

# Recent Advances with the Growing Hierarchical Self-Organizing Map

Michael Dittenbach<sup>1</sup>, Andreas Rauber<sup>2</sup>, and Dieter Merkl<sup>2</sup>

<sup>1</sup> E-Commerce Competence Center – EC3,  
Siebensterng. 21/3, A-1070 Vienna, Austria

<sup>2</sup> Institut für Softwaretechnik, Technische Universität Wien,  
Favoritenstraße 9-11/188, A-1040 Wien, Austria  
[www.ifs.tuwien.ac.at/~mbach, ~andi, ~dieter](http://www.ifs.tuwien.ac.at/~mbach, ~andi, ~dieter)

**Summary.** We present our recent work on the *Growing Hierarchical Self-Organizing Map*, a dynamically growing neural network model which evolves into a hierarchical structure according to the necessities of the input data during an unsupervised training process. The benefits of this novel architecture is shown by organizing a real-world document collection according to semantic similarities.

## 1 Introduction

The *Self-Organizing Map (SOM)* [3,4] has shown to be exceptionally successful in arranging high-dimensional input data along its 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*.

Despite the large number of research reports on applications of the *SOM* some deficiencies remained largely untouched. First, the *SOM* uses a static network architecture both in terms of number and arrangement of neural processing elements, which have to be defined prior to the start of training. Second, hierarchical relations between the input data are rather difficult to detect in the map display. So far, both issues are have been addressed separately by means of adaptive architectures, e.g. the *Growing Grid* [2], or hierarchies of independent *SOMs*, e.g. the *Hierarchical Feature Map* [6].

With the *Growing Hierarchical Self-Organizing Map (GHSOM)* [1] we proposed a novel neural network model that addresses both deficiencies as outlined above within one framework. 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 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 deeper layers of the neural network.

In this paper we report on our recent work on the *GHSOM* where particular emphasis is directed to the orientation of the various layers during

training. In particular, we discuss the positive effects on input data representation when the initial orientation of deeper layers is chosen according to their respective higher layer maps. We can thus maintain the already learned similarities between input data during the establishment of the hierarchical structure of the *GHSOM*. As a consequence, the negative effects of generating strictly disjoint clusters are alleviated because neighboring maps in deeper layers of the hierarchy show similar characteristics at their respective borders.

## 2 Growing Hierarchical Self-Organizing Map

### 2.1 Architecture and training

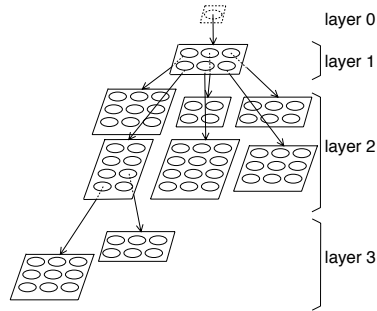
The key idea of the *GHSOM* is to use a hierarchical structure of multiple layers where each layer consists of a number of independent *SOMs*. One *SOM* is used at the first layer of the hierarchy. For every unit in this map a *SOM* might be added to the next layer of the hierarchy. This principle is 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 rather use an incrementally growing version of the *SOM*. This relieves us from the burden of predefining the network's size which is rather determined during the unsupervised training process. We start with a layer 0, which consists of only one single unit. The weight vector of this unit is initialized as the average of all input data. The training process basically starts with a small map of, say,  $2 \times 2$  units in layer 1, which is self-organized according to the standard *SOM* training algorithm.

This training process is repeated for a fixed number  $\lambda$  of training iterations. Ever after  $\lambda$  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. In between the error unit and its most dissimilar neighbor 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  $q_i$ , 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* based on the mean quantization error (*MQE*) of all units in the map. A map grows until its *MQE* is reduced to a certain fraction  $\tau_1$  of the  $q_i$  of the 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  $q_i$  which is above a threshold  $\tau_2$ . This threshold basically indicates the desired granularity level of data representation as a



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

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  $\tau_1$  of the respective higher layer unit's quantization error  $q_i$ . Note that this may not necessarily lead to a balanced hierarchy. The depth of the hierarchy will rather reflect the diversity in input data distribution which should be expected in real-world data collections.

Depending on the desired fraction  $\tau_1$  of *MQE* reduction we may end up with either a very deep hierarchy with 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.

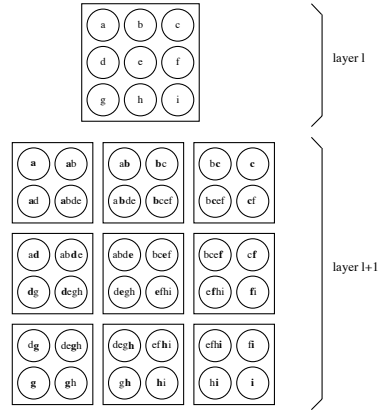
A graphical representation of a *GHSOM* is given in Figure 1. The map in layer 1 consists of  $3 \times 2$  units and provides a rough organization of the main clusters in the input data. The six independent maps in the second layer offer a more detailed view on the data. Two units from one of the second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

## 2.2 Orientation

The hierarchical structuring imposed on the data results in a separation of clusters mapped onto different branches. While this, in principle, is a desirable characteristic helping to understand the cluster structure of the data, it may lead to misinterpretations when large clusters are mapped and expanded on two neighboring, yet different units. Similar input data are thus rather arbitrarily separated in different branches of the hierarchy.

In order to provide a global orientation of the individual maps in the various levels of the hierarchy, their orientation must conform to the orientation of the data distribution on their parents' maps. This can be achieved by creating a coherent initialization of the units of a newly created map.

Let unit  $p$  be expanded to form a new  $2 \times 2$  map in the subsequent layer of the hierarchy. This map's four weight vectors  $s_1$  to  $s_4$  are initialized to



**Fig. 2.** Initialization of units for expanded maps

mirror the orientation of neighboring units of  $p$ . This is achieved by adding a fraction of the weight vectors in the neighborhood of  $p$ . Figure 2 provides an illustration of the initialization of new maps. Given a  $3 \times 3$  map with weight vectors  $a$  to  $i$ , the weight vectors of the four units representing the maps expanded from units  $a$  and  $e$  are initialized as provided in Expression 1.

$$\begin{array}{l}
 a_1 = a \\
 a_2 = a + \frac{(b-a)}{2} \\
 a_3 = a + \frac{(d-a)}{2} \\
 a_4 = a + \frac{(b-a)+(d-a)+(e-a)}{3}
 \end{array}
 \left|
 \begin{array}{l}
 e_1 = e + \frac{(a-e)+(b-e)+(d-e)}{3} \\
 e_2 = e + \frac{(b-e)+(c-e)+(f-e)}{3} \\
 e_3 = e + \frac{(d-e)+(g-e)+(h-e)}{3} \\
 e_4 = e + \frac{(f-e)+(h-e)+(i-e)}{3}
 \end{array}
 \right.
 \quad (1)$$

This initial orientation of the map is preserved during the training process. While new units may be inserted in between, the 4 corner units will still be the most similar to the respective corner units of the maps in neighboring branches.

### 3 Experiments

For the experiments presented hereafter we use a collection of 11,627 articles from the Austrian daily newspaper *Der Standard* covering the second quarter of 1999. To be used for map training, a vector-space representation of the single documents is created by full-text indexing. Instead of defining language or content specific stop word lists, we rather discard terms that appear in more than 813 (7%) or in less than 65 articles (0.56%). We end up with a vector dimensionality of 3,799 unique terms. The individual documents are then represented by feature vectors using a  $tf \times idf$ , i.e. term frequency times inverse document frequency, weighting scheme [7]. This weighting scheme assigns high values to terms that are important as to describe and discriminate between the documents. These feature vectors are used to train the *GHSOM*.

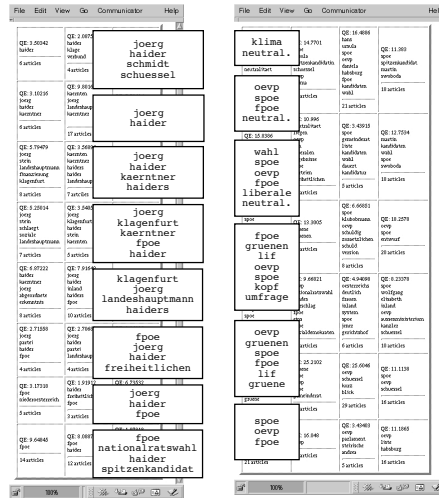


Fig. 3. Two neighboring second-layer maps on Internal Affairs

Training the *GHSOM* with parameters  $\tau_1 = 0.035$  and  $\tau_2 = 0.0035$  results in a shallow hierarchical structure of up to 7 layers. The layer 1 map grows to a size of  $7 \times 4$  units, all of which are expanded at subsequent layers. We find the most dominant branches to be, for example, *Sports*, located in the upper right corner of the map, *Internal Affairs* in the lower right corner, *Internet*-related articles on the left hand side of the map, to name but a few. However, due to the large size of the resulting first level map, a fine-grained representation of the data is already provided at this layer. This results in some larger clusters to be represented by two neighboring units already at the first layer, rather than being split up in a lower layer of the hierarchy. For example, we find the cluster on *Internal Affairs* to be represented by two neighboring units. One of these, on position (6/4), covers solely articles related to the *Freedom Party* and its political leader *Jörg Haider*, representing one of the most dominant political topics in Austria for some time now, resulting in an accordingly large number of news articles covering this topic. The neighboring unit to the right, i.e. located in the lower right corner on position (7/4), covers other *Internal Affairs*, with one of the main topics being the elections to the *European Parliament*. Figure 3 shows these two second-layer maps.

We also find, for example, articles related to the *Freedom Party* on the branch covering the more general *Internal Affairs*, reporting on their role and campaigns for the elections to the *European Parliament*. As might be expected these are closely related to the other articles on the *Freedom Party* which are located in the neighboring branch. Obviously, we would like them to be presented on the left hand side of this map, so as to allow the transition from one map to the next, with a continuous orientation of topics. Due to the initialization of the added maps during the training process, this contin-

uous orientation is preserved, as can easily be seen from the automatically extracted labels [5] provided in Figure 3. Continuing from the second layer map of unit (6/4) to the right we reach the according second layer map of unit (7/4) where we first find articles focusing on the *Freedom Party*, before moving on to the *Social Democrats*, the *People's Party*, the *Green Party* and the *Liberal Party*.

## 4 Conclusions

In this paper we have presented our recent work on the *Growing Hierarchical Self-Organizing Map*. The major features of this neural network are its hierarchical architecture, where the depth of the hierarchy is determined during the unsupervised training process. Each layer in the hierarchy consists of a number of independent *SOMs* which determine their size and arrangement of units during training. Thus, this model is especially well suited for applications which require hierarchical clustering of the input data. We have shown that significant improvement in data representation can be achieved by directing particular emphasis at the orientation of the various *SOMs* constituting the different branches of the hierarchy. Maps of neighboring branches now show the same orientation as the map they are derived from. Therefore the similarity of different clusters is presented in an intuitive fashion.

## References

1. M. Dittenbach, D. Merkl, and A. Rauber. The growing hierarchical self-organizing map. In *Proc Int'l Joint Conf Neural Networks*, Como, Italy, 2000.
2. B. Fritzke. Growing Grid – A self-organizing network with constant neighborhood range and adaption strength. *Neural Processing Letters*, 2(5), 1995.
3. T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 1982.
4. T. Kohonen. *Self-organizing maps*. Springer-Verlag, Berlin, 1995.
5. D. Merkl and A. Rauber. Automatic labeling of self-organizing maps for information retrieval. In *Proc Int'l Conf Neural Information Processing*, Perth, Australia, 1999.
6. R. Miikkulainen. Script recognition with hierarchical feature maps. *Connection Science*, 2, 1990.
7. G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, MA, 1989.