SOM of SOMs: An Extension of SOM from ‘Map’ to ‘Homotopy’

Tetsuo Furukawa

Kyushu Institute of Technology, Kitakyushu 808-0196, Japan,
furukawa@brain.kyutech.ac.jp,
WWW home page: http://www.brain.kyutech.ac.jp/˜furukawa

Abstract. This paper proposes an extension of an SOM called the “SOM of SOMs,” or SOM^2, in which objects to be mapped are self-organizing maps. In SOM^2, each nodal unit of a conventional SOM is replaced by a function module of SOM. Therefore, SOM^2 can be regarded as a variation of a modular network SOM (mnSOM). Since each child SOM module in SOM^2 is trained to represent an individual map, the parent map in SOM^2 generates a self-organizing map representing the continuous change of the child maps. Thus SOM^2 is an extension of an SOM that generates a ‘self-organizing homotopy’ rather than a map. This extension of an SOM is easily generalized to the case of SOM^n, such that “SOM^3 as SOM of SOM^{2}s”, corresponding to the n-th order of homotopy. This paper proposes a homotopy theory of SOM^2 with new simulation results.

1 Introduction

SOM^2 is an extension of Kohonen’s self-organizing map (SOM) aimed at generating a self-organizing map of a set of self-organizing maps [1, 2]. A SOM^2 consists of an assembly of basic (conventional) SOM modules arrayed on a lattice, which are the replacement of the reference vectors of the basic SOM. Thus SOM^2 is called the ‘SOM of SOMs’, the name may sound a bit eccentric, perplexing or even curious. However, despite its strange name, SOM^2 is a straightforward extension of a conventional SOM and is a powerful tool based on a sound mathematical theory.

Since a basic SOM represents a map from a high-dimensional data space to a low-dimensional feature one, the actual task of SOM^2 is to represent the continuous change of those maps, i.e., a homotopy. Thus, SOM^2 is an extension from a “self-organizing map” to a “self-organizing homotopy”. When a group of datasets is given, SOM^2 approximates their distributions by using a set of *child SOMs*, and simultaneously the *parent SOM* generates a map of those child maps. If two distributions of datasets are comparatively similar (or different), then those two datasets are located at nearer (or further) positions in the parent map.

Such architecture is useful when a set of data vectors observed from the same object forms a corresponding manifold in the data space. A typical example is face classification for a set of 2-dimensional photographs. In this case, a set of photographs taken of a single person from various viewpoints forms a manifold that is unique to that person. Therefore, if there are n people, then one obtains n face image manifolds that can be classified by a SOM^2. This ability of SOM^2 has been shown previously [1].

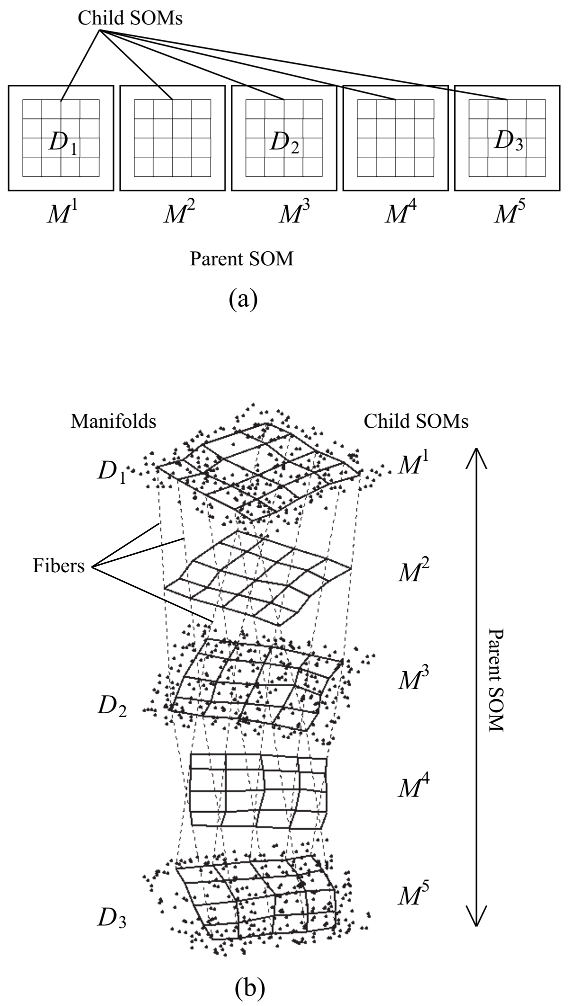


Fig. 1. (a) The architecture of SOM^2. In this case, the SOM^2 has 5×1 child SOMs, each of which has 5×5 reference vectors. Thus, the parent map space is 5×5, while the child map spaces are 5×5. (b) A simulation result when datasets $\{D_1, D_2, D_3\}$ are given. $\{M_1, \ldots, M_5\}$ are the child maps, and M_1, M_3, M_5 are the best matching maps (BMMs) of the datasets D_1, D_2, D_3, respectively. The dashed lines between child maps are called 'fibers', and these connect reference vectors with the same index of each child SOM.

The purpose of this paper is to propose a homotopy theory of SOM^2. In addition, a new application field of SOM^2 is also presented, namely, shape classification. The theory and an algorithm of SOM^2 are presented first, then followed by some simulation results.

2 Algorithm and Theory of SOM^2

2.1 What is SOM^2 ?

Like a conventional SOM, an SOM^2 has an arrayed structure of reference units on a lattice. In the case of the conventional one, each reference unit represents a vector in the data space, whereas the reference units in SOM^2 represents an SOM. Figure 1 (a) shows an architecture of SOM^2 that has 5×1 reference maps (i.e., child SOMs) $M^1, \ldots, M^5$, each of which has 5×5 reference vectors. In other words, the parent SOM has 5×1 map size, while the child SOM has 5×5 map size for each. SOM^2 is regarded as an SOM with modular structure, the modules of which are the basic SOMs. Therefore, SOM^2 is a variation of a modular network SOM (mnSOM), i.e., SOM-module-mnSOM [3–6].

To clarify the purpose of the algorithm, we take a typical case in which three families of sets $\{D_1, \ldots, D_3\}$ are given to the SOM^2 (Figure 1 (b)). In this case, the data vectors are distributed in 2-dimensional squares, which are topologically congruent but their positions and orientations are different in the 3-dimensional data space.

Figure 1 (b) also shows an actual simulation result. The child map that represents a class distribution best is called a 'winner map' or a 'best matching map' (BMM). In the case of Figure 1, the child maps M^1, M^3, M^5 became the BMMs of D_1, D_2, D_3, respectively. As a result, the three squares were arranged in the parent map in the desired order $D_1 \to D_2 \to D_3$. Child maps M^2 and M^4 that lost in the competition formed 'intermediate squares' to represent the continuous change of square positions and orientations. Thus, the homotopy was successfully organized in SOM^2.

The lines connecting the reference vectors with same indexes are the so-called 'fibers' (the dashed lines in Figure 1 (b)). As shown in the figure, the fibers connect the corresponding points of five child maps. Therefore, SOM^2 not only generates a map of objects, but it also finds out correspondences between given objects. In other words, SOM^2 can represent a set of data distributions by a bundle of fibers. Therefore, SOM^2 can be also regarded as an extension of an SOM that represents a "fiber bundle" rather than a manifold.

The task of SOM^2 described above can be summarized as follows. (i) For a given family of sets $\{D_1, D_2, \ldots\}$, representing those distributions using child maps $\{M^1, M^2, \ldots\}$. (ii) Mapping those datasets in the parent map. (iii) Finding out corresponding points in objects and connecting them by fibers. These three tasks are conducted simultaneously.

2.2 Algorithm of SOM^2

The algorithm of SOM^2 is based on the batch learning algorithm of the conventional SOM, which can be described as follows.

$$\mathbf{w}^l(t+1) = \sum_i \alpha_i^l(t)\mathbf{x}_i \tag{1}$$

$\mathbf{w}^l$ and $\mathbf{x}_i$ denote the l-th reference vector and i-th data Vector, respectively, and α_i^l is determined by the neighborhood function. Here α_i^l is normalized so as to be $\sum_i \alpha_i^l = 1$.

Thus,

$$\alpha_i^l = \frac{\exp[-\|\xi_i^* - \xi^l\|^2/2\sigma^2(T)]}{\sum_{i'} \exp[-\|\xi_{i'}^* - \xi^l\|^2/2\sigma^2(T)]}. \tag{2}$$

Here ξ_i^* and ξ^k are the coordinates of the BMU of $\mathbf{x}_i$ and the l-th child SOM in the parent map space.

The algorithm of SOM^2 has been described in an earlier work, but here I derive it another way. Suppose that an SOM^2 has K child maps, each of which has L reference vectors. Let $\mathbf{w}^{k,l}$ denote the l-th reference vector of the k-th child map, and let $W^k = (\mathbf{w}^{k,1}, \ldots, \mathbf{w}^{k,L})$ be a vector obtained by joining all reference vectors belonging to the k-th child map. Thus W^k is the joint reference vector of the k-th child map M^k. By regarding these joint reference vectors of the child SOMs as the reference vectors of the parent SOM, the entire SOM^2 can be regarded as a conventional SOM with reference vectors $\{W^1, W^2, \ldots, W^K\}$.

Suppose $\{D_1, \ldots, D_I\}$ is a family of sets observed from I objects. We now suppose that we have another set of conventional SOMs, the reference vectors of which are $V_i = (\mathbf{v}_i^1, \mathbf{v}_i^2, \ldots, \mathbf{v}_i^L)$. Further suppose that V_i learns only the dataset D_i. Here let us call V_i a "class map of D_i", since V_i organizes a map specialized to the i-th class.

Under this condition, a naive algorithm for SOM^2 is to train$\{V_1, V_2, \ldots, V_I\}$ by regarding them as ordinary data vectors. Thus, the class maps are calculated in advance, and then the joint reference vector W^k is updated as

$$W^k(t+1) = \sum_i \alpha_i^k(t) V_i. \tag{3}$$

This naive version of SOM^2 algorithm indicates good suggestions for better solutions, though it has a fatal defect. (i) Usually, it is not easy to define the distance measure between two manifolds. This method provides the definition of the measure as the Euclidian distance between two joint reference vectors. It is a natural definition because the distance also means the sum of the lengths of the fibers between two manifolds. (ii) It is also easy to define the "median point" of a set of manifolds, which is given by the median point of the joint reference vectors. However, the fatal defect is that there are several equivalent solutions of reference vectors organized by an SOM, e.g., a map with rotated 180 degrees and a map turned over. Therefore, it is nonsense to measure the distance between two manifolds without matching the corresponding points. This means that it is necessary to ascertain good correspondences between manifolds, i.e., to determine the good "fibers" between child maps and class maps.

To resolve this problem, one needs to simultaneously estimate both child and class maps. In such a case, an expectation maximizing (EM) algorithm is available, i.e., the class and the child maps are reciprocally estimated. In the initial state, both class and child maps are set to random, and the tentative class map is estimated from the datasets. Then the child maps are updated using the tentative class map; after which the class maps are estimated from the BMMs.

Now suppose that the i-th dataset D_i is picked up at time t. Then the child map with the least quantization error becomes the BMM of D_i, and the joint reference vector of the BMM is supposed to be $W_i^*(t)$. Next the class map $\tilde{V}_i$ is estimated from the BMM

$W_i^*(t)$. Thus, substituting $W_i^*(t)$ to $\tilde{V}_i(t)$ as the initial state, and then updating $\tilde{V}_i(t)$ by using the batch learning SOM algorithm. Here batch learning is assumed to be executed in one step as follows.

$$\tilde{\mathbf{v}}_i^l(t) = \sum_j \beta_{i,j}^l(t)\mathbf{x}_{i,j} \tag{4}$$

Here $\beta_{i,j}^l$ is given by the normalized neighborhood function that determines how $\mathbf{x}_{i,j}$ affects $\tilde{\mathbf{v}}_i^l$, and it satisfies $\sum_j \beta_{i,j}^l = 1$. Now the estimated class maps are obtained, then the child maps are updated from these estimated class maps.

$$W^k(t+1) = \sum_i \alpha_i^k \tilde{V}_i(t) \tag{5}$$

This equation is equivalent to (2) with the exception that the class maps are tentatively estimated ones. By combining (3) and (4), we obtain

$$\mathbf{w}^{k,l}(t+1) = \sum_i \sum_j \alpha_i^k(t)\, \beta_{i,j}^l(t)\, \mathbf{x}_{i,j} \tag{6}$$

$$= \sum_i \alpha_i^k(t) \left\{ \sum_j \beta_{i,j}^l(t)\, \mathbf{x}_{i,j} \right\}. \tag{7}$$

This is the algorithm for SOM^2. Please note that the estimated class maps$\{\tilde{V}_i\}$ are not necessary to update child maps $\{W^k\}$ anymore, because they are just introduced derive the algorithm. This updated algorithm is iterated, so reducing the neighborhood size until both parent and child maps achieve a steady state.

The updated algorithm (6) has a recursive structure like a Russian doll. Therefore, it is easy to extend SOM^3, SOM^4, ..., by further nesting.

2.3 Theory of SOM^2

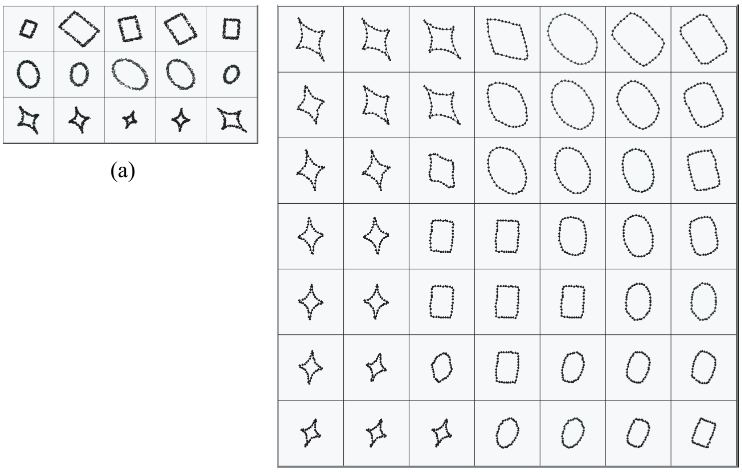
Here let us consider a theoretical aspect underlying SOM^2 from the point of view of topology. Let us assume that the data vectors dealt with by SOM^2 are distributed on a set of manifolds $\{U_i\}$ that are homotopic. Let us further suppose that the manifold U_i is obtained by a continuous surjective map φ_i from a base space $B = I^n$ ($n < m$, and n equals the dimension of child SOMs), as follows.

$$\varphi_i : B \to U_i \tag{8}$$

$$\xi \mapsto \mathbf{x}_i \tag{9}$$

Now let the nonlinear maps $\{\varphi_i\}$ be obtained by a continuous change of an intrinsic parameter θ. Thus,

$$\mathbf{x}_i = \Phi(\xi, \theta_i) = \varphi_i(\xi). \tag{10}$$



(a)

(b)

Fig. 2. A result of the shape classification task. (a) 15 contours are given to an SOM^2. The contours are represented by a set of dots that are the data vectors. (b) The map generated by the SOM^2. The SOM^2 successfully generated a map of contours, indicating the continuous changes of shapes, sizes and orientations.

Here $\Phi(\xi, \theta_i)$ is the homotopy. Under this condition, the distance between two manifolds U_1 and U_2 can be defined as follows.

$$L^2(U_1, U_2) \triangleq \int_{\xi \in B} \|\varphi_1(\xi) - \varphi_2(\xi)\|^2 p(\xi) d\xi \tag{11}$$

$$= \int_{\xi \in B} \|\Phi(\xi, \theta_1) - \Phi(\xi, \theta_2)\|^2 p(\xi) d\xi. \tag{12}$$

Here $p(\xi)$ gives the density of ξ. By employing this definition, the distance between a data class D_i and a child map W^k is approximated by

$$L^2(D_i, W^k) \simeq \frac{1}{L} \sum_{l=1}^{L} \left\| \mathbf{v}_i^l - \mathbf{w}^{k,l} \right\|^2 = \frac{1}{L} \left\| V_i - W^k \right\|^2, \tag{13}$$

and we obtain the algorithm of SOM^2, which is an unsupervised learning machine that ascertains the homotopy Φ from a family of set $\{D_1, \ldots, D_I\}$, and is also regarded as using a fiber bundle for representing data distributions, the sections of which represent the data classes.

3 Simulation Results

The ability of SOM^2 has been shown in cases of artificial manifolds and 2D images projected from 3D objects [1, 2]. SOM^2 has also been applied to facial image recognition [1].

Fig. 3. The map of alphabet generated by NG-SOM.

In this paper, another application field of SOM^2 is presented; namely, shape classification. It is known that a conventional SOM can be used for shape representation. In such cases, data vectors are assumed to be distributed on the surface of the object, and a conventional SOM learns the distribution of data vectors. If one has n objects, then one needs n conventional SOMs to represent those shapes. Consequently, an SOM^2 can make a map of these SOMs, i.e., a set of shapes. It is expected that objects with similar shapes are mapped nearer, while those with different shapes are mapped further in the SOM^2. The advantage of this method is that a user can directly deal with "shapes of objects" without employing any heuristic vectorization.

Figure 2 shows a result of a simulation of shape classification. In this case, 15 contours are given to an SOM^2. Each contour consists of a set of small dots that corresponds to the dataset. Thus, the i-th contour corresponds to the i-th dataset $D_i = \{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,J}\}$, and $\mathbf{x}_{i,j} = (x_{ij}, y_{ij})$ represents the coordinate of the j-th dot in the i-th contour. The child SOMs has a one-dimensional closed ring structure to represent a contour, and the parent SOM has 7×7 child SOMs. Figure 2 (b) shows a map generated by the SOM^2. The SOM^2 successfully generated a map of contours that shows continuous changes of shapes, sizes and orientations of the objects. An advantage of this method is that the result is robust to a small change of position or orientation of the contours.

Figure 3 is a tentative result of handwritten character classification. Since the topological structures of the characters are all different, neural gas networks are employed instead of child SOMs for this task; namely, an NG-SOM was used. The handwritten data were also represented by sets of small dots. In the case of Figure 3, 26 characters written by a person were given to the NG-SOM. After the map was generated, then another 9 sets of characters written by 9 people, i.e., $26 \times 9 = 234$ characters were given to the NG-SOM. The recognition rate was $93.2 \pm 6.7\%$ (mean±SD, $n = 9$). In this case, the training dataset has only 1 data for each alphabet, whereas the test dataset has 9 data for each alphabet with slightly different positions, shapes, sizes, and written by different people. Note that neither a heuristic preprocess nor an additional algorithm for feature extraction were used.

4 Conclusion

In this paper we have proposed an extension of an SOM called an SOM^2. Despite the eccentric impression given by the name of 'SOM of SOMs', SOM^2 is a straight forward extension of a conventional SOM from a 'map' to a 'homotopy'.

As a closing remark, I have some comments about SOM^2. First, some people may think that SOM^2 is a supervised algorithm, because it requires a labeled dataset. However, such an understanding is not realistic. The mapping objects of SOM^2 are distribution data vectors, and each distribution should be estimated by a set of data vectors. Each data vector in a conventional SOM case corresponds to each data distribution in an SOM^2.

Second, our aim is not to develop an alternative algorithm that supersedes the conventional one. The concept of SOM^2 tells us that we have a family of SOM^1 (the conventional SOM), SOM^2, SOM^3, $\ldots$, etc., and users can choose an appropriate order for the SOM^n family depending on their purpose. For some tasks an SOM^2 would be the best solution, and for others a conventional SOM would be appropriate. Therefore the idea of SOM^2 will further enlarge the application fields for SOMs.

Acknowledgement

This work was supported by a COE program (center #J19) granted by MEXT of Japan. This work was also partially supported by a Grant-in-Aid for Scientific Research (C) granted by MEXT of Japan.

References

1. Furukawa, T.: SOM^2 as "SOM of SOMs". Proc. of WSOM2005 (2005) 545-552
2. Furukawa, T.: SOM of SOMs: Self-organizing map which maps a group of self-organizing maps. Lecture Notes in computer Science, **3696** (2005) 391-396
3. Tokunaga, K., Furukawa, T., Yasui, S.: Modular network SOM: Extension of SOM to the realm of function space. Proc. of WSOM2003 (2003) 173–178
4. Furukawa, T., Tokunaga, K., Morishita, K., Yasui, S.: Modular network SOM (mnSOM): From vector space to function space. Proc. of IJCNN2005 (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 WSOM2005 (2005) 537-544
6. Furukawa, T.: Generalization of the Self-Organizing Map: From Artificial Neural Networks to Artificial Cortexes. Proc. of ICONIP2006 (2006)