Biol. Cybern. 67, 35–45 (1992)

Biological Cybernetics

Self-organizing maps: stationary states, metastability and convergence rate

E. Erwin, K. Obermayer, and K. Schulten

Beckman Institute and Department of Physics, University of Illinois at Urbana – Champaign, Urbana, IL 61801, USA

Received July 22, 1991/Accepted in revised form December 18, 1991

Abstract. We investigate the effect of various types of neighborhood function on the convergence rates and the presence or absence of metastable stationary states of Kohonen's self-organizing feature map algorithm in one dimension. We demonstrate that the time necessary to form a topographic representation of the unit interval [0, 1] may vary over several orders of magnitude depending on the range and also the shape of the neighborhood function, by which the weight changes of the neurons in the neighborhood of the winning neuron are scaled. We will prove that for neighborhood functions which are convex on an interval given by the length of the Kohonen chain there exist no metastable states. For all other neighborhood functions, metastable states are present and may trap the algorithm during the learning process. For the widely-used Gaussian function there exists a threshold for the width above which metastable states cannot exist. Due to the presence or absence of metastable states, convergence time is very sensitive to slight changes in the shape of the neighborhood function. Fastest convergence is achieved using neighborhood functions which are "convex" over a large range around the winner neuron and yet have large differences in value at neighboring neurons.

1 Introduction

The self-organizing feature map (SOFM) algorithm (Kohonen 1982a, b, 1988) is a biologically-inspired method for constructing a structured representation of data from an *input space* by prototypes, called *weight vectors*. The weight vectors are associated with selected elements, the neurons, of an *image space* where metric relationships are defined between the elements. For any given data-set, the SOFM algorithm selects weight vectors and assigns them to neurons in the network. The weight vectors as a function of neuron coordinates are called the *feature map*. Topologically ordered feature maps are characterized by the fact that prototypes which are neighbors in the input space are mapped onto neighboring neurons. The SOFM algorithm can successfully form feature maps which are topologically ordered, or nearly so, in a variety of applications, where the input space and image space have the same or different dimensionality. However, no complete theory of feature map formation has yet appeared. Such a theory would be useful for optimizing the algorithm in technical applications and for developing new algorithms for cases where the original SOFM algorithm is not adequate.

In a companion article (Erwin et al. 1992) we began to provide a framework for answering some questions about the SOFM algorithm. We have proven that the one-dimensional SOFM algorithm will converge to a topographic representation of the unit interval [0, 1] by a linear chain of neurons given only that the neighborhood function, by which the weight changes in neurons in the neighborhood of the winning neuron are scaled, be monotonically decreasing. We also demonstrated that the dynamics of the one- or multi-dimensional algorithm may be described using a set of energy function, such that the weight values associated with each neuron tend to move to decrease their energy.

In this paper we will consider the convergence process in more detail, in particular the rate of convergence and the presence and absence of metastable states, which correspond to non-global minima of the energy functions. In Sect. 2 we will briefly describe the one-dimensional SOFM algorithm. In Sect. 3 we will discuss the issue of *metastable states*, stationary states of the algorithm which do not coresond to the topologically ordered, optimal mappings. We will prove that for a certain class of neighborhood functions there exist no metastable states, while for other types of neighborhood functions metastable states are present regardless of the parameters of the algorithm. We will also show that for the case of the widely used Gaussian functions, there exists a threshold value for their width, above which the topologically ordered state is the only stationary state of

Correspondence to: E. Erwin

the algorithm. However, for more narrow Gaussian functions, or for the simple step function, metastable states exist and the mapping algorithm may become "stuck" temporarily in these non-ordered configurations.

In Sect. 4 we will provide numerical and analytical results on the speed of ordering as a function of the algorithm's parameters. Due to the presence or absence of metastable states, convergence time is very sensitive to slight changes in the shape of the neighborhood function, so that neighborhood functions which are close in some function-space metric may yet cause the SOFM algorithm to require very different amounts of time to converge. Fastest convergence is achieved using neighborhood functions which are "convex" over a large range around the winner neuron in the network, and yet have large differences in value at neighboring neurons. For the typically chosen Gaussian function, these competing interests balance to give the shortest convergence time when the width of the Gaussian is of the order of the number of neurons in the chain. The results of this section offer suggestions for avoiding metastable states in practical applications.

2 The one-dimensional SOFM algorithm

The one-dimensional SOFM algorithm employs a set of neurons which are arranged in a linear chain. The location of any neuron in the network is specified by the scalar index s, which we will allow to take integer values between 1 and N, the number of neurons. A weight vector w_s, in this case a scalar from the interval [0, 1], is assigned to each neuron s to form the *feature map*.

Given an initial set of weight vectors, feature map formation follows an iterative procedure. At each time step t, a *pattern* v, an element of the data manifold in input space, is chosen at random. The neuron r whose weight value w_r is metrically closest to the pattern is selected

$$|w_r - v| = \min_s |w_s - v|, \tag{1}$$

and the weight values of all neurons are then changed according to the feature map update rule,

$$w_s(t+1) = w_s(t) + \epsilon h(r, s)[v(t) - w_s(t)], \tag{2}$$

where ϵ, the *learning step width*, is some small constant ($0 < \epsilon \ll 1$). The function $h(r, s)$ is called the *neighborhood function*. For most applications it has a maximum value of one for $s \equiv r$ and decreases with increasing distance between s and r.

For convenience we define a *state* of the network as a particular set of weight values $\{w_s | s = 1, 2, \ldots, N; w_s \in [0, 1]\}$, and a *configuration* as the set of states which are characterized by the same order relations among the scalar weights.

The original form of the SOFM algorithm employed a step function as the neighborhood function $h(r, s)$

$$h(r, s) \equiv H(|r - s|) = \begin{cases} 1, & \text{if } |r - s| \leqslant 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Later it was discovered that the algorithm could be made to converge more quickly in practical applications if a gradually decreasing function were used (e.g. Ritter et al. 1989; Lo and Bavarian 1991). In a companion paper we proved that for the one-dimensional case, the algorithm can be guaranteed to converge for any monotonically decreasing neighborhood function. However, we also observed that other properties of the neighborhood function seemed to affect the efficiency of the algorithm. In particular we will show here that the algorithm is more efficient when a so-called *convex* neighborhood function is used.

We define a neighborhood function to be *convex* on a certain interval of integer values $I \equiv \{0, 1, 2, \ldots, N\}$, if

$$|s - q| > |s - r|, |r - q| \Rightarrow [h(s, s) + h(s, q)] < [h(s, r) + h(r, q)] \tag{4}$$

holds for all $|s - q|$, $|s - r|$, $|r - q|$ within the interval I, and to be *concave* otherwise. If the indices r, s and q are allowed to be real numbers and the interval I is taken to be the set of all real numbers, this definition fits the usual notion of convexity. However, due to edge effects, the definition (4) allows some additional neighborhood functions to be classified as convex even though their second derivative is positive over a part of the range of allowed arguments.

To illustrate the consequences of neighborhood function shape, we choose specific neighborhood functions which do or do not need condition (4). We have selected the step function (3) above and the "Gaussian", "concave exponential", "compressed Gaussian", and "compressed concave exponential" neighborhood functions (Fig. 1) given by:

$$h(r, s) \equiv H(|r - s|) = \begin{cases} \exp(-(r - s)^2/\sigma^2), & (5) \\ \exp(-\sqrt{|r - s|}/\sigma), & (6) \\ 1 - \lambda(1 - \exp(-(r - s)^2/\sigma^2)), \text{ and} & (7) \\ 1 - \lambda(1 - \exp(-\sqrt{|r - s|}/\sigma)), & (6) \end{cases}$$

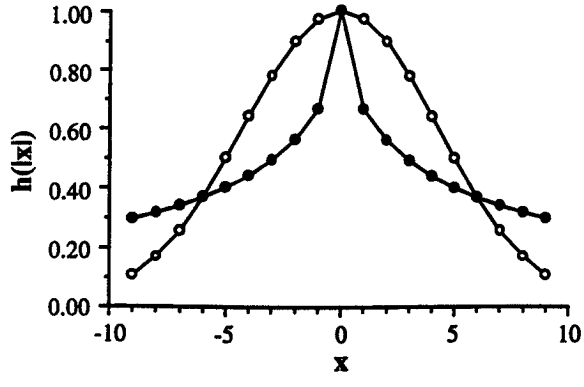

Fig. 1. Examples of the neighborhood functions used: Gaussian (open symbols), concave exponential (filled symbols); $\sigma = 6$ in both cases. The Gaussian function here is convex (4) over the interval $-9 < x < 9$

respectively. The constant λ in the compressed functions must be in the range $0 < \lambda \leqslant 1$, and will usually be small. For $\lambda = 1$, (7) or (8) reduce to the Gaussian (5) or concave exponential functions (6), respectively. Note that Gaussian functions (5) and (7) are convex within an interval around their maximum; for large enough values of σ, the Gaussian functions will be convex over the full range of their possible arguments, which are integer values less than N. An approximate lower bound on the value of σ is given by $\sigma > N\sqrt{2}$, although slightly lower values of σ may still result in neighborhood functions satisfying (4). The concave exponential functions (6) and (8) are concave for all values of the parameter σ.

3 Metastable states

It is known that the ordered state for the one-dimensional SOFM algorithm is stationary and stable (Kohonen 1982a; Ritter and Schulten 1986). However, little attention has been given to the existence of other stationary states, which correspond to stable "topological defects" of the feature meap. Here we will show that the existance of metastable states is linked to the shape of the neighborhood function. Metastable states with topological defects exist for any neighborhood function which is not "convex" (4).

For the derivation of these results it will be convenient to relabel the weight values so that their indices are arranged in ascending order in the input space. To avoid confusion we introduce new symbols u_x to refer to weight values laveled such that $x < y \rightarrow u_x < u_y$, $(x, y \in \{1, 2, \ldots, N\})$ and use the "old" symbols w_s to label weight values by the position of the corresponding neurons in the network. "New" indices can be converted to "old" indices by a permutation function $s = \mathscr{P}(x)$, which, however, is different for each configuration of the network. Thus we may write:

$$u_x \equiv w_{\mathscr{P}(x)} \equiv w_s \,. \tag{9}$$

Note that the arguments of the neighborhood function are always the indices s of w_s, since the neighborhood function is defined by neighborhood relationships in the image space (network), not in the input space. We will use the abbreviation $\hat{h}(s, y)$ for $h(\mathscr{P}(x), \mathscr{P}(y))$.

Let us denote the probability density of choosing a pattern v by $P(v)$. Then the average change $V_x[u] \equiv \langle u_x(t+1) - u_x(t) \rangle$ of the weight value u_x in one iteration, with the average taken over all possible patterns v, is given by

$$V_x[u] = \epsilon \int_0^1 \hat{h}(x, y)(v - u_x)P(v)\, \mathrm{d}v \,, \tag{10}$$

where y is the label of the winner neuron. The quantity $V_x[u]$ may be interpreted loosely as the average force acting to either increase or decrease the value of the weight u_x at the next iteration. Expanding (10) into a sum of integrals over all possible winner neurons y yields

$$V_x[u] = \epsilon \sum_{y=1}^{N} \hat{h}(x, y) \int_{v \in \Omega(y)} (v - u_x)P(v)\, \mathrm{d}v \,, \tag{11}$$

where each integral is evaluated only over the *Voronoi tessellation cell* of the neuron y, i.e. the area of the input space mapped onto neuron y. The Voronoi tessellation cell may be expressed as

$$\left.\begin{aligned} \Omega(1) &= \{v | 0 < v < \tfrac{1}{2}(u_1 + u_2)\} \,, \\ \Omega(y) &= \{v | \tfrac{1}{2}(u_{y-1} + u_y) < v < \tfrac{1}{2}(u_y + u_{y+1})\}, \\ &\qquad\qquad \text{for } 1 < y < N \,, \\ \Omega(N) &= \{v | \tfrac{1}{2}(u_{N-1} + u_N) < v < 1\} \,. \end{aligned}\right\} \tag{12}$$

We define a *stationary state* to be a set of weight values $\{u_x\}$ or $\{w_s\}$ which are characterized by vanishing forces: $V_x = 0$ for all x. The stationary states could correspond to local minima or maxima of the energy functions (given in Erwin et al. 1992), but we will show that all known stationary states correspond to local minima of the energy functions. Later we will differentiate between *stable* stationary states which belong to the absorbing, ordered configurations, and *metastable* states which belong to configurations with topological defects.

Let us consider the simplest case, where $P(v)$ is a constant. The condition $V_x[u] \equiv 0$ for the existence of a stationary state then simplifies to

$$0 = \sum_{y=1}^{N} \hat{h}(x, y) \int_{v \in \Omega_y} (v - u_x)\, \mathrm{d}v \,. \tag{13}$$

If the neighborhood function $h(r, s)$ is also constant, there exists only one stationary state with weight values given by $u_x = 1/2$, for all x.

Let us next consider neighborhood functions which can be written as a perturbation

$$h(x, y) \equiv 1 - \lambda g(x, y) \tag{14}$$

on this trivial case. Note that the "compressed" neighborhood functions (7) and (8) are of this form. Then we may derive the following theorem:

Theorem 1. *Given a constant probability distribution and a neighborhood function $h(x, y) \equiv 1 - \lambda g(x, y)$, such that $g(x, y) \equiv G(|x - y|)$ is positive and of order $\mathcal{O}(1)$, and $0 < \lambda \ll 1$, then*

1. *Any permutation function $\mathscr{P}(x)$ which when inserted into the right hand side of*

$$u_x = \frac{1}{2} + \frac{\lambda}{8}(\hat{g}(1, x) - \hat{g}(N, x)) + \frac{\lambda^2}{16}(\hat{g}(1, x)^2 - \hat{g}(N, x)^2) + \mathcal{O}(\lambda^3) \tag{15}$$

   leads to weights $\{u_x\}$ in ascending order, with the differences between adjacent weight values being greater than $\mathcal{O}(\lambda^3)$, describes a configuration which contains one stationary state.
2. *The stationary state $\{u_x^0\}$ is given by the r.h.s. of* (15).

The derivation of (15) in Theorem 1 is given in Appendix A. Note that the "compressed" neighborhood functions (7) and (8) meet the requirements of Theorem 1 when the factor λ is sufficiently small.

For an ensemble of maps with weight values located at one of the stationary states, application of the update rule (2) to each map, with patterns v independently chosen from the input space with the probability density $P(v)$, leads to no change in the average of each weight value across the ensemble. The weight values in any individual map must, however, change after each application of the update rule. The average effect of the update rule on maps whose weight values are near a stationary state is considered in the following theorem:

Theorem 2. *Given a constant probability distribution, then for the neighborhood functions of Theorem* 1

$$|\partial V_x/\partial u_y| < 0, \quad \forall x, y \in \{1, 2, \ldots, N\}, \tag{16}$$

for all sets of weight values $\{u_x\}$ *such that the derivatives exist, i.e. for all maps such that* $u_x \neq u_y$, $\forall x \neq y$.

This theorem may be proven by performing the indicated derivative and using the property $u_x > u_y$, $\forall x > y$ of the stationary state (15).

Theorem 2 states that the weight values of maps in a configuration containing a stationary state are more likely to move toward that stationary state than away from it, and thus they fluctuate around the values given by Theorem 1. However, unless the weights are in an ordered configuration, there is always a finite probability that a series of input patterns will cause an individual map to "escape", i.e. to change their configuration of weights. After a change in configuration, the weights will again be attracted to a stationary state, but this may not be the one in the previous configuration. For these reasons, we call stationary states *stable* only if they belong to the absorbing, ordered configuration, and *metastable* otherwise.

We may make several extensions of Theorem 1 concerning the existence of stationary states. Our first corollary states the conditions under which no metastable states exist.

Corollary 1. *If the conditions of Theorem* 1 *hold and* $h(x, y)$ *is convex in the sense of* (4), *then and only then all stationary states belong to ordered configurations.*

The proof of Corollary 1 is given in Appendix B. Although expansion (15) is no longer valid for neighborhood functions with $\lambda \approx 1$, we found empirical evidence from numerical simulations that even when the expansion (15) is not valid, metastable states exist for concave, but not for convex neighborhood functions.

When the conditions of Theorem 1 are met, it follows from $u_{x+1} > u_x$ and (15) that

$$g(\mathscr{P}(1), \mathscr{P}(x+1)) - g(\mathscr{P}(N), \mathscr{P}(x+1)) > g(\mathscr{P}(1), \mathscr{P}(x)) - g(\mathscr{P}(N), \mathscr{P}(x)), \tag{17}$$

for any two weights u_x and u_{x+1}. After $\mathscr{P}(1)$ and $\mathscr{P}(N)$ have been specified, there is at most one possible set of values for the other $\mathscr{P}(x)$'s which will fulfill (17). Thus

Corollary 2. *If the conditions of Theorem* 1 *are met, and* $h(x, y)$ *is concave, then non-ordered stationary states, i.e. metastable states, exist. If the number of neurons is* N, *then* (15) *predicts at most a total of* $N(N-1)$ *stable and metastable stationary states.*

When the conditions of Theorem 1 do not hold, it is no longer correct to neglect the terms of order $\mathcal{O}(\lambda^3)$ in (15). Corollary 2 no longer holds and more than one stationary state may exist for each pair of $\mathscr{P}(1)$ and $\mathscr{P}(N)$ values, thus the number of stationary states is no longer limited to $N(N-1)$. Indeed, for $\sigma \to 0$ in any of the four neighborhood functions, $N!$ metastable states exist, one for each configuration of the weights.

The conditions of Theorem 1 are met for the compressed neighborhood functions (7) and (8) for small λ and all but the smallest values of σ. For very small σ, $\hat{g}(1, x) - \hat{g}(N, x) < \mathcal{O}(\lambda^3)$ and the expansion to order λ^3 is not sufficient. The conditions of Theorem 1 are also met by the Gaussian (5) and concave exponential (6) functions, if their width σ is sufficiently large. In this case expansion of (5) and (6) with respect to $1/\sigma$ leads to the perturbations $\lambda G(|x|) \approx x^2/\sigma^2$, or $\lambda G(|x|) \approx \sqrt{|x|}/\sigma$, respectively. Since Gaussian neighborhood functions with broad enough σ are convex over the full range of their possible arguments, application of Corollaries 1 and 2 to Guassian functions yields that there exists a threshold value for σ above which no metastable states exist. An equation for determining this threshold value will be given in Sect. 4.3.

Figure 2 illustrates these results for a 10-neuron Kohonen chain and a compressed Gaussian neighborhood function (7) with $N = 10$, $\lambda = 10^{-8}$. For a 10-neuron

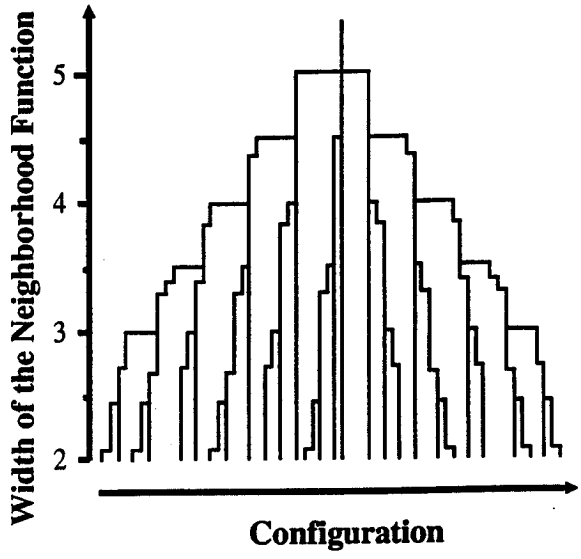

Fig. 2. Schematic diagram demonstrating how the number of metastable states increases for a 10-neuron chain as the width σ of the Gaussian neighborhood function is decreased. The horizontal axis is a somewhat abstract "configuration" axis. Only half of the configuration space is shown – each vertical line thus represents two configurations in one of the classes of metastable states listed in Table 1. The central line represents the class of ordered configurations. Class B of metastable states branches off from the central line at $\sigma \approx 5.02$; classes C and D becomes metastable at $\sigma \approx 4.53$

chain a maximum of 90 metastable configurations can be found from (15). These configurations can be grouped into the 25 classes which are listed in Table 1 along with the range of σ over which they are stable. All members of a class are related to each other by two elementary symmetry operations, i.e. by replacing all indices s by $(N-s+1)$, or replacing all weight values w_s by $(1-w_s)$.

For large σ, the perturbative part of (7) is convex, and the only stationary states are the two ordered stationary states, but as σ is lowered this term becomes concave and metastable configurations also begin to appear. As σ is reduced, the first set of metastable states to appear are the ones in the class labeled B in Table 1. In this configuration all of the weights are ordered except for the two corresponding to the neurons at the end of the chain, i.e. $(w_1 < w_2 < w_3 \cdots < w_{10} < w_9)$. Three symmetrical metastable states appear at the same value of σ, namely with $(w_1 > w_2 > \cdots w_{10} > w_9)$; $(w_2 < w_1 < w_3 < \cdots < w_9 < w_{10})$; and $(w_2 > w_1 > w_3 > \cdots > w_9 > w_{10})$. As σ is lowered further, more metastable states appear, with greater disorder. Some metastable states, such as B, remain metastable as σ is lowered, others are only stable over a limited range of σ. As σ goes to zero for any of the neighborhood functions considered, the conditions of Theorem 1 do not hold, and expansion (15) is no longer sufficient. Note that for the concave exponential function (8), a certain set of meatstable states exist for all values of σ; some of their configurations are indicated in Table 1.

Table 1. Table of metastable states for a compressed Gaussian neighborhood function (7) ($\lambda = 10^{-8}$). The range of values of σ over which a metastable state exists in each given configuration was calculated from (15). Since this equation is no longer valid for small σ, we only indicate states which are metastable for values of σ greater than 1.2. Configurations marked by a star (*) also contain metastable states for the concave exponential neighborhood function (8)

Class	Range of σ	Number of Members	Prototype									
			$\mathscr{P}(1)$	$\mathscr{P}(2)$	$\mathscr{P}(3)$	$\mathscr{P}(4)$	$\mathscr{P}(5)$	$\mathscr{P}(6)$	$\mathscr{P}(7)$	$\mathscr{P}(8)$	$\mathscr{P}(9)$	$\mathscr{P}(10)$
A*	∞–1.2000	2	1	2	3	4	5	6	7	8	9	10
B*	5.0245–1.2000	4	1	2	3	4	5	6	7	8	10	9
C*	4.5289–1.2000	2	2	1	3	4	5	6	7	8	10	9
D*	4.5289–4.3733	4	1	2	3	4	5	6	7	10	9	8
E	4.3733–1.2000	4	1	2	3	4	5	6	10	7	9	8
F*	4.0258–3.8473	4	2	1	3	4	5	6	7	10	9	8
G	4.0258–3.8473	4	1	2	3	4	5	10	6	9	8	7
H	3.8473–1.2000	4	2	1	3	4	5	6	10	7	9	8
I	3.8473–3.3954	4	1	2	3	4	5	10	9	6	8	7
J*	3.5132–3.3024	2	3	2	1	4	5	6	7	10	9	8
K	3.5132–3.3024	4	2	1	3	4	5	10	6	9	8	7
L	3.5132–3.3954	4	1	2	3	4	10	9	5	8	7	6
M	3.3954–1.2000	4	1	2	3	4	10	5	9	6	8	7
N	3.3954–3.3024	4	1	2	3	10	4	9	5	8	7	6
O	3.3024–1.2000	2	3	2	4	1	5	6	10	7	9	8
P	3.3024–2.6862	4	2	1	3	4	5	10	9	6	8	7
Q	3.3024–2.6862	4	1	2	3	10	4	9	8	5	7	6
R	2.9889–2.7269	4	3	2	1	4	5	10	6	9	8	7
S	2.9889–2.7269	4	2	1	3	4	10	9	5	8	7	6
T	2.9889–2.7269	4	1	2	3	10	9	8	4	7	6	5
U	2.7269–1.2000	4	3	2	4	1	5	10	9	6	8	7
V	2.7269–1.2000	4	2	1	3	4	10	9	8	5	7	6
W	2.7269–1.2000	4	1	2	3	10	9	8	7	4	6	5
X	2.6862–1.2000	4	2	1	3	4	10	5	9	6	8	7
Y	2.6862–1.2000	4	1	2	3	10	9	4	8	5	7	6
Z	2.4489–2.0864	2	4	3	2	5	1	10	6	9	8	7
AA	2.4489–2.0864	4	3	2	1	4	10	9	5	8	7	6
AB	2.4489–2.0864	4	2	1	3	10	9	8	4	7	6	5
AC	2.4489–2.0864	4	1	2	10	9	8	7	3	6	5	4
AD	2.0864–1.2000	2	4	3	5	2	1	10	9	6	8	7
AE	2.0864–1.2000	4	3	2	4	1	10	9	8	5	7	6
AF	2.0864–1.2000	4	2	1	3	10	9	8	7	4	6	5
AG	2.0864–1.2000	4	1	2	10	9	8	7	6	3	5	4
AH*	1.8857–1.2000	4	4	3	2	1	5	10	9	8	7	6
AI*	1.8857–1.2000	4	3	2	1	4	10	9	8	7	6	5
AJ*	1.8857–1.2000	4	2	1	3	10	9	8	7	6	5	4
AK	1.8857–1.2000	4	1	2	10	9	8	7	6	5	4	3
AL*	1.2810–1.2000	2	5	4	3	2	1	10	9	8	7	6
AM*	1.2810–1.2000	4	4	3	2	1	10	9	8	7	6	5
AN*	1.2810–1.2000	4	3	2	1	10	9	8	7	6	5	4
AO*	1.2810–1.2000	4	2	1	10	9	8	7	6	5	4	3
AP*	1.2810–1.2000	4	1	10	9	8	7	6	5	4	3	2

4 Rate of convergence

4.1 Ordering time

Let us define *ordering time* as the number of time steps required for a given map to reach an ordered configuration. In the case of the neighborhood functions (5)–(8), for $\sigma \to 0$ or $\sigma \to \infty$ the relative order of the weight values is not affected by the update rule and ordering time becomes infinite. For all other values of σ, the neighborhood function is a monotonously decreasing function of the distance $|r - s|$, and the ordering time must be finite. Thus there must exist an intermediate value $\sigma_{\min}$ which corresponds to a minimum in ordering time. The minimal value and its location differ between the neighborhood functions (5)–(8) and, in fact, crucially depend on the shape of the neighborhood function.

Figure 3a shows the average ordering time as a function of σ for the Gaussian neighborhood function (5) and an ensemble of 10-neuron Kohonen chains. Note that time is scaled by ϵ, since ordering time is proportional to ϵ for $\epsilon \ll 1$. Ordering time is minimal for $\sigma \approx 9$. For $\sigma \to \infty$ ordering time rises slowly and approaches a logarithmic dependence. The standard deviation of the number of time steps required (indicated by the error bars) is small and approximately constant. For small σ average ordering time rises very rapidly and the standard deviation is large. The large average ordering time with a large standard deviation is due to the presence of metastable states, which appear as the neighborhood function becomes progressively more concave. Constant standard deviation, and a logarithmic increase in ordering time with σ, is due to an initial "contraction" phase of the network. Ordering in the separate regimes of large and small σ will be discussed in detail below.

Similar graphs result for chains and a larger number of neurons N, but ordering time scales differently with N in the regimes of large and small σ. For large N ordering time becomes a function of σ/N for large σ. However, this is not true for small σ where the presence of metastable states causes ordering time to increase faster than a linear function of N. Figure 3b shows that the compressed Gaussian function gives results similar to the Gaussian function, except that time of ordering rises more rapidly as σ approaches zero.

For the case of the concave exponential (6) and compressed concave exponential (8) neighborhood functions, however, the presence of metastable states for all values of σ gives rise to much longer ordering times; so much longer, in fact, that is was infeasible to construct graphs similar to Fig. 3a, b. Ordering times for independent simulations with identical parameters spanned several orders of magnitude. Note that although the compressed Gaussian and compressed concave exponential functions differ only slightly (in some function-space metric) they give rise to ordering times which differ by several orders of magnitude. Hence the shape of the neighborhood function, e.g. its concavity or convexity, crucially influences the learning dynamics. In the following sections we will discuss the different convergence phenomena in the regimes where metastable states are, or are not present.

4.2 Contraction of weights

For convex functions, e.g. Gaussian functions with large σ, metastable states do not exist; the increase in ordering time as σ increases is completely due to the increase in the amount of time spent in the initial (random) configuration.

For $\sigma \to \infty$, the change in weight differences, $\Delta w_{rs} = (w_r - w_s)_{t+1} - (w_r - w_s)_t$, per iteration approaches zero as $1/\sigma^2$. In the initial random configuration the weights cover the unit interval completely, weight differences are much larger than $1/\sigma^2$ and no rearrangement of weights can take place. Therefore map formation must proceed in two steps: first the range of weight values must shrink until $\Delta w_{rs} = \mathcal{O}(1/\sigma^2)$ while maintaining the initial configuration, and then the weights must rearrange to form an ordered mapping.

Figure 4 shows the average ordering time t, the time t_c spend in the initial configuration, and the rearrangement time, $t - t_c$, as a function of σ for the Gaussian neighborhood function (5). The increase in ordering time above $\sigma_{\min}$ is completely due to an increase in t_c which empirically fits well with the logarithmic function

$$t_c \approx (1/\epsilon)\ln(l/l_c[h(r, s)]) \tag{18}$$

where l denotes the length of the interval spanned by the initial weight values and l_c denotes the length of the interval covered by the weights (*critical* length) after which the first rearrangements occur (see Fig. 5). The

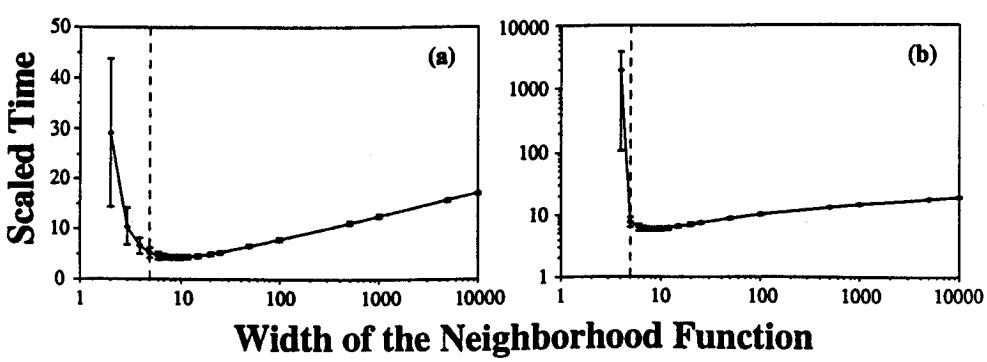

Fig. 3. Ordering time as a function of the width σ of the neighborhood function, for a Gaussian (**a**) and a "compressed" Gaussian (**b**) function ($\lambda = 0.1$). Ordering time is averaged over 1000 independent simulations for each data point. Error bars represent one standard deviation. Metastable states exist below $\sigma = 5.0254$ (*dotted line*)

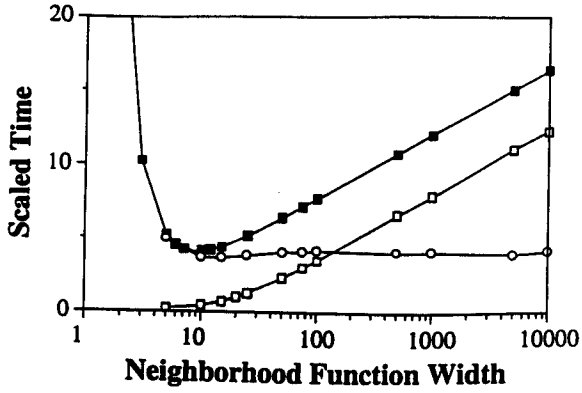

Fig. 4. Ordering time (*filled squares*), time spent in the initial configuration (*open squares*), and rearrangement time (*open circles*) as a function of σ for a ten-neuron Kohonen chain with a Gaussian neighborhood function

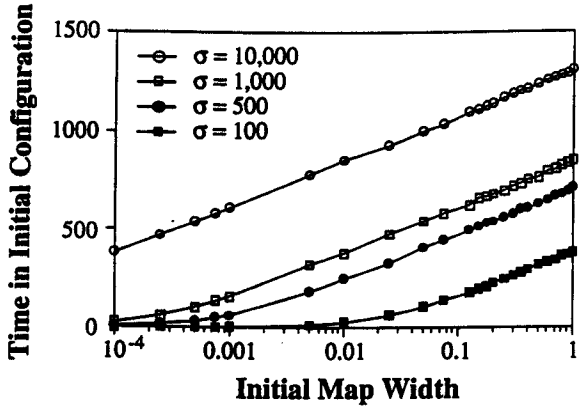

Fig. 5. The number of time steps spent in the initial configuration t_c is plotted against the length of the unit interval spanned by the weight values in the initial map, $l = \max(|w_i - w_j|)$, for a 10-neuron Kohonen chain and a Gaussian neighborhood function at several values of σ. The curves can be fit to the function $t_c \approx (1/\epsilon)\ln(l/l_c[h(r, s)])$, for large l, where l_c is an empirical constant which is a functional of $h(r, s)$

distance l_c is a functional of $h(r, s)$. Its dependence on σ determines the shape of the ordering-time curves for large σ.

Based on the mistaken assumption that for a Gaussian neighborhood function the number of "kinks" in the map, $N(t)$, could not increase in any time step, Geszti et al. (1990) inferred that the rate-limiting step in the ordering process was the elimination of the last "kink" in the map. They modelled this process as a random walk of the location of this kink in the chain of neurons, but failed to correctly predict ordering time as a function of σ. We can now see why this approach fails. For large σ, the rate-limiting step is the initial contraction of the range of the weights; after this step the ordering time is independent of σ. For small σ, the effect of metastable states on the ordering time must be considered.

4.3 Effect of metastable states

For the concave exponential functions (6) and (8), and for the Gaussian functions (5) and (7) with small σ, the long ordering times combined with a large standard deviation may be explained by the presence of metastable states. In some simulations the map will get "stuck" in metastable configurations for a large number of time steps, whereas in some simulations the algorithm will, by chance, avoid the metastable states.

For the compressed neighborhood functions (7) and (8), Theorem 1 may be used to predict when metastable stationary states should be present. Their effect on the ordering process may be observed in Fig. 6, which shows the percentage of maps in a particular configuration as a function of time for the compressed neighborhood functions (7) and (8). In each plot the heavy curve represents the percentage of maps, out of an ensemble of 10,000 initially random maps, which have reached one of the two ordered configurations, class A in Table 1, at a given time. The thin solid curves represent the percentage of maps in the configurations of classes B–G defined in Table 1, and the dashed curve which starts near 100 percent represents the total percentage of maps in all other configurations.

Figure 6a, b shows the population diagram for a compressed Gaussian neighborhood function. With large σ ($\sigma = 1000$) the only stationary states of the mapping alborithm should be the ordered configurations, and indeed we see in Fig. 6b that the mapping remains in the initial random configurations (dashed curve) until $t \approx 190$ while undergoing the contraction of weights discussed in the previous section. After $t \approx 190$, a rapid rearrangement of the weights takes place until $t \approx 220$ when all maps in the ensemble have reached the ordered configurations (heavy curve). At no time do the populations of any of the non-ordered configurations become significant.

The situation is quite different for the compressed Gaussian (7) with small σ ($\sigma = 3.5$). Figure 6a shows that for this small value of σ, the period of contraction of weights is not necessary, and rearrangement starts almost immediately. After a very short time most of the maps are trapped in one of the metastable configurations – which for this value of σ are of the form of classes B, C, and E-L of Table 1.

Figure 6c, d shows similar diagrams for the compressed concave exponential (8) with $\sigma = 3.5$, and $\sigma = 1000$, respectively. For both of these neighborhood functions metastable states exist in configurations B–D, and F, and in a few other configurations which are not plotted (see Table 1). As expected we do see a large percentage of the maps trapped in these metastable configurations, particularly configurations of the type B and D, for intermediate times. For both of these neighborhood functions, the percentage of maps in the ordered configurations approaches 100 percent as a logarithmic function of time, but for each the average ordering time is far greater than for the corresponding convex Gaussians in Fig. 6a, b.

Plots of the compressed Gaussian (7) and compressed concave exponential (8) functions against distance for $\sigma = 1000$ appear quite similar at short distances from the winner neuron. The concave exponential may at first appear to be an even better choice for the neighborhood function since it does decrease

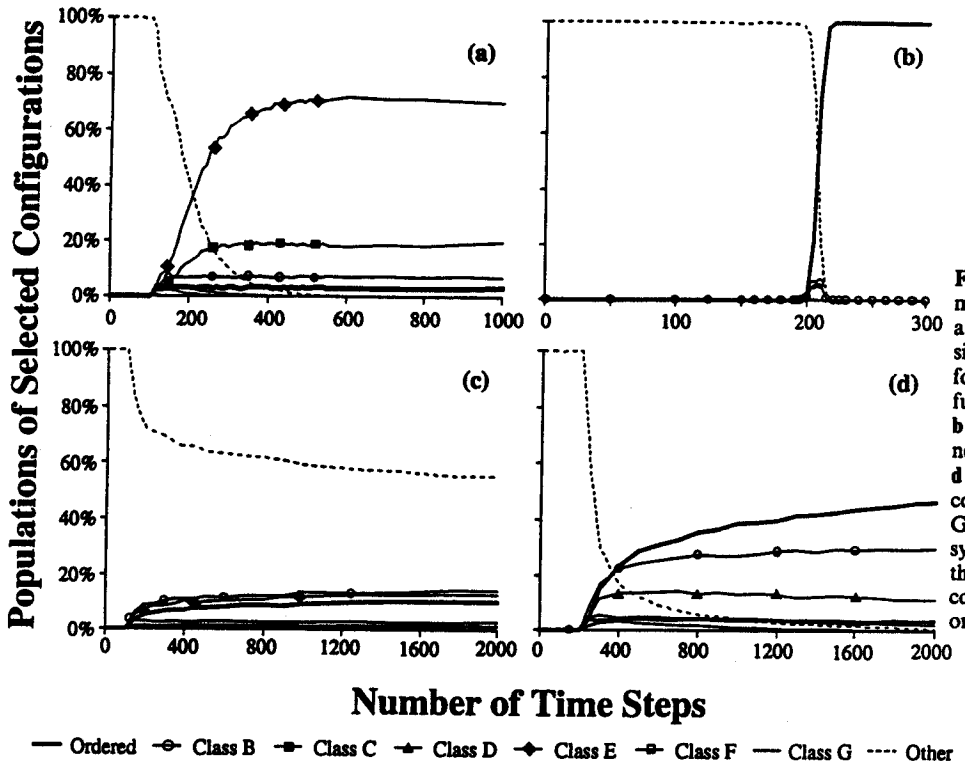

Fig. 6a–d. Percentage of ordered maps and maps in disordered configurations, out of an ensemble of 10,000 independent simulations, is shown as a function of time for a "compressed" Gaussian neighborhood function ($\epsilon = 0.1$, $\lambda = 10^{-4}$) with **a** $\sigma = 3.5$, **b** $\sigma = 1000$, and for a concave exponential neighborhood function with **c** $\sigma = 3.5$, and **d** $\sigma = 1000$. The curves for the disordered configurations in classes B, C, D, E, F, and G from Table 1 are denoted by the symbols above. The dashed curve represents the total percentage of maps in all other configurations. Plot symbols are ommitted on some curves to avoid clutter

more rapidly at small distances from the winner neuron. However, a 10-point one-dimensional array of neurons can self-organize into a map of the unit interval in fewer than 140 iterations with this Gaussian neighborhood function, while the ordering time with the concave exponential can be many orders of magnitude longer.

Although Theorem 1 and its corollaries hold only for the compressed functions, or for the Gaussian with large σ, we have empirically observed that Corollaries 1 and 2 correctly predict the existence or non-existence of metastable states for all monotonically decreasing neighborhood functions. Metastable states appear to exist for any concave neighborhood function, and not to exist for convex neighborhood functions. Figure 7a and b shows populations diagrams for the (non-compressed) Gaussian (5) function with $\sigma = 2$ and 1000, respectively, in the same format as Fig. 6. (Further examples can be seen in Erwin et al. 1991.) By using a convex function, such as a broad Gaussian, we can optimize ordering time and avoid having the algorithm get "stuck" in a metastable state. Since the first metastable state to appear as σ in lowered has the weights w_N and w_{N-1} in reverse order in the input space, i.e. configuration B in Table 1, we may find the threshold value of sigma above which no metastable state can exist by finding the lowest value of σ for which the neighborhood function between the three neurons w_1, w_{N-1} and w_N is convex (4). This threshold value can be found from the relation

$$h(N, N) + h(1, N) = h(1, N-1) + h(N-1, N-2) \tag{19}$$

$$1 + H(N-1) = H(N-2) + H(1) \tag{20}$$

by numerical methods.

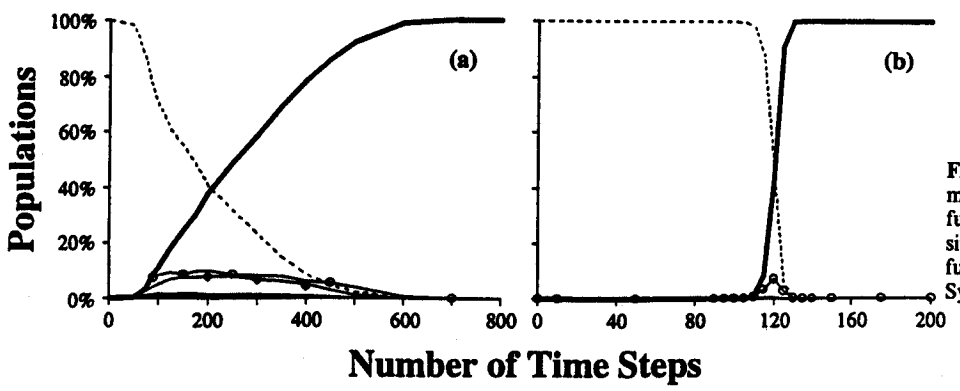

Fig. 7. Percentage of ordered maps and maps in disordered configurations as a function of time for 10,000 independent simulations for a Gaussian neighborhood function with **a** $\sigma = 2$ and **b** $\sigma = 1000$. Symbols as in Fig. 6

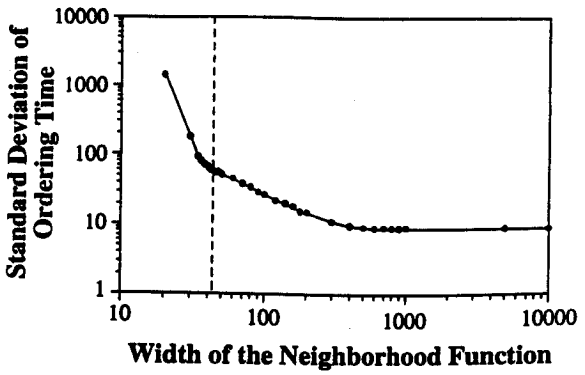

Fig. 8. The standard deviation of the ordering time plotted against σ for a 100-neuron Kohonen chain. For large σ the standard deviation of the ordering time approaches a constant value. At intermediate values, the standard deviation of the ordering time is inversely proportional to σ. For small values of σ the ordering time and its standard deviation increase rapidly. The dotted line shows the calculated value of $\sigma \approx 43.214$, below which metastable states should exist

Further evidence that metastable states arise for concave functions even when the expansion in (15) does not hold can be seen in Fig. 8. In this figure the standard deviation of the ordering time is plotted against σ for a 100-neuron Kohonen chain. For intermediate values of σ, the standard deviation of the ordering time is inversely proportional to σ; however for small values of σ, the ordering time, and its standard deviation both increase rapidly. The value of σ which marks the transition between these two behaviors appears to correspond to $\sigma \approx 43.214$ (shown as a dotted line in Fig. 8) below which value the neighborhood function becomes concave. The same behavior is observed in maps with a different number of neurons.

The earliest form of the SOFM algorithm employed a step function (3) for the neighborhood function (Kohonen 1982a, b, 1988). Later it was discovered that gradually decreasing functions, such as Gaussians, give rise to faster convergence of the algorithm in practical applications (e.g. Ritter et al. 1989; Lo and Bavarian 1991). Since it resembles the shape of a narrow Gaussian function, we might expect the step neighborhood functio to lead to metastable stationary states which result in the longer ordering times. However, Theorem 1 cannot be used to either confirm nor deny the existance of metastable stationary states in this case. Since the low-order terms in the expansion (15) evaluate to zero for many values of the parameter x, the higher-order terms in the expansion cannot be neglected. Evidence for the existence of metastable states may be obtained by making plots such as those shown in Figs. 6 and 7. Such graphs reveal that the ordering process follows a similar time course to that followed when metastable states are present. After a short period of contraction, the maps quickly rearrange so that the majority are in one of a few configurations, at least some of which probably contain metastable states. The configuration which holds the greatest number of non-ordered maps as the majority of maps reach the ordered configuration is the configuration E of Table 1.

5 Summary and discussion

The one-dimensional SOFM algorithm itself is of little practical importance. However, a thorough understanding of its properties is a necessary first step towards understanding the higher-dimensional versions of the algorithm which are used in applications. In this paper and its companion (Erwin et al. 1992) we have focused on the simplest form of the SOFM algorithm. This enabled us to provide exact results, such as a proof of ordering for general monotonically decreasing neighborhood functions, and a method of describing the behavior of the algorithm in terms of a set of energy functions. However, our results also reveal how complicated a full description of the SOFM algorithm must be. We proved that the algorithm, even in the one-dimensional form, cannot be described as following a gradient descent on any energy function. The dynamics of the algorithm must instead be described using a set of energy functions, which can be very complicated for the multi-dimensional case.

In this paper we have proven analytically that, for the one-dimensional SOFM algorithm, the "shape" of the neighborhood function, in particular its "convexity" determines the presence or absence of non-ordered stationary states. Ordering time may vary by orders of magnitude depending on the type of neighborhood function chosen. Broad, convex neighborhood functions, such as broad Gaussians, lead to the shortest ordering times. Narrow Gaussian functions, step functions, and other concave functions all allow non-ordered stationary states which delay the convergence of the algorithm on average. Other, more subtle properties of the neighborhood function which we have not studied may also turn out to be useful for efficient multi-dimensional map formation. For example, Geszti (1990) has suggested that anisotropic neighborhood functions should reduce the ordering time of one- and multi-dimensional feature mappings.

In this paper we have shown that the Gaussian functions, which are the most commonly used neighborhood functions, give best results if the width of the Gaussian function is large enough to avoid the presence of metastable states. For the one-dimensional case the value of σ below which the first metastable state appears may be found from the relation (20). An optimal value of σ exists, which is slightly higher than this threshold value. It is difficult to give an equation for the optimal value of σ for a Gaussian function, since the two competing effects, due to the presence of metastable states and the contraction of weights, scale differently with the number of neurons. As a quick approximation, the value of σ which gives the optimal ordering time is of the order of the number of neurons. It is probably wise to employ similar guidelines in choosing neighborhood functions in multi-dimensional applications.

From our discussion of ordering time for the one-dimensional algorithm, we can understand the empirically observed fact that in practical applications with high-dimensional forms of the algorithm, ordered maps

may often be formed more quickly by employing a neighborhood function which begins with a large width which is slowly reduced. Starting with a broad neighborhood function allows rapid formation of an ordered map, due to the absence of metastable stationary states. After the ordered map has formed, the width of the neighborhood function may be reduced until the map expands to fill the input space more fully.

Acknowledgements. This research has been supported by the National Science Foundation (grant number NSF 90-15561) and by the National Institute of Health (grant number P41RR05969). Financial support to E. E. by the Beckman Institute and the University of Illinois, and of K. O. by the Boehringer-Ingelheim Fonds is gratefully acknowledged. The authors would like to thank H. Ritter for stimulating discussions.

Appendix A: derivation of the stationary state equation

To derive the stationary state Eq. (15), we set $V_x = 0$ using the definition of V_x in (13). Inserting

$$\hat{h}(x, y) \equiv 1 - \lambda \hat{g}(x, y)\,, \text{ and} \tag{21}$$

$$u_x = 1/2 + \lambda\theta_x + \lambda^2\phi_x \tag{22}$$

into (13) and keeping only terms up to order λ^2, we find

$$\begin{aligned}
0 &= \lambda[(\hat{g}(1, x) - \hat{g}(N, x))/4 - 2\theta_x] \\
&\quad + \lambda^2 \Bigg\{ \sum_{y=2}^{N-1} (\theta_{y+1} - \theta_{y-1}) \\
&\quad + (\theta_{y-1} + 2\theta_y + \theta_{y+1} - 4\theta_x)/4 \\
&\quad + [(\theta_1 + \theta_2 - 4\theta_x)(\theta_1 + \theta_2 - \hat{g}(1, x)) \\
&\quad + \hat{g}(1, x)(\theta_1 + \theta_2) - 4\phi_x]/4 \\
&\quad + [\hat{g}(N, x)\theta_x - (\theta_{N-1} + \theta_N)^2/4 \\
&\quad + \theta_x(\theta_N + \theta_{N-1}) - \phi_x] \Bigg\} \\
&\quad + \mathcal{O}(\lambda^3) \\
&= \lambda[(\hat{g}(1, x) - \hat{g}(N, x))/4 - 2\theta_x] \\
&\quad + \lambda^2[\theta_x(\hat{g}(1, x) + \hat{g}(N, x)) - 2\phi_x] + \mathcal{O}(\lambda^3)
\end{aligned}$$

So from the first-order term we get

$$\theta_x = (\hat{g}(1, x) - \hat{g}(N, x))/8\,, \tag{23}$$

and from the second-order term we get

$$\begin{aligned}
\phi_x &= \theta_x(\hat{g}(1, x) + \hat{g}(N, x))/2 \\
&= (\hat{g}(1, x)^2 - \hat{g}(N, x)^2)/16\,.
\end{aligned} \tag{24}$$

Inserting this into (22) gives the result (15). Note that the third order terms (omitted here) cannot be written in such a simple form.

Appendix B: proof that all stationary states are ordered for convex $g(r, s)$

From $u_1 < u_2 < \cdots < u_N$ and (15) we can conclude that the following relationship must hold for the values of $\hat{g}(x, y)$

$$\begin{aligned}
\hat{g}(1, 1) - \hat{g}(N, 1) < \hat{g}(1, 2) - \hat{g}(N, 2) \\
< \cdots < \hat{g}(1, N) - \hat{g}(N, N)
\end{aligned} \tag{25}$$

Now assume that $g(x, y)$ has the following properties: $\hat{g}(x, y) = \hat{g}(y, x) = G(|\mathscr{P}(x) - \mathscr{P}(y)|)$, and $G(|x|)$ is monotonically increasing from zero with $|x|$. Also assume that

$$\hat{g}(z, x) > \hat{g}(z, y) + \hat{g}(y, x)\,, \tag{26}$$

will hold if and only if neuron $w_{P(y)}$ is located between neurons $w_{P(x)}$ and $w_{P(z)}$ in the neural chain, i.e. if

$$|\mathscr{P}(z) - \mathscr{P}(x)| > |\mathscr{P}(z) - \mathscr{P}(y)|, |\mathscr{P}(y) - \mathscr{P}(x)|\,. \tag{27}$$

Condition (26) may also be written as $(1 + \hat{h}(z, x)) < (\hat{h}(x, y) + \hat{h}(y, x))$ which together with (27) is our definition of convexity (4).

From (25) we know that for any x, $(\hat{g}(1, 1) - \hat{g}(N, 1)) < (\hat{g}(1, x) - \hat{g}(N, x)) < (\hat{g}(1, N) - \hat{g}(N, N))$, but since $\hat{g}(1, 1) = \hat{g}(N, N) = 0$, we may write

$$\hat{g}(x, N) < \hat{g}(N, 1) + \hat{g}(1, x)\,, \text{ and} \tag{28}$$

$$\hat{g}(1, x) < \hat{g}(1, N) + \hat{g}(N, x)\,. \tag{29}$$

From (28) and (26) we known that $\mathscr{P}(1)$ is not located between $\mathscr{P}(N)$ and $\mathscr{P}(x)$, and from (29) and (26) we know that $\mathscr{P}(N)$ is not located between $\mathscr{P}(1)$ and $\mathscr{P}(x)$. Therefore we may conclude

$$\mathscr{P}(x) \text{ is located between } \mathscr{P}(1) \text{ and } \mathscr{P}(N), \text{ for all } 1 < x < N\,. \tag{30}$$

From (25) we know that for all $x' > x$,

$$\hat{g}(1, x) - \hat{g}(N, x) < \hat{g}(1, x') - \hat{g}(N, x')\,, \text{ or rather}$$

$$\hat{g}(1, x) + \hat{g}(N, x') < \hat{g}(1, x') + \hat{g}(N, x)\,. \tag{31}$$

From (30) we know that both $\mathscr{P}(x)$ and $\mathscr{P}(x')$ are located between $\mathscr{P}(1)$ and $\mathscr{P}(N)$. Now suppose that $\mathscr{P}(x')$ were between $\mathscr{P}(1)$ and $\mathscr{P}(x)$. Then it follows from the assumption (26) that

$$\hat{g}(1, x) > \hat{g}(1, x') + \hat{g}(x, x') \Rightarrow \hat{g}(1, x) > \hat{g}(1, x')\,, \text{ and}$$

$$\hat{g}(x', N) > \hat{g}(x, x') + \hat{g}(x, N) \Rightarrow \hat{g}(x', N) > \hat{g}(x, N)\,, \text{ thus}$$

$$\hat{g}(1, x) + \hat{g}(x', N) > \hat{g}(1, x') + \hat{g}(x, N)\,. \tag{32}$$

Byt this contradicts (31). Therefore our supposition that $\mathscr{P}(x')$ is located between $\mathscr{P}(1)$ and $\mathscr{P}(x)$ is inconsistent with the assumption that the neighborhood function be convex. We conclude that for a convex neighborhood function

$$\mathscr{P}(x') \text{ is located between } \mathscr{P}(x) \text{ and } \mathscr{P}(N), \quad \forall x < x' < N\,. \tag{33}$$

Conditions (30) and (33) together imply that either $\mathscr{P}(1) < \mathscr{P}(2) < \cdots < \mathscr{P}(N)$ or $\mathscr{P}(1) > \mathscr{P}(2) > \cdots > \mathscr{P}(N)$. Therefore the only stationary states for a convex neighborhood function are the two states where the weights are ordered in either ascending or descending order.

References

Erwin E, Obermayer K, Schulten K (1991) Convergence properties of self-organizing maps. In: Kohonen T et al. (eds) Artificial neural networks, vol I, North Holland, Amsterdam, pp 409–414

Erwin E, Obermayer K, Schulten K (1992) Self-organizing maps: ordering, convergence properties and energy functions. Biol Cybern (this issue)

Geszti T (1990) Physical models of neural networks. World Scientific, Singapore

Geszti T, Csabai I, Czakó F, Szakács T, Serneels R, Vattay G (1990) Dynamics of the Kohonen map. In: Statistical mechanics of neural networks: Proceedings, Sitges, Barcelona, Spain, Springer, Berlin Heidelberg New York, pp 341–349

Kohonen T (1982a) Analysis of a simple self-organizing process. Biol Cybern 44:135–140

Kohonen T (1982b) Self-organized formation of topologically correct feature maps. Biol Cybern 43:59–69

Kohonen T (1988) Self-organization and associative memory. Springer, New York Berlin Heidelberg

Lo ZP, Bavarian B (1991) On the rate of convergence in topology preserving neural networks. Biol Cybern 65:55–63

Ritter H, Schulten K (1986) On the stationary state of Kohonen's self-organizing sensory mapping. Biol Cybern 54:99–106

Ritter H, Martinetz T, Schulten K (1989) Topology conserving maps for learning visuomotor-coordination. Neural Networks 2:159–168