat{\boldsymbol{x}}^k$ is the projection of $\boldsymbol{x}$ to the manifold U^k, the distance between C_i and U^k can be defined as the expectation of the square error $\|\boldsymbol{x} - \hat{\boldsymbol{x}}^k\|^2$ that is approximated by the average error of the sample data, as follows.

$$L^2(C_i, M^k) = \int \|\boldsymbol{x} - \hat{\boldsymbol{x}}^k\|^2 \, p_i(\boldsymbol{x}) d\boldsymbol{x} \tag{1}$$

$$\simeq \frac{1}{J} \sum_{j=1}^{J} \|\boldsymbol{x}_{i,j} - \hat{\boldsymbol{x}}^k_{i,j}\|^2 \equiv E_i^k \tag{2}$$

The BMM of C_i is defined as the least mean square module M_i^*. Thus, by letting k_i^* be the index of M_i^*, k_i^* is given by

$$k_i^* = \arg\min_k E_i^k \tag{3}$$

The tentative estimation of $\boldsymbol{\xi}_i$ is given by the BMM, i.e. $\boldsymbol{\xi}_i^* = \boldsymbol{\xi}^{k_i^*}$. Now let us assume that the real $\boldsymbol{\xi}_i$ is distributed around $\boldsymbol{\xi}_i^*$ as given by Gaussian distribution, as follows[1].

$$p(\boldsymbol{\xi}_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^*\|^2}{2\sigma^2}\right] \tag{4}$$

Although $\boldsymbol{\xi}_i$ in (4) is a continuous variable, the modules are located at discrete positions. Consequently $\boldsymbol{\xi}_i$ is quantized to the nearest module position $\boldsymbol{\xi}^k$. Now let R^k denote the area that belongs to M^k. Then

[1] $p(\boldsymbol{\xi}_i)$ should more precisely be written as $p(\boldsymbol{\xi}_i|\boldsymbol{\xi}_i^*)$. However the condition part is omitted for simplification.

the discrete probability that $\boldsymbol{\xi}_i$ belongs to M^k, i.e., $P(\boldsymbol{\xi}_i \in R^k) = P(M^k|\boldsymbol{\theta}_i)$ is given by,

$$P(M^k|\boldsymbol{\theta}_i) = \int_{\boldsymbol{\xi} \in R^k} p(\boldsymbol{\xi}|\boldsymbol{\theta}_i) d\boldsymbol{\xi} \simeq \frac{S(R^k)}{\sqrt{2\pi}\sigma} \exp\left[-\frac{\|\boldsymbol{\xi}^k - \boldsymbol{\xi}_i^*\|^2}{2\sigma^2}\right] \tag{5}$$

Here $S(R^k)$ gives the size of R^k, which is suppose to be equal for all modules. The equation (5) is the so-called neighborhood function in a conventional SOM. By applying Bayes' rule, the posterior probability is given by,

$$P(\boldsymbol{\theta}_i|M^k) \equiv \Phi_i^k = \frac{\exp\left[-\|\boldsymbol{\xi}^k - \boldsymbol{\xi}_i^*\|^2/2\sigma^2\right]}{\sum_{i'=1}^{I} \exp\left[-\|\boldsymbol{\xi}^k - \boldsymbol{\xi}_{i'}^*\|^2/2\sigma^2\right]} \tag{6}$$

Now the expectation of the square error of the kth module $\langle E^k \rangle$ is evaluated as follows.

$$\langle E^k \rangle = E\left[\|\boldsymbol{x} - \hat{\boldsymbol{x}}^k\|^2\right] \tag{7}$$

$$= \sum_{i=1}^{I} \int \|\boldsymbol{x} - \hat{\boldsymbol{x}}^k\|^2 \, p(\boldsymbol{x}|\boldsymbol{\theta}_i) P(\boldsymbol{\theta}_i|M^k) d\boldsymbol{x} \tag{8}$$

$$= \sum_{i=1}^{I} \Phi_i^k \int \|\boldsymbol{x} - \hat{\boldsymbol{x}}^k\|^2 \, p_i(\boldsymbol{x}) d\boldsymbol{x} \simeq \sum_{i=1}^{I} \Phi_i^k E_i^k \tag{9}$$

The evaluation of $\langle E^k \rangle$ corresponds to the E-step. After that each ANN module is trained so as to minimize the expectation error $\langle E^k \rangle$ by the gradient descent algorithm. Thus the weight vectors are updated iteratively by the following equation.

$$\Delta \boldsymbol{w}^k = -\eta \frac{\partial \langle E^k \rangle}{\partial \boldsymbol{w}^k} = -\eta \sum_{i=1}^{I} \Phi_i^k \frac{\partial E_i^k}{\partial \boldsymbol{w}^k} \tag{10}$$

This process corresponds to the M-step. (10) is the conventional backpropagation algorithm with the learning rate $\eta\Phi_i^k$ for each class. Therefore the BMM M_i^* and its neighbors learn the data D_i more than other modules.

The above equations can be summarized as follows. (2) and (3) are the competitive process in which the BMM of each class is determined, and (6) is the cooperative process in which the learning rates $\{\Phi_i^k\}$ are calculated. At last the weight vectors are updated by (10) in the adaptive process. These 3 processes are iterated with decreasing σ until the network reaches the stable state.

3 Simulation Results

3.1 A Map of the Sine-Wave Family

The first simulation is to demonstrate that the 3L-ANN-mnSOM is equivalent to the ASSOM, though this has been already reported by Zhang et al. in a recognition task involving hand written digits[6]. Here we used a much easier task, the aim being to map a sine-wave family. In this task, each class consisted of sampled sine-waveforms with various phases and amplitudes, while the frequency was

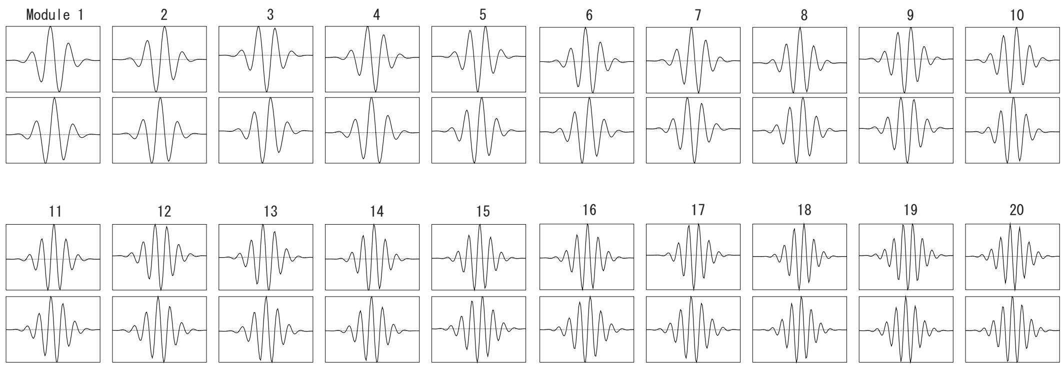


Figure 2: One dimensional map generated by 3L-ANN-mnSOM. Each pair of waveforms represents the pair of basis functions that emerged in the corresponding ANN module

kept fixed for each class. Thus the sampled waveform $s_{ij}(t)$ and the corresponding data vector $\boldsymbol{x}_{i,j}$ were given by,

$$s_{ij}(t) = A_{ij}\sin(2\pi f_i\, t + \phi_{ij}) \tag{11}$$

$$\boldsymbol{x}_{i,j} = [s_{ij}(0), \ldots, s_{ij}(T-1)]^T \tag{12}$$

The frequency of each class is defined by,

$$f_i = f_0 \cdot 2^{i/12} \qquad (i = 0, \ldots, 12) \tag{13}$$

Therefore the classes $\{C_0, \ldots, C_{12}\}$ corresponded to the scale of {C, C♯, D, D♯,. . . , B♭, B, C(2)}, and the frequency was the hidden parameter. Since the data vectors of each class D_i were distributed in a 2-dimensional linear subspace, a 3L-ANN with 2 bottleneck layer units was expected to represent the distribution of a single class. In the simulation, the mnSOM had 20 modules of such 3L-ANNs arrayed on a one dimensional map space. Further, the sampled waveforms were filtered by a Gaussian window so that the boundary conditions all became zero, though this process was not essential.

The result is shown in Fig. 2. The figure shows pairs of the basis functions that emerged in the ANN modules. As shown in the figure, the basis functions lined up in order of the frequency. In addition, the phases of the paired two basis functions differed by approximately 90 degree for all classes. These results demonstrated that the 3L-ANN mnSOM worked in the same manner as the ASSOM[2].

3.2 A Map of the Periodical Waveforms

The task of the second simulation was almost same as the first. Here, the periodical waveforms were increased to three different types, namely, sinusoidal, rectangular and pyramidal ones. Each waveform had 13 classes corresponding to 13 notes from C to C(2), thus there were 39 classes. Since the data vectors of each class were distributed in a 2-dimensional nonlinear subspace, 5L-ANN modules with 2 bottleneck layer units were employed here. The map space was 2-dimensional and 13×13 modules were arrayed.

Fig. 3 shows the map generated by the ANN-mnSOM. The map showed a concentric structure in which classes belonging to each waveform were located on a circle, whereas the classes belong to each frequency were lined radially. Thus, the parameter space consisted of frequency and waveform

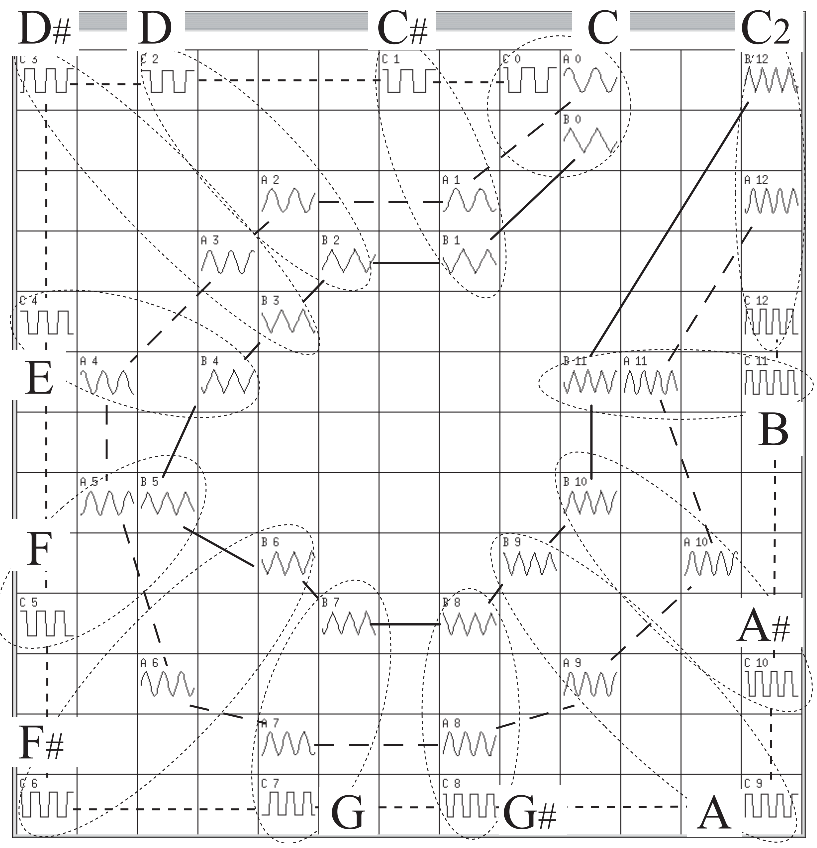


Figure 3: Map of periodical waveforms generated by the 5L-ANN-mnSOM

were preserved in the map space. Therefore, it can be concluded that the 5L-ANN-mnSOM worked as a nonlinear ASSOM.

3.3 Map of 3D Objects From 2D Projected Images

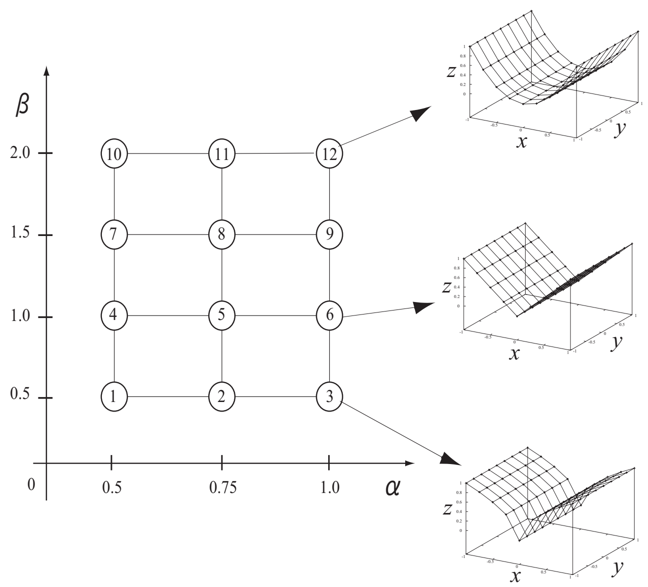
The third simulation dealt with a set of 3D objects. Each class corresponded to a 3D object, which consisted of a group of 2D images taken from various viewpoints. Therefore, the task is to generate a map of 3D objects from a set of photographs. Since the 2D images formed a 2-dimensional manifold, the 5L-ANN-mnSOM ($N_{\mathrm{BN}} = 2$) was used. It should be noted that an mnSOM does not know how to reconstruct 3D objects from photographs.

In the simulation, each object was assumed to be formed by a flexible grid. The shape of the object was given by a set of 3D coordinates of the nodes, $\{(x_{i,n}, y_{i,n}, z_{i,n})\}$, which were generated by the following equation.

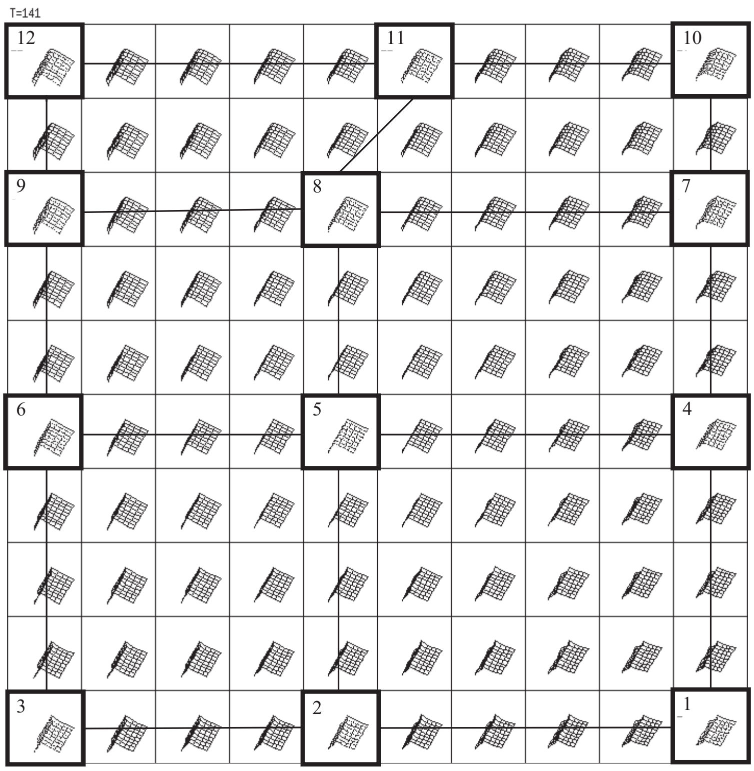
$$z_{i,n} = \alpha_i |x_{i,n}|^{\beta_i} \tag{14}$$

Here two hidden parameters α_i and β_i determined the depth and the roundness, respectively, of the objects, and the 12 objects presented in Fig. 4(a) were used. Each sample vector consisted of a set of 2D coordinates representing the node positions in the 2D images taken from a viewpoint.

Fig. 4(b) shows the map of the 3D objects generated by the 5L-ANN-mnSOM. The 3D object depicted in each box corresponds to the one represented by the ANN-module. The numbered mod-



(a)



(b)

Figure 4: (a) The parameters of the 12 training objects used in the simulation. (b) A map of 3D objects generated by the 5L-ANN-mnSOM from the sets of 2D images

ules are the BMMs of the training classes corresponding to the objects in Fig. 4(a). The map space perfectly captured the topology of the hidden parameter space (α, β). Furthermore, the intermediate ANN-modules represented the intermediate 3D objects that were not given during the training.

4 Conclusion

In this paper, we have proposed a modular network SOM made up of autoassociative neural networks (ANN-mnSOM). Theoretical and simulation results have shown that an 5L-ANN-mnSOM is equivalent to a nonlinear-ASSOM. The simulation results also suggest that (i) an ANN-mnSOM estimates the distribution of the given classes, (ii) an ANN-mnSOM generates a map space in which the topology of the hidden parameters is preserved, and (iii) an ANN-mnSOM interpolates the data distributions between the training classes. From these results, it can be seen that an ANN-mnSOM is a powerful tool for tasks of multi-class data analysis, as a nonlinear ASSOM.

Acknowledgement

This work was supported by a COE program (center #J19) granted to Kyushu Institute of Technology by MEXT of Japan.

References

[1] Kohonen, T. (1993) Generalization of the Self-organizing map. Proc. of IJCNN93, pp.457–462

[2] Kohonen, T., Kaski, S., Lappalainen, H. (1993) Self-organized formation of various invariant-feature filters in the adaptive-subspace SOM. Neural Computation **Vol.9**, pp.1321–1344

[3] Tokunaga, K., Furukawa, T., Yasui, S. (2003) Modular network SOM: Extension of SOM to the realm of function space. Proc. of WSOM2003, pp.173–178

[4] Furukawa, T., Tokunaga, K., Morishita, K., Yasui, S. (2005) Modular network SOM (mnSOM): From vector space to function space. Proc. of IJCNN2005 (*accepted*)

[5] Furukawa, T., Tokunaga, K., Kaneko, S., Kimotsuki, K., Yasui, S. (2004) Generalized self-organizing maps (mnSOM) for dealing with dynamical systems. Proc. of NOLTA2004, pp.231–234

[6] Zhang, B., Fu, M., Yan, H., Jabri, M.A. (1999) Handwritten digit recognition by adaptive-subspace self-organizing map (ASSOM). IEEE Trans. Neural Networks, **Vol.10** No.4, pp.939–945