The Adaptive-Subspace SOM (ASSOM) and its Use for the Implementation of Invariant Feature Detection

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

Abstract. A new SOM architecture called the ASSOM (Adaptive-Subspace SOM) is shown to create sets of wavelet- and Gabor-type filters when randomly displaced or moving input patterns are used as training data. No analytical functional forms for these filters is thereby postulated. Different kinds of filters are formed by the ASSOM when pictures are zoomed during learning, or when they are rotated. The ASSOM can thus act as a universal, learning feature-extraction stage for pattern recognizers, being able to adapt to many sensory environments.

1 The Problem

In genuine pattern recognition, the input items have to be identified under various transformations of their representations. The present view is that recognition must be based on *features* that are more or less invariant with respect to different transformation groups inherent in natural observations.

Contemporary neural-network research mainly concentrates on decision-making systems and associative-memory network architectures, whereas the fundamental functions associated with the preprocessing of observations have often been ignored. This paper is a step toward theories that are expected to explain *the automatic emergence of invariant-feature detectors.*

2 Learning Linear Combinations

Around 1973 – 1976 this author pursued research on various linear neuron models that were supposed to become "habituated" to a set of input patterns (cf., e.g., [1], [2]). It turned out among other things that when such models learned a *finite* number of samples, they at the same time learned *the general linear combination* of the samples.

The "habituation" effect was demonstrated in a general form in the neural-network architecture called the Novelty Filter [2]. When an input vector $x = x(t) \in \Re^n$ was applied to such a system, the output from the system was another vector $y = y(t) \in \Re^n$ that depended on $x(t)$ and the earlier $x(t')$, $t' < t$ in a particular way. If, namely, only a finite set of *arbitrary* input patterns

$\{x(t_k)|k = 1, \ldots, K\}$ had occurred as inputs, then y was shown to be of the form

$$y = \gamma(t) \cdot \tilde{x} \ , \tag{1}$$

where $\gamma(t)$ is some scalar function of time, and $\tilde{x}$ is the residual vector in the decomposition $x = \hat{x} + \tilde{x}$, where $\hat{x}$ is the *orthogonal projection* of x on the subspace spanned by the $x(t_k)$, $k = 1, \ldots, K$. In other words, this system was shown to become habituated *to the general linear combination of the* $x(t_k)$.

The ideas presented in this paper are based on a similar effect. First, however, the linear-subspace formalism has to be discussed in more detail.

3 Linear-Subspace Neural Units

Consider N linearly independent basis vectors $b_1, \ldots, b_N \in \Re^n$ that are said to *span* the linear subspace $\mathcal{L} = \mathcal{L}(b_1, \ldots b_N)$. In other words, subspace $\mathcal{L}$ is identical with the set of all vectors $x \in \Re^n$ that satisfy the condition

$$x = \sum_{i=1}^{N} \alpha_i b_i \ \text{ for any scalars } \ \alpha_1, \ldots, \alpha_N \ . \tag{2}$$

For instance, in a three-dimensional space $\Re^3$, a plane that passes the origin can be defined by two basis vectors b_1 and b_2, and all points of this plane then constitute a two-dimensional subspace $\mathcal{L}$.

If $\mathcal{L}$ is now regarded as a *class*, then the *distance* of an *arbitrary* vector $x \in \Re^n$ from $\mathcal{L}$ can be defined in terms of *orthogonal projections.* It is always possible to decompose x as

$$x = \hat{x} + \tilde{x} \ , \tag{3}$$

where $\hat{x} \in \Re^n$ is a particular linear combination of the $b_1 \ldots b_N$, whereas the Euclidean norm of $\tilde{x} \in \Re^n$, denoted $||\tilde{x}||$ in the sequel, is minimum. Further $\hat{x}$ and $\tilde{x}$ are mutually orthogonal, i.e., their dot product is zero. The orthogonal projections $\hat{x}$ and $\tilde{x}$ can conveniently be computed by the traditional Gram-Schmidt method [3].

Whereas $\hat{x}$ and $\tilde{x}$ are vectorial variables, the distance of x from $\mathcal{L}$ is the scalar entity $||\tilde{x}||$. When we then define different classes $\mathcal{L}_i$ using different basis vectors $b_{i1} \ldots b_{iN}$, input x is considered to belong to that class $\mathcal{L}_i$ from which its distance $||\tilde{x}_i||$ is minimum.

In view of the Gram-Schmidt expansion it is now possible to think of a neural network consisting of two layers, as in Fig. 1. The input "neurons" are linear, and their weight vectors are identified with the b_{ih}, where i is the index of one of the units separated by the dashed lines and h is the index on an input neuron within that unit. The transfer function of each output neuron is quadratic (Q). In practical computations we tend to keep the basis vectors normalized and mutually as orthogonal as possible, whereby the output "neuron" only has to form the sum of squares of its inputs.

Formally we consider the b_{ih} as being the basis vectors of the linear subspace $\mathcal{L}_i$ that is associated with the "neural" unit i. In mathematical terms (cf. [3]) the output of unit i is always $||\hat{x}_i||$, the projection length of input x on subspace $\mathcal{L}_i$. Notice that Fig. 1 is only meant for illustration; it is possible that many other system architectures will implement the theory discussed in this paper.

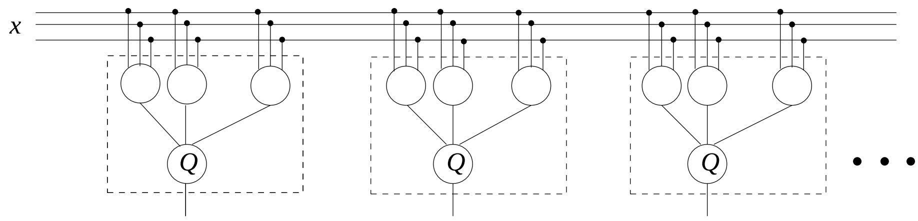


Figure 1: Neural network consisting of linear-subspace units.

4 The Adaptive-Subspace SOM (ASSOM)

Let us now revert to the problem of a neural network producing invariant responses to an input item, although the signal representation of the latter undergoes arbitrary transformations from a particular transformation group. The principle that is commonplace in biological sensory systems and is becoming popular in the theory of neural networks, too, is that observations of items are decomposed into and recognized on the basis of more or less invariant *elementary features.* For one-dimensional patterns such as acoustic waveforms one has introduced *wavelets* [4] for elementary features; two-dimensional patterns are often decomposed into *Gabor functions* [5].

If discrete sample vectors $x(t)$ are derived from some time function $f(t)$ as

$$x = x(t) = [f(t-t_1), f(t-t_2), \ldots, f(t-t_n)]^{\mathrm{T}} \in \Re^n \; , \tag{4}$$

then the central result in the following is that if we can decompose $f(t)$ into pieces of almost sinusoidal components, each of the neural units defined below will be tuned to one cluster of them, independent of the phases of the components, and thus develop into a particular invariant-feature detector.

For two-dimensional images with intensity $I(r)$ at the image coordinate $r \in \Re^2$, the input vector to the SOM is

$$x = x(r) = [I(r-r_1), I(r-r_2), \ldots, I(r-r_n)]^{\mathrm{T}} \in \Re^n, \tag{5}$$

where now the $r_1, \ldots, r_n \in \Re^2$ form a two-dimensional *sampling lattice.*

Emergence of sets of invariant-feature filters ensues from competitive learning, if we use the training samples and define the "winner" in a particular way. In this work we shall exclusively apply the Self-Organizing Map (SOM) architecture consisting of "neural units" of the above kind, whereby the units will be tuned to a one- or two-dimensional order along with the invariant features they are supposed to detect. The particular SOM described in this work is dubbed the *ASSOM (Adaptive-Subspace SOM).*

Assume that each neural unit i in a SOM array (like the neural network depicted in Fig. 1) represents a subspace $\mathcal{L}_i$ defined by k basis vectors $b_{ih} \in \Re^n$, $h = 1, 2, \ldots, k$. The "winner" unit c for a sample vector x can be defined in terms of the orthogonal components of x,

$$||\hat{x}_c|| = \max_i \{||\hat{x}_i||\} \;\text{ or }\; ||\tilde{x}_c|| = \min_i \{||\tilde{x}_i||\} \; . \tag{6}$$

Another concept we need is the *"representative winner" over a set of subsequent sample vectors $x(t)$ ("episode").* For every $x(t)$ we shall define the usual "winner" node $c = c(t)$, but what we actually want a particular node and its topological neighborhood in the network to do is *to learn the general linear combination of the temporally adjacent sample vectors $x(t)$.*

Consider the sample vectors $x(t_1), \ldots, x(t_k)$ taken from the same signal waveform close to each other. Let the respective projections on subspace $\mathcal{L}_i$ be $\hat{x}_i(t_1), \ldots, \hat{x}_i(t_k)$. The "representative winner" c_r may be defined as

$$c_r = \arg\max_i \left\{ \sum_{p=1}^{k} ||\hat{x}_i(t_p)||^2 \right\} . \tag{7}$$

There now seems to exist a paradox that whereas the "representative winner" was determined by one set of sequences, these samples are no longer available for learning later on. This paradox, however, is more apparent than real. In many practical cases, e.g., when analyzing ergodic waveforms, the correlation of having the same "representative winner" during subsequent episodes is high. Thus one could use one period of time for determining the "winner" and the immediately subsequent one for learning, respectively. *In the simulations reported below, however, learning was simply based on the same set of samples that was used for finding the "representative winner".*

Learning in the SOM is in general defined to occur in the "topological neighborhood" N_c of cells in the neural network, around the winner cell c. In the case of a "subspace SOM," learning means that the subspaces $\mathcal{L}_i$, i.e., the basis vectors of the $\mathcal{L}_i$ in the neighborhood set N_{c_r} around the "representative winner" must be *rotated* toward input x; but the degree of rotation must be a proper function of the distance of x from the subspaces. After some considerations [3], [6] we defined the rotations of the basis vectors as

$$b_{ih}(t) = \prod_{t_d \in E} \left[I + \alpha(t - t_d) \frac{x(t - t_d) x^{\mathrm{T}}(t - t_d)}{||\hat{x}_i(t - t_d)|| \; ||x(t - t_d)||} \right] \cdot b_{ih}(t - t_d) \ , \tag{8}$$

where $\hat{x}_i$ is the projection of x on $\mathcal{L}_i$, E is the set of sequence vectors used for this training step ("episode"), and the t_d are the relative displacements of the training vectors in time. The sample vectors should preferably be normalized; this can be done for each sample separately, or for some larger set of samples (i.e., dividing by their average norm).

5 Automatic Formation of Wavelet-Type Filters

In the experiment reported below we automatically generated wavelet-type filters for time-domain speech waveforms obtained from the TIMIT database, in which the signals have been sampled at 12.8 kHz. We formed the training vectors $x \in \Re^{64}$ from the samples by picking them up from randomly selected time windows of 64 adjacent points, equivalent to time intervals of 5 ms. We had a SOM with 24 units arranged in a one-dimensional array. The training process consisted of 30000 steps, whereby the radius of N_{c_r} changed linearly in time from twelve to one array spacings, and $\alpha(t)$ was of the form $A/(B+t)$, where $A = 18$ and $B = 6000$. At each step the episode consisted of eight windows displaced randomly over 64 adjacent sampling instants. A number of further details was introduced to the ASSOM process. First, in order to enchance signal energy at high frequencies, the signals in the TIMIT database were *high-pass filtered* by taking the differences of successive values. The data vectors x were then normalized. In this experiment we had *two* basis vectors, b_{i1} and b_{i2}, for each subspace. They were orthonormalized after each rotation operation. (This orthonormalization could have been made more seldom, say, once at the end of each tenth episode.) Third, if we would not *weight* the input samples in any way, the wavelet function might be formed anywhere within the sampling window; so, in order to stabilize the wavelets into the center of the window, the middle samples were provided with higher weights, and a Gaussian weighting function over the sampling "window" was used. The weighting was started using a narrow Gaussian weighting function having the full width at half maximum of eight sampling points and centered in the window. The narrow Gaussian stabilizes the high-frequency filters first, whereby also all filters are coarsely ordered. When the Gaussian is thereafter let to broaden up to 50 sampling points during the learning phase, the low-frequency filters acquire their final correct values. Fourth, after all the previous precautions we still noticed a peculiar instability: some filters, especially those close to the minima of the speech spectrum, were prone to be tuned to *two* frequency bands. In principle the filters could freely be tuned to several frequencies. However, due to the competitive-learning function present in this experiment, *different* filters in general tend to be tuned to *different* frequency components. We found out that the following simple *dissipative effect* was a sufficient remedy for producing reasonably smooth, well-distributed and asymptotically stable filters *having a single peak*: after each orthonormalization of the basis vectors, a small constant value (equivalent to $\alpha/50$) was subtracted from the *amplitude* of each component of the basis vectors (after they were orthonormalized at each step). This subtraction was saturable: the components did not change their signs.

Figure 2 shows the pairwise basis vectors formed in the self-organizing process at each "neural" unit.

6 Automatic Formation of Gabor-Type Filters

In the present experiment we used equidistant sampling with 225 sampling points over a square area of the closest pixels. At each learning cycle, this lattice was shifted randomly in four closeby positions, and the pattern vectors $x(t)$ thereby obtained constituted the two-dimensional "episodes" over which determination of the "representative winner", and updating of the corresponding N_{c_r}, were made. The learning cycle was repeated for other parts of the image, and for different images.

Discretized photographic images were used as training data. High frequencies were emphasized by subtracting local averages from the pixels. The sampling lattice was further weighted by a Gaussian function for better initial concentration of the filters. The Gaussian was flattened during learning. Fig. 3 illustrates the pairs of Gabor-type filters thereby produced.

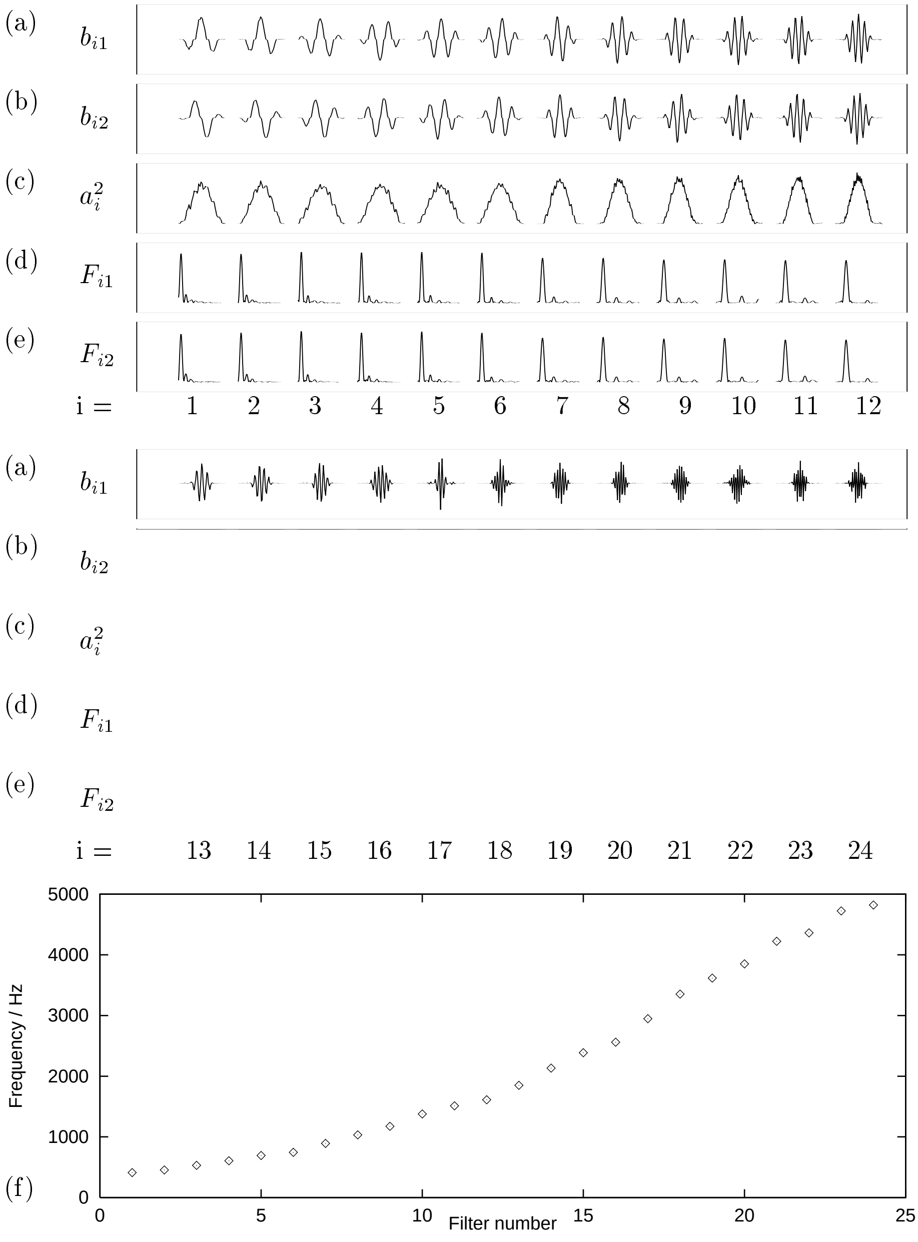
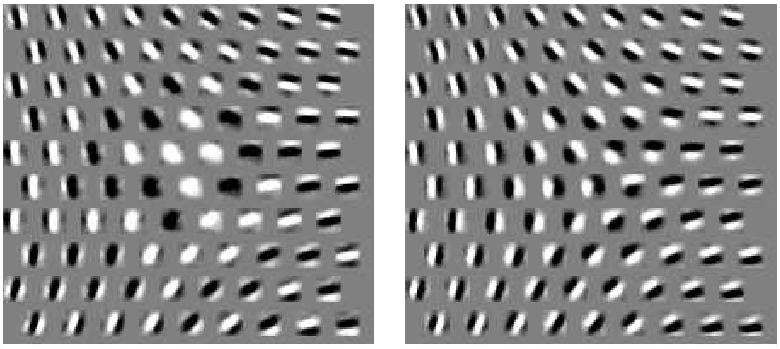


Figure 2: Notice that the diagrams (a) through (e) are in two parts. (a) Cosine-type wavelets (b_{i1}), (b) Sine-type wavelets (b_{i2}), (c) $a_i^2 = b_{i1}^2 + b_{i2}^2$, (d) Fourier transforms (F_{i1}) of the b_{i1}, (e) Fourier transforms (F_{i2}) of the b_{i2}, (f) Distribution of the average frequencies of the filters. The steeper part around 2500 Hz corresponds to a broad minimum of the speech spectrum!



(a) (b)

Figure 3: An ASSOM that has formed Gabor-type filters: (a) The b_{i1}, (b) The b_{i2}.

7 Automatic Formation of Approach- and Rotation-Invariant Detectors

It should be noted that an almost identical learning algorithm was used in Secs. 5 and 6; the filters developed differently because the input data in the training episodes were different. We shall now give two further examples of emergence of specific filters when the algorithm was still the same, but the input images were *zoomed* or *rotated* with respect to a fixed point, without otherwise translating the sampling lattice. In the example of Sec. 6 the optic-flow field was uniform over the sampling lattice. In the first experiment of this section, the optic flow was radiating from the zooming center, and its magnitude was directly proportional to distance from the said center. In the second experiment, the optic flow was circulating around the rotation center, and its magnitude was proportional to distance from the center, too. We had to high-pass filter the images in the temporal domain, by taking differences of successive images; in this way the almost static intensity around the center was attenuated automatically and the high-optic-flow parts were utilized better. If we did not do this, some randomly generated "parasitic" filters developed in the almost static surroundings of the center. It was also necessary to use sampling lattices with circular boundaries. No Gaussian weighting was used.

Fig. 4 shows the one-dimensional ASSOM for zoomed patterns. There were 316 points in the sampling lattice, and the input pattern $I(r)$ consisted of *random pixels*, low-pass filtered spatially at 0.2 (pixel)$^{-1}$. The episode was constructed of 5 zoomed patterns. The total range of the zoom was 1:2, and the 5 patterns of the episode were selected at random from this range. It is clearly visible that the feature detectors became invariant to radial motion, corresponding to approaching or withdrawing objects.

The ASSOM for rotated patterns is shown in Fig. 5. The sampling lattice and the primary patterns with spatial prefiltering were similar as in the zooming experiment, except that now the pattern was rotated over a range of 60 degrees

(a) b_{i1}

(b) b_{i2}

i = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Figure 4: Section of a one-dimensional ASSOM. (a) Cosine-type “radial wavelets” (b_{i1}), (b) Sine-type “radial wavelets” (b_{i2}).

(a) b_{i1}

(b) b_{i2}

i = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

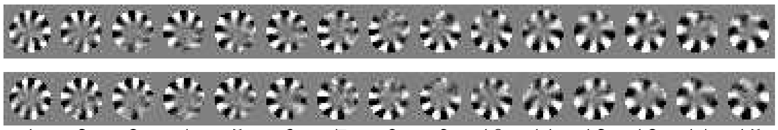


Figure 5: Section of a one-dimensional ASSOM. (a) Cosine-type “azimuthal wavelets” (b_{i1}), (b) Sine-type “azimuthal wavelets” (b_{i2}).

around the center of the sampling lattice. The 5 patterns in the episode were selected from this range at random. We can see that the feature detectors have become invariant to rotation.

8 Conclusion

It was demonstrated in this paper that wavelet- and Gabor-type filters are formed automatically by the ASSOM architecture if the primary patterns that define the successive input sample vectors in an episode are translated. If the patterns are zoomed or rotated, different kinds of feature detectors will emerge, although the algorithm is the same.

Sometimes it may be beneficial to introduce three or more basis vectors for each subspace, and the ASSOM array might be, say, three-dimensional. The ASSOM architecture may be used as a genuine, adaptive *feature-extraction layer* in a pattern recognition system [3]. It is also possible to exercise decision-directed training of the feature-extraction layer (see the so called FASSOM architecture discussed in [3]).

References:

[1] T. Kohonen, P. Lehtiö, J. Rovamo, Ann. Acad. Scient. Fennicae, Series A:V Medica, No. 167 (1974).
[2] T. Kohonen, E. Oja, Biol. Cybern. 21, 85-95 (1976).
[3] T. Kohonen, *Self-Organizing Maps.* Springer 1995.
[4] J. Daugman, IEEE Trans. Syst., Man, Cybern. 13, 882 (1983).
[5] D. Gabor, J. IEE 93, 429 (1946).
[6] T. Kohonen, Proc. WCNN'95 (in press).