Quantifying the Neighborhood Preservation of Self-Organizing Feature Maps

Hans-Ulrich Bauer and Klaus R. Pawelzik

Abstract— **Neighborhood preservation from input space to output space is an essential element of such self-organizing feature maps as the Kohonen map. However, a measure for the preservation or violation of neighborhood relations, which is more systematic than just visual inspection of the map, has been lacking. We show that a topographic product P, first introduced in nonlinear dynamics, is an appropriate measure in this regard. It is sensitive to large-scale violations of the neighborhood ordering, but does not account for neighborhood ordering distortions caused by varying areal magnification factors. A vanishing value of the topographic product indicates a perfect neighborhood preservation; negative (positive) values indicate a too small (too large) output space dimensionality. In a simple example of maps from a 2-D input space onto 1-D, 2-D, and 3-D output spaces we demonstrate how the topographic product picks the correct output space dimensionality. In a second example we map 19-D speech data onto various output spaces and find that a 3-D output space (instead of 2-D) seems to be optimally suited to the data. This is in agreement with a recent speech recognition experiment on the same data set.**

I. Introduction

MAPS constitute an important class of neural information processing systems, both natural and artificial [1]–[3]. They project a pattern in an input space onto a position in an output space, in this way coding the information as the location of an activated node in the output space. A few examples of maps in the nervous system are retinotopic maps in the visual cortex [4], tonotopic maps in the auditory cortex [5], and maps from the skin onto the somatosensoric cortex [6]. Of these, retinotopic maps have been modeled in a number of contributions, including aspects such as orientation preference and ocular dominance [7]–[10]. In the domain of artificial neural networks, applications of maps include motor control tasks for robot arms [11] and/or phoneme recognition (using the "learning vector quantizer" refinement (LVQ) of the basic feature map [12]).

An essential property common to all these maps is the preservation of neighborhood relations. Nearby features in the input space are mapped onto neighboring locations in the output space. It is this aspect of maps which serves as an organizing principle for map formation algorithms, one of which, the Kohonen algorithm, we will discuss in the second section.

Manuscript received June 19, 1991; revised December 5, 1991. This work was supported by the Deutsche Forschungsgemeinschaft (Sonderforschungsbereich 185 "Nichtlineare Dynamik," TP A10).

The authors are with the Institut für Theoretische Physik and SFB "Nichtlineare Dynamik," Universität Frankfurt, Robert-Mayer-Str. 8-10, W-6000 Frankfurt am Main 11, Germany.

IEEE Log Number 9106423.

However, it is difficult to quantitatively characterize how "good" this preservation actually is. In fact, large-scale neighborhood violations in maps are usually detected by visually inspecting the map; a method which is restricted to sufficiently low-dimensional input spaces (not to mention its arbitrariness). The main point of our contribution is to close this diagnostic gap by introducing a topographic product as a measure of the preservation of neighborhood relations. The topographic product has already proved useful for the analysis of embeddings for chaotic attractors in nonlinear dynamics [13]. A strange attractor can only be embedded if there is a continuous map with a continuous inverse from the original phase space to the reconstructed space. In the third section we will give a detailed discussion of the different terms entering the product and we will show how this measure cannot be fooled by local stretchings of the map owing to varying input pattern densities.

We then continue in the fourth section to a very simple example for a self-organizing feature map (SFM)—the mapping from a square with constant stimulus density onto an output space consisting of a line, a square, or a cube of neurons. This very simple example includes the essential effects of a dimensionality mismatch between input space and output space, and has been analyzed in detail by Ritter and Schulten [14]. Using the topographic product as a measure, we show how the matching output space dimensionality for this problem can be found in an unambiguous way.

In the last two sections we then turn to the question of why an optimal preservation of neighborhood relations means an advantage in applications, and is not merely a theoretical excursion. To this purpose we first mimic a speech recognition experiment with SFM's. The recognition is based on the classification of trajectories in the output space of the map. We show how the recognition performance can severely degrade if there is a dimension mismatch between input space and output space. Then we analyze real speech data and identify the optimal output space dimension. This turns out to be in agreement with a comparable speech recognition experiment [15], where Kohonen maps of different output space dimension have been used for preprocessing. The recognition was performed by classifying the resulting trajectories in the output space with a dynamic time warping algorithm.

II. Kohonen Algorithm for Self-Organizing Feature Maps

The Kohonen algorithm is a well established learning rule for self-organizing feature maps and can easily be imple-

1045–9227/92$03.00 © 1992 IEEE

mented. It has been described in numerous publications, in particular in the book by Kohonen [2]. Here it should suffice to give only a short account of the algorithm, which will provide the notation used in the subsequent section. In order to simplify comparisons between different papers, we adhere in this paper to the notation of Ritter *et al.* [3], [14].

The algorithm describes a map Φ from an input space V into an output space A. The output space consists of nodes j, which are arranged in some topological order (e.g. as nodes on a line or as vertices of a two-dimensional lattice). For each position j in A there is a pointer, $\boldsymbol{w}_j$, into V, which can be regarded as the center of the receptive field associated with the neuron at j. If a stimulus $\boldsymbol{v}$ occurs in the input space, it is mapped onto that neuron i in A, the "receptive field pointer" $\boldsymbol{w}_i$ of which lies closest to $\boldsymbol{v}$, i.e.,

$$i: \quad d^V(\boldsymbol{w}_i, \boldsymbol{v}) = \min_{j \in A} d^V(\boldsymbol{w}_j, \boldsymbol{v}). \tag{1}$$

Here $d^V(\boldsymbol{w}_j, \boldsymbol{v})$ means the distance in the input space V between $\boldsymbol{w}_j$ and $\boldsymbol{v}$.

During the learning phase, the map Φ is formed by sucessive adjustments of the vectors $\boldsymbol{w}_j$. During one learning step, a stimulus $\boldsymbol{v}$ is (randomly) chosen and mapped onto an output node i according to (1). Then both the pointer $\boldsymbol{w}_i$ and all the pointers $\boldsymbol{w}_j$ of nodes in the vicinity of i are shifted a small step toward $\boldsymbol{v}$:

$$\delta \boldsymbol{w}_j = \epsilon h^0_{j,i}\big(d^A(j,i)\big)(\boldsymbol{v} - \boldsymbol{w}_j) \qquad \forall j \in A. \tag{2}$$

The function $h^0_{j,i}$ determines the size of the vicinity of i which takes part in the learning. It depends on the distance $d^A(j,i)$ between output nodes j and i, measured in the output space. The function $h^0_{j,i}\big(d^A(j,i)\big)$ has a maximum at $d^A(i,i) = 0$, and decreases with increasing $d^A(j,i)$. A typical choice for $h^0_{j,i}$ is

$$h^0_{j,i}(d) = e^{-d^2/2\sigma^2}. \tag{3}$$

The complete learning phase consists in a (random) initialization of the $\boldsymbol{w}_j$, followed by a number of the above-described learning steps. For the learning to converge, it is helpful to slowly decrease the step-size $\epsilon(t)$ as well as the width $\sigma(t)$ of $h^0_{j,i}$ during the learning process. Even though not optimal, an exponential decay,

$$\epsilon(t) = \epsilon_0 e^{-t/\tau_\epsilon} \tag{4}$$

$$\sigma(t) = \sigma_0 e^{-t/\tau_\sigma}, \tag{5}$$

most often turns out to be sufficient [16].

The final map is usually visualized in the input space. All pointers $\boldsymbol{w}_j$ are shown as dots, and pointers of neighboring nodes are connected with lines. An undistorted graph without foldings, as in Fig. 1, is the signature of a map which preserves neighborhood relations.

So far, we have not elaborated on the spaces V and A. As far as V is concerned, we assume a D^V dimensional, continuous space. The distance measure $d^V(\boldsymbol{v}, \boldsymbol{v}')$ in V has already been used in (1). The output space A is usually assumed discrete, not only as a consequence of the discrete nature of neurons, but also because only a finite, and therefore discrete, map can

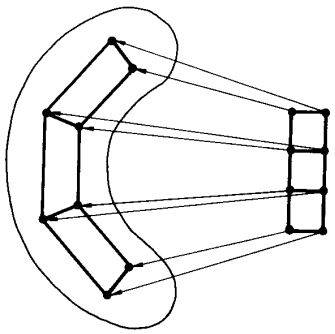


Fig. 1. The self-organizing feature map can be visualized by marking pointer positions in the input space as dots, and by connecting such pointers, which belong to nearest neighbor nodes in the output space. Compared with this figure, in a usual visualization all auxiliary lines, including the whole output space, are not shown, but only the bended ladder (or a corresponding figure) in the input space.

be simulated on a computer. The nodes in A are assumed to be ordered like a D^A dimensional lattice. The distance $d^A(i,j)$ is assumed to be the Euclidean distance on this lattice. Modeling nervous maps from some peripheral sensory input space onto a cortical area, one chooses $D^A = 2$, since cortical areas are quasi-two-dimensional. However, in nonbiological applications, e.g. in speech recognition or robot control, D^A can be chosen freely. In particular, it can be optimized with regard to the preservation of neighborhood relations. It is important to note that $D^A_{\text{opt}} \neq D^V$ in general, since the input stimuli need not fill the whole of V, but can lie in subspace with lower dimension $\tilde{D}^V$. Clearly an output space dimensionality $D^A < \tilde{D}^V$ will be suboptimal, because the output space has to fold itself into the input space. This rather sloppy expression will become self-evident when we consider the examples of Section IV. On the other hand one might be tempted to chose a large D^A, perhaps even $D^A = D^V > \tilde{D}^V$. This choice can turn out suboptimal as well, considering that folding of the input space into the output space can also occur. This case is visualized in Fig. 2, which shows a map of a line onto a square.

We should note here that other map formation algorithms have been proposed which induce nontrivial output space topologies [17], [18]. As long as the output spaces provide distance measures, the following discussion of the topographic product applies to these cases as well.

III. Topographic Product

The topographic product is a measure of the preservation of neighborhood relations in maps between spaces of possibly different dimensionality. It was introduced (under the name "wavering product") in the context of nonlinear dynamics and time series analysis [13]. There it was used for purposes similar to those in this paper: it served as a tool to select optimal embedding parameters (dimension and delay time) for the reconstruction of chaotic attractors from one-variable time series via delay coordinates. We will now derive the

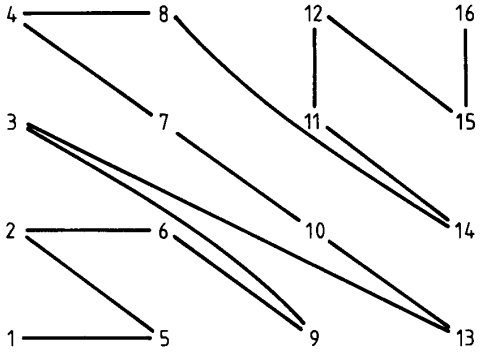


Fig. 2. Map from a one-dimensional input space onto a square output space. In this figure we show not the usual pointer positions in the input space, but rather the ordering of pointers on the line, shown in the output space. Connected nodes in the output space are nearest neighbors in the input space. The input space "folds" itself into the output space in order to cope with the dimension mismatch. This is exactly the inverse situation of the map of a square onto a line, which gives the peano curve picture as in Fig. 4(a).

topographic product step by step and explain the reasoning for each part of the formula in detail.

First we introduce a notation for nearest-neighbor indices. Let $n_k^A(j)$ denote the kth nearest neighbor of node j, with the distances measured in the output space, i.e.,

$$\begin{aligned} n_1^A(j): \quad & d^A\left(j, n_1^A(j)\right) = \min_{j' \in A \setminus \{j\}} d^A(j, j') \\ n_2^A(j): \quad & d^A\left(j, n_2^A(j)\right) = \min_{j' \in A \setminus \left\{j, n_1^A(j)\right\}} d^A(j, j') \\ & \vdots \end{aligned} \tag{6}$$

In the same way let $n_k^V(j)$ denote the kth nearest neighbor of j, but with the distances measured in the input space between $\boldsymbol{w}_j$ and $\boldsymbol{w}_{n_k^V(j)}$:

$$\begin{aligned} n_1^V(j): \quad & d^V\left(\boldsymbol{w}_j, \boldsymbol{w}_{n_1^V(j)}\right) = \min_{j' \in A \setminus \{j\}} d^V(\boldsymbol{w}_j, \boldsymbol{w}_{j'}) \\ n_2^V(j): \quad & d^V\left(\boldsymbol{w}_j, \boldsymbol{w}_{n_2^V(j)}\right) = \min_{j' \in A \setminus \left\{j, n_1^V(j)\right\}} d^V(\boldsymbol{w}_j, \boldsymbol{w}_{j'}) \\ & \vdots \end{aligned} \tag{7}$$

Using this nearest-neighbor indexation, we next define the ratios

$$Q_1(j,k) = \frac{d^V\left(\boldsymbol{w}_j, \boldsymbol{w}_{n_k^A(j)}\right)}{d^V\left(\boldsymbol{w}_j, \boldsymbol{w}_{n_k^V(j)}\right)} \tag{8}$$

$$Q_2(j,k) = \frac{d^A\left(j, n_k^A(j)\right)}{d^A\left(j, n_k^V(j)\right)}. \tag{9}$$

From this definition we will have $Q_1(j,k) = Q_2(j,k) = 1$ only if the nearest neighbors of order k in the input and the output space coincide. Any deviation of Q_1 and Q_2 from 1 points to a violation of the nearest-neighbor ordering because of the map. However, this is too sensitive a measure for the preservation of neighborhood relations, since a locally stretched map can preserve neighborhood relations

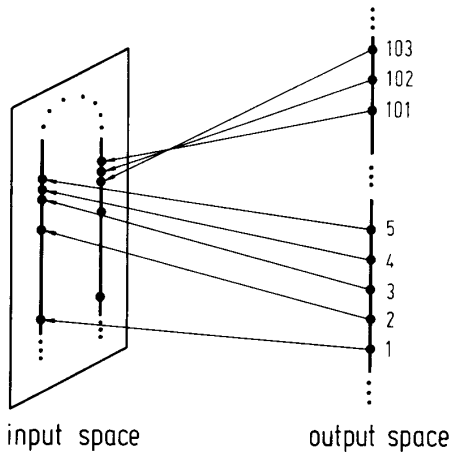


Fig. 3. Part of a map from a two-dimensional input space onto a one-dimensional output space. Two pieces of the folded output space line are shown (nodes 1–5 and nodes 101–103) which are nearby in the input space. The nearest neighbors of node 3, measured in the input space, are $4, 5, 2, 102, \cdots$. Of these, node 102 lies far from node 3 in the output space, causing a deviation of the topographic product from the value 1.

even though the nearest-neighbor ordering is violated. A local stretch of the map can be induced by a gradient in the input stimulus density. Such a situation is shown in Fig. 3. Consider there the nearest neighbors of node 3 in input space and output space. The nearest neighbors

$$n_1^V(3) = 4$$
$$n_1^A(3) = 4$$

coincide (not regarding degeneration), but not so the second nearest neighbors:

$$n_2^V(3) = 5$$
$$n_2^A(3) = 2.$$

Therefore

$$Q_1(3,2) = \frac{d^V(\boldsymbol{w}_3, \boldsymbol{w}_2)}{d^V(\boldsymbol{w}_3, \boldsymbol{w}_5)} > 1, \tag{10}$$

with $Q_2(3,2) < 1$ analogously. On the other hand, the pointers in the input space form a line in the same way as the nodes in the output space; i.e., the neighborhood relations for nodes 2, 3, 4, and 5 are preserved.

This problem can be overcome by multiplying the $Q_\nu(j,k)$ for all orders k. With proper normalization this gives

$$P_1(j,k) = \left(\prod_{l=1}^{k} Q_1(j,l)\right)^{1/k} \tag{11}$$

$$P_2(j,k) = \left(\prod_{l=1}^{k} Q_2(j,l)\right)^{1/k}. \tag{12}$$

For the new variables P_1 and P_2, we have

$$P_1(j,k) \geq 1$$
$$P_2(j,k) \leq 1.$$

In P_1 and P_2 a different ordering of nearest neighbors is canceled, as long as the first k nearest neighbors of j in V

and A coincide (not regarding their order). Picking up the example from above, we find

$$P_1(3,3) = \left(\frac{d^V(\boldsymbol{w}_3, \boldsymbol{w}_4) d^V(\boldsymbol{w}_3, \boldsymbol{w}_2) d^V(\boldsymbol{w}_3, \boldsymbol{w}_5)}{d^V(\boldsymbol{w}_3, \boldsymbol{w}_4) d^V(\boldsymbol{w}_3, \boldsymbol{w}_5) d^V(\boldsymbol{w}_3, \boldsymbol{w}_2)} \right)^{1/3} = 1, \tag{13}$$

with $P_2(3,3) = 1$ analogously. The P_1, P_2 are sensitive only to severe neighborhood violations, e.g. if two pointers are found to be closeby in the input space but their corresponding roots lie far apart in the output space. In Fig. 3, this is the case for node 3 and its fourth-nearest neighbor $n_4^V(3) = 102$, which have nearby pointers owing to a distortion of the map. What are the effects on P_1 and P_2? We now find

$$\begin{aligned} P_1(3,4) = & \\ & \left(\frac{d^V(\boldsymbol{w}_3, \boldsymbol{w}_4)\, d^V(\boldsymbol{w}_3, \boldsymbol{w}_2)\, d^V(\boldsymbol{w}_3, \boldsymbol{w}_5)\, d^V(\boldsymbol{w}_3, \boldsymbol{w}_1)}{d^V(\boldsymbol{w}_3, \boldsymbol{w}_4)\, d^V(\boldsymbol{w}_3, \boldsymbol{w}_5)\, d^V(\boldsymbol{w}_3, \boldsymbol{w}_2)\, d^V(\boldsymbol{w}_3, \boldsymbol{w}_{102})} \right)^{\frac{1}{4}} \\ & = \left(\frac{d^V(\boldsymbol{w}_3, \boldsymbol{w}_1)}{d^V(\boldsymbol{w}_3, \boldsymbol{w}_{102})} \right)^{\frac{1}{4}} \\ & > \approx 1, \end{aligned} \tag{14}$$

$$\begin{aligned} P_2(3,4) &= \left(\frac{d^A(3,4) d^A(3,2) d^A(3,5) d^A(3,1)}{d^A(3,4) d^A(3,5) d^A(3,2) d^A(3,102)} \right)^{1/4} \\ &= \left(\frac{d^A(3,1)}{d^A(3,102)} \right)^{1/4} \\ &\ll 1. \end{aligned} \tag{15}$$

That is, we find a small deviation from 1 for P_1 and a very strong deviation for P_2.

Constant magnification factors of the map do not change next-neighbor orderings; therefore the products P_1 and P_2 have the important property of being insensitive to constant gradients of the map. Spatially varying stimulus densities induce spatially varying magnification factors of the Kohonen map, which correspond to nonvanishing second derivatives. A change of the local magnification factor may induce changes in the next-neighbor orderings. As long as these second-order contributions average out locally, the products $P_1(k)$ and $P_2(k)$ remain close to 1 individually if one multiplies up to sufficient values of k (in the above example, we had $P_1(3,3) = 1$, even though $P_1(3,2) > 1$). For the case where second derivatives do not average out locally, we combine P_1 and P_2 multiplicatively in order to find

$$P_3(j,k) = \left(\prod_{l=1}^{k} Q_1(j,l) Q_2(j,l) \right)^{1/2k} \tag{16}$$

This last step has the effect that, as a consequence of the inverse nature of P_1 and P_2, the contributions of curvatures are suppressed while violations of neighborhoods are detected by $P_3 \neq 1$ (in the above example, we had $P_1(3,4) \approx 1$, while $P_2(3,4) \ll 1$). A further important reason for this definition of P_3, however, is that $P_1 > 1/P_2$ if the input space folds itself into the output space, and $P_1 < 1/P_2$ if the output folds itself into the input space (as in Fig. 2). In other words, the deviation of P_3 above or below 1 indicates whether the embedding dimension D^A is too large or too small, respectively.

All that remains is to define a suitable averaging of $P_3(j,k)$. To this purpose there are several ways imaginable. First, one could look pointwise for all nodes j in order to obtain a spatially resolved estimate of topology violations. Second, one could build histograms in order to obtain the probability for strong neighborhood violations together with an estimation for the global variance of P_3. In clear cases, such as those depicted in Figs. 4–8, we find broad distributions with a cutoff at $P_3 = 1$ and a tail below or above this value for the cases with $D^A < D^V$ or $D^A > D^V$, respectively. For $D^A = D^V$ the distribution becomes centered at $P_3 = 1$ with a small variance. The most simple way of averaging, however, consists in summing over all nodes and all neighbor orders, a method which suffices for most practical purposes. Being only interested in deviations from 1, we average the logarithm of P_3 and finally arrive at the full-blown formula for the topographic product P:

$$P = \frac{1}{N(N-1)} \sum_{j=1}^{N} \sum_{k=1}^{N-1} \log(P_3(j,k)). \tag{17}$$

IV. Example I: Maps of a Quadratic 2-D Input Space

We now turn to an example to make the abstract ideas of the last section more concrete, and to show the type of quantitative results application of (17) yields. We chose to make this example as simple as possible, for two reasons. First we do not want to confuse the results of dimensionality mismatch with other influence factors; second we want to check the quantitative results of the topographic product method for plausibility. The latter reason requires an example, for which an intuitive expectation for the optimal output space dimension D^A exists. This certainly is the case for maps from a quadratic input space with flat stimulus distribution onto 1-D, 2-D, or 3-D output spaces. This map can easily be visualized since the input space is only two-dimensional. We can expect that an output space with exactly the same quadratic shape would preserve the neighborhood relations best. Although the example bears no surprises, it is representative for all cases where the stimuli lie on a hypersurface in a high-dimensional input space with a very small variance in the orthogonal directions.

A variant of this example has been investigated by Ritter *et al.* with regard to the occurrence of instabilities in the map, which are driven by dimension mismatch [14]. The mismatch occurs if the input space dimension exceeds the output space dimension and if the variance of the stimuli in the additional input space dimensions exceeds a critical value.

In Fig. 4(a) an SFM of a square onto a line with $N = 256$ nodes is depicted. We see that the curve given by the connected pointer positions in the input space is very distorted in order to fill the square as densely as possible. It resembles a peano curve. In Fig. 4(b) the components $P_1(k), P_2(k)$, and $P_3(k)$ are shown for all values of k (averaged over all nodes j). We find $P_1 \geq 1$ and $P_2 \leq 1$, as discussed in the previous section. The combined product, P_3 lies well below 1 in nearly the whole range of k values. According to (17), the topo-

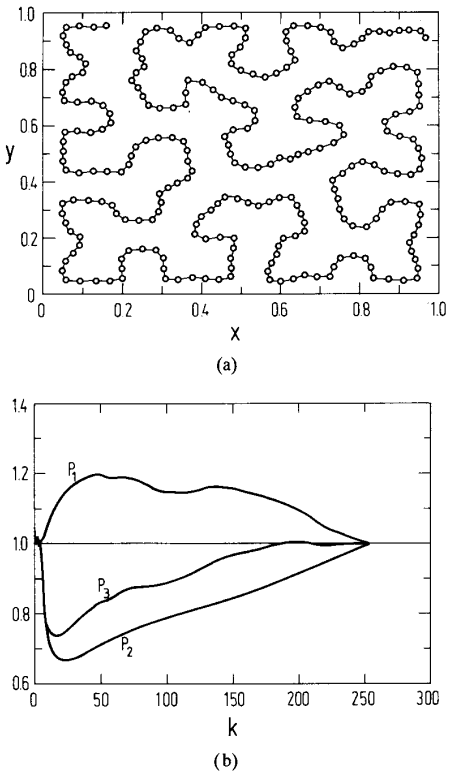


Fig. 4. (a) Map of a square input space onto a line with $N = 256$ nodes. The output space line folds itself like a peano curve in order to fill the input space as densely as possible. (b) Components $P_1(k)$, $P_2(k)$, and $P_3(k)$ of the topographic product, evaluated for the map of Fig. 4(a).

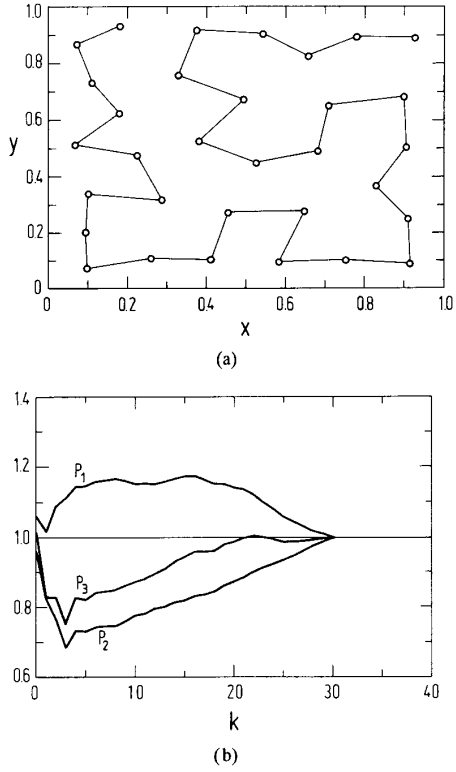


Fig. 5. (a) As in Fig. 4(a), but for $N = 32$ nodes. (b) As in Fig. 4(b) but for the map of Fig. 5(a).

graphic product P follows from this curve by taking the logarithm and averaging over k, resulting in a value of $P = -0.09026$. This value indicates an output space dimension that is too small.

The topographic product P and the shape of the $P_1(k)$, $P_2(k)$, and $P_3(k)$ curves do not depend on the number of nodes in the output space as can be seen in parts of (a) and (b) of Fig. 5. Here everything coincides with (a) and (b) in Fig. 4, the sole exception being that we have $N = 32$ output nodes only. Consequently the resulting "peano curve" is much less distorted (Fig. 5(a)). Nevertheless, the shape of the curves in Fig. 5(b) is about the same as in Fig. 4(b), with the horizontal axis rescaled linearly with the number of nodes. (The maximum neighborhood order has a value of $N - 1$.) As a consequence, the topographic product for the map of Fig. 5(a) has about the same value ($-$ 0.081) as the map of Fig. 4(a) ($-$ 0.090). As a more systematic result, we give in Table I the topographic products for nets with $N = 32$, 64, 128, and 256 nodes, averaged over four nets in each case. The values seem to converge even on a logarithmic scale, in this way justifying the heuristic averaging in (17). A more rigorous argument for finite size scaling effects of P would require a detailed (analytic) understanding of the finite size scaling behavior of the map itself.

TABLE I
TOPOGRAPHIC PRODUCT P FOR THE MAP FROM A SQUARE INPUT SPACE ONTO A LINE OF N NODES (VALUES FOR P AVERAGED OVER FOUR NETWORKS EACH)

N	P
16	-0.074 ± 0.009
32	-0.084 ± 0.003
64	-0.092 ± 0.007
128	-0.097 ± 0.005
256	-0.105 ± 0.010

Parts (a) and (b) of Fig. 6 show the map for a very elongated (64 $\times$ 4 node) output space. This long rectangle is rolled up a bit in order to fit into the square input space (comparable to a rather short line ($N = 16$)).The topographic product of -0.067 is increased relative to the 1-D output space, but still lies below 1, indicating either too small a dimension or an aspect ratio of the output space that does not fit. Parts (a) and (b) of Fig. 7 show the next case, a square (16 $\times$ 16) node output space. The perfectly regular inverse map of Fig. 7(a) indicates strongly that this output space preserves the neighborhood relations in an optimal way. This heuristic argument is in perfect agreement with the topographic product, which yields a value of $P = 0.000569$. Going beyond 2-D, we finally show the map onto a 6 $\times$ 6 $\times$ 6 cube with $N = 216$ nodes ((a) and (b) of Fig. 8). Again we have a dimension

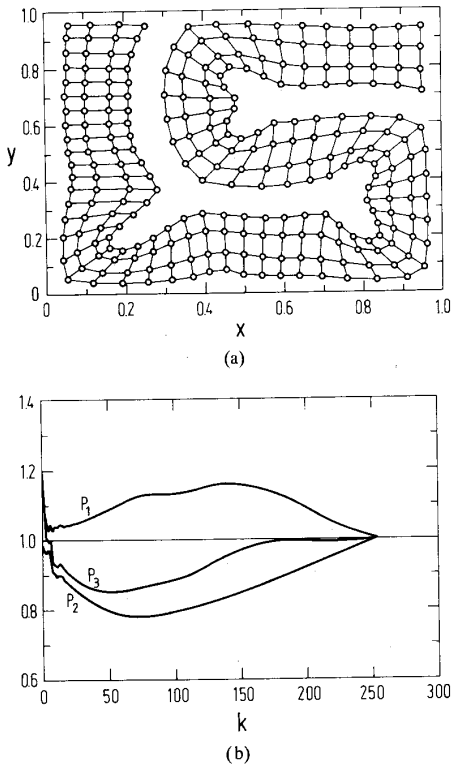


Fig. 6. (a) As in Fig. 4(a), but for $N = 64 \times 4$ nodes. (b) As in Fig. 4(b), but for the map of Fig. 6(a).

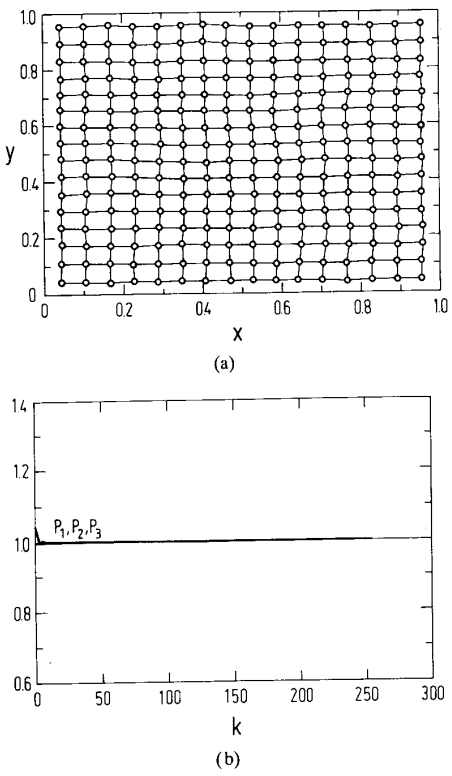


Fig. 7. (a) Map of a square input space onto a square output space with $N = 16 \times 16$ nodes. (b) As in Fig. 4(b), for the map of Fig. 7(a). The curves $P_1(k)$, $P_2(k)$, and $P_3(k)$ are very close to 1 in nearly the whole range of k, thus indicating a perfect preservation of neighborhood relations.

mismatch, this time with the input space folding itself into the output space. As a consequence, Fig. 8(a) shows a rather irregular inverse map. The $P(k)$ curves in Fig. 8(b) show again $P_1 \geq 1, P_2 \leq 1$, as we had, for example, in Fig. 4(b) for the 1-D output space. The combined product $P_3(k)$, however, now has values $P_3 \geq 1$, thereby indicating too large an output space dimension.

The results for this example are summarized in Table II, supplemented by a few intermediate topologies, for which we did not show extra figures. The results demonstrate that the topographic product picks the same output space topology as visual inspection of the inverse maps suggests. Even though for this example the intuitive approach seems adequate, we note that the topographic product method is an interpretation-free, quantitative approach that will prove its value for more complicated mapping problems whenever the "intuitive approach" fails. One possible reason is an input space of dimensionality $D^V > 3$, because then the map cannot be visualized, even if the stimuli lie in a lower dimensional subspace.

V. Recognition of Sequences

This section is meant as a demonstration that a perfect preservation of neighborhood relations can have a value in applications, which goes way beyond certain aesthetic considerations of theorists. To this purpose we need not introduce a new example, but we can use the results from the last section, which dealt with the mapping from a square input space onto output spaces of different dimension.

One important area of application for SFM's is speech recognition. If the recognition problem is just a feature vector classification, as in phoneme recognition, neighborhood relations between the output nodes are of no importance for the performance of the net. This is because only one output node is activated during a classification process, and the application does not involve any measure of distance between the output nodes. In applications of this kind, an LVQ fine-tuning of the map should turn out to be advantageous, as has been pointed out by Kohonen several times [12]. Even though the LVQ refinements do not involve neighborhood relations between nodes, one might consider whether or not a map with a better preservation of neighborhood relations provides a better starting point for LVQ fine-tuning. However, this is not the point we are interested in here. We want to consider an application where the postprocessing makes explicit use of the distances between nodes in the output space. This is the case for a word recognition scheme, where the sequence of feature vectors in the high-dimensional input space is replaced after mapping by a sequence of active nodes in the low-dimensional output space. Different versions of the same words are mapped onto nearby trajectories or, in some cases, the same trajectory

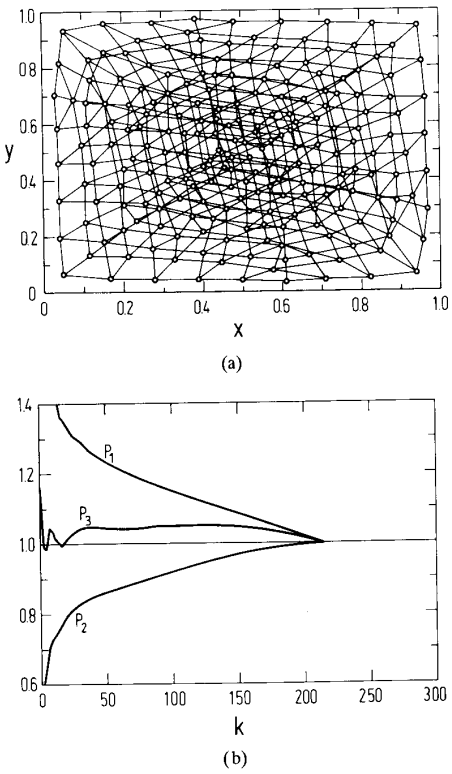


Fig. 8. (a) As in Fig. 4(a), but for the map from a square input space onto a cubic 6 × 6 × 6 node output space. (b) As in Fig. 4(b), but for the map of Fig. 8(a).

TABLE II
TOPOGRAPHIC PRODUCT P FOR THE MAP FROM A SQUARE INPUT SPACE ONTO A LINE, RECTANGLES, AND CUBES OF N NODES (VALUES FOR P AVERAGED OVER FOUR NETWORKS EACH)

N	P
256	-0.105 ± 0.010
64 × 4	-0.066 ± 0.002
32 × 8	-0.0301 ± 0.0001
16 × 16	0.0005 ± 0.00002
10 × 10 × 2	0.0076 ± 0.0003
6 × 6 × 6	0.0382 ± 0.00006

in the output space. For the classification of the output space trajectories, for example, a dynamic time warping (DTW) algorithm can be used. Compared with a DTW classification in the input space, a performance increase can be observed. However, for this increase to take place it is important to preserve as much as possible the neighborhood relations from the input space into the output space. In order to substantiate the latter claim, we mimic the speech recognition experiments in a simple way. Instead of the different words, we consider four reference trajectories $\{\boldsymbol{v}_i^\nu\}$ through a square input space (Fig. 9), which are mapped onto the output space trajectories $\{j_i^\nu\}$:

$$\Phi(\boldsymbol{v}_1^\nu, \boldsymbol{v}_2^\nu, \cdots, \boldsymbol{v}_N^\nu) \rightarrow (j_1^\nu, j_2^\nu, \cdots, j_N^\nu), \qquad \nu = 1, \cdots, 4. \tag{18}$$

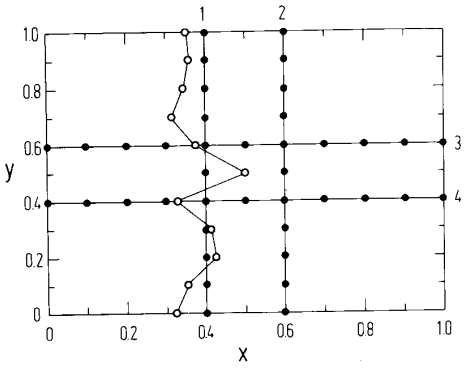


Fig. 9. Square input space with the four reference trajectories (solid dots) and one noisy trajectory (circles), which is a variation of reference trajectory 1. Each trajectory has $N = 11$ points. The noise is applied only to the x coordinates in the case of reference trajectories 1 and 2 and only to the y coordinates for trajectories 3 and 4. The noise for the additional test trajectory shown in the figure was taken from a flat distribution in the interval $[-0.1, 0.1]$.

Different version of the same "word" are mimicked by the addition of noise $\{\delta_i^\nu\}$ to the individual stimuli, and lead to the test trajectories $\{k_i^\nu\}$:

$$\Phi(\boldsymbol{v}_1^\nu + \delta_1^\nu, \boldsymbol{v}_2^\nu + \delta_2^\nu, \cdots, \boldsymbol{v}_N^\nu + \delta_N^\nu) \rightarrow (k_1^\nu, k_2^\nu, \cdots, k_N^\nu), \quad \nu = 1, \cdots, 4. \tag{19}$$

For the mapping Φ we use the maps into 1-D, 2-D, and 3-D output spaces, which were discussed in the last section. A noisy trajectory is identified with a reference trajectory μ by minimizing it mean square deviations, $d^{\nu'}$, from all reference trajectories in the output space:

$$d^{\nu'} = \sum_{i=1}^{N} d^A\left(j_i^{\nu'}, k_i^\nu\right) \tag{20}$$

$$d^\mu = \min_{\nu'} d^{\nu'}. \tag{21}$$

Classification is judged according to

$$\mu = \nu: \quad \text{correct recognition}$$
$$\mu \neq \nu: \quad \text{incorrect recognition.}$$

As can be seen in Fig. 10, the violation of neighborhood relations in the 1-D map induces the inclusion of output nodes into the output space trajectory, which are very far from their corresponding reference trajectory node, and thus leads to misclassification. Fig. 11, which shows the test trajectory in the D^A= 2-dimensional output space, has no such excursions.

In Fig. 12, the classification performance for 1-D, 2-D, and 3-D maps is shown as a function of the noise level. Both the 1-D map and the 3-D map show a substantial decrease of performance with increasing noise level, whereas the 2-D map performs very well even up to rather high noise levels. This performance difference between the nets is due to the global neighborhood violations, to which we made the topographic product particularly sensitive. Concluding this section, we note that preservation of neighborhood relations is more than

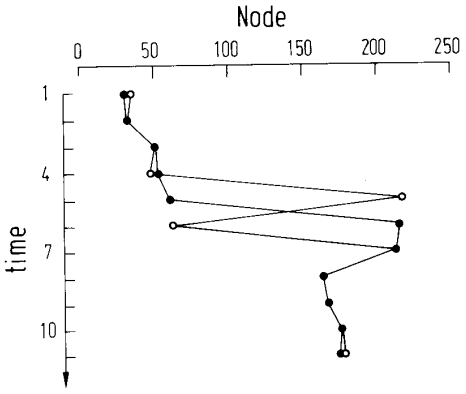


Fig. 10. Trajectories from Fig. 9 after mapping onto a 1-D output space with $N = 256$ nodes. For better visualization, the time coordinate runs in the negative y direction. The line connecting the solid dots gives reference trajectory 1; the line connecting the open circles shows the test trajectory from Fig. 9. At times steps 6 and 7, the slight deviations in the input space lead to large deviations in the output space, which might lead to a misclassification of the test trajectory.

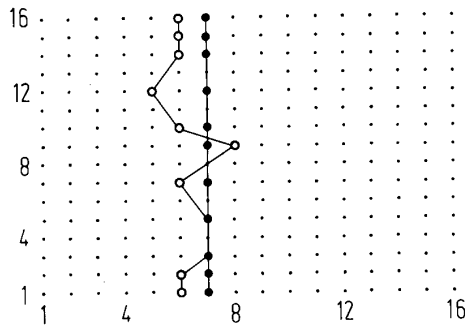


Fig. 11. As in Fig. 10, but for the two-dimensional output space. The two coordinates shown now coincide with the two coordinates of the output space.

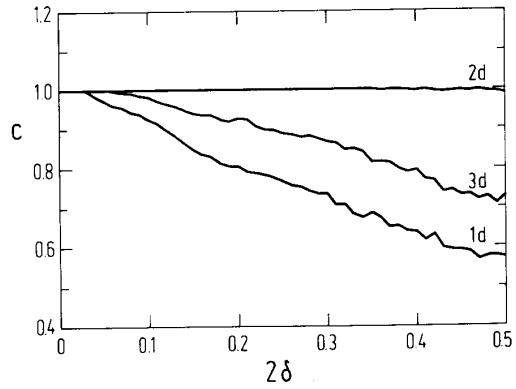


Fig. 12. Recognition rate, c, for the four test trajectories as a function of noise level δ. The noise is applied as described in the figure caption of Fig. 9; the values are randomly chosen from the interval $[-\delta, \delta]$.

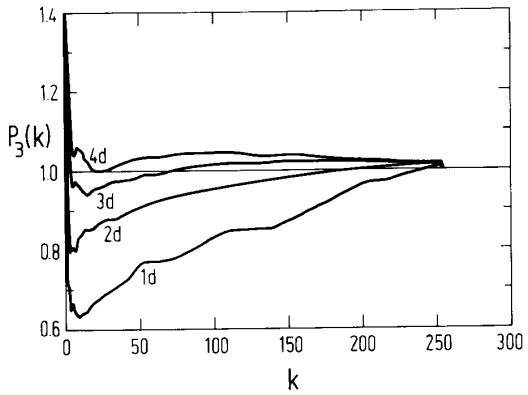


Fig. 13. $P_3(k)$ curves for maps of speech data onto an output space with $N = 256$ nodes (1-D), $N = 16 \times 16$ nodes (2-D), $N = 7 \times 6 \times 6$ nodes (3-D) and $N = 4 \times 4 \times 4 \times 4$ nodes (4-D). The 3-D curve yields the smallest absolute values and, consequently, corresponds to the best preservation of neighborhood relations.

a theoretical concept and can have a significant effect on the performance of SFM's in applications. Consequently the choice of an appropriate topology of the output space can also have a substantial effect on performance figures.

VI. Example II: Topographic Maps of Acoustical Data

Finally, we discuss the application of our method to a speech data set. The data set stems from the DPI data base, which has been accumulated at the III. Physikalisches Institut, University of Göttingen, Germany. From the data base of 40 German words, each spoken ten times by ten different speakers, we chose the ten German digits "Null" through "Neun," spoken by one male speaker (A.M.). For each digit we used the ten available versions. Recording and preprocessing of the data are described elsewhere [19]. Here we merely note that each word consists of 40–50 feature vectors, each feature vector giving the amplitudes of 19 (Bark-) frequency channels. Altogether the input data for our simulations consisted of 4500 vectors in a 19-dimensional input space. These data were mapped onto 256 output nodes, arranged as a line, a 16×16 square, a $7 \times 6 \times 6$ cube (252 output nodes in this case), and a $4 \times 4 \times 4 \times 4$ hypercube. The $P_3(k)$ values for the map are shown in Fig. 13, the averaged values, P, being $P = -0.166$ for 1-D, $P = -0.041$ for 2-D, $P = 0.019$ for 3-D, and $P = 0.034$ for 4-D. The smallest value of these is $P = 0.019$ for the 3-D case. However, this value contains contributions with $P_3(k) < 1$ as well as $P_3(k) > 1$. For this reason, the numerical value of P should be taken with a grain of salt. We note, however, that we have $P_3(k) < 1$ for the 2-D case in nearly the whole range of k and that $P_3(k) > 1$ for the 4-D case. So the somewhat indecisive 3-D case is framed by a too low-dimensional and a too high dimensional output space. This observation, together with the numerical evaluation, indicates that in a 3-D output space the data are represented in the most topology-conserving way.

Considering the classification example of the previous section, one can now suppose that a word recognizer based on a Kohonen map with a subsequent trajectory recognizer would perform better if the Kohonen map were arranged in 3-D. This is also the result of a recent speech recognition experiment

TABLE III
RECOGNITION RESULTS FOR SPEAKER-INDEPENDENT WORD RECOGNITION EXPERIMENT WITH THE TEN GERMAN DIGITS (DATA COMPUTED BY BRANDT *et al.* [15])

N	Recognition Performance
11×11	0.72
20×15	0.725
$6 \times 5 \times 4$	0.7725
$9 \times 7 \times 6$	0.795

carried out by Brandt *et al.* [15]. Using the same DPI data base, they preprocessed the speech data with Kohonen maps of different output space dimensionality, but about the same number of nodes. The authors then classified the trajectories in output space, which resulted from following the course of isolated words in the input space. The classification was performed with a dynamic time-warping algorithm, which eliminated fluctuations of the speaker velocity. (Apart from the time-warping aspect, we built the demonstration scheme of the last section following their approach.) The authors found their classification performance to increase from 0.725 to 0.795 if they increased their network dimension from 2-D to 3-D (Table III). This is a coincidence of two independent investigations into the effects of varying output space dimensionality in SFM's, which underscores the importance of topography in such maps.

VII. SUMMARY AND DISCUSSION

In this contribution, we solved the problem of the quantitative characterization of neighborhood preservation in self-organizing feature maps. To this purpose we considered a topographic product which had originally been introduced in order to estimate the dimension of the embedding space for strange attractors in nonlinear dynamics. We showed that this topographic product can readily be applied to the analysis to topographic feature maps of the Kohonen type. For the analysis only the weight vectors of the output space nodes are required; no knowledge of the stimulus distribution in the input space is necessary.

We demonstrated, in the very intuitive example of the map from a square input space onto output spaces of various dimensionality, how the vanishing of the topographic product points to the output space that best preserves the neighborhood relations. In different tests (not included in this paper), the applicability of the topographic product also for nonflat stimulus distributions, inducing varying areal magnification factors, was demonstrated.

The virtue of the preservation of neighborhoods was demonstrated in a sequence classification test which mimicked a speech recognition strategy using SFM's and DTW. The map with the best matching dimensionality clearly scored the highest recognition result. A detailed analysis of this effect might lead to an objective function which would connect the preservation of neighborhoods with some performance measure. Such an objective function would represent a valuable contribution to the discussion of the merits of neighborhood preservation, and would in this way provide a theoretical background for the method presented in this paper.

In a final example we were able to show that for speech recognition purposes an output space dimensionality of $D^A = 3$ instead of the usual $D^A = 2$ is better suited to the data. This result is in accord with performance results for single-word recognition on the same data set.

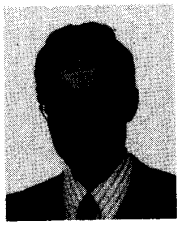
We expect the topographic product to prove a valuable tool in designing the topology of SFM's in applications. The technique needs neither statistics on the input data nor backprocessing such as DTW as a performance measure, since it rests only on the weights constituting the map. In this way, optimizing the network performance by optimizing the network topology, a strategy which proved very successful for MLP's will be easier to implement for SFM's as well.

ACKNOWLEDGMENT

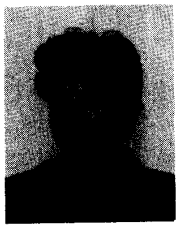
The kind hospitality of the Institute for Scientific Interchange, Torino, Italy shown to the authors during a stay there is gratefully acknowledged.

REFERENCES

[1] E. I. Knudsen, S. du Lac, and S. D. Esterly, "Computational maps in the brain," *Ann. Rev. Neurosci.*, vol. 10, pp. 41–65, 1987.
[2] T. Kohonen, *Self-Organization and Associative Memory*, 3rd ed. New York: Springer, 1989.
[3] H. Ritter, T. Martinetz, and K. Schulten, *Neuronale Netze*. Reading, MA: Addison Wesley, 1990.
[4] G. Blasdel and G. Salama, "Voltage sensitive dyes reveal a modular organization in monkey striate cortex," *Nature*, vol. 321, pp. 579–585, 1986.
[5] N. Suga and W. E. O'Neill, "Neural axis representing target range in the auditory cortex of the mustache bat," *Science*, vol. 206, pp. 351–353, 1979.
[6] J. H. Kaas, R. J. Nelson, M. Sur, C.-S. Lin, and M. M. Merzenich, "Multiple representations of the body within the primary somotosensory cortex of primates," *Science*, vol. 204, pp. 521–523, 1979.
[7] C. von der Malsburg, "Self-organization of orientation sensitive cells in the striate cortex," *Kybernetik*, vol. 14, pp. 85–100, 1973.
[8] D. J. Willshaw and C. von der Malsburg, "How patterned neural connections can be set up by self-organization," *Proc. Roy. Soc. London.*, vol. 194, pp. 431–445, 1976.
[9] R. Durbin and G. Mitchison, "A dimension reduction framework for understanding cortical maps," *Nature*, vol. 343, pp. 644–647, 1990.
[10] K. Obermayer, H. Ritter, and K. Schulten, "A principle for the formation of the spatial structure of cortical feature maps," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 87, pp. 8345–8349, 1990.
[11] T. Martinetz, H. J. Ritter, and K. J. Schulten, "Three-dimensional neural net for learning visuomotor coordination of a robot arm," *IEEE Trans. Neural Networks*, vol. 1, pp. 131–136, 1990.
[12] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, pp. 1464–1480, 1990.
[13] W. Liebert, K. Pawelzik, and H. G. Schuster, "Optimal embeddings of chaotic attractors from topological considerations," *Europhys. Lett.*, vol. 14, pp. 521–526, 1991.
[14] H. Ritter and K. Schulten, "Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability and dimension selection," *Biol. Cybern.*, vol. 60, pp. 59–71, 1988.
[15] W. D. Brandt, H. Behme, and H. W. Strube, "Bildung von Merkmalen zur Spracherkennung mittels phonotopischer Karten," in *Fortschritte der Akustik-DAGA 91*. Bad Honnef, Germany: DPG GmbH, pp. 1057–1060, 1991.
[16] H. Ritter, T. Martinetz, and K. Schulten, *Neuronale Netze*. Reading, MA: Addison Wesley, 1990, pp. 203–205.
[17] T. Martinetz and K. Schulten, "A neural gas network learns topologies," in *Proc. ICANN 91* (Helsinki), 1991.
[18] J. A. Kangas, T. K. Kohonen, and J. T. Laaksonen, "Variants of self-organizing maps," *IEEE Trans. Neural Networks*, vol. 1, pp. 93–99, 1990.
[19] T. Gramss and H. W. Strube, "Recognition of isolated words based on psychoacoutics and neurobiology," *Speech Comm.*, vol. 9, pp. 35–40, 1990.



Hans-Ulrich Bauer was born in Germany. He studied physics and received a master's degree from the University of California at San Diego, a Diplom degree from the Technical University of Munich, and a Ph.D. from the University of Frankfurt in 1990. His interest in neural systems began several years ago, and he pursued it in the stability analysis in recurrent networks and their application to speech processing during his graduate work. Currently he is at the University of Frankfurt, where he became a member of the Sonderforschungsbereich "Nichtlineare Dynamik." There he applies ideas from nonlinear dynamics to the analysis of neural systems. One of his particular areas of interest involves self-organizing maps. In a different line of research, he investigates oscillatory neural response, e.g., in cat visual cortex. He has published several articles.



Klaus R. Pawelzik was born in 1959 in Leverkusen, Germany. He received the Diplom degree in physics in 1987 and the Ph.D. in theoretical physics 1990 from the University of Frankfurt, Germany.

He is a fellow of the Volkswagen Foundation at the Institute of Theoretical Physics in Frankfurt. Since 1985 he has worked in the field of nonlinear dynamics, in particular, time series analysis and system identification. In 1991 he became member of the "Sonderforschungsbereich Nichtlineare Dynamik 185" of the German Research Society (DFG). Since 1987 he has collaborated with the Max Planck Institute for Brain Science in Frankfurt in the area of analysis and modeling of complex neuronal systems with methods from nonlinear dynamics and statistical physics.