Dense learning by high-dimensional self-organizing maps for interactive image segmentation

Hotaka Takizawa^{a,*}

^a University of Tsukuba, 1-1-1 Tennodai, Tsukuba, 305-8573, Japan

*Corresponding Author: takizawa@cs.tsukuba.ac.jp

Abstract

This paper proposes an interactive image segmentation method based on self-organizing maps (SOMs). In this paper, not only two dimensional SOMs but also higher dimensional SOMs are used for image segmentation. The proposed method was applied to actual images. The experimental results demonstrated that higher dimensional SOMs were able to achieve more accurate segmentation.

Keywords: Dense learning, high dimensional self-organizing maps, artificial neural network, high-density connections of units, interactive image segmentation

1. Introduction

The concept of a self-organizing map (SOM) was proposed by T.Kohonen^(1,2). The SOM is a type of artificial neural network that is composed of neural units arranged in a grid pattern. The neural units have reference vectors whose dimensions are the same as those of input vectors in a dataset. The SOM can perform un-supervised classification for the input vectors. The similar input vectors are mapped near to each other in the SOM.

SOMs have been used for data visualization⁽³⁾, image segmentation^(4,5), and so forth. Many of these studies used SOMs whose units were arranged in two dimensional grid pattern. Such two dimensional SOMs were useful to visualize trends in data distribution. In this paper, higher dimensional SOMs are used to segment gray-scale and color images interactively, and the segmentation accuracy of these SOMs are compared with each other.

Figure 1. Two, three, and higher dimensional SOMs.

2. High dimensional SOM

Let $U(\mathbf{m})$ denote a unit in an N dimensional SOM. $\mathbf{m} = (m_1, m_2, \dots, m_n, \dots, m_N)^T$ is an index to indicate the position of the unit in the SOM, where $n \in \{1, 2, \dots, N\}$ and $m_n \in \{1, 2, \dots, M_n\}$. If the SOM is a torus-type SOM⁽⁶⁾, $U(\dots, 1, \dots) = U(\dots, M_n + 1, \dots)$. Each unit has a D dimensional reference vector, $\mathbf{z}(\mathbf{m}) = (z_1(\mathbf{m}), z_2(\mathbf{m}), \dots, z_d(\mathbf{m}), \dots, z_D(\mathbf{m}))^T$, and also a label, $\gamma(\mathbf{m}) \in \mathbb{Z}$, for a segmented image region. Figure 1 depicts two, three, and higher dimensional SOMs. Circles represent units, and lines represent links to connect two adjacent units directly. Each unit in a higher dimensional SOM has more links. It means that higher-density connections are formed in a higher dimensional SOM. In this paper, learning based on high dimensional SOMs is called *dense learning*. The neighborhood of a unit is defined to be a set of neighboring units in a (hyper) cuboid composed of $L_1 \times L_2 \times \dots \times L_N$ units.

3. Interactive segmentation

In this paper, SOMs are used to segment original (color

or gray-scale) images into several regions on the basis of seed regions in seed images. The seed regions are interactively extracted in advance, and are given unique labels. The original images are segmented by extracting pixels that have similar features as the seed regions.

3.1 Self organization

An original image is raster-scanned by use of a patch composed of $D' \times D'$ pixels to create input vectors, $\mathbf{x}(k) = (x_1(k), x_2(k), \dots, x_d(k), \dots, x_D(k))^T$, where $D = D' \times D'$ and $k \in \{1, 2, \dots, K\}$. The central coordinate of the k -th patch is denoted by $c_p(k)$.

There are two learning algorithms for SOMs: online learning (OL) and batch learning (BL)⁽⁷⁾. The OL algorithm updates the reference vectors by entering the input vectors into a SOM in order, whereas the BL algorithm updates the reference vectors considering all the input vectors simultaneously. The OL algorithm has a risk of biased learning, and the BL algorithm is memory consuming. In this paper, the OL algorithm is used because many input vectors should be inputted to the SOM.

3.2 Posteriori learning

After self-organization, each $\mathbf{x}(k)$ is fed again into the SOM, and the best match unit $U(\mathbf{m}^*; k)$ is obtained for $\mathbf{x}(k)$. The label of the pixel at $c_p(k)$ in a seed image is set to $\gamma(\mathbf{m}^*; k)$.

3.3 Segmentation

An output image of the same size is created beforehand. Each $\mathbf{x}(k)$ is input again to the SOM, and the best match unit $U(\mathbf{m}^*; k)$ is obtained. The label $\gamma(\mathbf{m}^*; k)$ is set to the pixel at $c_p(k)$ in the output image.

4. Experiments

4.1 Conditions

Two, three, four, five, and six dimensional SOMs were used in this paper. The sizes of the SOMs ($M_1 \times \dots \times M_N$) and the neighborhoods ($L_1 \times \dots \times L_N$) were set as listed in Table 1. The numbers of units in the SOMs were set to be as equal to each other as possible, and the numbers of units in the neighborhoods were set in the same manner. The reference vectors were initialized by random values. In the OL algorithm, the sizes of neighborhoods were lineally decreased with increasing the iteration time, and the weights of the neighborhoods were exponentially decreased with

Table 1. The sizes of SOMs and neighborhoods (the numbers of units). “ N ” means the dimensions of the SOMs.

N	SOMs	Neighborhoods
2	181×181	57×57
3	$32 \times 32 \times 32$	$15 \times 15 \times 15$
4	$12 \times 13 \times 14 \times 15$	$7 \times 7 \times 7 \times 9$
5	$8 \times 8 \times 8 \times 8 \times 8$	$5 \times 5 \times 5 \times 5 \times 5$
6	$5 \times 5 \times 6 \times 6 \times 6 \times 6$	$3 \times 3 \times 3 \times 5 \times 5 \times 5$

increasing the iteration time.

4.2 Sample 1

Figure 2 shows sample and seed images whose sizes are 147×191 pixels. The sample image is composed of nine blocks, each of which has twenty five segments. The directions of the segments in the nine blocks are 0, 10, ..., and 80 degrees, respectively. The seed image includes nine seed regions that correspond to the nine blocks, respectively. The seed regions were manually extracted in advance, and were given unique labels ranging from 1 to 9. Blue, azure, ..., and red regions correspond to the labels, 1, 2, ..., and 9, respectively.

(a) Sample image.

(b) Seed image.

Figure 2. Sample and seed images.

Figure 3(a)-(e) show the segmentation results obtained by applying the two, three, four, five, and six dimensional SOMs to the sample and seed images, respectively. The iteration time in the OL algorithm was 5, and the sizes of patches were 15×15 pixels. The colors of pixels in the segmentation results correspond to those of the seed regions in Figure 2(b).

Figure 4 shows the truth image of the sample image, and Figure 5 shows relations between segmentation accuracy and the dimensions of the SOMs. The segmentation accuracy

Figure 3. Segmentation results of the 2, 3, 4, 5, and 6 dimensional SOMs.

Figure 4. Truth image for the sample image.

Figure 5. Relations between segmentation accuracy and the dimensions of SOMs.

is defined to be a ratio of the number of correctly segmented pixels to that of the pixels in the corresponding truth region. The colors of lines in Figure 5 correspond to those of the seed regions in Figure 2(b). A black line represents the mean accuracy. The calculation time was approximately 369 minutes.

4.3 Sample 2

Figure 6 shows another sample image and its seed image. The image sizes are 632×347 pixels. Two

(a) Sample image.

(b) Seed image.

Figure 6. Another sample image and its seed image.

(a) N=2.

(b) N=3

(c) N=4

(d) N=5

(e) N=6

Figure 7. Segmentation results of the 2, 3, 4, 5 and 6 dimensional SOMs.

Figure 8. Truth image for the sample image.

clothes are taken in the sample image. The main colors of the both clothes are red, but they have different texture patterns. Seed regions in Figure 6(b) were extracted manually in advance.

Figure 9. Relations between segmentation accuracy and the dimensions of SOMs.

Figure 7(a)-(e) show the segmentation results. The iteration time was 3, and the sizes of patches were 21×21 pixels. Figure 8 shows the truth image, and Figure 9 shows relations between segmentation accuracy and the dimensions of the SOMs. The calculation time was approximately 589 minutes.

5. Discussion

In the experiments, the numbers of units in the SOMs were set to be almost the same; nevertheless the higher dimensional SOMs were able to achieve more accurate segmentation in average. It implies that higher dimensional SOMs can produce more effective maps in the self-organizing process. In many studies, two dimensional SOMs

have been used, but it would be useful to consider to use higher dimensional SOMs as well.

In this paper, we investigated only up to six dimensional SOMs. In the future, we should investigate more than six dimensional SOMs.

The proposed SOMs have label components as well as reference vectors in the units. The study ⁽²⁾ introduced supervised SOMs whose reference vectors are composed of input (data) and output (label) components. We should investigate high dimensional SOMs composed of such fusion vectors.

6. Conclusion

This paper proposed an interactive image segmentation method based on high dimensional SOMs. The proposed method was applied to actual gray-scale and color images. The experimental results demonstrated that higher dimensional SOMs were able to achieve more accurate segmentation.

References

- (1) T. Kohonen, Self-organized formation of topologically correct feature maps, *Biological Cybernetics* 43 (1) , 59–69, 1982.
- (2) T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin Heidelberg, 2001.
- (3) Brian S. Penn, Using self-organizing maps to visualize high-dimensional data, *Computers & Geosciences*, Volume 31, Issue 5, 531-544, 2005,
- (4) Dongxiang Chi, Self-Organizing Map-Based Color Image Segmentation with k-Means Clustering and Saliency Map, *ISRN Signal Processing*, vol. 2011, Article ID 393891, 18 pages, 2011.
- (5) Jiang, Yuan, Chen, Ke-Jia, Zhou, Zhi-Hua, SOM Based Image Segmentation, *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, Springer Berlin Heidelberg, 640—643, 2003.
- (6) Masahiro Ito, Tsutomu Miyoshi, Hiroshi Masuyama, The characteristics of the torus self organizing map, *Faji Shisutemu Shinpojiumu Koen Ronbunshu*. 16. 373-374, 2000.
- (7) Makoto Kinouchi, Naoshi Takada, Yoshihiro Kudo, Toshimichi Ikemura, Quick Learning for Batch-Learning Self-Organizing Map, *Genome Informatics*, 13, 266-267, 2011,