Alternative Ways for Cluster Visualization in Self-Organizing Maps

Dieter Merkl, Andreas Rauber

Institut für Softwaretechnik, Technische Universität Wien
Resselgasse 3/188, A-1040 Wien, Austria
{dieter, andi}@ifs.tuwien.ac.at

Abstract

We present two enhanced visualization techniques for the self-organizing map allowing the intuitive representation of input data similarity. The general idea of both approaches is to visualize the relationship of nodes to facilitate the detection of cluster boundaries without modifying the architecture or the basic training process of SOM. One approach mirrors the movement of weight vectors during the training process within a two-dimensional (virtual) output space, whereas the second results in a grid of connected nodes where the intensity of the connection mirrors the similarity of the underlying data items. Both approaches can be combined to allow improved analysis of the inherent structure of high-dimensional input data and an intuitive recognition of cluster boundaries without the necessity of substantial prior knowledge concerning the input patterns.

1 Introduction

The self-organizing map allows the mapping of high-dimensional input data onto a two-dimensional output space while preserving the structure of input data as faithfully as possible. The resulting topographic map represents similarity between input vectors in terms of the distance between the respective units located in a fixed grid. It still represents a non-trivial task, however, to determine the boundary between unrelated parts of the output space within the limited possibilities provided by the standard representation. In other words, the various clusters of similar input data often may only be detected when the person examining the map representation has some prior knowledge of the input structures. This clearly is one deficiency of the self-organizing map.

Only recently, variations on the basic idea of self-organizing maps with a special focus on the facilitation of cluster boundary detection have been proposed. One of those approaches based solely on post-processing a trained SOM to create a 3-dimensional representation is the so-called *U-matrix* [12], using the distances encountered in a trained SOM to visualize clusters. Another stream of research relies on adaptive architectures as for instance *incremental grid growing* [1, 2] and *growing cell structures* [3, 4]. These architectures incrementally grow and split according to the specific requirements of the actual input data. As a result, these architectures produce isolated clusters of similar input data items. Other authors suggest the utilization of multiple independent self-organizing maps as for example *multiple 2-dimensional SOM* [13], or *hierarchical feature maps* [9]. The central feature of these architectures is that the independent self-organizing maps specialize on different clusters of the input data during the learning process. All of these approaches enable an intuitive visual representation of cluster boundaries. We should mention, however, that the approaches relying on changed architectures require more "magic numbers", i.e. training parameters, to be specified before the training process starts. This makes them much more susceptible to minor variations of the learning parameters as compared to the extremely robust model of self-organizing maps. Moreover, the recognition of inter-cluster relationship is not as easy as in SOMs, to say the least.

These problems raise the necessity for novel approaches addressing the problem of cluster boundary detection while not interfering with the robustness offered by the standard SOM model. In this paper we present two approaches which provide a straight-forward visualization of the inherent structure of input data. Being add-ons to the standard SOM model, its robustness is preserved. Information obtained during the training process or present in any trained map but not used in the standard representation is used to create enhanced representations aiding in exploratory data analysis. The first approach mirrors the movement of a unit's weight vector in input space during the training process in a two-dimensional (virtual) output space, leading to an image of the clustering of nodes found in input space. The second technique utilizes the distance between weight vectors of neighboring nodes to determine their degree of connectivity. Depending on their distance the nodes are connected by using connections of different intensity where the intensity corresponds to the nodes' mutual similarity.

The remainder of the paper is organized as follows. In Section 2 we present a brief review of these two approaches. In Section 3 we present the experimental results of both approaches applied to a popular benchmark data set. A discussion of both visualization techniques is provided in Section 4. Finally, our conclusions are presented in Section 5.

2 SOM, Adaptive Coordinates and Cluster Connections

Both approaches to simliarity visualization that are presented thereafter are based on the standard architecture and learning procedure of Kohonen's self-organizing map (SOM) architecture [5, 6] consisting of a grid of units with n-dimensional weight vectors. The input signals are presented to the map in random order and an activation function (usually based on the Euclidean distance between the weight vector and the presented signal) is used to determine the winning unit. The weight vector of the winner together with the weight vectors of neighboring units are modified in order to represent the input signal more closely.

With the so-called *Adaptive Coordinate*, AC, approach we extended the basic learning rule in order to capture the movements of the various weight vectors within a two-dimensional 'virtual' output space for subsequent visualization of the classification result. Initially, each of the output units of the self-organizing maps is shown in its initial position within the two-dimensional grid defined by the neural network architecture. In other words, the initial coordinates $\langle ax_i, ay_i \rangle$ of the unit i are identical to the unit's prosition within the grid of the map. During each learning step the distance between the weight vectors and a randomly selected input pattern are stored in a table, i.e. $Dist(t)$. After the adaptation of weight vectors a similar distance table is calculated, i.e. $Dist(t+1)$. The obvious next step is to compute the relative change in distance for every unit i as given in Expression 1.

$$\Delta Dist_i(t+1) = \frac{Dist_i(t) - Dist_i(t+1)}{Dist_i(t)} \tag{1}$$

The movement of the various weight vectors within the input space due to the adaptation process is performed analogously within the 'virtual' output space. Pragmatically speaking, the adaptive coordinates $\langle ax_i, ay_i \rangle$ are used in order to mimic the movement of this unit's weight vector during the training process. Since the presented input signal was mapped onto the winning unit c, no adaptation of the position of the winning unit is performed, with c now being representative of the selected input signal in terms of the 'virtual' output space. The adaptive coordinates of all units but the winner are now moved by the fraction given in $\Delta Dist_i(t+1)$ towards the position of the winning unit c given by $\langle ax_c, ay_c \rangle$. In Expression 2 we provide the exact formulation of adaptation performed with the ax-coordinate ($ay_i(t+1)$ is calculated analogously).

$$ax_i(t+1) = ax_i(t) + \Delta Dist_i(t+1) \cdot (ax_c(t) - ax_i(t)) \tag{2}$$

Thus, the clustering of units around the winning unit resembles the clustering of the units' weight vectors around the presented input signal after the current training cycle. After convergence of the training process the clusters learned by the self-organizing map can be visualized by using the coordinates $\langle ax_i, ay_i \rangle$ to plot the unit i in the 'virtual' output space. A more detailed analysis of the AC approach may be found in [10, 8].

While AC try to obtain additional information concerning the clustering of nodes during the training process, *Cluster Connections*, CC, are based on post-processing the information contained in the various weight vectors of a trained SOM. Pragmatically speaking, the distances between weight vectors of neighboring nodes are used to determine whether these nodes belong to the same cluster and should thus be shown as connected, or are located along a cluster boundary and should thus be not connected. In order to enhance the visual representation, a set of thresholds is used to visualize different degrees of similarity. As a result our visualization technique is capable of, first, revealing the inter-cluster relation of the input data items as well as, second, visualizing qualitative information on cluster membership.

More formally, the Cluster Connection technique relies on the analysis of the distances between the weight vectors of neighboring units. A closer look at the distances d_{ij} between pairs of nodes i, j in a SOM reveals, that three more or less distinct types of relationships between neighboring nodes can be found. Pragmatically speaking, these three types may be described as given below.

1. Highly similar nodes: The weight vectors of a pair of nodes representing highly similar input items may be expected to be highly similar, too. Analogously, weight vectors of a pair of nodes where the one represents an input whereas the other does not, albeit being specialized to a similar region of the input space, may be expected to be highly similar as well.

2. Medium similar nodes: The weight vectors of pairs of nodes representing input data belonging to the same cluster may be expected to exhibit medium similarity.

3. Dissimilar nodes: The weight vectors of pairs of neighboring nodes with a cluster boundary going right in between them may be expected to be remarkably dissimilar.

Based on the analysis of the distances encountered in a trained SOM by a distance diagram, a set of thresholds or a monotonous grey-scale representation can be used to visualize the closeness of relationship of neighboring nodes by connections of differing intensity. Various implementations of this basic idea which may serve the purpose of cluster visualization are provided in [10].

3 AC - CC: A Comparison

In this section we provide simulation results for both approaches applied to a self-organizing map trained with an artificial input data set, namely the *Animals* data set given in Table 1 (cf. [11]). The SOM implementation we used for the experiments described below is presented in [7]. In the representations the winning units are labeled by the corresponding input signal.

Figure 1a.) depicts an 6×6 SOM trained with the input data given in Table 1. Note that the input vectors were normalized prior to the training process. Furthermore, please mind that we omitted the encoding of the animals names, resulting in 2 pairs of identical input signals, namely *Horse* - *Zebra* and *Hawk* - *Owl*. The map was trained for 10000 iterations with an initial learning rate of 0.8 and an initial neighbourhood range of 3.5. In spite of the fact, that the map presents a topographic ordering of the input signals (e.g. all birds are mapped around the upper right corner of the map), cluster boundaries are not easily observable from this standard representation, unless one has sufficient prior knowlege concerning the input data. Just to give one example, the *Eagle* seems to be as closely related to the *Hawk* as to the *Cow*.

Contrary to that, the corresponding adaptive coordinate representation as provided in Figure 1b.) clearly shows the distinct clusters learned by the map: the birds are separated from all other animals in the upper right corner, the cluster itself being substructured into two groups consisting

	Attribute	Dove	Hen	Duck	Goose	Owl	Hawk	Eagle	Fox	Dog	Wolf	Cat	Tiger	Lion	Horse	Zebra	Cow
is	small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
likes to	hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	fly	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Table 1: Input Data Set: Animals

a.) b.)

Figure 1: $6 \times 6 SOM$: a.) Standard Representation b.) AC Representation

of the hunting birds *Hawk* and *Eagle* and the non-hunting birds *Duck, Hen, Goose* and *Dove*. *Zebra* and *Cow* on the lower right corner are separated from both the birds and the big hunters *Tiger* and *Lion*. In Figure 1b.) the high similarity of various groups of animals results in an almost overlapping graphical representation. This, however, is merely an inconvenience of printed figures. It is straight-forward to allow the user to zoom in the output space in order to reveal the necessary details of similarity visualization.

On the other hand, Figure 2 depicts the corresponding CC representation with a set of three thresholds being set to $t_1 = 0.2$, $t_2 = 0.26$ and $t_3 = 0.36$. The utilization of these thresholds results in visualization of increasing connectivity by means of grey-scale connections with increasing darkness. The extremes are marked by 'non-connection' in case $d_{ij} \geq t_3$ and 'black-connection' in case $d_{ij} < t_1$. Again the cluster boundaries are clearly visible from the resulting representation, with the birds being clearly separated from the mammals.

4 Discussion

Adaptive Coordinates and Cluster Connections both provide a method to visualize and detect the clusters learned by a self-organizing map as an extension to both the standard training process and standard fixed grid representation. The resulting representations allow a straight-forward analysis of the inherent structure and intuitive assignment of clusters within the input data. Still the robustness of the standard SOM model is not sacrificed, since both approaches work as add-ons to the basic training and visualization process. Thus both AC and CC can be combined to obtain

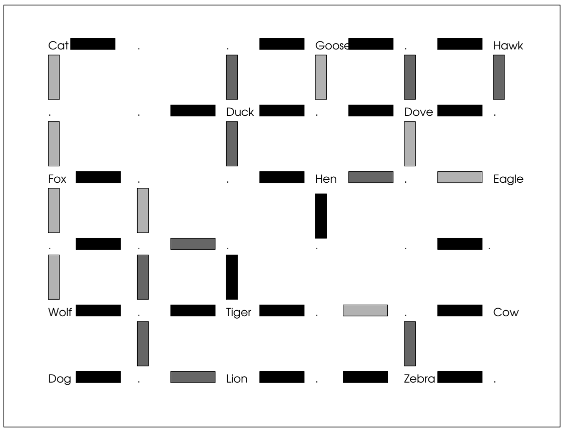

Figure 2: $6 \times 6 SOM$: CC Representation

additional benefits in exploratory data analysis. Furthermore, the computational cost to obtained the enhanced visualizations is neglectable, consisting in simple post-processing for CC and in the adaptation of a pair of adaptive coordinates per unit in every training cycle for the AC approach.

Although both techniques address the same problem, the resulting representations offer different possibilities serving different purposes. With AC the structure of the input data as analyzed by the resulting mapping is clearly visible as a set of clusters within the map. Each cluster in turn may consist of a set of sub-clusters providing a finer granularity for exploratory data analysis. On the other hand, the clusters themselves are organized in the map corresponding to their mutual similarity, i.e. similar clusters are located more closely to each other than distinct ones. By mirroring the movements of weight vectors through input space, the location of nodes resembles the clustering of the nodes' weight vectors thus not requiring any further interaction from the user. However, the fixed grid representation is broken and no explicit cluster assignment can be derived from the clusters obtained using the adaptive coordinates, yet it certainly serves the user in representing similarity in a highly intuitive fashion.

On the other hand, Cluster Connections allow a very flexible approach to the cluster detection and visualization process. Visualization can be achieved fully automatically, be it by creating a set of thresholds based on an analysis of the distance diagram of the map (i.e. minimum, maximum and average distances between the nodes' weight vectors) or by using the distances encountered directly to assign colors or gray-scales to represent the connections. Moreover, a variety of user interfaces offer itself to assist the user in selecting a set of thresholds manually, allowing her the creation of a hierarchy of clusters of varying granularity as well as enabling her to see the structure inherent in the input data by interactively moving the threshold values. While automatic visualization allows a quick feedback on the structures learned by the SOM, manual interaction with the thresholds and thus uncovering their influence on the growing and shrinking of connected regions, provides an indispensable tool for the exploratory analysis of the underlying input data set. Both the basic cluster structure as well as the overall structure present in the input data can be visualized using Cluster Connections. Qualitative information about the relationship between linked nodes is given by the usage of different colors or gray-scales to indicate looser and closer connections. The same applies to the overall structure, i.e. the similarity of different clusters can be visualized within the limits of a fixed grid representation by setting thresholds to create connections of different intensity between clusters. Once a set of thresholds is selected, clear cluster assignment can be given for every node based on the set of connected nodes.

5 Conclusion

In this paper we have presented and compared two enhanced visualization techniques for self-organizing maps. The development of these visualization techniques referred to as *Adaptive Coordinates* and *Cluster Connections* is motivated by the fact that detection of cluster boundaries as well as the inherent structure of the input data is not as intuitively possible in the standard output representation of self-organizing maps as it could be. Our extension enables both the visualization of similarity of input data comprising one cluster and the visualization of similarity between different clusters. While AC provides an intuitive representation without any further interaction from the user, no automatic cluster assignment can be derived directly from the resulting representation. On the other hand, CC allows strict cluster assignment for every node once a set of thresholds is selected to obtain a grid of connected nodes. Both visualization techniques combined offer an indispensable tool for the analysis of the structures encountered in high-dimensional input data.

References

[1] J. Blackmore and R. Miikkulainen. Incremental Grid Growing: Encoding high-dimensional structure into a two-dimensional feature map. In *Proc IEEE Int'l Conference on Neural Networks*, San Francisco, CA, 1993.

[2] J. Blackmore and R. Miikkulainen. Visualizing high-dimensional structure with the Incremental Grid Growing network. In *Proc Int'l Conference on Machine Learning*, Lake Tahoe, NV, 1995.

[3] B. Fritzke. Kohonen feature maps and Growing Cell Structures: A performance comparison. In *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann, San Mateo, CA, 1993.

[4] B. Fritzke. Growing Cell Structures: A self-organizing network for unsupervised and supervised learning. *Neural Networks*, 7(9), 1994.

[5] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 1982.

[6] T. Kohonen. *Self-organizing maps*. Springer-Verlag, Berlin, 1995.

[7] D. Merkl. The effect of lateral inhibition on learning speed and precision of a self-organizing map. In *Proc Australian Conf on Neural Networks*, Sydney, Australia, 1995.

[8] D. Merkl and A. Rauber. On the similarity of eagles, hawks, and cows – Visualization of similarity in self-organizing maps. In *Proc Int'l Workshop Fuzzy-Neuro-Systems'97*, Soest, Germany, 1997.

[9] R. Miikkulainen. Script recognition with hierarchical feature maps. *Connection Science*, 2, 1990.

[10] A. Rauber. Cluster Visualization in Unsupervised Neural Networks. Diplomarbeit, Technische Universität Wien, Austria, 1996.

[11] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61, 1989.

[12] A. Ultsch. Self-organizing neural networks for visualization and classification. In *Information and Classification - Concepts, Methods and Applications*, Springer Verlag, Berlin, 1993.

[13] W. Wan and D. Fraser. Multiple Kohonen self-organizing maps: Supervised and unsupervised formation with application to remotely sensed image analysis. In *Proc Australian Conference on Neural Networks*, Brisbane, Australia, 1994.