Topology Preservation in Self-Organizing Maps

Kimmo Kiviluoto
Helsinki University of Technology
Neural Networks Research Centre
Rakentajanaukio 2 C, FIN-02150 Espoo, FINLAND
email: kimmo.kiviluoto@hut.fi

ABSTRACT

This paper concentrates on the following issues: 1. Discussion on what kind of mapping is produced by the SOM algorithm, 2. Introduction of a quantitative measure of continuity for the mapping produced by SOM, 3. Introduction of a variant of SOM, called the AdSOM, with locally adapting neighborhood radii.

1. Introduction

The Self-Organizing Map (SOM) is an unsupervised neural network algorithm that has been succesfully used in a wide variety of applications. The algorithm forms a mapping from the input space onto a lattice of neural units. The dimension of the input space is often very high, whereas that of the neuron lattice is usually only two, so the mapping performs a dimensionality reduction. One of the most fascinating characteristics of the mapping is that it tries to do this preserving topology, to the degree possible.

The current discussion has been motivated by Speckmann et al.'s paper [1], where it was argued that one should have the dimension of the neuron lattice the same as the natural dimension[1] of the input space in order to get good learning results. This seems plausible, but the approach is not without its drawbacks; for instance, displaying the results on a two-dimensional plane would no longer be possible.

As a matter of fact, confining the dimensions of the lattice to two is extremely important in many applications as well as it would be in prospective hardware implementations of the SOM – and it also seems that much of the information processing in biological neural networks is indeed based on two-dimensional maps. Therefore, one should consider, what happens if the dimension of the lattice is lower than the natural dimension of the input space. Is it still possible to get good results – and what exactly do we mean by "good"?

2. Quantifying the Goodness of the Mapping Formed by the SOM

Generally, the goodness for a mapping formed by the SOM may be evaluated using the following criteria:

1. To what degree the mapping is continuous?
2. What is the resolution of the mapping?
3. How does the mapping reflect the probability distribution of the input space?

In the following, we concentrate on the first two items of the above list. These are usually the most important questions in the applications, as the basic SOM usually reflects the probability distribution "well enough". Briefly characterizing, in a *continuous* mapping, input vectors that are close in input space are mapped close in output space; in a mapping of good *resolution*, no vectors that are distant in input space are mapped close in output space.

When the neural lattice of the SOM is lower-dimensional than the input manifold, the topology can not be perfectly preserved in the mapping, and there is a tradeoff between the continuity and resolution of the mapping. The SOM tries to approximate the higher dimension of the input manifold by folding itself

[1] We use term *natural dimension* to emphasize that we are talking about the dimension q of the input space $M \subset \mathbb{R}^n$, which is an intrinsic property of the set M and is independent of the dimension n of the embedding space. Speckmann et al. used actually terms *information dimension* and *fractal dimension*, meaning roughly the same thing.

like a Peano-curve, but this results in discontinuities[2] in the mapping. This behaviour has been termed by Kohonen as "automatic selection of feature dimensions" [2], and it is a very valuable property of SOM, when high resolution is desired.

However, in some applications the continuity of the mapping is more important than good resolution. Then, one should use a map that is flexible enough to find the possibly non-linear "principal components" of the input space, but so stiff that it does not fold itself, trying to represent also the "minor components". The stiffness of the map can be controlled by adjusting the width of the neighborhood function, as proposed by Speckmann et al. [3]; this behaviour has been more thoroughly analyzed by Ritter and Schulten [4].

To find the optimal balance between continuity and resolution, one should have a way to quantify these properties. Quantifying resolution is easily accomplished by using as a measure the *quantization error* $\mathcal{E}_q$, which is defined as the average distance from sample vectors $\mathbf{x}$ to their nearest weight vectors $\mathbf{w}$ (see e.g. [5]). Quantifying continuity, on the other hand, is a more involved task; in the following, we consider some measures proposed earlier and introduce a new measure, the *topographic error* $\mathcal{E}_t$.

2.1. Topographic Product

The first attempt to quantify the continuity of the SOM mapping – often referred to as quantifying the topology preservation or the neighborhood preservation – was the *topographic product* introduced by Bauer and Pawelzik [6]. It compares the weight vectors of the neurons on the map, and if it finds folds on the map, it takes this as an indicator that the map is trying to approximate a higher-dimensional input space, thus producing topographic error.

However, as shown by Villmann et al. [7], the topographic product fails to differentiate between the correct folds that are caused by the map following a folded input space, and the folds that are actually erroneous. Therefore, it gives correct results only when the input space is nearly linear; if the input space is e.g. U-shaped, topographic product yields a nonzero or error-indicating value also for maps that co form a continuous mapping. Thus, results based on topographic product – such as those reported by Speckmann et al. [1, 3] – may or may not be valid, depending on the linearity of the input space.

This illustrates well a more general point. As we do not want to restrict ourselves to linear subspaces, we cannot always rely on Euclidean metrics. In a curvilinear input space, the distances must be measured ultrametrically, "following the input space". However, *locally* the Euclidean metrics usually gives us a good enough approximation. The point is made more precise with the following assumption:

Assumption 1 *Assume that the input space $M \subset \mathbb{R}^n$ is regular enough with respect to the number N of sample vectors $\mathbf{x}_i \in M$, $i = 1, 2, \ldots, N$ available, so that the line segment connecting any two sample vectors that are nearest neighbors in Euclidean metrics may be considered as being contained in M. Moreover, assume that the SOM has been fitted to the data well enough, so that also the line segments connecting sample vectors $\mathbf{x}_i \in M$ to the two nearest weight vectors $\mathbf{w}_i, \mathbf{w}_j$ may be considered as being contained in M.*

2.2. Topographic Function

Villmann et al. introduced another measure of continuity of the mapping: the *topographic function* [7]. Define first the receptive field of a neuron n_i as $R_i = V_i \cap M$, with M denoting the input space and V_i the Voronoi polyhedron[3] of the neuron n_i. The topographic function $\Phi_L^M(s)$ is then defined as the number of neurons that have adjacent receptive fields in the input space, but a city-block distance greater than s on the map:

$$\Phi_L^M(s) = \sum_{i \in L} \#\{n_j \mid j \in L,\, \|n_i - n_j\| > s, n_i \text{ and } n_j \text{ adjacent}\}, \tag{1}$$

where $\#$ denotes the cardinality of a set, and L is the index set for the neural units on the lattice.

[2]Here we use the concept of continuity somewhat loosely, neglecting the unavoidable discontinuities that are caused by the discrete output space. A mapping is considered continuous, if the data vectors that are very close in the input manifold are mapped either to the same or to adjacent neural units.

[3]A Voronoi polyhedron of neuron n_i having a weight vector $\mathbf{w}_i \in \mathbb{R}^n$ is the set $V_i = \{\mathbf{z} \mid \mathbf{z} \in \mathbb{R}^n;\ \|\mathbf{z} - \mathbf{w}_i\| < \|\mathbf{z} - \mathbf{w}_j\|\ \forall j \neq i\}$.

Calculating the topographic function is straightforward. Assuming that there are enough sample vectors and that assumption 1 holds, if two neurons n_i and n_j have adjacent receptive fields, some of the sample vectors $\mathbf{x} \in M$ must have $\mathbf{w}_i$ as their nearest weight vector and $\mathbf{w}_j$ as their second-nearest weight vector. Then, by going through all the sample vectors $\mathbf{x} \in M$ we find which neurons are adjacent, and are thus able to plot the topographic function.

The above assumptions are usually fairly well filled, and the topographic function gives correct results also for nonlinear input spaces. However, there are a few problems with the topographic function. For instance, how to compare two different topographic functions? It would be easier to have simply one number as a measure instead of a function plot, even when the latter would incorporate more information of the characteristics of the mapping.

Another problem is that the topographic function may give misleading results – a single sample vector is enough to render two receptive fields adjacent, but this may be too strict a criterion. As defined, the topographic function does not differentiate between the adjacency of receptive fields in areas where the sample vectors are dense, and in areas where they are sparse. Neither does it differentiate between the adjacency for a long or a short distance. Still, in both these examples, the latter kind of adjacency would imply better continuity.

2.3. Topographic Error

A measure of continuity of the mapping called *topographic error* $\mathcal{E}_t$ is obtained, when the emphasis is shifted from the adjacency of receptive fields towards the proportion of sample vectors that indicate a local discontinuity of the mapping.

Given a sample vector $\mathbf{x} \in M$, let us denote its nearest weight vector with $\mathbf{w}_i$ and second-nearest with $\mathbf{w}_j$. Then, if assumption 1 holds, some of the points in M between $\mathbf{x}$ and $\mathbf{w}_j$ are mapped to $\mathbf{w}_i$, while the rest are mapped to $\mathbf{w}_j$. If the corresponding neurons n_i and n_j are adjacent, the mapping is locally continuous; if they are non-adjacent, there is a local discontinuity, or a local topographic error. The topographic error $\mathcal{E}_t$ for the whole mapping is then obtained by summing up the number of local topographic errors for all sample vectors and normalizing:

$$\mathcal{E}_t = \frac{1}{N}\sum_{k=1}^{N} u(\mathbf{x}_k), \text{ where } u(\mathbf{x}_k) = \begin{cases} 1, & \text{best- and second-best-matching units non-adjacent} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Defined this way, $\mathcal{E}_t$ only gives us an idea of what portion of the local neighborhoods are mapped correctly. It does not describe the kind of incorrect mappings: given two points very close in input space, there is no difference between mapping them one neuron apart, or to opposite corners of the lattice. This seems justified, as otherwise it would be difficult to know whether a high $\mathcal{E}_t$ value indicates that nearly all neighborhoods contain short-range topographic errors, or that in certain few points there is a large discontinuity spanning over the whole map.

3. AdSOM: SOM with locally adapting neighborhood radii

To summarize the previous discussion:

1. If the natural dimension of input space is larger than that of the SOM lattice, the map tries to approximate the higher dimension by folding itself into the input space.
2. The degree of folding depends on the stiffness of the map, which is governed by the width of the neighborhood function: the wider the neighborhood function, the more the map tolerates inputs from directions not well represented on it. On the other hand, too wide neighborhood function results in poor resolution and erroneous averaging of the map.
3. Excessive folding, that results in topographic error, manifests itself so that the best-matching and second-best-matching weight vectors are no longer adjacent.

Then, a potential solution to make the map preserve topology while retaining as much flexibility as possible would be to make the neighborhood radius dependent on the *local* degree of folding. When the neighborhood radius in a certain area of the map shrinks, folds in that area start to grow, until they are big enough to induce topographic error that can be observed. Then, the neighborhood function width in that area is increased so that the topographic error disappears. This is the basic idea behind AdSOM, a variant of the SOM with locally adapting neighborhood radii.

3.1. Implementing the AdSOM

The AdSOM algorithm is otherwise just like the normal SOM, only the width of the neighborhood function is specified for each neural unit separately. In this experiment, only one-dimensional neural lattice was used, but the similar idea is easily extended to larger-dimensional maps.

The neighborhood function $h_i(r)$ is the convex middle part of the Gaussian function, normalized so that its value on the center is unity when the width parameter $\sigma_i = 1$:

$$h_i(r) = \begin{cases} \dfrac{\exp\left[-\frac{1}{2}\left(\frac{r}{\sigma_i}\right)^2\right] - \exp\left(-\frac{1}{2}\right)}{\sigma_i\left[1-\exp\left(-\frac{1}{2}\right)\right]} & \text{if } r < \sigma_i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Here r denotes the lattice distance from the best-matching unit, and subscript i is used to emphasize that the neighborhood function is specific to the ith neural unit.

There is a twofold motivation for cutting off the concave tails of the Gaussian. The first is the result of Erwin et al. [8], that when the neighborhood function is convex, the ordering of the map takes place easier. The second is that some thresholding seems to be necessary for effective implementation, be it biological or hardware, to have the neighborhood function nonzero in a rather small area only.

The exact form of the neighborhood function is probably not very critical. However, having the neighborhood function smooth seems to lead to better results than using the binary-valued "bubble neighborhood" defined as

$$h(r) = \begin{cases} 1/\lfloor\sigma\rfloor, & \text{when } r \le \sigma \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $\lfloor\cdot\rfloor$ denotes rounding to the largest integer smaller than or equal to the argument. For comparisons between the different neighborhood functions, see the simulation results below.

In the AdSOM, each neuron i has its own neighborhood width parameter σ_i associated with it. When the training of the AdSOM is started, the parameter σ_i is set to half the diameter of the lattice for all i. During training, sample vectors are presented to the AdSOM just like to the basic SOM, and the weights are adjusted according to the familiar update rule, with decreasing learning rate. However, the width of the neighborhood is determined by the σ_i value of the best-matching unit.

The parameter σ_i are determined by the local topographic errors. If for a sample vector $\mathbf{x} \in M$ the two nearest weight vectors are $\mathbf{w}_j$ and $\mathbf{w}_k$, and the corresponding best-matching units (BMUs) n_j and n_k are not adjacent, there is local topographic error. Then, for units n_i that are near the two BMUs, a new value for the neighborhood radius σ_i is calculated:

$$\sigma_i = \begin{cases} \|n_j - n_k\| & \text{if } \max\{\|n_i - n_j\|, \|n_i - n_k\|\} \le \|n_j - n_k\| \\ \|n_j - n_k\| - s & \text{otherwise, when } s := \min\{\|n_i - n_j\|, \|n_i - n_k\|\} < \|n_j - n_k\|, \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

Thus, for units between the two BMUs, the new σ_i is equal to the distance on the map between the BMUs; outside that area, the σ_i decreases linearly to one. Should there be another topographic error in the same area, the σ_i values are calculated again, and the larger of the old and new σ_i is stored for each unit.

When N_{rec} sample vectors have been presented, the neighborhood radii σ_i are decreased according to the rule

$$\sigma_i := (\sigma_i)^\beta, \quad 0 \ll \beta < 1 \tag{6}$$

The decrease parameter β must be large enough relative to N_{rec} – if the neighborhood radii shrink too fast, the map has no time to unfold, which results in topographic error.

The computational cost of the AdSOM is almost the same as that of the basic SOM. The second-best-matching unit is found as a by-product of the search for the best-matching unit, so the “inner loop” of the AdSOM requires only a few more comparisons than that of the SOM. The recalculation of the neighborhood radii is also computationally light, and as learning proceeds, it is not performed very often.

3.2. Experimental results

The AdSOM was compared with the basic SOM using a data set that consisted of 3 000 sample vectors and had nonuniform dimension. The neural lattice was one-dimensional, with 65 neural units. To see the effect of the form of neighborhood function, both the bubble neighborhood and the modified Gaussian neighborhood defined in equation (3) were used. The AdSOM parameter values were $N_{\mathrm{rec}} = 100$ and $\beta = 0.97$.

Two experiments were conducted. In the first, the map was originally unordered. During the first 3 000 steps, the learning rate α decreased linearly from 0.99 to 0.05 in all three maps (SOM with bubble and Gaussian neighborhoods, and AdSOM). Simultaneously, the neighborhood width σ decreased linearly from 32 to 1 in the bubble-SOM and to 2 in the Gaussian SOM; the AdSOM σ_i were calculated as described above, with the initial value 32. After the initialization phase, the parameters were frozen in their final values for the next 27 000 steps, except the σ_i of the AdSOM.

In the second experiment, the most severely folded map of the first experiment, produced by the Gaussian SOM, was unfolded. This was accomplished by setting the σ of the bubble and Gaussian SOM to 9; the AdSOM σ_i were again initialized to 32. The learning rate α was 0.05 throughout the experiment, which consisted of 30 000 training steps.

The topographic error $\mathcal{E}_t$ and quantization error $\mathcal{E}_q$ were computed every 500 steps. These are plotted in figure 1, with the pictures of the final weight vectors superpositioned on the sample vectors.

The two basic SOM maps are very close to each other, when the final neighborhood radius is small. However, when the radius is large, there are differences: the bubble neighborhood lumps the weight vectors into clusters, which results in larger quantization error. In these clusters, the weight vectors are close to each other, but they do not lie on the same line segment, so the topographic error does not vanish, as it does when the Gaussian neighborhood is used.

The AdSOM keeps the topographic error almost negligible, without sacrificing too much of the resolution. A very practical feature is that this is accomplished without extensive experimenting with different final values of the width parameter σ. While the learning proceeds, the AdSOM also gives hints of the natural dimension of the input space: in areas where the map has lower dimension than the input space, topology preserving requires wider neighborhoods, so the σ_i parameter values are large even after long training period.

References

[1] H. Speckmann, G. Raddatz, and W. Rosenstiel, “Considerations of geometrical and fractal dimension of SOM to get better learning results”, in *Proc. ICANN'94, Int. Conf. on Artificial Neural Networks*, Maria Marinaro and Pietro G. Morasso, Eds., London, UK, 1994, vol. I, pp. 342–345, Springer.

[2] Teuvo Kohonen, *Self-organization and Associative Memory*, Springer Series in Information Sciences 8. Springer, Berlin Heidelberg New York, 1984.

[3] H. Speckmann, G. Raddatz, and W. Rosenstiel, “Relations between generalized fractal dimensions and Kohonen's self-organizing map”, in *Proc. of NEURONIMES94*, 1994.

[4] H. Ritter and K. Schulten, “Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability, and dimension selection”, *Biological Cybernetics*, vol. 60, no. 1, pp. 59–71, Nov. 1988.

[5] Teuvo Kohonen, *Self-Organizing Maps*, Springer Series in Information Sciences 30. Springer, Berlin Heidelberg New York, 1995.

[6] Hans-Ulrich Bauer and Klaus R. Pawelzik, “Quantifying the neighborhood preservation of self-organizing feature maps”, *IEEE Transactions on Neural Networks*, vol. 3, no. 4, pp. 570–579, July 1992.

[7] Th. Villmann, R. Der, and Th. Martinetz, “A new quantitative measure of topology preservation in Kohonen's feature maps”, in *Proc. ICNN'94, the IEEE Int. Conf. on Neural Networks*, Orlando, Florida, USA, June 1994, IEEE, pp. 645–648.

[8] E. Erwin, K. Obermayer, and K. Schulten, “Self-organizing maps: stationary states, metastability and convergence rate”, *Biological Cybernetics*, vol. 67, no. 1, pp. 35–45, 1992.