A Simple Learning Algorithm for Growing Self-Organizing Maps and Its Application to the Skeletonization

Hiroki Sasamura
EECE Dept, HOSEI Univ, Tokyo, 184-8584, Japan
Email: sasamura@nonlinear.k.hosei.ac.jp

Toshimichi Saito
EECE Dept, HOSEI Univ, Tokyo, 184-8584, Japan
Email: tsaito@k.hosei.ac.jp

Abstract—This paper presents a simple learning algorithm for growing self-organizing maps (ab. SOMs) and consider its application to the skeletonization. In order to adapt the shape of the input data, the map can have partial tree and loop topology. In the algorithm, the map can grow and the topology can change based on occasional inspection of learning history of each cell and MST. If the control parameters are selected suitably, the algorithm can be applied effectively for skeletonization of Japanese characters.

I. INTRODUCTION

Self-organizing feature maps (ab.SOMs [1]) are known as typical unsupervised learning systems which can extract features from input data. The SOMs have variety of applications including speech recognition and data compression. In order to develop the functions of SOMs, it is important to control size and topology adaptively to the feature of the input data. Refs. [4]-[9] present interesting learning algorithms for growing and reducing structure of cells. If the control parameters are selected suitably in the algorithm, the classification functions can be improved and the algorithm can be applied to various problems including data visualization [4], vector quantization [4] [6], knowledge discovery [9], traveling salesperson problems [8] and skeletonization [13] [14].

This paper presents a simple learning algorithm for growing self-organizing map and consider its application to the skeletonization. In order to adapt the shape of the input data, the map can have partial tree and loop topology. In our algorithm, the map can grow and the topology can change based on occasional inspection of learning history of each cell and minimum spanning tree computation (ab. MST). The algorithm is applied to the skeletonization of Japanese characters. Basic experimental results suggest that the algorithm can realize suitable skeletonization if the control parameters are selected suitably. Such the skeletonization is hard without the growing structure and variable topology.

The skeletonization is usually used to extract important features in a shape of connected objects located in an image. They are used in applications such as character recognition, automatic inspection of circuit boards, and fingerprint identification [12] [13] [14]. For the skeletonization, some SOM-based methods are proposed [10] [11]. However, in the existing algorithms, the topology of the map is fixed and the initial state setting is complicated. Our algorithm may be developed into an efficient algorithm for practical and flexible applications.

II. ALGORITHM

Step 1: Initialization
Let \( t \) be the discrete learning time and let \( N(t) \) be the number of the cells at time \( t \). Let \( t = 0 \). We give initial conditions \( N(0), C_i(0) \) and \( w_i(0) \), where \( i \in \{1, \cdots, N(0)\} \).

Step 2: Input signal
Apply input signal \( x \in X \) according to some distribution.

Step 3: Determination of winner cell
Cell \( c \) is selected as a winner at time \( t \) if its synaptic vector \( w_c(t) \) is the closest to the input \( x \) at time \( t \).

\[
\|x(t) - w_c(t)\| = \min_i \|x(t) - w_i(t)\| \tag{1}
\]

where \( \| \cdot \| \) denotes the Euclidean distance.

Step 4: Update of synaptic vectors and counters
According to Equation (2), synaptic vectors of the winner and its neighbors are updated and the other synaptic vectors are preserved at time \( t \).

\[
w_i(t + 1) = \begin{cases} 
  w_i(t) + \alpha_i(x(t) - w_i(t)) & \text{if } i = c \\
  w_i(t) + \alpha_i(x(t) - w_i(t)) & \text{if } i \in N_c \\
  w_i(t) & \text{otherwise} 
\end{cases} \tag{2}
\]

Fig. 1. SOM with variable topology
Learning

No

Yes

Input

Insert a cell

Every \( T_{int} \) steps

Every \( T_{mst} \) steps

MST

Loop topology

where \( i \) is modulus \( N(t) \). The learning rate \( \alpha_c \) and \( \alpha_n \) are the 1st and 2nd control parameters of this algorithm. The signal counter of the winner is updated and the other signal counters preserve their values at time \( t \).

\[
C_i(t + 1) = \begin{cases} 
C_i(t) + 1 & \text{if } i = c \\
C_i(t) & \text{otherwise}
\end{cases}
\] (3)

**Step 5: Insertion of a cell**

Our algorithm inserts one new cell at every \( T_{int} \) learning times hereby the map can grow. \( T_{int} \) is the 3rd parameter of the algorithm. According to Equation (4), we find cell \( q \) which has the maximum counter.

\[
C_q(nT_{int}) \geq C_i(nT_{int}) \quad \text{for all } i.
\] (4)

If there exist plural maximum counter values we randomly select one of them. This is an inspection of the learning history. Then we find the farthest direct neighbor cell of \( q \). In Fig. 3 \( f \) is the cell. If Equation (5) is satisfied, a novel cell \( r \) is inserted between \( q \) and \( f \).

\[
\| w_q - w_f \| > \varepsilon_{ins} d_{ave}
\] (5)

where \( \varepsilon_{ins} \) is the 4th parameter, \( d_{ave} \) is average length of branches in the map. The synaptic vector of \( r \) is initialized as

\[
w_r = 0.5(w_q + w_f)
\] (6)

Counter values of \( q \) and \( r \) are re-assigned as

\[
C_q(t+1) = 0.5C_q(t), \quad C_r(t+1) = 0.5C_q(t).
\] (7)

After the insertion, the number of cells increases: \( N(t + 1) = N(t) + 1 \).

**Step 6: Variable topology**

Our map has variable topology at every \( T_{mst} \) learning times hereby the map topology can include partial tree and loop. The \( T_{mst} \) is the 5th (last) parameter of this algorithm. First, using Prim’s algorithm of the MST, we construct a tree topology as shown in Fig. 4.

Next, in order to make (sub)loop, we find the first and second nearest cells \( m_a \) and \( m_b \) from each cell \( m_i \). If \( m_a \) and \( m_b \) are not connected, we connect \( m_i \) and \( m_b \). In this manner we can make loop as shown in Fig. 5.

**Step 7: Termination**

Let \( t = t + 1 \). If \( t < T_{max} \) then go to step 2, otherwise the learning is terminated, where \( T_{max} \) is the maximum learning limit.

It should be noted that this algorithm has 5 parameters: \( \alpha_c, \alpha_n, \varepsilon_{ins}, T_{int}, T_{mst} \). If these parameters are selected suitably, this algorithm can realize the skeletonization as shown in the next section.

Ref. [4] presents a learning algorithm for networks with growing cell structure (ab. GCS). The GCS includes a complex subroutine for the network growing and the network.
topology is fixed. As compared with this, our algorithm is simpler and the networks topology can vary flexibly.

III. EXPERIMENT

In order to consider an algorithm to the skeletonization, we have performed two basic experiments. In the experiments, input signal in Step 2 is applied according to uniform distribution on the image.

Fig. 6 shows a Japanese character that is the first object. We have selected the parameters trial-and-errors and have succeed in the skeletonization from this character with tree topology.

Fig. 7 shows the second object that includes loop. For this complicated image, our algorithm has operated successfully.

$T_{mst}$ is an important principal parameter that controls frequency to modify the topology of the map. In order to consider dependence of this parameter, we investigate the skeletonization performance for various values of $T_{mst}$. Fig. 8 shows typical data. Based on the data, we have set $T_{mst} = 100$ in Fig. 7. Our simple algorithm have operate successfully in order to get the skeleton from the images, however, such skeletonization is hard without the variable topology. As shown in Fig. 9, the original SOM algorithm can not make the skeleton.

IV. CONCLUSION

We have presented a simple learning algorithm for growing self-organizing map with variable topology and have considered its application to the skeletonization. Performing basic experiments, we have clarified efficiency of the algorithm. Future problems include automatic adjustment of the parameters and statistical analysis of the learning process.

REFERENCES

Fig. 6. Skeletonization for Japanese character. \( T_{\text{max}} = 25000, N(0) = 3, \alpha_c = 0.25, \alpha_n = 0.01, T_{\text{int}} = 100, T_{\text{mst}} = 100, \varepsilon_{\text{ins}} = 0.94 \)

Fig. 7. Skeleton of shapes with loop. \( T_{\text{max}} = 20000, N(0) = 3, \alpha_c = 0.20, \alpha_n = 0.01, T_{\text{int}} = 100, T_{\text{mst}} = 100, \varepsilon_{\text{ins}} = 0.95 \)

Fig. 8. The dependence of the re-construction time \( T_{\text{mst}} \). \( T_{\text{max}} = 20000, N(0) = 3, \alpha_c = 0.20, \alpha_n = 0.01, T_{\text{int}} = 100, \varepsilon_{\text{ins}} = 0.95 \)

Fig. 9. Experiment using original SOM algorithm \( T_{\text{max}} = 20000, N(0) = 40, \alpha_c = 0.25, \alpha_n = 0.01 \)

790