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.

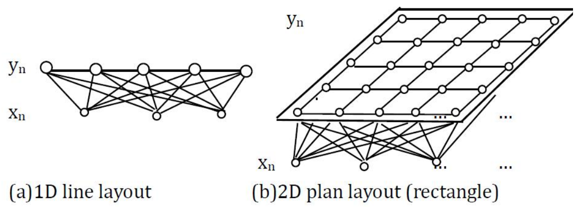

Figure 1. Conventional 1D and 2D arrangements.

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].

978-1-4673-4519-4/12/$31.00 ©2012 IEEE

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 x, the formula can be represented:

$$i(x) = \arg\min_j \|\mathbf{x} - \mathbf{w}_j\|,\ j=1,2,\ldots,l \qquad (1)$$

In formula (1), x is the input vector, w_j is the j^{th} 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:

$$h(j,\ i) = \exp(-\,d_{j,i}^{2} / 2\sigma^{2}) \qquad (2)$$

In formula (2), $h(j,i)$ is the topological area centered around the wining neuron i. The $d_{j,i}$ is the lateral distance between winning neuron i and cooperating neuron j, and σ is the radius influence.

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 neurons in the neighborhood of the winner are updated:

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

In formula (3), η is a learning rate, i is the index of winning neuron, w_j is the weight of the neuron j. 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 Leontitsis'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 Leontisis' 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.

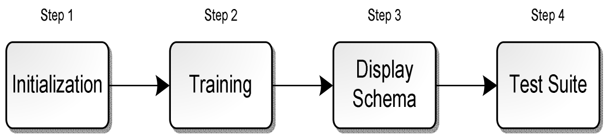

Figure 2. The general flow of CSSOM.

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.

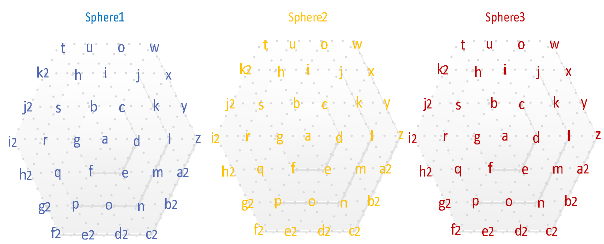

Figure 3. 2D output grid units map of 3 layers of CSSOM.

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$_2$**. 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

```
initialize the neighborhood's data structure,
assign n spheres of original neighborhood data structure C to a new one Cnew;
//rsize is the radius size, spheres is the number of spheres,
//nsize is the number of units, rsDfl is defualt radius size of C.
for i = 1 to rsize do
      for each sphere j do
            for each neuron k do
                  update the right index of neighbors in the same sphere;
//because initialization just expands the size of Cnew, but the index of neighbors is not correct.
            end for
      end for
end for
if rsize bigger or equal spheres then
    assign spheres - 1 to rsize;
end if
if no reminder of shperes devided by 2 then
    assign(spheres / 2 - 1) to rsize;
else
    assign((spheres + 1) / 2 - 1) to rsize;
end if
if rsize is larger than rsDfl then
          for i = 1 to(rsize - rsDfl) do
              for each sphere j do
                  for k = 1 to nsize do
                      assign Cnew{k + nsize * (j - 1), i + size(C, 2)} to empty value;
                      //to prevent the subscript exceeds, when assigning values next loop
                  end for
              end for
          end for
end if
assign Cnew to temporary variable CC;
for each sphere l do
       for each neuron i do
              for j = 1 to rsize do
                     for k = 1 to j do
                            if j - k equal to 0 then
                                   add the most adjacent spheres corresponding index to current Cnew;
                            else then
                                   add k adjacent spheres' corresponding neuron CC 's r = j - k neighbor to current Cnew;
                            end if
                     end for
              end for
       end for
end for
```

IV. Experiment and Results

A. Experiment Description

In this experiment, the main task is to analyze and evaluate the quality of different topologies. There are two figures below used to observe SOMs' quality in each emphasis. The first one compares the quantization errors and topological errors among Kohonen's SOM, Sangole's S-SOM and concentric SSOM using dataset "IRIS". The second table will focus on comparing S-SOM and CSSOM using the more complex dataset "ECSH".

IRIS: This dataset consist of 150 patterns which 50 are Iris Setosa, 50 are Iris Versicolour and 50 are Iris Virginica. There are four attributes for the input data which are sepal length, sepal width, petal length and petal width respectively. More detailed descriptions and the data files can be found on the UCI University of California Irvine machine learning data set repository website[1].

ECSH: This dataset is for a participant in a larger study who read paragraphs in Easy, Calm, Stressful, Hard order. There are two sheets where one sheet has data for a Calm paragraph and the other for Stressful, and in this experiment, the data is for a Calm paragraph. It reflects 60Hz recordings and spans one minute. There are 3,641 input vectors each of which has 7 attributes as listed below:

xGaze: x gaze point on a 1680*1050 monitor

yGaze: y gaze point on a 1680*1050 monitor

pupilLDiam: pupil diameter of left eye

pupilRDiam: pupil diameter of right eye

ecg: ECG

gsr: GSR

bp: blood pressure

The table below shows the summary of those datasets.

TABLE I.
SUMMARY OF DATASETS

Dataset	Number of input dimensions	Number of observations	Missing value	Data characteristics
IRIS	4	150	No	Multivariate
ECSH	7	3641	Yes	Time sequence

[1] http://archive.ics.uci.edu/ml/index.html

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:

$$\text{QE}=\frac{1}{n}\sum_{j=1}^{n}\left\|\vec{x}_j - m_{\vec{x}_j}\right\| \tag{4}$$

In formula 4.2, n is the number of input vectors, $\vec{x}_j$ is the j^{th} input vector, and $m_{\vec{x}_j}$ 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 (BMU) are not adjacent vectors [15]. The formula is expressed as:

$$\text{TE}=\frac{1}{n}\sum_{j=1}^{n}\mu(\vec{x}_j) \tag{5.a}$$

$$\mu(\vec{x}_j)=\begin{cases} 0 & \textit{if data vector's } 1st \textit{ and } 2nd \textit{ best} \\ & \textit{matching units are adjacent} \\ 1 & \textit{if data vector's } 1st \textit{ and } 2nd \textit{ best} \\ & \textit{matching units are not adjacent} \end{cases} \tag{5.b}$$

In (5.a) and (5.b), $\vec{x}_j$ is jth input vector and n is the number of input vectors. Therefore, lower topological error has better topology preservation. In contrast, a high topology error indicates that the output space is complex and it is hard to preserve the topology, so it recommends reducing the size of the network [3]. That is because as the size of the network reduces, the possibility that first-best and second-best matching units are adjacent in output space increase, so the TE becomes lower.

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 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 closet to 162). Every value in Table 4 is the average of all values in 20-times repeated runs. The results are shown below:

TABLE II.
QE AND TE OF SOMS USING IRIS DATASET

Error	SOM	SSOM	CSSOM (14S)
QE	0.257	0.213	0.253
TE	0.027	0.018	0.018

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 some 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 patters. 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:

TABLE III.
QE AND TE OF SSOM AND CSSOM USING ECSH DATASET

Error	SSOM	CSSOM (4S)	CSSOM (15S)	CSSOM (61S)	CSSOM (214S)
QE	193.77	381.71	188.25	587.73	426.14
TE	0.101	0.328	0.179	0.092	0.105

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 word, 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

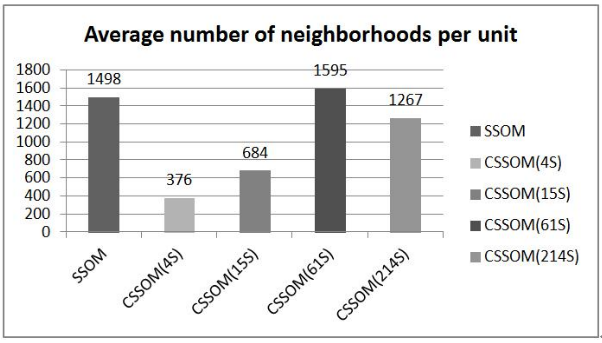

Figure 4. The average number of neighborhoods per units in SSOM and CSSOM.

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.

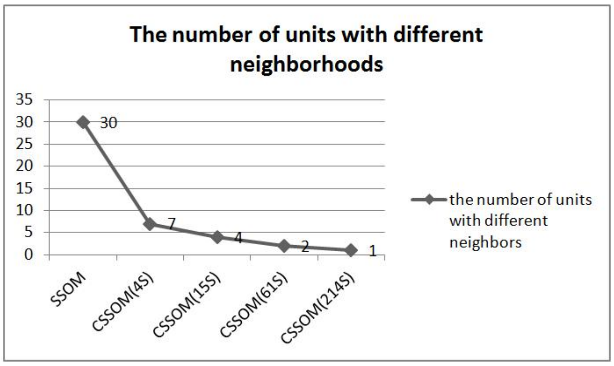

Figure 5. The number of units with different neighborhoods.

Compared to SSOM, besides reflecting the connections and similarity in the same sphere, the visual view of CSSOM has a more visible representation on the connections between the spheres. Figure 7 shows the view of "CSSOM (15S)" in Table III above, one of the experiments with 30.69 in QE and 0.176 in TE. It clearly shows that there are more similarities between spheres 4 to 11.

V. CONCLUSION

In this paper, we propose a new approach for the arrangement of neurons in spherical SOMs, which is CSSOM with arbitrary number of spheres. Users can select arbitrary number of spheres according to the objectives of various experiments. First, CSSOM has better quality than conventional SOM, since it has lower QE and TE. Second, the QE and TE among SSOM and CSSOMs (different number of spheres, but with the approximate number of units in total) are different. If a better topology preservation and greater uniformity of units' neighborhood are desired, the CSSOM with fewer units in one sphere is recommended. 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.

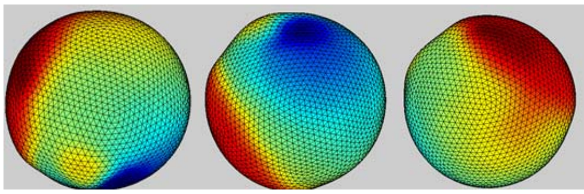

Figure 6. Visual view of SSOM from different angles

ACKNOWLEDGMENT

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

REFERENCES

[1] Leontitsis, A & Sangole, AP 2006, 'Estimating an Optimal Neighborhood Size in the Spherical Self-Organizing Feature Map', International Journal of Computational Intelligence, vol 18, no. 35, pp. 192-196.

[2] Nishio, H, Altaf-Ui-Amin, MD, Kurokawa, K & Kanaya, S 2006, 'Spherical SOM and Arrangement of Neurons Using Helix on Sphere', IPSJ Digital Courier, vol.2, pp. 133-137.

[3] Bação, F & Lobo, V 2010, Introduction to Kohonen's Self-Organizing Maps, Universidade Nova De Lisboa, Portugal, http://edugi.uji.es/Bacao/SOM%20Tutorial.pdf

[4] ce, vol 18, no. 35, pp. 192-196.

[5] Nishio, H, Altaf-Ui-Amin, MD, Kurokawa, K & Kanaya, S 2006, 'Spherical SOM and Arrangement of Neurons Using Helix on Sphere', IPSJ Digital Courier, vol.2, pp. 133-137.

[6] Bação, F & Lobo, V 2010, Introduction to Kohonen's Self-Organizing Maps, Universidade Nova De Lisboa, Portugal, http://edugi.uji.es/Bacao/SOM%20Tutorial.pdf

[7] Sangole, AP and Leontitsis, A 2006, 'Spherical Self-Organizing Feature Map: an Introductory Review', International Journal of Bifurcation and Chaos, vol.16, no. 11, pp. 3195-3206.

[8] Wu, Y and Takatsuka, M 2006, 'Spherical self-organizing map using efficient indexed geodesic data structure', Neural Networks, vol. 19, no. 6, pp. 900-910

[9] Dayan, P 1999, 'Unsupervised learning', in Wilson, RA & Keil, K (ed.), The MIT Encyclopedia of the Cognitive Sciences.

[10] Ramasubramanian, V & Paliwal, KK 2000, 'Fast nearest-neighbor search algorithms based on approximation-elimination search', Pattern Recognition, vol. 33, no. 9, pp. 1497-1510.

[11] Haykin, S 1999, Neural Networks: A Comprehensive Foundation, 2nd edn, Prentice-Hall, Inc., New Jersey, pp. 443-465.

[12] Kohonen, T 1982, 'Self-Organized Formation of Topologicallly Correct Feature Maps', Biological Cybernetics, vol.43.

[13] Kohonen, T 1995, 'Self-Organizing Maps', Springer-Verlag.

[14] Kohonen, T 2001, 'Self-Organizing Maps', *Springer series in information sciences 3rd ed.*

[15] Uriarte, EA & Martin, FD 2005, 'Topology Preservation in SOM', *Interational Journal of Mathematical and Computer Science*, vol.1, no.1, pp. 1-4.

[16] Sarle, WS 2002, SOM FAQ.

[17] Boudjemai, F, Enberg, PB & Postaire, JG 2003, 'Self organizing spherical map architecture for 3D object modeling', in *Proceedings of workshop on self-organizing maps*.

[18] Lawrence, J 1994, *Introduction to neural network: design, theory and applications*, 6th edn, California Scientific Software Press.

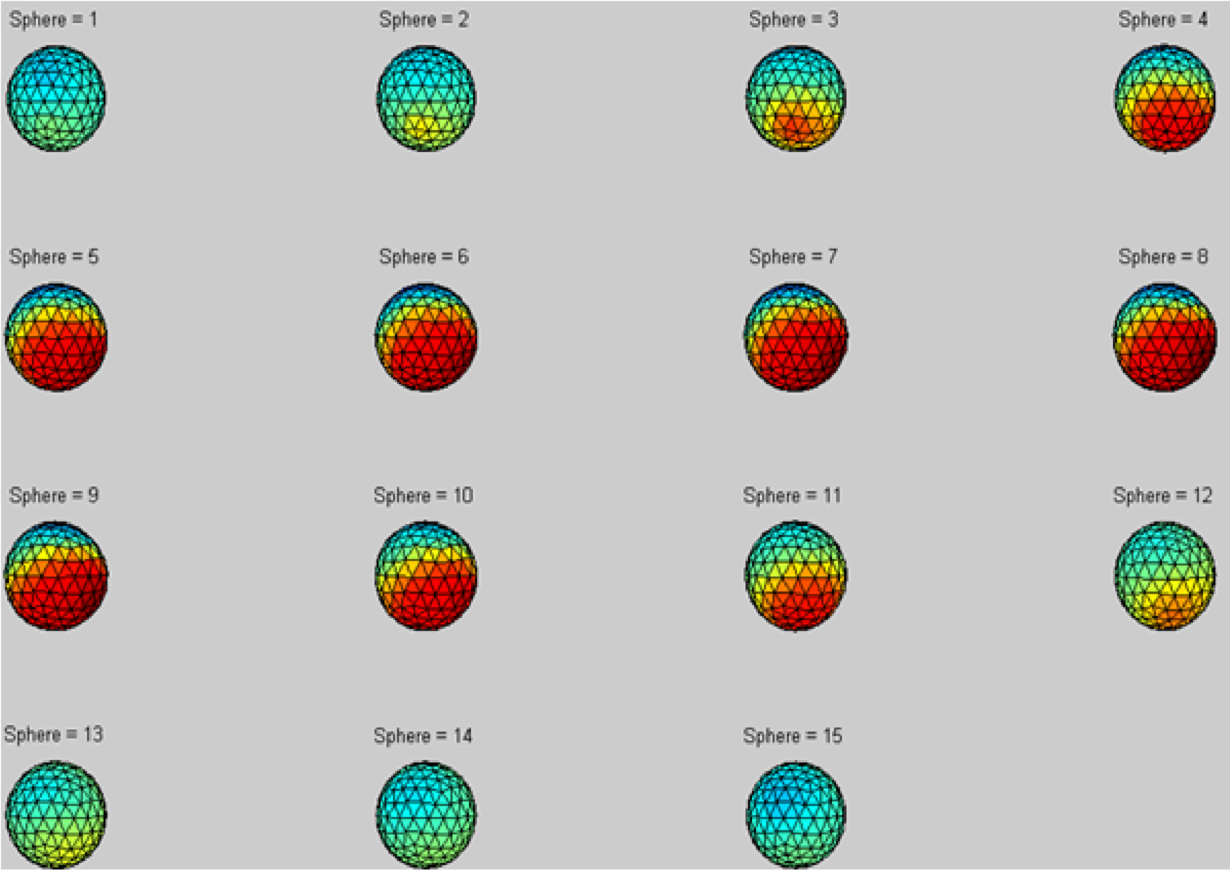

Figure 7. Visual view of CSSOM(15S) from sphere 1 to sphere 15