Instabilities in Kohonen's self-organizing feature map

1994 J. Phys. A: Math. Gen. 27 1665

(http://iopscience.iop.org/0305-4470/27/5/029)

J. Phys. A: Math. Gen. 27 (1994) 1665–1681. Printed in the UK

Instabilities in Kohonen's self-organizing feature map

G A van Velzen†

Utrechts Biofysica Instituut, Princetonplein 5, PO Box 80 000, 3508 TA Utrecht, The Netherlands

Received 21 September 1992, in final form 1 October 1993

Abstract. The topology-preserving representation of a rectangular part of space onto a square network of formal neurons is studied using the Kohonen algorithm. Linear stability analysis shows that there is a critical ratio for the sides of the rectangle. For larger ratios the map becomes unstable. The value of the critical ratio depends on the actual shape of the adjustment function. The problems *cannot* be scaled away in the case of inhomogeneously sampled input space. The results of the analysis are compared with computer simulations.

1. Introduction

In this paper we study a model for topological map formation. Such models might describe observed activity patterns in the brain. On the other hand, the specific model we study has applications in robotics and data segmentation and classification tasks. Let us, however, start with the biologically inspired models that have been proposed. We give only an outline; for more details and arguments we refer the reader to the literature.

Studies of topology conserving maps usually involve a two-layer, feedforward network of neurons. The two layers interact through (modifiable) synaptic couplings. The first layer is the sensory layer, the second is a layer which, subject to input from the first layer, develops a structured activity pattern. Henceforth we will refer to the sensory layer neurons as 'sensors'. Both sensors and neurons are elements with continuous responses. The mechanism responsible for the formation of the activity pattern in the second layer has been proposed to be a short-range excitatory and long-range inhibitory in-layer interaction between neurons. This interaction is realized through in-layer synaptic couplings. Actually, this second layer is short-hand for two layers, one with excitatory, the other with inhibitory neurons [1–4, 6].

The in-layer synaptic couplings of the second layer are usually kept constant. Each neuron receives signals both from other neurons and from the sensors (sensory neurons). Its activity depends on the total input it receives, and the dynamics can be introduced in a straightforward manner, and has been investigated in [2]. Topological map formation is further pictured as follows. Relaxation subject to the sensory input leads to clustering of activity, to some degree [2]. Subsequently, a Hebbian rule (see e.g. [5]) is

† Present address: Grontmij Consulting Engineers, Department of Physical Planning, PO Box 203, 3730 AE De Bilt, The Netherlands.

0305-4470/94/051665+17$19.50

used to modify the synaptic strengths of the between-layer couplings. It is in these couplings that the information is stored. The time scale for neuron dynamics is much shorter than that for synaptic dynamics. After presenting sufficiently many randomly chosen examples, the synaptic strengths, through self-organization, will have become such that the activity in the network tends to the same topological structure as the input that is presented to it through the sensors. In case of non-matching dimensionality of input space and network, the most significant dimensions will be represented.

Although one usually thinks in terms of continuously varying firing rates of the neurons, it is also possible to model the neurons by binary elements [6].

Theoretical analysis of the system described above is rather complicated. In order to circumvent some of the problems, Kohonen has proposed the well known 'winner takes all' algorithm [7, 8]. From a physiological point of view, however, this approach has the drawback of involving a non-local mechanism in the form of the 'supervisor' determining which neuron is the 'winner'. Nevertheless, the model has nice properties, and is widely used in applications like robot learning, topographic map formation, classification of x-ray data, etc [9, 10]. It is thus worthwhile to study the convergence and stability of this algorithm. Points of interest have been the required behaviour of learning parameters, metastable states, form of the equilibrium configuration etc [11–13].

In this paper we study the robustness of the Kohonen algorithm against arbitrary sensor characteristics. As an elementary property of the sensors we take the ranges of their outputs. We find that the linear stability depends quite critically on the ratio of the ranges of different sensors. The consequence of this instability is that the map becomes of a different nature. In fact we find that the map may develop in such a way that the key property of the model, i.e. preservation of topology, *breaks down*.

The paper is organized as follows. In section 2, we define the model. In sections 3 and 4 we give an analysis of the fluctuations around the equilibrium map, a generalization of a study carried out by Ritter and Schulten [12]. In section 5 we investigate which modes become unstable for which ratio. In section 6 we present several simulations to compare with the theory. Section 7 contains comments and conclusions.

2. Model

In general, the Kohonen model [7, 8, 11, 12] maps a multi-dimensional input space onto a d-dimensional network. The situation in the cortex suggests the study of the $d=2$ case. For real-space representation in robotics one conveniently takes $d=3$. The input usually lies on a hyperplane parametrized with few parameters, e.g. many muscle spindle signals depending only on position and velocity of limbs.

We restrict ourselves here to the case in which the input is taken from a rectangular part of two-dimensional space. It is mapped through two sensors, just measuring Cartesian coordinates, to a two-dimensional $N_x \times N_y$ network. The network thus contains $N \equiv N_x N_y$ units. We take the connectivity of the network to be square. The units are labeled by r and distances in the network are measured in units of the lattice spacing. Every unit is coupled to the two spatial directions of the input space by adaptive weights $J_r = (J_{1r}, J_{2r})^T$, where the superscript ^{T} denotes transposition of a vector or matrix. The activity of unit r, upon presenting an input s to the network, depends monotonically on the inner product $s \cdot J_r$, e.g. the post-synaptic potential. The maximally responding unit is then said to 'represent' the input s.

Hebbian learning is realized here by replacing J_r by $J_r+\varepsilon s$ for the maximally responding unit *and* its neighbours. ε is the (small) learning parameter. In order to prevent weights from growing infinitely, one could include a synaptic decay process, or, following Kohonen, a normalization of the synaptic weights J_r. The latter costs one degree of freedom, so for a two-dimensional representation of a two-dimensional input, one thus needs *three* components of s and the J_r. These steps are repeated for every s, drawn subsequently from the input space.

This was the scheme Kohonen originally proposed [7].

One easily shows that the problem can be reformulated, for small ε, in terms of distances. In this case no normalization is needed. For a theoretical analysis, this is a more convenient formulation, and we will use it from here on.

In this approach, the point $s=(s_1, s_2)^T$ is represented by unit r, given by

$$\|J_r-s\|=\min_{r'}\|J_{r'}-s\| \tag{2.1}$$

where $\|\cdot\|$ denotes the usual square vector norm. Unit r represents all input points s that are in its *feature set*:

$$F_r\equiv\{s|\forall r'\neq r\colon \|J_r-s\|<\|J_{r'}-s\|\}. \tag{2.2}$$

Upon exposure to an example s from input space, learning proceeds for every r' according to

$$J_{r'}\rightarrow J_{r'}+\varepsilon h^0_{rr'}(s-J_{r'}) \tag{2.3}$$

if s is in the feature set of neuron r. $h^0_{rr'}$ is the network *neighbourhood function*, which is usually taken to depend symmetrically on the difference $r-r'$ and decays with increasing distance. The occurrence of these neighbourhood relations in the learning rule is the origin of topological map formation. (Note that $J_{r'}$ inside the brackets cannot be considered as synaptic decay, because this 'decay' is subject to the neighbourhood function $h^0_{rr'}$.) For example, a typical choice for the neighbourhood function, for which only nearest neighbours are 'dragged along' with the 'winner', would be

$$h^0_{rr'}=\delta_{r,r'}+\sum_{n=\pm e_x,\pm e_y}\delta_{r+n,r'} \tag{2.4}$$

where $\delta_{r,r'}$ is the Kronecker delta function. e_x and e_y are unit vectors (in the network) in the x- and y-direction.

If the s are drawn *randomly* from the input space, according to some probability density $P(s)$, the steps described above define a Markov process. Starting from random couplings J, and meeting certain conditions (which we will indicate later on), a topologically correct representation of input space will emerge. Recently, the occurrence of metastable states has been investigated in more detail [15].

3. Fokker–Planck equation

The set of couplings $J\equiv(J_{r_1}, J_{r_2}, \ldots, J_{r_N})$ determines the state of the system, and we will henceforth refer to J as the *state*. In every Markov step this state changes from J' to J according to (2.3), or formally:

$$J=T(J', s, \varepsilon). \tag{3.1}$$

The transition probability for going from state J' to J is given by

$$Q(J, J') = \sum_r \int_{F_r(J)} \delta(J - T(J', s, \varepsilon)) P(s)\, \mathrm{d}s. \tag{3.2}$$

In order to apply methods from statistical mechanics, one considers an ensemble of these systems, whose states J at iteration time t are distributed according to a distribution function $\tilde{S}(J, t)$. The evolution of the distribution function $\tilde{S}$ is described by the Chapman–Kolmogorov equation [14]

$$\tilde{S}(J, t+1) = \int \mathrm{d}J'\, Q(J, J') \tilde{S}(J', t). \tag{3.3}$$

This expression has been further analysed in [12] as follows. One assumes the existence of a stationary expectation value $\langle J \rangle$, which is the solution of

$$\lim_{\varepsilon \downarrow 0} \int \mathrm{d}s\, P(s) T(\langle J \rangle, s, \varepsilon) = \langle J \rangle. \tag{3.4}$$

As long as ε does not equal zero, the map will fluctuate around this stationary solution. The fluctuations are proportional to $\sqrt{\varepsilon}$ (see later, e.g. equation (3.17) and equations (4.18) and (4.19)).

Next, $\tilde{S}$ is expanded in deviations $j \equiv J - \langle J \rangle$ from this stationary state, keeping only derivatives up to the second order, and of these only the leading order in ε. This requires the learning parameter ε to be small enough such that the individual Markov steps are sufficiently small. The resulting equation is given by [12]

$$\frac{1}{\varepsilon} \partial_t S(j, t) = \sum_{rr'} \nabla_{j_r} \cdot B_{rr'} j_{r'} S(j, t) + \frac{\varepsilon}{2} \sum_{rr'} \nabla_{j_r} \cdot D_{rr'}(\langle J \rangle) \nabla_{j_{r'}} S(j, t). \tag{3.5}$$

This is known as the multivariate linear Fokker–Planck equation (see e.g. [14]). The centre of the distribution function has been shifted to the origin:

$$S(j, t) \equiv \tilde{S}(\langle J \rangle + j, t). \tag{3.6}$$

The constant 2×2 matrix $B_{rr'}$ is given by

$$B_{rr'} \equiv \left. \frac{\partial V_r(j)}{\partial J_{r'}} \right|_{J = \langle J \rangle} \tag{3.7}$$

where

$$V_r(J) \equiv \sum_{r'} h_{rr'} (J_{r'} - \langle s \rangle_{r'}) \bar{P}_{r'}(J) \tag{3.8}$$

with

$$h_{rr'} \equiv h^0_{rr'} / (1 - \varepsilon h^0_{rr'}) \tag{3.9}$$

$$\langle s \rangle_r \equiv \frac{1}{\bar{P}_r(J)} \int_{F_r(J)} s P(s)\, \mathrm{d}s \tag{3.10}$$

and

$$\bar{P}_r(J) \equiv \int_{F_r(J)} P(s)\, \mathrm{d}s. \tag{3.11}$$

The difference between $h_{rr'}$ and $h^0_{rr'}$ is only of order ε. One easily interprets $-\varepsilon V_r(J)$ as the amount by which, given a state J, the individual J_r is modified. The 2×2 matrix $D_{rr'}$ is given by

$$D_{rr'}(J) \equiv \sum_{r''} h_{rr''} h_{r'r''} \Bigg[(J_r - \langle s \rangle_{r''})(J_{r'} - \langle s \rangle_{r''}) \bar{P}_{r''}(J) \tag{3.12}$$

$$+ \int_{F_{r''}(J)} (ss^{\mathrm{T}} - \langle s \rangle_{r''} \langle s \rangle_{r''}^{\mathrm{T}}) P(s)\, \mathrm{d}s \Bigg]. \tag{3.13}$$

By B, D, etc, we will denote the corresponding $2N\times 2N$ matrices. The first-order term in (3.5) is the restoring force, the second-order term is the diffusion term. An initial $j(t=0)$ will decay according to

$$\langle j \rangle(t) = \mathrm{e}^{-\varepsilon t B} j(0). \tag{3.14}$$

The first moment of the distribution thus depends only on B. The second moment also depends on the diffusion term in the Fokker–Planck equation. The system is stable if the matrix B is positive definite. If it is, the infinite time expectation value is given by $\langle j \rangle = \langle J - \langle J \rangle \rangle = \mathbf{0}$. Whether or not the eigenvalues of B are positive will be the main issue of this paper. If they are, and choosing a δ-distribution as initial condition: $S(j, 0) = \delta(j) = \Pi_r \delta(j_r)$, the distribution function $S(j, t)$ will be a Gaussian [14]:

$$S(j, t) = \det(2\pi C)^{-1/2} \exp(-\tfrac{1}{2} j^{\mathrm{T}} C^{-1} j) \tag{3.15}$$

where C is the correlation matrix

$$C(t) = \langle j j^{\mathrm{T}} \rangle \tag{3.16}$$

which depends on B and D, in the context of the Fokker–Planck equation (3.5). With certain requirements for the long-time behaviour of $\varepsilon(t)$ (decreasing to zero, but with divergent time integral [12]), the equilibrium state (3.4) will be reached.

In the present study we are interested in the behaviour of the system subject to a constant learning rate ε. We will encounter situations in which the matrices B and D commute. In that case, the long-time correlation matrix is simply given by

$$C(t=\infty) = \varepsilon^2 \int_0^\infty \mathrm{e}^{-\varepsilon B \tau} D\, \mathrm{e}^{-\varepsilon B^{\mathrm{T}} \tau}\, \mathrm{d}\tau = \varepsilon (B + B^{\mathrm{T}})^{-1} D \tag{3.17}$$

where the commutation is used in the last equality. Note that these results only hold in the stable case, i.e. for positive definite B.

4. Rectangle-to-rectangle mapping

As a representative case, we will henceforth assume that the input is taken from a rectangle of size $A_x N_x \times A_y N_y$ with uniform sampling density $P(s) = (A_x N_x A_y N_y)^{-1}$. The stationary configuration is then one of eight possible configurations, as the algorithm is invariant under reflection and rotation, where the rotation group has four elements for the geometry considered.

Next, we impose periodic boundary conditions in input space as well as in the network, such that the stationary state becomes translationally invariant. We measure the distance in the network in units of the nearest-neighbour inter-neuron spacing. We

are further free to take the orientation in the network such that the x- and y-direction correspond to those in input space. After these remarks, we can write the stationary state as

$$\langle J\rangle_r = Ar \tag{4.1}$$

where A is a diagonal matrix with elements A_x and A_y. Then, the feature sets are of size $A_x \times A_y$. Equation (4.1) satisfies equation (3.4). We are interested in the stability of this stationary configuration. The form of the distribution function is specified if we calculate the correlation matrix (3.16), which in special cases reduces to (3.17).

Due to the translational invariance, the 2×2 matrices $B_{rr'}$ and $D_{rr'}$ depend only on the difference $r-r'$. Therefore, we can decouple the Fokker–Planck equation (3.5) if we represent $S(j)$ in terms of Fourier mode amplitudes

$$\hat{j}_k = \frac{1}{\sqrt{N}} \sum_r e^{ik\cdot r} j_r \tag{4.2}$$

etc. Recall that $N = N_x N_y$. By separation of variables, i.e. setting

$$\hat{S}(\hat{j}) = \prod_k \hat{S}_k(\hat{j}_k) \tag{4.3}$$

the resulting mutually independent Fokker–Planck equations are:

$$\frac{1}{\varepsilon}\partial_t \hat{S}_k(j_k, t) = \nabla_{\hat{j}_k} \cdot \hat{B}(k)\hat{j}_k \hat{S}_k(\hat{j}_k) + \frac{\varepsilon}{2}\nabla_{\hat{j}_k} \cdot \hat{D}(k)\nabla_{\hat{j}_k}\hat{S}_k(\hat{j}_k) \tag{4.4}$$

where

$$\hat{B}(k) = \sum_r e^{ik(r-r')} B_{rr'} = \sum_r e^{ik\cdot r} B_{r0}$$

$$= \frac{1}{N}[\hat{h}(\mathbf{0})\mathbb{1} - \hat{h}(k)\hat{a}(k) - i(A\nabla_k \hat{h}(k))\hat{b}(k)^T] \tag{4.5}$$

$$\hat{D}(k) = \frac{1}{N}[(A\nabla_k \hat{h}(k))(A\nabla_k \hat{h}(k))^T + M\hat{h}(k)^2] \tag{4.6}$$

with

$$M = \int_{F_r(\langle J\rangle)} ds(ss^T - \langle s\rangle_r \langle s\rangle_r^T) = \frac{1}{12N}\begin{pmatrix} A_x^2 & 0 \\ 0 & A_y^2 \end{pmatrix} \tag{4.7}$$

where M is the correlation matrix of the inputs s over feature set F_r. Due to translational invariance, the feature sets differ only by their centroids, but are otherwise identical. Hence M does not depend on r. The Fourier transform of the neighbourhood function is

$$\hat{h}(k) = \sum_r e^{ik\cdot r} h_{rr'} \tag{4.8}$$

independent of r'. The matrix a and the vector b are given by

$$a_{rr'} = \left.\frac{\partial\langle s\rangle_r}{\partial J_{r'}}\right|_{J=\langle J\rangle} \tag{4.9}$$

$$b_{rr'} = \left.\frac{1}{\bar{P}_r}\frac{\partial \bar{P}_r(j)}{\partial J_{r'}}\right|_{J=\langle J\rangle}. \tag{4.10}$$

a measures the shift of the centre of a feature set and b the change of the corresponding volume, under small deformations of the stationary state. They do not depend on the neighbourhood function, but only on the geometry of the stationary configuration, which is related to the network geometry. Carefully doing the geometry yields:

$$a_{rr'} = \delta_{r,r'} \operatorname{Diag}(\tfrac{1}{2}+\tfrac{1}{6}R^{-2}, \tfrac{1}{2}+\tfrac{1}{6}R^{2}) + (\delta_{r',r+e_x} + \delta_{r',r-e_x}) \operatorname{Diag}(\tfrac{1}{4}, -\tfrac{1}{12}R^{2}) + (\delta_{r',r+e_y} + \delta_{r',r-e_y}) \operatorname{Diag}(-\tfrac{1}{12}R^{-2}, \tfrac{1}{4}) \tag{4.11}$$

where $\operatorname{Diag}(d_1, d_2)$ is a diagonal matrix with elements d_1 and d_2, and R denotes the ratio of the sides of the feature set:

$$R \equiv \frac{A_y}{A_x}. \tag{4.12}$$

Further:

$$b_{rr'} = \tfrac{1}{2} \sum_{n=\pm e_x, \pm e_y} A^{-1} n (\delta_{r+n,r'} - \delta_{r,r'}). \tag{4.13}$$

The Fourier transforms are

$$\hat{a}(k) = \operatorname{Diag}(\tfrac{1}{2}+\tfrac{1}{2}\cos k_x + \tfrac{1}{6}R^{-2}(1-\cos k_y), \tfrac{1}{2}+\tfrac{1}{2}\cos k_y + \tfrac{1}{6}R^{2}(1-\cos k_x)) \tag{4.14}$$

$$\hat{b}(k) = -\mathrm{i} \begin{pmatrix} A_x^{-1} \sin k_x \\ A_y^{-1} \sin k_y \end{pmatrix}. \tag{4.15}$$

With constant ε, the matrix $\hat{B}(k)$ gives the speed with which a fluctuation with wavevector k decays exponentially with time. From here on we will only deal with infinite time quantities, i.e. when the distribution function S has reached its stationary form. For the case of commuting $\hat{B}$ and $\hat{D}$, the variances of the fluctuations are the diagonal elements of the correlation matrix, which is given by

$$C_k = \langle \hat{j}_k^{*} \hat{j}_k^{\mathrm{T}} \rangle = \varepsilon (\hat{B}(k) + \hat{B}^{\mathrm{T}}(k))^{-1} \hat{D}(k) \tag{4.16}$$

where * denotes complex conjugation.

For symmetry reasons, we only have to consider k-vectors in the interval $[0, \pi] \times [0, \pi]$. It is convenient to consider a representative case. We choose the wavevector k along the x-axis. The Cartesian components of the deviation vectors can then be distinguished in a component parallel to the wavevector and one perpendicular to it. The situation with k along the y-axis can be obtained by interchanging (A_x, N_x) with (A_y, N_y). For an isotropic system we could have taken any direction of k, but the square lattice is not isotropic.

Note that in general the only off-diagonal elements in $\hat{B}(k)$ occur in the last term of (4.5). One easily checks that for any wavevector k along a coordinate axis the matrices $\hat{B}(k)$ and $\hat{D}(k)$ are diagonal, so (4.16) applies and also $C(k)$ is diagonal. This is due to the fact that $\hat{b}$ is an odd function of k and $\hat{h}$ an even function of k.

If we introduce the notation, with $k = (k, 0)^{\mathrm{T}}$,

$$A_{\parallel} = A_x \qquad A_{\perp} = A_y \qquad \hat{h}(k) \equiv \hat{h}((k, 0)^{\mathrm{T}}) \tag{4.17}$$

the resulting expressions for the variances, i.e. the diagonal elements of $C(k)$, are

$$\langle|\hat{j}_{\parallel}(k)|^2\rangle \equiv \frac{\varepsilon D_{\parallel}}{2B_{\parallel}} = \varepsilon A_{\parallel}^2 \frac{\frac{1}{2}\hat{h}'(k)^2 + \frac{1}{24}\hat{h}(k)^2}{\hat{h}(0) - \frac{1}{2}\hat{h}(k)(1+\cos k) - \hat{h}'(k)\sin k} \tag{4.18}$$

$$\langle|\hat{j}_{\perp}(k)|^2\rangle \equiv \frac{\varepsilon D_{\perp}}{2B_{\perp}} = \varepsilon A_{\perp}^2 \frac{\frac{1}{24}\hat{h}(k)^2}{\hat{h}(0) - \hat{h}(k)(1+\frac{1}{6}R^2(1-\cos k))}. \tag{4.19}$$

The overall factors $A_{\parallel}^2$ and $A_{\perp}^2$ ensure a correct scaling behaviour. Apart from this factor, the longitudinal variance does not depend on the dimensions of input space, unlike the transverse variance. The denominators in these expressions are the eigenvalues of the matrix $\hat{B}(k)$, which we have denoted by $B_{\parallel}(k)$ and $B_{\perp}(k)$, respectively. $\hat{h}(k)$ is a sum of cosines of multiples of k. So for $k \to 0$ we have $\hat{h}(0) - \hat{h}(k) = \mathcal{O}(k^2)$, corresponding to a $1/k^2$ divergence of the variances for small k, i.e. zero-eigenvalues of the matrix $\hat{B}(\mathbf{0})$. An overall translation of the J (i.e. a $k = \mathbf{0}$ mode) is not restored. This is caused by the translational invariance of the system.

Let us, for example, consider the typical neighbourhood function (2.4), with Fourier transform

$$\hat{h}(\boldsymbol{k}) = 1 + 2\cos k_x + 2\cos k_y \qquad \hat{h}(k) = 3 + 2\cos k \tag{4.20}$$

we have

$$\langle|\hat{j}_{\parallel}|^2\rangle = \varepsilon A_{\parallel}^2 \frac{44\sin^2 k + 12\cos k + 13}{12(1-\cos k)(11+6\cos k)} \tag{4.21}$$

$$\langle|\hat{j}_{\perp}|^2\rangle = \varepsilon A_{\perp}^2 \frac{(3+2\cos k)^2}{4(1-\cos k)(12-3R^2-2R^2\cos k)}. \tag{4.22}$$

Analysis of the denominator of the last expression, essentially the eigenvalue $B_{\perp}$, shows that the *linear stability* of the transverse mode *breaks down* for a critical ratio of the feature set dimensions:

$$R_c^0 = \sqrt{\frac{12}{5}} \simeq 1.549. \tag{4.23}$$

For $R = A_{\perp}/A_{\parallel} > R_c^0$, the small-$k$ or long-wavelength modes become unstable. For example, for $R = 1.65$ all modes with $k < \pi/4$ are linearly unstable, for $R = 2$ those with $k < \pi/2$, and for $R = 3$ those with $k < 0.81\pi$. The same value was found in [12] for a mapping from a *three*-dimensional input space to a *two*-dimensional network of formal neurons, which we will comment on briefly. The first two dimensions of the input concern a square part of space and the network too has equal numbers of units in either direction. As long as the additional third dimension is negligible, the square-to-square map is trivially given by (4.1). However, with the third dimension becoming more important, the two-dimensional variety containing the ends of the weight vectors J has to account more and more for that third dimension. At some point, when the size ratio equals $\sqrt{12/5}$, the variety really starts to fold into the third dimension. In the Fourier analysis this corresponds to initially some and finally all modes becoming unstable. As it is a representation of a three-dimensional space by a two-dimensional network, the property of topology preservation is not the same as the property of preservation of ordering.

In the present case of a two-dimension-to-two-dimension mapping, however, ordering and topology are closely related. Our analysis implies that it may happen that the feature sets of two adjacent network units are interchanged, while the relative positions of the feature sets of surrounding units remain unaltered. Consequently the topology of the network is no longer the same as the topology described by the ends of the weight vectors.

We stress that this instability and possible *topology violation* occurs for relatively small ratios of the input space dimensions: the first modes become linearly unstable already at $R \simeq 1.549$, which is of the order of unity. It is thus not necessary to go to high or extreme values for R.

In the following we will study this effect in more detail for general neighbourhood functions.

5. Critical ratios

In this section we analyse the critical ratios for general neighbourhood functions $h_{rr'}$. Intuitively, one expects the long-range fluctuations to be damped more effectively if the range of $h_{rr'}$ is wider; this would correspond to a larger critical ratio. We investigate this in more detail.

Before we start, we remark that for ratios smaller than one, there is no instability of the transverse mode. This is clear from (4.21) and (4.22), and is also true for the general case given in (4.18) and (4.19). But of course, for $R<1$ it is more interesting to take the wavevector k along the y-direction, so that $A_\parallel = A_y$, $A_\perp = A_x$ and thus $R>1$. We conclude that we should only consider $R \geqslant 1$. Unless, of course, the critical ratio is smaller than unity, in which case even the square-to-square map is unstable.

First we look at the modes $k=(k, 0)$. Later we will also consider $k_y \neq 0$.

5.1. Modes $k=(k, 0)^T$

Let us write $h_{r0}=h_{lm}$, where $l=r_x$, $m=r_y$, satisfying the symmetry: $h_{lm}=h_{|l|,|m|}$. Then

$$\hat{h}(k)=\sum_l e^{ilk} \sum_m h_{lm} = \sum_{l=0}^{\infty} H_l \cos lk \tag{5.1}$$

where

$$H_l \equiv (2-\delta_{l0})\left(h_{l0}+2\sum_{m=1} h_{lm}\right) \tag{5.2}$$

The eigenvalue $\hat{B}_\perp(k)$ (see (4.19)) then vanishes if

$$\sum_{l=1} H_l(1-\cos lk) = \tfrac{1}{6}R^2(1-\cos k)\sum_{l=0} H_l \cos lk. \tag{5.3}$$

This equation determines the ratio $R_c(k, 0)$ for which the modes with wavevector $(k, 0)^T$ become unstable. The quantity we are finally after is the minimum

$$\mathscr{R}_c \equiv \min_{k_x, k_y} R_c(k_x, k_y). \tag{5.4}$$

Here we have $k_y=0$. If we further define

$$R_c^0 \equiv \lim_{k\to 0} R_c(k, 0) \tag{5.5}$$

a *small k* analysis yields

$$R_c^{02}=6\sum_{l=1} l^2 H_l \Big/ \sum_{l=0} H_l \tag{5.6}$$

i.e. proportional to the second moment of h in the parallel direction. (Note that $\Sigma_{l=0} H_l=\Sigma_{l,m} h_{lm}=\hat{h}(\mathbf{0})$.) For this ratio the $\mathcal{O}(k^2)$-term of the eigenvalue $\hat{B}_\perp$ vanishes. The $\mathcal{O}(k^4)$-term is positive. For larger ratios, however, the $\mathcal{O}(k^2)$-term becomes negative, and the stationary state (3.4) is unstable.

It is clarifying to consider some simple examples for the neighbourhood function. We can take for instance a '*cross*' *neighbourhood*, i.e. $h_{lm}=1$ only on the axes $l=0$ and $m=0$, up to $n_\parallel(n_\perp)$ in the parallel(perpendicular) direction. Then [15]

$$R_c^{02}=\frac{2n_\parallel(n_\parallel+1)(2n_\parallel+1)}{1+2n_\parallel+2n_\perp}. \tag{5.7}$$

Or we can take the '*rectangle*' *neighbourhood* where $h_{lm}=1$ only on a rectangle of size $(1+2n_\parallel)\times(1+2n_\perp)$ and otherwise zero. This is an example of a neighbourhood function which is separable. In the case of separability, i.e. when we have the form

$$h_{lm}=f_\parallel(l)\,f_\perp(m) \tag{5.8}$$

the upper limit in the sum in (5.2) is independent of l and the relevant quantity H_l/H_0 in (5.3) reduces to $f_\parallel(l)/f_\parallel(0)$. The extent of the neighbourhood in the perpendicular direction is irrelevant under these conditions. Note that the Gaussian is an example of this class of functions. For the rectangular neighbourhood function with $f_\parallel(l)=$constant up to $l=n_\parallel$, we find

$$R_c^{02}=2n_\parallel(n_\parallel+1) \tag{5.9}$$

independent of $n_\perp$.

Note that $R_c^0\sim\mathcal{O}(n_\parallel)$ for large $n_\parallel$.

As another toy case we consider $h_{0,0}=1$, $h_{lm}=\beta$ if $l^2+m^2=1$, and zero otherwise. Then

$$R_c^{02}=\frac{12\beta}{1+4\beta}. \tag{5.10}$$

a monotonic function of β, with maximum 3 for $\beta\to\infty$.

It is also useful for the sequel to have analytic results for $k=\pi$. Just substituting this value in (5.3) yields critical ratios for the 'cross' and 'rectangle' neighbourhood functions. The results are:

$$R_c^2(\pi,0)=3\,\frac{1+2n_\parallel-(-1)^{n_\parallel}}{2n_\perp+(-1)^{n_\parallel}} \tag{5.11}$$

and

$$R_c^2(\pi,0)=6n_\parallel \tag{5.12}$$

respectively. The latter expression (for 'rectangle') only applies for even $n_\parallel$. For odd $n_\parallel$ the mode $k=(\pi,0)^T$ is stable for all R. Note that both $k=\pi$ results behave as $R_c(\pi,0)\sim\mathcal{O}(\sqrt{n})$. More neighbours have to be 'dragged along' in order to remove this instability with respect to the $k=\mathbf{0}$ instability.

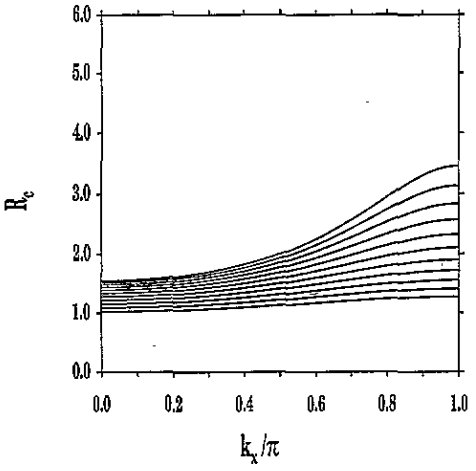

Figure 1. Phase diagrams for the typical neighbourhood function (2.4), i.e. $n_\parallel = n_\perp = 1$, with h_{ri} proportional to a Gaussian: $\exp(\alpha(r_x^2 + r_y^2))$. Upper line for $\alpha = 0$, others for $\alpha = 0.2, 0.4, \ldots, 2.0$.

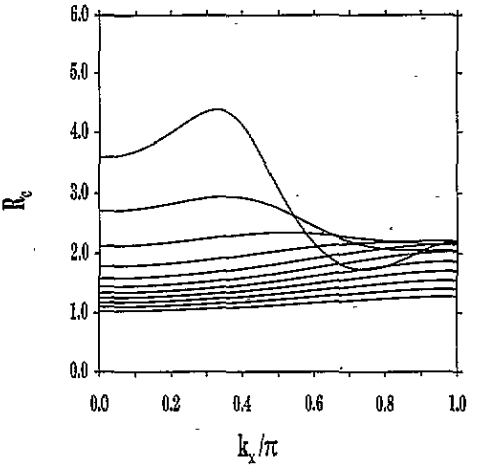

Figure 2. Same as figure 1 for the 7×7 'cross' neighbourhood function, i.e. $n_\parallel = n_\perp = 3$.

One could argue that the small-k instabilities may be considered those with the greatest impact, as they are relevant for large scales. For the typical neighbourhood function (2.4) they are the only ones to occur. However, for other forms of the neighbourhood function modes with finite k may become unstable at a smaller ratio. So we need to consider at least the complete expression (5.3). In figures 1 and 2 we give examples for a 'cross' neighbourhood, and in figure 3 for a rectangular neighbourhood.

Some examples for which finite-$(k_x, 0)$ modes become first unstable are the following. In these examples $h_{lm} = 1$ or 0.

- Rectangle: with $n_\parallel = 3$ with $R_c^0 \simeq 4.90$ and $R_c(0.72\pi, 0) \simeq 4.42$.

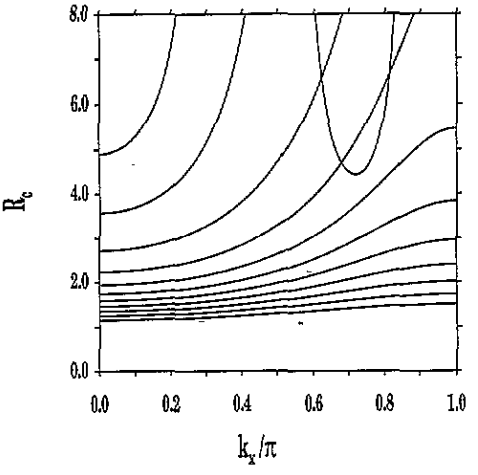

Figure 3. Same as figure 1 for the 7×7 'rectangular' neighbourhood function, i.e. $n_\parallel = n_\perp = 3$.

- 'Cross' neighbourhood:

$$(n_\parallel, n_\perp) = (2, 2) \text{ with } R_c^0 \simeq 2.58 \text{ and } R_c(\pi, 0) \simeq 1.55.$$

$$(n_\parallel, n_\perp) = (3, 3) \text{ with } R_c^0 \simeq 3.60 \text{ and } R_c(0.74\pi, 0) \simeq 1.73.$$

Finally, we multiply h_{lm} by $\exp(-\alpha(l^2+m^2))$. For α sufficiently large, the precise form of the region where h is non-zero is not relevant. Recall that convergence to equilibrium is obtained for $\varepsilon \to 0$ and the width of h to zero. This would correspond to $\alpha \to \infty$. In figure 1, figure 2 and figure 3 we have also plotted curves with non-zero α. For large α, corresponding to a very peaked h, the critical ratio becomes even less than 1 for all k.

A very peaked h may also be described by (5.10) with β small. We observe that for $\beta > \beta_c = \frac{1}{8}$ the critical ratio is smaller than one, so even the $R=1$ case is unstable. Including the next-nearest neighbours with strength β we would obtain, using (5.2) and (5.6), $\beta_c = \frac{1}{28}$, and by including them with strength β^2 we would have $\beta_c = \frac{1}{10}$. Here we have the important result that apparently the 'dragging' along of the neighbouring synapses has a lower bound for the map to be linearly stable.

Table 1. Critical ratios, and the modes which become unstable first. The first column contains the neighbourhood function h, denoted as $(n_\parallel, n_\perp)$r or $(n_\parallel, n_\perp)$c for rectangular or 'cross' neighbourhood functions, respectively. The second column contains the minimal critical ratio, and the third column the k_x-value for which this minimum is attained. In the fourth and rightmost column we give the ratio at which the $k=0$-mode becomes unstable.

h	$R_c\|_{k_y=0}$	k_x	$R_c\|_{k_x=k_y=0}$
(1, 1)c	1.549	0	1.549
(2, 2)c	1.549	π	2.582
(3, 3)c	1.735	2.332	3.595
(4, 4)c	1.633	π	4.602
(5, 5)c	1.710	2.612	5.606
(6, 6)c	1.664	π	6.609
(7, 7)c	1.706	2.746	7.611
(8, 8)c	1.680	π	8.613
(1, 1)r	2	0	2
(2, 2)r	3.464	π	3.464
(3, 3)r	4.426	2.257	4.899
(4, 4)r	4.899	π	6.325
(5, 5)r	5.579	2.574	7.746
(6, 6)r	6	π	9.165
(7, 7)r	6.547	2.724	10.583
(8, 8)r	6.928	π	12

Some typical values of the critical ratios are given in table 1, together with the corresponding neighbourhood functions and the k_x-value of the mode which is the first to become unstable.

5.2. *Modes* $k = (k_x, k_y)^T$

The above analysis determines the ratios $R_c(k)$ for which the modes $(k, 0)^T$ become unstable. However, for the system to be declared stable one usually requires *all* individual modes to be stable. At the moment we only have derived upper bounds for the ratios at which there is stability.

The longitudinal components of the modes investigated above is always stable (because $\hat{B}_{\parallel} > 0$), so *a priori* we do not expect a general mode (k_x, k_y) with $k_y \neq 0$ to become unstable for yet smaller ratios, because neither Cartesian component of $\boldsymbol{j}(\boldsymbol{k})$ is completely perpendicular to the wavevector. These arguments would apply, however, to an isotropic system, which our square lattice of formal neurons unfortunately is not. So in order to determine at what critical input space ratio the system looses its linear stability we should carry out the stability analysis for general modes.

For $k_y \neq 0$ and $k_x \neq 0$, $\hat{B}(\boldsymbol{k})$ and $\hat{D}(\boldsymbol{k})$ are not diagonal; $\hat{B}(\boldsymbol{k})$ is not even symmetric. In general these matrices do not commute and therefore the last equality in (3.7) does not hold. Hence the analysis becomes much more complicated, see e.g. [18]. We have not carried out this analysis.

6. Simulations

To test some of the above theoretical results, we have performed some numerical simulations. We used square networks with $N_x \times N_y$ units, where $N_x = N_y$. The network was initialized with the stationary state given in (4.1). We used periodic boundary conditions. Every N_s iteration steps we took a snapshot and calculated the Fourier amplitudes. N_s was chosen such that on the average every unit was updated about 10 times between two successive snapshots. N_s depends on the number of units and on the form of the neighbouring function h. We only considered modes with $k_y = 0$.

We give a comparison between simulations and theory for some typical values of the ratio $R = A_{\perp}/A_{\parallel} > 1$ with small ε. Recall that for $R < 1$ nothing dramatic is happening and we should take the wavevector in the other lattice direction, and replace R^{-1} by R, which is then taken larger than 1. The square case ($R = 1$) has already been done in [12]. Simulations for rectangular input sets agree rather well with the theory. It is more interesting to test the theory with the more exotic 'cross' neighbourhood function. We present the stable cases 'cross' $(n_{\parallel}, n_{\perp}) = (1, 1)$ with $R = 1.5$ and 'cross' $(n_{\parallel}, n_{\perp}) = (3, 3)$ with $R = 1.67$ in figure 4 and figure 5, respectively. We plotted the variance of the orthogonal component. The theoretical curve for the cases presented is not reached for every k. This may be due to the *higher-order terms* which we left out of the analysis. They may prevent the system from moving away too far from its stationary state.

In figure 6 we give the parallel component for the latter case. We also show the unstable case $R = 2$ in figure 7. Although the system is unstable and the theory does not apply, the 'stable' modes still seem to follow the theoretical curve.

A map with a very large ratio ($R = 10$) is presented in figure 8. Almost every mode is linearly unstable here. It may be that for such high ratios the higher-order terms also fail to keep the solution sufficiently close to its stationary form. For one or both of these reasons, it turns out that at some points in the map the topological order is violated ('folds in the fishing net'). The large-scale order, however, is preserved. The formation of these topological maps is an essentially nonlinear phenomenon. A *linear* stability analysis, as presented in this paper, only applies close to the stationary state (3.4), and does not explain the preservation of global order. However, the present study provides a better understanding of the nature of the map.

7. Discussion

The Kohonen algorithm aims at ordering *any* input according to its major axes. The general scheme for obtaining this ordered representation is as follows. One starts with

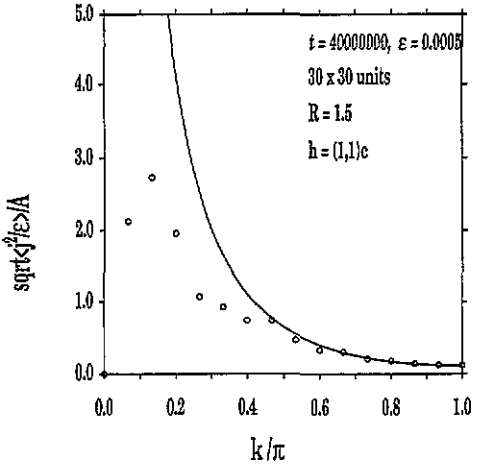

Figure 4. Comparison of theory and simulations for the neighbourhood function (2.4) and $R=1.5$, while $R_c=R_c^0\simeq 1.549$. Plotted is the variance of the fluctuations orthogonal to $k=(k,0)^T$.

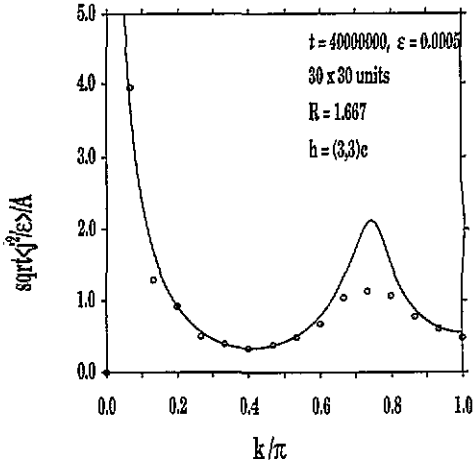

Figure 5. Comparison of theory and simulations for the 3×3 'cross' neighbourhood function with $R=1.67$, while $R_c^0=3.60$ and $R_c\simeq R_c(0.74\pi,0)\simeq 1.73$. Plotted is the variance of the fluctuations orthogonal to $k=(k,0)^T$.

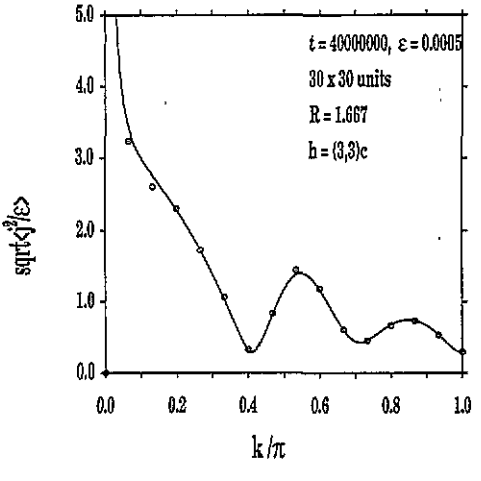

Figure 6. The fluctuations parallel to $k=(k,0)^T$, corresponding to figure 5.

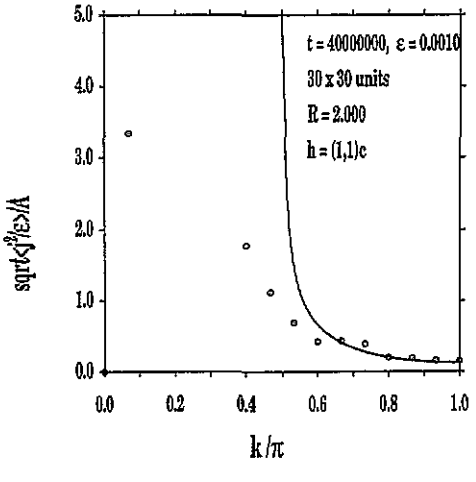

Figure 7. Transverse fluctuations of model with neighbourhood function (2.4) and $R=2$. This situation is unstable. The 'stable' modes, i.e. with $k>\frac{1}{2}\pi$ still follow the theory rather well.

a rather wide neighbourhood function $h_{rr'}$ and a rather large learning parameter ε. This prevents the algorithm from ending up in a 'twist' or 'butterfly', a metastable state in which for instance J_r is a monotonically *increasing* function of r_x on one end of the network (say $r_y=0$), and a monotonically *decreasing* function of r_x at the other end of the network ($r_y=N_y$). The metastable states of the one-dimensional Kohonen map have recently been studied [13]. Final convergence of the map is obtained by decreasing the range of h, and decreasing $\varepsilon(t)$.

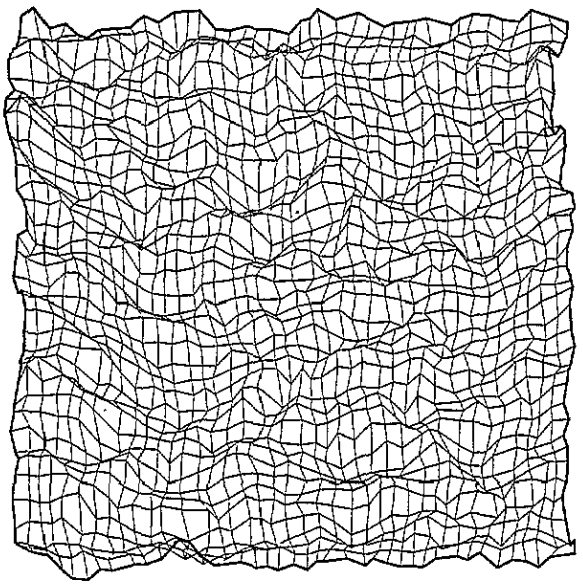

Figure 8. Map with $R=10$ (horizontal size/vertical size). 40×40 units, $\varepsilon=0.1$ h as in (2.4). Started from the stationary state, 10^6 steps are taken. One easily identifies points where topological order is violated.

The equations for the stationary map were given in [11]; $\varepsilon(t)$ has to decay to zero for $t\to\infty$, while $\int \varepsilon(t)\,dt$ has to diverge [12].

The stationary map equations were derived (i) for a continuous network with (ii) width of h to zero and (iii) $\varepsilon\to0$. However, in a realistic situation none of these will be the case. First because the number of units used in a computer simulation is finite. Probably the same remark applies for the small part of the brain which the Kohonen model usually models. So h with small width necessarily has also a finite number of units. Second, in order to let the system interact with its environment by adapting to changing circumstances, one has to maintain the learning parameter at some finite value. The present study incorporates these two facts.

The results described in this paper indicate the difficulties that may arise in applications of the Kohonen map.

We provide arguments that the ranges and units in which the sensory input is measured and subsequently presented to the network, have to be chosen with some care. Otherwise, the map becomes linearly unstable. We have also given upper bounds for the 'peakedness' of h, in order for the regular case $R=1$ to be stable.

The instabilities we find lead to *violations* of the important property of topology preservation. This is unlike the instability found earlier [12] for the representation of a three-dimensional input space by a two-dimensional network.

Inclusion of higher-order terms in the theory complicates it dramatically, because then Fourier transformation does not decouple the equations for different $\boldsymbol{k}$.

The instabilities we have found can only occur for dimensionality greater than one. In one dimension there is only one mode, the longitudinal one, which is always stable.

Some remarks can be made concerning the implications of our results.

If inputs are taken from the input set with homogeneous sampling density, one could simply rescale the input ranges in order to prevent instabilities; the anisotropy

of the feature sets has to remain within bounds given in this paper. However, this assumes prior knowledge of the inputs, contrary to what the model was 'invented' for.

More serious are the problems when the input set is sampled in an inhomogeneous manner. This causes the stationary state (see for instance [11]) to have feature set anisotropies that vary over the network. Simple rescaling is impossible then. In a general application this will be the case. Take for instance the map formed on the basis of muscle spindle information, or even for the hypothetical case that arm position information is available through 'joint angle sensors' [16]. Even in the case of linear sensor characteristics, this involves a non-trivial Jacobian of the transformation from Cartesian coordinates to joint angles. Additional transformations arise from the nonlinearity of the sensor characteristics; see for instance [17]. In addition the space that is to be represented is likely to be sampled in an inhomogeneous manner.

The difficulties might be dealt with by a modification of the algorithm. One can let the neighbourhood function depend locally on the map formed thus far. The width in a certain direction should then be chosen as a function of the local ratio of the size of the feature set perpendicular and parallel to this direction. We have not tested this.

A further interesting point to investigate is whether the stability of the model is increased if the square network connectivity is replaced by a hexagonal connectivity.

We have not ruled out that for certain neighbourhood functions the first mode to become unstable may *not* have k along one of the axis. This would possibly originate from non-isotropy of the network, in which case it would be interesting to check if a network with hexagonal connectivity also has this property.

We conclude by summarizing the results. We have demonstrated that the topological map formed by the Kohonen algorithm may not be linearly stable if the feature sets of a square network of formal neurons are rectangular instead of square. We have given examples for which this leads to violations of order in the map. Higher-order terms are assumed to be responsible for the mere formation and the *large-scale* stability of the map. We have investigated some typical forms of the neighbouring function, and the corresponding critical ratios, for which certain modes become unstable. Finally we have suggested a local technique that lifts the instability.

Acknowledgments

The author would like to thank J J Denier van der Gon and H J J Jonker for interesting discussions, and E Orlowski for carrying out preliminary simulations.

References

[1] Amari S 1990 *Proc. IEEE* **78** 1443
[2] Amari S 1977 *Biol. Cybernetics* **27** 77
[3] Wilson H R and Cowan J D 1973 *Kybernetik* **13** 55
[4] von der Malsburg C 1973 *Kybernetik* **14** 85; 1976 *Proc. R. Soc.* B **194** 431
[5] Amit D J 1989 *Modeling Brain Function* (Cambridge: Cambridge University Press)
[6] van Velzen G A 1992 *Physica A* **185** 439
[7] Kohonen T 1982 *Biol. Cybern.* **43** 59
[8] Kohonen T 1988 *Self-Organization and Associative Memory* (Berlin: Springer)
[9] Domany E, van Hemmen J L and Schulten K (eds) 1991 *Models of Neural Networks* (Berlin: Springer)
[10] Kohonen T, Mäkisara K, Simula O and Kangas J (eds) 1991 *Proceedings of International Conference on Artificial Neural Networks* (Amsterdam: North Holland)

[11] Ritter H and Schulten K 1986 *Biol. Cybern.* **54** 99
[12] Ritter H and Schulten K 1988 *Biol. Cybern.* **60** 59
[13] Erwin E, Obermayer K and Schulten K 1992 *Biol. Cybern.* **67** 35
[14] van Kampen N G 1981 *Stochastic Processes in Physics and Chemistry* (Amsterdam: North Holland)
[15] Gradshteyn I S and Ryzhik I M 1980 *Table of Integrals, Series and Products* (New York: Academic)
[16] Coiton Y, Gilhodes J C, Velay J L and Rou J P 1991 *Biol. Cybern.* **66** 167
[17] Hasan Z 1983 *J. Neurophysiol.* **49** 989
[18] Khalil H K 1992 *Nonlinear Systems* (New York: MacMillan)