

Investigation of Alternative Strategies and Quality Measures for Controlling the Growth Process of the Growing Hierarchical Self-Organizing Map

Michael Dittenbach
iSpaces Group
eCommerce Competence Center – ec3
Donau-City-Straße 1, A-1220 Wien, Austria
E-mail: michael.dittenbach@ec3.at

Andreas Rauber, Georg Pözlbauer
Department of Software Technology
and Interactive Systems
Vienna University of Technology
Favoritenstraße 9–11/188, A-1040 Wien, Austria
E-mail: {rauber,poelzlbauer}@ifs.tuwien.ac.at

Abstract—The *Self-Organizing Map (SOM)* is a very popular neural network model for data analysis and visualization of high-dimensional input data. The *Growing Hierarchical Self-Organizing Map (GHSOM)* – being one of the many architectures based on the *SOM* – has the property of dynamically adapting its architecture during training by map growth as well as creating a hierarchical structure of maps, thus reflecting hierarchical relations in the data. This allows for viewing portions of the data at different levels of granularity. We review different *SOM* quality measures and also investigate alternative strategies as candidates for guiding the growth process of the *GHSOM* in order to improve the hierarchical representation of the data.

I. INTRODUCTION

The *Self-Organizing Map (SOM)* [1], [2] has shown to be exceptionally successful in mapping high-dimensional input data to a two-dimensional output space such that similar inputs are mapped onto neighboring regions of the map. In other words, the similarity of the input data is preserved as faithfully as possible within the representation space of the *SOM*. However, its static nature in terms of map size and the fact that the size must be determined prior to training usually implies multiple runs until optimal results are reached. Especially when large data sets are to be clustered at a fine-grained map resolution, i.e. the map consists of a large number of units, this can be very a time consuming task. Furthermore, the two-dimensional map representation inherently disregards potential hierarchical characteristics of the data.

Among the large number of research publications discussing the *SOM* (see [3] and [4]), different variants and extensions have been introduced. Some of the extensions of the *SOM* algorithm and architecture address the disadvantages of fixed size and missing hierarchical representation. One of the *SOM*-based models implementing an algorithm dealing with both issues is the *Growing Hierarchical Self-Organizing Map (GHSOM)* [5], [6], [7]. The *GHSOM* is a neural network architecture combining the advantages of two principal extension of the *Self-Organizing Map*, dynamic growth and hierarchical structure. Basically, this neural network model is composed of independent *SOMs*, each of which is allowed to grow in size during the training process until a certain quality criterion

regarding data representation is met. This growth process is further continued to form a layered architecture such that hierarchical relations between input data are further detailed at lower layers of the hierarchy. Consequently, the structure of this adaptive architecture automatically adapts itself according to the structure of the input space during the training process.

So far, the *quantization error* has been used as a measure to automatically guide the growth process of the architecture, both in terms of map and hierarchical growth. In other words, the single maps are allowed to grow until a certain quality criterion depending on the quantization error of a higher-layer unit is reached. Moreover, the expansion of the hierarchy into further layers also depends on the quantization errors of the single units on a map. Consequently, each layer deeper in the hierarchy contains maps that represent the data at a higher level of granularity. Depending on the main parameter that guides the training process, the resulting structure is either a flat hierarchy with rather large maps or a deep hierarchy with rather small maps.

However, in the case of the *GHSOM* being used as a tool for providing hierarchically structured access to, e.g. a text document collection, it would be more beneficial to create rather small maps in the upper layers of the hierarchy providing a very coarse overview of the main topics and larger maps in the lower layers for a fine-grained distinction between the data points. In this paper we investigate various *SOM* quality measures reported in literature for their suitability to be used for controlling the growth process of the *GHSOM* as an alternative to the quantization error. We also discuss the necessity of more complex criteria for certain application domains where the *GHSOM* can be used for visualization. We report on their effects of representation quality as well as on the structure of the architecture by comparing the characteristics of trained *GHSOMs* using an inherently hierarchically structured artificial data set.

The remainder of this paper is structured as follows. In Section II we discuss related growing, as well as hierarchical neural network models also being based on the *SOM*. The principles of the original *Growing Hierarchical Self-Organizing*

Map algorithm are reviewed in Section III followed by a discussion of alternative quality measures and strategies to guide map and hierarchy growth in Section IV. In Section V the effects on different growth strategies on the outcome of the training process are shown. Finally, some conclusions are drawn in VI.

II. RELATED WORK

SOM-based neural networks that grow during map training have in common that the training process starts with a rather small number of units. New units are inserted into the network at certain iterations until a stopping criterion, e.g. a predefined number of iterations or level of mapping quality, is reached. In some models, links between units are being added or removed during training, thus, influencing their neighborhood relations and allowing for a stronger separation of clusters. The *Incremental Grid Growing* [8], for example, initially consists of four connected units in a rectangular grid structure. During the training process, the structure as well as the connectivity of the network is dynamically adapted by adding new units at the border of the network adjacent to the unit having the maximum quantization error, in order to provide more map space for a better representation of the input data in this specific area of the network. Similarly, during the training of the *Growing Cell Structures* introduced by Fritzsche [9], units are added and the state of connections changed, but with more freedom regarding the topology of the map space. *Growing Neural Gas* [10] uses a similar algorithm but implements a different learning rule. A dynamically growing network model, again adhering to a rectangular grid structure, is the *Growing Grid* [11]. Complete rows and columns of units are added to the network maintaining a rectangular grid until the training process is terminated by reaching a certain number of map units or if a certain quality criterion is met. New units are inserted periodically after a number of training iterations between the unit with the highest number of “hits” and its most dissimilar direct neighbor in terms of weight vector distance. The connections between the units remain untouched. A growing network model with an adaptive hyper-cubical output space is presented in [12].

A second type of *SOM* variants are hierarchical models such as the *Hierarchical Feature Map* [13]. This model consists of a pyramidal hierarchy of *SOMs* with one map at the top layer and for each of its units, a map is present at the second layer. This principle is repeated with the third and any further layers. Map training is performed top-down according to the standard *SOM* training algorithm, but the input for maps in the second and subsequent layers are only the respective portions of the data which have been mapped onto the according upper-layer units. Despite the computational advantage of this architecture, the size of the single maps and the hierarchical structure itself has to be determined prior to training. Hence, the structure of the representing hierarchy is imposed on the data rather than vice versa. The *Tree Structured SOM* [14] is a hierarchical model primarily with computational speed-up in mind in order to train large maps in shorter time by

exploiting the hierarchy for efficient fast winner search. Quite recently the *Evolving Tree* has been introduced, a *SOM*-based network overcoming the constraints of map-like topologies with the units being arranged in a growing tree topology [15]. An architecture based on the *GHSOM* and *Growing Neural Gas* with parameter self-adjustment focusing on robustness regarding clustering of non-stationary data is the *Dynamic Adaptive Self-Organizing Hybrid Model* [16].

III. GHSOM ARCHITECTURE AND TRAINING

The key idea of the *GHSOM* is to use a hierarchical structure of multiple layers where each layer consists of a number of independent *SOMs*. One *SOM* is used at the first layer of the hierarchy. For every unit in this map a *SOM* might be added to the next layer of the hierarchy. This principle is repeated with the third and any further layers of the *GHSOM*.

Since one of the shortcomings of *SOM* usage is its fixed network architecture we preferred to use an incrementally growing version of the *SOM* similar to the *Growing Grid*. This relieves us of the burden of predefining the network’s size, which is rather determined during the unsupervised training process. We start with a layer 0 consisting of only one single unit. The weight vector of this unit is initialized as the average of all input data. The training process then basically starts with a small map of 2×2 units in layer 1 that is self-organized according to the standard *SOM* training algorithm.

This training process is repeated for a number λ of training iterations. λ is determined by the number of input data to be trained on the map. Ever after λ training iterations the unit with the largest deviation between its weight vector and the input vectors represented by this very unit is selected as the error unit e . In between the error unit e and its most dissimilar neighbor d in terms of the input space either a new row or a new column of units is inserted. The weight vectors of these new units are initialized as the average of their neighbors.

An obvious criterion to guide the training process is the *quantization error* qe_i , calculated as the sum of the distances between the weight vector of a unit i and the input vectors mapped onto this unit. It is used to evaluate the mapping quality of a *SOM* that is based on the *mean quantization error* (*MQE*) of all units on the map. A map grows until its *MQE* is reduced to a certain fraction τ_1 of qe_i of unit i in the preceding layer of the hierarchy. Thus, the map now represents the data mapped onto the higher layer unit i in more detail.

As outlined above, the initial architecture of the *GHSOM* consists of one *SOM*. This architecture is expanded by another layer in case of dissimilar input data being mapped on a particular unit. These units are identified by a rather high quantization error qe_i which is above a threshold τ_2 . This threshold basically indicates the desired granularity of data representation as a fraction of the initial quantization error at layer 0. In such a case, a new map will be added to the hierarchy and the input data mapped on the respective higher layer unit are self-organized in this new map, which again grows until its *MQE* is reduced to a fraction τ_1 of the respective higher layer unit’s quantization error qe_i .

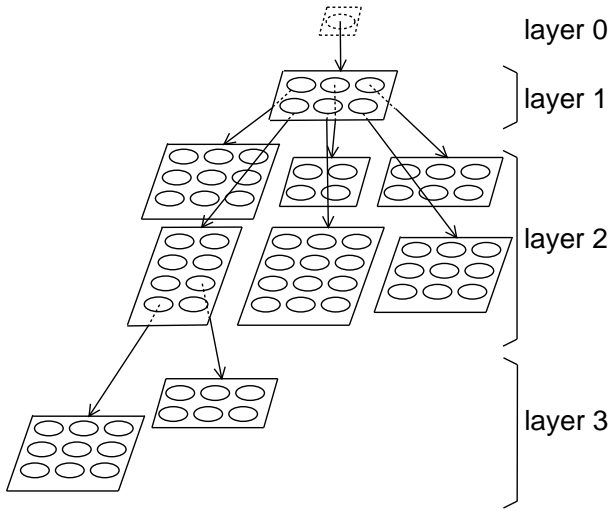


Fig. 1. *GHSOM* reflecting the hierarchical structure of the input data.

A graphical representation of a *GHSOM* is given in Figure 1. The map in layer 1 consists of 3×2 units and provides a rough organization of the main clusters in the input data. The six independent maps in the second layer offer a more detailed view on the data. Two units from one of the second layer maps have further been expanded into third-layer maps to provide a sufficiently granular data representation.

Depending on the desired fraction τ_1 of *MQE* reduction, we may end up with either a very deep hierarchy consisting of small maps, a flat structure with large maps, or – in the most extreme case – only one large map, which is similar to the *Growing Grid*. The growth of the hierarchy is terminated when no further units are available for expansion. It should be noted that the training process does not necessarily lead to a balanced hierarchy in terms of all branches having the same depth. This is one of the main advantages of the *GHSOM*, because the structure of the hierarchy adapts itself according to the requirements of the input space. Therefore, areas in the input space that require more units for appropriate data representation create deeper branches than others.

The growth process of the *GHSOM* is mainly guided by the two parameters τ_1 and τ_2 , which merit further consideration. Parameter τ_2 controls the minimum granularity of data representation, i.e. no unit may represent data at a coarser granularity. If the data mapped onto one single unit still has a larger variation, a new map will be added originating from this unit representing this unit's data in more detail at a subsequent layer. This absolute granularity of data representation is specified as a fraction of the inherent dissimilarity of the data collection as such, which is expressed in the *MQE* of the layer 0 map that is effectively equal to the quantization error of its single unit. If we decide after the termination of the training process that a yet more detailed representation would be desirable, it is possible to resume the training process from the respective lower level maps, continuing to both grow them horizontally as well as adding new lower level maps until

a stricter quality criterion is satisfied. This parameter thus represents a global termination and quality criterion for the *GHSOM*.

Parameter τ_1 controls the actual growth process of the *GHSOM*. Basically, hierarchical data can be represented in different ways favoring either (a) lower hierarchies with rather detailed refinements presented at each subsequent layer or (b) deeper hierarchies, which provide a stricter separation of the various sub-clusters by assigning separate maps. In the first case we will prefer larger maps in each layer that explain larger portions of the data in their flat representation, allowing less hierarchical structuring. In the second case, however, we will prefer rather small maps, each of which describes only a small portion of the characteristics of the data, and rather emphasizes the detection and representation of hierarchical structure.

Thus, the smaller the parameter τ_1 , the larger will be the degree to which the data has to be explained with one single map. This results in larger maps as the map's mean quantization error (*MQE*) will be lower the more units are available for representing the data. If τ_1 is set to a rather high value, the *MQE* does not need to fall too far below the quantization error of the upper layer's unit it is based upon. Thus, a smaller map will satisfy the stopping criterion for the horizontal growth process, requiring the more detailed representation of the data to be performed in subsequent layers.

In a nutshell we can say, that the smaller the parameter value τ_1 , the more shallow the hierarchy, and that the lower the setting of parameter τ_2 , the larger the number of layers in the resulting *GHSOM* network will be.

Apart from the advantage of automatically determining the number of units required for data representation and the reflection of the hierarchical structure in the data, a considerable speed-up of the *GHSOM* training process as compared to standard *SOM* training has to be noted. The reasons for this are twofold. First, at the transition from one layer to the next, vector components that are (almost) identical for all data items mapped onto a particular unit can be omitted for training of the according next layer map, because they do not contribute to differentiation between them. Hence, shorter input vectors lead directly to reduced training times because of faster winner selection and weight vector adaptation. Secondly, a considerable speed-up results from smaller map sizes, as the number of units that have to be evaluated for winner selection is smaller at each map. This results directly from the fact that the spatial relation of different areas of the input space is maintained by means of the network architecture rather than by means of the training process.

IV. QUALITY MEASURES AND ENHANCED GROWTH STRATEGY

Because of the determining characteristic of the *SOM* to provide a topology-preserving mapping from some input data space to a usually two-dimensional output space, most quality measures assess the degree of topological ordering of the map. The *topographic error* [17] is calculated as the number of data

points for which the best-matching unit and the second best-matching unit are not adjacent divided by the total number of data points. Hence, a value of zero means that no topological distortions are present on the map. *Trustworthiness* as introduced in [18], quantifies the mapping quality by measuring if the neighborhoods in the output space are properly preserved in the input space. The *topographic product* [19] only takes the units' weight vectors into account but also determines the degree of topology preservation of the mapping from input to output space. This measure can be used to determine the optimal size of the map as well as the appropriate dimension of the output space. Villmann et al. present an improvement of the *topographic product* in [20]. Contrary to most other measuring approaches, the *topographic function* takes the topology of the input data space into account. For an in-depth theoretical review of quality measures specifically for *Self-Organizing Maps* we refer to Polani [21] and for topographic mappings in general to Goodhill and Sejnowski [22]

However, these quality measures have in common that information about quantization is not taken into account. Moreover, since the estimation of an initial quality value (layer 0) needed as a starting point for *GHSOM* training, the measure has to be computable for single units. Hence, it is not possible to use the topology preservation measures mentioned above.

To reach the desired property of a coarse overview in higher layers and a fine-grained representation in lower layer maps, we introduce an additional factor modifying parameter τ_1 . In order to reach an even higher resolution at lower layers, τ_1 is multiplied by a factor $0 < \beta < 1$ that decreases with each level of the hierarchy. Consequently, the mean quantization error a specific map has to reach is smaller than compared to the standard algorithm.

V. EXPERIMENTS

A. Data Set

In order to effectively demonstrate the effects of the different growth strategies, we have created an artificial data set with obvious hierarchical structure. The application of the *Growing Hierarchical Self-Organizing Map* for organizing a real-world document collections has been shown in [7]. The data set, as depicted in Figure 2, consists of 3,250 two-dimensional data points that have been drawn from different normal distributions. These 65 small clusters consisting of 50 data points each, are grouped to form larger clusters at various levels of granularity constituting six large clusters at the coarsest level. The six large clusters have been labeled with numbers 1 to 6 consisting of 400, 150, 950, 1000, 200 and 500 data points, respectively. Additionally, 26 *second-level* labels have been assigned to the data points in order to be able to evaluate the cluster quality of the second-layer maps.

B. Results

We have trained different *Growing Hierarchical Self-Organizing Maps* setting to show the effects of parameter selection as well as the proposed extension of the growth strategy. The mapping quality itself, i.e. the purity of class

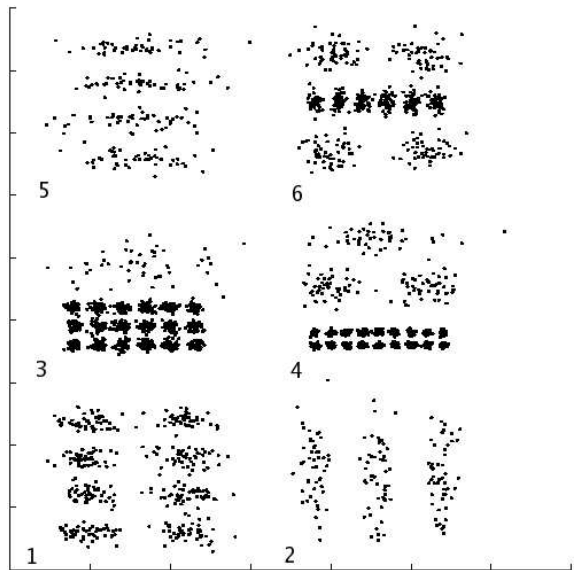


Fig. 2. Artificial two-dimensional data set.

distribution on each node, is nearly optimal, because the input space is of the same dimensionality as the output space. Figures 3 and 4 show the resulting hierarchy of maps with τ_1 set to a high and a low value respectively. The *GHSOM* in Figure 3 consists map of size 3×2 , of which all units have been expanded to the second layer. This number of units on the top-layer map matches exactly the six main clusters of the data set which is desirable in this case. When operating on real world data, such as in document clustering, the inherent number of clusters is of course not that clear, since the data is usually more noisy and clusters are potentially overlapping. Boxes shaded gray denote units that have been expanded to the next layer. Furthermore, the percentages indicate the cluster purity of the map units, i.e. the number of data points from the predominant class on the unit divided by the total number of data points on the unit.

The second-layer maps are, again, rather small and most of the units have further been expanded into the third layer. Please note that the cluster purity values are based on the *second-level* class labels.

Contrary, the second hierarchy as shown in Figure 4 has grown into a larger map, which explains the data in more detail, but the number of data points per unit is still too high for a fine-grained representation. Nevertheless, the quantization errors of most units are low enough, so that only five second-level maps have been created showing the respective parts of the data in more detail. Here, the coarse overview in the top level is not optimally provided as the main clusters are already distributed over different numbers of units.

Consequently, using a factor β to increase the required quality criterion defined by τ_1 provides the solution for an optimal display of the complete data set in a convenient and concise manner as shown in Figure 5.

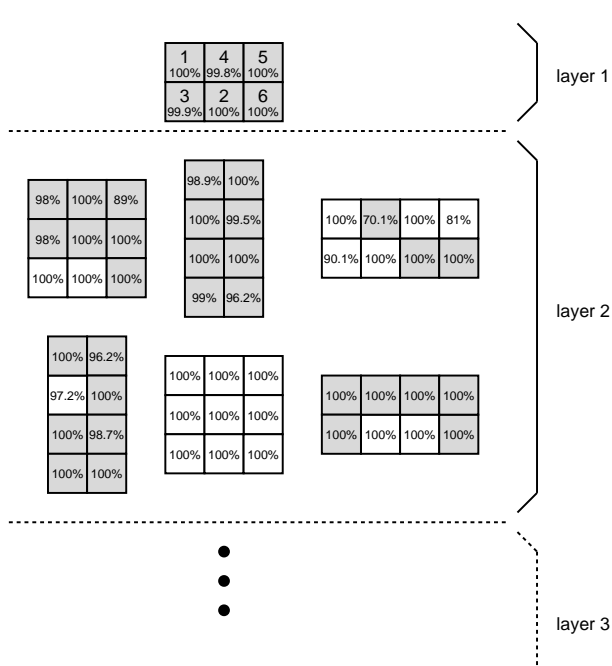


Fig. 3. GHSOM hierarchy with small maps. Percentages denote cluster purity values of the units.

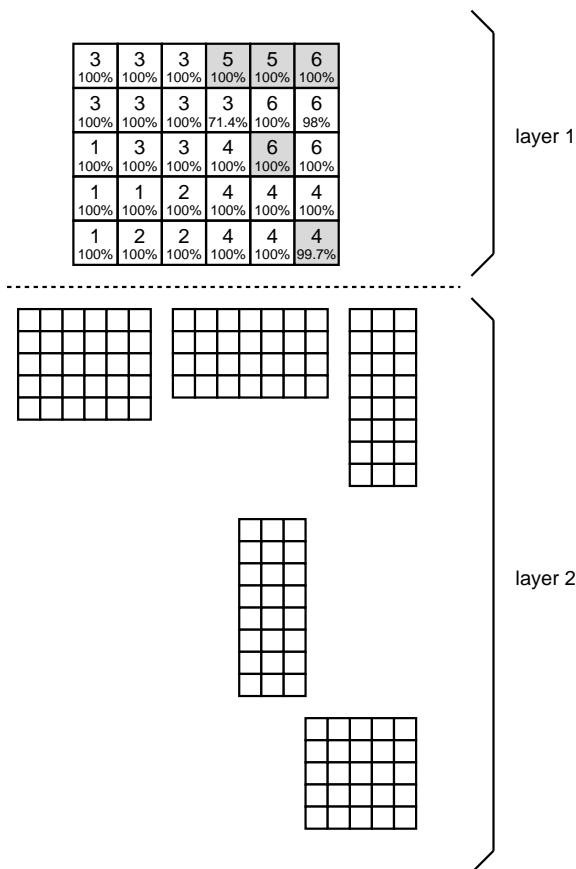


Fig. 4. GHSOM with large maps. All units in the second and subsequent layers have a cluster purity of 100%.

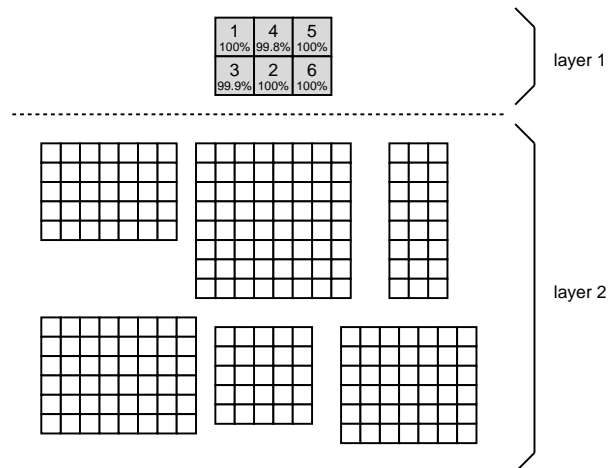


Fig. 5. GHSOM with enhanced growth strategy. All units in the second and subsequent layers have a cluster purity of 100%.

VI. CONCLUSIONS

In this paper we have revisited the *Growing Hierarchical Self-Organizing Map (GHSOM)*, a dynamically growing, hierarchical neural network architecture based on the *Self-Organizing Map*. We have found that most of the quality measures found in literature are not suitable for the purpose of improving the growth process due to their nature of focusing on measuring topology preservation rather than quantization properties of such maps. Hence, we have considered a different strategy regarding the determination of the desired level of data representation granularity at the different levels of the hierarchy again based on the quantization error only. This strategy improves the quality of data representation especially for the task of providing an interface for visual data inspection where a rough classification at the top level and a more fine-grained representation at the lower layers is desired as it is the case for hierarchically organizing large text document collections. Nevertheless, the combination of a quality measure for map growth and a different measure for expanding units to subsequent layers merits further consideration.

However, it is clear that an additional parameter is the downside of an improved outcome of the *GHSOM* training process for certain application domains, such as document clustering. Additionally, it has to be noted that quantitatively assessing the proposed improvement is difficult, because a large real-world data set – which justifies using the *GHSOM* because of the number of data points, e.g. documents – that includes class labels at different levels of a hierarchical class structure is not known to the authors.

REFERENCES

- [1] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, 1982.
- [2] —, *Self-Organizing Maps*, 3rd ed., ser. Springer Series in Information Sciences. Springer-Verlag, 2001, vol. 30.
- [3] S. Kaski, J. Kangas, and T. Kohonen, "Bibliography of self-organizing map (SOM) papers: 1981–1997," *Neural Computing Surveys*, vol. 1, no. 3&4, pp. 1–176, 1998.

- [4] M. Oja, S. Kaski, and T. Kohonen, "Bibliography of self-organizing map (SOM) papers: 1998–2001 addendum," *Neural Computing Surveys*, vol. 3, pp. 1–156, 2003.
- [5] M. Dittenbach, D. Merkl, and A. Rauber, "The growing hierarchical self-organizing map," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000)*, S. Amari, C. L. Giles, M. Gori, and V. Puri, Eds., vol. 6. Como, Italy: IEEE Computer Society, July 24–27 2000, pp. 15–19.
- [6] M. Dittenbach, A. Rauber, and D. Merkl, "Recent advances with the growing hierarchical self-organizing map," in *Advances in Self-Organizing Maps*, N. Allinson, H. Yin, L. Allinson, and J. Slack, Eds. Lincoln, GB: Springer-Verlag, June 13-15 2001, pp. 140–145.
- [7] —, "Uncovering the hierarchical structure in data using the growing hierarchical self-organizing map," *Neurocomputing*, vol. 48, no. 1–4, pp. 199–216, November 2002.
- [8] J. Blackmore and R. Miiikkulainen, "Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map," in *Proceedings of the International Conference on Neural Networks (ICNN 1993)*, vol. 1, San Francisco, CA, 1993, pp. 450–455.
- [9] B. Fritzke, "Growing cell structures – A self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol. 7, no. 9, pp. 1441–1460, 1994.
- [10] —, "A growing neural gas network learns topologies," in *Advances in Neural Information Processing Systems 7*, G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds. MIT Press, 1995, pp. 625–632.
- [11] —, "Growing Grid – A self-organizing network with constant neighborhood range and adaption strength," *Neural Processing Letters*, vol. 2, no. 5, 1995.
- [12] H.-U. Bauer and T. Villmann, "Growing a hypercubical output space in a self-organizing feature map," *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 218–226, 1997.
- [13] R. Miiikkulainen, "Script recognition with hierarchical feature maps," *Connection Science*, vol. 2, 1990.
- [14] P. Koikkalainen and E. Oja, "Self-organizing hierarchical feature maps," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 1990)*, San Diego, CA 1990.
- [15] J. Pakkanen, J. Iivarinen, and E. Oja, "The Evolving Tree – A novel self-organizing network for data analysis," *Neural Processing Letters*, vol. 20, no. 3, pp. 199–211, 2004.
- [16] C. Hung and S. Wermter, "A dynamic adaptive self-organising hybrid model for text clustering," in *Proceedings of The 3rd International Conference on Data Mining (ICDM 2003)*. Melbourne, FL: IEEE Computer Society, December 2003, pp. 75–82.
- [17] K. Kiviluoto, "Topology preservation in self-organizing maps," in *Proceedings of the International Conference on Neural Networks (ICNN 1996)*. Washington DC, USA: IEEE Press, June 1996, pp. 294–299.
- [18] J. Venna and S. Kaski, "Neighborhood preservation in nonlinear projection methods," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN 2001)*, ser. LNCS, G. Dorffner, H. Bischof, and K. Hornik, Eds., no. 2130. Vienna, Austria: Springer-Verlag, August 21–25 2001, pp. 485–491.
- [19] H.-U. Bauer and K. 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.
- [20] T. Villmann, R. Der, M. Herrmann, and T. Martinetz, "Topology preservation in self-organizing feature maps: Exact definition and measurement," *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 256–266, 1997.
- [21] D. Polani, "Measures for the organization of self-organizing maps," in *Self-Organizing Neural Networks: Recent Advances and Applications*, ser. Springer Studies In Fuzziness And Soft Computing, U. Seiffert and L. C. Jain, Eds. Physica-Verlag, 2002, pp. 13–44.
- [22] G. J. Goodhill and T. J. Sejnowski, "Quantifying neighbourhood preservation in topographic mappings," in *Proceedings of the 3rd Joint Symposium on Neural Computation*, Pasadena, CA, 1996, pp. 61–82.