2004 International Symposium on Nonlinear
Theory and its Applications (NOLTA2004)
Fukuoka, Japan, Nov. 29 - Dec. 3, 2004

Generalized Self-Organizing Maps (mnSOM) for Dealing with Dynamical Systems

Tetsuo Furukawa†, Kazuhiro Tokunaga†, Syuji Kaneko†, Kenji Kimotsuki† and Syozo Yasui†

†Department of Brain Science and Engineering, Kyushu Institute of Technology
2-4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196 Japan
Email: furukawa@brain.kyutech.ac.jp

Abstract—The modular network SOM (mnSOM) is a generalization of Kohonen's Self-Organizing Map (SOM) in which each nodal unit is replaced by a function module instead of a weight vector. The mnSOM has expanded the application area of SOM from those involving vectorized data to those dealing with more general classes of dataset relevant to functions, systems, manifolds and so on. By employing a recurrent neural network (RNN) as the function module, the mnSOM acquires an ability to deal with dynamical systems. The RNN-module-mnSOM learns the dynamics from a set of time series data in parallel with generating a feature map which describes the underlying models as well as their relationships. Thus, the mnSOM works as systems identifier and systems classifier at the same time. This paper overviews the mnSOM and then reports application results to some linear- and nonlinear-dynamical phenomena such as damped oscillation and chaotic behavior of a neuron model.

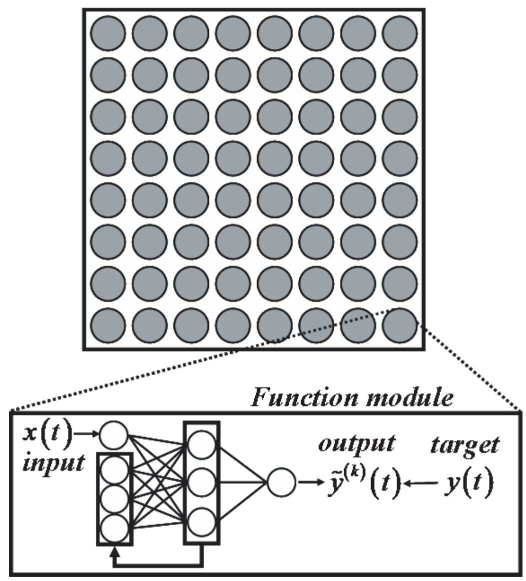

Figure 1: The architecture of the mnSOM.

1. Introduction

Difficulty of grouping and classifying a set of time-developing phenomena is likely due to the fact that direct comparisons between observed signal waveforms usually provide only partial clues. A better classification requires extraction of the essence underlying the dataset, i.e., identifying the dynamics in terms of mathematical models, and comparing each other's. In that sense, a good classifier of time series data should be a good system identifier, and this requirement is likely to make the task harder. On the other hand, our brain has an excellent ability to feel both resemblance and difference between time series. For example, it is not difficult for us to feel a common feature of Bach's melodies even if the temporal changes of musical notes look quite different, and there is no room to confuse a Beatles' melody. Such a dynamical information processing might be one of key functions unique to our brain.

In this paper, an artificial neural network technology called *modular network Self-Organizing Map* (mnSOM) is introduced as an identifier and a classifier of dynamical systems[1]. The mnSOM proposed by the present authors is based on Kohonen's Self-Organizing Map (SOM) which transforms a set of data vectors from a higher-order space to a lower-order one called 'feature map'[2]. Such a map preserves the topology of data distribution in the original space; therefore, SOM reduces the data dimension in such a manner that it minimizes loss of the feature information. Another advantage of SOM is that the learning is an unsupervised one, which involves Hebbian learning as well as competitive and cooperative interactions between its vector units.

The mnSOM is a generalization of SOM in which each vector unit is replaced by a function module such as a neural network. By choosing an appropriate function module according to the task, the mnSOM deals with not merely vector data but also deals with functions, systems and manifolds. For example, by employing a multi-layer perceptron (MLP) as the function modules, the mnSOM learns the input-output relations as a set of functions, and simultaneously the mnSOM generates the feature map representing them. In this sense, the mnSOM works as a SOM in function space rather than vector space. If recurrent neural network modules are used, then the mnSOM learns the underlying dynamics from time series dataset, and at the same time it tells the topological relationships between the dynamical systems on the feature map. Because of these aspects, the mnSOM can be regarded both as a supervised system identifier and as an unsupervised system classifier. Therefore, the mnSOM is expected to be a powerful tool for classifying time developing phenomena in a manner similar to what our brain does.

2. Architecture and Algorithm of mnSOM

The architecture of the mnSOM is shown in Fig.1. The mnSOM consists of an array of function modules on a lattice. The lattice, whose dimension is usually one or two for the sake of visualization, represents the coordinate of the feature map. For applying the mnSOM to dynamical systems, a recurrent neural network (RNN) such as the Elman network is appropriate as the module type. Now let us assume that there are M dynamical systems whose characteristics are unknown. The first task of the mnSOM is to identify the system dynamics from the observed input-output time series data. Since one RNN module learns one of the systems for its dynamics, the mnSOM which has N modules ($N > M$) is expected to reconstruct all systems. The second task is to map the M systems onto the feature map. It means that if the i-th system and the i'-th system have similar characteristics, then the RNN modules specialized to those two systems are located at near positions on the lattice.

The algorithm of the mnSOM consists of four processes: *evaluative process*, *competitive process*, *cooperative process*, and *adaptive process*. Let us consider the case in which $\{(x_i(t), y_i(t))\}$ ($i = 1 \ldots M$) is a set of observed input-output signals of a dynamical system. *In the evaluative process*, $x_i(t)$ is entered to all modules as the input, and the corresponding outputs $\{\tilde{y}_i^{(k)}\}$ are evaluated. Here, k denotes the module number. *In the competitive process*, the module which reproduces $y_i(t)$ best is decided as the tentative winner for the i-th system. In the next *cooperative process*, the learning rates of the modules are calculated by using the neighborhood function. Usually, the winner and its neighbor modules gain larger learning rates than other modules. In the last *adaptive process*, all modules are trained by the backpropagation learning as follows:

$$\Delta \mathbf{w}^{(k)} = -\eta \sum_{i=1}^{M} \psi_i^{(k)} \frac{\partial E_i^{(k)}}{\partial \mathbf{w}^{(k)}} \tag{1}$$

Here, $\mathbf{w}^{(k)}$ is the connection weight matrix of the k-th module, and $E_i^{(k)}$ is the mean square error between the module output $\tilde{y}_i^{(k)}$ and the actual output $y_i(t)$ of the system. $\psi_i^{(k)}$ is the learning rate given by

$$\psi_i^{(k)} = \frac{\phi_i^{(k)}}{\sum_j \phi_j^{(k)}} \tag{2}$$

where $\phi_i^{(k)}$ is the neighborhood function given by

$$\phi_i^{(k)} = \exp\left[-\frac{{l_i^{(k)}}^2}{2\sigma(T)^2} \right]. \tag{3}$$

Here, $l_i^{(k)}$ denotes the distance between the winner module and the k-th module on the lattice. The preceding equation tells that the neighborhood function shrinks with the calculation time T.

The architecture of the RNN modules, especially the definition of input and output, should be designed carefully depending on the target systems and the task. Usually, $x(t)$ is the input and $y(t)$ is the desired output of the modules, however, better results are expected in some cases when both $x(t)$ and $y(t)$ are entered to the modules as the input and $\Delta y(t) = y(t + 1) - y(t)$ is given to the target output.

3. Simulation Results

3.1. Application to Linear Damped Oscillatory Systems

The linear damped oscillatory systems described by the following equation are considered.

$$M\ddot{y}(t) + f\dot{y}(t) + ky(t) = x(t) \tag{4}$$

While keeping the mass $M = 1$, thirteen different combination of parameter values for the spring constant k and the viscous friction rate f were considered, as shown in Fig.2(a) together with the corresponding impulse responses. Nine out of thirteen systems (1–9) were used for the training, while the remaining four systems (A–D) were used for the test after training. These systems were driven by the white-noise input $x(t)$ as the external force, and the corresponding outputs $y(t)$ were observed. The data given to the mnSOM modules were $x(t)$ as the input and $y(t)$ as the target. Therefore, the mnSOM was not taught as to what kind of impulse responses the systems had.

Fig.2(b) shows the simulation result. The curve depicted in each box represents the impulse response acquired by the corresponding module after the training. The dark and light gray modules are the winner for the training and test data, respectively. The topology in the parameter space (k, f) of Fig.2(a) is preserved on the map of Fig.2(b) for both training and test data. Moreover, the impulse response waveforms acquired by the modules changed continuously all over the map. This fact means that the mnSOM network not merely learned the nine training systems, but also captured the rest of ninety one systems which were not presented to the network at all. This suggests that the mnSOM has powerful interpolation and generalization abilities.

3.2. Reconstructing the Bifurcation Diagram of Logistic Map

In the case of linear-systems like damped oscillations, the system dynamics and the output waveforms are continuously altered in accordance with continuous shift of parameter values. As opposed to such linear situations, the output waveforms of a nonlinear system may change drastically at certain parameter values called bifurcation points.

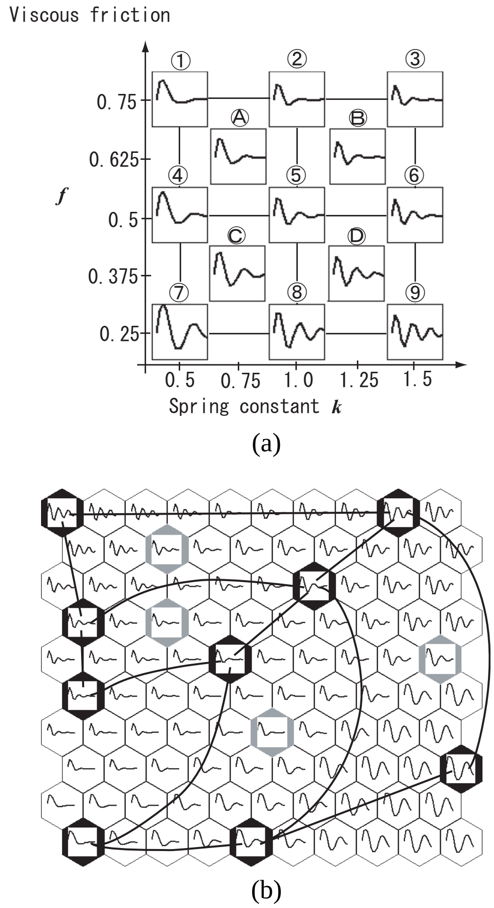

Figure 2: (a) A family of damped oscillatory systems for the simulation. (b) The feature map showing the impulse responses acquired by the mnSOM after the training.

To examine what happens to the mnSOM in such nonlinear cases, the discrete time series produced by the logistic map were considered. The relevant equation is described as follows:

$$x_{n+1} = -ax_n^2 + a - 1. \quad (5)$$

By changing the parameter a in the range of $1 < a < 2$, the characteristic of $\{x_n\}$ alters from a stable point to a chaotic state via periodical oscillation. The purpose of this simulation is to 'discover' such bifurcation structure by training the mnSOM. By giving several sets of time series whose values for a are different but are not told to the network, the mnSOM is expected (i) to identify the systems dynamics, (ii) to sort the all time series in the order of the parameter values, and (iii) to discover the waveforms of intermediate parameter values by interpolating between the values used for the training.

The Fig.3(a) shows the various time series generated by the logistic map with the value of parameter a ranging from 1.6 to 1.9. Five out of them (surrounded by thick frames) were given to the mnSOM as the training data. In this simulation, one dimensional feature map ($N = 16$) was used.

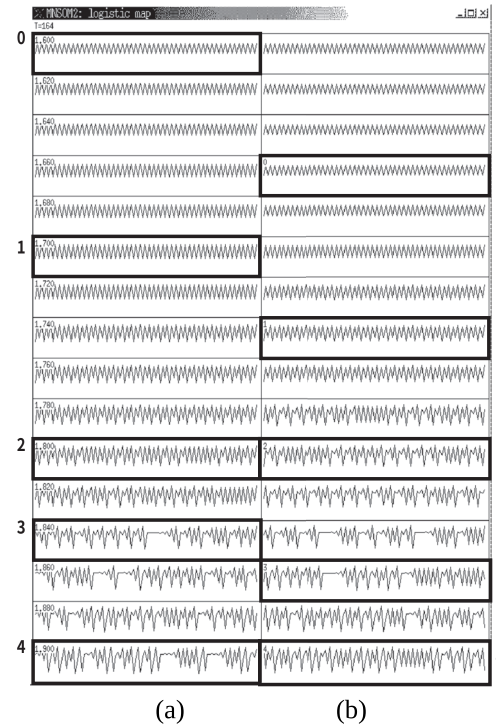

Figure 3: (a) Waveforms generated by the logistic map. Thick boxes for the training data. (b) Reproduced waveforms and the feature map generated by the mnSOM. Thick boxes for the winners of the training data.

After the training, the modules were made to generate time series auto-regressively by giving the initial value x_0. Thus, sixteen reproduced waveforms were obtained as presented in Fig.3(b). Modules of 1–4 are the winners for the corresponding training data of Fig.3(a). The mnSOM succeeded both in identifying and in sorting the systems in accordance with the order of the parameter values. Moreover, the mnSOM created the waveforms of intermediate parameter values not used for the training. In this respect, the feature map generated by the mnSOM can be regarded as an approximation of the bifurcation diagram.

3.3. Mapping the Parameter Space of the BVP Neuron Model

Finally, the BVP neuron model is considered. The differential equations are described as follows:

$$\dot{x} = c\left(x - \frac{x^3}{3} + y - Az\right) \quad (6)$$

$$\dot{y} = -\frac{x + by - a}{c} \quad (7)$$

Here, $x(t)$ represents the membrane potential and $y(t)$ means the refractoriness. The model was driven by a periodical pulse train $z(t)$, whose value was 0 or 1 according to a pulse was absent or present at time t. The intrinsic parameters a, b, c were set to be 0.7, 0.8 and 3.0, respectively.

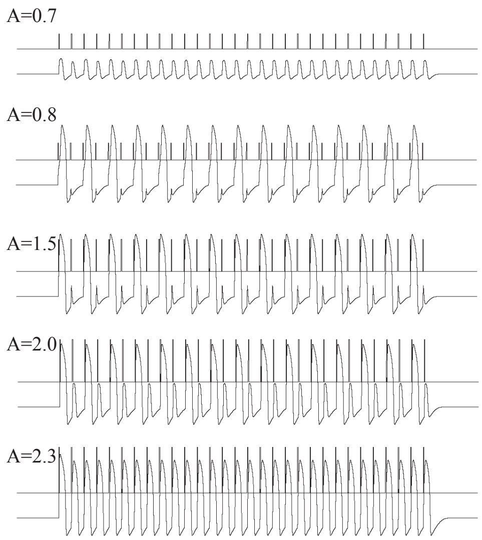

Figure 4: Waveforms produced by the BVP model with various pulse amplitudes.

In this simulation, the period and the duration of the input pulse were fixed at 10.0 and 0.3 respectively, whereas the pulse amplitude A was changed in order to generate various waveforms[3]. Fig.4 shows five waveforms of the BVP models generated by the parameter A = 0.7, 0.8, 1.5, 2.0 and 2.3, which were given to the mnSOM as the training data.

To learn the dynamics of the BVP model, the Elman type RNN was employed as the function modules of the mnSOM. Here, the inputs to the mnSOM were the membrane potential $x(t)$ and the pulse input $z(t)$, whereas the target output was $\Delta x(t) = x(t+\Delta t) - x(t)$. Since $z(t)$ was 0 or 1, the amplitude parameter A was hidden from the mnSOM. The refractoriness $y(t)$ was neither given to the network, therefore, the mnSOM should recreate the hidden variable $y(t)$ within the feedback units. The map was one-dimensional as in the case of the logistic map (N = 10).

The Fig.5 shows the feature map generated by the mnSOM together with the waveforms $\Delta x(t)$ reproduced by the modules. The mnSOM learned the dynamics of the BVP model correctly. At the same time, the mnSOM arranged the map in the order of values of the hidden parameter A.

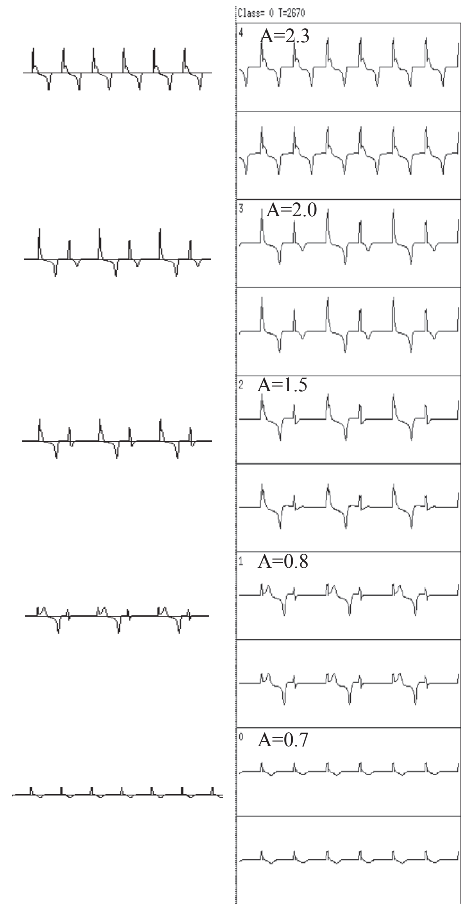

Figure 5: The feature map of BVP models acquired by the mnSOM. The waveforms on the left side are the training data.

4. Conclusion

The mnSOM consistently showed high abilities which can be summarized as follows: (i) The mnSOM is a parallel system identifier. It learns the dynamics of multi-systems simultaneously, including the case where the systems have hidden variables. (ii) The mnSOM is a hidden parameter finder. The mnSOM arranges the systems on the feature map in the order of hidden parameters. (iii) The mnSOM is a system interpolator. The mnSOM discovers intermediate systems between the trained ones, in the sense of interpolating hidden parameters. The mnSOM is not a simple interpolator of waveforms. It extracts the essence of time series, i.e., underlying dynamics, finds out hidden variables and parameters, and shows the relationships on the map between unknown systems being discovered. Examining these abilities for real nonlinear systems remains to be made.

References

[1] K. Tokunaga, T. Furukawa and S. Yasui, "Modular network SOM: Extension of SOM to the realm of function space," *Proc. of Workshop on Self-Organizing Maps (WSOM2003)*, pp.173–178, 2003.

[2] T. Kohonen, "The Self-Organizing Maps," Springer Verlag, 1995.

[3] S. Doi and S. Sato, "The global bifurcation structure of the BVP neuronal model driven by periodic pulse trains," *Mathematical Biosciences*, Vol.125, pp.229–250, 1995.