Modular network SOM : Theory, algorithm and applications

Kazuhiro Tokunaga and Tetsuo Furukawa

Kyushu Institute of Technology, Kitakyushu 808-0196, Japan
{tokunaga, furukawa}@brain.kyutech.ac.jp

Abstract. The modular network SOM (mnSOM) proposed by authors is an extension and generalization of a conventional SOM in which each nodal unit is replaced by a module such as a neural network. It is expected that the mnSOM will extend the area of applications beyond that of a conventional SOM. We set out to establish the theory and algorithm of a mnSOM, and to apply it to several research topics, to create a fundamental technology that is generally usable only in expensive studies. In this paper, the theory and the algorithm of the mnSOM are reported; moreover, the results of applications of the mnSOM are presented.

1 Introduction

Kohonen's self-organizing map (SOM) performs a topology-preserving transformation from a higher-dimensional vector space to a lower one, which is usually two-dimensional, and generates a map that can display visually the similarity between vectors [1]. In addition, the units in a SOM can interpolate the intermediate vectors between the input vectors. Since a SOM has these features, it has been applied in various fields such as in medical treatment, informational-communication, control systems, and image and speech analysis. Despite that the SOM has been used in these various areas, objects that the SOM deals with are limited to vector data that are distributed in vector space. In the conventional SOM, it is difficult to generate a map of objects such as a set of time-series data or a set of manifolds. Therefore, it is necessary to propose a generalized SOM that can generate a map corresponding to various object types. Further, if it is possible to generate the intermediate objects self-organizationally by a generalized SOM, then SOMs will become more powerful tools.

Kohonen has described necessity of the generalization of the SOM [2], and proposed a self-organizing operator map (SOOM) as the generalization of the SOM [2]. In the SOOM, each nodal unit in the SOM is replaced to a linear operator such as an AR model in order to generate a map for dynamic signals. In other words, the network structure of the SOOM is equal to a modular network in which a module unit is composed of a linear operator. Kohonen tried to derive a general principle of the SOM from the SOOM. However, we believe that it can't truly be described as a generalization of the SOM since it is written in respect to theory limited to the SOOM, each module of which is a linear operator.

Recently, we proposed a modular network SOM (mnSOM) in which each nodal unit in the conventional SOM is replaced by a module such as a neural network [3]. The module of the mnSOM can be freely designed to correspond to the objects that are performed topology-preserving transformation to a map. For example, when the module of the mnSOM is composed of multi-layer perceptrons (MLP), which represent a nonlinear function, then the mnSOM generates a map in function space [4]. Moreover, when the module is designed as recurrent neural network that represents a function of a dynamical system, then the map generated by the mnSOM gives the interrelationships between the functions of dynamical systems [5]. Therefore, it is considered that the mnSOM has the characteristics of an extension and generalization of the SOM of Kohonen; moreover, it is expected that the mnSOM will become a fundamental technology for neural network models as well as expanding the fields to which an SOM as a generalized SOM can be applied.

In our past studies, the theory and the algorithm of our mnSOM based on the MLP module (MLP-mnSOM), the theoretical framework of which is simple, were established to prove that the mnSOM behaves as a generalized SOM. Moreover, not only the MLP-mnSOM but also mnSOMs based on various kinds of modules have been applied to a variety of research topics. As a result, it has been proven that the characteristic of the maps of mnSOMs and SOMs are equal. This suggests that our mnSOM is a generalization of an SOM. Further, it has been noted that the mnSOM has characteristics of both supervised and unsupervised learning; that is, the mnSOM performs not only supervised learning corresponding to the given target signals, but also unsupervised learning in which intermediate objects are generated self-organizationally.

In this paper, the theory and the algorithm of the MLP-mnSOM are presented. Moreover, the characteristics of the maps in mnSOMs and SOMs, and the results from various research areas with various variations of mnSOMs are shown.

2 Theory and algorithm

2.1 Theory and framework

The mnSOM has the structure of a modular network, the modules of which are arranged in a lattice (Fig.1(a)). In this paper, each module of an mnSOM, called "MLP-mnSOM", is composed of a multi-layer perceptron (MLP) that represents a nonlinear function. The MLP-mnSOM generates a map that presents similarity-relationships between functions. In other words, the neighboring modules in a mnSOM acquire similar functions through training, while distant modules acquire different functions.

Fig.1(b) shows the framework of the MLP-mnSOM. Suppose that there are I systems (static), in which input-output characteristics are defined with functions $f_i(\cdot)$ $i = 1, ..., I$ and which have d_{in} inputs and d_{out} outputs. In addition, suppose that input-output vector datasets $D_i = \{(\boldsymbol{x}_{ij}, \boldsymbol{y}_{ij})\}$ $j = 1, ..., J$ are observed

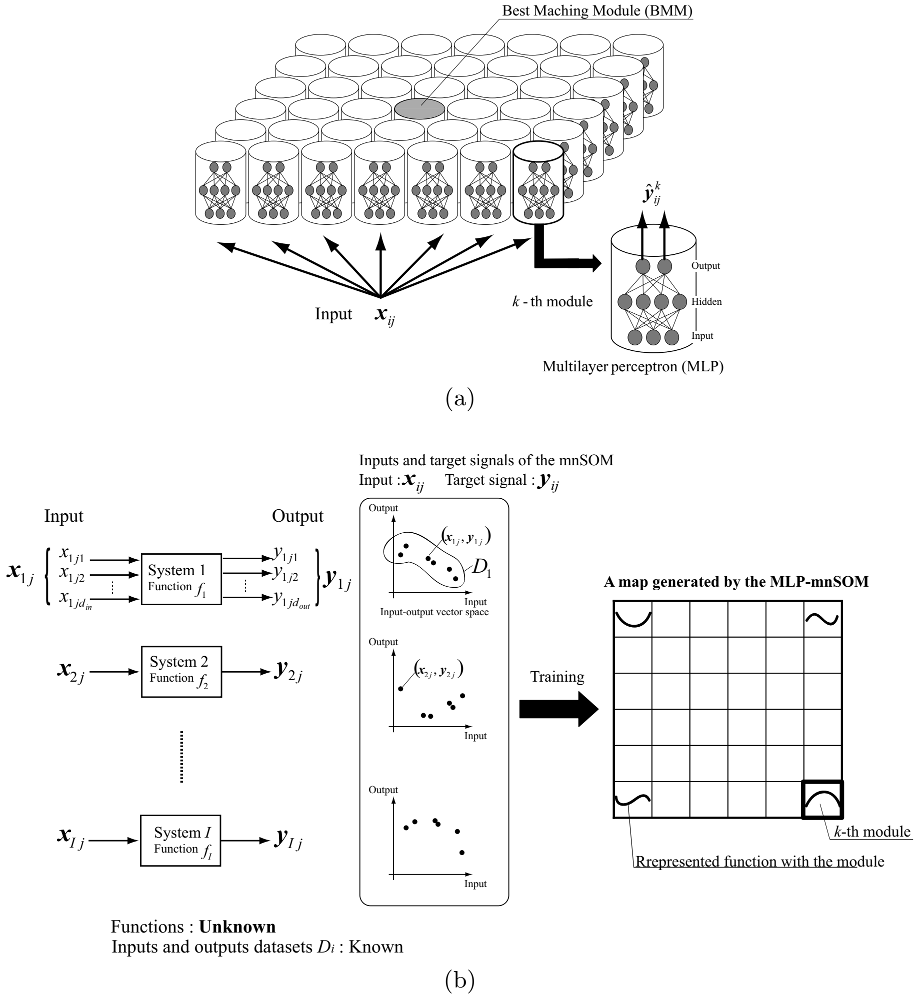


Fig. 1. (a)The architecture of an MLP-mnSOM. (b)The framework of an MLP-mnSOM.

from the systems, so that $\boldsymbol{y}_{ij} = f_i(\boldsymbol{x}_{ij})$. Known information is only the input-output vector datasets D_i. The functions $f_i(\cdot)$ of the systems are unknown. While, the mnSOM is composed of M MLP modules, each of which has d_{in} input-layer units, d_{hidden} hidden-layer units and d_{out} output-layer units. In these conditions, the MLP-mnSOM simultaneously executes the following tasks .

(1) To identify function $f_i(\cdot)$ from datasets D_i $\;i = 1, ...I$
(2) To generate a map based on the similarity-measures between functions.

(1) means that the functions of the systems are expected from the learning of the Best Matching Module (BMM). Here, a BMM means a module in which

output errors are minimized to the desired outputs of the dataset of a system. (2) means that the i-th and the i'-th systems have similar characteristics; Thus, two corresponding BMMs are near to each other by their positions in the lattice. Here, the similarity-measure $L^2(g, f)$ between functions $g(\boldsymbol{x})$ and $f(\boldsymbol{x})$ is defined as follows :

$$L^2(g, f) = \int \|g(\boldsymbol{x} - f(\boldsymbol{x})\|^2 p(\boldsymbol{x}) d\boldsymbol{x}. \tag{1}$$

$p(\boldsymbol{x})$ denotes the probability function of $\boldsymbol{x}$. Moreover, the modules between the BMMs of the i-th and the i'-th systems become "intermediate systems" by interpolation. These two tasks are processed in parallel.

2.2 Algorithm

The algorithm for the mnSOM consists of four processes: (1)*evaluative process*, (2)*competitive process*, (3)*cooperative process*, (4)*adaptive process*. In this paper, the algorithm of the MLP-mnSOM is shown.

(1)*evaluative process*

The outputs of all modules are evaluated for each input-output data vector pair. Suppose that an input data vector $\boldsymbol{x}_{ij}$ is picked up, then the output of the k-th modules $\tilde{\boldsymbol{y}}_{ij}^k$ is calculated for that input. This calculation process is repeated for $k = 1, ..., M$ using the same input $\boldsymbol{x}_{ij}$. After evaluating all outputs for all inputs, the errors of all modules for each data class are then evaluated. Now let E_i^k be the error of the k-th module for the dataset from i-th system, *i.e.*,

$$E_i^k = \frac{1}{J} \sum_{j=1}^{J} \|\tilde{\boldsymbol{y}}_{ij}^k - \boldsymbol{y}_{ij}\|^2 = \frac{1}{J} \sum_{j=1}^{J} \|g^k(\boldsymbol{x}_{ij}) - f_i(\boldsymbol{x}_{ij})\|^2. \tag{2}$$

If J is large enough, then the distance between the k-th module and the i-th system in the function space is approximated by the error E_i^k as follows :

$$L^2(g^k, f_i) = \int \|g^k(\boldsymbol{x}) - f_i(\boldsymbol{x})\|^2 p_i(\boldsymbol{x}) d\boldsymbol{x} \simeq E_i^k. \tag{3}$$

In this paper it is assumed that $\{p_i(\boldsymbol{x})\}$ for $i = 1, ..., I$ are approximately the same as $p(\boldsymbol{x})$, due to normalization of the data distribution for each class.

(2)*competitive process*

The module which reproduces $\{\boldsymbol{y}_{ij}\}$ best is defined as the BMM, *i.e.* the winner module for the i-th system. Thus, let k_i^* be the module number of the BMM for the i-th system, then k_i^* is defined as :

$$k_i^* = \arg\min_k E_i^k. \tag{4}$$

(3) *cooperative process*

The learning rate of each module is calculated by using the neighborhood function. Usually, a BMM and its neighbor modules gain larger learning rates than other modules. Let $\psi_i^k(T)$ denote the learning rate of the k-th module for the i-th system at learning time T. Then $\psi_i^k(T)$ is given by :

$$\psi_i^k = \frac{h(l(k, k_i^*); T}{\sum_{i'}^{I} h(l(k, k_{i'}^*); T)} \tag{5}$$

$$h(l; T) = \exp\left[-\frac{l^2}{2\sigma^2(T)}\right]. \tag{6}$$

Here, $l(k, k_i^*)$ expresses the distance between the k-th module and the BMM for the i-th system in the map space, *i.e.* the distance on the lattice of the mnSOM. $h(l; T)$ is a neighborhood function which shrinks with the calculation time T. Moreover, $\sigma^2(T)$ means the width of the neighborhood function $h(l; T)$ at the calculation time T. The $\sigma^2(T)$ is monotonously decreased at time T as:

$$\sigma^2(T) = \sigma_{min} + (\sigma_{max} - \sigma_{min}) \exp\left[-\frac{T}{\tau}\right] \tag{7}$$

σ_{max}, σ_{min} and τ are constants.

(4) *adaptive process*

All modules are trained by the backpropagation learning algorithm as:

$$\Delta \boldsymbol{w}^k = -\eta \sum_{i=1}^{I} \psi_i^k \frac{\partial E_i^k}{\partial \boldsymbol{w}^k} = -\eta \frac{\partial E^k}{\partial \boldsymbol{w}^k}. \tag{8}$$

Here, $\boldsymbol{w}^k$ denotes the weight vector of the k-th MLP module and $E^k = \sum_i \psi_i^k E_i^k$. Note that E^k has a global minimum point at which $g^k(\boldsymbol{x})$ satisfies:

$$g^k(\boldsymbol{x}) = \sum_{i=1}^{I} \psi_i^k f_i(\boldsymbol{x}). \tag{9}$$

Therefore, $g^k(\boldsymbol{x})$ is updated so as to converge to the interior division of $\{f_i(\boldsymbol{x})\}$ with the weights $\{\psi_i^k\}$. During training MLPs, each input vector $\boldsymbol{x}_{ij}$ is presented one by one as the input, and the corresponding output $\boldsymbol{y}_{ij}$ is presented as the desired signal.

3 Computer simulation

This section presents the results of several simulations: the maps of a family of cubic functions with MLP-mnSOM, and the results of various research topics using various variations of mnSOM.

3.1 Maps of cubic functions with MLP-mnSOM

First, the MLP-mnSOM generated the maps corresponding to the datasets as observed from the systems, in which the input-output characteristics were defined by the cubic functions: $y = ax^3 + bx^2 + cx$. The simulations were made under two different conditions. In the first case (simulation 1), there were a small number of datasets ($I = 6$) with a large number of data samples($J = 200$), whereas in the second condition (simulation 2) there was a large number of datasets ($I = 126$) with a small number of data samples($J = 8$). Each dataset $D_i = \{(x_{ij}, y_{ij})\}$ was sampled randomly, with the probability density function of $p(x)$ distributed uniformly between $[-1, +1]$. In addition, Gaussian white noise was added to $\{y_{ij}\}$ (standard deviation of noise $\sigma_{noise} = 0.04$). Figs.2 and 3(a) show examples of the datasets in simulations 1 and 2. It is easy to identify individual functions from each of the datasets in Fig.2(a); whereas, in Fig.3(a) the identifying of individual functions is difficult. However, it is considered that the interpolating between datasets by the cooperative processes of the MLP-mnSOM facilitates the identification of the true functions. The MLP module has three layers with one input, and eight hidden and one output units. Other details are presented in Table 1.

Figs.2 and 3 (b) show the results of simulations 1 and 2, respectively. The curve depicted in each box represents the function acquired by the corresponding module after training. The MLP-mnSOM generated similar maps in both cases. The neighbor modules acquired similar function forms, and the modules in the corner show opposite functions. The modules indicated by thick frames in Fig.2 (b) are the BMMs of the given six datasets. All other functions acquired by the rest of the modules were interpolated by the mnSOM in order to make a continuous map. Also, in simulation 2 (Fig.3(b)), the mnSOM succeeded in generating a map of cubic functions, despite there being only a small number of data samples. These results from simulations 1 and 2 suggest that the identification of functions was performed with not only with the supervised learning in each module, but also with unsupervised learning.

Incidentally, it is important that the essences between the maps generated by the mnSOM and the SOM are equal. If the essence of the map in the SOM is lost by replacing each unit in the SOM with a module, then it can not be confidently said that the mnSOM is the generalization of the SOM. To investigate the essence of the map in the mnSOM, we considered a case in which an orthonormal functional expansion is employed for vectorization. Thus, let $\{P_i(\cdot)\}$ be a set of orthonormal functions. Then the function $f(\cdot)$ is transformed to a coefficient vector $\boldsymbol{a} = (a_0, ..., a_n)$, where $f(x) = a_1 P_1(x) + a_2 P_2(x) + ... + a_n P_n(x)$.

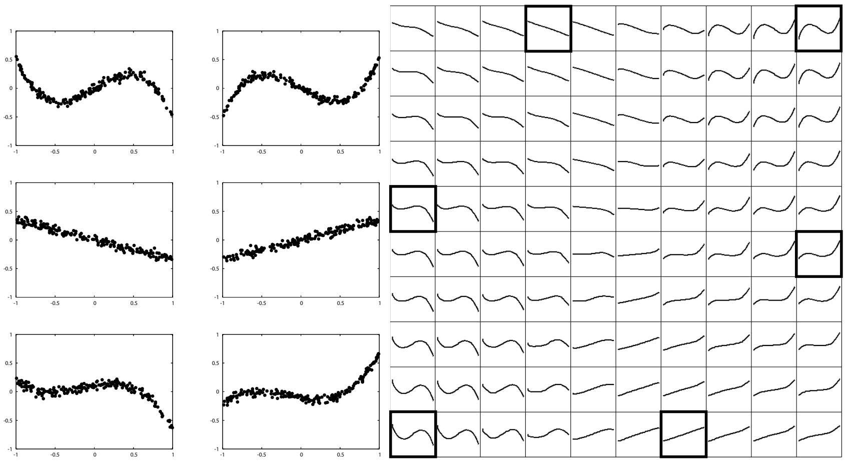
Terms higher than the n-th order are assumed to be negligible. Under this condition, the distance L_f in the function space is identical to that in the coefficient vector spaceL_v as follows :

$$L_f^2(f_i, g^k) = (a_{i1} - b_{k1})^2 + ... + (a_{in} - b^{kn})^2 = L_v^2(\boldsymbol{a}_i, \boldsymbol{b}^k). \tag{10}$$

Here $\boldsymbol{a}_i$ and $\boldsymbol{b}^k$ denote the coefficient vectors of the given i-th function and the k-th module of the mnSOM, respectively. In this situation, the mnSOM and

Table 1. Experimental conditions

Input layer	1	Map size K	$100(10 \times 10)$
Hidden layer	8	σ_0	10.0
Output layer	1	σ_∞ (simulation 1)	2.0
Learning rate η	0.05	σ_∞ (simulation 2)	1.0
		τ	300



(a) (b)

Fig. 2. (a) Training datasets sampled from the six cubic functions in simulation 1. (b) Map of cubic functions generated by the MLP-mnSOM in simulation 1.

SOM should produce the same results since the learning algorithm is identical for the two types, the SOM and the mnSOM. Therefore, the mnSOM and SOM share the same essences of the map.

In simulation 1, six functions obtained the coefficient vectors $\{\boldsymbol{a}_i\} = \{(a_{i1}, a_{i2}, a_{i3})\}$ by orthogonal expansion (In this paper, Legendre expansions are used). The map generated by a SOM for $\{\boldsymbol{a}_i\}$ is shown in Fig.4 (a). The figure shows the position of each reference vector in the coefficient vector space. The map generated by the mnSOM is shown in Fig.4 (b). The figure shows the map of the coefficient vectors $\boldsymbol{b}^k = \{(b^{k1}, b^{k2}, b^{k3})\}$ of the Legendre polynomial corresponding to the functions acquired with modules. The results in Fig.4(a) and (b) are roughly equal. Since the accuracy of the function approximation in the MLP is low, distortion is caused in the map (Fig.4(b)).

3.2 Applications of the mnSOM

In our past studies, we have applied the mnSOM to various research topics (Fig.5). The designs of modules in the mnSOMs are different in individual appli-

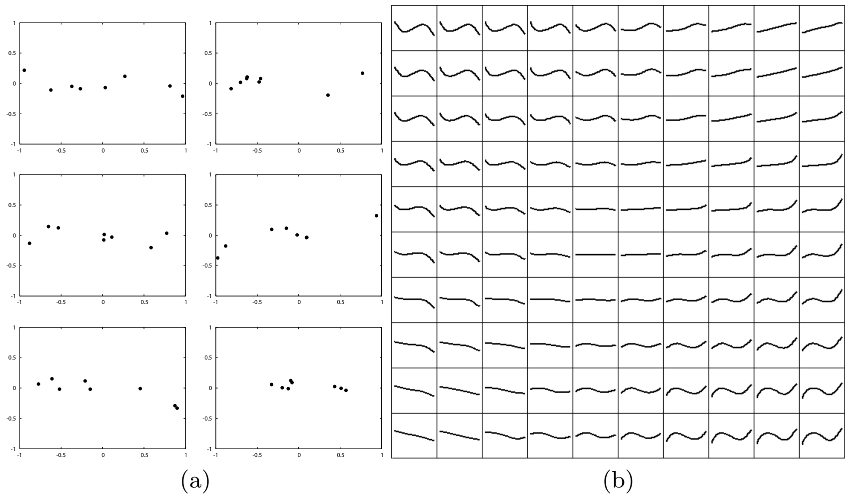


Fig. 3. (a) Example of training datasets in simulation 2. (b) Map of cubic functions generated by the MLP-mnSOM in simulation 2.

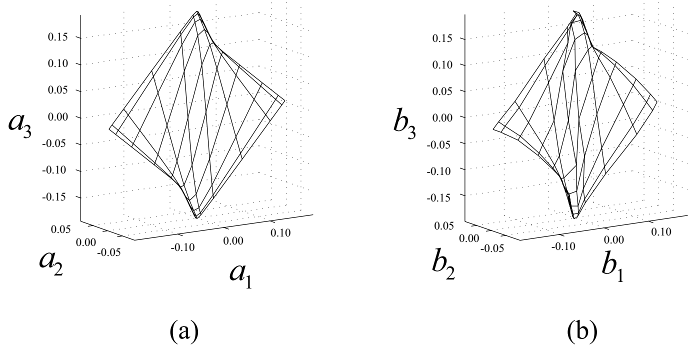


Fig. 4. Map of cubic functions plotted in the coefficient vector space of the Legendre expansion. (a) Map generated by the SOM. (b)Map generated by the mnSOM.

cations; whereas, the architecture and the algorithm are same in all applications. The results of these applications are described later.

Fig.5(a) shows "Map of weather in the Kyushu area of Japan" by the mnSOM in which each module is composed of a neural network that predicts the dynamics of weather. In this simulation, the mnSOM was merely given the weather datasets of nine cities in the Kyushu area of Japan, despite that the mnSOM represented the relationship of the geographic position of each city in Kyushu on the map.

Fig.5(b) shows "Map of damped oscillators". In this simulation, each module in the mnSOM is composed of the recurrent neural network that represents a function of a dynamical system. The mnSOM is merely given the datasets obtained from 9 damped oscillators, in which the individual input-output characteristics are different. Each module of the mnSOM acquired the characteristics of the 9 damped oscillators by training; moreover, the intermediate characteristics between the 9 damped oscillators were also acquired.

Fig.5(c) shows "Map of periodical waveforms" generated by the mnSOM based on a five layer autoassociative neural network module. This map is generated by giving several periodical waveforms to the mnSOM and training it on them. As the result, a map on which waveforms and frequencies are divided was generated.

Fig.5(d) shows "Map of 3D objects". In this simulation, sets of 2-dimensional images are merely given to the mnSOM. Despite that the mnSOM is not taught the method for rebuilding the solid from 2-dimensional images, the mnSOM generated a map of 3D objects.

4 Concluding remarks

We have presented the theory and the algorithm for our mnSOM. From the results, we proved that the characteristics of the maps of mnSOMs and SOMs are equal. Therefore, it is considered that our mnSOM is the generalization of the SOM of Kohonen. Additionally, it is expected that the mnSOM will become a fundamental technology of neural network models, since the mnSOM has characteristics of both supervised and unsupervised learning. The mnSOM can be applied to various research fields in which it is too difficult to apply conventional means. We expect to be able to use our mnSOM for extensive studies. A specific theory for a generalized SOM is explained by Furukawa.

Acknowledgments

This work was supported by a COE program (center #J19) granted to Kyushu Institute of Technology by MEXT of Japan. This work was also partially supported by a Grant-in-Aid for Scientific Research (C) granted by MEXT of Japan. This work was also supported by a The Okawa Foundation for Information and Telecommunications.

References

1. Kohonen, T.: Self-organizing maps. 3rd ed. Springer-Verlag, berlin Heidelberg New York (2003)
2. Kohonen, T.: Generalization of the self-organizing map. Proc. of International Joint Conference on Neural Networks (1993) 457–462
3. Tokunaga, K., Furukawa, T., Yasui, S.: Modular network SOM: Extension of SOM to the realm of function space. Proc. of Workshop on Self-Organizing Maps (2003) 173–178

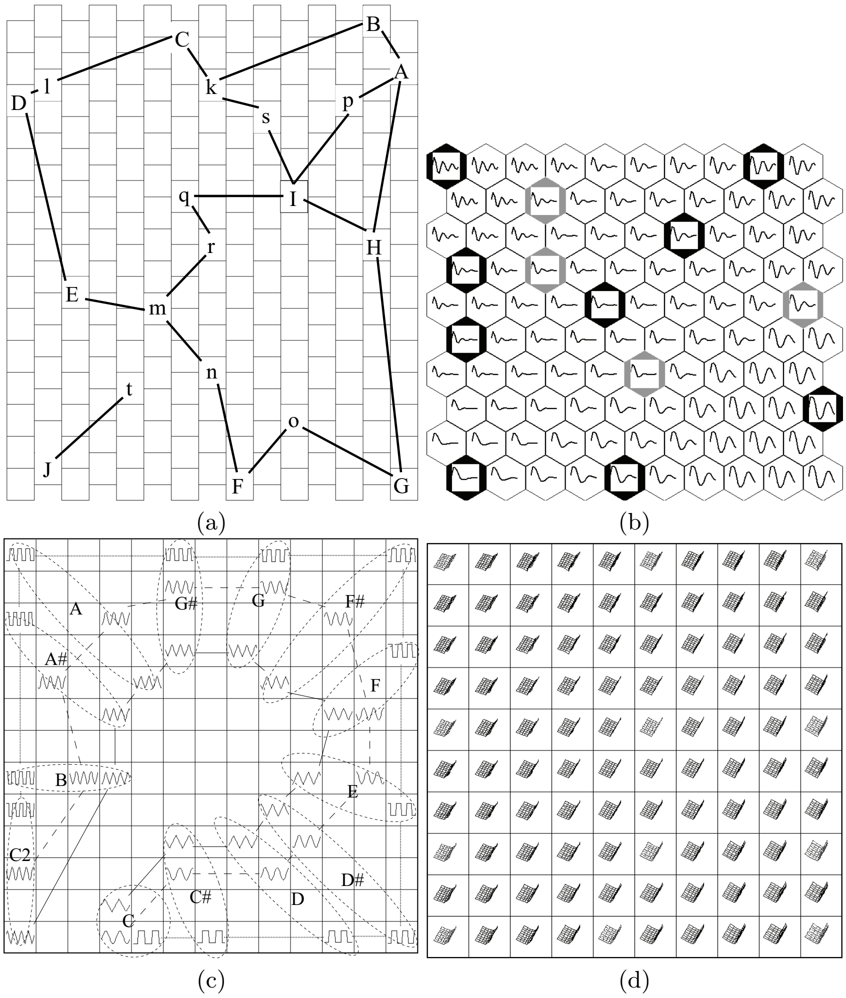


Fig. 5. Results of applications in the mnSOM. (a)Map of weather in the Kyushu area of Japan. (b)Map of damped oscillators. (c)Map of periodical waveforms. (d)Map of 3D objects.

4. Furukawa., T., Tokunaga, K., Morishita, K., Yasui, S.: Modular network SOM(mnSOM): From vector space to function space. Proc. of International Joint Conference on Neural Networks (2005) 1581–1586
5. Kaneko, S., Tokunaga, K., Furukawa, T.: Modular network SOM: The architecture, the algorithm and applications to nonlinear dynamical systems. Proc. of Workshop on Self-Organizing Maps (2005) 537–544