To appear in: Proc. of the ICANN-91 Helsinki

LET IT GROW – SELF-ORGANIZING FEATURE MAPS WITH PROBLEM DEPENDENT CELL STRUCTURE

Bernd FRITZKE

Institut für Mathematische Maschinen und Datenverarbeitung
Lehrstuhl für Programmiersprachen
Universität Erlangen-Nürnberg
D-8520 Erlangen, Germany

Abstract: The self-organizing feature maps introduced by T. Kohonen use a cell array of fixed size and structure. In many cases this array is not able to model a given signal distribution properly. We present a method to construct two-dimensional cell structures during a self-organization process which are specially adapted to the underlying distribution: Starting with a small number of cells new cells are added successively. Thereby signal vectors according to the (usually not explicitly known) probability distribution are used to determine where to insert or delete cells in the current structure. This process leads to problem dependent cell structures which model the given distribution with arbitrary high accuracy.

1 The Model of Kohonen

The self-organizing feature maps introduced by T. Kohonen [1] are one of the most successful models in the area of unsupervised learning. It has been used for vector quantization [2], biological modelling [3] and speech recognition [4]. In an extended version Ritter and Schulten used it for learning robot arm movements in an unsupervised fashion [5].

Kohonen's model builds up a mapping from the n-dimensional vector space of real numbers $\Re^n$ to a usually two-dimensional array S of cells. Each cell is given a virtual position in $\Re^n$. This is done by attaching an n-dimensional *position vector* $\text{pos}(c)$ to every cell c. An arbitrary n-dimensional vector x is then mapped onto the cell which lies nearest to it in $\Re^n$. This is the cell bmu (best matching unit) which satisfies

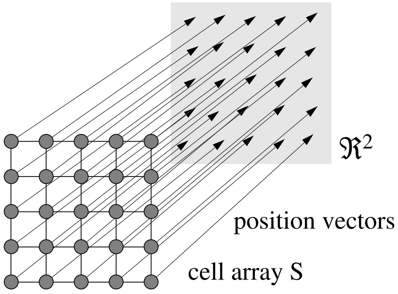
$$\| bmu - x \| \leq \| c - x \| \quad \forall c \in S \tag{1}$$

Thereby denotes $\| \, . \, \|$ the Euclidean Norm. We will sometimes denote the position vector of a cell as its *position* and changes of this vector as *movements* of the cell. However, one should keep in mind that no physical movements take place. The cells remain unchanged in the structure S.

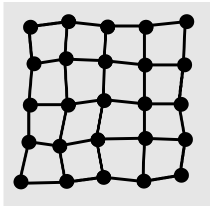
Let now P(X) be an unknown probability distribution on $\Re^n$ from which we are able to take arbitrary many sample vector. The purpose of the self-organization process described by Kohonen is to find values for the position vectors such that the resulting mapping is topology-preserving and distribution-preserving.

By a *topology-preserving* mapping we mean that adjacent vectors in $\Re^n$ are mapped on adjacent (or identical) cells in the array. Furthermore adjacent cells in the array have similar position vectors in $\Re^n$.

By a *distribution-preserving* mapping we mean, that for a random vector according to P(X) each cell has the same probability of being the target of the mapping. Stated otherwise this means, that the relative density of position vectors in $\Re^n$ approximates the probability density of P(X).



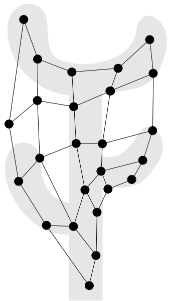
a) P(X) is a uniform distribution in the shaded sub area of $\Re^2$. Also the position vectors are distributed rather uniform in the concerning area. Furthermore neighboring cells have similar position vectors.



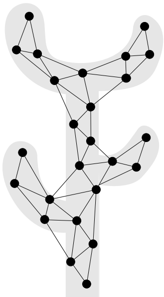
b) Usually the situation in (a) is figured as shown here: The cells are drawn in the vector space at the positions, their associated vectors point to. Neighboring cells are connected by edges.

Figure 1: Example of the result of Kohonen's method

An example of the result of Kohonen's method is shown in Fig. 1. In this case the structure S is a square array of 25 cells and P(X) is a uniform distribution in a square sub area (shaded) of $\Re^2$. In the example in Fig. 1 the previously mentioned conditions for the mapping are fulfilled satisfactory. Unfortunately this is not always the case, as Fig. 2a) shows.



a) Usual rectangular cell structure (after [6])



b) Imaginary cell structure which models this distribution better

Figure 2: Result of the self-organizing process for a structured probability distribution

Here P(X) is a uniform distribution in the cactus-shaped area. Again the structure S is a 5x5 array. Now the correspondence between the cell positions and the underlying distribution is not as good as before:

- Some cells are at locations, where the probability density is zero.
- Some cells which are direct neighbors in the array have rather distant positions in $\Re^2$.

It is obvious, that in this case no rectangular structure can be positioned such that the resulting mapping is both topology- and distribution-preserving. The situation changes, if we allow arbitrary formed two-dimensional structures. Fig. 2b) shows a cell structure with the same number of cells which models the cactus-shaped distribution much better: Here all neighboring cells have similar positions in the $\Re^2$ and cells are only at positions where the probability density is not zero. The problem is, that we need a different cell structure for every probability distribution P(X). Moreover, for generally P(X) is not known explicitly we must be able to build up the cell structures solely by evaluating sample vectors according to P(X).

2 Growing Cell Structures

We want to develop two-dimensional cell structures and distribute them in $\Re^n$ (in the sense of chapter one) such that the resulting mapping is topology-preserving and distribution-preserving with respect to an n-dimensional probability distribution P(X). Fig. 3 shows the basic idea: We start with a triangle of cells at random positions in $\Re^n$. This triangle is distributed as good as possible over the

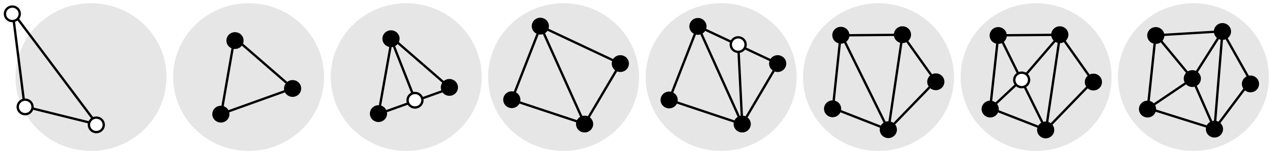


Figure 3: Development of a cell structure for a circular probability distribution

area where P(X) differs from zero. Then a new cell is inserted by splitting one of the edges in the structure. This new cell is connected to the cell opposite to the splitted edge so that again we have a structure consisting of (now 2) triangles. The structure is redistributed. Another cell is inserted and so on.

We now have to answer two questions: (i) How can we (re)distribute a cell structure such that the relative density of cells approximates the probability density of P(X)? (ii) Where should new cells be inserted?

(i) The redistribution is done analogously to the method of Kohonen: We generate an input vector according to P(X). Then we move the nearest cell (the *bmu* , see chapter 1.) and its direct topological neighbors towards this vector (see Fig. 4). The amount of movement is a fraction ϵ_{bmu} and $\epsilon_{neighbor}$

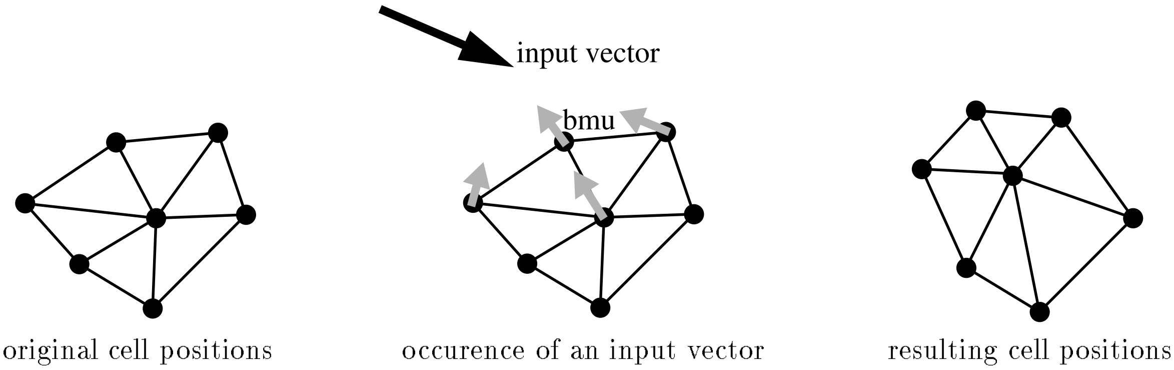


Figure 4: One step during the redistribution: the *bmu* and its direct topological neighbors are moved towards the position of the input vector (also denoted as *input signal*).

of the total distance, respectively. This step, consisting of input vector generation and cell movement, is repeated for a fixed number $n_{distribution}$ of times. According to [1] the positions of the cells thereby tend to change towards an equilibrium state in which the cell distribution roughly approximates P(X).

(ii) During the redistribution information can be compiled about the place where to insert the next cell: Each time when a cell c is *bmu*, we add the distance between c and the input vector to an attached "error" variable err(c). By this means we are able to determine the error that is caused by every cell over the time. A high error value indicates, that in the surrounding of the corresponding cell in $\Re^n$ the density of cells is comparatively low. Therefore after the redistribution phase we search for the cell bs ("black sheep") with maximum error value. Then we determine the furthest of the direct neighbors of bs, say a cell f, and split the edge which connects both: a new cell c_{new} is inserted halfway between bs and f in $\Re^n$. Thus its position vector is initialized as follows:

$$\mathrm{pos}(c_{new}) = 0.5\,(\mathrm{pos}(bs) + \mathrm{pos}(f)) \tag{2}$$

The cell $c_{new}$is connected with the surrounding cells in such a way, that again a plane structure of triangles is built. In doing so two different cases can occur depending on whether the split edge lies on the rim of the structure S or not (see Fig. 5). The error variable of c_{new} is initialized as:

$$\mathrm{err}(c_{new}) \;=\; \frac{1}{d+1}\sum_{i=1}^{d}\mathrm{err}(neighbor_i) \tag{3}$$

a) Either bs or f (or both) lies inside the structure S. The edge which connects them, is member of two triangles. This edge is removed. Then c_{new} is connected to bs, f, and the opposite corners of both of the two triangles. Thus c_{new} gets four direct neighbors.

b) Both bs and f lie on the rim of S. In this case the edge connecting them is part of only one triangle. Again this edge is removed. c_{new} is connected to bs, f, and the opposite corner of the triangle and gets by that three direct neighbours.

Figure 5: Insertion of a new cell between bs and f

Thereby d is the number of direct neighbors of c_{new} (three or four as illustrated in Fig. 5) and $neighbor_i$ is the i-th of these cells. It is desirable to keep the total error of the involved cells constant. Therefore the error values of the direct neighbors of c_{new} are diminished by the appropriate amount:

$$\mathrm{err}(neighbor_i) \;=\; \frac{d}{d+1}\,\mathrm{err}(neighbor_i) \quad \forall\, i \in \{1, 2 \ldots d\} \tag{4}$$

A consequence of this error redistribution is, that the direct neighbor cells (among them the current bs) will less likely become the "black sheep" the next time. In the long run this will lead to a cell density which approximates the probability density of P(X). Fig. 6 shows an example of the resulting structures. In this case the probability distribution consisted of two separate circular areas with uniform probability density.

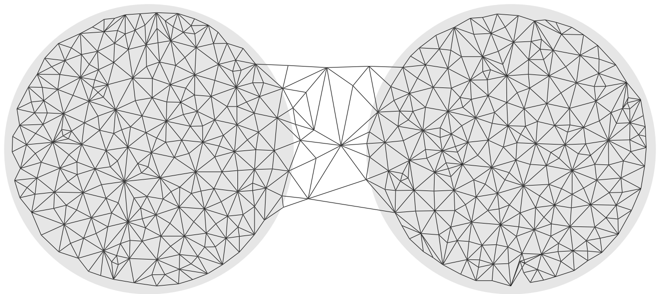


Figure 6: Cell structure for a probability distribution with disconnected regions after 25000 input vectors (simulation parameters: $n_{distribution} = 50$, $\epsilon_{bmu} = 0.1$, $\epsilon_{neighbor} = 0.003$)

3 Selective Removal of Cells

So far we have only extended our cell structures. A necessary consequence thereof is, that each cell is connected directly or indirectly to every other. If we have to model a distribution with disconnected regions of positive probability density (as in Fig. 6) a problem occurs: either we get some cells at positions where P(X) equals zero or we get some very long edges, which indicate that the mapping is not really topology-preserving in the corresponding area of $\Re^n$.

A solution can be found by introducing the removal of cells from the structure. An obvious criterium for a cell to be removed would be, that it has a position in an area of the $\Re^n$, where the probability density is zero. However, since P(X) is not given explicitly, we can only decide this by evaluating sample vectors according to P(X). The general idea is: the longer a cell was not bmu , the more likely it lies in an area of $\Re^n$ where the probability density of P(X) is zero.

Let now n be the current number of cells in the structure. We assume, that P(X) is uniform in a sub area B of $\Re^n$ and that the cells are uniformly distributed in B. Let c be an arbitrary cell with a position somewhere in B. A new cell has been generated after every $n_{distribution}$ input vectors. Thus k input vectors "ago" the number of cells was $n - q$ with

$$q = \left\lfloor \frac{k}{n_{distribution}} \right\rfloor \tag{5}$$

Thus the probability for an arbitrary cell to exist since k input vectors is

$$p_1 = \frac{n-q}{n} \tag{6}$$

The probability for one of n cells to be not bmu for $n_{distribution}$ successive input vectors is:

$$\left(\frac{n-1}{n}\right)^{n_{distribution}} \tag{7}$$

We have to take in account, that the number of cells is increased every $n_{distribution}$ input vectors. Then the probability, that a cell that *does exist* since k successive input vectors has never been bmu is

$$p_2 = \prod_{i=1}^{q} \left(\frac{n-i}{n-i+1}\right)^{n_{distribution}} = \left(\frac{n-q}{n}\right)^{n_{distribution}} \tag{8}$$

The probability that a cell exists since k input vectors *and* has never been bmu is then

$$p_3 = p_2 p_1 = \left(\frac{n-q}{n}\right)^{n_{distribution}+1} \tag{9}$$

We want to be sure to a degree p_{sure} not to remove a cell lying in B. Since we have n cells the relation

$$\overline{p_3}^{n} > p_{sure} \qquad \text{respectively} \qquad \left(1 - \left(\frac{n-q}{n}\right)^{n_{distribution}+1}\right)^{n} > p_{sure} \tag{10}$$

must be satisfiesd. If we replace q with $k/n_{distribution}$ and resolve with respect to k, this leads to:

$$k > n\, n_{distribution} \left(1 - \left(1 - p_{sure}{}^{1/n}\right)^{1/(n_{distribution}+1)}\right) \tag{11}$$

Now define k_{remove} as the smallest integer satisfying this condition. We then can realize the removal of cells by counting the input vectors and storing with every cell the number of the input vector for which it was bmu most recently. We periodically compute k_{remove} for the current number n of cells and check if the difference between the current number of input vectors and the number stored with a cell exceeds k_{remove}. In this case we remove the cell and all adjacent edges. If this is necessary to get again a structure of triangles we remove further nodes and edges. The sequence in Fig. 7 illustrates that now a wide range of distributions can be modelled properly.

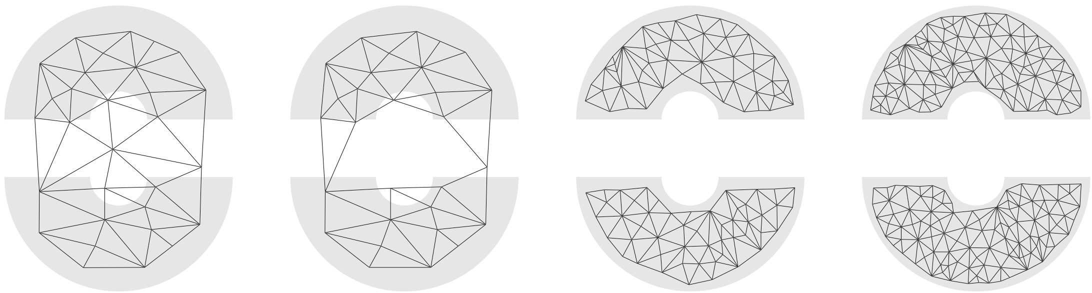


Figure 7: Sequence of cell structures obtained with selective removal of cells. The structures shown consist of 30, 29, 100 and 200 cells (simulation parameters: $n_{distribution} = 50$, $\epsilon_{bmu} = 0.1$, $\epsilon_{neighbor} = 0.005$, $p_{sure} = 0.9995$).

4 Conclusion And Further Research

We presented a self-organizing scheme to construct two-dimensional cell structures specially adapted to a given probability distribution. Our model improves the underlying model of Kohonen in several important aspects:

- The cell structure has not to be specified in advance. This is very important in the frequent case, that the probability distribution of the input vectors is unknown.
- The problem dependent cell structures developed by our approach are capable of modeling properly a far wider range of distributions than any array with predefined size and structure.
- The accuracy of the feature maps we construct can be increased nearly continuously by simply letting the self-organization process continue and thereby insert successively more cells into the structure.

We succesfully applied the principles described in this contribution to cluster analysis [7] and optimization problems [8].

In currently ongoing studies we explore the possibility of constructing three and higher-dimensional cell structures with the same principles. Further investigations are being made to directly compare our approach with the model of Kohonen for various applications.

Acknowledgements

I'd like to thank my colleagues at the chair of programming languages for their very helpful hints and proposals while I was preparing this manuscript.

References

[1] Teuvo Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics* 43 (1982), 59–69.

[2] Teuvo Kohonen, "Learning Vector Quantization for Pattern Recognition," Helsinki University of Technology, TKK-F-A601, 1986.

[3] K. Obermayer, H. Ritter & K. Schulten, "Large-Scale Simulations of Self-organizing Neural Networks on Parallel Computers: Application to Biological Modelling," *Journal of Parallel Computation* (1990).

[4] Teuvo Kohonen, K. Mäkisara & T. Saramäki, "Phonotopic maps, – insightful representation of phonological features for speech recognition," *Proc. 7th Int. Conf. on Pattern Recognition*, Montreal (1984).

[5] H. Ritter, T. M. Martinetz & K. Schulten, "Topology-Conserving Maps for Learning Visuo-Motor-Coordination," *Neural Networks* 2 (1989), 159–168.

[6] Teuvo Kohonen, *Self-organization and associative memory*, Springer, Heidelberg, 1984.

[7] Bernd Fritzke, "Unsupervised Clustering with growing Cell Structures," *to appear in: Proc. of IJCNN-91*, Seattle (1991).

[8] Bernd Fritzke, "Solving large Travelling-Salesman Problems with an adaptive growing Ring," (to be published).