Investigating the Quality of Different Self-Organizing Map Topologies for Complex Data

Huajie Wu, Tom Gedeon and Dingyun Zhu
Research School of Computer Science,
College of Engineering and Computer Science,
The Australian National University,
Acton, Canberra, ACT 0200, Australia.

Abstract—Self-Organizing Maps (SOM) are useful tools for visualizing high dimensional data. However, conventional SOM suffer from the “border effect”. Therefore, Spherical Self-Organizing Maps (SSOM) have been developed to remove such negative effects. In this paper, we extend the topology of SSOM by reconstructing the neighbors to propose the concept of Concentric Spherical Self-Organizing Maps (CSSOM). The major improvement of CSSOM is that it allows using an arbitrary number of spheres and such a topology could be applied in analyzing sequential and time series data. We conducted experiments using these SOM topologies on several datasets. The display schemas and several measures for the quality of SOMs are discussed with the experimental results. The comparison of the results indicates that the quality of SOM is improved through using specified CSSOM depending on the characteristics of the dataset.

I. INTRODUCTION

Self-Organizing Maps (SOM) are primarily used for clustering, classification, sampling and visualizing high dimensional data [3]. This technique has been widely applied in many ways, for instance clustering high-frequency financial data. The conventional neighborhood arrangements are planar SOM made of two-dimensional rectangular or hexagonal lattices. However, the planar SOM has a disadvantage which is the “border effect” [13]. During training, the neurons compete with others. The weight of the winning neuron and its neighbor are updated. Ideally, all the units have the same chance to be updated. However, in the planar map, the units at the border of the map have fewer neighbors than the inside units. At the end of training, the map may not form expected similar regions of the data space, since there are many units with unequal chances of being modified during training [5]. Therefore, many spherical SOMs were proposed in order to solve that problem.

The motivation of this research is to provide a method that users can use an arbitrary number of spheres, and to observe the results of clustering data as well as the quality of SOMs on multiple spheres. In this paper, we consider the Spherical SOM introduced in [4] as the base to be extended to our Concentric Spherical SOM (CSSOM) topology.

The disadvantage of common methods is that the number of neurons in the map is not arbitrary [2]. Concentric Spherical Self-Organizing Maps based on SOM implement multiple layers of SSOM. Compared with that single layer of SSOM, the number of neurons is more able to be varied. More importantly, the standard SOM also has no way to represent sequential data which could be done in multiple layers in CSSOM.

II. RELATED WORK

A. SOM

The Self-Organizing Maps topology was originally introduced by Teuvo Kohonen in 1982 [9]. Generally, Kohonen’s later publications in 1995 [10] and 2001 [11] are regarded as the major references on SOM. It is described as a tool of visualization and analysis of high-dimensional data. Additionally, it is useful for clustering, classification and data mining in different areas. SOM is an unsupervised learning method, the key feature of which is that there are no explicit target outputs or environmental evaluations associated with each input [6]. During the training process, there is no evaluation of correctness of output or supervision. First, it is different from other neural networks, and it only has two layers which are input layer and output layer (or called competition layer) respectively. Every input in input space connects to all the output neurons in the map. The output arrangements are mostly of two dimensions. The Figure 2 shows below conventional 1D and 2D arrangements.

In Figure 1, \( x_n \) represents the input neurons in input space, \( y_n \) represents the outputs in the output space. Figure 1 (a) shows a one dimensional arrangement in form of a line layout. Figure 2 (b) shows a two-dimensional arrangement in form of rectangular layout. It shows that compared to general NN, SOM has no hidden neurons and the discrete layout of the inputs map to output space in a regular arrangement. Besides the rectangular layout, 2D SOM also has the form of hexagonal arrangement.

The main process of Self-Organizing Maps (SOM) consists of three main phases which are competition, cooperation and adaptation [15].
Competition: The output of the neuron in self-organizing map neural network computes the distance (Euclidean distance) between the weight vector and input vector. Then, the competition among the neurons is based on the outputs that they produce, where $i(x)$ indicates the optimal matching input vector, and $i$ is the winning neuron’s weight vector. It uses nearest neighbor search, which is interpreted as proximity search, similarity search or closest point search, consists in finding closest points in metric spaces [7]. The neuron $j$ which satisfies the above condition is called the “winning neuron”.

Cooperation: the winning neuron is located at the center of the neighborhood of topologically cooperating neurons. The winning neuron tends to activate a set of neurons at lateral distances computed by a special function. The distance function must satisfy two requirements: 1) it is symmetric; 2) it decreases monotonically, as the distance increases [8]. A distance function $h(n,i)$ which satisfies the above requirements is Gaussian:

$$h(n,i) = \exp(-\frac{d^2}{2\sigma^2})$$ (2)

Adaption: it is in this phase that the synaptic weights adaptively change. Since these neural networks are self-adaptive, it requires neuron $j$’s synaptic weight $w_j$ to be updated toward the input vector $x$. All neurons in the neighborhood of the winner are updated as well in order to make sure that adjacent neurons have similar weight vectors. The following formula state the weights of each neuron in the neighborhood of the winner are updated:

$$w_j = w_j + \eta h(j,i)(x - w_j)$$ (3)

In formula (3), $\eta$ is a learning rate, $i$ is the index of winning neuron, $w_i$ is the weight of the neuron $i$. The $h(j,i)$ function has been shown in equation (2).

These three phases are repeated during the training, until the changes become less than a predefined threshold.

B. Spherical SOM

Compared to 2D normal SOM, spherical SOMs eliminate the “border effect”. Furthermore, the spherical SOMs have more effective visualization. That is because all neurons have equal geometrical treatment and people may prefer to read the maps from the spheres.

There are a number of spherical SOMs which have been implemented and applied to various datasets. The main spherical SOMs topologies are the followings: GeoSOM, S-SOM, 3D-SOM and H-SOM. GeoSOM was proposed by Wu and Takatsuka [5], which uses a 2D rectangular grid data structure to store the icosahedron-based geodesic dome. In Sangole and Leontitis’s S-SOM work [4], every grid unit stores the list of its immediate neighbors. Boudjemai [14] applied it in 3D modeling as 3D-SOM, whereas Hirokazu [2] developed H-SOM to arrange the neurons along a helix, which is divided into equal parts. Hirokazu’s method allows arbitrary numbers of neurons, but calculating neighbors is still difficult.

III. CONCENTRIC SPHERICAL SOM

The Concentric Spherical Self-Organizing Maps can be interpreted as multiple layers of Sangole and Leontitis’s S-SOM [4], and a tool of more effective visualized representation for sequence data. The CSSOM is composed of four modules which are the initialization module, training module, display schema module and test suite module. The following diagrams show the flow and sequence of the modules and provide an overview of CSSOM’s operation principles.

In the initialization process, all data is saved as variables in the workspace such as X for input vectors, C for the neighbor lists. Based on “X”, “C”, “P” (the Cartesian coordinates) and “spheres”, resize the P and reorganize the neighborhood lists.

Before the training, based on “P”, “C” and “spheres” (representing the cartesian coordinates, neighborhood lists, and number of layers respectively), “P” and “C” need to be resized and modified. This section focuses on the modifications of the neighborhood structure. It can be demonstrated according to the following diagram.

![Figure 2. The general flow of CSSOM.](image-url)

In Figure 3, in order to distinguish the cluster units in different spheres, we use different colors, which are also involved in following explanations. Blue represents Sphere 1, Yellow is for Sphere 2, and Red is for Sphere 3.

In Sangole & Leontisis’ SSOM code, there is only one sphere which we assume is Sphere1. If a in Sphere1 is the winning neuron, in a’s neighborhood list, when $r=1$ ($r$ for radius), the immediate neighbors are $b…g$ (from b to g), when $r=2$, the neighbors exactly 2 away are $h…s$,
when \( r=3 \), the neighbors exactly 3 away are \( t...k \). Normally, the initial value of \( r \) depends on the number of units, which spreads over half of the sphere \([4]\). If it is implemented in CSSOM and the number of layers is 3, the neighbors of a should be added, which contain the units in Sphere2 and Sphere3. The units in different spheres have connections to each other. Therefore, for \( r=1 \), besides \( b...g \) (from b to g in Sphere1), the neighbors of a include a in Sphere2 and a in Sphere3 (Sphere3 is also adjacent to Sphere1, because all the spheres are considered to be in a loop). When \( r=2 \), besides \( h...s \) (from h to s in Sphere1), the neighbors also include \( b...g \) (from b to g in Sphere2) and \( b...g \) (from b to g in Sphere3), and so on.

Next, the algorithm for updating neurons’ neighbors on the data structure is below:

**Algorithm: updating neurons’ neighbors on the data structure**

1. initialize the neighborhood's data structure,
2. assign n spheres of original neighborhood data structure C to a new Cnew;
3. //rsize is the radius size, spheres is the number of spheres,
4. //nsize is the number of units, rsDfl is defualt radius size of C.
5. for i = 1 to rsize do
6. for each sphere j do
7. for each neuron k do
8. update the right index of neighbors in the same sphere;
9. //because initialization just expands the size of Cnew, but the index of neighbors is not correct.
10. end for
11. end for
12. end for
13. if rsize bigger or equal spheres then
14. assign spheres - 1 to rsize;
15. end if
16. if no reminder of spheres devided by 2 then
17. assign(spheres / 2 - 1) to rsize;
18. else
19. assign((spheres + 1) / 2 - 1) to rsize;
20. end if
21. if rsize is larger than rsDfl then
22. for i = 1 to(rsize - rsDfl) do
23. for each sphere j do
24. for k = 1 to rsize do
25. assign Cnew\{k + nsize \* (j - 1), i + size(C, 2)\} to empty value;
26. end for
27. end for
28. end for
29. end for
30. end if
31. assign Cnew to temporary variable CC;
32. for each sphere l do
33. for each neuron i do
34. for j = 1 to rsize do
35. for k = 1 to j do
36. if j - k equal to 0 then
37. add the most adjacent spheres

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of input dimensions</th>
<th>Number of observations</th>
<th>Missing value</th>
<th>Data characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRIS</td>
<td>4</td>
<td>150</td>
<td>No</td>
<td>Multivariate</td>
</tr>
<tr>
<td>ECHS</td>
<td>7</td>
<td>3641</td>
<td>Yes</td>
<td>Time sequence</td>
</tr>
</tbody>
</table>

B. Quantization Error (QE) and Topological Error (TE)

Quantization error and topological error are two widely used measurements which evaluate the quality of a Self-Organizing Map.

First, quantization error evaluates how well the neural network map fits the input patterns. This error is the average distance between all data vectors and their best matching units (winning neurons). The formula can be expressed as:

$$\text{QE} = \frac{1}{n} \sum_{j=1}^{n} \| \bar{x}_j - m_{j} \|$$  \hspace{1cm} (4)

In formula 4.2, \( n \) is the number of input vectors, \( \bar{x}_j \) is the \( j^{th} \) input vector, and \( m_{j} \) is the best matching unit of the \( j^{th} \) input vector.

However, the quantization error cannot describe the topological order of the map, or in other words, it is difficult to measure the topology preservation. Topology preservation is a measurement of how continuous the mapping from input space to the map grid is. Topological error is used to evaluate the complexity of the output space. This error measures the proportion of all data vectors for which first and second best-matching units are adjacent in the size of the network [15]. Those values selected are the optimal ones based on repeated experiments. For ease of the comparisons between those topology methods, similar number of units in each is selected. The appropriate number is 162, especially in CSSOM, 14 spheres of 12 units’ grid (=168 the closest to 162).

In Table II, we can see that both QE and TE of SSOM, CSSOM (14 spheres of 12 units’ grid) are lower than the conventional SOM’s. That is because SSOM and CSSOM have solved the “border effects” problem, and every unit has much the same chance to update the weights. As a result, the errors are decreased. However, QE of CSSOM is slightly higher than SSOM. It is probably because the complexity of the neighborhood structure has been increased when the CSSOM is used. There are not any differences in TE. For ease of comparisons between SSOM and CSSOM, we use more complex data to obtain more results. The “ECSH” dataset is used which has 3,641 patterns. Therefore, the appropriate number of units is 2,562 in SSOM, because it is closest to the number of input patterns. For CSSOM, we select a number of units which is closest to 2,562. The parameter “epoch” is set to 20. The parameter “neighborhood size(s)” is set to 0.5. The results are shown below:

C. Experimental Process, Results and Discussion

First, we use the “IRIS” dataset to compare the quantization errors and topological errors of SOM, SSOM and CSSOM. The parameter “epoch” is set to 44. The parameter “neighborhood size(s)” is set to 0.5. Those parameters are set according to the “nature of dataset”. However, inferences in this regard cannot be made based on existing methods for estimating the SOFM parameters. Error! Reference source not found. Therefore, the values selected are the optimal ones based on repeated 20-times repeated runs. The results are shown below:

In Table III, SSOM is a single SOM with a 2,562-units grid map; CSSOM (4S) represents 4 spheres of 642-units grid map (4*642=2,568); CSSOM (15S) represents 15 spheres of 162-units grid map (16*162=2,582); CSSOM (61S) represents 61 spheres of 42-units grid map (61*42=2,562); CSSOM (214S) represents 214 spheres of 12-units grid map (214*12=2,568).

It is found that TE in Table III is directly related to the average number of neighbors per units shown in Figure 4. Obviously, CSSOM (61S) has the smallest topological error (TE), since the average number of neighbors per unit in CSSOM (61S) is the largest. In other words, as the average number of neighborhoods per unit increases, TE will decrease. That is because TE measures the proportion of with different number of neighbors decreases, as the number of units in a singer layer decreases. Obviously, all 1,267 units in CSSOM (214S) have the same number of neighbors, whereas the units in SSOM have 30 variants with different neighborhoods. The number of units with different numbers of neighbors is larger, so the uniformity is worse. Therefore, CSSOM (214S) is most uniform, and the others are less. The uniformity of the arrangement is one of the factors impacting the QE. As the number of
units with different number of neighbors becomes larger, more and more units have unequal chances to be updated. Therefore, it will lead to influences on forming expected similar regions of data space. At the same time, other factors like the complexity of the units-grid map and the connections between the units have an impact on QE. Therefore, it might be the reason CSSOM (15S) has the smallest QE.

Next, we analyzed the quality of SSOM and CSSOM (15S) in the form of visual representations. The representations of distortions and colors reflect on the magnitude of the similarity measure. The representation in Figure 6 is from “SSOM” in Table III above, one of the experiments with 75.96 in QE and 0.116 in TE.

Therefore, users should have the balance between them patterns might not fit the neural map well (high QE). However, the input patterns might not fit the neural map well (high QE). Therefore, users should have the balance between them according to different demands.

ACKNOWLEDGMENT
This research was supported by the Information and Human Centred Computing (iHcc) research group in RSCS at ANU.

REFERENCES
[16] Sarle, WS 2002, SOM FAQ.
Figure 7. Visual view of CSSOM(15S) from sphere 1 to sphere 15