Self-Organized Formation of Various Invariant-Feature Filters in the Adaptive-Subspace SOM

Teuvo Kohonen, Samuel Kaski, and Harri Lappalainen

Helsinki University of Technology
Neural Networks Research Centre
Rakentajanaukio 2 C, FIN-02150 Espoo, Finland

Neural Computation, 9:6 (August 15,1997), pp. 1321-1344.

Self-Organized Formation of Various Invariant-Feature Filters in the Adaptive-Subspace SOM

Teuvo Kohonen, Samuel Kaski, and Harri Lappalainen

Helsinki University of Technology
Neural Networks Research Centre
Rakentajanaukio 2 C
SF-02150 Espoo, FINLAND

Abstract — The Adaptive-Subspace SOM (ASSOM) is a modular neural-network architecture, the modules of which learn to identify input patterns subject to some simple transformations. The learning process is unsupervised, competitive, and related to that of the traditional SOM (Self-Organizing Map). Each neural module becomes adaptively specific to some restricted class of transformations, and modules close to each other in the network become tuned to similar features in an orderly fashion. If different transformations exist in the input signals, different subsets of ASSOM units become tuned to these transformation classes.

1 Introduction

A long-standing problem in artificial perception has been to recognize patterns subject to simple transformations such as translation, rotation, or scaling. A frequently held view is that invariances can be achieved by filtering of the pattern features that then become mapped into respective invariance classes. Another, equally important and intriguing problem then ensues: How to determine a set of effective feature filter functions for arbitrary statistics of dynamic signals? These problems seem to have been solved in biological systems in some effective way. In this article we devise a corresponding technical solution.

In traditional image analysis and pattern recognition techniques, the primary observations are usually preprocessed in a separate stage that extracts a number of "invariant features". On the basis of these features, a statistical-decision-making machine is then able to identify or classify the target invariantly. In applications where no prior standardization of the samples is possible, for instance when the targets are moving or their illumination is varying, the wavelet transforms such as the Gabor transforms (Gabor 1946; Daubechies 1990) have become popular as extractors of approximately invariant *local* features. These transforms, although optimizable by standard techniques, have the limitation that their mathematical forms must be fixed *a priori*.

The learning scheme discussed in this paper differs from all the previous approaches in that the *forms* of the filter functions are learned directly from short segments or episodes of dynamical signals or pattern samples. If the samples of the episode can be derived from each other by some linear transformation such as translation, they span a special *manifold* in the input signal space, namely, a *linear subspace*.

The ASSOM (Adaptive-Subspace Self-Organizing Map) architecture discussed in this article was introduced by one of the authors in 1995 (Kohonen 1995a; Kohonen 1995b; Kohonen 1995c; Kohonen 1996); the present work is a continuing study of it. In this scheme, the various feature filters emerge in learning and become tuned to those manifolds of input patterns that are defined only transiently in short segments of signals, but occur sufficiently often. These filters thus do not learn the pattern prototypes as such, but the different low-dimensional manifolds spanned by the consecutive signal pattern vectors.

The present discussion is believed to constitute a viable alternative to the standard PCA (principal-component analysis) methods of feature extraction (Hotelling 1933) for which neural approaches have been presented in, e.g., (Oja 1983, 1992; Rubner and Tavan 1989; Cichocki and Unbehauen 1993). The latter extract average features that reflect the global stationary statistical properties of the pattern sequence. The ASSOM, on the contrary, creates a set of statistical representations referring to different temporal events, and by competitive (winner-take-all) selection and learning, each of the representations selectively concentrates on a different class of temporal episodes. Moreover, due to the *spatial interactions* between the processing units of the ASSOM during learning, characteristic of all the SOM (Self-Organizing Map) methods (Kohonen 1982; Kohonen 1995c) in general, the various feature filters emerging in this process will be organized *spatially*, processing units being ordered along some characteristic feature coordinates.

One should mention a couple of recent works (Földiák 1991; Wallis et al. 1993) in which neural units learn sequences of simple patterns. In these works a fixed input layer of a two-layer architecture projects to a competitive second layer, where the winning unit learns a trace of successive inputs. The results obtained with the model do not, however, explain the emergence of the input-layer filters, which are essential for learning arbitrary *manifolds*. Effective and organized non-trivial invariant-feature filters can only ensue from the cooperation of several kinds of neural components, such as those constituting the ASSOM network architecture discussed in this work.

2 Auxiliary Concepts

First we set out to discuss the following subproblems in this section: (i) Invariant recognition of a pattern as invariant recognition of all its feature components. (ii) Definition of invariance classes for feature components as linear subspaces. (iii) Adaptive learning of the invariance classes from short sequences of input patterns called the episodes.

2.1 Decomposition of a Pattern into Features that are Mapped into Invariance Classes

Vectors that represent dynamic sensory environments are usually distributed along *manifolds*, which are continuous subsets or zones in the vector space. Consider tentatively that the input pattern represents some fundamental *feature component* of a more complex object, and is thus very simple, such as the Gabor functions are. Its variations, caused by a particular class of transformations of the basic pattern such as translation, can then be thought to span a low-dimensional manifold. For the simplest transformations the manifold can even be represented by a *linear subspace*, in which every vector is linearly dependent on the others. The class of linear transformations of the same fundamental component will be mapped into the same linear subspace that is then regarded as an *invariance class*.

If then a more complex pattern can be decomposed into a sum of such simple feature components, approximately at least, then all terms in the sum become mapped into their respective linear subspaces invariantly, whereby also the whole pattern will be represented invariantly with respect to this same linear transformation.

2.2 Sample Vectors

The input to a neural network model may be described by a real vector $\mathbf{x} \in \Re^n$. Here we consider *sample vectors* that consist of scalar samples collected from some time-domain intensity function I at different time lags v_i referred to some instant of time t:

$$\mathbf{x}(t) = [I(t - v_1), I(t - v_2), \ldots, I(t - v_n)]^{\mathrm{T}}. \tag{1}$$

Alternatively, the input vector $\mathbf{x}$ may be composed of samples of a two-dimensional input image, collected from points with intensity I around a time-dependent reference point $\mathbf{r} = \mathbf{r}(t) \in \Re^2$. These sampling points are displaced relative to $\mathbf{r}$ by the amounts $\mathbf{s}_i \in \Re^2$:

$$\mathbf{x}(t) = \mathbf{x}[\mathbf{r}(t)] = [I(\mathbf{r}(t) - \mathbf{s}_1), I(\mathbf{r}(t) - \mathbf{s}_2), \ldots, I(\mathbf{r}(t) - \mathbf{s}_n)]^{\mathrm{T}}. \tag{2}$$

The v_i and $\mathbf{s}_i$, respectively, form a regular or irregular *sampling grid*. The time series and the images, as well as patterns eventually representable in higher-dimensional domains, will be discussed below in the framework of the general vector-space formalism.

2.3 Episodes, Signal Subspaces, and Orthogonal Projections

One of the basic problems is thus how the linear subspaces, the invariance classes, can be defined, and another how to test whether an unknown pattern belongs to a subspace or not. Consider, for instance, time-domain signals. If N sample vectors ($N < n$) are collected from successive reference points t_p, the set of the nearby sample vectors $\{\mathbf{x}(t_p)\}$ spans some linear signal subspace of the total input space. The set of the reference time instants $\mathcal{S} = \{t_p\}$ will be called an *episode*. The maximum dimensionality of this subspace is equal to the number of sampling instants in $\mathcal{S}$, but the dimensionality can also be much smaller, if the sample vectors are dependent, as is often the case. The connection between subspaces and invariances lies in realizing that the *signal subspace itself is an invariant feature of the input vector* $\mathbf{x}$.

A linear subspace $\mathcal{L}$ of dimensionality H is in general completely defined by the general linear combination of the linearly independent basis vectors $\mathbf{b}_1 \dots \mathbf{b}_H$. However, the choice of the $\mathbf{b}_h$ is not unique: there exist infinitely many equivalent combinations of the $\mathbf{b}_h$ for the same $\mathcal{L}$, in some of which the $\mathbf{b}_h$ can be orthonormal. If a set of basis vectors are known, a set of equivalent orthonormal basis vectors for $\mathcal{L}$ can be computed by the familiar *Gram-Schmidt process* (for a textbook account, cf., e.g., Kohonen 1984, 1995c).

The basic operation to determine whether an unknown vector $\mathbf{x}$ belongs to $\mathcal{L}$ is the *orthogonal projection* of $\mathbf{x}$ on $\mathcal{L}$, denoted $\hat{\mathbf{x}}$. If $\hat{\mathbf{x}} = \mathbf{x}$, then $\mathbf{x}$ belongs to $\mathcal{L}$. For an arbitrary $\mathbf{x}$ one can define its *distance* from $\mathcal{L}$ as $\|\tilde{\mathbf{x}}\| = \|\mathbf{x} - \hat{\mathbf{x}}\|$, where the norm is Euclidean.

The orthogonal projection $\hat{\mathbf{x}}$ can be computed as either of the following simple expressions, provided that the b_h are orthonormal:

$$\hat{\mathbf{x}} = \left(\sum_{h=1}^{H} \mathbf{b}_h {\mathbf{b}_h}^{\mathrm{T}} \right) \mathbf{x} = \sum_{h=1}^{H} ({\mathbf{b}_h}^{\mathrm{T}} \mathbf{x}) \mathbf{b}_h \; . \tag{3}$$

The latter expression is computationally lighter.

Even though $\mathbf{x}$ does not belong to any subspace exactly, it can be *identified* with the closest one. The factor in the first parenthesis of (3) is a *projection operator matrix* abbreviated P, and there holds $P^2 = P$, $P^{\mathrm{T}} = P$. The pattern space is divided into *pattern zones*, each of which contains one of the above subspaces. The optimal separating surface between the ith and jth zone is the set of those points $\mathbf{x}$, the projections of which on the respective subspaces $\mathcal{L}^{(i)}$ and $\mathcal{L}^{(j)}$ have equal norms. The separating surface is thus defined by the following quadratic equation in $\mathbf{x}$:

$$\mathbf{x}^{\mathrm{T}} {P^{(i)}}^2 \mathbf{x} - \mathbf{x}^{\mathrm{T}} {P^{(j)}}^2 \mathbf{x} = \mathbf{x}^{\mathrm{T}} (P^{(i)} - P^{(j)}) \mathbf{x} = 0 \; , \tag{4}$$

where $P^{(i)}$ and $P^{(j)}$ are the projection operators to $\mathcal{L}^{(i)}$ and $\mathcal{L}^{(j)}$, respectively.

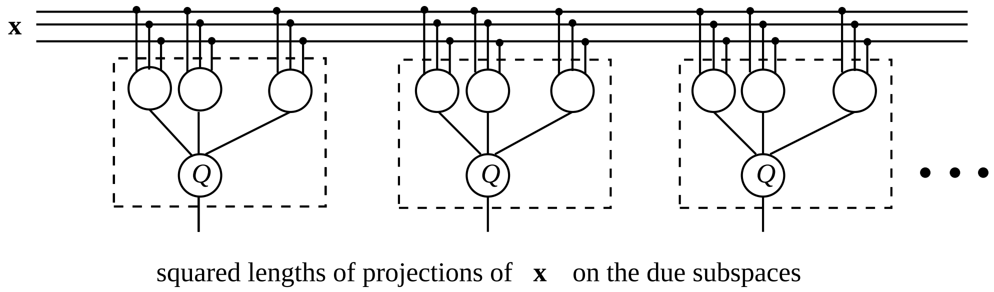

Figure 1: Neural network that consists of linear-subspace modules. The modules are separated by dashed lines. Q: quadratic neuron.

2.4 The Network Model

It may be helpful to imagine that the set of subspaces, the invariance classes, corresponds to a two-layered neural architecture, organized into *modules* that function as processing units (Fig. 1). Each of the modules represents a subspace denoted by $\mathcal{L}^{(i)}$. The neurons in the input layer are linear; if the weight vector of neuron h in the input layer of module i is denoted by $\mathbf{b}_h^{(i)}$ and the input by $\mathbf{x}$, respectively, the output of this neuron is $\mathbf{x}^{\mathrm{T}}\mathbf{b}_h^{(i)}$. The weight vectors $\mathbf{b}_h^{(i)}$ form the basis of the subspace $\mathcal{L}^{(i)}$ and they will be referred to as *basis vectors*. The neurons in the output layer have some quadratic transfer function Q. If in practical simulations the vectors $\mathbf{b}_h^{(i)}$ in each module are kept mutually orthonormal, Q is simply the sum of squares of the inputs of the second-layer neuron.

The sum of squares formed by the output neuron in each module equals the *squared norm of the projection* $\hat{\mathbf{x}}^{(i)}$ *of the input on the subspace* $\mathcal{L}^{(i)}$, or $\|\hat{\mathbf{x}}^{(i)}\|^2$.

The output of the module can then be interpreted as the degree of matching between the input and the subspace. *Thus, the output of each module is invariant* (neglecting eventual changes in the norm of the input) *to restricted linear transformations that occur within the subspace represented by the module.*

In the competitive learning process discussed below, the modules learn to represent different invariance classes.

Competitive learning in this network also involves neighborhood interactions between the modules similar to the neighborhood function in the standard SOM algorithm (Kohonen 1982; Kohonen 1995c). Due to these interactions, neighboring modules learn to represent similar features, or similarly transformed features, and the array of the modules becomes tuned to the features in an orderly fashion. In the scale of the whole network, due to these interactions, different areas in the network may become specialized to different input types, e.g., transforms.

3 Theory of the ASSOM

It has to be emphasized that in this work we not only want to create sets of feature filters: we also want these filters to become self-organized into a *spatial order* that reflects the similarity of the feature values they detect. Not only is then such a "map" easier to inspect, but the artificially produced "maps" can then be better compared with those existing in the neural realms.

3.1 Learning

The ASSOM units learn to become invariant, approximately at least, to the transformations that exist in their environment, by modifying their own subspaces to improve matching with the input signal subspaces. The latter are spanned by the successive samples in the episodes $\mathcal{S}$, as discussed in Sec. 2.3.

In order to approximate the input episodes by subspaces of low dimensionality, in the experiments described in this work, we fixed the ASSOM modules to have two filters, the minimum number required to extract invariant features; cf. Gabor filters. It is also quite possible to use more filters, however. When these modules are made to compete on the input signal subspaces, the module that represents a given signal subspace best wins the competition, and it, together with its neighbors in the array, is adapted to represent the input subspace even better. In this way the different modules become gradually specialized to represent different types of input subspaces.

As the modules *in the neighborhood of the winner* are also adapted to represent the input better, the neighboring modules gradually learn to represent similar inputs; this effect can be regarded as a kind of smoothing of the neighboring representations. Such "neighborhood learning" was also the basis of the Self-Organizing Map algorithm, where the weight vectors of processing units become spatially ordered in the map array.

The ASSOM learning process can be summarized in general terms as follows:

1. Locate the module ("winner") c, the subspace $\mathcal{L}^{(c)}$ of which is closest to the signal subspace of the input episode $\mathcal{S}$.
2. Adjust the subspace $\mathcal{L}^{(c)}$ of the winning module and modules in its neighborhood in the ASSOM array closer to the signal subspace.

Direct comparison of the signal subspace with the subspaces of the "neural" modules could be rather difficult, for instance, because the dimensionality of $\mathcal{S}$ may be defined vaguely due to noise in the samples. A simpler and much more robust definition of match between the input subspace and the "neural" subspaces is comparison of each input sample of the episode separately with the "neural" subspaces, and computation of the "energy" (sum of squares) of the projections of the input samples of the episode on each of the subspaces of the ASSOM modules. The module with the largest energy, i.e., sum of the squared projections of the samples collected during an episode, wins the competition. Denote the projections of the samples collected during $\mathcal{S}$ by $\hat{\mathbf{x}}^{(i)}(t_p)$. The "*representative winner*" for the whole episode $\mathcal{S}$, again indexed by c, is then defined by the expression

$$c = \arg\max_i \left\{ \sum_{t_p \in \mathcal{S}} \|\hat{\mathbf{x}}^{(i)}(t_p)\|^2 \right\} ,$$

or equivalently,

$$c = \arg\min_i \left\{ \sum_{t_p \in \mathcal{S}} \|\tilde{\mathbf{x}}^{(i)}(t_p)\|^2 \right\} . \tag{5}$$

The final goal in the competitive learning of subspaces is *to minimize the average expected error* when a stochastically selected input subspace is represented by the subspace of the winning module. A somewhat related error principle was also used in the classical vector quantization (Gersho 1979; Gray 1984; Makhoul et al. 1985). The distance between the subspace spanned by the input samples and the subspace $\mathcal{L}^{(i)}$ will here be measured in terms of the "energy" of the *relative* projection errors of the input samples.

Consider tentatively that no neighborhood interactions between the modules are yet taken into account and that the input vectors are normalized. The average expected distance between the input subspace and the subspace of the winning module, computed over the space of all possible Cartesian products X of input samples collected during the episodes $\mathcal{S}$, can be measured in terms of the projection error $\tilde{\mathbf{x}}^{(c)}$ and is expressed by the objective function

$$E = \int \sum_{t_p \in \mathcal{S}} \|\tilde{\mathbf{x}}^{(c)}(t_p)\|^2 p(X) dX . \tag{6}$$

Here $p(X)$ is the probability density of the X, and dX is a shorthand notation for a volume differential in the integration space, the Cartesian product of all the samples of the episode.

The index of the winning module c depends on the whole episode, as well as on the subspaces $\mathcal{L}^{(i)}$ of all modules i. Thus the exact minimization of (6) by differentiation may be a very complicated task (Kohonen 1991; Kohonen 1995c; for recent alternative treatments, cf., e.g. Buhmann and Kühnel 1993; Luttrell 1994). Therefore we shall resort to the classical *Robbins-Monro stochastic approximation* (Robbins and Monro 1951) in minimizing (6), in the same way as in the derivation of the traditional SOM algorithms (Kohonen 1991; Kohonen 1995c). It has to be emphasized that the objective function need not be a Lyapunov or energy function; nonetheless the convergence of the stochastic approximation has been established generally, cf. (Albert and Gardner 1967; Kushner and Clark 1978; Wasan 1969).

The objective function (6) describes the goal of pure unordered competitive learning. In (Kohonen 1991; Kohonen 1995c), to enforce *ordering* to the old vector quantization problems, the integrand in the error function E was *smoothed locally* with respect to the processing modules, which leads to the SOM algorithms. The same modification can also be made here by multiplying the integrand with the *neighborhood kernel* $h_c^{(i)}$, which is a decreasing function of the distance between the modules c and i in the ASSOM array. The new objective function then reads

$$E_1 = \int \sum_i h_c^{(i)} \sum_{t_p \in \mathcal{S}} \|\tilde{\mathbf{x}}^{(i)}(t_p)\|^2 p(X) dX \; . \tag{7}$$

Although attempts to optimize cost functions of the type (7) have not succeeded in closed form, it has been possible to use the Robbins-Monro stochastic approximation to derive the SOM algorithm (Kohonen 1991). (In the latter work, comparison of the exact and approximate optima was also made.) In this approximation, the gradient of E_1 is evaluated on the basis of the last input $\mathbf{x}$ and the last values of $\mathbf{b}_h^{(i)}$ during the sampling sequence. Here the whole set $X = X(t)$ is regarded to constitute a "sample" in the stochastic approximation. Thus, instead of optimizing the original objective function E_1, in this approximation a step in the direction of the negative gradient of the *sample function*, $E_2(t)$, is taken:

$$E_2(t) = \sum_i h_c^{(i)} \sum_{t_p \in \mathcal{S}(t)} \|\tilde{\mathbf{x}}^{(i)}(t_p)\|^2 \; . \tag{8}$$

Let us recall that $\tilde{\mathbf{x}}^{(i)} = \mathbf{x} - \hat{\mathbf{x}}^{(i)}$, where $\hat{\mathbf{x}}^{(i)}$, as a function of the $\mathbf{b}_h^{(i)}$, is given by (3). If the basis vectors are kept orthonormal, the gradient of the sample function with respect to basis vector $\mathbf{b}_h^{(i)}$ is

$$\frac{\partial E_2}{\partial \mathbf{b}_h^{(i)}}(t) = -2 h_c^{(i)} \sum_{t_p \in \mathcal{S}(t)} \left[\mathbf{x}(t_p)\mathbf{x}(t_p)^{\mathrm{T}} \right] \mathbf{b}_h^{(i)} \; . \tag{9}$$

Comment 1. As index c would be changed abruptly when the signal subspace passes the border between its two closest subspaces $\mathcal{L}^{(i)}$, it will be necessary to stipulate that the signal subspace does not belong to the infinitesimal neighborhood of such a border in (9). For continuous, stochastic signals, this possibility can be neglected. Cf. also a similar discussion in (Kohonen 1991; Kohonen 1995c).

Comment 2. As the basis vectors that span a subspace are not unique, it might seem that the gradient-descent method cannot be used, because the optimum is not unique. Notwithstanding it may be clear that *any* of these optima (for which E_2 is minimized) is an equivalent solution, so it will suffice if *one* optimal basis is found.

With a step length of $1/2\lambda(t)$ in the direction of the negative gradient we obtain the rule

$$\mathbf{b}_h^{(i)}(t+1) \;\; = \;\; \mathbf{b}_h^{(i)}(t) - \frac{1}{2}\lambda(t) \frac{\partial E_2}{\partial \mathbf{b}_h^{(i)}}(t)$$

$$= \left[I + \lambda(t) h_c^{(i)} \sum_{t_p \in \mathcal{S}(t)} \mathbf{x}(t_p)\mathbf{x}(t_p)^{\mathrm{T}} \right] \mathbf{b}_h^{(i)}(t) \; . \tag{10}$$

The parameter $\lambda(t)$ is called the *learning-rate factor*. In stochastic approximation, the $\lambda(t)$ must satisfy the following conditions (Robbins and Monro 1951; Kushner and Clark 1978): $\sum_{t=0}^{\infty} \lambda(t) = \infty$, $\sum_{t=0}^{\infty} \lambda^2(t) < \infty$.

In the learning law (10) the operator

$$R_1(\lambda(t)) = I + \lambda(t) h_c^{(i)} \sum_{t_p \in \mathcal{S}(t)} \mathbf{x}(t_p)\mathbf{x}(t_p)^{\mathrm{T}} \tag{11}$$

is applied to the basis vector $\mathbf{b}_h^{(i)}$. In the derivation of the learning law the input vectors were tentatively assumed to be normalized. If they are not, the normalization can be incorporated into the objective function (7) by considering *relative projection errors* $\|\tilde{\mathbf{x}}^{(i)}(t_p)\|^2 / \|\mathbf{x}(t_p)\|$. Then the rotation operator (11) becomes

$$R_2(\lambda(t)) = I + \lambda(t) h_c^{(i)} \sum_{t_p \in \mathcal{S}(t)} \frac{\mathbf{x}(t_p)\mathbf{x}(t_p)^{\mathrm{T}}}{\|\mathbf{x}(t_p)\|^2} \; . \tag{12}$$

Earlier (Kohonen 1995c; Kohonen 1996) we have shown that the following slightly different law, a product of unnormalized rotation operators, can be used for the competing "neural" units to learn the different input subspaces:

$$R_3(\lambda(t)) = \prod_{t_p \in \mathcal{S}(t)} \left[I + \lambda(t) h_c^{(i)} \frac{\mathbf{x}(t_p)\mathbf{x}(t_p)^{\mathrm{T}}}{\|\mathbf{x}(t_p)\|^2} \right] \; . \tag{13}$$

When $\lambda(t)$ is small, (12) and (13) can immediately be found equivalent. In fact, (12) can be seen as a batch version of (13), where the whole episode is used in one operation. One of the authors has earlier proved the so-called *Adaptive-Subspace Theorem* (Kohonen 1996), which states that if there is only one subspace $\mathcal{L}^{(i)}$, it will converge to $\mathcal{L}^* \subseteq \mathcal{L}^{(s)}$ almost surely, where $\mathcal{L}^{(s)}$ is the signal subspace from which the samples $\mathbf{x}(t_p)$ are picked at random.

We shall henceforth use the operator $R_3(\lambda(t))$, because it applies the input samples $\mathbf{x}(t_p)$ in succession and some extra operations explained in Sec. 3.2 shall be made after each operation. The operator $R_3(\lambda(t))$ in fact corresponds to Hebb-type learning: The modification of a synapse (a component of a basis vector) is proportional to the product of its input and the output of the neuron.

Since the expressions (12) and (13) do not preserve the norms of the basis vectors, for good performance the basis vectors should be orthonormalized, at least after some number (e.g., a few dozen) of learning steps.

3.2 Enhancement of the Organizing Power

It has to be emphasized that in order to achieve a sufficient stability in the self-organizing process and to obtain well-distributed filter sets, the advice given in this section seems to be necessary.

First of all, to ensure effective self-organization, any correction made on a neural unit must be a *monotonically increasing function of the error*. If this were not the case, the corrections made in neighboring units would strongly depend on their basis vectors before correction, which would lead to unstable self-organization.

The magnitude of the correction made on the subspace $\mathcal{L}^{(i)}$ should be a monotonically increasing function of the true error $\|\tilde{\mathbf{x}}^{(i)}\| = \|\mathbf{x} - \hat{\mathbf{x}}^{(i)}\|$, or a monotonically decreasing function of $\|\hat{\mathbf{x}}^{(i)}\|$. In the projective methods, the correction (cf. eqn 10) would be proportional to $\mathbf{x}(t_p)^{\mathrm{T}}\mathbf{b}_h^{(i)}$, which *decreases* with increasing angle between $\mathbf{x}(t_p)$ and $\mathbf{b}_h^{(i)}$. A simple method for guaranteeing that the correction is an increasing function of error (cf. Kohonen 1996) is to divide the learning-rate factor $\lambda(t)$ by the scalar value $\|\hat{\mathbf{x}}^{(i)}\|/\|\mathbf{x}\|$, which only changes the effective learning rate. The operator (13) then becomes

$$R^{(i)}(t) = \prod_{t_p \in \mathcal{S}(t)} \left[I + \lambda(t) h_c^{(i)} \frac{\mathbf{x}(t_p)\mathbf{x}(t_p)^{\mathrm{T}}}{\|\hat{\mathbf{x}}^{(i)}(t_p)\| \|\mathbf{x}(t_p)\|} \right] . \tag{14}$$

In the course of our simulations, reported in Sec. 4.2, it turned out that the organization of the ASSOM was not always perfect even when the modified rotation operator (14) was used. For instance, when the input consisted of speech signals, some modules were prone to become tuned to *two* separate frequency bands. An explanation of this phenomenon is that a high-dimensional filter in general can easily learn a rather complicated passband function, even one with many passbands, and it is mainly by the virtue of the competitive-learning process that single passbands at the filters are at all formed, because the process tries to distribute them in an orderly fashion over the frequency domain. When such two-band filters were formed in self-organization, the frequency range between these two bands remained unrepresented by the ASSOM array, i.e., a gap in the distribution was formed (cf. Sec. 4.2).

Without further speculation it may be mentioned that this instability can be eliminated by the following simple method, which has been empirically demonstrated to produce filters that are much better distributed in the frequency domain. If, after each learning step, the magnitude of small components of the basis vectors $\mathbf{b}_h^{(i)}$ is set to zero, then the basis vectors will be forced to approximate the stronger and more fundamental frequency components. If we denote the basis vectors by $\mathbf{b}_h^{(i)} = [b_{h1}^{(i)}, \ldots, b_{hn}^{(i)}]^{\mathrm{T}} \in \Re^n$, then a very simple "dissipation effect" of this kind can be described by the equation

$$b'^{(i)}_{hj} = \mathrm{sgn}(b_{hj}^{(i)}) \max(0, |b_{hj}^{(i)}| - \varepsilon) , \tag{15}$$

where ε $(0 < \varepsilon < 1)$ is a very small term. The amount of dissipation should be a fraction of the magnitude of the correction: $\varepsilon = \varepsilon_h^{(i)}(t) = \alpha|\mathbf{b}_h^{(i)}(t) - \mathbf{b}_h^{(i)}(t-1)|$, where α is a small scalar parameter. The dissipation is applied after each rotation and before normalization.

Also other regularizing principles have increased the stability of self-organization in our experiments. For instance, there exist excessive "parasite" degrees of freedom in the representation of basis vectors because of various symmetries. These "parasite" degrees of freedom often result in asymmetric filters: for instance, the filters may become centered at any point in the sampling grid. In order to avoid the ASSOM from being organized along irrelevant feature dimensions, there exists a simple method for favoring the middle of the sampling grid without significantly affecting the learning process in other respects, viz., weighting the lattice points by a Gaussian function centered in the middle. If the function is wide enough, it only affects the form of the filters quite near the edges. Since all input vectors are weighted similarly, this weighting does not alter the transformation group inherent in the input vectors but enhances symmetry.

To speed up the organization process, the Gaussian weighting function can also be made time-dependent. In the beginning of the organizing process the function could be selected as rather narrow, in order to fix the middle points first. After that the width of the Gaussian is increased and the modules learn to represent their inputs more and more accurately using more and more peripheral points of the sampling grid.

Still another method for enhancing the organizing power, a standard method for warranting good organization in ordinary self-organizing maps, is to make the neighborhood kernel dependent on time, $h_c^{(i)} = h_c^{(i)}(t)$. In the beginning of learning, the kernel should be very wide, effectively comprising half or even more of the map. When the width then gradually decreases to its final value, the map approximates its inputs more closely, while at the same time preserving the global ordering achieved in the beginning. In our simulations we used a simple box-shaped kernel. The modules up to a certain radius from the winner were thereby updated equally ($h_c^{(i)}(t) = 1$), whereas the other modules were not updated at all during this learning step ($h_c^{(i)}(t) = 0$).

3.3 Summary: The ASSOM Learning Algorithm

For each learning episode $\mathcal{S}(t)$ consisting of successive time instants $t_p \in \mathcal{S}(t)$ do the following:

- Find the winner, indexed by c:

$$c = \arg\max_i \left\{ \sum_{t_p \in \mathcal{S}(t)} ||\hat{\mathbf{x}}^{(i)}(t_p)||^2 \right\} .$$

- For each sample $\mathbf{x}(t_p)$, $t_p \in \mathcal{S}(t)$:

1. Rotate the basis vectors of the modules:

$$\mathbf{b}_h^{(i)}(t+1) = \left[I + \lambda(t) h_c^{(i)}(t) \frac{\mathbf{x}(t_p)\mathbf{x}(t_p)^{\mathrm{T}}}{\|\hat{\mathbf{x}}^{(i)}(t_p)\| \|\mathbf{x}(t_p)\|} \right] \mathbf{b}_h^{(i)}(t) \ .$$

2. Dissipate the components $b_{hj}^{(i)}$ of the basis vectors $\mathbf{b}_h^{(i)}$:

$$b'^{(i)}_{hj} = \mathrm{sgn}(b_{hj}^{(i)}) \max(0, |b_{hj}^{(i)}| - \varepsilon) \ ,$$

   where

$$\varepsilon = \varepsilon_h^{(i)}(t) = \alpha |\mathbf{b}_h^{(i)}(t) - \mathbf{b}_h^{(i)}(t-1)| \ .$$

3. Orthonormalize the basis vectors of each module.

The orthonormalization can also be made more seldom, say, after every hundred steps. Above, $\lambda(t)$ and α are suitable small scalar parameters.

4 Experiments

We shall show next what kinds of invariant-feature filters will emerge in the ASSOM learning process in different sensory environments, i.e., when the input to the ASSOM is subjected to various transformations. The emerging filters reflect statistical characteristics of short sequences of input patterns, but not waveforms as such. The filters will only learn such pattern components present in short input sequences that satisfy certain linear constraints, i.e., lie on a linear subspace. Patterns that do not satisfy these constraints will be averaged out in the long run. The resulting filters in fact represent *kernels* of transforms that produce the invariant outputs.

Before reporting the final results, we shall first define some details of simulations. Thereafter we demonstrate the enhancement of the organizing power due to the non-linear “dissipation” effect introduced in Sec. 3.2.

4.1 Details of the Simulations

In this section we summarize briefly what kinds of input patters were used, and how they were transformed.

Translated Speech Signal. In the first experiments the input signal consisted of digitized time-domain speech waveforms from the TIMIT (1988) speech data set, which contains phrases spoken by a large number of U.S. speakers, sampled at 12.8 kHz. Before picking up the input vectors, high frequencies in the signal were enhanced by taking differences of successive samples (high-pass filtering).

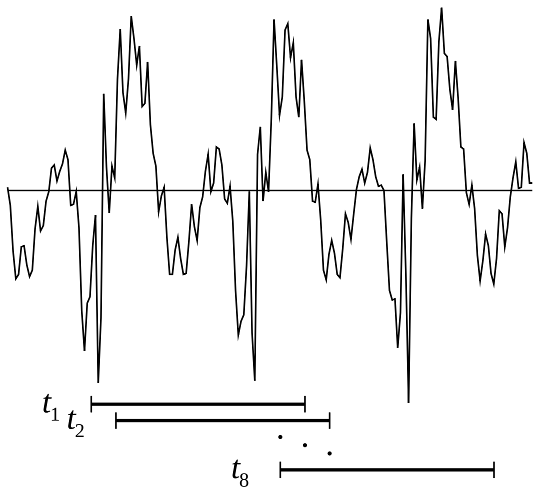

Figure 2: A sample speech waveform. The locations of the eight input vectors (with starting times $t_1, \ldots, t_8$) belonging to an episode are identified with the line segments under the speech waveform.

The input vectors consisted of 64 successive samples of the speech waveform. Each learning episode began at a randomly chosen time instant and consisted of eight vectors, each displaced by a random amount (eight samples on the average) from the previous one. Thus, the transformation group inherent in the data consisted of translations in time. A segment of the original signal and the eight input vectors forming an episode are exemplified in Fig. 2.

Two-Dimensional Colored Noise. Earlier (Kohonen 1995c) we have demonstrated that two-dimensional filters can be formed in response to photographic images. Here we wanted to specify the images statistically. Thus, in the rest of the experiments reported here, two-dimensional patterns consisting of colored noise were used. The patterns were formed by low-pass filtering of white noise with a second-order Butterworth filter that had the cut-off frequency at 60 per cent of the Nyquist frequency. The image was sampled with a circular, uniformly spaced sampling grid consisting of 316 pixels.

In the experiments the inputs were either *translated horizontally, rotated*, or *scaled.* The center of both the rotation and the scaling operations was fixed to be the center of the sampling grid. The signal subspaces spanned by the samples of the episode can be regarded as low-dimensional only if the transformations between the samples are modest. The amount of translation was varied from 0 to 10 pixels, using equal steps and a 10 per cent random displacement at each. The amount of rotation ranged from 0 to 1 radians, divided into equal steps and having 10 per cent random rotation at each step. The amount of scaling used in the episode increased with equal steps from 1.0 to 1.5, with a 10 per cent random variation applied at each step. The translated images were weighted by a vertical Gaussian kernel, to force all filters to become vertically centered at the same location (cf. Sec. 3.2). The other kinds of inputs, scaled and rotated, were not weighted.

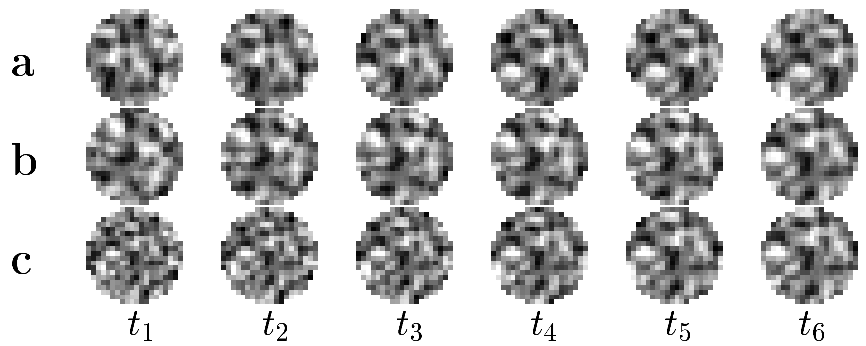

Figure 3: Samples of transformed colored noise patterns. **a** Translated, **b** rotated, and **c** scaled inputs. Each partial figure forms an episode.

The inputs were high-pass filtered in time by taking differences of successive input vectors to enhance the changing parts of the input. High-pass filtering is necessary for, e.g., scaled images where the center of the scaling remains approximately constant. A well-functioning invariance filter would then model predominantly this invariant area, instead of the changes in the stimuli produced by scaling. Before filtering, the number of samples in each episode was six, and the number decreased to five after taking the differences.

Exemplary episodes of translated, rotated, and scaled colored noise are shown in Fig 3.

Methods. The ASSOM lattice consisted, unless otherwise stated, of 24 units organized as a one-dimensional array. Each ASSOM unit contained two basis vectors $\mathbf{b}_1^{(i)}$ and $\mathbf{b}_2^{(i)}$, which were initialized with random values before learning. All inputs were normalized to unit length. The learning process was summarized in Sec. 3.3. The neighborhood kernel $h_c^{(i)}$ was box-shaped with $h_c^{(i)} = 1$ in the neighborhood of the winner, and $h_c^{(i)} = 0$ for the other units. The basis vectors were orthonormalized after each adaptation step. In case the computing time is critical, orthonormalization can be made more seldom.

During the learning process the parameters changed according to Fig. 4.

4.2 Demonstration of the Enhancement of the Organizing Power

Preliminary Experiment without Dissipation. In the first experiment with the ASSOM algorithm, the input consisted of translated speech signals. After 2000 learning episodes the ASSOM units represented different frequency components in an orderly fashion (Fig. 5). The two basis vectors in each unit, $\mathbf{b}_1^{(i)}$ and $\mathbf{b}_2^{(i)}$, became tuned to an approximately similar frequency band and had a 90-degree phase shift with respect to each other. Each output (squared projection of the input on the subspace of a unit) of the units became thus selective to a certain frequency band and was invariant to translations in time, as long as the frequency content of the input did not change.

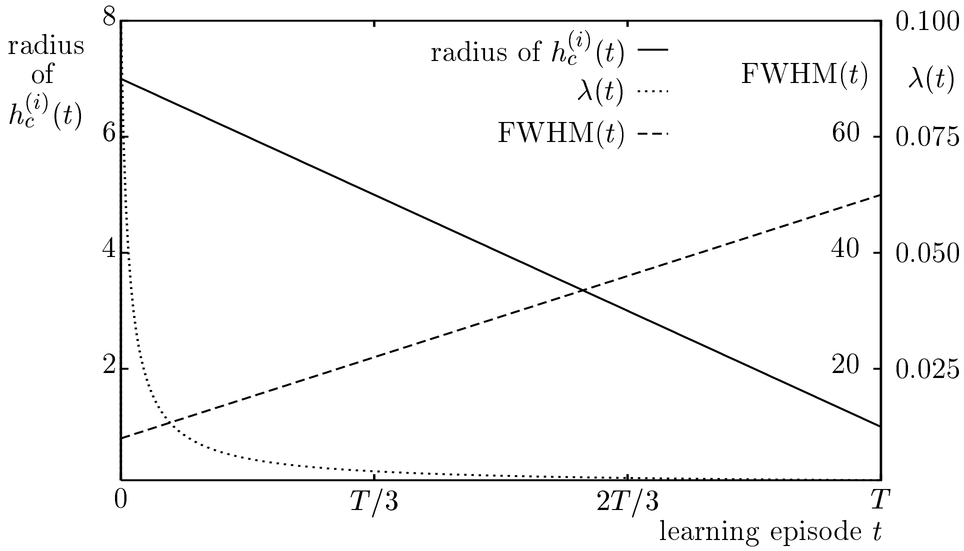

Figure 4: The learning parameters vs. number of learning episodes. The learning-rate factor $\lambda(t)$ was of the form $0.1 \cdot T/(T + 99t)$, where T is the total number of episodes during learning. In T steps the radius of the box-shaped neighborhood decreased linearly from 7 lattice spacings to one spacing. The Gaussian function used for weighting the sampled input pattern changed according to the curve marked by FWHM(t), full width at half maximum, expressed in terms of sampling points. The FWHM increased linearly from eight to 50 sampling points.

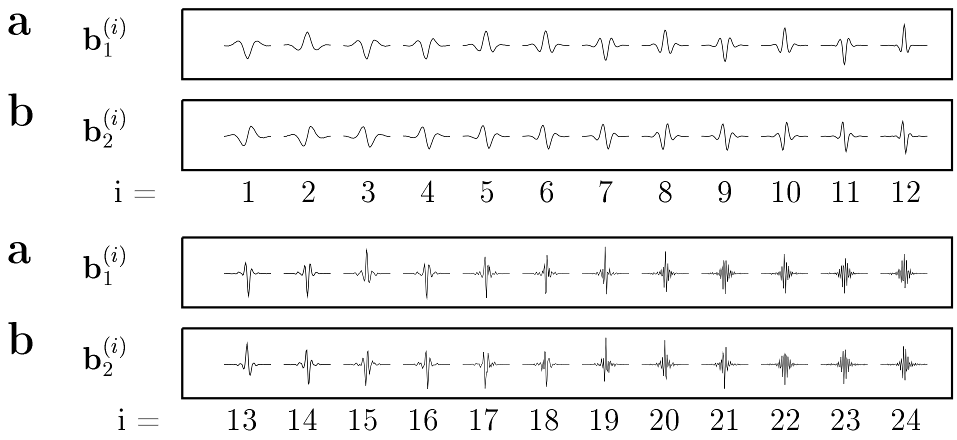

Figure 5: The tentative results with translated speech waveforms as inputs. Notice that the diagrams are in two parts. **a** The first basis vectors of the ASSOM units ($\mathbf{b}_1^{(i)}$). **b** The second basis vectors ($\mathbf{b}_2^{(i)}$).

If the learning period was extended, it turned out that the above organizing result was not stable. In continued learning a *gap* appeared in the representation. Some units had the tendency of representing several, usually two distinct frequency components instead of one compact frequency band, and some range of the input frequencies was left unrepresented. This occurred especially at the minima of the frequency spectrum. Fig. 6 shows the filters after a learning process consisting of 100,000 episodes. There is a clear defect in the organization: A wide gap at frequencies 2000 through 3500 Hz in the distribution of the dominant frequencies of the units (Fig. 6e). It is notable that the frequency response, defined as the output of a unit as a function of the input frequency, of unit number 18 (that is closest to the gap) consists of two separate peaks corresponding to the borders of the frequency gap (Fig. 6d). The ability of the ASSOM filters to represent several frequencies at the same time was discussed in Sec. 3.2.

Reducing Excessive Degrees of Freedom using Dissipation. After the dissipation term discussed in Sec. 3.2 was introduced, the frequency responses of all units obtained a single, relatively narrow passband (Fig. 7d). The value 0.02 was used for the dissipation constant α. The gap in the distribution of the dominant frequencies of the ASSOM units was thereby closed (Fig. 7e). The steepest slope in the distribution of filters corresponds to a local minimum in the frequency spectrum.

If Fig. 7a–b is inspected carefully, the effect of nonlinearity due to dissipation can be seen as small shoulders in the basis vectors at small values.

It may be interesting to note that an effective Gaussian-shaped "time window" was determined by the wavelet shapes (Fig. 7c), and its width seemed to be optimized for each filter separately.

4.3 Formation of Various Invariant-Feature Filters

The environment, i.e., the transforms that modify the inputs, determines what kinds of filters emerge in the learning process. In the previous section, where the input consisted of translated speech signals, the filters developed into wavelet-like sinusoidally modulated waves. In the following experiments we show that different types of invariant-feature filters are learned when the input is translated horizontally, rotated, or scaled. Since the system model was the same all the time, and the input samples in all of these experiments consisted of the same type of colored noise, the differences in the resulting filters are totally determined by the characteristics of the *transformation type*.

A total of $T = 30,000$ episodes consisting of 5 transformed sample patterns each were presented. The value of the dissipation constant α was 0.0001.

If the inputs are *translated* randomly in the image plane, in both the horizontal and the vertical direction, it has been shown earlier that basis vectors resembling two-dimensional Gabor-functions will emerge (Kohonen 1996). In the present work we show for comparison that if the translation is mainly restricted to one direction, say

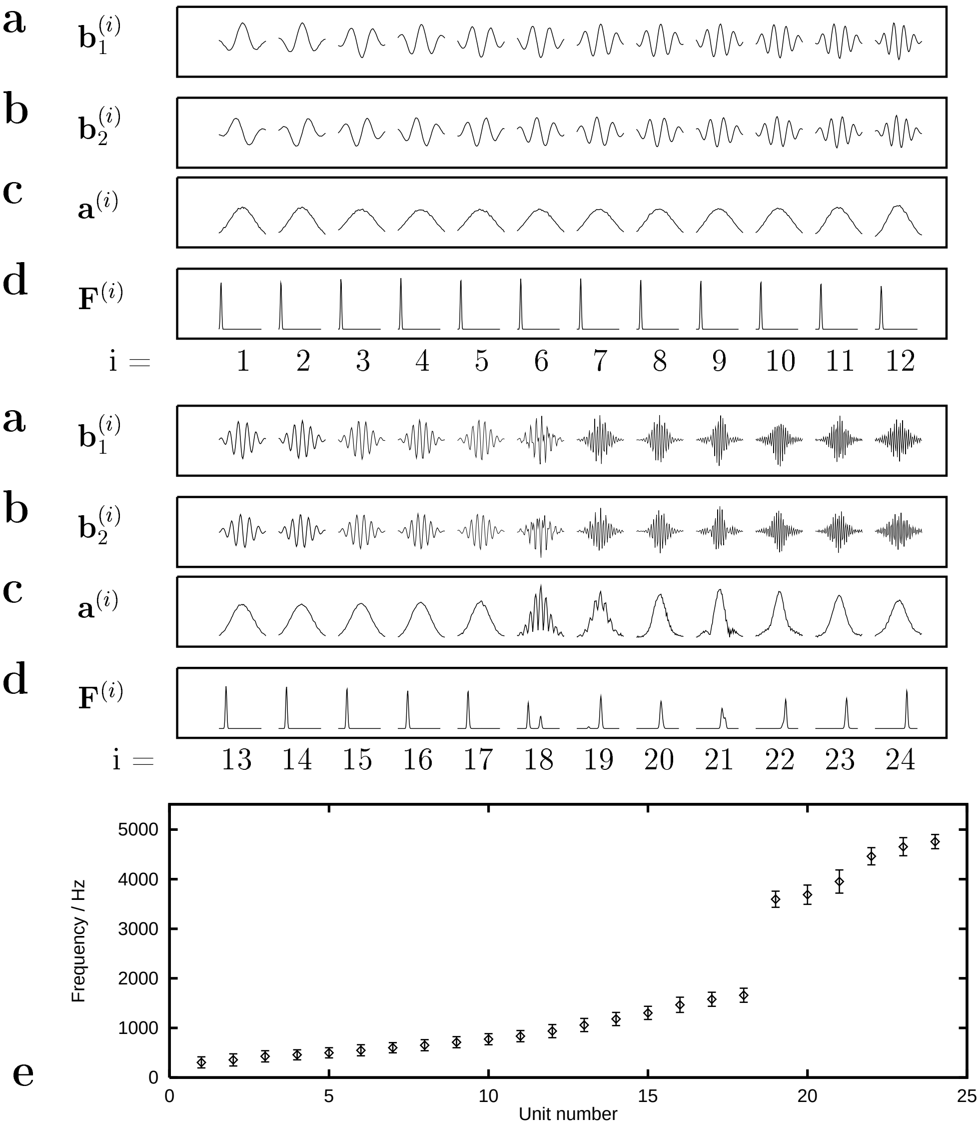

Figure 6: Results after a longer learning process: The tentative results turned out to be unstable in the long run. Notice that the diagrams a–d are in two parts. **a** The first basis vectors of the ASSOM units ($\mathbf{b}_1^{(i)}$). **b** The second basis vectors ($\mathbf{b}_2^{(i)}$). **c** $a_j^{(i)} = (b_{1j}^{(i)})^2 + (b_{2j}^{(i)})^2$, where $\mathbf{b}_1^{(i)} = [b_{1,1}^{(i)}, \ldots, b_{1,64}^{(i)}]^{\mathrm{T}}$ etc. **d** The response of the units to different frequencies, $F_j^{(i)} = (f_{1j}^{(i)})^2 + (f_{2j}^{(i)})^2$, where $\mathbf{f}_1^{(i)}$ and $\mathbf{f}_2^{(i)}$ are the Fourier transforms of the $\mathbf{b}_1^{(i)}$ and $\mathbf{b}_2^{(i)}$, respectively. **e** The dominant frequencies of the units. The bars indicate the FWHM of the frequency responses $\mathbf{F}^{(i)}$.

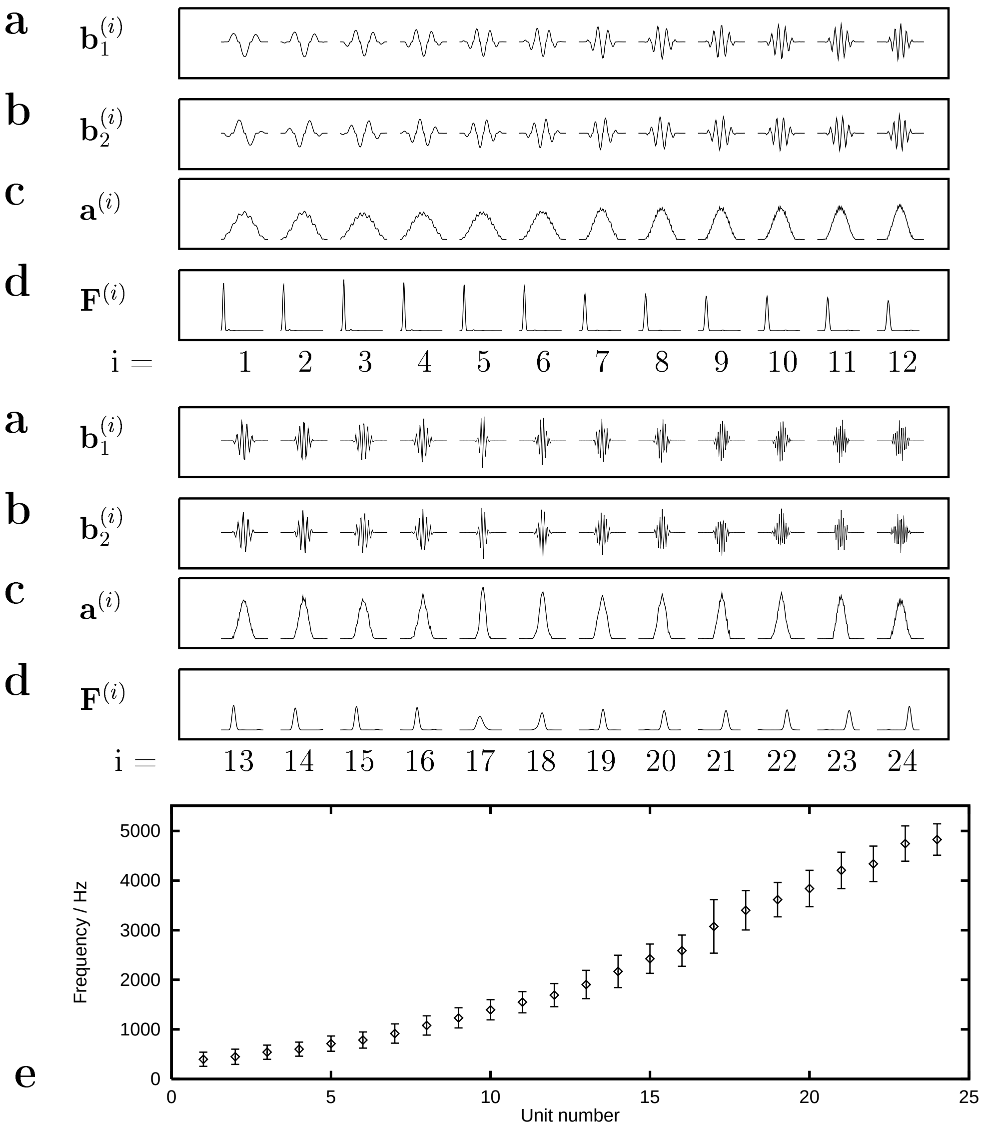

Figure 7: The final results with speech waveforms, when the nonlinear dissipation effect was included. Notice that the diagrams a–d are in two parts. **a** The first basis vectors of the ASSOM units ($\mathbf{b}_1^{(i)}$). **b** The second basis vectors ($\mathbf{b}_2^{(i)}$). **c** $a_j^{(i)} = (b_{1j}^{(i)})^2 + (b_{2j}^{(i)})^2$. **d** The response of the units to different frequencies, $F_j^{(i)} = (f_{1j}^{(i)})^2 + (f_{2j}^{(i)})^2$, where $\mathbf{f}_1^{(i)}$ and $\mathbf{f}_2^{(i)}$ are the Fourier transforms of the $\mathbf{b}_1^{(i)}$ and $\mathbf{b}_2^{(i)}$, respectively. **e** The dominant frequencies of the units. The bars indicate the FWHM of the frequency responses $\mathbf{F}^{(i)}$.

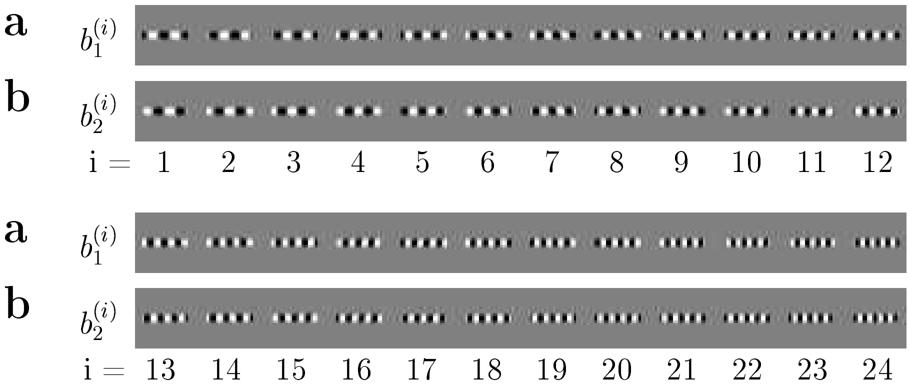

Figure 8: **a** Basis vectors $\mathbf{b}_1^{(i)}$, and **b** basis vectors $\mathbf{b}_2^{(i)}$ of an ASSOM that is invariant to horizontal translations. Notice that the diagrams are in two parts. During learning the FWHM of the vertical Gaussian used for weighting the inputs changed from 0.2 to 1.0.

horizontal, then the filters will become invariant to movements in that direction. Filters that have emerged when the input was translated horizontally and thereby weighted with a vertical Gaussian to eliminate the asymmetries (cf. Sec. 3.2) are shown in Fig. 8. The filters are sinusoidally modulated in the horizontal direction, and the two basis vectors in each unit have a 90-degree phase shift with each other.

In an environment where *differently rotated* input patterns are present, sinusoidally modulated basis vectors again emerge. This time the modulation occurs in the *azimuthal* direction (Fig. 9). The phases of the two basis vectors $\mathbf{b}_1^{(i)}$ and $\mathbf{b}_2^{(i)}$ have a 90-degree phase shift with respect to each other, whereby the responses of the units are invariant to rotation of the inputs.

The ASSOM subspaces can readily represent invariance classes of *linear* transformations, like rotation and translation. Also nonlinear transformations can be approximately represented linearly, the better the smaller the amount of transformation. Such linear approximations to a nonlinear operation emerge in the ASSOM when the inputs are *scaled* with respect to the center of the sampling grid. The basis vectors acquire a sinusoidal modulation in the *radial* direction outwards from the scaling center (Fig. 10). The phases of the two basis vectors in each unit again have a 90-degree phase shift with each other, whereby the responses of the units are invariant to the changing scale of the inputs.

4.4 Competition on Different Transformations

If the input patterns are subjected to several transformation types, each of them active during different episodes, the ASSOM units will also compete on these *different transformations*. Due to the neighborhood interactions during learning, neighboring

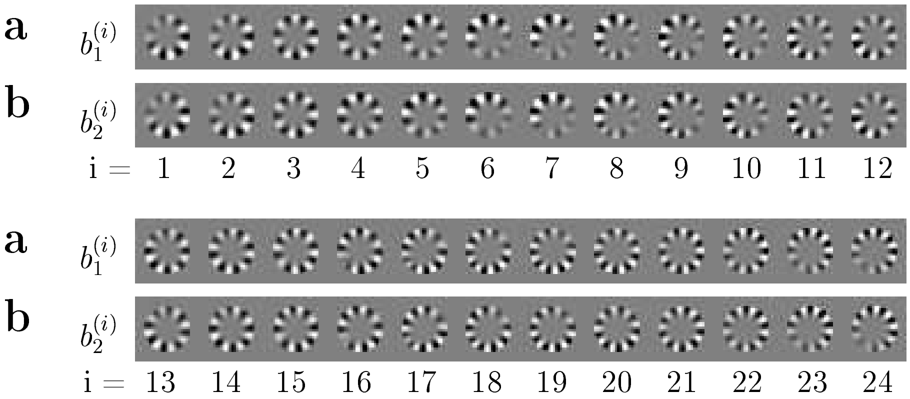

Figure 9: **a** Basis vectors $\mathbf{b}_1^{(i)}$, and **b** basis vectors $\mathbf{b}_2^{(i)}$ of the rotation-invariant ASSOM. Notice that the diagrams are in two parts.

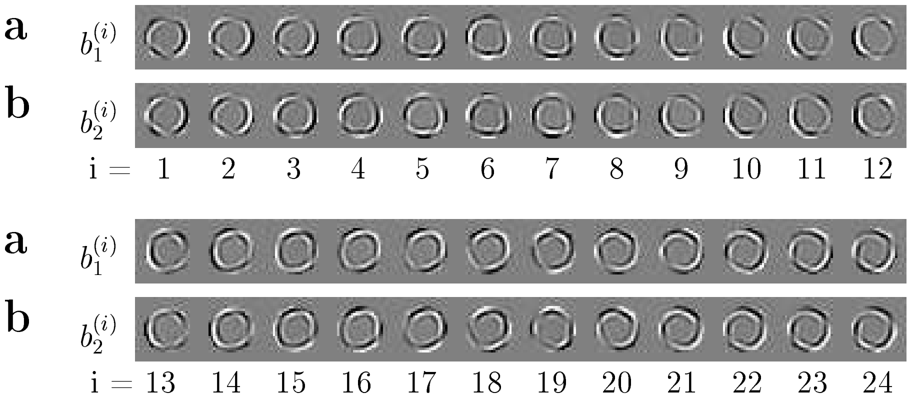

Figure 10: **a** Basis vectors $\mathbf{b}_1^{(i)}$, and **b** basis vectors $\mathbf{b}_2^{(i)}$ of the scale-invariant ASSOM. Notice that the diagrams are in two parts.

units will become invariant to similar transformations, but also *areas that are devoted to different transformations* will emerge.

To demonstrate the formation of areas specializing to different transformation types, we transformed the input patterns using three different transformation types, namely, horizontal translation, rotation, and scaling, respectively, during different episodes. The transformations were defined precisely like in the previous section, except that no Gaussian weighting was applied to the inputs. Vertical centering of the filters during horizontal translation was favored by varying the tilt of the pattern with respect to the sampling grid after each translation by a random amount (uniformly distributed between -0.5 and 0.5 radians). The ASSOM lattice consisted of 48 units, and the radius of the neighborhood kernel decreased from 14 to 1 units during learning. Otherwise the learning proceeded exactly as before.

After the learning process, three distinct areas can be discerned in the ASSOM (Fig. 11). The units depicted on the first row have mainly become invariant to scaling, and the units on the last row mainly to rotation. The units in the middle of the ASSOM, especially units 25-30 are best describable as being invariant to horizontal translations.

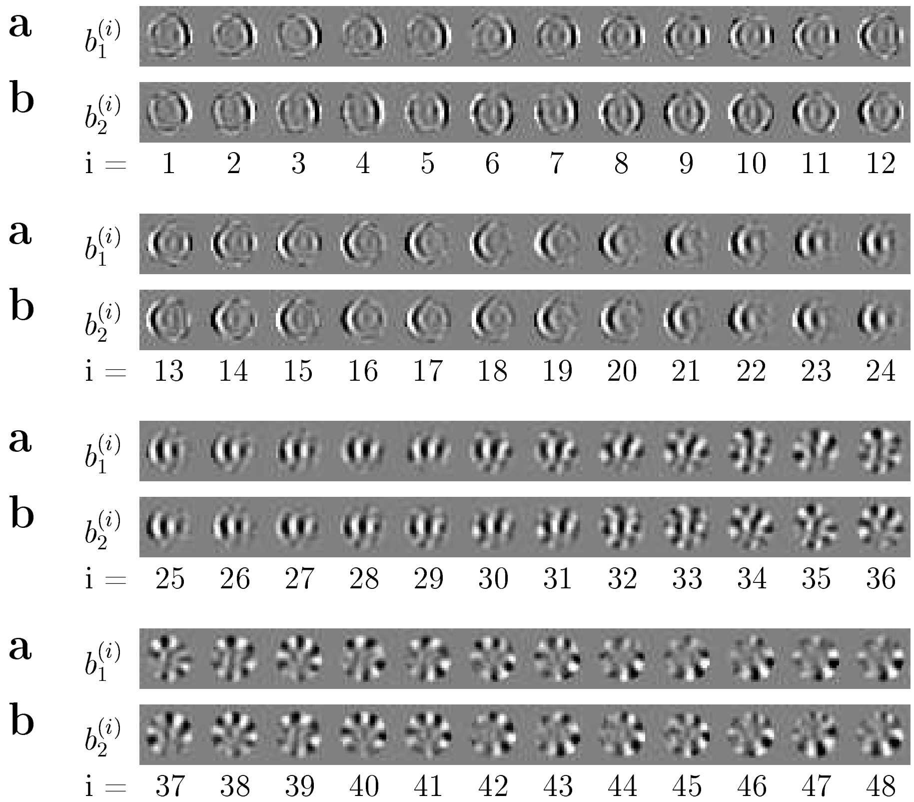

Figure 11: Different areas on an ASSOM have become invariant to different transformations, when the input consisted of horizontally translated, rotated, and scaled patterns. **a** The first basis vectors of the ASSOM units ($\mathbf{b}_1^{(i)}$). **b** The second basis vectors ($\mathbf{b}_2^{(i)}$). Notice that the diagrams are in four parts.

5 Discussion

In this article we demonstrated that invariant-feature filters and areas specializing to different transformations can emerge in the ASSOM as a result of Hebbian-type competitive learning. The learning rule was derived by minimizing an error function with the Robbins-Monro stochastic approximation. The objective function describes the average expected error of a random sequence of input samples compared with the corresponding best "neural" representation (subspace of the best-matching processing module).

With the simple nonlinear modifications of the learning rule described in Sec. 3.2, the invariant-feature filters were ordered robustly and reliably according to the similarity of the transformations. The capability of the network to learn invariant features was demonstrated for both natural speech signals and broadband random images.

Each processing module in the network can be made to become invariant to one transformation type, and decode a certain range of features invariantly of this transformation. Different competing modules specialize to represent different kinds of invariant features. The network can thus function as a *learning* preprocessing stage for invariant-feature extraction.

The number of basis vectors in the neural modules was two in our experiments. The resulting two-dimensional subspaces then corresponded to those of the basic wavelet filters. There exists, however, no restriction on the number of basis vectors in the algorithm, and subspaces of larger dimensions can readily be used to represent more complex features.

An intriguing possibility that the neurons in the ASSOM architecture might have biological counterparts ensues from the forms and operation of the filters in the input and output layers. The receptive fields of the *simple cells* in the mammalian primary visual cortex have been hypothesized to be describable by Gabor-type functions (Daugman 1980, 1985; Marcelja 1980; Jones and Palmer 1987), and the ASSOM process also produces Gabor-type filters in the first-layer neurons. The output-layer neurons respond invariantly to moving and transforming targets and in this sense resemble the *complex cells*.

Acknowledgement. The authors would like to thank Professor Erkki Oja for useful discussions.

References

Albert, A. E., and Gardner Jr., L. A. 1967. *Stochastic Approximation and Nonlinear Regression.* The M.I.T. Press, Cambridge, MA.

Buhmann, J., and Kühnel, H. 1993. Vector quantization with complexity costs. *IEEE Trans. Information Theory* **39**, 1133–1145.

Cichocki, A., and Unbehauen, R. 1993. *Neural networks for optimization and signal processing.* John Wiley, New York.

Daubechies, I. 1990. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. Information Theory* **36**, 961–1005.

Daugman, J. 1980. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Res.* **20**, 847–856.

Daugman, J. 1985. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A* **2**, 1160–1169.

Földiák, P. 1991. Learning invariance from transformation sequences. *Neural Comp.* **3**, 194–200.

Gabor, D. 1946. Theory of communication. *J. IEE* 93:429–457.

Gersho, A. 1979. Asymptotically optimal block quantization. *IEEE Trans. Information Theory* **25**, 373–380.

Gray, R. M. 1984. Vector quantization. *IEEE ASSP Magazine*, April, pp. 4–29.

Hotelling, H. 1933. Analysis of a complex of statistical variables into principal components. *J. Educat. Psych.* **24**, 498–520.

Jones, J., and Palmer, L. 1987. An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *J. Neurophysiol.* **58**, 1233–1258.

Kohonen, T. 1982. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **43**, 59–69.

Kohonen, T. 1984. *Self-Organization and Associative Memory.* 3rd ed. 1989. Springer-Verlag, Berlin.

Kohonen, T. 1991. Self-organizing maps: Optimization approaches. In *Artificial Neural Networks*, Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., eds., Vol. II, pp. 981–990. North-Holland, Amsterdam, Netherlands.

Kohonen, T. 1995a. The adaptive-subspace SOM (ASSOM) and its use for the implementation of invariant feature detection. In *Proc. ICANN'95, Int. Conf. on Artificial Neural Networks*, Fogelman-Soulié, F. and Gallinari, P., eds., Vol. 1, pp. 3–10. EC2 et Cie, Paris.

Kohonen, T. 1995b. Emergence of invariant-feature detectors in self-organization. In *Computational intelligence. A dynamic system perspective*, Palaniswami, M., Attikiouzel, Y., Marks II, R. J., Fogel, D., and Fukuda, T., eds., pp. 17–31. IEEE Press, New York.

Kohonen, T. 1995c. *Self-Organizing Maps.* Springer, Berlin.

Kohonen, T. 1996. Emergence of invariant-feature detectors in the adaptive-subspace SOM. To appear in *Biol. Cybern.*.

Kushner, H. J., and Clark, D. S. 1978. *Stochastic approximation methods for constrained and unconstrained systems.* Springer-Verlag, New York.

Luttrell, S. P. 1994. A bayesian analysis of self-organizing maps. *Neural Comp.* **6**, 767–794.

Makhoul, J., Roucos, S., and Gish, H. 1985. Vector quantization in speech coding. *Proc. IEEE* **73**, 1551–1588.

Marcelja, S. 1980. Mathematical description of the responses of simple cortical cells. *J. Opt. Soc. Am.* **70**, 1297–1300.

Oja, E. 1983. *Subspace methods of pattern recognition.* Research Studies Press, Letchworth, England.

Oja, E. 1992. Principal components, minor components, and linear neural networks. *Neural Networks* **5**, 927–935.

Robbins, H., and Monro, S. 1951. A stochastic approximation method. *Ann. Math. Statist.* **22**, 400–407.

Rubner, J., and Tavan, P. 1989. A self-organizing network for principal-component analysis. *Europhys. Lett.* **10**, 693–698.

TIMIT 1988. CD-ROM prototype version of the DARPA TIMIT acoustic-phonetic speech database.

Wallis, G., Rolls, E., and Földiák, P. 1993. Learning invariant responses to the natural transformations of objects. In *Proc. IJCNN'93, Int. Joint Conf. on Neural Networks, Nagoya*, pp. 1087–1090. IEEE Service Center, Piscataway, NJ.

Wasan, M. T. 1969. *Stochastic Approximation.* Cambridge University Press, Cambridge, Great Britain.