ABSTRACT

Title of Dissertation: One-Shot Multi-Winner Self-Organizing Maps

Reiner Schulz, Doctor of Philosophy, July 22, 2004

Dissertation directed by: Dr. James Reggia
Department of Computer Science

There exist two different approaches to self-organizing maps (SOMs). One approach, rooted in theoretical neuroscience, uses SOMs as computational models of biological cortex. The other approach, taken in computer science and engineering, views SOMs as tools suitable to perform, for example, data visualization and pattern classification tasks. While the first approach emphasizes fidelity to neurobiological data, the latter stresses computational efficiency and effectiveness.

In the research reported here, I developed and studied a class of SOMs that incorporates the multiple, simultaneous winner nodes implicit in many biologically-oriented SOMs, but determines the winners using the same efficient one-shot algorithm employed by computationally-oriented, single-winner SOMs. This was achieved by generalizing single-winner SOMs, using localized competitions. The resulting one-shot multi-winner SOM was found to support the formation of multiple adjacent, mirror-symmetric topographic maps. It constitutes the first computational model of

mirror-image map formation, and raises questions about the role of Hebbian-type synaptic changes in the formation of mirror-symmetric maps that are often observed in the sensory neocortex of many species, including humans. The model unexpectedly predicted the occasional occurrence of adjacent, rotationally symmetric maps. It is natural to speculate that such atypically oriented maps might contribute to abnormal cortical information processing in some neurodevelopmental disorders.

Traditional SOMs lack applicability to problems where the inputs are not single patterns, but temporal sequences of patterns. Several SOM extensions have been proposed as a remedy, but there is no standard for processing temporal sequences with SOMs. I focused on the task of learning unique spatial representations for non-trivial sets of temporal sequences. The one-shot multi-winner SOM extended by temporally-asymmetric Hebbian synapses proved effective when applied to this task. The learned representations retained information about sequence similarity. The feature maps that formed show that temporal sequence processing and map formation are not mutually exclusive. Since the sequence processing one-shot multi-winner SOM was trained with phonetic transcriptions of spoken words, the results can be related to the internalization of spoken words during language acquisition. A final redesign of the network and the subsequent multi-objective optimization of its parameters using a genetic algorithm produced a more effective system.

One-Shot Multi-Winner Self-Organizing Maps

by

Reiner Schulz

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
July 22, 2004

Advisory Committee:

Dr. David Jacobs
Dr. David Mount
Dr. Donald Perlis
Dr. David Poeppel
Dr. James Reggia, Chair/Advisor

ACKNOWLEDGEMENTS

This work was supported by NIH award NS35466.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The self-organizing map (SOM) is an artificial neural network whose main characteristic is the association of each node in its output layer with a physical position in an output space which is typically a plane. The output nodes are usually regularly spaced so that if one connects each output node with a straight line to its closest neighbors, the nodes give rise to, for example, a square (grid-like) or hexagonal tessellation in the plane (see Figure 1.1). In very general terms, the SOM learns, in an unsupervised fashion, to systematically map inputs from an arbitrary and potentially high-dimensional input space to patterns of activation over the output lattice, the discrete "surface" laid out by the output nodes. The activation patterns of most SOMs are very simple in a winner-takes-all sense: each pattern comprises exactly one output node that is maximally active while all other output nodes are inactive. In these cases, the output for a particular input is completely specified by the position of the active output node in the lattice so that for higher-dimensional inputs, the SOM's mapping can be viewed as a dimensionality-reducing operation.

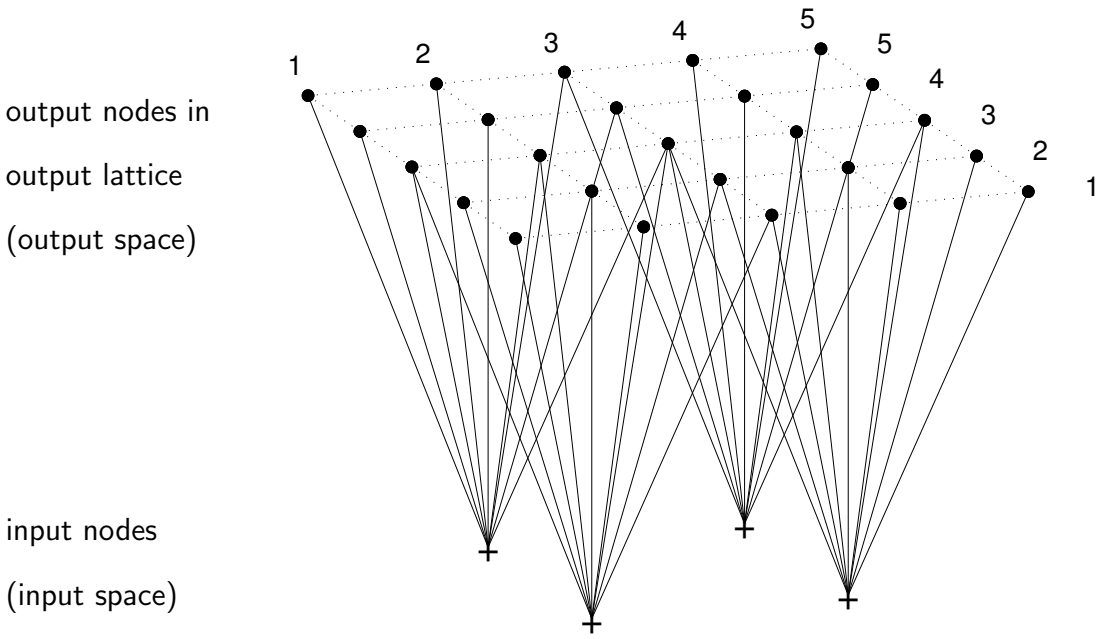


Figure 1.1: An example of the SOM network architecture. The SOM's 25 output nodes are located at the points of intersection in a five by five regular grid in a plane. The input layer consists of four nodes (marked with '+' signs). In this example, each input node sends connections to only a subset of all output nodes. Given a global competition for activation among the 25 output nodes with only a single winner for every input, the SOM performs a dimensionality-reducing operation by mapping each 4D input to a particular output node, that is, a 2D location in the plane.

1.1 Two Classes of SOMs

The specific properties of the mapping and the mechanism by which it is learned vary widely depending on what purpose a particular SOM serves, but essentially, two fundamentally different classes of SOMs currently exist. Both are popular research subjects, but since the study of each is motivated differently, SOM research has diverged and today forms two, for the most part disconnected, areas.

Originally, the SOM was conceived as a computational model of cortical information processing where the emphasis is on fidelity to neurobiological data. The earliest

work on SOMs was a model of feature map formation in the visual cortex of cats by von der Malsburg (1973), but this approach has been used since to model information processing in other cortical areas (e.g., Bednar and Miikkulainen (2000); Cho and Reggia (1994); Li (2002); Pearson et al. (1987); Reggia et al. (2001); Sutton et al. (1994)). The goal with this type of SOMs is to test and improve theories on how biological cortex represents and processes information. In this context, the SOM is seen as an explicit expression of a theory's assumptions, and by comparing the results of training the SOM (the consequences of the theory) with data that are reported in experimental studies of cerebral cortex, cognition, or behavior, the weaknesses and strengths of the underlying theory can be identified. I will refer to the class of SOMs that is the product of this line of biologically-oriented research as *iterative multi-winner SOMs*. They are iterative because, typically, complex systems of nonlinear differential equations that can only be solved via iterative simulations describe their activation dynamics. The equations implement a competition for activation among the output nodes of the network that typically results in activation patterns with more than one winner, which explains the second qualifying attribute of the name for this class of SOMs. The iterative nature of these SOMs is associated with a high computational complexity that limits their scalability. Large-scale, perhaps multi-modular computational models, albeit desirable, fast become very costly. The high complexity of iterative multi-winner SOMs also works against the goal of computational modeling to provide as simple an explanation of the observed experimental observations as possible.

The second, more recent and larger class of SOMs, which will be called *one-shot single-winner SOMs*, is an offspring of the iterative multi-winner SOM due to Kohonen (1982), and is much more widely used in computer science (Kohonen, 2001).

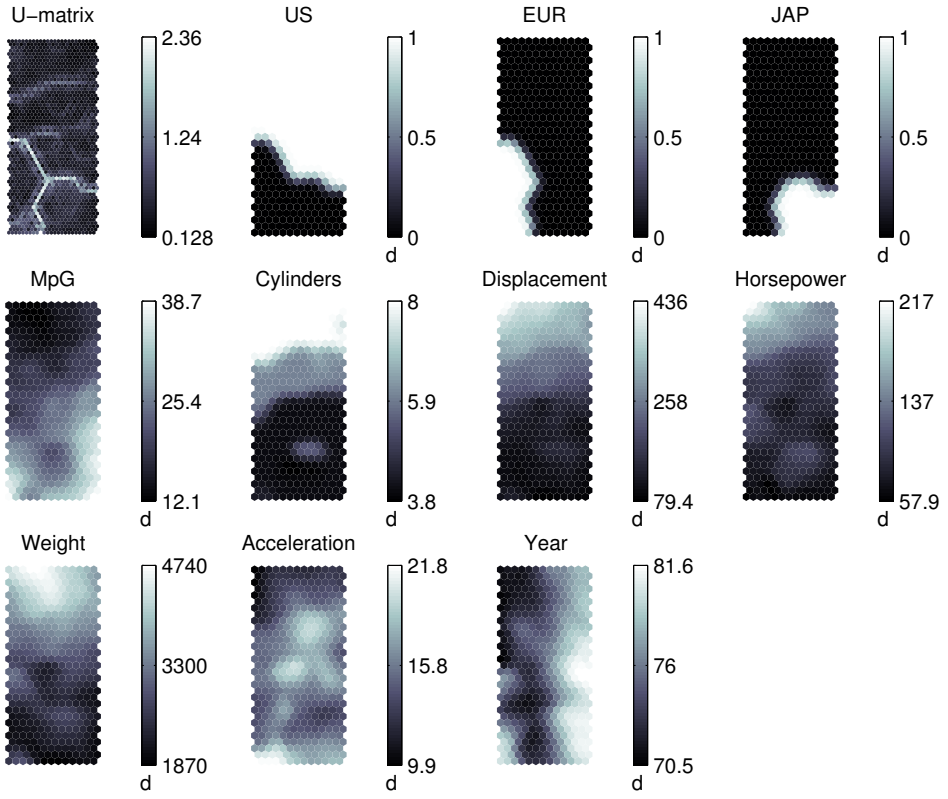


Figure 1.2: Maps resulting from training a one-shot single-winner SOM with the technical specifications of cars. Each car was represented as a vector of the form $\vec{x} =$ (made in USA, made in Europe, made in Japan, miles per gallon, number of cylinders, displacement, horsepower, weight, acceleration, year of make), the first three components being indicator variables (either 0 or 1). The similarity of adjacent output nodes in terms of their weight vectors is mapped in the top-left corner, revealing three tight (dark) clusters of cars that correspond to the three regions of origin (US at the top, Europe to the bottom-left, Japan to the bottom-right). The other maps show the nodes shaded according to the values of one of their weight vector components, making visible correlations between variables (= gradients in similar directions; e.g., displacement and horsepower) and independencies (= perpendicular gradients; e.g., origin and year of make).

When looked at from a technical perspective, a key simplification, albeit reducing neurobiological plausibility, renders the SOM a computationally efficient and practical tool in application domains like data visualization (e.g., Alhoniemi et al. (1999); Kaski et al. (1998a); Manduca (1996); Vesanto (1999)), feature detection (e.g., Morris et al. (1990); Munoz and Muruzabal (1998); Toivanen et al. (2003)), and pattern classification (e.g., Andrade et al. (1997); Callan et al. (1999); Takacs and Wechsler (1997)). This simplification is achieved by replacing the locally competitive activation dynamics with a global, non-iterative selection of a single winning node, which turns the iterative multi-winner SOM into a non-iterative SOM that I will refer to as the *one-shot single-winner SOM*. The improvement, from a technical point of view, is twofold. Computational efficiency is improved by orders of magnitude, and the representation learned by a one-shot single-winner SOM resembles the result of a principal curves analysis (Hastie and Stuetzle, 1989) or Sammon projection (Sammon, 1969) of the data cloud or manifold that the inputs to the network form in input space (see Yin (2003) for a review of non-linear data projection methods, including the SOM). Not only is this representation useful as a compact approximation of the input distribution that retains the most characteristic features of the input manifold, but it also, due to the representation's typically two-dimensional, map-like form, provides an intuitive visualization of dependencies between these features (see Figure 1.2). From a neurobiological point of view, the representation is implausible, because it encodes each input in a non-distributed and non-redundant manner, in stark contrast to the high memory capacity and fault-tolerance of biological cortex. This is not a concern in most studies of the one-shot single-winner SOM which focus on further reducing its computational cost, testing its applicability to specific practical problems, and formulating, in strict mathematical terms, the conditions that are necessary for

training to converge on an optimal representation.

1.2 One-Shot Multi-Winner SOMs

The existence of these two classes of SOMs naturally raises the question of whether the efficiency of one-shot single-winner SOMs can be easily combined with the distributed encoding that is inherent in iterative multi-winner SOMs to form a new class of SOMs that exhibits interesting and potentially useful properties. Specifically, it is of interest whether such a new class of SOMs supports map formation and, if so, what the properties of these maps are and how they relate to the maps that are formed by one-shot single-winner and iterative multi-winner SOMs.

Another issue is that, while each of the two main classes of SOMs described above has certain advantages and disadvantages in the context of specific applications, both classes of SOMs share a major limitation: they have been designed only to process time-invariant inputs. Without modifications, they are not capable of processing the relations between the elements of a temporal sequence of input patterns. Biological individuals are constantly facing the task of analyzing temporal sequences of sensory stimuli and adapting their behavior accordingly (e.g., tracking or avoiding an object whose relative motion is perceived visually, auditorily and/or through tactile senses). Similarly, the processing of temporal sequences is very important in modern technology. For example, some mobile robots must autonomously maintain their balance based on streams of feedback information delivered by sensors measuring accelerations, pressures, torques, etc. (Katic and Vukobratovic, 2003; Kun and Miller, 1999), while in some chemical plants, the flow of input materials requires continuous adjustments in order to maintain optimal operation conditions based on sequential feedback information (Bhat and McAvoy, 1990; Henson, 1998). These technical control prob-

lems typically require immediate reactions in response to the development of feedback signals over a period of time.

The brain, and in particular the cerebral cortex, constitutes a proof-of-existence of a solution for problems that involve temporal sequence processing (e.g., language, motor control, etc.). It is therefore an important next step to try to extend the SOM method, which has been shown to be both a somewhat faithful model of cortex and a useful technical tool, in a way that is once again patterned after cortex and provides the computational power necessary to process temporal sequences. This has been recognized by several past investigators who have proposed a variety of extensions that aim to make the one-shot single-winner SOM in particular applicable to temporal input sequences (e.g., Carpinteiro (1999); Chappell and Taylor (1993); Kangas (1990); Varsta et al. (1997)). However, no single uniformly applicable mechanism for processing temporal sequences with SOMs has been identified, partly due the many different specific tasks that fall into the category of temporal sequence processing.

In this dissertation, I develop and study a SOM methodology that efficiently and effectively combines elements of both the iterative multi-winner and the one-shot single-winner SOMs to form a new class of SOMs that I will call *one-shot multi-winner SOMs*. The one-shot multi-winner SOM is of low computational complexity and features a robust and coding-efficient distributed representation of the result that is computed for each input. With further extensions that are motivated by experimental findings in support of temporally asymmetric Hebbian learning at biological synapses (Bi and Poo, 2001, 1998; Markram et al., 1997; Zhang et al., 1998), the one-shot multi-winner SOM becomes capable of processing temporal sequences.

To gain an understanding of this novel SOM methodology, its properties, potential and limitations, one-shot multi-winner SOMs were developed and their properties

determined by pursuing the following specific aims:

1. Determine the properties of the one-shot multi-winner SOM, when input is time-invariant, and relate the results to the existing iterative multi-winner and one-shot single-winner SOM classes.

2. Determine conditions under which the one-shot multi-winner SOM forms multiple maps of the input space and examine the relationships of these maps to one another.

3. Study the one-shot multi-winner SOM when its task is to transform variable-length temporal input sequences into sequence-specific spatial representations. In particular, assess the memory capacity of the system and the nature of the learned representations.

4. Explore the performance limits of the temporal sequence processing one-shot multi-winner SOM by applying automatic optimization techniques to the problem of determining values for the parameters of the system that lead to better and ideally, near-optimal performance.

1.3 Summary of Results and Overview

In pursuing these aims, I obtained the following main results. The one-shot multi-winner SOM that I developed indeed supports map formation. In particular, networks with a sufficiently large output lattice formed multiple topographic maps of the input space. Moreover, maps that were adjacent in the output lattice were overwhelmingly mirror symmetric with respect to their shared boundary. This is consistent with experimental observations about the formation of multiple mirror image topographic maps in biological cortex, across different sensory modalities and species, including

humans (Drager, 1975; Engelien et al., 2002; Formisano et al., 2003; Merzenich et al., 1978; Newsome et al., 1986; Sur et al., 1982; Tiao and Blakemore, 1976). The one-shot multi-winner SOM thus constitutes the first computational model that produces multiple mirror image map formation similar to that seen in biological cortex. It does so purely on the basis of competitive Hebbian learning. These model results are intriguing as they relate to the ongoing debate about the extent to which biological topographic maps are learned or genetically determined (Cohen-Cory, 2002; Grove and Tomomi, 2003). The one-shot multi-winner SOM provides evidence suggesting activity-dependent synaptic changes may be more important in the formation of mirror image maps than is generally recognized. From the perspective of computer science, the one-shot multi-winner SOM constitutes a redundant and therefore more robust (less sensitive to damage and statistical noise) version of the already well-established one-shot single-winner SOM method (Kohonen, 2001).

When extended to the processing and representation of temporal input sequences, the one-shot multi-winner SOM proved capable of learning a unique spatially distributed representation for almost every distinct sequence in the relatively large training set of variable-length sequences. Since the training set comprised phonetic transcriptions of English nouns where each phoneme was represented as a high-dimensional phoneme feature vector, the network can be interpreted as a simplified model of unsupervised learning of word pronunciation. In addition, the one-shot multi-winner SOM maintained multiple map formation in terms of single phoneme features, which shows that the unique spatially distributed representation of sequential inputs and multiple map formation are not mutually exclusive phenomena. The results were consistent with what is known about biological cortex where similar inputs typically evoke similar distributed activation patterns (Haxby, 2001; Riesenhuber

and Poggio, 2002), and where temporal processing also takes place in areas that are occupied by topographic and/or feature maps (Ahissar and Arieli, 2001; Hoshi and Tanji, 2000; Sahyoun et al., 2004; Schrater et al., 2000). Certain design changes and the application of a multiobjective genetic optimization algorithm significantly improved the original and manually-optimized temporal sequence processing in the one-shot multi-winner SOM so that distinct sequences generally led to unique spatial representations, a transformation that can be exploited by subsequent stages in a larger temporal sequence processing system.

In short, the one-shot multi-winner SOM can explain a large number of phenomena that are associated with information processing in biological cortex. This fidelity to neurobiology combined with conceptual simplicity and computational efficiency should make the one-shot multi-winner SOM an attractive computational modeling tool that would allow for systematic studies of large scale multi-modular neural models, which formerly were associated with prohibitively large computational costs. On the other hand, the one-shot multi-winner SOM with its distributed representation of computation results may prove useful in application contexts which, in addition to computational efficiency, require a high degree of tolerance toward faults and/or statistical noise.

In the following, Chapter 2 provides background information on the self-organizing map, including the related phenomena that occur in biological cortex. It gives an overview of previous related work on the self-organizing map, with a special emphasis on self-organizing maps for sequence processing. Chapter 3 introduces the new class of one-shot multi-winner SOMs and investigates their properties when inputs are static in time, including the results about mirror-image map formation. Chapter 4 makes a transition to temporal sequence processing with the one-shot multi-winner

SOM. It explains the extensions to the one-shot multi-winner SOM that make it fit for temporal processing and studies its performance when applied to the task of learning unique spatial representations for sequential inputs. Chapter 5 is dedicated to the improvement of the temporal sequence processing by one-shot multi-winner SOMs by means of design changes and the subsequent multiobjective optimization of the network parameters using a genetic algorithm. Chapter 6 comprises a review and discussion of the research results, and an outlook on possible directions for future research on or using the one-shot multi-winner SOM.

Chapter 2

Background

As noted in the previous chapter, research on SOMs has historically been divided into two largely disconnected fields, each of which is concerned with its own research goals and uses its own type of SOM. Notation and nomenclature have diverged over the years. Both types of SOMs have in common that their architecture and dynamics are based, albeit to differing degrees, on biological cortex and that their behavior relates to cortical map formation. This chapter therefore starts out with an introduction to map formation in biological cortex, followed by a technical discussion of the shared features and differences of the two SOM classes. The last section reviews previous research efforts that relate to temporal sequence processing extensions to neural networks in general and SOMs in particular.

2.1 Maps in Biological Cortex

One of the most intriguing aspects of the SOM, whose architecture and low-level dynamics have been inspired by biological cortex, is the emerging high-level behavior of the system, specifically the formation of ordered maps of its inputs that can resemble those seen in biological cortex (Bauer, 1995; Kohonen, 1989; Martinetz et al., 1989; Obermayer et al., 1990, 1992a,b; Palakal et al., 1995; Ritter and Schulten, 1986),

suggesting that the SOM, despite being an extremely simplified model of cortex, can capture some fundamental principles of cortical self-organization. In biological cortex, map formation occurs in many primarily sensory areas, that is, cortical regions that are dedicated to the processing of incoming sensory information. In the neurosciences, a distinction is made between two types of cortical maps: topographic maps and computational or feature maps.

A *topographic map* constitutes a roughly topographically-correct point-to-point mapping of a two-dimensional sensory surface onto a continuous surface area of cortex. The stimulation of a point on the sensory surface activates a corresponding location on the cortical surface so that the relative distances between points on the sensory surface are roughly preserved in the corresponding distances between activated locations of cortex. Examples for this type of map are the somatotopic (Dykes and Ruest, 1984; Killackey et al., 1995), retinotopic and tonotopic maps that are located in primary somatosensory, visual and auditory cortex, respectively. These maps are illustrated in Figures 2.1, 2.2 and 2.3.

The other class of cortical maps, computational or *feature maps*, are systematically ordered mappings of sensory stimuli onto the surface of cortex where the order is with respect to a particular feature of the sensory stimuli other than their location on the sensory surface. In the primary visual cortex, for example, there exist ordered mappings according to ocular dominance and orientation sensitivity (Hubel and Wiesel, 1962, 1968, 1979). In the former case, the map, when visualized via microelectrode readings or cortical staining, takes the form of alternating bands where each band consists of cortical columns that are preferentially activated by inputs from the same eye. Cortical columns in visual cortex also exhibit a preference with respect to the orientation of line segments within visual stimuli. Each cortical column tends

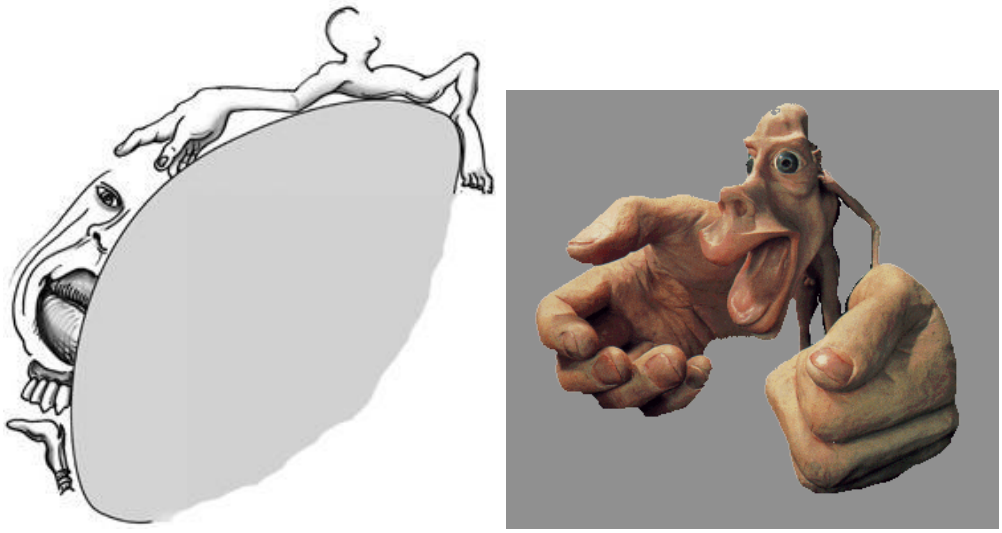


Figure 2.1: The left drawing outlines one of the cortical hemispheres cut vertically and from side to side (gray area) so that the curved (black) line of intersection with the cortical surface corresponds to the midline of the somatosensory area which extends like a band from medial to lateral across the cortical surface. This area hosts a roughly topographic map of the (here human) body surface whose orientation is indicated by the sketch of a human form following the outline of the hemisphere. The preservation of the body's topology is not perfect: mouth and face are represented laterally while the remaining body is represented medially so that face and hand are adjacent in cortex. The representations of the lips and hands occupy a disproportionately large cortical area which is an example of the magnification effect: regions of sensory surface with a relatively higher density of sensors and/or frequency of stimulation tend to be represented in more detail over relatively more cortical surface area. This is illustrated further by the human figure to the right, an area-proportional reverse projection of the cortical somatotopic representation. After Penfield and Rasmussen (1950). From Strobel (2000).

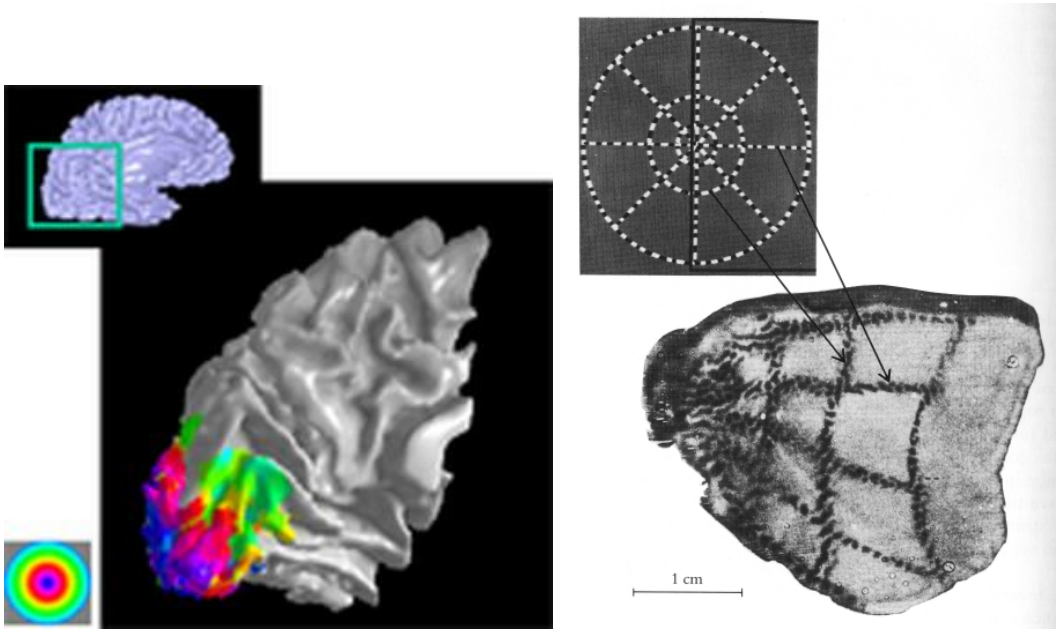


Figure 2.2: Both illustrations show that cortex, specifically the primary visual area which occupies the posterior lobes of both hemispheres (boxed-in area at the top of the picture to the left), contains a roughly topographic representation of the retinal surface and thus, the visual field. The picture to the left shows the visual field partitioned into differently colored concentric rings (bottom-left). The order among the colors is mostly preserved in the cortical representation (bottom-right; computed from fMRI data), indicating a roughly topographic cortical map of the visual field. In the picture to the right (from Tootell et al. (1982)), concentric circular and straight lines partition the visual field (top). Along those lines, point-like visual stimuli were applied. The ensuing activity at the responding cortical locations stained the cortical tissue in a topographically correct manner (bottom), recreating, on the surface of the visual cortex, the line pattern along which the visual field had been partitioned (arrows indicate corresponding line segments).

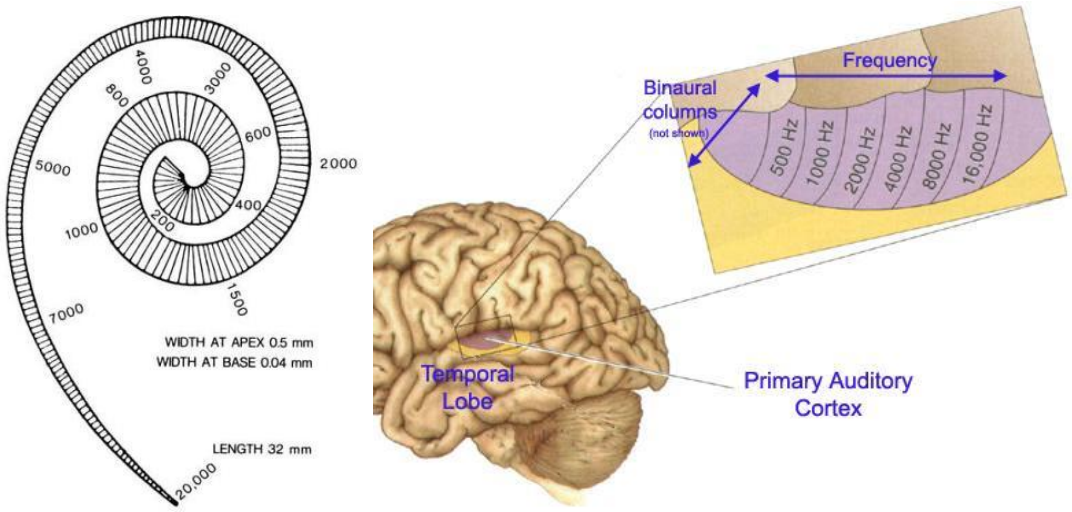


Figure 2.3: The left picture shows the spiral-shaped basilar membrane, the essentially one-dimensional auditory sensory surface (from Stuhlman (1952)). It is covered with mechanical sensors (hair cells) that are activated by sound waves. The location of a sensor determines which pure tone frequency activates it the most. Sensors at the center of the spiral preferably respond to the lowest perceptible frequencies, while toward the tip of the outer spiral arm, sensors are activated the most by tones of progressively higher frequency. This order is preserved in the frequency-sensitive bands of the basilar membrane's map representation in primary auditory cortex (Heschl's gyrus) which is shown to the right.

to be activated the most by a particular orientation stimulus, and columns that are relatively close in cortex are typically sensitive to similar orientations which gives rise to an overall very smooth and continuous mapping of orientation sensitivities onto the surface of cortex.

From a theoretical point of view, the distinction between topographic maps and feature or computational maps is largely artificial. The location of a stimulus on a

sensory surface is essentially just another feature of the stimulus. From this perspective, a topographic map becomes a consequence of the basic principle that seems to unify all cortical maps: nearby cortical locations represent stimuli that are similar with respect to a particular feature which, in the case of topographic maps, is the location feature. However, there are exceptions to this rule like, for example, the discontinuity in cortical somatotopic maps where the representation of the face is adjacent to the representation of the hand (see Figure 2.1, and Dykes and Ruest (1984)), or the pinwheel patterns in orientation sensitivity maps where adjacent cortical columns can be sensitive to very dissimilar, that is, perpendicular line orientations (Ohki et al., 2000).

Two more observations are often made with respect to in particular topographic cortical maps. First, the cortical area that the representation of a particular region of a sensory surface occupies is not strictly proportional to the region's surface area. Relatively more often stimulated and/or more sensitive regions of a sensory surface typically occupy a disproportionately large area of cortex (Azzopardi and Cowey, 1993; Creutzfeldt, 1978; Dykes and Ruest, 1984; Sereno et al., 1995). This is called the magnification effect which is very apparent in, for example, somatotopic maps where, for example in primates, the lips and hands are area-wise overrepresented (see Figure 2.1 and Dykes and Ruest (1984)). The other observation is the existence of multiple topographic maps of the same sensory surface, often in neighboring cortical areas and oriented so that adjacent maps are mirror symmetric with respect to their common boundary (e.g., Engelien et al. (2002); Formisano et al. (2003); Merzenich et al. (1978); Sereno et al. (1995); Sur et al. (1982)).

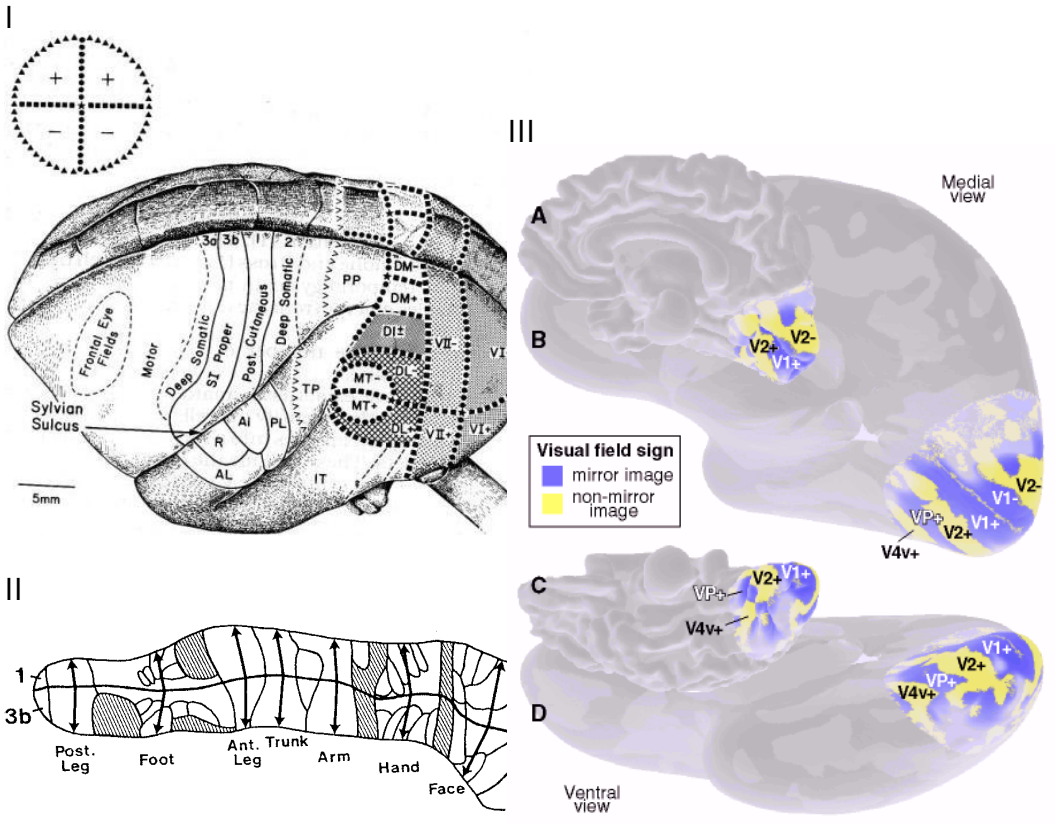


Figure 2.4: **I** Lateral view of the left cortical hemisphere with various cortical areas delineated, in particular the somatosensory areas 3a, 3b, 1, and 2. From Allman (1981). **II** Cortical somatotopic maps in the somatosensory region SI, composed of the adjoining areas 3b and 1, of the squirrel monkey, based on multi-unit microelectrode readings. The arrow pairs indicate that each of the two areas is home to a complete nearly topographic map representation of the body surface that is roughly mirror symmetric to the map in the adjacent area, where the axis of reflection corresponds to the border between areas 3b and 1. From Sur et al. (1982). **III** In primary and secondary human visual cortex, several areas (VI, VII, VP, and V4v) are home to topographic maps of the visual field. In addition, adjacent maps are again mirror symmetric to each other with respect to visual field topography. From Sereno et al. (1995).

2.2 Self-Organizing Maps

The common denominator of the two SOM types is that they are both neural network methods for the unsupervised acquisition of a mapping from an often high-dimensional input vector space into a space of patterns over a discrete, usually two-dimensional surface which consists of the output nodes of the neural network arranged in a regular lattice, e.g., the rectangular grid in Figure 1.1. Each node in the output lattice is not necessarily fully connected to the nodes that make up the input layer. Each connection an output node receives from the input layer carries a weight, so every output node is associated with a weight vector located in the input space. These weight vectors are often initialized randomly. During training, which is solely based on the repeated input of vectors from a representative sample of the input space, the weight vectors are adjusted slowly in response to each input according to an unsupervised learning rule. Hebbian, e.g., in Miikkulainen (1991), or competitive, e.g., in Kohonen (1982), learning methods are most common.

An intuition of the cumulative effect of training is easier to convey if one assumes each output node to be fully connected to the input layer, i.e., all weight vectors are located in the complete input space. Training essentially orders the initially random weight vectors, and thus the output nodes, such that they form a two-dimensional map which is characteristic of the distribution of vectors in the input space as represented by the training samples. For map examples, see Figures 1.2 and 2.5.

The ordering that emerges in a SOM during learning is a combination of statistical properties of the input distribution and local interactions between the output nodes. While two output nodes only interact directly if the distance between them is relatively small, overall the order tends to be 'smooth' across the entire lattice because any two output nodes that are immediate neighbors tend to have similar weight vectors.

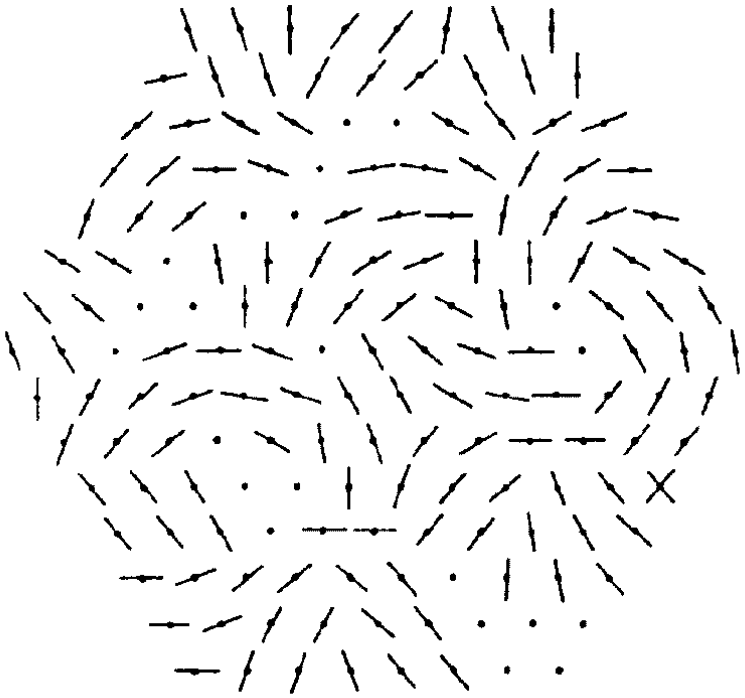


Figure 2.5: The hexagonal lattice of output nodes used in von der Malsburg's model of a small patch of visual cortex (area V1). The network was trained with line segments in the input space ("retina") having nine different orientations. The first orientation is a vertical line segment. The following eight orientations roughly correspond to eight consecutive clockwise rotations of the line segment around its center by 40 degrees. In the figure, each node is labeled with a line segment in the orientation that is the median of an interval of consecutive orientations for which the activation level of the node exceeds a threshold. The activation level of a node without a line segment does not exceed the threshold for any of the nine different orientations. A node with two line segments responds to two separate intervals of consecutive orientations above threshold. From von der Malsburg (1973).

Irregularities in the input distribution, in particular dense clusters, do however cause this smoothness to be disturbed at times. In such a case one observes a subdivision of the output lattice into internally relatively smooth areas that are sharply separated along their mutual borders by an abrupt change in the value of weight vectors as one crosses the border from one area into another. Each area is usually attributable to a particular cluster of patterns in the input training data (e.g., the clustering of cars according to origin in Figure 1.2). Within an area, the cluster is broken down further, and since the size of an area is correlated with the density of the cluster, denser clusters tend to be resolved in more detail and preferably along their most variable dimensions.

What has been said so far applies to both classes of SOMs. There are however significant differences between the two. These differences are summarized in Table 2.1, and discussed in detail in the next two sections.

2.3 Iterative Multi-Winner SOMs (Malsburg Maps)

Christoph von der Malsburg was probably the first to simulate the basic properties of SOMs in his neural model of the self-organized formation of maps in the visual cortex of cats and monkeys (von der Malsburg, 1973; von der Malsburg and Willshaw, 1976), which is composed of orientation-sensitive cortical columns (Hubel and Wiesel, 1962, 1963, 1968, 1979). He conceived the model SOM as a neurobiologically grounded computational model of cortex. The model reproduced the characteristic (mostly smooth and continuous) two-dimensional order with respect to orientation sensitivity that had been observed among the cortical columns of visual cortex. Thus, the model supports the theory that this phenomenon arises from unsupervised Hebbian-type learning.

Table 2.1: Typical Features of the Two Types of Self-Organizing Maps

SOM type →	Iterative Multi-Winner	One-Shot Single-Winner
seminal work	von der Malsburg, 1973	Kohonen, 1982
primary applications	neuroscience: usually modeling neocortex	computer science: data visualization, feature detection, pattern classification etc.
input-to-output connectivity	divergent, but localized	full
intra-lattice connectivity	lateral (excite immediate neighbors, inhibit more distant ones)	none (implicit neighborhoods)
activation dynamics	multiple winners: non-linear differential equations	single global winner: node most activated by the input
learning rule	Hebbian/competitive	Hebbian/competitive
computational cost	high	low
memory capacity	high	low
examples	Bednar and Miikkulainen (2000); Cho and Reggia (1994); Li (2002); Pearson et al. (1987); Reggia et al. (2001); Sutton et al. (1994); von der Malsburg (1973)	Callan et al. (1999); Kaski et al. (1998a); Kohonen (1982); Kokkonen and Torkkola (1990); Principe et al. (1998)

In addition to the two-dimensional structure of cortex, von der Malsburg's model incorporates another feature of cortex: lateral connections of limited range between cortical columns. These connections are set up as to facilitate localized competitions for activation among the nodes of a circumscribed neighborhood. When an input vector is presented to the network, the initial level of activation of an output node is roughly proportional to the similarity between its weight vector and the input vector. Some nodes are more activated by the input than others. The initial activation level of a node evolves over time according to a non-linear differential equation which takes into account the activation levels of connected nodes as well as the strength and nature (excitatory versus inhibitory) of these connections. This activation dynamics amplifies the activation levels of nodes that initially are very active relative to other nodes in their neighborhood and suppresses the activation of initially less active nodes. Overall this causes the initially diffuse distribution of activation across the lattice to evolve into a pattern composed of focused peaks of activation ('Mexican-hat' patterns of activation) that are sometimes centered at initially locally maximally active output nodes. This behavior of the model is consistent with electrophysiological measurements of the activation pattern over an area of cortex in response to external stimulation (Donoghue et al., 1992; Georgopoulos et al., 1988).

Learning in von der Malsburg's model is based on the final focused activation pattern. The weight vector of an output node is made more similar to the input vector to a degree that is proportional to the node's activation level. Based on these basic principles a map like that in Figure 2.5 emerges which resembles orientation-sensitivity maps observed in visual cortex when measuring the sensitivity of cortical columns to light bars of different orientations (Hubel and Wiesel, 1962, 1963, 1968, 1979).

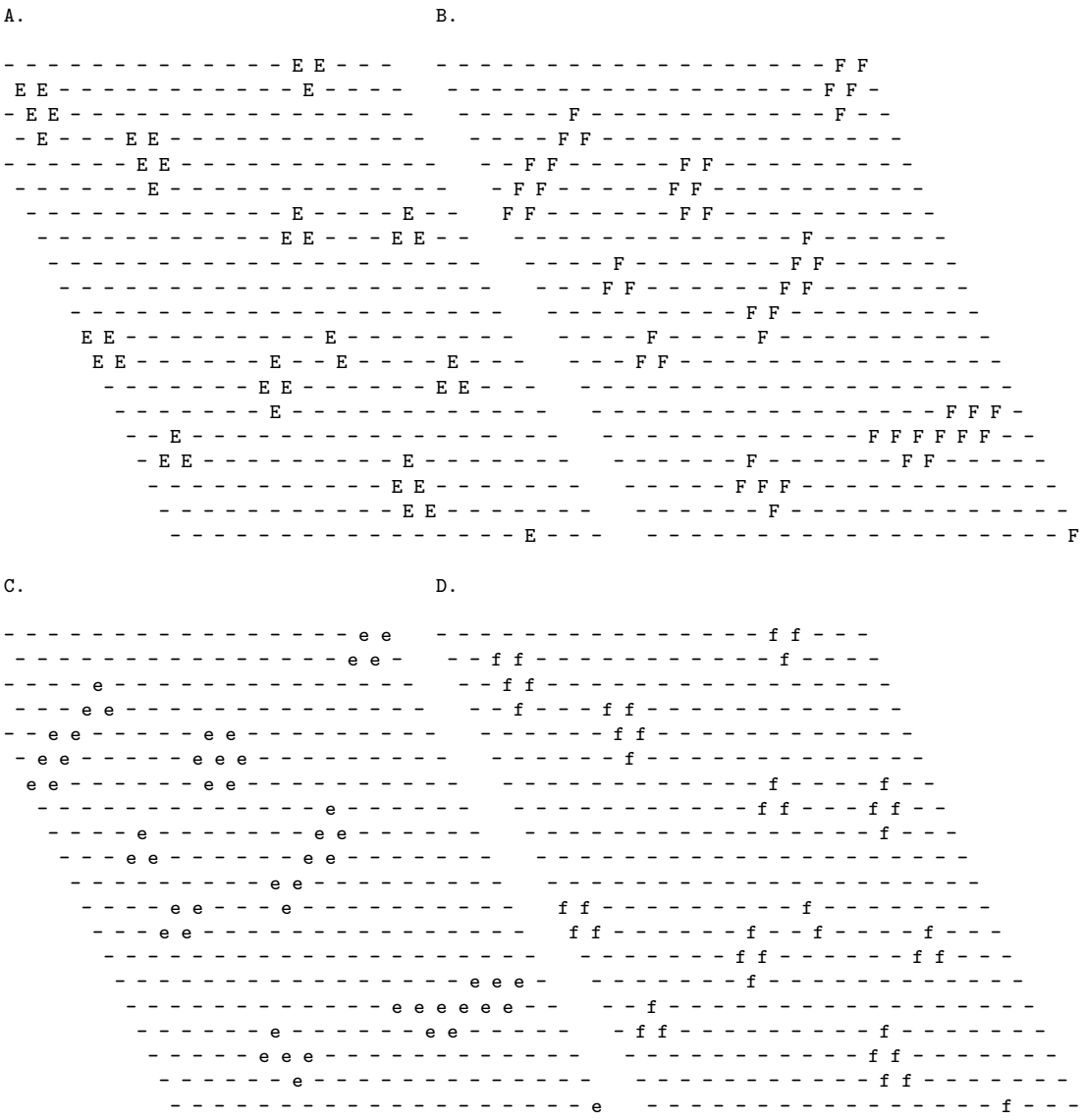


Figure 2.6: The tuning of output nodes in the proprioceptive cortex model of Chen and Reggia (1996). **A** elements tuned to the lengthening of the upper arm extensor, **B** elements tuned to the lengthening of the upper arm flexor, **C** elements tuned to tension in the upper arm extensor, **D** elements tuned to tension in the upper arm flexor.

Many other iterative multi-winner SOM-based neural models have been built since then. For example, Chen and Reggia (1996) studied a model composed of two SOMs interacting via the environment, which in this case is the simulated position of an arm. The first SOM modeled the primary motor cortex and sent muscle contraction signals to the arm causing it to change its position. The simulated arm translated its position into sensory information about the amount of tension in the arm muscles. These sensations were sent back to the second SOM which modeled the proprioceptive sensory cortex. Figure 2.6 shows some of the maps that emerged in the proprioceptive sensory SOM which eventually, i.e., after unsupervised training, reflected the correlations and anti-correlations between components of the sensory input that were consequences of the physical constraints built into the simulated arm. For example, the co-occurrence of a lengthened upper arm extensor and a tense upper arm flexor (the extensor's antagonist) was a consequence of the simulated arm positions, and after training, this was truthfully reflected in the almost exact alignment of maps (A) and (D). In contrast to that, maps (A) and (B) were complementary which was consistent with the fact that upper arm extensor and flexor could not be lengthened at the same time.

What most iterative multi-winner SOMs, including those described above, have in common is a computationally expensive implementation of the competitive activation dynamics by means of iteratively simulating a system of non-linear differential equations (Bednar and Miikkulainen, 2000; Cho and Reggia, 1994; Grajski and Merzenich, 1990; Li, 2002; Sirosh and Miikkulainen, 1992; Reggia et al., 2001; Sutton et al., 1994). This hinders efforts to investigate large scale neural models composed of several interacting SOMs. Therefore the issue arises as to whether it is possible to create computationally less costly, but nevertheless powerful neural models where the

simulation of differential equations is replaced by an instantaneous approximation of the simulation's outcome.

A different, still-debated issue is the role of short-distance lateral connections in cortex. A common assumption is that they are responsible for the competitive activation dynamics of cortex. Based on this assumption, the weights of these connections in von der Malsburg's and many other neural models are prescribed such that they cause a competitive dynamics. This view has gained support from studies which evolve or learn the lateral connection weights necessary to evoke this behavior (Sirosh and Miikkulainen, 1992; Ayers and Reggia, 2001). However, it has been shown that a competitive *distribution* of activation (Reggia, 1989) from the input layer is capable of producing the same effect in SOMs without lateral connections (Cho and Reggia, 1994; Sutton et al., 1994; Reggia et al., 1992). This prompts the question of what other role short-distance lateral connections in cortex might play. One hypothesis is that they enable cortex to process information with a temporal dimension by storing spatially and temporally local correlations between activation patterns that are distributed across cortex.

2.4 One-Shot Single-Winner SOMs (Kohonen Maps)

That the SOM can be formulated as an effective practical information processing tool in computer science was first suggested by Teuvo Kohonen (Kohonen, 1981, 1982). For the purpose of processing and visualizing digitized speech signals he constructed a significantly more efficient and better performing version of von der Malsburg's iterative multi-winner SOM.

In a SOM of the Kohonen-type (one-shot, single-winner) the competitive dynamics is cut short by simply declaring a single, initially maximally active output node the

winner of the global competition for activation. This not only decreases the computational cost of training by potentially orders of magnitude, but it also tends to eliminate redundancy in the final map representation of the input vector distribution. This redundancy is inherent in iterative multi-winner SOMs. For example, in Figure 2.5 there exist multiple nodes which respond most to vertical line segments and are far apart from one another. From a technical perspective this is undesirable. For data processing applications the virtue of the one-shot single-winner SOM lies in its ability to learn efficiently and in an unsupervised fashion to reduce the dimensionality of the usually high-dimensional input vectors so that the topology of the distribution of vectors in the input space is roughly preserved.

The meaning of the phrase 'topology-preserving' in connection with the SOM has not yet been unambiguously characterized in mathematical terms (Bauer and Pawelzik, 1992; Göppert and Rosenstiel, 1993; Kiviluoto, 1996; Ritter and Schulten, 1988; Villmann, 1999; Villmann et al., 1997). There are, however, many intuitive examples like that in Figure 2.7 which show that the one-shot single-winner SOM performs a form of distortion-minimizing projection (akin to, e.g., the projection of Sammon (1969)) of a typically high-dimensional input vector distribution onto the discrete, usually two-dimensional surface that is the SOM's output lattice. The example also demonstrates that the topology-preserving property of the one-shot single-winner SOM is volatile with respect to changes in the training parameters. That in general it is a hard problem for a two-dimensional one-shot single-winner SOM to preserve the topology of some input vector distributions, even if the input space is only three-dimensional, is illustrated in Figure 2.8. Despite these difficulties the one-shot single-winner SOM has become a popular data processing tool with applications in domains as diverse as computer vision (Deschenes and Noonan, 1995;

Manduca, 1996; Morris et al., 1990; Takacs and Wechsler, 1997; Toivanen et al., 2003), robotics (Cervera and del Pobil, 1999; Faldella et al., 1997; Heikkonen and Koikkalainen, 1997), signal/speech processing (Callan et al., 1999; Kangas, 1991; Kohonen et al., 1984), economics (Deboeck and Kohonen, 1998; Kaski et al., 1998a), and bioinformatics (Andrade et al., 1997; Ferrán and Ferrara, 1991; Hanke and Reich, 1996; Schuchhardt, 1996). Overall, the bibliography on, for the most part, the one-shot single-winner SOM consists of more than 5300 entries representing 30 years of research (Kaski et al., 1998b; Oja et al., 2003).

Figure 2.7 (next page): Each of the plots displays both the SOM and the input vectors used for its training in the three-dimensional input space. The training vectors point to locations on the surface of the unit sphere which are marked by a '+'. These points are arranged in a slightly skewed grid. Each output node of the SOM is plotted at the position on the unit sphere that corresponds to its weight vector. Two output nodes are connected by a line if they are immediate neighbors in the 2D output lattice. The topmost plot shows the SOM prior to training when each node's weight vector points to a random location on the unit sphere. The SOM was trained twice independently for two only slightly different parameter settings. The plots in the center show the SOM after 100 epochs, the plots at the bottom after 500 epochs of training. The SOM to the left eventually almost perfectly captures the topology of the input vector distribution, whereas the SOM on the right becomes intertwined early (develops a "fold"), and further training does not correct this defect.

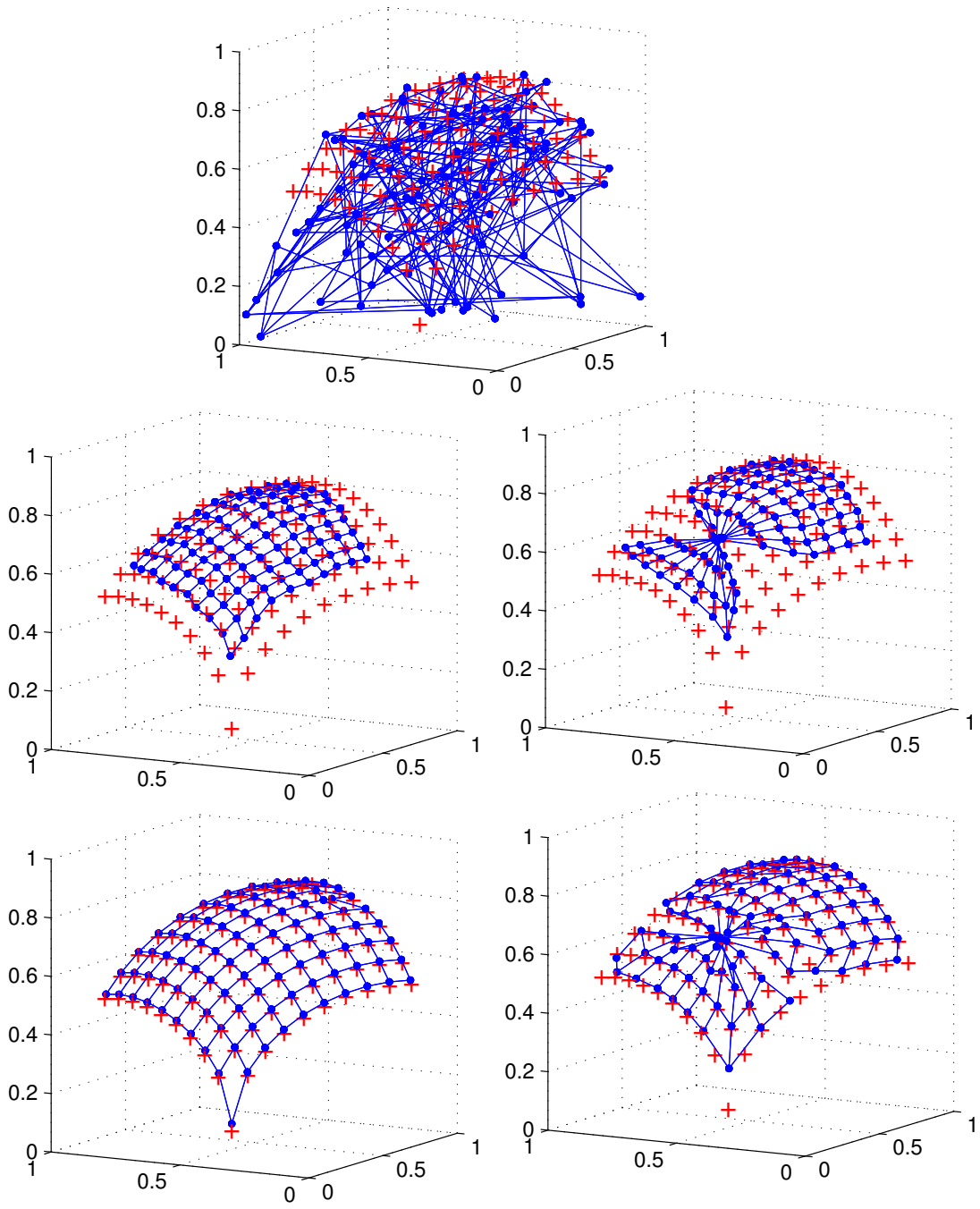


Figure 2.7: Caption on previous page

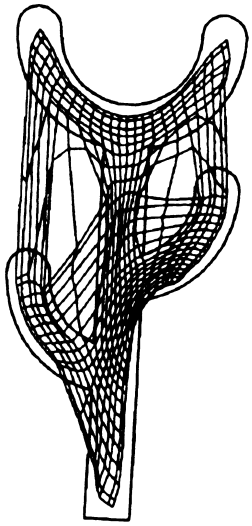


Figure 2.8: The two-dimensional lattice of a trained one-shot single-winner SOM as it is located in the three-dimensional input space according to the values of the three-dimensional weight vectors. The network was trained with vectors uniformly distributed across the complex input manifold, that is, the surface of the cactus. For the SOM's 2D output lattice, it is impossible to approximately cover the cactus' surface and create a topology-preserving projection of it onto the 2D lattice: there are output nodes that are adjacent in the lattice (connected by a line) but are far apart with respect to the cactus' surface. Taken from Kohonen (1989).

2.5 SOMs for Sequence Processing

Work aimed at extending the applicability of neural networks to temporal sequences has resulted in a wide range of proposals on how to achieve this. Within the domain of supervised learning, a simple but effective approach is the incorporation of context information in a standard feedforward multi-layer perceptron: at each time step, the original input vector and a context vector, which is roughly composed of the activation

levels of hidden and/or output units at the previous time step, are concatenated and together form the actual input to the network (Jordan, 1986; Elman, 1990). The network is trained via an unmodified standard error-backpropagation learning rule. In contrast to that, it is this rule which needs to be generalized in order to process temporal sequences with *arbitrary* recurrent error-backpropagation networks. This was done first for networks with a regular activation dynamics (Pineda, 1987), and the result was later generalized to networks with a competitive activation dynamics (Cho and Reggia, 1993). The derivations in principle used techniques (steepest descent) similar to those employed in the crafting of the standard error-backpropagation rule (Rumelhart et al., 1986), but taking into account arbitrary recursive connections made this a challenging task.

The temporal dimension of time-varying inputs can be captured independently of the training method as well. The 'leaky-integrator' model of a neuron (Cohen and Grossberg, 1983; Hopfield, 1984), which, via a membrane time constant, takes into account that a neuron is a capacitor, has been used in neural networks to give them temporal processing power (Chappell and Taylor (1993); Euliano and Principe (1999); Lambrinos et al. (1995); Mozer (1989); for an in-depth review of temporal neuron models see Gerstner (1995)). A similar approach is to use a separate 'short term memory' (STM) to enrich the original time-varying input with information about its temporal history. Each of the three most popular STMs (Mozer, 1993), i.e., the tapped delay line, the exponential trace memory and the gamma memory, is equivalent to the addition of nodes and/or constantly weighted connections to the input layer of a neural network.

Like many other neural network architectures, past SOM models have usually been designed to process a single time-invariant input vector at a time, but not a

sequence of time-varying vectors. This shortcoming has been addressed by only a small fraction of the SOM literature. The use of STMs and leaky-integrator neurons as the SOM's output nodes is widespread (Carpinteiro, 1999; Chappell and Taylor, 1993; Euliano and Principe, 1999; Koskela et al., 1998; Lambrinos et al., 1995). The approach of Carpinteiro (1999) is also an example of a network composed of two layered SOMs where the top SOM receives the activation pattern of the bottom SOM as input. The technique of stacking SOMs (often associated with the introduction of a time delay) to process temporal sequences has been used by other authors as well (Kangas, 1990; Morasso, 1991). In addition to using leaky-integrator neurons, the one-shot single-winner SOM in Euliano and Principe (1999) is made sensitive to the temporal dimension of the input via wavefronts of activation which spread and attenuate over time. In essence, the activation of an output node at a particular time step increases the chances of its immediate neighbors to win the competition for learning at subsequent time steps. This has the effect that the output nodes become ordered not just according to the similarity between the input vectors but also with respect to the temporal order in which they are presented to the SOM.

A different approach is taken in Kohonen (1991) where two sets of input weights exist: pattern weights and context weights. The input to the network consists of a pattern (a small sliding window) and the context (a larger sliding window) in which it appears within the input sequence. The context of a pattern pre-activates a subset of output nodes, among which the final winner node is determined by the pattern part of the input. An idea related to this "hypermap" for the spatial representation of temporal sequences, is presented in Kangas (1992). Given a one-shot single-winner SOM, the vector preceding the current input vector in a temporal sequence pre-activates a circumscribed neighborhood of output nodes. Only output nodes

within that region participate in the following competition. The winner then becomes the center of the next pre-activated neighborhood of nodes. The location of the winner node hence always depends on the current and all previous vectors of the input sequence, i.e., the location encodes both certain features of the current vector and the history of past vectors from the input sequence. The trajectory (on the 2D map) of winner nodes that unfolds for a sequence is its spatial representation. Having multiple winner nodes on the same 2D map is identified as a possible topic for future research. All the aforementioned efforts have been successful in training an essentially unmodified one-shot single-winner SOM to visualize and categorize/cluster (but not recall) temporal sequences by representing each sequence via a single output node. The problem of time series prediction has been addressed in Principe et al. (1998) via a one-shot single-winner SOM where each output node corresponds to a local linear model of the time series. The node/model best matching a fixed number of successive values from the series is then used to predict the next value of the series.

One of the rare examples of a SOM (iterative, essentially single-winner and with full lateral intra-map connectivity) for memorization and recall of temporal sequences composed of two-dimensional vectors is reported in Kopecz (1995). Storage and recall are greatly limited in that the same vector may not occur multiple times within the same sequence and that two different sequences may not share the same vector. These restrictions together with the full lateral connectivity limit the network's usability for many applications of interest and make it an implausible model for the representation of sequences in biological cortex.

None of the above research efforts has been undertaken with the explicit goal of training a SOM to find a unique, spatially distributed representation for each sequence in a large and unrestricted set of variable-length temporal sequences. Most efforts

have focused on extending the existing one-shot single-winner SOM methodology to temporal sequences which implies that each sequence is represented in a non-distributed fashion by a single output node. Because of this inefficient coding scheme, the one-shot single-winner SOM is a poor choice to try to achieve the above goal. By allowing multiple winners to exist (one-shot multi-winner SOM) the representations become distributed which promises to increase representation capacity, i.e., it should become easier to uniquely represent large sets of inputs. This idea, investigated later in this dissertation, has not been pursued before. Another novel aspect of the one-shot multi-winner SOM is the use of local, i.e., short-range, lateral intra-lattice connections which learn to guide the flow of activation over time such that the spatial representation for a temporal sequence is likely to be unique. Spreading wavefronts of activation (Euliano and Principe, 1999) and pre-activated neighborhoods (Kangas, 1992; Kohonen, 1991) can have a similar effect, but these mechanisms are static (they do not undergo training), and hence are indifferent to characteristic temporal properties of the training data.

Chapter 3

One-Shot Multi-Winner Self-Organizing Maps

This chapter introduces the one-shot multi-winner SOM, forms hypotheses on how the computational properties of this new class of SOM relate to map formation in biological cortex, and discusses the results of the experiments that were conducted to test these specific hypotheses and shed light on the computational properties of the one-shot multi-winner SOM in general. The first section provides a detailed description of the one-shot multi-winner SOM's architecture and dynamics. This is followed by a review of the literature on map formation in biological cortex which spawns the central hypotheses about how the one-shot multi-winner SOM might relate to the experimentally observed biological phenomena. Prior to the subsequent presentation of the computational simulation results that were obtained with the one-shot multi-winner SOM, a brief section on quantitative measures of map formation provides some necessary additional technical background. The final section of this chapter argues that the simulation results suggest that the one-shot multi-winner SOM's behavior make it an interesting computational model of cortical topographic map formation, especially of the mirror-image relationships that occur between biological neocortical maps. Alternative views and certain assumptions and consequences of the model are also discussed.

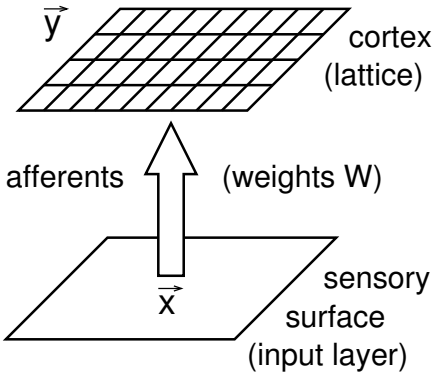


Figure 3.1: Architecture of the model cortical region used in this chapter. An input pattern $\vec{x}$ encoding the stimulation of a point on the sensory surface is modulated by afferent synaptic strengths W to produce an activation pattern $\vec{y}$ over a lattice of output nodes representing the neocortical surface. During Hebbian learning of W, a map of the input patterns and hence, assuming a suitable encoding is used, of the underlying sensory surface forms in the cortical region.

3.1 The Basic Model Architecture and Dynamics

The basic architecture of the multi-winner SOM, illustrated in Figure 3.1, is identical to that of a standard SOM. The output nodes are arranged in a regular, rectangular lattice of R rows by C columns. The *distance between two output nodes* i and i' at positions (r, c) and (r', c') in the lattice is measured using the box-distance metric, that is, $d_{\text{lattice}}(i, i') = \max(|r - r'|, |c - c'|)$. Each output node i receives an afferent connection from each of the P nodes in the *input layer*. Every afferent connection carries a non-negative, real-valued weight, w_{ij} on the connection from the j^{th} input to the i^{th} output node, and $\vec{w}_i \in \mathcal{R}^{+P}$ represents the afferent *weight vector* to the i^{th} output node. The level of activation of an input or output node ranges between 0 (inactive) and 1 (fully active). The activation levels of all P input nodes make up

the *input pattern*, a vector $\vec{x} \in [0,1]^P$ of unit length. Similarly, the activation levels of all output nodes form the *output pattern*, a vector $\vec{y} \in [0,1]^{RC}$.

In general, an input pattern $\vec{x}$ encodes the stimulation of a point on a *sensory surface*, a two-dimensional surface that is densely packed with sensors. To avoid biases due to unequal length input vectors, the planar sensory surface inputs were normalized in length by their projection onto the surface of the unit sphere. Specifically, given a point $p = [p_x, p_y]$ on the unit square, its image on the unit sphere is point $q = \left[q_x = \frac{p_x}{a}, q_y = \frac{p_y}{a}, q_z = \frac{b}{a}\right]$ where $a = (p_x^2 + p_y^2 + b^2)^{1/2}$ and $b = \sqrt{2} - (p_x^2 + p_y^2)^{1/2}$. The images on the unit sphere of the 441 points at the intersections in a regular grid of 21 rows by 21 columns covering the unit square (as visualized in Figure 3.2A) were used for training, randomly ordered.

Given an input pattern $\vec{x}$, the output pattern is determined by the same computationally efficient process employed by the standard SOM (Kohonen, 2001), except that it is generalized in a natural and biologically plausible way that causes the simultaneous existence of multiple winners. First, the *net input* h to each output node i is computed as $h_i = \vec{w}_i^T \vec{x}$ where T indicates the transpose of the column vector $\vec{w}_i$. I approximate the computationally-expensive, iterative competitive activation dynamics (Mexican Hat pattern) that is often implemented via the numerical solution of differential equations and iteratively transforms $\vec{h}$ into $\vec{y}$ (Cho and Reggia, 1994; Pearson et al., 1987; Reggia et al., 1992; Sutton et al., 1994; von der Malsburg, 1973) by a one-shot selection of winners in one step. However, unlike in the standard SOM, multiple winners can occur where each output node i which receives a net input greater than that to each of the N neighboring output nodes closest to i (ties resolved arbitrarily) is taken to be a winner. N for all output nodes, including those near or on the edges of the SOM's lattice, is taken to be the number of other output

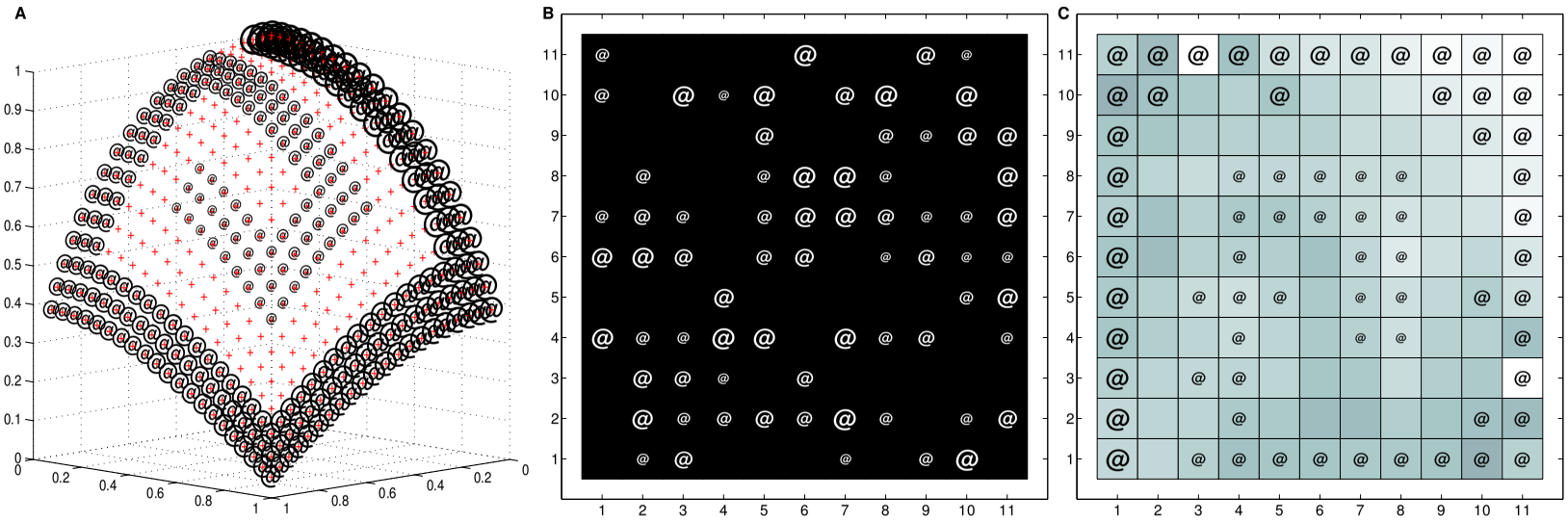


Figure 3.2: **A.** The input patterns encode representative points (the points of intersection in a grid) on a planar 2D sensory surface (shown here projected on the unit sphere's surface to normalize vector lengths). For illustrative purposes, parts of this surface are labeled with an inward clockwise spiral of '@' characters of decreasing size. **B.** The disorganized pre-training representation of the sensory surface by an 11 by 11 multi-winner SOM (one square cell per output node), showing that the labels are not arranged in any order due to the initially random weights. A cell's brightness indicates how similar the output node's weight vector is to the weight vectors of its immediate neighbors. All cells are dark indicating almost no similarity between the weight vectors of neighboring output nodes ($M = .35$). **C.** As expected, the post-training representation is like that seen with a standard single-winner SOM (Kohonen, 2001): a single topology-preserving map of the sensory surface that covers the SOM's entire lattice (indicated by @'s showing how the spiral on the sensory surface has become topographically projected onto the lattice). Since neighboring output nodes have come to represent neighboring regions on the sensory surface, the weight vectors of adjacent output nodes now are very similar (lightly shaded cells; $M = .98$).

nodes within a fixed *radius of competition* r_{comp} from the output node at location $(\lceil R/2 \rceil, \lceil C/2 \rceil)$ in the center of the lattice. Note how having the same N for output nodes along the lattice's edges is different from letting each output node compete with all other output nodes within a fixed radius from its position, which would introduce a bias favoring output nodes located near the lattice boundary. Since parameter r_{comp} is usually chosen to be small relative to the size of the lattice, typically multiple winner output nodes occur throughout the lattice in response to each input pattern. Each winner is made the central 'peak' of an 'island' of activation. The distribution of activation on a single island is such that the winner at the center of the island (output node i) is maximally active ($y_i = 1$), and the activation level of each output node j that competed with i decreases exponentially with increasing distance between j and i. Specifically, if the set V of winners is:

$$V \;=\; \{i \mid \forall j \neq i : j \text{ competes with } i \Rightarrow h_j(t) < h_i(t)\} \tag{3.1}$$

then the activation of output node j is:

$$y_j \;=\; \gamma^{\mathrm{d}(i,j)} \quad \text{with } i \in V, \text{ and } \forall k \in V,\ \mathrm{d}(k,j) \geq \mathrm{d}(i,j) \tag{3.2}$$

where $\gamma \in [0,1]$ determines the shape of each island of activation (lower γ means faster drop off from the peak). Two or more islands of activation may partially overlap. In that case, the activation level of an output node j in the region of overlap is determined by the island whose peak is closest to j. Unless stated otherwise, the parameter values used in the experiments reported here are $C = 11$ and $r_{\text{comp}} = 7$ (γ is described below).

Before training, each weight is independently initialized with a random value from the interval $[0,1]$, and each weight vector is then normalized to unit length. During training, the SOM learns by adjusting the weights on the incoming connections in

response to each input of a vector from the training set, presented in a random order that is different for each epoch. The number of training epochs used will depend on the specific training data, and will be taken to be 2000 in the following, unless explicitly noted otherwise.

For each output node i the learning rule is:

$$\vec{w}_i = \vec{w}_i + \mu y_i \vec{x} \tag{3.3}$$

$$\vec{w}_i = \vec{w}_i / ||\vec{w}_i||_2 \tag{3.4}$$

Eq. 3.3 implements typical Hebbian learning where $\mu \in (0, 1]$ is the *learning rate*. Normalization in Eq. 3.4 restricts $\vec{w}_i$ to move across the surface of the unit hypersphere, generally in the direction of the current input $\vec{x}$, and may result in a net decrease of a connection's efficacy due to competition with the other connections terminating at output node i.

Typical during the training of a standard single-winner SOM, the values of certain parameters in the above learning rule depend on how far training has progressed (Kohonen, 2001). For example, training is often divided into two phases: a rough ordering phase corresponding to large values for γ and μ, and a convergent phase corresponding to small values for γ and μ. Analogously for the one-shot multi-winner SOM, parameters γ and μ monotonically decrease in a non-linear fashion from some initial value to a smaller final value. For example, in the simulations described in the rest of this chapter, $\gamma(t) = \gamma_{\text{fin}} + (\gamma_{\text{init}} - \gamma_{\text{fin}})/(1 + e^{(t-\gamma_{\text{infl}})/\gamma_\sigma})$ where t is the fraction of completed training epochs, $\gamma_{\text{init}} = 0.9$ ($\gamma_{\text{fin}} = 0.0$) determines γ's initial (final) value, $\gamma_{\text{infl}} = 0.33$ is the point of inflection, and $\gamma_\sigma = 0.1$ determines the rate of decline. A similar function is used for μ where $\mu_{\text{init}} = 0.5$, $\mu_{\text{fin}} = 0.0$, $\gamma_{\text{infl}} = 0.5$, and $\mu_\sigma = 0.1$.

As an example, consider a one-shot multi-winner SOM as described above. Given

a radius of competition r_{comp} of 7, a one-shot multi-winner SOM of 15 by 15 or fewer output nodes ($15 = 2r_{\text{comp}} + 1$) is equivalent to a standard one-shot single-winner SOM. This is because each output node competes with all other output nodes for activation and learning under these conditions, and hence there is always only a single winner for a particular input (Kohonen, 2001). Given the input patterns shown in Figure 3.2A (they constitute a representative sample of a square planar sensory surface), Figure 3.2B shows a typical example of the initial disorganized state of an 11 by 11 SOM's representation of this sensory surface prior to training that is due to the random initialization of the SOM's afferent weights. Figure 3.2C shows, for the same SOM, the ordered map representation that was formed by training the network with the patterns from Figure 3.2A. As expected, when 11 by 11 (and 11 by 15) one-shot multi-winner SOMs were trained, each self-organized into a single topology-preserving map of the sensory surface that covered the entire lattice (Figure 3.2C), just as would be expected with Kohonen-style SOMs.

The one-shot multi-winner SOM was implemented in Matlab and C. C was chosen for the computationally very costly core training algorithm, while Matlab was used for the simulation framework which included the management, analysis and visualization of the training and simulation data. Each simulation typically involved a large batch of training runs where each run corresponded to a single operating system process. The processes were distributed across the eight CPUs of a networked pool of seven Sun Microsystems workstations (Ultra II/V and Blade 1000/1500/2000 models) running the Solaris operating system. A Perl script handled the automatic distribution and bookkeeping of the processes. On the fastest of these machines (Sun Blade with 900MHz UltraSparc processor and 1GB RAM), the training of a single one-shot multi-winner SOM with an output lattice of 30 by 20 nodes over 2000 epochs using the

441 3D input patterns described above took more than an hour of CPU time. I now turn to what occurs with larger one-shot multi-winner SOMs, after first considering the occurence of multiple adjacent topographic maps in biological cortex.

3.2 Multiple Mirror-Image Cortical Maps and a Hypothesis

Experimental studies have repeatedly established the existence of multiple neighboring cortical maps where the layout or topology of adjacent maps is mirror symmetric. Familiar examples of adjacent mirror image cortical maps include multiple representations of the body surface in primary somatosensory cortex of monkeys as illustrated in Figure 2.4II (Merzenich et al., 1978; Sur et al., 1982) and several mirror image tonotopic maps in primary auditory cortex (Heschl's gyrus) in humans (Engelien et al., 2002; Formisano et al., 2003). If one considers not only primary but also secondary sensory cortex (which also receives thalamocortical projections), numerous other mirror image maps have been found in somatosensory (Beck et al., 1996; Krubitzer and Calford, 1992; Krubitzer et al., 1995; Nelson et al., 1980), visual (Drager (1975); Newsome et al. (1986); Sereno et al. (1995); Tiao and Blakemore (1976); see also Figure 2.4III), and auditory (Imig et al., 1986; Pantev et al., 1995; Rauschecker et al., 1995; Talavage et al., 2000) cortex in a variety of species. In addition, mirror image movement representations have been found in the motor cortex of the macaque monkey (Gentilucci et al., 1989). While there are many different hypotheses about why multiple and sometimes apparently redundant maps occur so often (separation of spatial/temporal processing, parallel processing of different sensory attributes, evolutionary factors, minimization of connection distances, etc.) (Kaas, 1988; Cowey, 1981; Jones, 1990), there has been little speculation as to why such maps often exhibit reflection symmetry, and the mechanisms by which multiple, mirror image

maps arise during evolution and neurodevelopment remain unclear. Past computational models of self-organizing neocortical topographic maps (Kohonen, 2001; Ritter et al., 1992; Sutton et al., 1994; Pearson et al., 1987; Sirosh and Miikkulainen, 1994) have generally been limited to single maps and thus do not shed substantial light on this issue.

Given the basic one-shot multi-winner SOM described above, I hypothesized that multiple adjacent mirror-symmetric maps would arise from Hebbian synaptic changes whenever the distribution radius of afferents to the output (or cortical layer) sufficiently exceeds that of horizontal intracortical connections (Brown et al., 2001). Further, I expected that these maps would turn out to be mirror images of one another due to the basic properties of Hebbian learning. These hypotheses were inspired by the adjacent mirror-image maps in biological cortex. There, a stimulated area does not show an activation pattern involving a single-winner situation where only one cortical column and its immediate neighbors are active while all others are inactive, at least if the area considered is sufficiently large. Typically each initially highly active location retains or further increases its activity while inhibiting the activity of the less active regions that surround it, producing a more distributed, multi-focal pattern of activation (Donoghue et al., 1992; Georgopoulos et al., 1988; Pei et al., 1994). Carried over to the SOM, this corresponds to each output node competing only locally, that is, only with output nodes that are close enough (do not exceed some maximum distance) in the lattice. As a consequence, there will be multiple winners, distributed across the lattice, widely separated from one another, and learning concurrently from the same input, a behavior that is implicit in the "Mexican Hat" patterns of activity occurring in some more biologically realistic, but also more complex and computationally expensive models of cortex (Cho and Reggia, 1994; Pearson et al., 1987; Sutton

et al., 1994; von der Malsburg, 1973).

My hypotheses can thus be viewed as stating that the otherwise unaltered standard SOM learning method, when generalized to multiple winners, is sufficient to produce adjacent mirror image maps that are qualitatively similar to those observed in experimental studies. More specifically, I postulated that Hebbian learning combined with range-limited competition for activation and learning alone can explain the existence of mirror image maps in cortex. The precise circumstances under which this occurs are of special interest, and may provide testable predictions as to some of the conditions that prevail in biological cortex.

3.3 Quantitative Measures of Map Formation with Multiple Maps

To test the plausibility of the hypotheses given in the previous section, I examined a series of simulations using the one-shot multi-winner SOM formulation that had sufficiently large output/cortical lattices to permit multiple maps to form during learning. Before discussing the results of these simulations, it is important to clarify how map formation can be measured quantitatively when multiple maps are present.

Measures of the "goodness" of map formation such as the topographic product (Bauer and Pawelzik, 1992) or the topographic function (Villmann et al., 1997) have of course previously been devised to quantify the 'goodness' of a map in terms of how well the topology of the sensory surface (or, in general, the input space) is preserved on the SOM's lattice. In a standard SOM, a single output node i wins the global competition for activation for all input vectors $\vec{x}$ which satisfy that $\forall j \neq i$: $\vec{w}_i^T \vec{x} \geq \vec{w}_j^T \vec{x}$ (ties are resolved arbitrarily). This region of the input space is called the receptive field of output node i. The set of all receptive fields corresponds to the Voronoi tessellation of the input space where each weight vector is at the center of

one of the Voronoi cells. This Voronoi tessellation gives rise to a natural definition of of what it means for two output nodes to be adjacent *in the input space*: the corresponding two receptive fields or Voronoi cells have to be adjacent to each other (i.e., they share part of their boundaries). Intuitively, the classic definition of topology preservation demands that each pair of two in the lattice adjacent output nodes have to be adjacent in the input space *and vice versa*. Existing measures of topology preservation essentially count, weigh and sum the violations of this definition that occur in a particular map to express in a single number how close the map is to being perfectly topology preserving.

Figure 3.3 (next page): **A.** A multi-winner SOM where the lattice is a 1D string of seven output nodes. The dashed lines indicate adjacency between output nodes with respect to the lattice. **B.** The rectangular 2D sensory surface or input space. The representative input vectors have been labeled 'I', 'II', 'III' or 'IV' so that the input space is subdivided into four square sectors. **C.** The multi-winner SOM shown in the input space. The output nodes have been placed according to the positions of their weight vectors. The output nodes define a Voronoi tessellation of the input space where each output node is at the center of a Voronoi cell. The boundaries between the cells are shown as solid lines. **D.** The multi-winner SOM where each output node carries the label of the input vector that is closest to the output node's weight vector (like in Figures 3.2A and B, 3.4 and 3.5). The SOM has folded in the input space, resulting in two maps that are mirror images of each other where output node 4 corresponds to the 'axis' of reflection and is shared by both maps. **E., F.** These

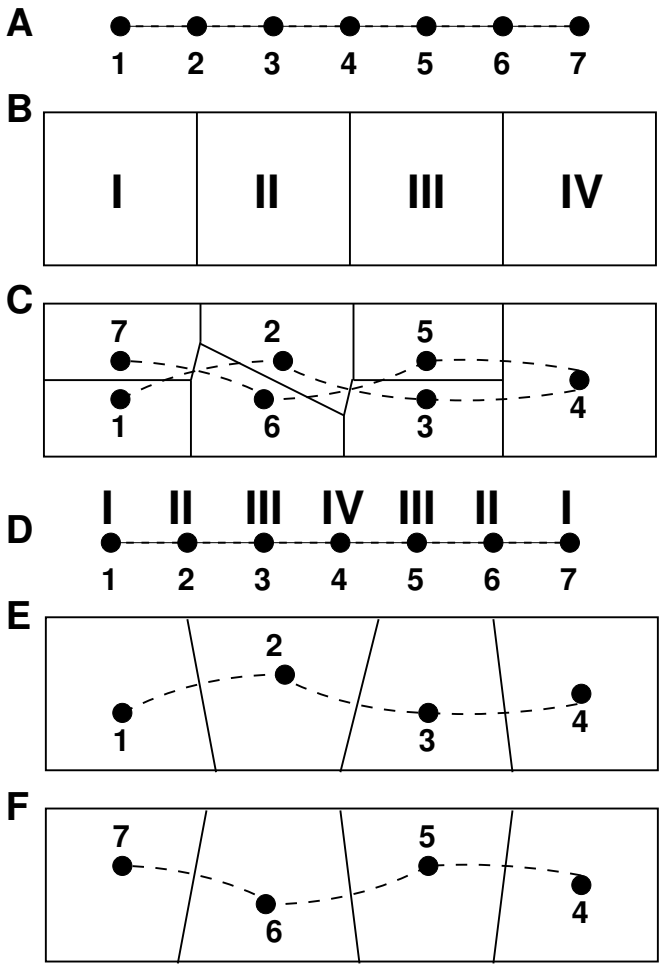


Figure 3.3: Caption starts on previous page

Voronoi diagrams show that each map by itself perfectly preserves the topology of the input space, i.e., output nodes that are adjacent in the network lattice are adjacent in the input space and vice versa. However, the diagram for all output nodes shown in **C.** contains violations of the input topology: output nodes 1 (5) and 2 (6) are adjacent in the lattice, but not in the input space; output nodes 1 (2, 3) and 7 (6, 5) are adjacent in the input space, but not in the lattice.

Unfortunately, the conditions contained in the above definition are too rigorous to be sensible with respect to the multi-winner SOM and hence, so are existing measures of topology preservation. Figure 3.3 illustrates how both directions of the definition can be violated by a multi-winner SOM, even though it has formed two individually perfectly topology preserving maps of the input space. However, Figure 3.3C suggests that if a multi-winner SOM has formed multiple individually topology preserving maps of the input space then two output nodes that are adjacent in the lattice are still relatively close in the input space, even if they are not immediately adjacent there. For these reasons, I devised a measure $M \geq 0$ to quantify whether and to what degree a particular multi-winner SOM has self-organized into multiple individually topology preserving maps of the input space. M is the mean of the *smallest* 2% of entries in the collection of all pairwise dot-products between the weight vectors of in the lattice adjacent output nodes, that is, if $\vec{c} = (c_1, c_2, \ldots, c_K)$ is a vector with its components in ascending order that comprises all dot-products of the form $\vec{w}_i^T \vec{w}_j$ where output node i is adjacent to output node j, then $M = 1/\lceil .02K \rceil \sum_{k=1}^{\lceil .02K \rceil} c_k$. The dot-product between the weight vectors of two output nodes is inversely proportional to the distance between the nodes' receptive field centers, that is, the two locations on the sensory surface to which the weight vectors point. Thus, M is inversely proportional to the average distance with respect to receptive field centers between in the lattice adjacent output nodes, but limited to those parts of the lattice where these distances are greatest (i.e., larger values of M indicate better map formation). Note that M is only sensible if the input space has a roughly uniform density, and it does not take into account the weight-vector-induced Voronoi tessellation of the input space so that it is not a direct measure of topology preservation.

In addition to the quantitative and objective measure M, map formation can be

assessed subjectively by visual inspection. Thus, solely for the purpose of visualizing map formation, the center of the location of each input pattern is associated with a label which is either the character '@' of a particular size or the blank character. When these labels are shown on the sensory surface at their associated positions as seen in Figure 3.2A, they form an inward clockwise spiral with the size of the '@' labels decreasing toward the center of the spiral. This distinctive superimposed pattern allows one to reliably judge, via visual inspection of the SOM's lattice, whether or not a roughly topology-preserving map of the sensory surface has formed, and if so, its orientation. In the following, the modeled cortical surface is shown as a 2D array of square cells, one cell for each output node. For each output node i, the corresponding cell carries the label of the input pattern $\vec{x}_j$ to which the node is most sensitive, that is, $j = \text{argmax}_k(\vec{w}_i^T \vec{x}_k)$. Each topology-preserving or well-formed map of the sensory surface thus shows up on the lattice as a projected image of the spiral pattern. The image may be rotated or slightly distorted, and/or may show a reversal of the spiral's direction from clockwise to counterclockwise, since these transformations do not violate the topology of the sensory surface.

3.4 Appearance and Relationships of Multiple Maps

This section presents the results of the simulations that were conducted with the the one-shot multi-winner SOM. The presentation includes four sets of results: the observed numbers of topographic maps that networks of different sizes formed and the symmetry relationships between those maps, the quantitative differences in terms of map formation between the observed types of symmetry, the impact of a non-uniform distribution of input patterns across the input surface on the relative orientation of multiple maps, and, finally, the robustness of simulation results to changes in the

parameters of the one-shot multi-winner SOM.

3.4.1 Number and Symmetry Relations

Fourteen separate experiments were conducted, each corresponding to a specific SOM lattice size ($R = 11, 15, 20, \ldots, 75$; $C = 11$), and for each size lattice 20 independent runs were executed. Each run consists of training the network, recording the number of resulting individual maps, and, in the case of multiple maps, documenting any symmetries between immediately adjacent maps. The runs of each single experiment were independent from one another: in each, the network was initialized with different random weights and a different random order was used for the presentation of the input patterns during training.

As can be seen from the left half of Table 3.1, for a sufficiently small $R \leq 20$, the one-shot multi-winner SOM was essentially equivalent to a standard single-winner SOM, and consequently, only a single map of the sensory surface formed, covering the entire lattice of output nodes (as in Figure 3.2C). With $R \geq 25$, multiple well formed maps appeared, such as those illustrated in Figure 3.4. In general, the number of maps formed increased proportional to R: approximately one additional map was formed for each additional 15 rows. This suggests that in general, the number of additional rows required to accommodate an additional map roughly equals the 'diameter' of competition, that is, the number of rows $2r_{\text{comp}} + 1$ (for $R \geq 2r_{\text{comp}} + 1$) of other output nodes with which each output node has to compete for activation and learning (e.g., 15 for $r_{\text{comp}} = 7$). All the maps in any one instance where multiple maps occurred were generally of the same size. Two adjacent maps were usually immediately adjacent, that is, there were no lattice parts in between them that were not part of the two adjacent maps, regardless of R.

Table 3.1: Averages over 20 Runs of Numbers and Symmetries of Maps

	Number of Maps			Pairwise Symmetries*		
R	mean	min.	max.	m	g	r
11	1.00	1	1	-	-	-
15	1.00	1	1	-	-	-
20	1.00	1	1	-	-	-
25	2.00	2	2	1.00	.00	.00
30	2.00	2	2	.90	.10	.00
35	2.12	2	3	.63	.37	.00
40	2.95	2	3	.92	.03	.05
45	3.00	3	3	.78	.10	.13
50	3.40	3	4	.73	.19	.08
55	3.83	3	4	.81	.15	.04
60	4.00	4	4	.93	.02	.05
65	4.50	2+	6	.77	.20	.03
70	5.07	2+	7	.82	.05	.14
75	5.18	3+	6	.77	.12	.11

*m = mirror reflection, g = glide reflection, r = 180^o rotation, + = unorganized areas also present

Three types of map-to-map symmetries were observed. In the overwhelming majority (82%) of cases, the two adjacent maps were mirror images of each other (e.g., Figure 3.4A). The second type of symmetry observed, found in 11% of the cases, was again essentially a mirror reflection, but now the axis of reflection was tilted so that the boundary between the two maps was no longer of minimal length (Figure 3.4B). In addition, the maps were translated in opposite directions along their common tilted boundary so that the resultant transformation is better characterized as a glide reflection. Thus, in 93% of the cases, adjacent maps exhibited mirror symmetry or distorted mirror symmetry reminiscent of that seen in biological neocortex. In the remaining 7% of map pairs[1], each individual map was characterized as a rotation relative to the other of 180 degrees around a symmetry point at the center in between the two maps (Figure 3.4C). The rightmost three columns of Table 3.1 show the fractions of mirror (m), glide (g) and rotation (r) symmetries between adjacent maps for different lattice sizes R, averaged over 20 independent runs, respectively. For a complete account of the training results for each individual run (non-averaged results), see Table A.1 in Appendix A.

Map visualizations like those in Figure 3.4 also revealed that the three symmetry types exhibited distinct patterns of similarity among the output nodes along an inter-map boundary. Output nodes along the boundaries between mirror symmetric maps typically were similar to their neighbors in the lattice, that is, their afferent weight vectors and thus, their receptive field centers were close to one another. In the example shown in Figure 3.4A, this becomes manifest in the form of lightly shaded cells along the inter-map boundary. Dissimilar output nodes (darkly shaded cells)

[1]The phrase 'map pair' always refers to two maps that are adjacent on the modeled cortical surface, i.e., the lattice of output nodes.

as part of the inter-map boundary were characteristic for both glide reflection and rotationally symmetric maps. However, while output nodes like that were present all along the boundary between glide reflection symmetric maps (Figure 3.4B), their presence was limited to the outer reaches of the inter-map boundary in the rotationally symmetric case (Figure 3.4C).

Figure 3.4 (next page): Representative instances where the cortical lattice formed multiple maps (top) and their corresponding schematic representation (bottom) for each of the three observed types of symmetry between adjacent maps of the sensory surface. The spatial organization of the maps is indicated by how a single spiral painted on the sensory surface (see Figure 3.2A) is replicated and oriented on the map. **A.** a mirror symmetric, **B.** a glide reflection symmetric ('distortedly' mirror symmetric), and **C.** a rotationally symmetric map pair. In the schematic representations at the bottom, the thin lines in **A** and **B** and the point in **C** indicate the symmetry axis and the center of rotation, respectively.

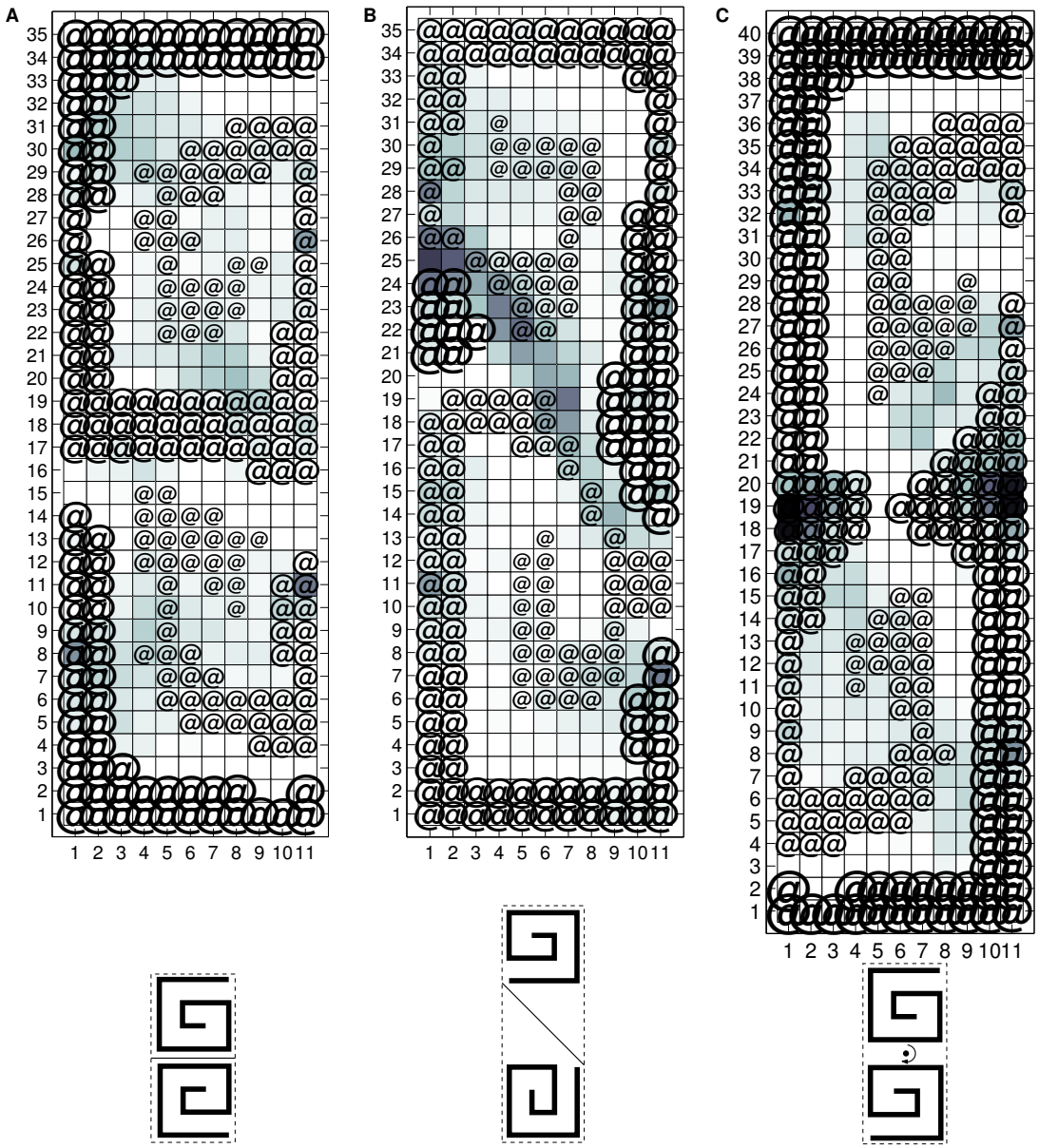


Figure 3.4: Caption on previous page

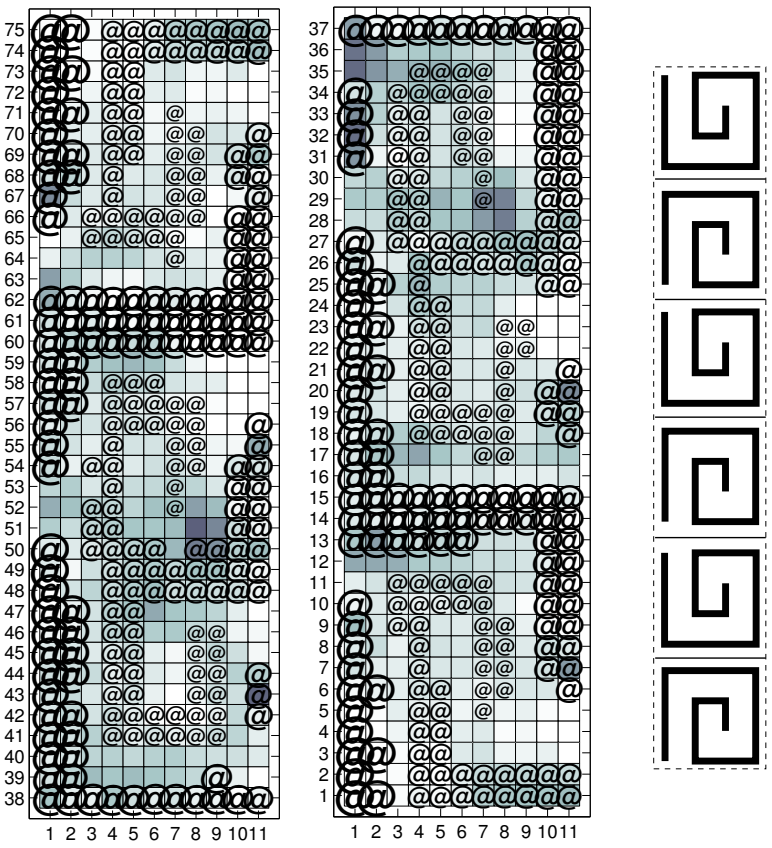


Figure 3.5: A single cortical lattice on which six maps of the sensory surface appeared where every two adjacent maps are always mirror images of each other. For illustrative purposes, the lattice has been split in the middle, with the top half shown on the left and the bottom half shown on the right. A schematic representation is given on the right.

There was some tendency for the largest fractions of mirror symmetric maps ($>$ 80%) to occur when R was a multiple of 15 or slightly smaller than that ($R = 25, 30, 40, 55, 60, 70$ in Table 3.1). So, for the formation of n mirror symmetric map pairs, an R equal to or slightly less than n times the 'diameter' of competition seems to be optimal. Under these optimal conditions, the height of a single map was roughly

15. However, I found several cortical lattices on which exclusively mirror symmetric map pairs formed and the number of maps exceeded the expected value because they were smaller. The *single* cortical region in Figure 3.5 provides an example of this where six somewhat compressed maps formed on the 75 by 11 lattice.

In a small minority of cases, the network did not completely self-organize, and parts of the lattice remained disorganized after learning. For example, the entry 2^+ for $R = 65$ in Table 3.1 indicates that in one of the 20 simulations with networks of this size, only two representations of the sensory surface were found, with the rest of the lattice being disorganized (all other 19 simulations in this case exhibited at least 4 maps and no disorganized regions).

3.4.2 Measuring Map Formation and Types of Symmetries

For the 220 simulations with cortical regions sufficiently large for multiple maps to appear $(R \geq 25)$, the mean initial value of M prior to any learning was 0.31 (SD 0.02, minimum 0.23, maximum 0.38). Following learning, this increased to 0.97 (SD 0.02, min. 0.87, max. 0.98). Each cortical lattice that was in principle large enough for multiple well-formed maps to appear (all 220 runs in Table 3.1 for which $R \geq 25$) was assigned to one of three categories. A lattice's category depends first on whether it shows any disorganized regions. If so, the lattice belongs to the '?' category (20 runs, or 9%), even if well-formed maps were also present. The remaining lattices are divided into those in category 'm' that formed exclusively mirror symmetric map pairs (138, 62%) and those in category '$g|r$' (62, 29%) that showed at least one glide reflection or rotationally symmetric map pair.

Figure 3.6 shows, for each lattice category, the distribution of the M values. The mean M value was 0.980 (SD 0.002) for category m simulations and a significantly

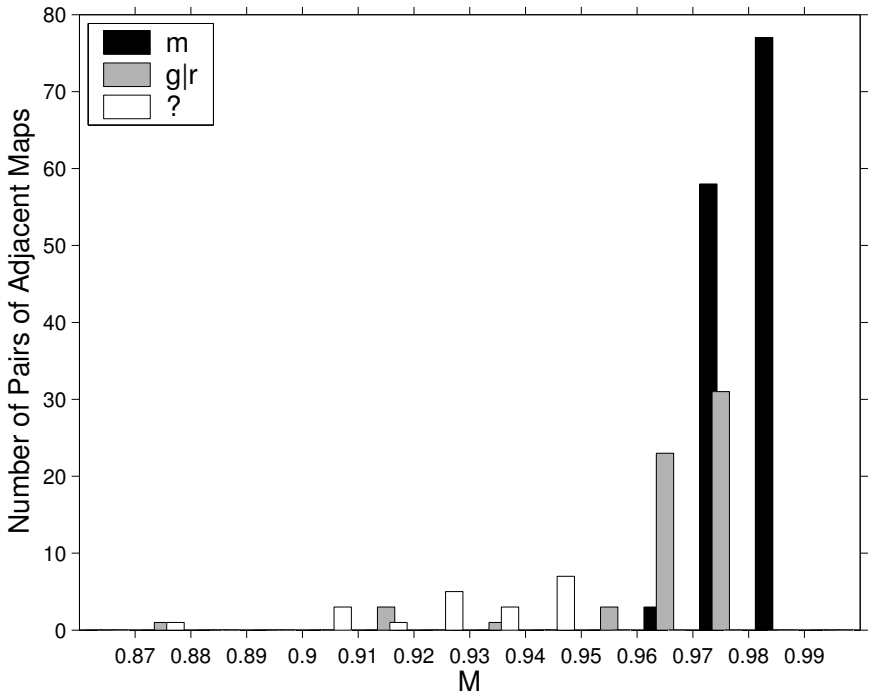


Figure 3.6: A histogram of the M values for each of the three symmetry categories. The values have been grouped into 16 consecutive intervals of width 0.01. To be comparable across the differently sized categories, each histogram shows, for each interval, the relative within-category frequency with which M fell within the limits of the interval. The histograms suggest that the means μ and standard deviations σ of the actual distributions of M values are ordered so that $\mu_m > \mu_{g|r} > \mu_?$ and $\sigma_m < \sigma_{g|r} < \sigma_?$.

different 0.965 (SD 0.018) for category g|r simulations ($p < 10^{-3}$ on t-test). On average, the M values were significantly greater for category 'm' than for category '$g|r$', and the spread of the values was smaller for category 'm' than for category '$g|r$'. Further, the average M values were significantly greater for category '$g|r$' than for category '?', with the spread being smaller for '$g|r$' than for '?'[2]. Since M primarily measures the organization along map boundaries when multiple maps are present, these results indicate that the same synaptic modifications responsible for individual map formation also tend to maximally preserve similarity of adjacent cortical element receptive fields along map boundaries by producing adjacent maps that are mirror symmetric. In contrast, other symmetry relationships (glide, rotational) are "local maxima" of M in which the map formation process becomes trapped during learning. Since category '$g|r$' lattices also exhibited mirror-symmetric map pairs, the differences observed between categories 'm' and '$g|r$' most likely would have been even more pronounced if each pair of adjacent individual maps had been manually categorized individually and if M had been measured separately for each pair (this was impractical to do).

3.4.3 Non-Uniform Density of Sensory Stimuli

In all of the above experiments, each representative point of the sensory surface was stimulated exactly once during a single epoch of training. This uniform distribution of input stimuli did nothing to bias which of the edges of mirror image maps became adjacent. Notwithstanding a reflection of the entire cortical lattice with respect to its

[2]Assuming that the average of M was the same for all three categories, the Jonckheere trend test gave a probability of $\ll .0001$ for the observed M values to be a product of chance, providing support for the alternative hypothesis that, on average, M was greater for category 'm' ('$g|r$') than for category '$g|r$' ('?').

vertical midline, there are only four different ways in which two adjacent and mirror symmetric individual maps may be oriented relative to each other. As is illustrated in Figure 3.7, each of these relative orientations (A, B, C and D) corresponds to a particular side of the square sensory surface (and hence, the superimposed spiral pattern) being represented by and coinciding with the inter-map boundary. Given a uniform distribution of sensory stimuli, each orientation should occur with roughly the same frequency.

Figure 3.7 (next page): Schematic drawings and examples illustrating the four distinct ways (**A**, **B**, **C**, and **D**) in which two adjacent mirror symmetric maps may be oriented relative to each other. No distinction is being made between the reflections of the map pair with respect to the vertical midline of the cortical lattice. The conceptual partitioning of the sensory surface into three equally sized, but potentially differently often stimulated regions is depicted in the schematic drawings where each individual image of the spiral pattern consists of three differently shaded strips, each an image of one of the three regions of the sensory surface.

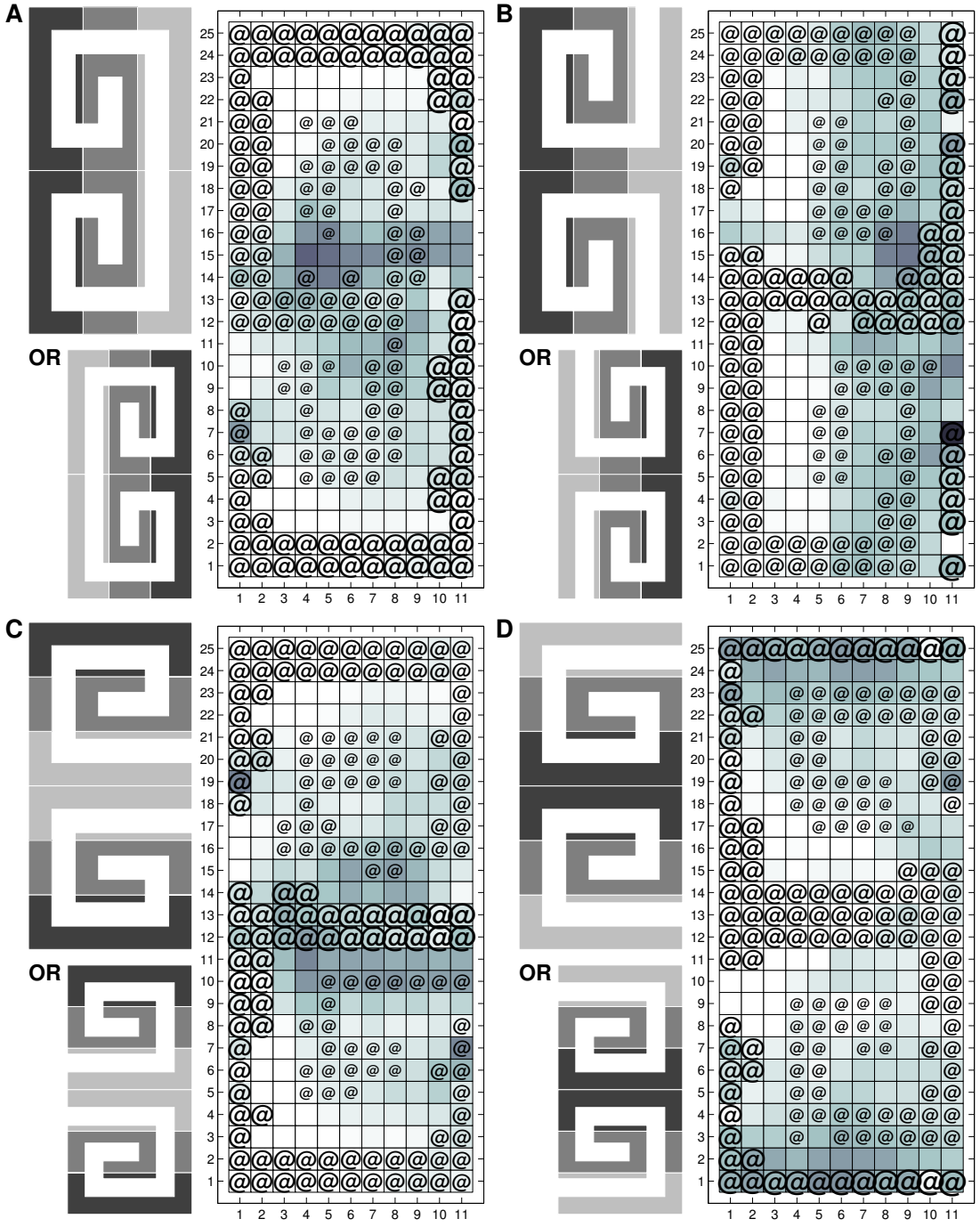


Figure 3.7: Caption on previous page

To see if the selection of which edges become adjacent during formation of mirror image maps could be biased, I altered the previously uniform probability distribution of stimuli over the sensory surface during learning. This was done by partitioning the regular 21 by 21 grid of points that serve as a representative sample of the sensory surface into three consecutive 21 by 7 sub-grids (I, II and III). Each point in sub-grid I was stimulated, as before, only once during a single epoch of training while each point in sub-grid II (III) was stimulated twice (three times) during the same period. During training, I used a fixed lattice size of 25 by 11 output nodes and coordinate-encoding input patterns, a combination which earlier had produced a pair of mirror image maps in 19 out of 20 runs (see Table A.1 in Appendix A) when uniformly probable input stimuli were used. Of these 19 mirror image map pairs, six (three; six; four) were composed of individual maps that were oriented like in Figure 3.7A(B; C; D) which is consistent with a uniform distribution ($\chi^2 = 1.42 < 28.87$ for the χ^2 goodness-of-fit test which, at the 0.05 significance level, confirms the hypothesis that the observed frequencies are from a uniformly distributed population). In comparison, of the 20 independent runs of training that were performed using the non-uniform sensory surface, 19 again produced mirror image map pairs, but now orientation A (B; C; D) occurred twice (never; never; 17 times) which corresponds to a significant preference for orientation D ($\chi^2 = 42.68 > 28.87$ for the χ^2 goodness-of-fit test, i.e., at the 0.05 significance level, the observed frequencies are inconsistent with a uniformly distributed population). At orientation D, the most frequently stimulated region of the non-uniform sensory surface is represented in both maps along the inter-map boundary (see Figure 3.7D), and the map representations of the least frequently stimulated region are the most removed from the boundary.

Biological sensory surfaces such as the skin or the retina exhibit a non-uniform

density of sensors (e.g., the macula versus the rest of the retina) as well as a non-uniform usage pattern where some regions are stimulated more often than others (e.g., the fingertips versus the back of the hand). Higher density regions and regions that are more frequently stimulated are typically magnified in cortical maps, that is, their cortical representation occupies a disproportionately large area of cortical surface (Azzopardi and Cowey, 1993; Creutzfeldt, 1978; Dykes and Ruest, 1984; Sereno et al., 1995). The single-winner SOM model is capable of reproducing this magnification effect, given inputs that model a non-uniform sensory surface (Grajski and Merzenich, 1990).

As expected, the images of the two more often stimulated sensory surface regions came to occupy a relatively larger area of the SOM's lattice (at the expense of the third least stimulated region) compared to when the three regions were equally frequently stimulated during training (uniform sensory surface). With the uniformly distributed sensory stimuli (the baseline), the map representations of the region corresponding to sub-grids I, II and III consumed, on average, 79.05, 104.0 and 91.95 output nodes with standard deviations of 13.04, 3.76 and 13.58, respectively. In comparison, with the non-uniformly distributed sensory stimuli, the averages for regions I, II and III were 57.58, 112.0 and 105.42 with standard deviations of 4.95, 3.30 and 7.70, respectively. This is a significantly different result. Given the hypothesis that the two samples are from the same population, the U-test returned a probability of 1.99e-07 (2.69e-06; 0.0056) in favor of the observed differences between the two samples being a chance event which provides strong support for the alternative hypothesis that the sample for the non-uniform sensory surface is from a population with a smaller (larger) mean.

3.4.4 Sensitivity to Model Changes

Certain model parameters had a substantial impact on whether or not a multi-winner SOM self-organized into well formed maps of the sensory surface, and what the likelihood of occurrence was for each of the three types of symmetries between adjacent maps.

For the radius of competition $r_{\text{comp}} = 7$ used in the above simulations, I observed the largest fraction of mirror symmetric map pairs for a lattice width of $C = 11$ (0.82m, 0.11g, 0.07r). Experiments with $C < 11$ resulted in a relatively larger number of glide reflections (e.g., 0.69m, 0.27g, 0.04r for $C = 9$). For $C > 11$, the relative number of rotational symmetries increased (e.g., 0.78m, 0.04g, 0.18r for $C = 13$, and 0.68m, 0.02g, 0.30r for $C = 15$). So, it seems that for a given radius of competition r_{comp}, a particular width C of the lattice is optimal for the formation of mirror symmetric map pairs. If C is smaller (greater) than the optimal value, the fraction of mirror symmetric pairs decreases while the fraction of glide reflection (rotationally) symmetric pairs increases.

The initial value of γ, γ_{init}, and γ_{infl} were also important. For $\gamma_{\text{init}} = .8$ (rather than .9 as in the experiments above), the fraction of mirror symmetric map pairs dropped to typically 60%. Rotation and glide reflection symmetry became more frequent with each occurring in roughly 20% of the cases. In general, values of γ_{init} smaller than 0.9 seem to disproportionately increase the fraction of rotationally symmetric map pairs. Delaying γ's descent by increasing γ_{infl} to 0.5 increased the number of cases in which self-organization failed partially or completely so that no well-formed maps were discernible in (parts of) the SOM's lattice. The effects of parameter changes pertaining to γ especially depend on how the learning rate μ changes over time during training. I made the above observations on the effects of changes to γ_{init} and γ_{infl}

while $\mu_{\text{init}} = 0.5$, $\mu_{\text{infl}} = 0.5$ and $\mu_{\text{sigma}} = 0.1$ were held fixed.

Two variations of Eq. 3.2 were implemented and tested as well. The first variant determines the activity of an output node by taking into account all winners (as opposed to just the closest one) and adding their activity contributions. So, Eq. 3.2 was replaced by

$$y_j = \sum_{i \in V} \gamma^{\mathrm{d}(i,j)} \tag{3.5}$$

which, in general, increases the activity of output nodes that are located in an area of the lattice where several islands of activation overlap. Given Eq. 3.5, it is actually possible that an output node in an area of overlap becomes more active than a winner. The second variant prevents this from happening by capping each output node's activation if it exceeds 1. Both variants were less conducive to the formation of mirror symmetric map pairs than the original rule in Eq. 3.2.

I used coordinate-encoding input patterns in most of my experiments since, especially during the search for suitable training parameters, computational efficiency was critical. However, in order to demonstrate that coordinate-encoding does not bias the model in favor of my hypotheses about map formation, the first 10 runs of each experiment were repeated, except now the networks were trained with high-dimensional *sensory activation patterns* as the input patterns. A sensory activation pattern is a vector with as many components as the number of sample sensory surface points (441). So, it is computationally much more expensive to use sensory activation patterns (*full encoding*) as the inputs to the model than it is to use 2D (3D after normalization) coordinate vectors (*coordinate encoding*) as I did in the original experiments which is why I did not repeat all of the experiments.

Each sensory activation pattern comprises the activation levels of all sensory surface points in response to the stimulation of one of the points. Stimulation of a point

Table 3.2: Averages over 10 Runs each of Numbers and Pairwise Symmetries of Learned Maps

	Number of Maps			Pairwise Symmetries*		
S	mean	min.	max.	m	g	r
11	1.00	1	1	-	-	-
15	1.00	1	1	-	-	-
20	1.00	1	1	-	-	-
25	2.00	2	2	1.00	.00	.00
30	2.00	2	2	1.00	.00	.00
35	2.22	2	3	1.00	.00	.00
40	3.00	3	3	1.00	.00	.00
45	3.00	3	3	.95	.00	.05
50	3.40	3	4	.92	.08	.00
55	4.00	2+	4	.94	.03	.03
60	4.00	3+	4	.79	.21	.00
65	4.78	3+	5	.94	.00	.06
70	5.20	5	6	.93	.07	.00
75	5.50	5	6	.82	.09	.09

*m = mirror reflection, g = glide reflection, r = 180^o rotation, + = unorganized areas also present

on the sensory surface evoked a bell-shaped activation pattern: maximum activation at the center and a monotonous decrease in activation with increasing distance from the center. Specifically, if the stimulation of point $p = (p_x, p_y, p_z)$ is encoded by $\vec{x}^{(p)}$ then $x_q^{(p)}$, the component of $\vec{x}^{(p)}$ corresponding to the activation level at point $q = (q_x, q_y, q_z)$, equals $\frac{p_x q_x + p_y q_y + p_z q_z}{(\sum_{q'}(p_x q'_x + p_y q'_y + p_z q'_z)^2)^{1/2}}$. This implies that the sensory activation patterns evoked by two separate and independent point stimuli are the more correlated the smaller the distance on the sensory surface is between the two points (this is true also if coordinate encoding is used). The formation of topographic maps in the brain then can be explained as the consequence of a tendency to reduce the distance on, for example, the surface of the neocortex between the representations of highly correlated afferent signals. The same principle can also explain the formation of computational or feature maps in the brain where the afferent signals do not originate from a sensory surface, but instead reside in an abstract internalized input or feature space.

The overall results, which are given in Table 3.2 (for detailed run-by-run results, see Table A.2 in Appendix A), were 91% mirror symmetric, 6% glide reflection symmetric and 3% rotationally symmetric map pairs, indicating a significant increase in the fraction of (distorted or undistorted) mirror symmetric map pairs increased significantly from 0.94 to 0.97. The statistical significance was verified with a one-sided χ^2-test. The sample size was 500 (272) map pairs for the coordinate (full) encoding, 465 (265) of which were mirror symmetric (distorted or undistorted). Consequently, $\chi^2 = 6.71$, that is, the observed difference between the two fractions is unlikely due to chance ($p < .005$). The average number of maps appearing on the cortical lattice was not significantly different at the 0.05 significance level, regardless of the specific lattice size R. According to the U statistical test, the difference in the average num-

ber of maps per lattice was closest to being significant for the 75 by 11 lattice size ($p = 0.08$). This provides experimental support for using coordinate-encoding input patterns as this did not influence the average number of well formed maps per lattice and, more importantly, did not bias the one-shot multi-winner SOM in favor of mirror symmetric adjacent maps.

3.5 Discussion

The one-shot multi-winner SOM introduced in this chapter, when trained with input patterns that encode the stimulation of points on a sensory surface, formed multiple, individually topologically correct maps of the sensory surface. As hypothesized, multiple maps arose whenever the distribution radius of cortical afferents sufficiently exceeded that of horizontal intracortical interactions (Brown et al., 2001). For a particular set of model parameters, adjacent maps were largely mirror symmetric with respect to their common boundary while for a wide range of model parameters, mirror symmetry was at least predominant. Two other types of symmetry, glide reflection and rotational symmetry, occurred between adjacent maps where the former is essentially a form of mirror symmetry, albeit somewhat distorted. When the sensory surface was subdivided into regions, some being stimulated more often during training, two adjacent maps, in addition to being mirror symmetric, were almost always oriented in such a way that their representations of the most often stimulated region were located next to each other at the inter-map boundary. The other regions were represented farther away from the inter-map boundary, following the gradient in the frequency of stimulation. Further, the more often stimulated regions were represented by a relatively larger area of modeled cortical surface in each of the individual maps, similar to the magnification of more frequently stimulated sensory regions that occurs

in biological maps (Azzopardi and Cowey, 1993; Creutzfeldt, 1978; Dykes and Ruest, 1984; Sereno et al., 1995).

The results of this study may have some significant implications in terms of an understanding of the occurrence of mirror image maps in the brain. They indicate that after the initial afferent and intracortical wiring, no genetic or other mechanisms beyond the competitive Hebbian learning used to produce topographic map formation are needed to explain the occurrence of multiple mirror image maps. The model's preference for the formation of multiple, individually topology-preserving maps that are pair-wise mirror symmetric can be explained by the tendency of competitive Hebbian learning to both represent the entire sensory surface *and* minimize the number of output node pairs that are relatively close in the lattice, but whose weight vectors are relatively far apart on the sensory surface. A single topology-preserving map of the sensory surface, like that typically formed by a standard single-winner SOM, is the optimal solution to this minimization problem. A single global competition for activation and learning is essential for the global self-organization of the entire lattice into a single topology-preserving map. However, with a multi-winner SOM, the information about which output node responds most to a particular input pattern is only locally available. The process of self-organization generally compensates for this lack of global knowledge by forming multiple small, but by themselves locally optimal solutions to the minimization problem, usually optimizing the transitions between them. The optimal transition manifests itself in the mirror symmetry that was observed between most adjacent maps of the sensory surface.

As illustrated in Figure 3.8A, *in the input space* this optimal transition corresponds to a perpendicular fold in the SOM's lattice. The other two types of symmetry that were observed constitute suboptimal transitions from one map to the next. A glide

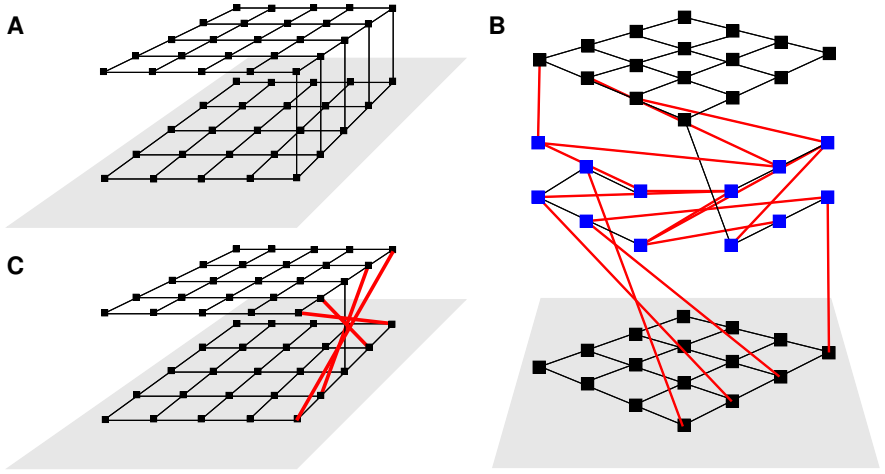


Figure 3.8: Various ways in which the cortical lattice became embedded in the input space, a 2D sensory surface which is shown as a light gray area in the x/y-plane. Two output nodes (small black rectangles) are connected by a solid line iff they are adjacent in the lattice. Each case involves two roughly topology-preserving maps of the sensory surface which have been spatially separated along the z-axis to illustrate the distortions of the lattice *in the input space*. For illustrative purposes, the lattices shown here are much smaller than those used in the actual experiments. **A.** Mirror symmetric maps (Figure 3.4A). The lattice of this 10 by 5 SOM folded in the process of self-organization. The fold is perpendicular to the longer sides of the lattice. Notice that in general, connected output nodes are close in the input space. **B.** Glide reflection symmetric maps (Figure 3.4B, on the right). The lattice of this 11 by 4 SOM contains a diagonal fold, and the output nodes alongside the fold line (highlighted in blue and separated from the two maps along the z-axis) have been forced to move counterclockwise (clockwise) in the input space around the top (bottom) map. Many of the output nodes along the fold line that are adjacent in the lattice become widely separated in the input space (the width of lines connecting such nodes has been increased and they have been highlighted in red). **C.** Rotationally symmetric maps (Figure 3.4C). The lattice has been twisted in addition to being folded in the same manner as in **A.** In this case, it is adjacent output nodes along the fold line *and* close to the edges of the lattice whose distance in the input space increases disproportionately.

reflection symmetry is the visible expression of a diagonal fold in the lattice combined with a shearing motion along the fold (Figure 3.8B; on the right in Figure 3.8) which causes the entire diagonal inter-map boundary to be non-optimal. Rotation symmetry indicates a combination of a perpendicular fold and a twist of the lattice (Figure 3.8C) so that only the central region of the inter-map boundary retains its optimality while near the edges of the lattice, the boundary is non-optimal. The key point illustrated by Figure 3.8 is that in the latter two cases, output nodes that are close in the lattice become uncorrelated, that is, far removed from one other in the input space, rendering these two transitions suboptimal and hence, significantly lowering the M measurements of lattices that produced these two transitions.

The one-shot multi-winner SOM as a model of biological cortex is a substantial simplification of biological reality. Nevertheless the model reproduces a number of features of map formation in biological cortex: the topologically correct representation of the sensory surface by an individual map, the magnification of more often stimulated or more innervated regions of the sensory surface, the formation of multiple maps of the same sensory surface, and the often-observed mirror symmetry between adjacent maps. Experimentally determined map pairs in biological cortex like that in Figure 2.4II show that the maps often are not perfectly mirror symmetric. Whether these imperfections are analogous to the glide reflection symmetric maps in my model is unclear and deserves further investigation.

I also obtained results for which I was unable to find analogs in the experimental neuroscience literature. The occasional occurrence of rotationally symmetric map pairs in biological cortex, to my knowledge, never has been reported in experimental studies. However, the model predicts that, although such an event should be uncommon, map pairs in biological cortex may occasionally exhibit this type of symmetry.

Its relative rarity might be the reason why it has not been mentioned or observed, either because it was not encountered at all or because it was discarded as an anomaly outside the respective study's focus. What causes my model to form rotationally symmetric map pairs is unclear at present, and this issue certainly would deserve to be addressed by future research, in particular if this type of symmetry is found in biological cortex and linked to neurological disorders.

The clear preference of my model to orient adjacent maps of the sensory surface such that their common edge represented the most often stimulated sensory regions at first glance seems to contradict the results of neurophysiological mapping studies. For example, the two representations of the hand in areas 3b and 1, respectively, of the somatosensory cortex of the owl monkey (Merzenich et al., 1978), the squirrel monkey (Sur et al., 1982), and the macaque (Nelson et al., 1980) are consistently oriented in such a way that the palm is represented next to the inter-map boundary while the fingertips, which are relatively more innervated and arguably more often stimulated in the adult animal, are represented farthest from the boundary. In addition, across these three species, other body parts like the trunk or the thigh show no clear preference with respect to whether their most innervated or stimulated regions are represented next to or distant from the inter-map boundary. However, the adult patterns of innervation and stimulation are arguably very different from the patterns that persist during development at the time when cortical map formation occurs so that the above observations need not be inconsistent with my results. In fact, the model is consistent with a testable prediction: when adjacent mirror image topographic maps occur in neocortex, their common edge should represent the region of sensory surface that develops and innervates first (i.e., that has the most frequent stimuli initially during map development).

In addition, note that in my experiments with the non-uniformly distributed sensory stimuli, the size of the SOM's lattice was fixed and allowed the formation of only two maps of the sensory surface. With a larger lattice that forms more than two maps of the sensory surface, it is unavoidable that some adjacent maps are oriented so that the least often stimulated region is represented closest to the inter-map boundary. Given the results for just two maps, it is a reasonable hypothesis that the model would tend to minimize the number of such map pairs. For example, on a lattice that produces four individual maps, only the central pair of adjacent maps should represent the least often stimulated sensory surface region at the inter-map boundary. Four successive maps of the body surface have been reported in areas 3a, 3b, 1 and 2, respectively, of the somatosensory cortex of, for example, the macaque where probably all pairs of adjacent maps (3a and 3b, 3b and 1, 1 and 2) are mirror symmetric (mirror symmetry between 3a and 3b is likely, but has not been established thoroughly) (Nelson et al., 1980). So, with respect to the representation of the hand, it could be that in this particular case, the hand's most innervated and stimulated regions are in fact represented next to the inter-map boundary as often (twice) as is optimal in the case of four successive individual maps.

Adjacent maps of the same sensory surface in biological cortex often receive their inputs from different sources, for example, from different sets of sensors (Dykes and Ruest, 1984; Jones, 1984; Rakic et al., 1991), although there is a considerable overlap in some cases (Nelson et al., 1980). In my model, there exists only one source of inputs. However, even if adjacent cortical maps receive their inputs from different sources, these sources would often still be correlated, especially with respect to the part of the signal that conveys the relative positions of stimulations (for example, cutaneous and deep tissue pressure sensors). Provided the level of correlation between

the different input sources is sufficiently high, the net effect on map formation would be the same as with a single input source. So, in a sense, I essentially do model multiple, but with respect to stimulation location information, very highly correlated input sources.

Some investigators explain observations about the structure of biological cortex in terms of the effects of wiring optimization (e.g., Dehay et al. (1996); Cherniak (1995); Welker (1990); Young (1992)). The cortical areas that underly adjacent mirror symmetric cortical maps tend to be interconnected in a roughly topographic manner, that is, most connections are between roughly corresponding points in the two maps (van Essen et al., 1986; Stepniewska and Kaas, 1996; Roe and Ts'o, 1995). In addition, it has been observed that inter-map boundaries often coincide with cortical folds (Welker, 1990). If the goal is to interconnect two adjacent cortical areas whose boundary coincides with a cortical gyrus in a one-to-one fashion, then the best strategy in terms of minimizing total connection length is to connect the areas in a mirror symmetric fashion. So, one can argue that mirror symmetric adjacent cortical maps are a mere consequence of the mirror symmetric connectivity between adjacent cortical areas. However, the connectivity between any two adjacent cortical areas never follows a strict one-to-one pattern. Rather, each point in one area projects to a circumscribed target region in the other area so that there is considerable overlap between targets. This divergence of intra-cortical connections is thought to be even more pronounced during early development which is characterized by an excess of connections, many of which are later pruned. That raises the question of whether mirror symmetric connectivity between adjacent cortical areas and even cortical folds are the effects of the earlier formation of mirror symmetric maps in these areas and later pruning of connections by competitive Hebbian learning.

It is being acknowledged that extreme variability exists across different individuals of the same species with respect to their cortical maps (to the extent where no corresponding points can be established in the maps of different brains) (Dykes and Ruest, 1984). This variability is hardly ever discussed in mapping studies. Cortical maps also show a remarkable ability to reorganize quickly in response to changes in stimulation frequencies, denervation and stroke damage (Allman, 1984; Dykes and Ruest, 1984). Despite these indications favoring an explanation that is at least in part based on learning, topographic map formation in the brain is often thought to be brought about by genetically-mediated molecular gradients that are present during development (Grove and Tomomi, 2003; Levitt, 2000; Zhou and Black, 2000). The existence of multiple neighboring topographic maps of the same sensory surface is sometimes conjectured to have evolved due to genetic mutations (Allman and Kaas, 1971; Allman, 1984; Krubitzer, 1995), and it has been suggested that they may provide fitness advantages due to separation of spatial/temporal processing, parallel processing of different sensory attributes, minimization of connection distances, and other factors (Kaas, 1988; Cowey, 1981; Jones, 1990). However, there has been little speculation as to why such maps often exhibit reflection symmetry, and the relative contributions of activity-dependent versus activity-independent mechanisms remain the source of some debate, even for individual maps (Cohen-Cory, 2002; Grove and Tomomi, 2003). In my computational model, the formation of multiple pair-wise mirror symmetric topographic maps relied entirely on Hebbian learning and range-limited competitions for activation and learning. So, the necessity of genetic and evolutionary mechanisms in map formation is perhaps limited to the layout of the computational substrate ("hardware"), that is, the parcellation of cortex into regions/areas (Sur and Leamey, 2001) and the specification of very coarse afferent

connectivity, thereby determining high level properties like, for example, the overall number of maps of a particular sensory surface that will form. For the actual process of map formation, my model raises the possibility that activity-dependent mechanisms like competitive Hebbian learning significantly contribute to it, in particular having an influence on the relative orientation of adjacent maps and perhaps affecting the evolution of their genetically-guided afferent connectivity.

Chapter 4

Sequential Inputs

In this chapter two biologically-inspired features are added to the previously introduced one-shot multi-winner SOM. These two features, local lateral connectivity and temporally asymmetric Hebbian learning, provide the necessary additional computational power to process temporal input sequences with the one-shot multi-winner SOM. The specific temporal sequence processing task considered is the creation of a unique spatial representation for sizeable sets of temporal sequences. The first section gives an overview of past work on temporal sequence processing with SOMs and views it from the perspective of this work's novel approach which is detailed in the subsequent section. A presentation of the experimental results that were obtained when the one-shot multi-winner SOM was applied to the task of representing phoneme sequences corresponding to word pronunciations follows. These results and directions for future research are discussed in the final section.

4.1 Past Self-Organizing Maps for Sequence Processing

As noted in Chapter 2, the vast majority of past work on SOMs, as well as related neural network methods (Bishop et al., 1998), has involved static, i.e., time-invariant, input patterns where a network's activation pattern in response to one input is not

influenced by previous inputs. The results of these studies do not carry over directly to temporal sequences of inputs, a significant shortcoming given that sequential inputs are very common (e.g., language, motion in visual fields, movement feedback).

In response to this problem, several extensions to the basic SOM method have been proposed during the last decade to support temporal sequence processing. These extended SOMs are very diverse, so I consider them first in terms of the tasks they address and second in terms of the methodologies they adopt.

The specific temporal processing tasks that have been addressed include prediction, recall, recognition and representation. *Prediction* is concerned with the accurate computation of the next element in a sequence from previously observed sequence elements. In Principe et al. (1998), for example, a SOM was successful at predicting artificial chaotic time series as well as controlling a wind tunnel which required the prediction of wind speed changes. In Rao and Sejnowski (2000), a SOM-like network of two recurrently connected chains of neurons learned to predict the next in a series of left-to-right or right-to-left moving stimuli. The *recall* task takes prediction a step further, requiring that the SOM reproduce all elements of a sequence in the correct temporal order when given an initial cue, for example the first element of the sequence. This has been accomplished in Kopecz (1995) and Abbott and Blum (1996) with 2D fully laterally connected SOMs for one or two low-dimensional sequences. In Gerstner et al. (1993), a fully laterally connected network of 1000 nodes (not arranged according to any topology) was shown to be capable of storing and retrieving four sequences, and its theoretical capacity estimated at 100 sequences.

Recognition of temporal sequences has generally focused on identifying a given input sequence as a member of a class by mapping it onto a particular output lattice location or locations which correspond to class prototypes learned from previously

seen sequences. There have been many efforts to achieve this, such as Chappell and Taylor (1993); Euliano and Principe (1999); Kangas (1990); Somervuo (1999, 2003); Varsta et al. (1997); Wiemer (2003). Finally, and most directly related to my work, is the problem of transforming temporal sequences into relatively unique *spatial representations*, i.e., into relatively unique final activation patterns on the output lattice that represent the sequences and thus might be viewed as reminiscent of "cell assemblies" (Hebb, 1949). Such a time-to-space representation may be beneficial in data visualization and as an initial input processing step in a larger neural system for sequence recognition (Chappell and Taylor, 1993). To my knowledge, the only other study to address this task was that of James and Miikkulainen (1995), but several of the sequence recognition models above are also necessarily concerned about how the prototypes are arranged on the output lattice relative to one another.

These past temporal sequence processing SOMs can also be viewed from the perspective of the diverse methodologies they have proposed. The simplest approach has been just to leave the original one-shot single-winner SOM model untouched and to preprocess sequential inputs via an external short term memory. For example, in some studies a fixed number of successive input patterns were concatenated to form a single static pattern (Kangas, 1990). Others have suggested averaging the patterns in a sequence over time and feeding the average as a static pattern to the network (Carpinteiro, 1999). However, these approaches assume that the range of inter-pattern relations across time is quite limited. Many other forms of short term memory are reviewed elsewhere (Barreto et al., 2003; Mozer, 1993). Another approach has employed 'leaky-integrator' or other temporal neuron models as the output nodes (Chappell and Taylor, 1993; Varsta et al., 1997), while yet another idea has been to capture temporal relations in the input via lateral connections between

the output nodes, rendering the SOM a truly recurrent neural network (Abbott and Blum, 1996; Gerstner et al., 1993; Kopecz, 1995; Rao and Sejnowski, 2000). Finally, in Euliano and Principe (1999) and Wiemer (2003), spreading wavefronts of activation (activity diffusion) were used to alter the typical activation dynamics of the one-shot single-winner SOM so that learning is characteristically affected by the temporal order of the inputs.

4.2 Adding Temporally-Asymmetric Hebbian Learning to the One-Shot Multi-Winner SOM

At present there is no general consensus as to how best to process sequences with SOMs, and this topic remains a very active focus of current neurocomputational research (Barreto et al., 2003). In this context, the goal of the work described here was to extend one-shot multi-winner SOMs *in a biologically plausible way* to make them more effective in processing and representing large sets of variable-length sequences. Unlike most past related work described above, I focus solely on the task of developing a unique spatial representation for each of the input sequences, with the idea that this is also a precursor for effective pattern recognition.

To achieve this goal, I extended the one-shot multi-winner SOM described in the previous chapter as follows. As a mechanism for supporting sequence processing, I introduced into SOMs, for the first time to my knowledge, the use of temporally asymmetric Hebbian learning to train *local*, or range-limited, intra-lattice connections. These local lateral connections are very different from those used in past multi-winner SOMs: they are not used to produce a "Mexican Hat" pattern of lateral interactions and they are adaptive. Further, their adaptation is temporally asymmetric in a fashion inspired by recent experimental evidence showing that changes in biological synaptic

efficacy in cortex (Markram et al., 1997) and other structures of the brain (Bi and Poo, 2001, 1998; Zhang et al., 1998) are sometimes due to temporally asymmetric Hebbian learning: a synapse is strengthened (LTP) if pre-synaptic action potentials precede excitatory post-synaptic potentials by typically 20-50ms, and weakened (LTD) if the time course is reversed.

While a few past modeling studies have used temporally asymmetric Hebbian learning to store and retrieve sequences (Abbott and Blum, 1996; Gerstner et al., 1993; Rao and Sejnowski, 2000), these past studies were not concerned with either map formation or the transformation of sequences into spatial representations as I consider here. The model described here can be distinguished from that of James and Miikkulainen (1995) which successfully dealt with the representation task but did not use multi-winner SOMs, lateral connectivity, or temporally asymmetric Hebbian learning, as I do here. My approach is also very different from the pattern recognition system of Somervuo (1999) which, after initial training of a standard one-shot single-winner SOM on non-sequential inputs, uses an external construction algorithm to convert the SOM into a network with connections between arbitrarily-distant nodes (i.e., its lateral connections are neither local nor learned with temporally asymmetric Hebbian learning). In summary, the *fundamental hypothesis* examined in this chapter is that training a one-shot multi-winner SOM whose short-range lateral connections undergo temporally asymmetric Hebbian learning transforms variable-length temporal sequences into reasonably unique spatial patterns of activity, even while map formation of the input space persists.

While the sequence processing SOM described here is very general, to assess its functionality I used specific sequences of feature vectors. Each vector in a sequence is the feature-based encoding of an English phoneme. Each sequence corresponds to

the phonetic transcription, based on the NetTalk corpus (Sejnowski and Rosenberg, 1987), of a 2 to 10 phoneme noun naming an object, taken from the widely-used Snodgrass-Vanderwart word corpus (Snodgrass and Vanderwart, 1980). For example, /h ɛ l ə k a p t ɚ r/ is the phonetic sequence transcription of 'helicopter', and /p/, the fourth from last phoneme in the transcription, is equivalent to a distinct tuple of 34 binary feature values: [consonantal=1, vocalic=0, compact=0, diffuse=1, grave=1, acute=0, nasal=0, oral=1, tense=1, lax=0, ...]. See Appendix B for a complete set of phoneme encodings and further details about the input data. In this context, the SOM's task is the unsupervised acquisition of an internal representation for the 'spoken' names of a set of objects, the representation for each name ideally being unique.

Initially, before the first vector of an input sequence is presented to the SOM, all output nodes are inactive. From this initial state, the activation pattern develops deterministically at discrete time steps (one time step per input phoneme vector, hence "one-shot") based on the current input vector and the activation pattern at the previous time step. This implies that, for example, in the case of 'bow' (/b o/) and 'bowl' (/b o l/), after the input of /o/, the respective activation patterns are identical. For 'bow', this is the final activation pattern, and hence its spatial representation. According to my hypothesis, the last feature vector /l/ of 'bowl' should trigger a change in the activation pattern across the output lattice of the trained one-shot multi-winner SOM so that the spatial representation for 'bowl' differs from that of 'bow'.

Figure 4.1A shows the basic architecture of the one-shot multi-winner SOM for sequence processing. The recurrent network transforms an input sequence of patterns into a final single static output pattern (the sequence's spatial representation) where

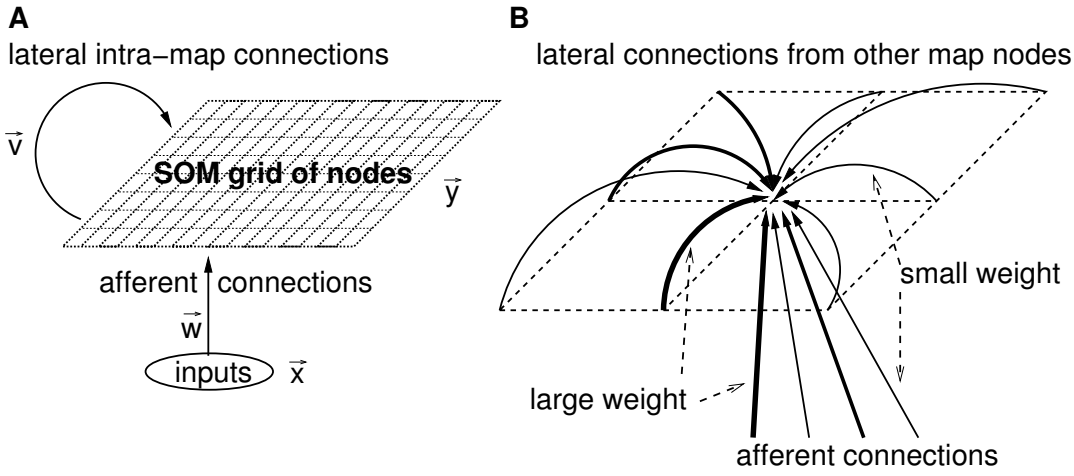


Figure 4.1: **A** The global architecture of the temporal sequence processing one-shot multi-winner SOM. **B** An output node with its weighted connections from the input layer and from the output nodes in its immediate 8-neighborhood. The different widths of the solid lines (arcs) indicates that in general the efficacies of the connections differ.

each component of the output corresponds to a node of the output lattice. The output lattice itself consists of a regular, rectangular grid of R rows by C columns of $Q = RC$ nodes. I again measure the *distance on the output lattice* between two output nodes i and i' at positions (r, c) and (r', c') using the box-distance metric, $\mathrm{d}(i, i') = \max(|r - r'|, |c - c'|)$.

In contrast to the one-shot multi-winner SOM of the previous chapter, the temporal sequence processing extension of it incorporates additional lateral intra-lattice connections between the nodes that form the output lattice. As illustrated in Figure 4.1B, an arbitrary output node i receives a connection from each node of the *input layer* as well as from each output node within a circumscribed *connection neighborhood*, $N_{\mathsf{conn}}(i) = \{j \mid \mathrm{d}(i, j) \leq r_{\mathsf{conn}}\}$, centered at and including i. Note that these

connections are *not* used to generate a Mexican Hat pattern of activation as was done, for example, in von der Malsburg's model of orientation sensitivity (von der Malsburg, 1973), but that they are entirely dedicated to temporal sequence processing. Every connection to i carries a real-valued synaptic weight indicating its efficacy. If the input layer consists of P nodes then the connection to the i^{th} output node from the j^{th} input is weighted by w_{ij}, and $\vec{w}_i \in \mathcal{R}^{+P}$ is the *afferent weight vector* of i. Analogously, the weights on the incoming lateral connections of output node i are stored in the *lateral weight vector* $\vec{v}_i \in \mathcal{R}^{+Q}$. In particular, v_{ij} corresponds to the weight on the lateral connection from j to i ($\mathrm{d}(i,j) \leq r_{\text{conn}}$), $v_{ii} = \beta \in \mathcal{R}$ is an immutable weight on the self-connection of i, and $v_{ij} = 0$ if i and j are not connected ($\mathrm{d}(i,j) > r_{\text{conn}}$).

The level of activation of an arbitrary input or output node ranges between 0 (inactive) and 1 (fully active). The activation levels of all P input nodes compose the afferent vector $\vec{x} \in [0,1]^P$, normalized to be of unit length. Similarly, the activation levels of all Q output nodes form a vector $\vec{y} \in [0,1]^Q$, the SOM's output or *map activation pattern*. For any output node i, only those components of $\vec{y}$ which correspond to activation levels of output nodes in i's connection neighborhood $N_{\text{conn}}(i)$ are directly relevant since i receives lateral connections only from those nodes. The activation levels of all nodes are updated synchronously at discrete time steps, *one step per input vector in a sequence*. Thus the afferent input vector as well as the map activation pattern are time-variant.

Given an input sequence $X = \vec{x}(1), \ldots, \vec{x}(k)$, the map activation pattern, initialized as $\vec{y}(0) = \vec{0}$, evolves over a period of k time steps. The final map activation pattern $\vec{y}(k)$ is then said to be the *spatial representation* of temporal sequence X. At the beginning of time step $t \geq 1$, the *net input* h is computed independently for

each output node i as:

$$h_i(t) = \alpha \vec{w}_i^T \vec{x}(t) + (1 - \alpha) \vec{v}_i^T \vec{y}(t-1) \tag{4.1}$$

where fixed parameter $\alpha \in [0, 1]$ determines the relative contributions of afferent versus lateral input vectors, and T indicates the transpose of column vectors $\vec{w}_i$ and $\vec{v}_i$.

To compute $\vec{y}(t)$ from $\vec{h}(t)$, a computationally efficient one-shot mechanism is used which approximates the competitive activation dynamics (Mexican Hat pattern) that has been implemented in some past iterative multi-winner SOMs via a computationally-expensive numerical solution of differential equations (Reggia et al., 1992). However, unlike with traditional one-shot single-winner SOMs, multiple winners occur: every output node i which receives a net input greater than that of all other output nodes within i's connection neighborhood is taken to be a winner. Since parameter r_{conn} is usually chosen to be small relative to the size of the entire output lattice, typically multiple winner nodes exist. Each winner is then made the center of a 'peak' of activation. The distribution of activation within a single peak is such that winner node i at its center is maximally active ($y_i = 1$), and the activation level of each non-winner node j within i's connection neighborhood decreases exponentially with increasing distance between j and i. The activation peak centered at i does not extend beyond the connection neighborhood of i. However, two or more peaks may partially overlap, in which case their contributions to the activation level of an output node in the region of overlap are added, but may not exceed 1. Specifically, and similar to what was done in the previous chapter, if the set $V(t)$ of winner nodes at time t is:

$$V(t) \;=\; \{i \mid \forall j \neq i : j \in N_{\text{conn}}(i) \Rightarrow h_j(t) < h_i(t)\} \tag{4.2}$$

then the activation of output node j is:

$$y_j(t) \;=\; \min\left(1,\ \sum_{i\in V(t)} \begin{cases} \gamma^{\mathrm{d}(i,j)} & \text{if } j \in N_{\mathsf{conn}}(i) \\ 0 & \text{otherwise} \end{cases}\right) \tag{4.3}$$

where $\gamma \in [0,1]$ determines the shape of each peak of activation (higher γ means slower drop off).

To test my central hypothesis, namely that my model learns to spatially represent the sequences in the training set fairly uniquely, I use the 1-norm to quantify the difference between two final activation patterns $\vec{y}$ and $\vec{y}'$: $d(\vec{y},\vec{y}') = ||\vec{y}-\vec{y}'||_1 = \sum_{i=1}^{Q}|y_i - y_i'|$. Using distance measure d, I assess the overall performance of my model by measuring over all distinct sequences X (of length k) and X' (of length k') from the training set, the distance between the spatial representations of X and X'. I use three quantitative measures of how the model performs overall in separating different sequences into unique final spatial representations. First, I count the number of pairs of distinct sequences in the training set for which the model ends up with the same final map activation pattern: $|Z| = |\{\{X,X'\} : X \neq X', d(\vec{y}_X(k),\vec{y}_{X'}(k')) = 0\}|$. The model uniquely represents all sequences if $|Z| = 0$, otherwise there are pairs of distinct sequences which the model 'confuses'. The second measure is the minimum distance d_{min} between two spatial representations computed over all pairs of distinct sequences in the training set: $d_{\mathsf{min}} = \min_{X\neq X'} d(\vec{y}_X(k),\vec{y}_{X'}(k'))$. Notice that $d_{\mathsf{min}} = 0$ for as long as $|Z| > 0$ and $|Z| = 0$ as soon as $d_{\mathsf{min}} > 0$, and that d_{min} and $|Z|$ are often complementary, not redundant. Training could, for example, significantly increase d_{min} from a pre-training value already greater than zero, while $|Z|$ remains 0. Or training may decrease $|Z|$ to a smaller value still greater than zero, while d_{min} remains 0. Finally, I measure the average distance between two spatial representations computed over all pairs of distinct sequences in the training

set: $\bar{d} = \frac{1}{|S|} \sum_{\{X,X'\},X \neq X'} d(\vec{y}_X(k), \vec{y}_{X'}(k'))$, where $|S|$ is the number of sequences in the training set.

Before training, each weight is independently initialized with a random value from the interval $[0, 1]$, each afferent weight vector is normalized to unit length, and each lateral weight vector is normalized such that $\forall i : \sum_{j \neq i} v_{ij} = 1$. The one-shot multi-winner SOM learns by adjusting its weights in response to the repeated input of all temporal sequences in the training set in random order. The number of training epochs is 1000. The input of a single arbitrary temporal sequence of length k causes the SOM to pass through k time steps. At the end of each time step t, after the construction of activation pattern $\vec{y}(t)$, the weights of the SOM are modified.

For the afferent weight vector $\vec{w}_i$ of the i^{th} output node, the learning rule is:

$$\vec{w}_i(t) = \vec{w}_i(t-1) + \mu y_i(t) \vec{x}(t) \tag{4.4}$$

$$\vec{w}_i(t) = \vec{w}_i(t) / ||\vec{w}_i(t)||_2 \tag{4.5}$$

Eq. 4.4 implements typical temporally symmetric Hebbian (or competitive) learning where $\mu \in (0, 1]$ is the *afferent learning rate*. Renormalization in Eq. 4.5 restricts $\vec{w}_i$ to move across the surface of the unit hypersphere, generally in the direction of the current afferent input $\vec{x}(t)$. In contrast, the learning rule for the lateral weights is unusual in being a temporally *asymmetric* variant of Hebbian learning. As noted earlier, recent experimental studies have found this learning rule to sometimes govern changes in synaptic efficacy in cortex (Markram et al., 1997) and other parts of the brain (Bi and Poo, 2001, 1998; Zhang et al., 1998). Given two output nodes i and j where $0 < \mathrm{d}(i, j) \leq r_{\text{conn}}$, the efficacy of the connection v_{ij} to i from j at time t is increased proportional to $y_j(t-1)$, the activity of j *at the previous time step*, times $\max(0, y_i(t) - y_i(t-1))$, the increase in the activity of i *relative to the previous time*

step:

$$v_{ij}(t) = \begin{cases} v_{ij}(t-1) + \ldots & \text{if } j \neq i \text{ and} \ldots \\ \ldots + \eta y_j(t-1)\max(0, y_i(t) - y_i(t-1)) & \ldots j \in N_{\text{conn}}(i) \\ v_{ij}(t-1) & \text{otherwise} \end{cases} \quad (4.6)$$

$$v_{ij}(t) = v_{ij}(t) / \sum\nolimits_{j \neq i} v_{ij}(t) \quad \text{for } i \neq j \quad (4.7)$$

where $\eta \in (0, 1]$ is the *lateral learning rate*. In general, Eq. 4.6 is intended to capture a notion of cause and effect across time: the connection to i from j is strengthened if j's preceding activity contributes to an increase in the activation of i. This is consistent with the results of a previous analysis of temporally asymmetric Hebbian learning which concludes that overall change in the synaptic efficacy is proportional to the rate of change in post-synaptic activity (Roberts, 1999). Note that the subsequent renormalization may result in a net decrease of a connection's efficacy due to competition for 'growth' with the other incoming lateral connections of i. This relates to previous observations that temporally asymmetric Hebbian learning is inherently competitive and self-stabilizing due to a balance of weight increases (LTP) and decreases (LTD) (Song et al., 2000; Royer and Pare, 2003). I use a simple method, that is, explicit renormalization, to implement such competition and stability.

As is typical for the training of traditional one-shot single-winner SOMs, the values of certain parameters in the learning rule depend on how far training has progressed. For example, training of one-shot single-winner SOMs is often divided into two phases: a rough ordering/self-organization phase corresponding to large values for γ and μ, and a convergent phase corresponding to small values for γ and μ. Analogously for the sequence processing one-shot multi-winner SOM, and much in the same way it was done for the one-shot multi-winner SOM of the previous chapter, parameters γ,

μ and η monotonically decrease in a non-linear fashion from some initial value to a smaller final value. For example, $\gamma(t) = \gamma_{\text{fin}} + (\gamma_{\text{init}} - \gamma_{\text{fin}})/(1 + e^{(t-\gamma_{\text{infl}})/\gamma_\sigma})$ where t is the fraction of completed training epochs, γ_{init} (γ_{fin}) determines γ's initial (final) value, γ_{infl} is the point of inflection, and γ_σ determines the rate of decline. Similar functions are used for μ and η.

4.3 Results of Using the Model to Learn Temporal Sequence Representations

I first demonstrate that, with a suitable set of parameters, training of the sequence processing one-shot multi-winner SOM improves the uniqueness of the transformation of sequences into spatial representations. Next, the changes in model performance are measured as the size of the training set is systematically varied, and the formation of feature maps and patterns in the lateral connectivity of the network are examined. Finally, the difference between the spatial representations of any two distinct sequences in the trained model is examined and found to relate to the similarity or dissimilarity of the two sequences.

4.3.1 Learning Unique Representations

An initial exploration of the parameter space using particle swarm optimization methods (Kennedy et al., 2001) readily established a value for each of the model parameters (see top of Table 4.1) such that training significantly improves the performance of the model across all three performance (uniqueness) measures. The great extent of the parameter search space and the computational cost of network training (the latter, due to the additional lateral connections, being even greater than in the previous chapter; see Section 3.1) forces one to limit both the size of the output lattice

Table 4.1: Best Parameter Set for the One-Shot Multi-Winner SOM

Parameter Set

r_{conn}	α	β	γ_{init}	γ_{fin}	γ_{infl}	γ_σ	μ_{init}	μ_{fin}	μ_{infl}	μ_σ	η_{init}	η_{fin}	η_{infl}	η_σ
4	.64	.05	.37	0	.2	.16	.44	0	.4	.0001	.62	0	.8	.04

Performance: 30 by 20 nodes, 60 training sequences

	$\bar{d}$	d_{min}	$\lvert Z\rvert$
pre-training	16.5 (1.003)	0 (0)	11.8 (4.27)
post-training	23.1 (0.998)	0.4 (0.88)	1.4 (1.27)

Performance: 40 by 30 nodes, 175 training sequences

	$\bar{d}$	d_{min}	$\lvert Z\rvert$
pre-training	31.1 (1.10)	0 (0)	58.9 (9.55)
post-training	45.1 (1.18)	0 (0)	11.0 (6.58)

and the training set to restrain the time needed to train the network repeatedly. The initial experiments were done with a 30 by 20 node network and 60 randomly chosen distinct sequences. I initially expected that strongly self-inhibitory output nodes (β strongly negative) and a very limited influence of the afferent inputs on the activation dynamics (small α_{fin}) would be critical to the formation of unique spatial representations. However, both the particle swarm optimization algorithm and a systematic manual trial-and-error exploration of the permissible range of values determined that self-inhibition was not optimal and that a much higher value (0.64) for α_{fin}, corresponding to a much stronger influence of the afferent input, produces better performance.

The performance results in the middle of Table 4.1 give the means and standard deviations (in parentheses) for the three performance measures, each averaged over

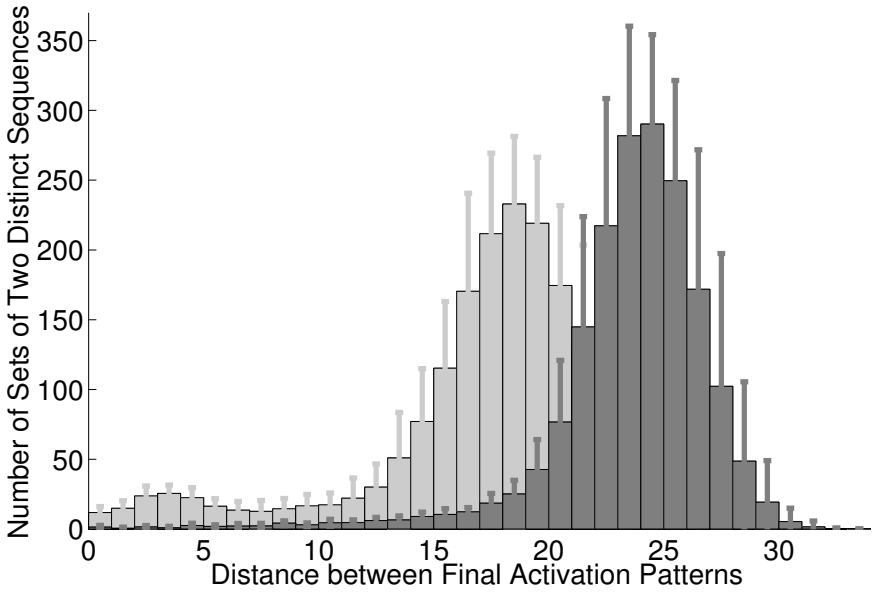


Figure 4.2: Pre-training (light gray) and post-training (dark gray) histograms of the distances between the spatial representations for every two distinct sequences from the training set containing 60 sequences. The histogram counts have been averaged over 20 independent experiments. The error bars represent one standard deviation. The shape of the obscured right tail of the pre-training histogram resembles the right tail of the post-training histogram.

20 independent experiments, obtained with the listed parameter values and the 30 by 20 node network. An independent experiment constitutes initializing the model using different random initial weights, measuring pre-training performance, training the model, and measuring post-training performance. Figure 4.2 shows that training significantly reduces the number of sequences that the model transforms into identical or very similar spatial representations. Training also increases the overall distance between the spatial representations of two distinct sequences in general, indicated in the figure by a post-training right-shift of the distance distribution's peak

in conjunction with the reduction of the distribution's left tail. As an example of how the one-shot multi-winner SOM learns to better separate two distinct but similar sequences, consider the two sequences /b ɔ l/ (ball) and /b o l/ (bowl). For the model, they are relatively hard to distinguish, since they are short and differ only in the intermediate similar phoneme. Nevertheless, on average, the distance d between the two sequences' final activation patterns increases from 3.0, prior to training, to 10.0, after training. This is illustrated in Figure 4.3 for a single representative experiment. The figure shows pre-training (top) and post-training (bottom) plots of the development over time of the activation pattern that the winner nodes form on the output lattice at each time step. Time proceeds along the horizontal axis, and the pattern of winner nodes ('•' for ball, 'x' for bowl) is shown after input of each phoneme on a grid that represents the output lattice. The initial pattern of winner nodes are identical for both sequences after seeing just the first identical phoneme, but the patterns diverge at subsequent time steps, leading to different final patterns which are shown on the right-most grid, even though the final afferent inputs to the network are identical (the phoneme /l/). The lines (dotted for 'ball', dashed for 'bowl') in this figure that sometimes connect winner nodes of subsequent time steps can be viewed as causal relationships: a winner node is connected to all nodes in its connected neighborhood that won at the previous time step, *only if* the node would not have won at the current time step without the lateral input from these previous winners.

Several qualitative differences between the pre-training and post-training activation dynamics of the model were observable in general. First, the total number of winner nodes on the output lattice tended to increase after training, indicating a more efficient use of the space that is available on the output lattice to provide an

encoding (not observable in Figure 4.3). Second, the fraction of winner nodes that win because of the lateral input from previous winners increases: after training, the lateral connectivity of the model apparently influences the activation dynamics much more strongly. Finally, and most importantly, the distance between the final activation patterns of two distinct sequences usually tends to increase. This suggests that the temporally-asymmetric training of the model's lateral connections is the major cause of the overall increase in uniqueness of the sequences' spatial representations. That this latter result is quite general, and much more dramatic for more different sequences than the two rather similar ones illustrated here, is seen in Figure 4.2.

Figure 4.3 (next page): Pre-training (**A**) and post-training (**B**) traces of winner nodes for the sequences /b ɔ l/ (ball; '•' and dotted lines) and /b o l/ (bowl; 'x' and dashed lines). If a winner node is connected by lines to winner nodes at the previous time step, then the node would not have won without the input it received from the previous winners. The final winner nodes or, equivalently, activation patterns (rightmost grid) are almost identical prior to training (**A**), but substantially more distinct after training (**B**). After training, more winners depend on the input they receive from previous winners via lateral connections (many more lines in **B**), indicating that the lateral connectivity much more strongly influences the activation dynamics after training, and is thus likely to be a major cause of the more unique final activation patterns.

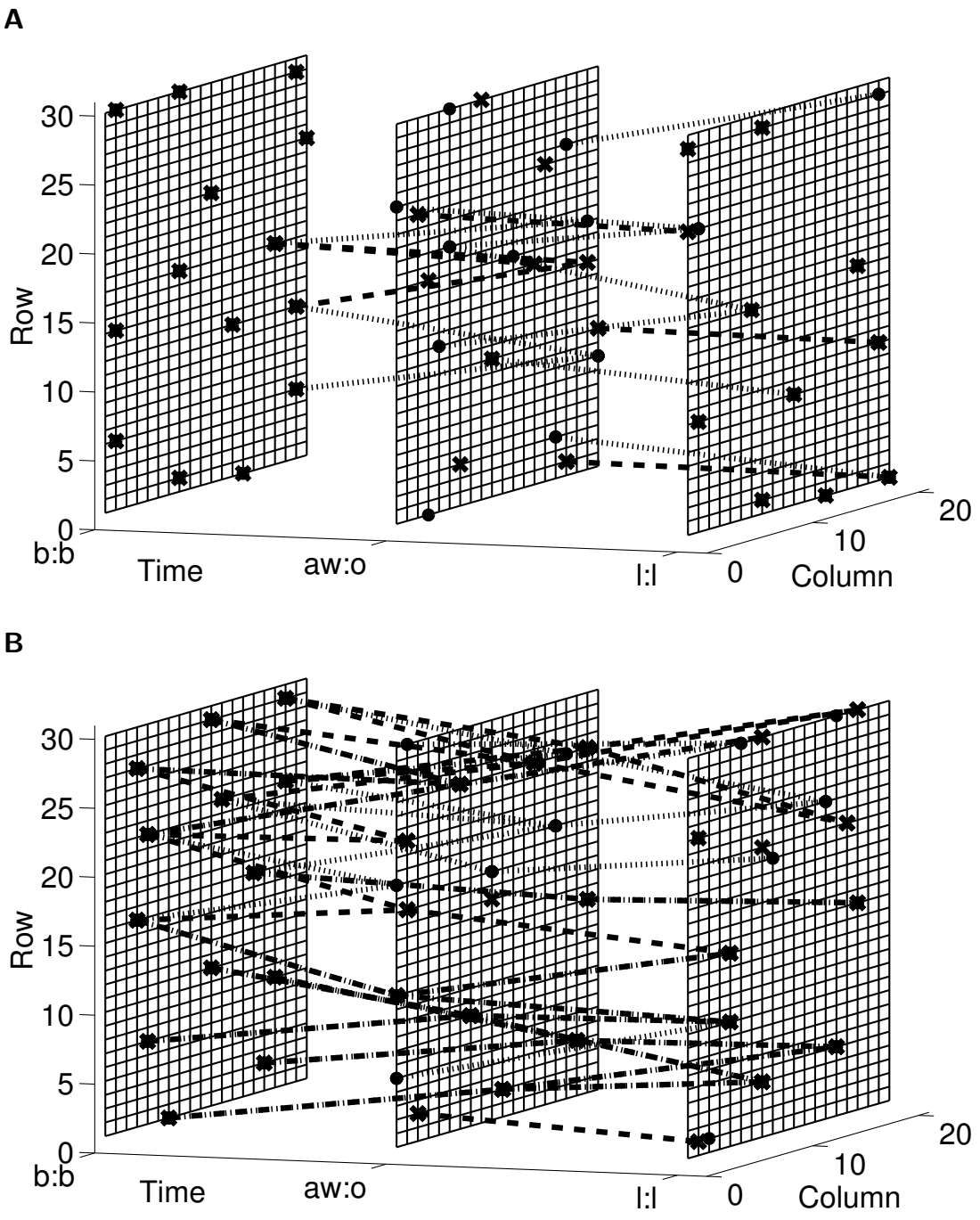


Figure 4.3: Caption on previous page

Despite a significant increase in performance, the relatively small networks used meant that all unique representations were learned in only 4 out of the 20 independent experiments conducted with 60 sequences. In the other 16 experiments, a total of only seven pairs of sequences were transformed into the same spatial representation after training. These pairs either ended in two successive consonants (horse/box, needle/eagle, iron/corn), shared a relatively long common suffix (sweater/helicopter, iron/corn), or both started with /k/ and ended with similar consonants (cup/couch, cup/coat, couch/coat).

A systematic exploration of varying parameters one at a time showed that the model's good performance was relatively insensitive to significant parameter changes. However, for parameter α which controls the relative influences of afferent and lateral inputs, I found only a narrow range of values $0.6 \leq \alpha \leq 0.7$ for which $|Z|$ reaches nearly minimal values and d_{min} exceeds zero. Unlike the other parameters, the choice of α thus appears critical to optimal performance.

I conducted seven independent experiments with a larger 40 by 30 version of the one-shot multi-winner SOM that was trained using the same parameter values with the complete set of 175 distinct sequences; the results are shown at the bottom of Table 4.1. A comparison of the performance values for the two different network sizes in Table 4.1 shows that the twice-as-large model trained with roughly three times the number of distinct sequences performed approximately the same. Training of the larger model increased $\bar{d}$ by 50%, did not increase d_{min} and decreased $|Z|$ by 84%, which compares to an increase of $\bar{d}$ by 40%, a very small increase of d_{min} and a decrease of $|Z|$ by 88% for the smaller model. The total number, over all experiments, of pairs of confused but distinct sequences increases by roughly a factor of six from seven for the smaller model trained with 60 sequences to 43 for the

larger model trained with 175 sequences (see Appendix C, Table C.1, for a list of all pairs of confused sequences). Note, however, that the total number of pairs of distinct sequences increases by roughly a factor of 8.6 from 1,770 to 15,225. Some representative examples of sequence pairs confused by the larger network model in different runs are: bell/ball, tiger/spider, tomato/potato, fly/butterfly, bottle/beetle, box/ox, and sweater/tiger.

4.3.2 Effects of Memory Load on Performance

Memory load as used here refers to the number of sequences in the training set in relation to the number of output nodes. To assess memory load effects on the performance of the smaller 30 by 20 network, using the parameter values from Table 4.1, the number of sequences in the training set was varied from 10 to 175. All training sets, except the set containing all 175 sequences, were generated by the successive subtraction (addition) of 10 randomly chosen distinct sequences from (to) the training set containing 60 sequences that I used in Section 4.3.1.

Figure 4.4 shows that after training the three performance measures for the one-shot multi-winner SOM react differently to variations in the size of the training set, measured as the number of pairs of distinct sequences that can be formed from sequences in the training set. The mean distance between the spatial representations of any two distinct sequences, $\bar{d}$ (d_{mean}), remains roughly constant over the entire range of tested training set sizes. The average minimum representation distance, d_{min}, quickly drops to zero, so with this small network there are often a few confused sequence pairs present. The data suggests an overall near-linear dependency of $|Z|$ on the number of pairs of distinct training sequences, or a roughly quadratic increase of $|Z|$ in terms of the number of training sequences.

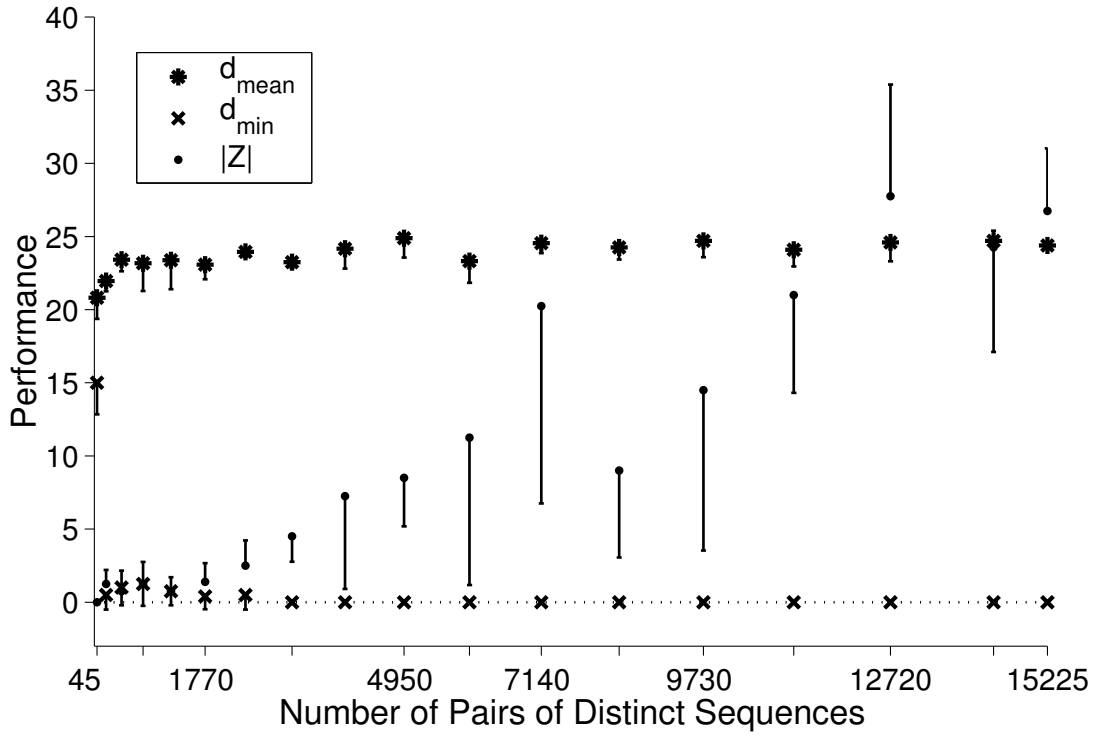


Figure 4.4: The influence of training set size on the three performance measures $d_{\text{mean}} = \bar{d}$, d_{min} and $|Z|$. For each training set size, four independent experiments were conducted, except for the set of size 60 for which the number of independent experiments was 20. Shown are performance means, each with an error bar of one estimated standard deviation (its direction is chosen to minimize clutter and is irrelevant otherwise). While $d_{\text{mean}} = \bar{d}$ remains constant, d_{min} very quickly drops to zero. $|Z|$ increases proportionally to the number of pairs of distinct sequences.

4.3.3 Map Formation

Past afferent input vectors have substantial influence, via recurrent lateral connections, on the activation dynamics of the one-shot multi-winner SOM during the processing of a sequence. This effect is non-existent in more typical one-shot single-winner SOMs which, when trained on a set of unsequenced input vectors, form maps where similar inputs (inputs that are close to one another in input space) are mapped onto or represented by nodes that are close to one another on the output lattice, and vice versa. I assumed that the 'reverberation' of past inputs in the network would disturb and prevent the formation of feature maps of the input phonemes. This assumption turned out to be incorrect.

As can be seen in Figure 4.5, which is a representative example, feature maps of single phonemes, that is, single input feature vectors, did form. For example, the model clearly separated clusters of vowels from consonants. These are the two top-level categories any reasonable clustering algorithm would identify because they are most dissimilar based on the set of feature vectors. Unlike in traditional single-winner SOMs, the maps formed by the one-shot multi-winner SOM exhibit substantial redundancy. For example, in Figure 4.5, twelve isolated clusters of vowels are visible where the clusters are similar to one another in terms of which particular vowels they contain. These redundant clusters arise due to the distributed representation used, and are reminiscent of the multiple redundant clusters sometimes seen in biological sensorimotor cortex (see, for example, Donoghue et al. (1992); Georgopoulos et al. (1988)). Internally clusters are more homogeneous than at their periphery, that is, nodes in the center of a cluster are more similar to their immediate neighbors (lighter cells) than nodes on the periphery of a cluster (darker cells). The same applies to the areas of the output lattice that have become sensitive to consonants. The result is

a multitude of small non-redundant feature maps with no discernible boundaries between them that form one globally redundant map. Surprisingly, qualitatively similar maps form for α-values as low as 0.2, that is, even if the afferent inputs have very little influence on the activation dynamics of the model.

Figure 4.5 (next page): The bottom half of a trained one-shot multi-winner SOM measuring 40 by 30 nodes (175 distinct sequences; network parameters as in Table 4.1). Each cell is one node that is labeled by the phoneme (white characters for vowels, dark characters for consonants) whose feature vector is closest to the node's afferent weight vector. Vowels have been separated on the map from consonants based on their phoneme feature vectors. Multiple vowel clusters or 'islands' can be seen at different locations in a 'sea' of consonants. A cell's background brightness corresponds to the average dot-product between the afferent weight vectors of the node it represents and that node's immediate neighbors: the brighter the node the more similar its input sensitivity is to that of its immediate neighbors. Lateral connections with a weight > 0.2 are shown as arrows pointing from the source toward the destination node. The length of an arrow is proportional to the square root of the connection weight. The distance of the destination node equals the number of concentric arcs at the arrow's base. The arrow is black if it points from a vowel to a consonant or vice versa; it is white if it points from a vowel (consonant) to a vowel (consonant). The pattern of strong lateral connections suggests that they represent frequent phoneme transitions in the training sequences. In the training sequences, a vowel is almost always followed by a consonant, and in the output lattice, most connections originating at "vowel nodes" indeed terminate at "consonant nodes" (black arrows).

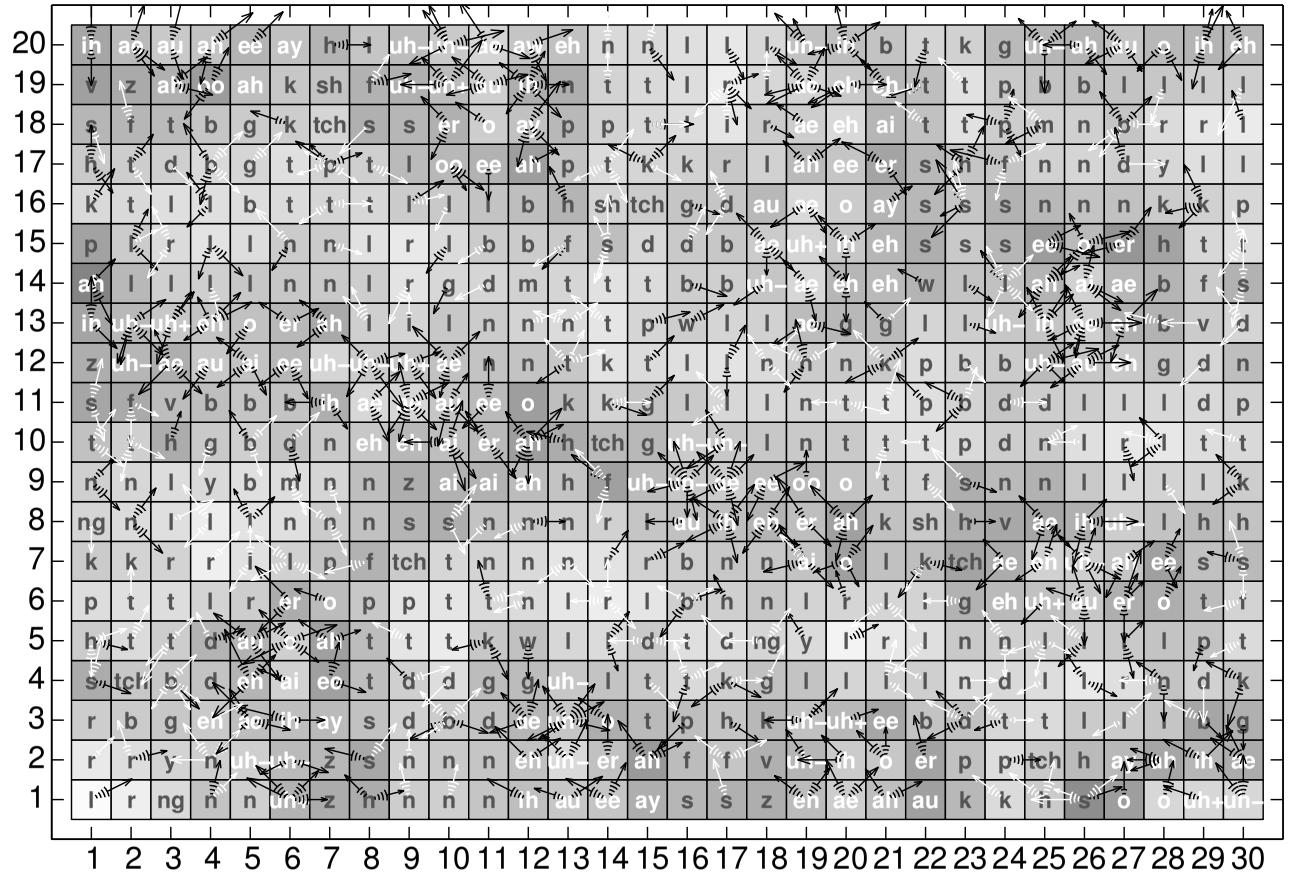


Figure 4.5: Caption on previous page

Figure 4.5 also shows all lateral connections whose weights have increased significantly during training[1]. A visual inspection suggests that nodes sensitive to vowels tend to send strong connections to nodes sensitive to consonants. It is one of the properties of the training set that in all but three cases (out of a total of 222), a vowel in a sequence is followed by a consonant. This gives rise to the hypothesis that strong lateral connections coincide with the frequent transition from a particular input phoneme to a particular next input phoneme in the sequences of the training set. To test this, I measured the correlation between lateral connection weights and phoneme transition frequencies. I recorded, for each possible input phoneme transition $\vec{x}(t)$ to $\vec{x}(t+1)$, the sum of the weights on all lateral connections from a node i to a node j ($i \neq j$) where $\vec{x}(t)$ maximizes $\vec{w}_i^T \vec{x}(t)$ and $\vec{x}(t+1)$ maximizes $\vec{w}_j^T \vec{x}(t+1)$. Each greater-than-zero sum[2] was then paired with the absolute frequency with which the respective phoneme transition $\vec{x}(t), \vec{x}(t+1)$ occurs in the training sequences. These pairs are the data points from which the correlation coefficient is computed. This was done repeatedly and independently for both a small (30 by 20 nodes trained with 60 distinct sequences; 20 independent experiments) and a large (40 by 30 nodes trained with 175 distinct sequences; 7 independent experiments) SOM output lattices, prior to and after training. Table 4.2 summarizes the results by providing the mean (and standard deviation) of each correlation coefficient, as estimated from the results of the respective number of independent experiments. Prior to training, the then random lateral connection weights are not correlated with phoneme transition frequencies. After training, the two quantities are very highly positively correlated,

[1]The threshold is 0.2. Prior to training for an output lattice of 40 by 30 (30 by 20) nodes, the mean lateral weight is 0.0008 (0.0017) with a standard deviation of 0.0040 (0.0058).

[2]Lateral connections with a weight equal to zero are considered non-existent. Hence, sums equal to zero are excluded from the analysis.

Table 4.2: Correlation between Lateral Weight Magnitude and Phoneme Transition Frequency

training set & model size →	60 sequences, 30x20 nodes	175 sequences, 40x30 nodes
pre-training	-0.0664 (0.0165)	-0.0536 (0.0141)
post-training	0.6939 (0.0277)	0.6263 (0.0311)

lending strong support to the hypothesis that strong lateral connections coincide with the frequent phoneme transitions.

4.3.4 Representation Distance and Sequence Similarity

I now consider the question of whether or not similar input sequences are transformed into similar spatial representations, that is, final map activation patterns. To measure the similarity of spatial representations, the 1-norm distance d that I have used all along plus a *winner separation distance* are both used. Recall that, at the end of training when the parameter γ determining activation peak widths approaches zero (see Eq. 4.3 and γ_{fin} in Table 4.1), only the winner nodes are significantly (and fully) active. Let the (row, column) positions of the winner nodes in the spatial representation $\vec{y}$ following the final phoneme of one input sequence be $(r_1, c_1), (r_2, c_2), \ldots, (r_k, c_k)$, and the positions in $\vec{y}'$ for a different input sequence be $(r'_1, c'_1), \ldots, (r'_{k'}, c'_{k'})$, and without loss of generality take $k \geq k'$. I then define the winner separation distance d_{sep} between $\vec{y}$ and $\vec{y}'$ to be the average distance on the output lattice from a winner node in $\vec{y}$ to the closest winner node in $\vec{y}'$: $d_{\text{sep}} = \frac{1}{k} \sum_{i=1}^{k} \min_{1 \leq j \leq k'} (|r_i - r'_j| + |c_i - c'_j|)$.

For comparison purposes, I also need a measure or measures of the similarity of any two input sequences of phonemes used for training. In general terms, the

similarity (dissimilarity) of two sequences is typically measured in terms of the two sequences' optimal alignment or "edit distance", and so I adopt this method here. The algorithm for computing optimal alignment is described in detail, for example, in Gusfield (1997). In short, an alignment of two sequences is a recipe for translating one sequence into the other using essentially two operations: the insertion of a special 'blank' element into a sequence and the substitution of an element in one sequence with an element at the same position in the other sequence. Each substitution operation in an alignment is associated with a cost or score which is a function of the two elements being substituted. The sum over all substitutions in an alignment is the score (cost) of the alignment. An optimal alignment maximizes (minimizes) the score (cost) of translating one sequence into the other. To avoid length-based biases, I normalize the score (cost) of each optimal alignment by its length. I adopt the convention that each inserted blank equals the blank's immediate predecessor in the sequence. As all input elements are binary-valued feature vectors, I adopt a Hamming distance cost measure (very similar results were also found using Euclidean distances instead). As a score measure, I use the Tversky feature count (Tversky, 1977; Tversky and Gati, 1978), a well established method in linguistics for measuring the similarity of phonemes. With this latter measure, if two phonemes are encoded by the feature vectors $\vec{x}$ and $\vec{x}'$, then their similarity equals the difference between the number of features they share and the number of features they do not share: $|\{i : x_i = x'_i = 1\}| - |\{i : x_i = 1, x'_i = 0\}| - |\{i : x_i = 0, x'_i = 1\}|$.

The correlation analysis was performed for each of the four possible combinations of a representation distance measure (d or d_{sep}) compared to a sequence similarity (Tversky) or dissimilarity (Hamming) measure. Two differently size versions of the model were used (30 by 20 nodes and 60 distinct sequences versus 40 by 30 nodes

Table 4.3: Correlation between Representation Distance and Sequence Similarity

training set & model size →	60 sequences, 30x20 nodes				175 sequences, 40x30 nodes			
pattern distance measure →	1-norm		winner separation		1-norm		winner separation	
pre- vs. post-training →	pre	post	pre	post	pre	post	pre	post
feature vector distance or similarity measure								
Hamming distance	*.3284*	*.2893*	**.3041**	**.3779**	.3874	.3594	.3863	.4141
Tversky feature count	*-.3438*	*-.2917*	**-.3214**	**-.4054**	-.3950	-.3641	**-.3946**	**-.4245**

and 175 distinct sequences). The small (large) instance of the model was initialized 20 (7) times with different random initial weights and subsequently trained. In each of these independent experiments, the four correlation coefficients were computed prior to and after training.

The correlation coefficients, averaged over the respective number of independent experiments, are listed in Table 4.3. Overall, these results show that both before and after training, there is a substantial positive correlation between input sequence

Hamming distances and their final activation pattern distances, and a substantial negative correlation between input sequence similarities (Tversky's measure) and their final activation pattern distances. Most intriguing is that the magnitudes of the correlation measured in terms of winner node separation are always increased by training. The bold (italic) table entries indicate a statistically significant ($p \leq 0.01$) increase (decrease) of the post-training relative to the pre-training absolute level of correlation.

4.4 Discussion

Most past work on self-organizing maps (SOMs) has focused on processing non-sequential input patterns and has used Kohonen's one-shot single-winner approach to map formation. As noted earlier, the latter bases learning on a single global winner node for each input pattern, and uses a one-shot "best match" winner selection process for computational efficiency. While very successful for the non-sequential tasks for which it is intended (Kaski et al., 1998a), various past approaches extending such SOMs have been and continue to be developed to process temporal sequences because of the importance of this issue (see Section 4.1).

In this chapter, I have examined the specific question of the extent to which the one-shot multi-winner SOM introduced in the previous chapter can be modified to learn a unique spatial representation or encoding of temporal sequences while still retaining traditional map formation properties. Two factors seem to be very important in facilitating sequence processing with SOMs, both being biologically plausible. First, instead of the global single-winner activation dynamics of more traditional Kohonen-style SOMs, I used multiple simultaneous winner nodes. Such a distributed or coarse representation is motivated by its potential to encode/represent a larger

number of temporal sequences. Using multiple local activation peaks like this is also more consistent with activity patterns in the cerebral cortex, and for this reason has been adopted in several past SOMs directed at modeling neurobiological observations (Bednar and Miikkulainen, 2000; Cho and Reggia, 1994; Li, 2002; Pearson et al., 1987; Reggia et al., 2001; Sutton et al., 1994; von der Malsburg, 1973). However, unlike these past studies with non-sequence processing tasks, I retained the one-shot winner selection of Kohonen SOMs for computational efficiency.

The second enhancement to traditional SOMs was to add local intra-lattice lateral connections that undergo temporally asymmetric Hebbian learning. The motivation for this type of connections was to enable the now recurrent network to capture temporal transitions via lateral shifts in activation peak locations. As discussed above, this extension also derives from biological data that has demonstrated such temporally asymmetric learning experimentally (Bi and Poo, 2001, 1998; Markram et al., 1997; Zhang et al., 1998). My learning rule (Eqs. 4.6, 4.7) intuitively tries to capture and enhance the causal relationships between activation peaks at one time instant and subsequent nearby activation peaks at the next time instance.

With these two extensions, the resulting sequence processing one-shot multi-winner SOM was found to be remarkably effective in developing unique spatial representatives (unique final activation patterns across the output lattice) for sizable sets of real-world temporal sequences. Even with the relatively small networks I used, maps could learn unique encodings for almost all 60 sequences, or 175 sequences for the somewhat larger maps. While not perfect (typically a very few sequences remained confused after training), the learning process clearly and consistently increased the uniqueness of representations over time. As similar input sequences tended to produce similar final activation patterns over the output lattice, not surprisingly the confused

input phoneme sequences often were similar, especially in their initial and/or final subsequences (e.g., ball/bell and spider/sweater).

A somewhat unexpected finding was that in spite of the sequential nature of the input, the multiple simultaneous winner nodes, and the lateral intra-lattice connectivity that influenced selection of winning nodes, well-organized maps of the individual phoneme input patterns still formed. These were reminiscent of maps seen with traditional one-shot (Kohonen) SOMs, with similar phonemes being generally adjacent to one another. For example, there was clear cut separation of vowel and consonant phonemes from one another. Of course, since unlike with traditional SOMs my model has multiple simultaneous winner nodes, multiple copies of such maps were present. This finding was very robust to variations in the weighting given to afferents vs. lateral connections (parameter α).

The findings of this study suggest that SOMs have a greater role to play as useful tools for sequence processing than is generally recognized. Still, there is room for future research to improve on the capabilities of SOMs in this regard. Perhaps most important, future theoretical and experimental studies are needed of ways to guarantee the uniqueness of the spatial representations that are learned for similar input sequences. While it might be true that using larger networks could resolve this issue, a more satisfying solution would use methods that increase the effectiveness of the time-to-space mapping without enlarging the maps. Some methods, which were not examined here, that might be explored include the use of noise during training to encourage more separation of the final activation patterns of very similar sequences, or increasing the time span of learning effects on lateral intra-lattice connections from one time step to two or three (reaching back further in time has proven effective in improving supervised sequence learning in some past non-SOM systems).

Chapter 5

Genetic Multi-Objective Optimization of a One-Shot Multi-Winner SOM

The overall goal of this chapter is to improve the performance of the sequence processing one-shot multi-winner SOM of the preceding chapter when applied to the task of learning unique spatial representations for large sets of variable-length temporal input sequences. This goal is pursued in two stages. First, possibilities for improvement via modifications of the network design are identified and then tested to determine the best combination of design alternatives. Subsequently, and using the best modified network design, the previously fixed parameters of the network that determine its activation and learning dynamics are optimized using a genetic multi-objective optimization algorithm. These efforts produce a system that outperforms the original network in terms of the arguably most important performance measure. However, the experimental results are not exclusively in favor of this new system as explained in the discussion at the end of this chapter.

5.1 Possibilities for Improvement

In the previous chapter, I introduced the temporal sequence processing one-shot multi-winner SOM as a method aimed at solving the problem of learning unique spatial

representations for large sets of temporal sequences. Overall, the method performed satisfactorily, learning unique representations for most of the temporal sequences in the training set, while still maintaining map formation. However, even with the best set of parametric values I could derive for determining aspects of the SOM's architecture and activation and learning dynamics (see Table 4.1), the method still confused an average of approximately 2% to 6% of the sequences in the training set with other training sequences, that is, the network transformed those sequences into non-unique spatial representations. Here I hypothesize that this behavior is due to shortcomings in the original design of the one-shot multi-winner SOM. To address these shortcomings, I propose and evaluate specific combinations of design alternatives.

Different combinations of design alternatives would be expected to lead to different sets of optimal parameter values, in particular different from those in Table 4.1. The search for optimal network parameters is further complicated by the fact that multiple objectives need to be taken into account. Specifically, the objectives are the maximization of the minimum distance d_{min} and the maximization of the average distance $\bar{d}$ between the spatial representations of two distinct sequences where minimum and average are computed with respect to all sets of two distinct sequences from a predetermined set (e.g., the training set like in Section 4.2). The effectiveness of the particle swarm optimization technique (Kennedy et al., 2001) that aided the discovery of the previously best parameter set (see Section 4.3.1) for the original sequence processing network is limited. Particle swarm optimization is not a multi-objective optimization technique per se and so, multiple objective functions need to be aggregated to form a single objective function whose optimization leads to the discovery of only a single solution where there are typically many different combi-

nations of decision variable (i.e., network parameter) values that are all optimal in the multi-objective sense. For this reason, in this chapter I describe work employing the recently developed NSGA-II genetic multi-objective optimization algorithm (Deb et al., 2002) instead of particle swarm optimization to determine optimal parameter settings for the initially most promising combination of one-shot multi-winner SOM design alternatives.

I now consider six possible design alternatives for the sequence processing one-shot multi-winner SOM. Three of the six considered design alternatives are inspired by neurobiological reality. First, the distributed, multi-focal patterns of activation found in biological cortex comprise foci of differing activation levels (Donoghue et al., 1992; Georgopoulos et al., 1988; Pei et al., 1994) where the maximum level of activation at each focal point presumably is directly related to how well the neural elements at the respective location are tuned to their particular inputs at the time. However, in the original one-shot multi-winner SOM, all foci of activation featured a homogeneous distribution of activity with the winner node at the center being maximally active. Alternatively, the activation level of each winner node can be made proportional to how well the weights on its incoming connections match the current inputs. This should lead to a smaller fraction of the training sequences being confused by the network since even if two distinct sequences cause the exact same output nodes to be the winners that comprise the final activation pattern, their respective activation levels now may differ, something that was not possible in the original design.

The second design alternative aims to improve the learning ability of the network by allowing a weight on a lateral intra-lattice connection to decrease in response to subsequent inputs activating the output node at which the connection terminates prior to activating the output node from which the connection originates. This is different

from the originally implemented learning rule for the weights on lateral intra-map connections (Eq. 4.6). With the original rule, a weight could decrease only due to the renormalization of lateral weight vectors (Eq. 4.7) which penalizes those weights that have grown relatively little or not at all since the last time step. The new learning rule more accurately models biological temporally asymmetric Hebbian learning where synaptic efficacy is actively reduced when the post-synaptic neuron fires prior to the pre-synaptic neuron (Bi and Poo, 2001, 1998; Markram et al., 1997; Zhang et al., 1998). Because of the increased specificity of the weight changes, the new learning rule is expected to increase the overall performance of the sequence processing one-shot multi-winner SOM. In addition, the proposed change to the learning rule will be seen to constitute a simplification of the network which lowers the computational cost of its training.

The third biologically-inspired alternative allows the formation of inhibitory lateral intra-lattice connections. The original rule explicitly prevented lateral weights from taking on negative values. However, in biological cortex, inhibitory lateral intra-cortical connections, while outnumbered by their excitatory counterparts, exist and apparently contribute to cortical information processing. By removing the biologically implausible constraint of all-excitatory intra-lattice connections, the one-shot multi-winner SOM's lateral learning rule becomes simpler and potentially more powerful.

The remaining three design alternatives are aimed at removing biases and inconsistencies. Output nodes near the lattice boundaries of the original sequence processing one-shot multi-winner SOM were found to be more likely to win the competitions for activation and learning than nodes near the center of the lattice. This was because nodes near the boundary had fewer competitors (unlike in Chapter 3 where this factor was controlled). The fourth alternative is thus to make to make each node compete

with (and receive connections from) its N closest neighbors in the lattice as with the non-sequence processing one-shot multi-winner SOM of Chapter 3, and as opposed to having a fixed radius of competition as in the previous chapter. To reduce computational cost, lateral weight vectors in the sequence processing network were originally normalized so that their components added to one. Optimizations in the network's implementation have caused other factors to dominate computational cost so that a fifth possible modification is to allow lateral weight vectors to be normalized to unit length, ensuring consistency with the normalization to unit length of all afferent weight vectors. Finally, if a node in the output lattice is relatively close to multiple winner nodes, the respective contributions of the winners to the node's activation level can be combined in different ways. In the original sequence processing network of the previous chapter, the contributions were accumulated, which sometimes rendered non-winner nodes maximally active. The activation level can alternatively be set to the maximum of the contributions as was done in Chapter 3, thus avoiding maximally active non-winner nodes.

The remainder of this chapter is organized as follows. After a brief introduction of basic concepts and terms that are central to multi-objective optimization, the design alternatives for the sequence processing one-shot multi-winner SOM are described formally in more detail and side-by-side with the original design choices. Thereafter the results of evaluating specific combinations of design alternatives and optimizing the network parameters are presented, followed by a discussion that relates the results to neurobiological reality and argues their relevance with respect to temporal sequence processing in computer science.

5.1.1 Multi-Objective Optimization

Recently, there has been a surge in research on new, and particularly genetic/evolutionary multi-objective optimization algorithms and their applicability to various optimization problems (Coello, 2001; Corne et al., 2000; Deb et al., 2002; Jensen, 2003; Knowles and Corne, 2000; Tan et al., 2002; Zitzler and Thiele, 1999). The application domains of multi-objective optimization techniques are diverse, ranging from the classic domain of engineering (Gaiddon et al., 2004; Kanazaki et al., 2004; Marseguerra et al., 2004) to, for example, physics and chemistry (Hennessy and Kelley, 2004), and medicine (Schreibmann et al., 2004).

The central concept in multi-objective optimization is domination (Coello, 2001; Deb, 2001). Two solutions to an optimization problem with multiple, often conflicting objectives are not comparable in general. One solution might be better with respect to one objective, while the other is superior with respect to a different objective. An important situation occurs when one solution is in fact better than another with respect to at least one objective and not worse with respect to the other objectives. In that situation, the better solution is said to *dominate* the other solution. The set of solutions that dominate all other solutions, but not each other, is called the *Pareto-optimal front* (Pareto, 1896).

The goal in multi-objective optimization is to find the Pareto-optimal front to a given problem. Typically, this is difficult or impossible to do analytically and so, heuristic search methods are employed that try to find solutions close to the Pareto-optimal front (Coello, 2001; Deb, 2001). Ideally, the solutions found are diverse, that is, they are approximately uniformly distributed along the entire Pareto-optimal front, and thus they are representative of the whole range of possible trade-offs between the different objectives, leaving the final choice of a 'production' solution from the

set of Pareto-optimal solutions with the user.

Several multi-objective genetic optimization algorithms have been developed (e.g., Corne et al. (2000); Deb et al. (2002); Knowles and Corne (2000); Zitzler and Thiele (1999)). In general, it is difficult to objectively compare the effectiveness of these algorithms (Zitzler et al., 2003). In Deb et al. (2002), the author compared his novel NSGA-II algorithm with the PEAS (Knowles and Corne, 2000) and SPEA (Zitzler and Thiele, 1999) algorithms on nine two-objective benchmark functions from the literature. NSGA-II was found to outperform the other algorithms on all but one of the benchmark functions. It generated more diverse solutions, maintained a better coverage of the Pareto-optimal front, and converged closer to the true theoretical Pareto-optimal front. Although comparative studies by the authors of a particular algorithm are always problematic, the fact that Deb et al. (2002) was one of ISI's fast breaking papers (Deb, 2004) and that since its publication, NSGA-II has been the subject of numerous mostly application-oriented studies indicates its effectiveness. In addition, NSGA-II is conceptually simple and thus, easy to implement.

Central to NSGA-II is the combined ranking of both the parent and offspring solutions in a generation according to their degree of Pareto-optimality (dominance sorting) and proximity to other solutions (crowding distance sorting). The best ranked individuals are selected to serve as the parents of the next generation, that is, NSGA-II is elitist: the offspring does not simply replace the parents, but competes with them so that the best found solutions are always retained in the population and never lost. With a set probability, the 'genomes' (real-valued vectors) of a pair of parent solutions that have been tournament-selected from the pool of all parents undergo recombination via simulated binary crossover (Deb and Beyer, 1999). Each of the two resulting solutions is, with a certain probability, subjected to a polynomial mutation

(Deb and Goyal, 1996). The parents and their offspring comprise the new generation of solutions. This process is repeated either until some convergence condition is met or a preset number of generations have passed.

In this chapter, NSGA-II will be used to evolve Pareto-optimal values for the parameters that determine the activation and learning dynamics of the sequence processing one-shot multi-winner SOM. Each evolved solution, i.e., set of parameter values, will be evaluated in terms of multiple post-training performance measures. That is, for each evolved set of parameter values, the one-shot multi-winner SOM, whose design will be fixed at this point, will first be trained and then its performance will be determined by measuring the objectives d_{min} and $\bar{d}$. These measurements then correspond to the objective function values for the respective evolved solution, which mostly determine the rank of the solution and thus, whether or not it is included in the next parent generation and has a chance to procreate.

5.2 Methods

This section will first make explicit which parts of the original sequence processing one-shot multi-winner SOM design are kept unchanged. Thereafter it will describe in detail the six design alternatives for the sequence processing one-shot multi-winner SOM, contrasting them with the original design choices from the previous chapter. The final part of this section details the experimental methods.

5.2.1 Unchanged Aspects of the Network

As before, the output or cortical nodes of the one-shot multi-winner SOM are arranged in a regular, rectangular lattice of R rows by C columns where the distance between two output nodes i and i' at positions (r, c) and (r', c') respectively is measured by

the box-distance metric, $\mathrm{d}(i, i') = \max(|r - r'|, |c - c'|)$. Each output node receives connections from all the nodes in the input layer. w_{ij} denotes the weight on the connection from the j^{th} input to the i^{th} output node, and $\vec{w}_i$ ($||\vec{w}_i||_2 = 1$) is the afferent weight vector of output node i, comprising the weights on all connections from the input layer. v_{ij} stands for the weight on the lateral connection from the j^{th} to the i^{th} output node, $v_{ij} = 0$ if i and j are not connected, and $v_{ii} = \beta \in \mathcal{R}$ is a fixed self-connection weight.

The first steps of the process that the network executes to determine the activation pattern $\vec{y}$ across the lattice of output nodes in response to an input vector $\vec{x}$ remain unchanged. First, the net input to each output node i at time step t is computed as $h_i(t) = \alpha \vec{w}_i^T \vec{x}(t) + (1 - \alpha)\vec{v}_i^T \vec{y}(t-1)$ where α determines the relative contributions of afferent ($\vec{x}(t)$) and lateral ($\vec{y}(t-1)$) inputs to the net input. Next, the set of winner nodes is determined as $V(t) = \{i \mid \forall j \neq i : j \in N_{\text{conn}}(i) \Rightarrow h_j(t) < h_i(t)\}$ where $N_{\text{conn}}(i)$ is the set of output nodes that send connections to output node i. The last step, i.e., computing $\vec{y}$ from $\vec{h}$ will be seen to largely depend on the design alternatives which are described below.

The high-level procedure for training the network is unchanged. The fixed training set consists of the same 60 distinct temporal sequences that were used to train the original network of the previous chapter. They had been selected at random from the set of all available sequences (phonetic transcriptions of 175 English nouns from the NetTalk and Snodgrass corpora (Sejnowski and Rosenberg, 1987; Snodgrass and Vanderwart, 1980), encoded as sequences of 34-dimensional phoneme feature vectors; see the previous chapter and Appendix B for examples and further details). The number of training epochs is 1000.

The learning rule that applies to the afferent weights of the network is identical

to the original rule, that is, the afferent weight vector $\vec{w}_i$ of output node i is updated at time step t according to

$$\vec{w}_i(t) = \vec{w}_i(t-1) + \mu y_i(t)\vec{x}(t) \quad (5.1)$$

$$\vec{w}_i(t) = \vec{w}_i(t)/||\vec{w}_i(t)||_2 \quad (5.2)$$

where t ranges from one (update in response to the first component vector of the current input sequence) to k (the length of the current input sequence) and $\mu \in (0, 1]$ is the afferent learning rate. In contrast, the design alternatives affect several changes in the original lateral learning rule which will be presented below.

As before, the values of the parameters γ, μ and η monotonically decrease according to the function $s(t) = s_{\text{fin}} + (s_{\text{init}} - s_{\text{fin}})/(1 + e^{(t-s_{\text{infl}})/s_\sigma})$ from some initial value s_{init} to a smaller final value s_{fin} where s_{infl} corresponds to the steepest point of the descent (the point of inflection) and s_σ determines the overall steepness. Here, s serves as a placeholder for γ, μ and η, respectively.

The measures of network performance that were defined in the previous chapter are reused here without any changes. The distance between two final activation patterns $\vec{y}$ and $\vec{y}'$ (the activation patterns in response to the last input vectors of two input sequences) is again measured in terms of the sum of the absolute componentwise differences between $\vec{y}$ and $\vec{y}'$, that is, $d(\vec{y}, \vec{y}') = ||\vec{y} - \vec{y}'||_1 = \sum_i |y_i - y'_i|$. The network uniquely represents all sequences of a set if for every two distinct sequences X and X' from a given set S, the distance between the two corresponding spatial representations (i.e., final activation patterns) is non-zero, that is, $|Z| = |\{\{X, X'\} \mid X \neq X', d(\vec{y}_X, \vec{y}_{X'}) = 0\}| = 0$. The minimum, $d_{\text{min}} = \min_{X \neq X'} d(\vec{y}_X, \vec{y}_{X'})$, and the average, $\bar{d} = \frac{1}{|S|} \sum_{\{X,X'\}, X \neq X'} d(\vec{y}_X, \vec{y}_{X'})$, of the distances between the spatial representations for every two distinct sequences from a given set serve as the performance measures for the sequence processing one-shot multi-winner SOM. The performance measure

d_{min} is arguably more important than $\bar{d}$ since it measures the degree to which the one-shot multi-winner SOM is able to distinguish the most difficult, that is, most similar pairs of distinct sequences.

5.2.2 Six Potential Design Alternatives

Table 5.1 provides a summary of the six potential alternatives in the design of the sequence processing one-shot multi-winner SOM, side-by-side with the original design choices that were made in the preceding chapter. Both alternatives A and F change the original formula for determining the activation level $y_j(t)$ of an arbitrary output node j at time step t, which is why they will be considered in conjunction here, even though they are independent of each other. If both alternatives were in effect, the following new formula would result:

$$y_j(t) \;=\; \max_{i \in V(t)} \begin{cases} h_i(t)\gamma^{\mathrm{d}(i,j)} & \text{if } j \in N_{\text{conn}}(i) \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

where $V(t)$ denotes the set of winner nodes at time step t and $N_{\text{conn}}(i)$ is the set of other output nodes from which i receives lateral intra-lattice connections. In Eq. 5.3, alternative A corresponds to the presence of the scaling factor $h_i(t)$, that is, the net input to winner node i, in front of each $\gamma^{\mathrm{d}(i,j)}$ term. Alternative F corresponds to the use of the maximum operator instead of the original summation operator, and the lack of an explicit upper bound on the result. The latter originally prevented non-winner nodes from becoming more active than winner nodes, an effect that cannot occur with the maximum operator. Intuitively, alternative F prescribes that an arbitrary output node's activity equal the maximum (as opposed to the sum) of all sources of activation in the node's connection neighborhood. Each source is still Gaussian-shaped with γ determining the rate of decay (like in preceding chapter), but with

Table 5.1: One-Shot Multi-Winner SOM Design Alternatives

	Design choices made for the sequence processing one-shot multi-winner SOM of the previous chapter	Potential alternatives
A	winner nodes are maximally active (Eq. 4.3)	a winner node's activation level is proportional to its net input (Eq. 5.3)
B	lateral weights cannot decrease prior to renormalization (Eq. 4.6)	lateral weights are allowed to decrease prior to renormalization (Eq. 5.4)
C	all lateral weights are initially non-negative and remain non-negative (Eq. 4.6)	lateral weights initially can be negative or non-negative and may change sign during training (Eq. 5.5)
D	each output node is connected to and competes with all other output nodes within a fixed radius	each output node sends connections to and competes with its N closest neighbors
E	lateral weight vectors are (re)normalized so that the sum of their components is one (Eq. 4.7)	lateral weight vectors are (re)normalized to unit Euclidean length
F	an output node's activation level is the sum of the contributions from all winner nodes within the connection neighborhood (Eq. 4.3)	an output node's activation level is the maximum of the contributions from all winner nodes within the connection neighborhood (Eq. 5.3)

alternative A in place, its peak amplitude now depends on the net input to the winner node at the center. Thus, in general, the Gaussian will be the taller the closer the afferent and lateral weight vectors of the respective winner node are to the afferent and lateral inputs at the current time step.

Alternatives B and C impact the learning rule that applies to the lateral intra-lattice connection weights of the network (formerly Eq. 4.6). Alternative B allows the weight of a lateral connection to decrease if the activity at the previous time step of the output node at which the connection originates ($y_j(t-1)$) coincides with a decrease in the activity of the target node ($y_i(t) - y_i(t-1)$). Specifically,

$$v_{ij}(t) = \begin{cases} \max(0, v_{ij}(t-1) + \ldots & \text{if } j \neq i \text{ and} \ldots \\ \ldots + \eta y_j(t-1)(y_i(t) - y_i(t-1))) & \ldots j \in N_{\text{conn}}(i) \\ v_{ij}(t-1) & \text{otherwise} \end{cases} \quad (5.4)$$

where $\eta \in (0, 1]$ is the lateral learning rate. Alternative C goes a step further by allowing lateral weights to stay or become negative, that is, inhibitory lateral connections are permitted. The formal rule in this case is

$$v_{ij}(t) = \begin{cases} v_{ij}(t-1) + \ldots & \text{if } j \neq i \text{ and} \ldots \\ \ldots + \eta y_j(t-1)(y_i(t) - y_i(t-1)) & \ldots j \in N_{\text{conn}}(i) \\ v_{ij}(t-1) & \text{otherwise} \end{cases} \quad (5.5)$$

As opposed to the other design alternatives, B and C are not independent of each other. Inhibitory (negative) lateral weights are permitted only if lateral weights are also allowed to decrease prior to normalization, that is, C subsumes B. If alternative C was in effect, but not B, then a positive lateral weight could never become negative, and a change in a negative lateral weight would always lead to either a negative lateral weight of a smaller magnitude or a positive lateral weight which would subsequently

stay positive. However, the intent of C is to allow all lateral weights to freely change sign, which requires alternative B to be in place also. That is why Eq. 5.5 includes the changes to the lateral learning rule that are contained in Eq. 5.4.

In the original sequence processing network, each output node received (sent) lateral intra-lattice connections from (to) all nodes within a fixed radius. In contrast, with alternative D in place, each output node receives lateral intra-lattice connections from all of its N closest neighbors (ties are resolved arbitrarily). Accordingly, the connection neighborhood of an arbitrary output node i is redefined as $N_{\text{conn}}(i) = \{j \mid j \text{ sends a connections to } i\}$. This ensures that all output nodes receive the same number of lateral connections, thus compete with the same number of other output nodes for activation and learning, and hence, have at least initially the same chance of being selected as winners via Eq. 4.2. Beyond this immediate effect, alternative D further influences the activation and learning dynamics of the network since the net input to a node, its activation level, and changes to the weights on its incoming lateral connections all depend on the node's connection neighborhood.

The last remaining design alternative E concerns the initial normalization and the re-normalization after an update of the lateral weight vectors. Each lateral weight vector comprising the weights on all incoming lateral connections to an output node, but excluding the weight on the node's self-connection, can be (re)normalized to unit length ($\forall i : ||\vec{v}_i||_2 = 1$) as opposed to the unit sum (re)normalization scheme from the previous chapter (Eq. 4.7). The latter, even though it was inconsistent with the normalization of all afferent weight vectors to unit length, was used originally to cut the computational cost of network training which has been optimized considerably since.

5.2.3 Experimental Procedures

Twelve specific combinations, that is, subsets of design alternatives were selected for an initial comparison in terms of network performance. Specifically, all six subsets of the biologically-inspired modifications (A, B and C in Table 5.1) that contain B whenever C is contained were selected. Each of these six combinations was used both in isolation and in addition to all of the other three design alternatives (D, E and F in Table 5.1). This selection includes the empty subset (no design changes) that corresponds to the original sequence processing one-shot multi-winner SOM and serves as the baseline for comparison. Based on the results of 10 independent training runs per combination, and in terms of the performance measures d_{min} and $\bar{d}$ (with respect to the training set of sequences) "best" combination was determined. During these training runs, the network parameters (α, β, etc.) were set to the values from Table 4.1, which previously had been found to work best with the original sequence processing one-shot multi-winner SOM. Note that *no* genetic optimization took place during this initial stage of the study.

Only after determining the best combination of design alternatives, the network parameters were optimized using the NSGA-II multi-objective genetic algorithm (Deb et al., 2002). Four objective functions were used: the performance measures d_{min} and $\bar{d}$, each evaluated twice after training of the network, once for the sequences in the training set (the same 60 sequences as in the previous chapter) and once for the sequences in the test set (the remaining 115 sequences). The number of test examples was so unusually high compared to the number of training examples because the high computational cost of training strictly limited the size of the training set, leaving a relatively large number of unused sequences that were then utilized for the test set. The initial population of parameter sets contained the previously best

parameter set and 39 slight random variations of it. The parent population size was 40 throughout the 50 generations for which the algorithm ran. Consequently, a total of 2000 individual networks needed to be trained and subsequently evaluated with respect to the four objective functions. In order to eliminate from the optimization results any variance resulting from differing initial random training conditions, each of the networks was initialized and trained using the same random number generator seed, that is, the initial set of weights and the order in which the training examples were presented was the same for all networks. This of course raises the question to what degree the optimization results are sensitive to changes in the initial random conditions of network training. This issue was addressed by conducting 19 additional training runs with different and distinct initial random conditions for each of the 40 parameter sets comprising the final generation of parents produced by the genetic algorithm (i.e., the best found solutions due to elitism). The results are presented below.

The genome of an individual was a real-valued vector where each component corresponded to one of the network parameters that were subject to genetic optimization. The genomes of a pair of parent individuals always underwent simulated binary crossover (crossover probability of 1; Deb and Beyer (1999)), and 10% of the genes (vector components) of the resulting offspring were subjected to polynomial mutation (Deb and Goyal, 1996) where the maximum magnitude of a mutation was limited to 0.1. The distribution indices of the polynomial probability distributions that underly the crossover and mutation operators were both set to 1 (Deb and Goyal, 1996; Deb and Beyer, 1999).

5.3 Results

The first part of this section presents the results of comparing several alternative designs of the sequence processing one-shot multi-winner SOM, including the original design that was used in the previous chapter, in terms of their performance when applied to the task of learning unique spatial representations for large sets of variable-length temporal input sequences. Based on these results, one design was selected and subsequently used as the basis for the optimization of the parameters that determine the networks activation and learning dynamics. This was done using a genetic multi-objective optimization algorithm, and the results of its application are detailed in the latter half of this section.

5.3.1 Manual Determination of the Best Combination of Design Alternatives

Table 5.2 shows how the sequence processing one-shot multi-winner SOM performed on average over 10 independent training runs (with the standard deviation given in parenthesis) for each of the selected 12 combinations of design alternatives, using the network parameters from Table 4.1 in the preceding chapter. The performance measures were d_{min} and $\bar{d}$ with respect to the training set and after training. The top row corresponds to the original unaltered sequence processing one-shot multi-winner SOM of the previous chapter. In the subsequent rows, the existence of particular design changes is indicated by 'x' markers in the respective columns to the left where the column headers correspond to the labels in Table 5.1.

With none of the investigated combinations of design alternatives did the network perform better with respect to both d_{min} and $\bar{d}$ than in its unaltered state. In general, $\bar{d}$ was less than for the unaltered network and was especially low for networks that

Table 5.2: Network Performance for Combinations of Design Alternatives

A	B	C	D, E & F	d_{min}	$\bar{d}$
				0.40 (0.88)	23.07 (1.00)
x				0.08 (0.09)	14.91 (0.62)
	x			0.50 (0.71)	22.60 (0.92)
x	x			0.06 (0.04)	14.76 (0.61)
	x	x		0.20 (0.63)	20.61 (1.15)
x	x	x		0.02 (0.02)	14.05 (1.10)
			x	0.00 (0.00)	19.15 (0.80)
x			x	0.44 (0.25)	17.52 (0.66)
	x		x	0.00 (0.00)	19.22 (1.12)
x	x		x	0.46 (0.14)	17.58 (0.72)
	x	x	x	0.00 (0.00)	19.42 (0.68)
x	x	x	x	0.32 (0.15)	16.87 (0.67)

incorporated net input proportional activation (A in Table 5.1) but not the modifications D, E and F. With the changes D, E and F in place, $\bar{d}$ recovered some, but still fell short of the values for the original sequence processing network.

For three specific combinations of design alternatives, the network performed better with respect to d_{min}. The largest d_{min} value (0.50) was measured when the only design change was to permit lateral weight decreases prior to weight renormalization (third row in Table 5.1). However, the variance of d_{min} in this case was relatively large (0.71^2).

The variance of d_{min} was comparatively small (0.14^2) for the combination of net input proportional activation (A), permitting lateral weight decreases prior to weight

renormalization (B) and including modifications D, E and F (third to last row in Table 5.1). The average of d_{min} (0.46) for this combination was the second highest of all the considered combinations, and the value was close to the observed maximum (0.50). The average performance in terms of $\bar{d}$ was lower (17.58) than for many of the other combinations of design alternatives. However, as mentioned earlier, d_{min} was considered the more important performance measure. Because of being the second best combination in terms of the most important performance measure, the average of d_{min}, and showing a comparatively small variance in the d_{min} values that promised robust results, the final decision was to change the design of the sequence processing one-shot multi-winner SOM accordingly, that is, to incorporate the modifications A, B, D, E, and F. This design is fixed from now on and in particular during the next stage of improving the sequence processing capabilities of the one-shot multi-winner SOM via the genetic optimization of the network parameters, which so far had been set to the previously best values for the original sequence processing one-shot multi-winner SOM (Table 4.1).

5.3.2 Genetic Multi-Objective Optimization of Network Parameters

Table 5.3 lists the network parameters that were subject to genetic optimization, gives the permissible range for each parameter during optimization, and reviews the parameters' functions. The optimization of these eleven network parameters with respect to the four objective functions (d_{min} and $\bar{d}$ evaluated on the training and test data sets) resulted in all 40 individuals comprising the final parent generation to be Pareto-optimal solutions. That is, of all the final evolved sets of parameter values (solutions), none was better than another in terms of all four objectives.

The positions of the solutions in the complete 4D objective function space cannot

Table 5.3: Network Parameters Subject to Genetic Optimization

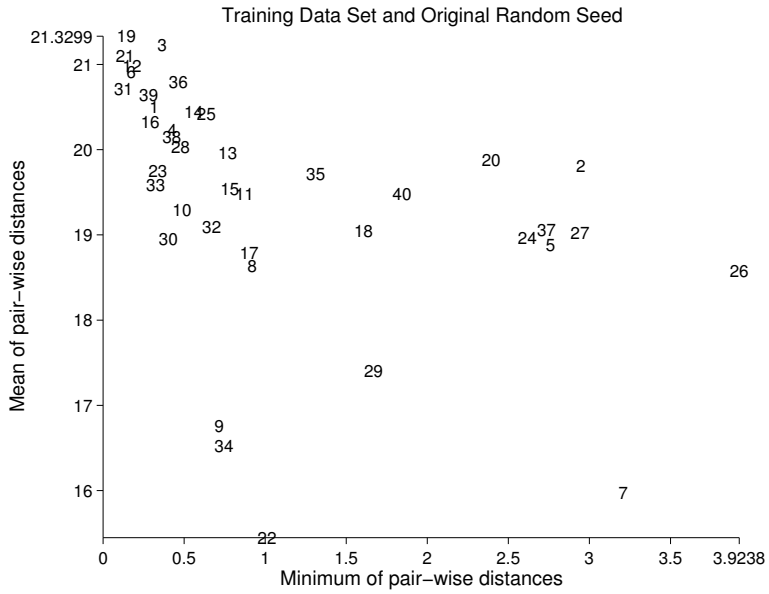
parameter	permissible range	function
α	[0, 1]	determines relative contributions of afferent vs. lateral inputs to the net input of an output node (in Eq. 4.1)
β	[-1, 1]	fixed weight on the self-connections of all output nodes
γ_{init}	[0, .8]	smoothness of the activation peaks centered at winner nodes at the beginning of training
γ_{infl}	[0, .6]	fraction of training epochs until the point of γ's steepest descend
γ_{σ}	[.01, .2]	overall steepness of γ's descend during training
μ_{init}	[0, .8]	afferent learning rate at the beginning of training
μ_{infl}	[0, .8]	fraction of training epochs until the point of μ's steepest descend
μ_{σ}	[.01, .1]	overall steepness of μ's descend during training
η_{init}	[0, .8]	lateral learning rate at the beginning of training
η_{infl}	[0, .8]	fraction of training epochs until the point of η's steepest descend
η_{σ}	[.01, .1]	overall steepness of η's descend during training

be visualized. However, one can show the solutions projected onto lower-dimensional subspaces of the objective function space that are of particular interest. For example, Figure 5.1A shows the solutions in the 2D subspace that is spanned by the two objectives d_{min} and $\bar{d}$ with respect to the training set of input sequences. In contrast, Figure 5.2A displays the solutions in a different 2D subspace: the one that is spanned by d_{min} and $\bar{d}$ when they are evaluated on the *test* set of input sequences.

Note that each of the data points in Figures 5.1A and 5.2A is the result of only a single training run. Recall that during optimization, every network that was trained started out with the same set of initial random weights and saw the training sequences in the same random order. So, during optimization, each evolved solution was evaluated only once in terms of network performance after a single training run, including the solutions of the last parent generation. For the latter, Figures 5.1A and 5.2A show the objective function values (performance measurements) obtained in this single evaluation. Consequently, the figures are not suitable for a direct comparison based on average performance with the original sequence processing one-shot multi-winner SOM. A comparison like that requires multiple evaluations of each final evolved set of network parameters based on multiple independent training runs.

Figure 5.1 (next page): Post-training performance of the evolved network parameter sets with respect to the training set. **A** Each of the 40 parameter sets comprising the last (50^{th}) parent generation is shown with its number plotted centered at the position that corresponds to the values of d_{min} (x-axis) and $\bar{d}$ (y-axis; does *not* start from zero) measured after training and with respect to the training set of input sequences. Note that this plot does *not* show the Pareto-optimal front. The Pareto-optimal front exists in the four-dimensional space that is spanned by all four objectives, while **A** is a projection of that space onto a subspace that is spanned by two of the objectives (d_{min} and $\bar{d}$ with respect to the training set). Only considering these two objectives, parameter sets 19, 3, 20, 2 and 26 are the non-dominated solutions. **B** Here, the location of each evolved parameter set (number) corresponds to the *average* values of d_{min} and $\bar{d}$ with respect to the training set, measured over 19 independent training runs per parameter set where the random number generator seed was always distinct and different from the seed that was used during the genetic optimization. The radii of the ellipsoid centered at each number correspond to the estimated variances of d_{min} and $\bar{d}$ for the respective parameter set. The special markers '$+$' (at the top) and 'x' (near the center) are for comparison. '$+$' stands for the performance of the original unaltered network in conjunction with the parameter set from Table 4.1. 'x' indicates the performance of that parameter set in combination with the new modified network.

A



B

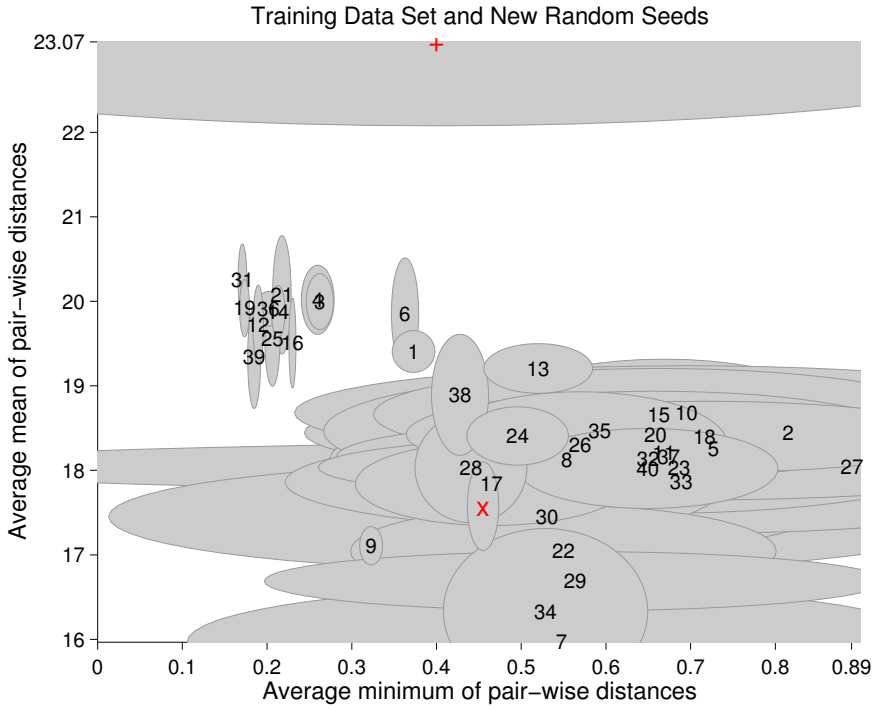


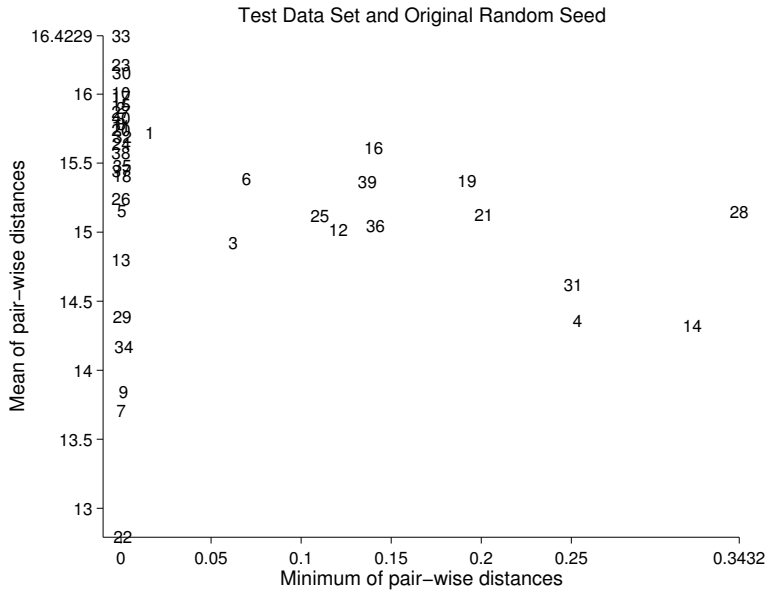
Figure 5.1: Caption on previous page

That is the purpose of the additional 19 independent training runs that were conducted for each of the 40 Pareto-optimal parameter sets making up the final parent generation. A different random number generator seed was used to initialize each training run, leading to a different set of initial weights and a different random order among the input sequences during training. The objective function values were computed after training and then averaged over the 19 independent runs per parameter set. Figures 5.1B and 5.2B show the result, that is, each parameter set is plotted at the position that corresponds to the estimated means of d_{min} and $\bar{d}$ (with respect to the training set in Figure 5.1B, and with respect to the test set in Figure 5.2B). In addition, the radii of an ellipsoid centered at the position of each parameter set indicate the estimated variance along each of the respective two dimensions. The best known parameter set for the original sequence processing one-shot multi-winner SOM ('+') and the same parameter set, but evaluated with the design alternatives A, B, D, E, and F in place ('x') were added to the figures for comparison.

For the training set (Figure 5.1B), all but one of the solutions that were discovered via genetic optimization performed better on average with respect to at least one of the two performance measures than the previously best known parameter set, but only when it was evaluated in combination with the new network design ('x' in Figure 5.1B). In direct comparison with the original unaltered network ('+' in Figure 5.1B), all found solutions on average performed worse in terms of $\bar{d}$, but the majority of the solutions was on average better with respect to d_{min} and, in addition, featured smaller variances along the d_{min} dimension.

The gap in terms of $\bar{d}$ between the original network and the optimized solutions for the new network design widened even further when they were evaluated on the test set of sequences instead of the training set (Figure 5.2B). For the original network,

A



B

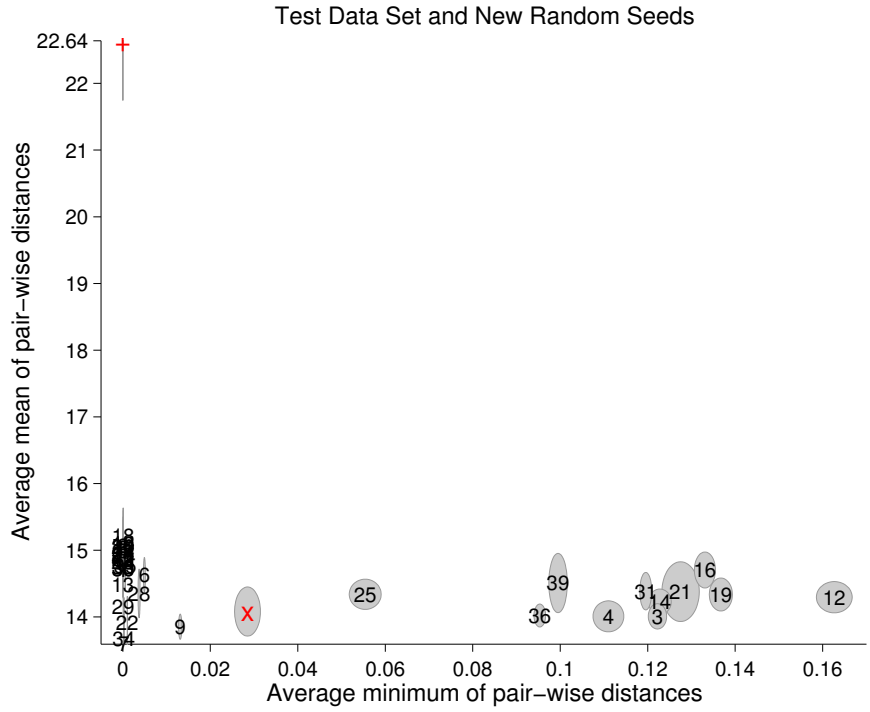


Figure 5.2: Post-training performance of the evolved network parameter sets with respect to the test set. **A** and **B** as in Figure 5.1, but with values based on the test set (as opposed to the training set) of input sequences.

the average of $\bar{d}$ decreased very little from 23.07 for the training to 22.64 for the test set. This is in stark contrast to the drop in $\bar{d}$ from around 20 to 15 for many of the optimized solutions, even though they were the product of optimizing network performance not only on the training set, but also on the test set. The parameter set for the original network (Table 4.1) had been (manually) optimized only for the training set. The situation was different for d_{min}. The decrease of d_{min} for the original network from 0.4 for the training to 0 for the test set was roughly an order of magnitude larger than the drop in performance from approximately 0.2 to between 0.13 and 0.16 for the in terms of d_{min} best optimized solutions in Figure 5.2B (e.g., solutions 12, 16 and 19).

A comparison of Figure 5.1A with Figure 5.1B and Figure 5.2A with Figure 5.2B indicates that d_{min} was much more sensitive to the initial random conditions prior to training (random number generator seed) than $\bar{d}$. Specifically, parameter sets for which d_{min} was relatively large at the end of optimization (e.g., solution 26 in Figure 5.1A and solution 28 in Figure 5.2A) suffered the most when a random number generator seed was used that was different from the seed used throughout the genetic optimization process. These solutions were typically associated with a relatively large variance along the d_{min} dimension in objective function space. This variance tended to diminish the better a solution performed on average along the $\bar{d}$ dimension. The variances along the $\bar{d}$ dimension did not exhibit a similar trend, that is, they seemed largely independent of the solutions' average d_{min} values.

The trade-off between the performance measures d_{min} and $\bar{d}$ was closely linked to parameter α which determines the relative influences of afferent, i.e. external, and lateral, i.e. lattice-internal, inputs on the activation dynamics of the network. With respect to the training set, there was a clear trend for solutions with high d_{min}

Table 5.4: The Non-Dominated Parameter Sets with respect to Average Performance on the Training Set

set	27	2	10	13	6	4	21	31	orig.
d_{min}	.89	.81	.70	.52	.36	.26	.22	.17	.40
STD	1.07	.76	.68	.25	.13	.14	.10	.07	.88
$\bar{d}$	18.04	18.44	18.68	19.21	19.86	20.02	20.08	20.26	23.07
STD	.55	.76	.75	.54	.81	.64	.84	.65	1.00
$\alpha \in [0, 1]$	.7335	.7160	.7072	.6533	.6098	.5022	.5051	.4820	.64
$\beta \in [-1, 1]$	-.0046	-.0293	-.0016	-.0976	-.1015	.0084	-.0024	.0229	.05
$\gamma_{\text{init}} \in [0, .8]$	.1653	.1286	.1353	.0482	.0847	.0247	.1372	.1269	.37
$\gamma_{\text{infl}} \in [0, .6]$	.2156	.1336	.2200	.0538	.0883	.1664	.1467	.1792	.2
$\gamma_\sigma \in [.01, .2]$	.1737	.1595	.1711	.1424	.1443	.1611	.1502	.1688	.16
$\mu_{\text{init}} \in [0, .8]$	.4590	.4560	.4651	.4330	.4352	.4589	.4205	.4465	.44
$\mu_{\text{infl}} \in [0, .8]$	.4658	.3757	.4698	.3827	.3716	.4034	.3446	.3675	.4
$\mu_\sigma \in [.01, .1]$	.0579	.0103	.0393	.0100	.0103	.0182	.0217	.0469	10^{-4}
$\eta_{\text{init}} \in [0, .8]$	.7358	.4540	.7130	.5684	.7055	.6318	.5947	.6061	.62
$\eta_{\text{infl}} \in [0, .8]$	.7633	.6651	.7654	.7123	.7110	.7248	.7533	.7961	.8
$\eta_\sigma \in [.01, .1]$	.0618	.0290	.0538	.0123	.0224	.0503	.0170	.0631	.04

values (and correspondingly low $\bar{d}$ values) to also have high α values. Table 5.4 contains example solutions that illustrate this. Specifically, the provided examples are the non-dominated solutions with respect to average performance on the training set, listed in ascending (descending) order of their $\bar{d}$ (d_{min}) values. In Figure 5.1B, this corresponds to a walk from one non-dominated solution to the closest next non-dominated solution, starting with solution 27 to the far right ($\alpha = .7335$) and ending

Table 5.5: The Non-Dominated and some Almost Non-Dominated Parameter Sets with respect to Average Performance on the Test Set

set	12	19	16	21	31	39	25	18	orig.
d_{min}	.16	.14	.13	.13	.12	.10	.06	.00	.00
STD	.06	.05	.05	.07	.04	.05	.06	.00	.00
$\bar{d}$	14.30	14.34	14.70	14.38	14.39	14.51	14.34	15.21	22.64
STD	.48	.50	.52	.67	.53	.67	.48	.54	.94
$\alpha \in [0, 1]$	.4807	.4836	.4795	.5051	.4820	.4539	.5815	.7205	.64
$\beta \in [-1, 1]$	-.0058	.0339	-.0021	-.0024	.0229	.0001	.0301	-.0078	.05
$\gamma_{\text{init}} \in [0, .8]$	.1595	.2859	.2411	.1372	.1269	.2586	.2126	.1098	.37
$\gamma_{\text{infl}} \in [0, .6]$	.1783	.1606	.1704	.1467	.1792	.1713	.2123	.2285	.2
$\gamma_{\sigma} \in [.01, .2]$	.1600	.1801	.1502	.1502	.1688	.1541	.1898	.1763	.16
$\mu_{\text{init}} \in [0, .8]$	.4493	.4447	.4224	.4205	.4465	.4314	.4408	.4432	.44
$\mu_{\text{infl}} \in [0, .8]$	.3336	.3602	.3041	.3446	.3675	.2597	.4658	.3960	.4
$\mu_{\sigma} \in [.01, .1]$	.0139	.0607	.0140	.0217	.0469	.0403	.0405	.0509	10^{-4}
$\eta_{\text{init}} \in [0, .8]$	.3489	.6285	.6931	.5947	.6061	.6180	.6001	.6464	.62
$\eta_{\text{infl}} \in [0, .8]$	.7382	.7986	.7314	.7533	.7961	.7482	.7680	.7571	.8
$\eta_{\sigma} \in [.01, .1]$	.0545	.0489	.0219	.0170	.0631	.0212	.0493	.0644	.04

with solution 31 to the far left ($\alpha = .4820$). None of the other parameters exhibited an obvious trend like α did. With the exceptions of γ_{init} and μ_{σ}, they were overall remarkably similar to the best known parameter values for the original unaltered network (last column in Table 5.4).

For the test data set, the trend of α was reversed (it is unclear why). Table 5.5 demonstrates this with example solutions that were either non-dominated or close

to non-dominated in terms of average performance on the test set. They are listed from left to right in descending order of their d_{min} values (which corresponds to a right-to-left walk in Figure 5.2). The best solutions along the d_{min} dimension (e.g., 12, 19 and 16) featured small α values around 0.5. α increased for solutions with d_{min} values closer to zero (e.g., 25 with $\alpha = .5815$) and, for solutions with $d_{\text{min}} \approx 0$ (e.g., 18 with $\alpha = .7205$), α reached the level that was characteristic of the best solutions with respect to d_{min} when evaluated on the training set (e.g., 27 and 2 in Figure 5.1 and Table 5.4). In fact, the best solutions with respect to d_{min} and the test set were among the worst in terms of d_{min} for the training set, but they typically outperformed the majority of the other solutions in terms of $\bar{d}$ for the training set.

Figure 5.3 (next page): The pre-training (light gray) and post-training (dark gray) distributions of the distances between the representations of all pairs of distinct input sequences for the evolved parameter set 13. **A** shows the distributions for when the network was evaluated on the set of sequences that was used for training. **B** is the diagram that resulted from evaluating the network using the remaining sequences that were *not* used for training, that is, the test set of sequences. Note that the wider separation of the pre-training and post-training distributions in comparison to Figure 4.2 is mostly the result of smaller pre-training distances between the sequence representations due to the net-input-proportional (as opposed to the originally always maximal) activation of winner nodes.

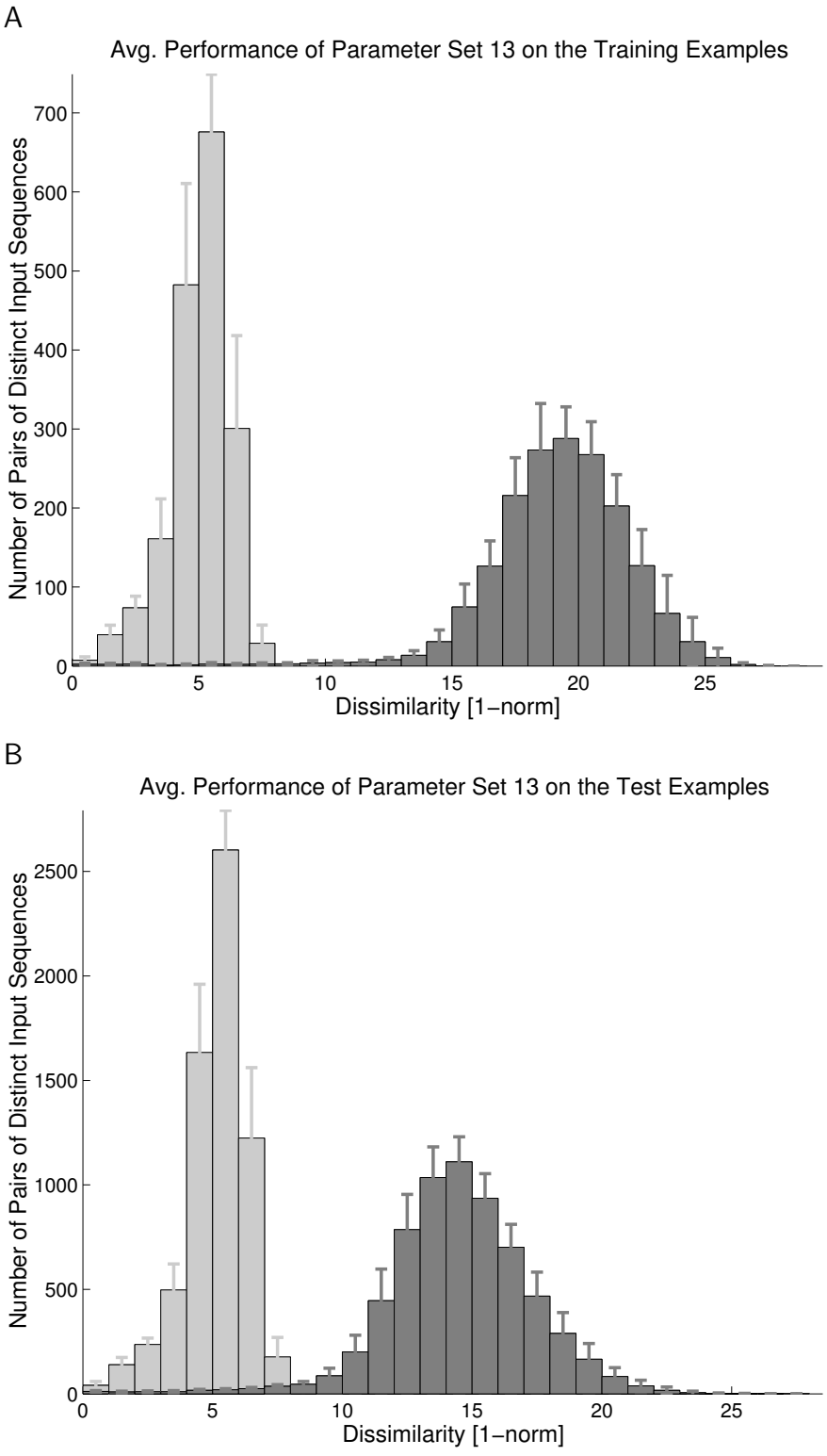


Figure 5.3: Caption on previous page.

Finally, one can compare the pre-training and post-training distributions of the distance between the two representations of two distinct sequences from the set of all distinct sequences, like was done in Figure 4.2 for the original sequence processing one-shot multi-winner SOM. Figures 5.3A and B are the equivalent plots for the modified network design, trained with the evolved parameter set 13, where Figure 5.3A shows the distributions for the training set and Figure 5.3B for the test set. Parameter set 13 was representative of the above mentioned drop in average performance by about five points in terms of $\bar{d}$, which becomes evident when comparing Figure 5.3A with Figure 5.3B. Also, the margin separating pre-training and post-training distributions was usually markedly larger than for the original network design (even for the test set; compare Figures 5.3A and B with Figure 4.2), owing to the net-input-proportional activation of the winner nodes which, prior to training, have weight vectors that are very dissimilar to the input vectors, resulting in low net input values, consequently low activation levels and thus, small initial representation distances. Training typically reduced the number of pairs of distinct sequences whose representations were so similar that their distance was smaller than one (left-most bar in the histograms of Figures 5.3A and B), just as with the original network. However, recall that with the original network, the distance between two representations was always an integer and thus, once the distance was smaller than one, it was in fact zero and the two representations were not just very similar but identical due to identical sets of maximally active winner nodes. This is not true for the new network design where the winner nodes are active proportional to their respective net input so that identical sets of winner nodes do not imply identical representations (but typically cause very similar representations).

5.4 Discussion

The genetic optimization did not yield a parameter set that, in connection with the new design for the sequence processing one-shot multi-winner SOM, performs better in all respects than the original network from the previous chapter. However, if one puts special emphasis on the performance measure d_{min}, then the genetic optimization process found many parameter sets that, in conjunction with the new network design, outperform the original network, either for the training or test data set (but not for both). With respect to performance on the training examples, parameter set 13 (see Table 5.4) perhaps is most desirable since it performed better on average than the original network in terms of d_{min}, it scored higher with respect to $\bar{d}$ than any of the other solutions that were better than the original network in terms of d_{min}, and both d_{min} and $\bar{d}$ varied relatively little from training run to training run for parameter set 13. However, if it is important that the network forms unique representations not only for the sequences in the training set, but that this capability generalizes to sequences not used for training, then parameter set 16 (see Table 5.5) is preferable because of high averages and small variances for d_{min} and $\bar{d}$ with respect to the test set.

Set 13 qualitatively differs from the best known parameter set for the original network (Table 4.1) in two aspects. First, the network's output nodes evolved to be relatively strongly self-inhibitory as opposed to having been mildly self-excitatory originally ($\beta = -.1$ vs .05 originally), the former being more in line with evidence of the predominantly inhibitory nature of cortical columns (Miller, 2003; Pinto et al., 2003). Second, the peaks of activation centered at the winner nodes were initially already extremely more focused ($\gamma_{\text{init}} = .05$ vs .37) and became even more focused much earlier ($\gamma_{\text{infl}} = .05$ vs .2) and more rapidly ($\gamma_{\sigma} = .14$ vs .16) than for the original parameter set. Thus, essentially only winner nodes ever became active. This

was true for most of the evolved parameter sets (see, for example, Tables 5.4 and 5.5), that is, limiting learning to the winners nodes (and not letting their neighbors participate in it) seems to be beneficial to network performance in general. Because the participation of the winners' neighbors in learning was thought to be crucial for map formation, no map formation was expected with γ values as low as in parameter set 13. However, even though map formation was somewhat impaired it still took place as can be seen in Figure 5.4 where a map that formed on the lattice of a SOM of the new design (trained using parameter set 13) is shown next to a map produced by a SOM of the original design (trained using the original parameter set from Table 4.1).

The evolved learning rate parameters $\{\mu|\eta\}_{\mathsf{init}|\mathsf{infl}|\sigma}$ were relatively similar to those in the original parameter set. That there was a difference of three orders of magnitude with respect to μ_σ is a result of the constraint $\mu_\sigma \in [.01, .1]$ that was enforced during genetic optimization (to limit the search space and to ensure that $\mu \approx 0$ toward the end of training). Whether $\mu_\sigma = .01$ or $\mu_\sigma \ll .01$ was assumed to not translate into qualitatively different network behavior since the shape of the sigmoidal descend of μ during training approaches a step function (is very steep at the point of inflection) in either case.

Figure 5.4 (next page): Map formation in the new versus the original sequence processing one-shot multi-winner SOM. The map to the left formed on the 30 by 20 lattice of a SOM of the new design that was trained using parameter set 13, that is, using very small values for γ throughout training ($\gamma_{\mathsf{init|infl|}\sigma} = \{.05|.05|.14\}$), meaning that only the winner nodes themselves but not their neighbors learn from a particular input vector. In that situation, map formation, which was thought to rely on a strong interaction between winner nodes and their neighbors, was unexpected. Nevertheless an ordered mapping of single phonemes appeared: output nodes sensitive to vowels (white text labels) and consonants (black text labels) have been spatially separated, the former forming multiple disconnected clusters that internally, with few exceptions, comprise only nodes sensitive to vowels. For comparison, to the right is a map produced by a SOM of the old design that was trained with the original parameter set from Table 4.1. That parameter set featured much larger values for γ during training ($\gamma_{\mathsf{init|infl|}\sigma} = \{.37|.2|.16\}$) so that winners *and* their neighbors made significant weight adjustments in response to an input. This causes an even more ordered map. For example, there are no consonants at all within the clusters of vowels, and adjacent output nodes are overall more similar in terms of their afferent weight vectors (fewer darkly shaded cells and note that due to different scales, the similarity values associated with the gray shades are much higher on the right than on the left).

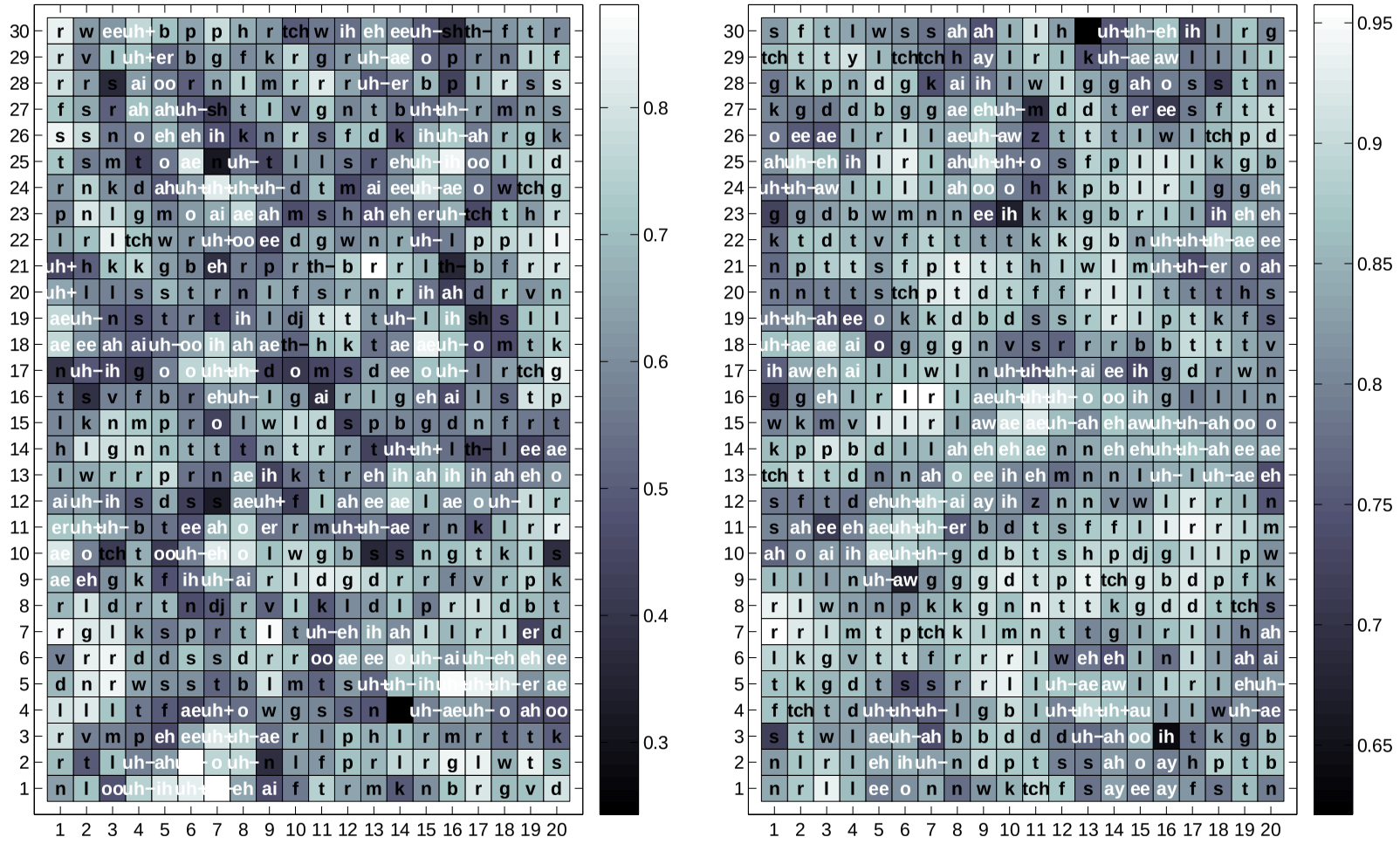


Figure 5.4: Caption on previous page

Set 16 is much more similar to the original parameter set in terms of β and $\{\gamma\}_{\mathsf{init|infl|}\sigma}$ (although the differences that exist are consistent with those of set 13, i.e., they go in the same direction). The main difference is a much smaller influence of the afferent network inputs on the network dynamics ($\alpha = .48$ vs .64 originally and .65 for set 13). But an influence of the afferent inputs that is still roughly on par with that of the lateral intra-map inputs is nevertheless much greater than what was originally expected to lead to good network performance, given the anatomical fact of many more short-range intra-cortical than afferent connections to cortical neurons that suggests a relatively small direct influence of the afferent inputs (Braitenberg and Schüz, 1991).

Even though the genetic algorithm was free to change the parameter values within their permitted ranges (see the first column of Table 5.4 or 5.5) and independently on one another, the evolved parameter sets were not radically different from the original. For example, for all evolved parameter sets it was true that the point of inflection of the afferent learning rate μ came before that of the lateral learning rate η, that is, learning on the afferent connections always ceased earlier than learning on the lateral connections. Also, even though β varied some among the found solutions, the observed variations were limited to the interval [-0.1040, 0.1847], while the permissible range of [-1, 1] was much wider. One reason for this is almost certainly the seeding of the initial population with slight variations of the original parameter set (maximally different by $+/-$ .1 in each parameter), introducing a bias toward finding better solutions near the original parameter set which makes it less likely for radically different solutions to emerge. The initial results of repeating the genetic optimization with different random initial conditions (a different random number generator seed) indicate that the region of parameter space around the original parameter set is indeed

not the only region harboring better solutions. The repeat optimization also produced 40 Pareto-optimal solutions, which, while at first glance not being radically better, did span, for example, a much larger range of the permissible interval for β ([-.6667, .6114]) and often reversed the order of afferent and lateral weight maturation. This unfortunately suggests that the fitness landscape underlying the problem of optimizing the network parameters is complex. Many more independent runs of the genetic optimization with larger population sizes should ideally have been conducted (some without biasing the initial population in any way) to delineate the promising regions of the parameter space with some certainty. However, the high computational cost of training at least 2000 networks per run, which took roughly three weeks to complete on the available cluster of workstations (see Section 3.1), made this impractical.

The modified network never performed better than the original network in terms of $\bar{d}$. The following is an attempt to explain this result. The original network had each output node compete for activation and learning with all other output nodes to which it was connected, that is, specifically, all other output nodes within a radius of $r_{\text{conn}} = 4$ on the output lattice, thereby favoring output nodes near the lattice boundaries which, as a consequence of this rule, had fewer competitors and therefore won more often. For a SOM lattice of 30 by 20 nodes and $r_{\text{conn}} = 4$, the maximum number of winner nodes that can be present on the lattice at the same time is 24 for the original network which is reached, for example, by the two configurations of winners on the lattice in Figure 5.5.

The two configurations' sets of winners are disjunct, i.e., there does not exist a node that is a winner in both configurations. Hence, the theoretically maximal distance (measured using the 1-norm metric) between two map activation patterns, and hence, between the spatial representations of any two input sequences, is 48 for

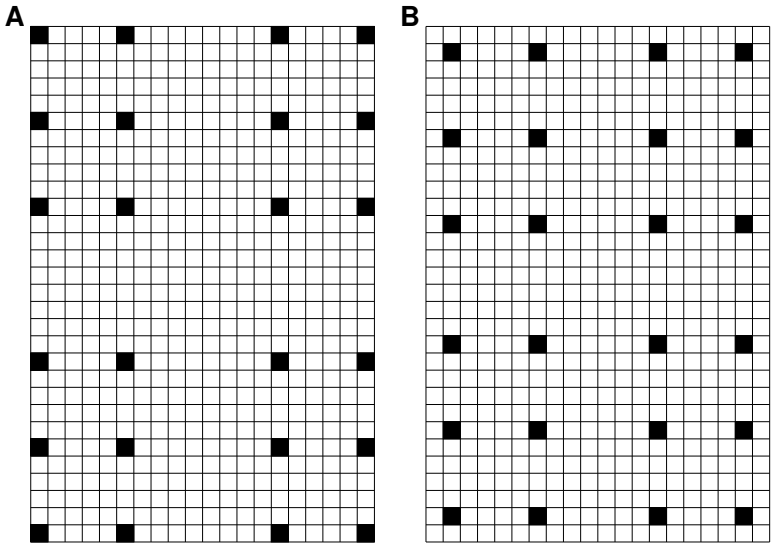


Figure 5.5: Two examples of output lattices composed of 30 by 20 nodes with a maximum number (24) of winner nodes, given that each output node competes with all other output nodes with a radius of $r_{\text{conn}} = 4$ on the lattice (like with the original sequence processing one-shot multi-winner SOM of the preceding chapter). Each square cell corresponds to an output node. The black filled squares are winner nodes, the empty squares are non-winner nodes. The minimum distance on the lattice between two distinct winner nodes is four, in accordance with the radius of competition. The sets of winners in **A** and **B** are disjunct, that is, the distance (1-norm) between these two activation patterns is 48, which is maximal.

the original network. All configurations achieving the maximum number of 24 winner nodes have in common that 16 of the 24 winner nodes are near the lattice boundary, that is, in some direction from each of these nodes the distance to the boundary is smaller than $r_{\text{conn}} = 4$. No more than 16 winner nodes can be near the boundary. By favoring nodes near the boundary in the competitions for activation and learning, the original network was biased toward representations with, on average, relatively many

winner nodes and thus, on average, relatively large distances between any two distinct representations. This would have had a positive effect on $\bar{d}$, which is an average over all pairs of distinct sequences, but would also have limited (probabilistically) the space of all possible representations, thereby making it more likely for two distinct sequences to be mapped to the same representation and thus, hurting network performance in terms of d_{min}.

One of the network design alternatives was the neutralization of boundary effects. Instead of having each node compete with all other nodes within a given radius, each node competes with a given number of nearest other nodes (the number being equal to the number of nodes within radius $r_{\text{conn}} = 4$ from a node in the center of the map, i.e., 81). To begin with, this reduces the maximum number of winner nodes and consequently, reduces the maximum distance between two representations, most likely negatively effecting $\bar{d}$. In addition, nodes near the boundary no longer have an advantage in the competitions for activation and learning. This removes the bias toward representations with relatively many winner nodes, which should compound the negative effect on $\bar{d}$ since, on average, two distinct representations will comprise fewer winners and hence, the distance between them will be reduced. On the other hand, the space of representations is no longer constrained (skewed toward representations with many winner nodes), which should help performance in terms of d_{min} since two distinct sequences are now less likely to be mapped to the same representation. This could explain why the modified network was able to outperform the original network with respect to d_{min}, but lacked behind in terms of $\bar{d}$.

Chapter 6

Discussion

This work has introduced a new class of one-shot multi-winner SOMs that combines the key features of the two existing classes of SOMs: the non-iterative, computationally efficient mechanism by which the winner node is determined in the one-shot single-winner SOM (Kohonen, 2001), and the distributed, neurobiologically more plausible and potentially more effective representation of inputs in the iterative multi-winner SOM (von der Malsburg, 1973). Seen from the perspective of computer science, this new class of SOMs represents a generalization of the SOM as a successful data processing method in various application domains. When viewed from the perspective of theoretical neuroscience, the one-shot multi-winner SOM, despite its simplicity, proves to be a surprisingly interesting computational model that can be linked to several complex phenomena in biological cortex.

6.1 The Computer Science Perspective

This dissertation first and foremost is a study of the basic properties of one-shot multi-winner SOMs. SOMs in general are one of the major approaches to unsupervised learning in artificial neural networks and they continue to be a popular research subject (Kaski et al., 1998b; Oja et al., 2003). Most past SOMs fall into one of two

distinct classes, one-shot single-winner SOMs (e.g., Callan et al. (1999); Kaski et al. (1998a); Kohonen (1982); Kokkonen and Torkkola (1990); Principe et al. (1998)) and iterative multi-winner SOMs (e.g., Bednar and Miikkulainen (2000); Li (2002); Pearson et al. (1987); Reggia et al. (2001); Sutton et al. (1994); von der Malsburg (1973)), depending on their primary purpose which determines key features of their architecture and dynamics. One-shot single-winner SOMs have received extensive attention due to their applicability to practical data processing tasks. They have been used successfully as data visualization, feature detection and pattern classification tools in a variety of application domains, for example, in computer vision (Deschenes and Noonan, 1995; Manduca, 1996; Morris et al., 1990; Takacs and Wechsler, 1997; Toivanen et al., 2003), robotics (Cervera and del Pobil, 1999; Faldella et al., 1997; Heikkonen and Koikkalainen, 1997), signal and specifically speech processing (Callan et al., 1999; Kangas, 1991; Kohonen et al., 1984), economics (Deboeck and Kohonen, 1998; Kaski et al., 1998a), and bioinformatics (Andrade et al., 1997; Ferrán and Ferrara, 1991; Hanke and Reich, 1996; Schuchhardt, 1996).

In this research, I have introduced a new class of one-shot multi-winner SOMs, a natural generalization of the highly successful one-shot single-winner SOM method that incorporates the biologically-inspired distributed representation of inputs, a key feature of the more biologically plausible but less computationally efficient class of iterative multi-winner SOMs. Central to the one-shot single-winner SOM is the computationally efficient selection in one step of the single winner in the global competition for activation and learning among the output nodes of the network in response to a particular input (Kohonen, 2001). In the one-shot multi-winner SOM, multiple localized competitions take place instead, resulting in multiple simultaneous winners that form a spatially distributed representation of the input across the output lattice

Table 6.1: The Typical Features of the Two Existing Classes of SOMs and One-Shot Multi-Winner SOMs

SOM type →	Iterative Multi-Winner	One-Shot Single-Winner	One-Shot Multi-Winner
seminal work	von der Malsburg (1973)	Kohonen (1982)	this dissertation
primary applications	neuroscience: modeling cortex	computer science: data visualization, feature detection, pattern classification etc.	potentially both computer science and neuroscience
input-to-output connectivity	divergent, but localized	full	full
intra-lattice connectivity	lateral (excite adjacent nodes, inhibit more distant ones ⇒ Mexican Hat activation patterns)	none (implicit neighborhoods)	none (static inputs; implicit neighborhoods) or range-limited lateral (sequential inputs; implicit neighborhoods)
activation dynamics	iterative solution of non-linear differential equations ⇒ multiple winners	one-shot selection of most activated node ⇒ single winner	one-shot selection of locally most activated nodes ⇒ multiple winners
learning rule	Hebbian/competitive	Hebbian/competitive	Hebbian/competitive (temporally asymmetric for lateral weights)
computational cost	high	low	low
memory capacity	high	low	high
further examples	Bednar and Miikkulainen (2000); Li (2002); Pearson et al. (1987); Reggia et al. (2001); Sutton et al. (1994)	Callan et al. (1999); Kaski et al. (1998a); Kokkonen and Torkkola (1990); Principe et al. (1998)	

of the network much like in iterative multi-winner SOMs (Cho and Reggia, 1994; Pearson et al., 1987; Sutton et al., 1994; von der Malsburg, 1973). However, the selection of these winners is still a one step process (as opposed to the iterative mechanism found in iterative multi-winner SOMs) so that computational efficiency is retained. Table 6.1 summarizes the differences and parallels that exist between the one-shot multi-winner SOM and the two previously existing SOM classes.

The systematic study of the one-shot multi-winner SOM revealed that its behavior constitutes a natural and principled generalization of the behavior of the one-shot single-winner SOM. Specifically, whenever the size of the one-shot multi-winner SOM's output lattice was sufficiently small relative to the extent of the local competitions, a single topology-preserving map of the network's inputs formed on the output lattice, a behavior identical to that of the one-shot single-winner SOM (see Figure 3.2C). However, as soon as the size of the output lattice was sufficiently larger than the extent of the local competitions, multiple neighboring topographic maps formed (e.g., Figure 3.5). The number of maps was roughly proportional to the size of the underlying output lattice. Moreover, the overwhelming majority of adjacent topographic maps were mirror symmetric relative to each other (see Tables 3.1 and 3.2). The formation of multiple, mostly mirror symmetric maps was robust to significant changes in the parameters that determine the activation and learning dynamics of the one-shot multi-winner SOM. These results are consistent with and further underpin the prevailing theory that explains the SOM's behavior in terms of the basic principle of having similar inputs be represented close to one another by nodes in the output lattice.

When a frequency gradient was introduced so that inputs from near one edge of the input space occurred three times more often than inputs from near the opposite

edge, the most frequent region of input space almost always became represented along the shared boundary between two adjacent maps, and the representation of the least frequent region was farthest removed from the boundary. Prior to that, with a uniform input distribution, no such bias had been observed, that is, all four edges of the input space had equally often been represented next to the inter-map boundary (see Figure 3.7). Finally, just like with one-shot single-winner SOMs (Grajski and Merzenich, 1990), the map representations of more frequent input regions occupied a disproportionately large area of the output lattice, that is, they were represented with a increased resolution at the expense of less frequent input regions.

In the process of obtaining and analyzing the above results, I developed a new objective metric for map formation that is especially useful when multiple adjacent maps are being studied (see Section 3.3). The principled and robust behavior of the one-shot multi-winner SOM may widen the applicability of SOMs, especially in situations that call for a high fault tolerance and/or confidence in the SOM's results. Both can be achieved by exploiting the multiple redundant map representations that form in a one-shot multi-winner SOM. The property that frequency gradients in the input space bias the orientation of adjacent maps may prove useful in data visualization.

Originally, SOMs were designed to process static, that is, time-invariant input patterns only, and still, the vast majority of the literature on SOMs assumes this type of input. The generalization of the SOM method to situations where each input is a temporal sequence of varying input patterns has only relatively recently received significant attention (e.g., Chappell and Taylor (1993); Euliano and Principe (1999); Kangas (1990); Somervuo (1999, 2003); Varsta et al. (1997); Wiemer (2003)). However, a generally accepted approach to temporal sequence processing with the SOM has not been established and in general, is an unlikely prospect due to the many

different specific tasks that fall into the category of temporal sequence processing.

The distributed, coding-efficient representations computed by the one-shot multi-winner SOM promised that the network would perform well when applied to the specific task of transforming each distinct sequence from a non-trivial set of variable-lengths sequences into a unique spatial representation. Some form of time-to-space transformation is almost always part of temporal sequence processing systems that involve neural networks (Barreto et al., 2003; Mozer, 1993), but the SOM has only once before been applied to the above specific task (James and Miikkulainen, 1995). In that study, the SOM was an unaltered one-shot single-winner SOM that simply remembered the winner node for each vector in the input sequence of vectors, a trivial but effective method. Here, a novel approach was taken, inspired by the architecture of biological cortex (Braitenberg and Schüz, 1991) and the learning dynamics at biological synapses (Bi and Poo, 2001, 1998; Markram et al., 1997; Zhang et al., 1998). Specifically, the one-shot multi-winner SOM was augmented with local lateral intra-lattice connections whose weights were trained using temporally asymmetric competitive Hebbian learning.

The thus extended one-shot multi-winner SOM was trained with temporal inputs in the form of sequences of high-dimensional feature vectors, each encoding the sequence of phonemes in an English noun naming an object. The fairly small network (30 by 20 or 40 by 30 output nodes) learned a unique distributed activation pattern across the nodes in the output lattice for, on average, 94% to 98% of the distinct sequences in the training set (60 or 175 sequences total). The sequences that the network mapped to non-unique spatial representation were typically short and/or very similar, that is, words with only two to three phonemes and/or sharing the first and/or last phonemes. In general, the more similar two sequences were the more similar their

spatial representations tended to be, which is a desirable property in cases where the spatial representations are destined for subsequent processing. An entirely unexpected result was the simultaneous formation of phoneme feature maps.

Following the promising results of this first investigation, a range of design alternatives for the sequence processing one-shot multi-winner SOM were examined with the goal to further improve the performance of the system. These efforts showed that the sequence processing performance of the one-shot multi-winner SOM is generally robust. The performance of one such design alternative was readily improved by tuning the parameters of the network using a genetic multiobjective optimization algorithm. Most notably, the genetically optimized sequence processing one-shot multi-winner SOM markedly outperformed the original system in terms of the most important performance measure, that of the degree to which the two most similar sequence representations differ.

6.2 Relevance to Neuroscience

Even though the one-shot multi-winner SOM was not an attempt to create a realistic and detailed model of cortical map development, the results obtained with it are intriguing in the context of current neuroscientific data. The input patterns to the one-shot multi-winner SOM in Chapter 3 can be viewed as encodings of point stimuli on a two-dimensional sensory surface like, for example, the skin or the retina. When trained with these inputs, multiple topologically-correct maps of the sensory surface formed on the SOM's output lattice, that is, the modeled cortical surface, provided the distribution radius of cortical afferents (i.e., the size of the output lattice) sufficiently exceeded the range of horizontal intracortical interactions. These conditions may indeed be present during brain development when thalamocortical afferent pro-

jections are more widespread than in adults (Brown et al., 2001; Mountcastle, 1998). Moreover, topographic maps that were adjacent on the output lattice overwhelmingly exhibited mirror symmetry, their common boundary being the axis of reflection. Regions of the input surface that were overrepresented in the sample of input patterns used for training became magnified, that is, their map representations occupied a relatively larger area of modeled cortical surface. These results persisted in the face of parameter variations and even a different representation of sensory stimuli as long as basic map self-organization was not disrupted.

The findings are consistent with observations of topographic maps and their relative orientations in biological cortex. Specifically, multiple adjacent, roughly mirror-image topographic maps are commonly observed experimentally in the sensory neocortical areas of many species, including humans (e.g., Drager (1975); Engelien et al. (2002); Formisano et al. (2003); Merzenich et al. (1978); Newsome et al. (1986); Sur et al. (1982)), and the magnification effect also is a well-documented property of many cortical maps (Azzopardi and Cowey, 1993; Creutzfeldt, 1978; Dykes and Ruest, 1984; Sereno et al., 1995). Thus, when viewed from the perspective of theoretical neuroscience, the one-shot multi-winner SOM comprises the first computational model of multiple mirror-image topographic map formation in biological cortex.

The model contributes to the ongoing debate within neuroscience on the degree to which topographic map formation is an activity-dependent (learning-based) or activity-independent (genetically-determined) process (Cohen-Cory, 2002; Grove and Tomomi, 2003). The initial parcellation of cortex into multiple regions/areas is generally believed to be due to genetically-determined chemical markers and independent of thalamocortical afferent activity (Sur and Leamey, 2001). However, it remains less clear as to why partially redundant cortical maps occur in these areas, why they are

so often oriented with reflection symmetry, and what role thalamocortical activity plays in their formation. Multiple adjacent maps are often hypothesized to arise during development due to genetically-mediated chemical gradients (Grove and Tomomi, 2003; Levitt, 2000; Zhou and Black, 2000). They are sometimes conjectured to have evolved due to genetic mutations (Allman and Kaas, 1971; Krubitzer, 1995), and it has been suggested that they may provide fitness advantages due to separation of spatial/temporal processing, parallel processing of different sensory attributes, minimization of connection distances, and other factors (Kaas, 1988; Cowey, 1981; Jones, 1990).

In contrast, multiple mirror-image map formation in the one-shot multi-winner SOM is driven entirely by a form of competitive Hebbian learning, an activity-dependent process. This is complementary to and consistent with the prevalent notion that activity-independent genetic factors initially determine cortical arealization and affect targeting of thalamocortical afferents. However, it raises the question of how genetic and activity-dependent synaptic changes might interact during development and even during evolution, as it seems improbable that evolutionary processes would hardwire adjacent cortical maps to be mirror images so often unless there was some advantage to this arrangement (such as consistency with local synaptic plasticity).

Finally, the model makes two specific, testable predictions that may or may not relate to biological cortical maps. First, when adjacent mirror-image topographic maps occur in neocortex, their common edge should represent the region of sensory surface that develops and innervates first (i.e., that has the most frequent stimuli initially during map development). This is consistent with, for example, the otherwise surprising location of fingers/toes in biological neocortex far from the symmetry axis in mirror image hand/foot representations in S1 (see Figure 2.4II), as these distal

digits appear late during development (Gilbert, 1994; Lonai, 1996). Second, adjacent maps may occasionally exhibit a very different rotational symmetry. If such previously unreported rotationally symmetric maps are ever observed experimentally in a small percentage of currently known cortical map regions, they would provide very strong support for the model. Such atypically oriented adjacent maps, in the context of normal connectivity between cortical regions, would be expected to cause abnormal cortical information processing, and it is natural to speculate that they might account for some of the cognitive deficits and functional imaging changes observed in neurodevelopmental disorders such as dyslexia or autism (Frank and Pavlakis, 2001; Papanicolaou et al., 2003; Temple et al., 2003). The rarity of such atypically oriented adjacent maps and the very limited experimental data on human maps may explain why they have not been reported experimentally.

Interestingly, the temporal sequence processing one-shot multi-winner SOM also exhibited multiple map formation (see Figures 4.5 and 5.4). In this case, individual maps were ordered projections of the high-dimensional feature space whose dimensions distinguished the distinct phonemes that comprised the input sequences (phonetic transcriptions of spoken words). Similarly redundant maps of complex features like, for example, the orientation of line segments, exist in biological visual cortex (Hubel and Wiesel, 1962, 1963, 1968, 1979). The intended purpose of the sequence processing one-shot multi-winner SOM was to learn unique spatial representations for its temporal input sequences. The simultaneous formation of feature maps suggests that temporal sequence processing and map formation are compatible.

Temporally asymmetric Hebbian learning of the weights on the lateral intra-lattice connections of the network proved to be an effective mechanism to learn unique spatially distributed representations for sizeable sets of temporal input sequences. Exper-

imental evidence for temporally asymmetric Hebbian changes at biological synapses in cortex (Markram et al., 1997) and other parts of the brain is accumulating (Bi and Poo, 2001, 1998; Zhang et al., 1998), but its functional role still needs to be established. The findings obtained with the SOM raise the possibility of a role in the distributed spatial representation of time-varying stimuli in biological cortex. The fact that the network tended to create similar spatial representations for similar input sequences is consistent with functional imaging studies of cortical areas in humans where similar visual stimuli were found to evoke similar spatially distributed activation patterns (Haxby, 2001; Riesenhuber and Poggio, 2002).

Finally, since phonetic transcriptions of spoken words naming objects were used to train the sequence processing one-shot multi-winner SOM, the training results can be related to cognitive science theories on human language processing, specifically, the internal representation of spoken words and their pronunciation. The sequence processing one-shot multi-winner SOM therefore has been adopted as part of a large scale neurocognitive network of naming and word repetition that is currently under development.

6.3 Going Further

As mentioned above, the computational properties of the one-shot multi-winner SOM may be useful in practical application settings. The identification of specific real-world applications for the one-shot multi-winner SOM is one issue that should be addressed by future research. Problems to which Kohonen's one-shot single-winner SOM has been applied successfully in the past would be a natural starting point for a usability study. The central question that needs to be answered in these cases is whether redundant map formation and/or mirror symmetry between adjacent maps

would be valuable additives to the standard solution found by a Kohonen SOM. For example, the fact that the orientation of two adjacent mirror maps can be biased by gradients in the input distribution may prove useful in data visualization whenever the identification of such gradients is of interest.

Redundant map formation should increase the fault tolerance and thus, the robustness of a system that uses a one-shot multi-winner SOM instead of a one-shot single-winner SOM. This increase should be quantifiable via systematic "lesioning" studies that involve the deactivation of nodes in the output lattice and the "denervation" of the network, that is, the cutting of connections from the input nodes to the output lattice. The degree of damage to the network could then be related to the network's performance, for example, the frequency of classification errors if the network was used for a pattern classification task. Lesioning studies could also be used to further determine the validity of the one-shot multi-winner SOM as a computational model of information processing in biological cortex for which experimental lesioning data is available that could be compared to lesioned model behavior.

In addition, it may be beneficial to the field of theoretical neuroscience to design and carry out a study that quantitatively compares the results of past modeling studies involving iterative multi-winner SOMs with the results obtained when a one-shot multi-winner SOM is used as the model instead. Two candidate studies are Chen and Reggia (1996) and von der Malsburg (1973). For both, implementation details of the respective iterative multi-winner SOM and the training data are available. The general issue that should be addressed is how the maps formed using the one-shot multi-winner SOM compare to those formed by iterative multi-winner SOMs. Should they be the same, then the one-shot multi-winner SOM provides a computationally more efficient shortcut, allowing larger scale models of cortical map formation to be

built and investigated. If the maps are different, then one needs to know in what ways to see which type of SOM is a more accurate model of the biological reality.

The genetic multiobjective optimization of the network parameters of the sequence processing one-shot multi-winner SOM yielded better performance in terms of the unique spatial representation of input sequences. However, the ability of the network to form feature maps was somewhat impaired (see Figure 5.4). An interesting basic question is whether the inclusion of map formation as one of the objective functions in the optimization process would restore or even improve the quality of the maps, while still leading to improved performance with respect to the temporal processing task.

Appendix A

Results of Individual Training Runs

Each entry in Tables A.1 and A.2 summarizes the outcome of a single run during the experiments described in Chapter 3 (each double-row holds the results for a single experiment). An entry consists of the number of individual maps of the sensory surface and, in the case of a multiple maps, subscripts that indicate, in order from the lattice's top to its bottom, the types of symmetry between adjacent maps: 'm' for mirror symmetry, 'g' for glide reflection symmetry, and 'r' for rotational symmetry. For example, $5_{2m,g,r}$ describes a lattice on which five well formed individual maps of the sensory surface appeared where, from top to bottom, the first and second maps as well as the second and third maps are mirror images of each other, the third and fourth maps exhibit glide reflection symmetry, and the fourth and fifth maps are rotationally symmetric.

In a small minority of runs, the network did not (completely) self-organize, that is, training did not result in a lattice completely partitioned into distinguishable and immediately adjacent maps, and these situations are indicated by question marks. Instead, parts of the SOM's lattice remained disorganized. The position of the question mark in entries with at least one number expression indicates the relative position on the lattice at which self-organization failed. For example, an entry like $2_r?2_g$ denotes

that the upper and lower parts of the lattice each formed two adjacent maps which exhibited rotational and glide reflection symmetry, respectively, but that there is a region in between the two map pairs where no recognizable self-organization took place.

The right-most column contains the double-row-wise average number of individual maps per lattice, and, as subscripts, the relative fractions of occurrence for each of the three symmetry types. Entries containing a question mark do not contribute to the average, but they do contribute to the relative fractions. The 'grand total' to the bottom-right of the table provides the relative fractions of occurrence for each of the three symmetry types over all experiments.

Table A.1: Number and Pairwise Symmetries of Learned Maps per Individual Run

R											average
11	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1.0
15	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1.0
20	?, 1	1, 1	1, 1	1, ?	1, 1	1, 1	1, ?	1, ?	1, 1	1, 1	1.0
25	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_m	
	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_m	?	$2.0_{1.0m}$
30	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_g	
	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_g	$2.0_{.9m,.1g}$
35	2_m	2_g	2_g	?	3_{2m}	2_g	?	2_m	2_g	2_g	
	2_m	2_m	2_g	2_m	2_m	2_m	2_m	3_{2m}	?	2_g	$2.12_{.63m,.37g}$
40	3_{2m}	3_{2m}	2_r	3_{2m}	$?2_g$	3_{2m}	3_{2m}	3_{2m}	3_{2m}	3_{2m}	
	3_{2m}	3_{2m}	3_{2m}	3_{2m}	3_{2m}	3_{2m}	3_{2m}	$3_{m,r}$	3_{2m}	3_{2m}	$2.95_{.92m,.03g,.05r}$
45	3_{2m}	$3_{g,m}$	3_{2m}	3_{2m}	$3_{m,g}$	3_{2m}	3_{2m}	$3_{m,g}$	3_{2m}	3_{2m}	
	$3_{g,m}$	$3_{r,m}$	3_{2m}	3_{2r}	3_{2m}	3_{2r}	3_{2m}	3_{2m}	3_{2m}	3_{2m}	$3.0_{.78m,.1g,.13r}$
50	4_{3m}	$4_{r,2m}$	4_{3m}	$3_{r,m}$	$3_{g,m}$	3_{2m}	4_{3m}	$4_{g,2m}$	3_{2m}	$3_{m,r}$	
	3_{2g}	$3_{m,g}$	4_{3m}	3_{2g}	$3_{m,r}$	3_{2g}	3_{2m}	4_{3m}	4_{3m}	3_{2m}	$3.4_{.73m,.19g,.08r}$
55	4_{3m}	4_{3m}	3_{2r}	3_{2g}	4_{3m}	4_{3m}	4_{3m}	$4_{g,2m}$	4_{3m}	$4_{g,m,g}$	
	?	3_{2g}	4_{3m}	4_{3m}	4_{3m}	$2_m?2_m$	4_{3m}	$4_{2m,g}$	4_{3m}	4_{3m}	$3.83_{.81m,.15g,.04r}$
60	4_{3m}	$4_{g,2m}$	4_{3m}	4_{3m}	4_{3m}	4_{3m}	4_{3m}	4_{3m}	4_{3m}	4_{3m}	
	?	4_{3m}	4_{3m}	4_{3m}	4_{3m}	4_{3m}	4_{3m}	$4_{m,2r}$	$4_{m,r,m}$	4_{3m}	$4.0_{.93m,.02g,.05r}$
65	$2_m?3_{2g}$	$2_m?2_m$	5_{4m}	$2_m?2_m$	$6_{g,m,g,2m}$	$5_{3m,g}$	4_{3m}	$4_{m,g,m}$	4_{3m}	5_{4m}	
	5_{4m}	$4_{m,r,m}$	5_{4m}	5_{4m}	$4_{m,2g}$	$4_{2m,g}$	$4_{r,g,m}$	$4_{g,m,g}$	$?2_g$	4_{3m}	$4.5_{.77m,.2g,.03r}$
70	5_{4m}	$5_{m,3r}$	5_{4m}	5_{4m}	$7_{m,g,m,g,2m}$	$2_m?$	5_{4m}	?	$2_m?2_g$	5_{4m}	
	5_{4m}	$5_{r,3m}$	5_{4m}	$2_m?$	$5_{2r,2m}$	5_{4m}	$?2_m$	$4_{r,2m}$	$5_{m,2r,m}$	5_{4m}	$5.07_{.82m,.05g,.14r}$
75	$5_{r,2m,g}$	$5_{2m,g,m}$	5_{4m}	$5_{m,g,2m}$	5_{4m}	6_{5m}	$5_{2m,r,m}$	$5_{g,3m}$	$5_{2m,g,m}$	6_{5m}	
	6_{5m}	5_{4m}	$5_{2r,2m}$	$2_m?3_{g,m}$	$5_{g,3m}$	$5_{m,2r,m}$	?	$5_{m,r,2m}$	$3_{r,m}?$	$5_{2m,2g}$	$5.18_{.77m,.12g,.11r}$
										Grand Total:	.82m, .11g, .07r

Table A.2: Number and Pairwise Symmetries of Learned Maps per Individual Run

R											average
11	1	1	1	1	1	1	1	1	1	1	1.0
15	1	1	1	1	1	1	1	1	1	1	1.0
20	1	1	?	1	1	1	1	1	1	1	1.0
25	?	2_m	2_m	2_m	2_m	2_m	2_m	?	2_m	2_m	$2.0_{1.0m}$
30	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_m	2_m	$2.0_{1.0m}$
35	2_m	?	2_m	2_m	3_{2m}	2_m	2_m	2_m	3_{2m}	2_m	$2.22_{1.0m}$
40	3_{2m}	3_{2m}	3_{2m}	3_{2m}	3_{2m}	3_{2m}	3_{2m}	3_{2m}	?	3_{2m}	$3.0_{1.0m}$
45	3_{2m}	3_{2m}	3_{2m}	3_{2m}	3_{2m}	3_{2m}	$3_{r,m}$	3_{2m}	3_{2m}	3_{2m}	$3.0_{.95m,.05r}$
50	3_{2m}	4_{3m}	3_{2m}	4_{3m}	4_{3m}	$3_{g,m}$	4_{3m}	$3_{m,g}$	3_{2m}	3_{2m}	$3.4_{.92m,.08g}$
55	$4_{m,r,m}$	4_{3m}	4_{3m}	4_{3m}	4_{3m}	4_{3m}	4_{3m}	4_{3m}	$4_{m,g,m}$	$2_m?2_m$	$4.0_{.94m,.03g,.03r}$
60	4_{3m}	$4_{g,2m}$	4_{3m}	4_{3m}	4_{3m}	$4_{m,g,m}$	$4_{2m,g}$	$?3_{g,m}$	$4_{m,2g}$	4_{3m}	$4.0_{.79m,.21g}$
65	5_{4m}	5_{4m}	5_{4m}	5_{4m}	5_{4m}	$4_{2m,r}$	4_{3m}	5_{4m}	$3_{m,r}?$	5_{4m}	$4.78_{.94m,.06r}$
70	5_{4m}	6_{5m}	5_{4m}	5_{4m}	$5_{m,g,2m}$	5_{4m}	5_{4m}	$5_{m,2g,m}$	6_{5m}	5_{4m}	$5.2_{.93m,.07g}$
75	6_{5m}	$5_{2m,g,m}$	5_{4m}	$5_{2m,2r}$	6_{5m}	$6_{m,g,r,2m}$	6_{5m}	$5_{m,2g,m}$	$5_{m,r,2m}$	6_{5m}	$5.5_{.82m,.09g,.09r}$

Grand Total: .91m, .06g, .03r

Appendix B

Sequential Training Data

The words (nouns) used in this work are derived from the Snodgrass-Vanderwart corpus (Snodgrass and Vanderwart, 1980) and their phonemes based on the NetTalk corpus (Berndt et al., 1994; Sejnowski and Rosenberg, 1987). The Snodgrass-Vanderwart corpus contains 260 names of physical objects (e.g., "apple"), from which we eliminated all multiword names (e.g., "spool of thread"), words for which, in experiments, subjects did not select the "correct" name for the corresponding picture at least 90% of the time (using % Corr(1) in (Snodgrass and Yuditsky, 1996)), and nouns that are not part of the NetTalk corpus. This leaves 175 nouns that we use as training data. The phoneme sequences corresponding to the selected nouns are taken from the NetTalk corpus. Altogether 27 consonants and 15 vowels and diphthongs occur in the NetTalk corpus, for a total of 42 phonemes. Three of the consonants, /ul/, /um/ and /un/, which rarely or never are part of a selected noun (14, 0 and 3 times), are not distinguished, but considered to be equivalent to /l/, /m/ and /n/.

Construction of distinctive feature vectors for each phoneme is challenging as sometimes experts in phonology/phonetics/linguistics disagree on what an ideal set of distinctive features should be (see, for example, (Frisch, 1996)). Our distinctive features were not based on any modeling considerations, but on well-known previously

published feature sets. They provide a unique representation for each distinguished phoneme that captures at least some of the regularities that make some phonemes similar to others. All 34 components of a feature vector (input pattern), prior to normalization, are binary valued: $+$ for a present feature (numerical value 1.0), and $-$ for an absent feature (0.0); see Tables B.1, B.2 and B.3. The *consonant features* were taken mostly from the Jakobson, Fant and Halle feature system (Jakobson et al., 1951), (Singh, 1976, pp. 34–40), augmented for completeness with additional phonemes (e.g., /r/) and features by S. Singh and colleagues (Singh and Black, 1966), (Singh, 1976, pp. 48–53). The *vowel features* include some of the same features as consonants, plus features based upon the F1 and F2 formants, each divided into six discrete frequency intervals (VH = very high, H = high, HM = high-medium, M = medium, LM = low-medium, L = low), taken from (Paget, 1976). The diphthongs such as /ai/ and /au/ were taken as the average of their two non-diphthong components for simplicity.

For normalization, each feature vector is projected onto the unit hypersphere in the next higher dimension. The additional component v_r stores the minimal distance between the original feature vector and the surface of the smallest hypersphere enclosing all feature vectors: $v_r = r - ||\vec{v}||_2$ where r is the length of the largest feature vector. The thus extended feature vectors are then normalized to unit length to prevent input vectors with relatively large norms from having a greater influence on the activation dynamics. The prior projection step preserves topological information such as the nearest neighbor relation between the vectors.

Table B.1: Distinctive Features: Vowels

IPA	o	a	e	u	ə	i	ɪ	ɛ	æ	ʌ	ʊ	ɔ	ɚ	ai	əʊ
Keyboard code	o	ah	ay	oo	uh-	ee	ih	eh	ae	uh+	u	aw	er	ai	au
Consonantal	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Vocalic	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Compact	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Diffuse	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Grave	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Acute	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Nasal	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Oral	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Tense	+	+	+	+	.	+	.	.	.	.	.	.	+	+	.
Lax	.	.	.	.	+	.	+	+	+	+	+	+	.	.	+
Continuant	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Interrupted	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Strident	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Mellow	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
+Voicing	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
–Voicing	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
+Duration	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
–Duration	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
+(Af)Frication	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
–(Af)Frication	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Liquid	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Glide	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Retroflex	.	.	.	.	.	.	.	.	.	.	.	.	+	.	.
$F_{2,VH}$	.	.	+	.	.	+	+	.	.	.	.	.	.	.	.
$F_{2,H}$	.	.	.	.	.	.	.	+	+	.	.	.	.	+	.
$F_{2,HM}$	.	.	.	.	+	.	.	.	.	+	.	.	+	.	.
$F_{2,LM}$	.	+	.	.	.	.	.	.	.	.	.	.	.	.	+
$F_{2,L}$	.	.	.	.	.	.	.	.	.	.	+	+	.	.	.
$F_{2,VL}/F_{1,VH}$	+	.	.	+	.	.	.	.	.	+	.	.	.	.	.
$F_{1,H}$	.	+	.	.	+	.	.	.	+	.	.	.	.	.	.
$F_{1,HM}$	.	.	.	.	.	.	.	+	.	.	.	+	.	.	.
$F_{1,LM}$	+	.	+	.	.	.	.	.	.	.	.	.	+	+	+
$F_{1,L}$	.	.	.	+	.	.	+	.	.	.	.	.	.	.	.
$F_{1,VL}$	.	.	.	.	.	+	.	.	.	.	+	.	.	.	.

Table B.2: Distinctive Features: Consonants, Part I

IPA	p	b	m	t	d	n	ʧ	ʤ	k	g	f	v	θ	ð
Keyboard code	p	b	m	t	d	n	tch	dj	k	g	f	v	th−	th+
Consonantal	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Vocalic	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Compact	.	.	.	.	.	.	+	+	+	+	.	.	.	.
Diffuse	+	+	+	+	+	+	.	.	.	.	+	+	+	+
Grave	+	+	+	.	.	.	.	.	.	.	+	+	.	.
Acute	.	.	.	+	+	+	.	.	.	.	.	.	+	+
Nasal	.	.	+	.	.	+	.	.	.	.	.	.	.	.
Oral	+	+	.	+	+	.	+	+	+	+	+	+	+	+
Tense	+	.	.	+	.	.	+	.	+	.	+	.	+	.
Lax	.	+	.	.	+	.	.	+	.	+	.	+	.	+
Continuant	.	.	.	.	.	.	.	.	.	.	+	+	+	+
Interrupted	+	+	.	+	+	.	+	+	+	+	.	.	.	.
Strident	.	.	.	.	.	.	+	+	.	.	.	.	.	.
Mellow	.	.	.	.	.	.	.	.	+	+	.	.	+	+
+Voicing	.	+	+	.	+	+	.	+	.	+	.	+	.	+
–Voicing	+	.	.	+	.	.	+	.	+	.	+	.	+	.
+Duration	.	.	.	.	.	.	.	.	.	.	.	.	.	.
–Duration	+	+	+	+	+	+	+	+	+	+	+	+	+	+
+(Af)Frication	.	.	.	.	.	.	+	+	.	.	+	+	+	+
–(Af)Frication	+	+	+	+	+	+	.	.	+	+	.	.	.	.
Liquid	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Glide	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Retroflex	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$F_{2,VH}$	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$F_{2,H}$	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$F_{2,HM}$	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$F_{2,LM}$	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$F_{2,L}$	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$F_{2,VL}/F_{1,VH}$	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$F_{1,H}$	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$F_{1,HM}$	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$F_{1,LM}$	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$F_{1,L}$	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$F_{1,VL}$	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Table B.3: Distinctive Features: Consonants, Part II

IPA	s	z	ʃ	ʒ	w	r	l	j	h	ŋ
Keyboard code	s	z	sh	zh	w	r	l	y	h	ng
Consonantal	+	+	+	+	+	+	+	+	+	+
Vocalic	.	.	.	.	.	+	+	.	.	.
Compact	.	.	+	+	.	.	.	.	.	+
Diffuse	+	+	.	.	.	.	.	.	.	.
Grave	.	.	.	.	.	.	.	.	.	.
Acute	+	+	.	.	.	.	.	.	.	.
Nasal	.	.	.	.	.	.	.	.	.	+
Oral	+	+	+	+	+	+	+	+	+	.
Tense	+	.	+	.	.	.	.	.	+	.
Lax	.	+	.	+	.	.	.	.	.	.
Continuant	+	+	+	+	+	.	.	.	.	.
Interrupted	.	.	.	.	.	.	.	.	.	.
Strident	+	+	.	.	.	.	.	.	.	.
Mellow	.	.	.	.	.	.	.	.	.	.
+Voicing	.	+	.	+	+	+	+	+	.	+
–Voicing	+	.	+	.	.	.	.	.	+	.
+Duration	+	+	+	+	.	.	.	.	.	.
–Duration	.	.	.	.	+	+	+	+	+	+
+(Af)Frication	+	+	+	+	.	.	.	.	+	.
–(Af)Frication	.	.	.	.	+	+	+	+	.	+
Liquid	.	.	.	.	.	+	+	.	.	.
Glide	.	.	.	.	+	.	.	+	.	.
Retroflex	.	.	.	.	.	+	.	.	.	.
$F_{2,VH}$	.	.	.	.	.	.	.	.	.	.
$F_{2,H}$	.	.	.	.	.	.	.	.	.	.
$F_{2,HM}$	.	.	.	.	.	.	.	.	.	.
$F_{2,LM}$	.	.	.	.	.	.	.	.	.	.
$F_{2,L}$	.	.	.	.	.	.	.	.	.	.
$F_{2,VL}/F_{1,VH}$	.	.	.	.	.	.	.	.	.	.
$F_{1,H}$	.	.	.	.	.	.	.	.	.	.
$F_{1,HM}$	.	.	.	.	.	.	.	.	.	.
$F_{1,LM}$	.	.	.	.	.	.	.	.	.	.
$F_{1,L}$	.	.	.	.	.	.	.	.	.	.
$F_{1,VL}$	.	.	.	.	.	.	.	.	.	.

Appendix C

Pairs of Confused Sequences

Table C.1: Pairs of Confused Sequences
30 by 20 map, 60 distinct training sequences

/h ɔ r s/ (horse)	/b a k s/ (box)
/n i d l/ (needle)	/i g l/ (eagle)
/ai r n/ (iron)	/k ɔ r n/ (corn)
/s w ɛ t ɚ r/ (sweater)	/h ɛ l ə k a p t ɚ r/ (helicopter)
/k ʌ p/ (cup)	/k o t/ (coat)
/k ʌ p/ (cup)	/k əʊ ʧ/ (couch)
/k əʊ ʧ/ (couch)	/k o t/ (coat)

40 by 30 map, 175 distinct training sequences

/b ɛ l/ (bell)	/b ɛ r/ (bear)
/f ɛ n s/ (fence)	/b a k s/ (box)
/f a k s/ (fox)	/b a k s/ (box)
/f l ai/ (fly)	/b ə t ɚ r f l ai/ (butterfly)
/p æ n t s/ (pants)	/f a k s/ (fox)
/f a k s/ (fox)	/f ɛ n s/ (fence)
/p æ n t s/ (pants)	/f ɛ n s/ (fence)
/t ai g ɚ r/ (tiger)	/s p ai d ɚ r/ (spider)
/p æ n t s/ (pants)	/b a k s/ (box)
/f l əʊ ɚ r/ (flower)	/f l ŋ g ɚ r/ (finger)
/p æ n t s/ (pants)	/æ k s/ (axe)

/k i / (key)
/b ɛ l/ (bell)
/b o/ (bow)
/b a t l/ (bottle)
/h æ m ɚ r/ (hammer)
/h ɔ r s/ (horse)
/k ai t/ (kite)
/h ɔ r s/ (horse)
/h æ m ɚ r/ (hammer)
/p æ n t s/ (pants)
/ʌ n y ə n/ (onion)
/h ɔ r s/ (horse)
/t ə m e t o/ (tomato)
/l æ d ɚ r/ (ladder)
/k ʌ p/ (cup)
/h ɔ r s/ (horse)
/s w ɛ t ɚ r/ (sweater)
/r u l ɚ r/ (ruler)
/t ai g ɚ r/ (tiger)
/t ai g ɚ r/ (tiger)
/s p ai d ɚ r/ (spider)
/t ai g ɚ r/ (tiger)
/k æ t/ (cat)
/b a k s/ (box)
/s p ai d ɚ r/ (spider)
/s w ɛ t ɚ r/ (sweater)
/s w ɛ t ɚ r/ (sweater)
/s w ɛ t ɚ r/ (sweater)
/s w ɛ t ɚ r/ (sweater)
/p ai n æ p l/ (pineapple)
/s w ɛ t ɚ r/ (sweater)
/t ai g ɚ r/ (tiger)

/d a ŋ k i / (donkey)
/b ɔ l/ (ball)
/b i / (bee)
/b i t l/ (beetle)
/f l ŋ g ɚ r/ (finger)
/b a k s/ (box)
/k o t/ (coat)
/f a k s/ (fox)
/f l ʊ ɚ r/ (flower)
/h ɔ r s/ (horse)
/l ai ə n/ (lion)
/æ k s/ (axe)
/p ə t e t o/ (potato)
/f l ʊ ɚ r/ (flower)
/k æ p/ (cap)
/f ɛ n s/ (fence)
/f l ʊ ɚ r/ (flower)
/f l ʊ ɚ r/ (flower)
/f l ʊ ɚ r/ (flower)
/r u l ɚ r/ (ruler)
/f l ʊ ɚ r/ (flower)
/f l ŋ g ɚ r/ (finger)
/k æ p/ (cap)
/æ k s/ (axe)
/r u l ɚ r/ (ruler)
/r u l ɚ r/ (ruler)
/f l ŋ g ɚ r/ (finger)
/t ai g ɚ r/ (tiger)
/s p ai d ɚ r/ (spider)
/æ p l/ (apple)
/h æ m ɚ r/ (hammer)
/h æ m ɚ r/ (hammer)

Bibliography

Abbott, L. and Blum, K. (1996). Functional significance of long-term potentiation for sequence learning and prediction. *Cerebral Cortex*, 6(3):406–16.

Ahissar, E. and Arieli, A. (2001). Figuring space by time. *Neuron*, 32:185–201.

Alhoniemi, E., Hollmen, J., Simula, O., and Vesanto, J. (1999). Process monitoring and modeling using the self-organizing map. *Integrated Computer-Aided Engineering*, 1(6):3–14.

Allman, J. (1981). Visual topography and function. In Woolsey, C., editor, *Cortical Sensory Organization: Multiple Visual Areas*. Humana Press, Clifton, NJ.

Allman, J. (1984). Evolution of neocortex. In Jones, E. and Peters, A., editors, *Cerebral Cortex*, volume 8A, pages 269–83. Plenum Press, New York.

Allman, J. and Kaas, J. (1971). A representation of the visual field in the caudal third of the middle temporal gyrus of the owl monkey. *Brain Research*, 31:85–105.

Andrade, M., Casari, G., Sander, C., and Valencia, A. (1997). Classification of protein families and detection of the determinant residues with an improved self-organizing map. *Biological Cybernetics*, 76(6):441–50.

Ayers, D. and Reggia, J. (2001). Evolving columnar circuitry for lateral cortical inhibition. In *Proc INNS-IEEE Int Joint Conf Neural Networks*, pages 278–83. IEEE Press.

Azzopardi, P. and Cowey, A. (1993). Preferential representation of the fovea in the primary visual cortex. *Nature*, 361(6414):719–21.

Barreto, G., Arajo, A., and Kremer, S. (2003). A taxonomy for spatiotemporal connectionist networks revisited: The unsupervised case. *Neural Computation*, 15(6):1255–320.

Bauer, H.-U. (1995). Development of oriented ocular dominance bands as a consequence of areal geometry. *Neural Computation*, 7(1):36–50.

Bauer, H.-U. and Pawelzik, K. (1992). Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks*, 3(4):570–9.

Beck, P., Pospichal, M., and Kaas, J. (1996). Topography, architecture, and connections of somatosensory cortex in opossums: Evidence for five somatosensory areas. *J Comparative Neurology*, 1(366):109–33.

Bednar, J. and Miikkulainen, R. (2000). Tilt aftereffects in a self-organizing model of the primary visual cortex. *Neural Computation*, 12(7):1721–40.

Berndt, R., D'Autrechy, C., and Reggia, J. (1994). Functional pronunciation units in English words. *J Experimental Psychology: Learning, Memory and Cognition*, 20:977–91.

Bhat, N. and McAvoy, T. (1990). Use of neural nets for dynamic modeling and control of chemical process systems. *Computers & Chemical Engineering*, 14(4–5):573–83.

Bi, G. and Poo, M. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *J Neuroscience*, 18(24):10464–72.

Bi, G. and Poo, M. (2001). Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience*, 24:139–66.

Bishop, C., Svensén, M., and Williams, C. (1998). GTM: The generative topographic mapping. *Neural Computation*, 10:215–34.

Braitenberg, V. and Schüz, A. (1991). *Anatomy of the Cortex*. Springer, Berlin.

Brown, M., Keynes, R., and Lumsden, A. (2001). *The Developing Brain*. Oxford University Press.

Callan, D., Kent, R., Roy, N., and Tasko, S. (1999). Self-organizing map for the classification of normal and disordered female voices. *J Speech, Language, and Hearing Research*, 42(2):355–66.

Carpinteiro, O. (1999). Hierarchical self-organizing map model for sequence recognition. *Neural Processing Letters*, 9(3):209–20.

Cervera, E. and del Pobil, A. (1999). A SOM-based sensing approach to robotic manipulation tasks. In Oja, E. and Kaski, S., editors, *Kohonen Maps*, pages 207–18. Elsevier, Amsterdam.

Chappell, G. and Taylor, J. (1993). The temporal Kohonen map. *Neural Networks*, 6:441–5.

Chen, Y. and Reggia, J. (1996). Alignment of coexisting cortical maps in a motor control model. *Neural Computation*, 8:731–55.

Cherniak, C. (1995). Neural component placement. *Trends in Neuroscience*, 18(12):522–7.

Cho, S. and Reggia, J. (1993). Learning competition and cooperation. *Neural Computation*, 5:242–59.

Cho, S. and Reggia, J. (1994). Map formation in proprioceptive cortex. *Int J Neural Systems*, 5(2):87–101.

Coello, C. (2001). A short tutorial on evolutionary multiobjective optimization. In *Evolutionary Multi-Criterion Optimization*, volume 1993 of *Lecture Notes in Computer Science*, pages 21–40. Springer.

Cohen, M. and Grossberg, S. (1983). Stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man and Cybernetics*, 13:815–26.

Cohen-Cory, S. (2002). The developing synapse: construction and modulation of synaptic structures and circuits. *Science*, 298:770–6.

Corne, D., Knowles, J., and Oates, M. (2000). The Pareto envelope-based selection algorithm for multiobjective optimization. In *Parallel Problem Solving from Nature*, volume 1917 of *Lecture Notes in Computer Science*, pages 839–48. Springer.

Cowey, A. (1981). Why are there so many visual areas? In Schmitt, F., editor, *The Organization of the Cerebral Cortex*, pages 395–413. MIT Press, Cambridge MA.

Creutzfeldt, O. (1978). The neocortical link: Thoughts on the generality of structure and function of the neocortex. In *Architectonics of the Cerebral Cortex*, pages 357–83. Raven Press, New York.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, New York.

Deb, K. (February 2004). Kalyanmoy deb answers a few questions about this month's fast breaking paper in field of Engineering. *Fast Breaking Papers*. `http://esi-topics.com/fbp/2004/february04-KalyanmoyDeb.html` (as of 6/16/04).

Deb, K. and Beyer, H. (1999). Self-adaptive genetic algorithms with simulated binary crossover. Technical Report CI-61/99, University of Dortmund, Department of Computer Science, Dortmund, Germany.

Deb, K. and Goyal, M. (1996). A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 26(4):30–45.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–97.

Deboeck, G. and Kohonen, T. (1998). *Visual Explorations in Finance with Self-Organizing Maps*. Springer-Verlag, London.

Dehay, C., Giroud, P., Berland, M., Killackey, H., and Kennedy, H. (1996). Contribution of thalamic input to the specification of cytoarchitectonic cortical fields in the primate: Effects of bilateral enucleation in the fetal monkey on the boundaries, dimensions, and gyrification of striate and extrastriate cortex. *J Comparative Neurology*, 367:70–89.

Deschenes, C. and Noonan, J. (1995). Fuzzy Kohonen network for the classification of transients using the wavelet transform for feature extraction. *Information Sciences*, 87(4):247–66.

Donoghue, J., Leibovic, S., and Sanes, J. (1992). Organization of the forelimb area in squirrel monkey motor cortex. *Experimental Brain Research*, 89:1–19.

Drager, U. (1975). Receptive fields of single cells and topography in mouse visual cortex. *J Comparative Neurology*, 160:269–90.

Dykes, R. and Ruest, A. (1984). What makes a map in somatosensory cortex? In Jones, E. and Peters, A., editors, *Cerebral Cortex*, volume 5, pages 1–29. Plenum Press, New York.

Elman, J. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

Engelien, A., Yang, Y., Engelien, W., Zonana, J., Stern, E., and Silbersweig, D. (2002). Physiological mapping of human auditory cortices with a silent event-related fMRI technique. *Neuroimage*, 4(16):944–53.

Euliano, N. and Principe, J. (1999). A spatio-temporal memory based on SOM's with activity diffusion. In Oja, E. and Kaski, S., editors, *Kohonen Maps*, pages 253–66. Elsevier, Amsterdam.

Faldella, E., Fringuelli, B., Passeri, D., and Rosi, L. (1997). A neural approach to robotic haptic recognition of 3-D objects based on a Kohonen self-organizing feature map. *IEEE Transactions on Industrial Electronics*, 44(2):267–9.

Ferrán, E. and Ferrara, P. (1991). Topological maps of protein sequences. *Biological Cybernetics*, 65(6):451–8.

Formisano, E., Kim, D., and Salle, F. (2003). Mirror-symmetric tonotopic maps in human primary auditory cortex. *Neuron*, 40:859–69.

Frank, Y. and Pavlakis, S. (2001). Brain imaging in neurobehavioral disorders. *Pediatr. Neurol.*, 25:278–287.

Frisch, S. (1996). *Similarity and Frequency in Phonology*. Ph.D. dissertation, Northwestern University, Evanston, USA.

Gaiddon, A., Knight, D., and Poloni, C. (2004). Multicriteria design optimization of a supersonic inlet based upon global missile performance. *J Propulsion and Power*, 20(3):542–58.

Gentilucci, M., Fogassi, L., Luppino, G., Matelli, M., Camarda, R., and Rizzolatti, G. (1989). Somatotopic representation in inferior area 6 of the macaque monkey. *Brain, Behavior and Evolution*, 2–3(33):118–21.

Georgopoulos, A., Kettner, R., and Schwartz, A. (1988). Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the directions of movement by a neural population. *J Neuroscience*, 8:2928–37.

Gerstner, W. (1995). Time structure of the activity in neural network models. *Physical Review E*, 51:738–58.

Gerstner, W., Ritz, R., and van Hemmen, J. (1993). Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns. *Biological Cybernetics*, 69(5–6):503–15.

Gilbert, S. (1994). *Developmental Biology*. Sinauer.

Göppert, J. and Rosenstiel, W. (1993). Topology-preserving interpolation in self-organizing maps. In *Proc Neuro-Nimes*, pages 425–34, Nanterre, France. EC2.

Grajski, K. and Merzenich, M. (1990). Neural network simulation of somatosensory representational plasticity. In Touretzky, D., editor, *Advances in Neural Information Processing Systems 2*, pages 52–9. Morgan Kaufmann.

Grove, E. and Tomomi, F. (2003). Generating the cerebral cortical area map. *Annual Review Neuroscience*, 26:355–80.

Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences*, chapter 11, Core String Edits, Alignments, and Dynamic Programming, pages 215–53. Cambridge University Press.

Hanke, J. and Reich, J. (1996). Kohonen map as a visualization tool for the analysis of protein sequences – multiple alignments, domains and segments of secondary structures. *Computer Applications in the Biosciences*, 12(6):447–54.

Hastie, T. and Stuetzle, W. (1989). Principal curves. *J American Statistical Association*, 84(406):502–16.

Haxby, J. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539):2425–30.

Hebb, D. (1949). *The Organization of Behavior: A Neuropsychological Theory*. John Wiley & Sons, New York.

Heikkonen, J. and Koikkalainen, P. (1997). Self-organization and autonomous robots. In Omidvar, O. and van der Smagt, P., editors, *Neural Systems for Robotics*, pages 297–337. Academic Press, San Diego, CA.

Hennessy, M. and Kelley, A. (2004). Using real-valued multi-objective genetic algorithms to model molecular absorption spectra and Raman excitation profiles in solution. *Physical Chemistry Chemical Physics*, 6(6):1085–95.

Henson, M. (1998). Nonlinear model predictive control: current status and future directions. *Computers & Chemical Engineering*, 23(2):187–202.

Hopfield, J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proc National Academy of Sciences USA*, 81:3088–92.

Hoshi, E. and Tanji, J. (2000). Integration of target and body-part information in the premotor cortex when planning action. *Nature*, 408(6811):466–70.

Hubel, D. and Wiesel, T. (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *J Physiology*, 160:106–54.

Hubel, D. and Wiesel, T. (1963). Receptive fields of cells in striate cortex of very young, visually inexperienced kittens. *J Neurophysiology*, 26:994–1002.

Hubel, D. and Wiesel, T. (1968). Receptive fields and functional architecture of monkey striate cortex. *J Physiology*, 195:215–43.

Hubel, D. and Wiesel, T. (1979). Brain mechanisms of vision. *Scientific American*, 241(3):150ff.

Imig, T., Reale, R., and Brugge, J. (1986). Topography of cortico-cortical connections related to tonotopic and binaural maps. In Lepore, F., editor, *Two Hemispheres - One Brain*, pages 103–15. Alan Liss, New York.

Jakobson, R., Fant, G., and Halle, M. (1951). *Preliminaries to Speech Analysis: the Distinctive Features and their Correlates*. MIT Press.

James, D. and Miikkulainen, R. (1995). SARDNET: a self-organizing feature map for sequences. In Tesauro, G., Touretzky, D., and Leen, T., editors, *Advances in Neural Information Processing Systems 7*, pages 577–84, Cambridge, MA, USA. MIT Press.

Jensen, M. (2003). Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms. *IEEE Transactions on Evolutionary Computation*, 7(5):503–15.

Jones, E. (1984). Connectivity of the primate sensory-motor cortex. In Jones, E. and Peters, A., editors, *Cerebral Cortex*, volume 5, pages 113–83. Plenum Press, New York.

Jones, E. (1990). Modulatory events in the development and evolution of primate neocortex. In Peters, A., editor, *Cerebral Cortex*, pages 311–51. Plenum.

Jordan, M. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In *Proc Conf Cognitive Science Society*, pages 531–46. Lawrence Erlbaum Associates.

Kaas, J. (1988). Why does the brain have so many cortical areas? *J Cognitive Neuroscience*, 1:121–34.

Kanazaki, M., Obayashi, S., and Nakahashi, K. (2004). Exhaust manifold design with tapered pipes using divided range moga. *Engineering Optimization*, 36(2):149–63.

Kangas, J. (1990). Time-delayed self-organizing maps. In *Proc Int Joint Conf Neural Networks*, volume II, pages 331–6, Los Alamitos, CA. IEEE Computing Society Press.

Kangas, J. (1991). Time-dependent self-organizing maps for speech recognition. In Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., editors, *Proc Artificial Neural Networks*, volume II, pages 1591–4, Amsterdam, Netherlands. North-Holland.

Kangas, J. (1992). Temporal knowledge in locations of activations in a self-organizing map. In Aleksander, I. and Taylor, J., editors, *Proc Artificial Neural Networks*, volume I, pages 117–20, Amsterdam, Netherlands. North-Holland.

Kaski, S., Honkela, T., Lagus, K., and Kohonen, T. (1998a). WEBSOM – self-organizing maps of document collections. *Neurocomputing*, 21(1):101–7.

Kaski, S., Kangas, J., and Kohonen, T. (1998b). Bibliography of self-organizing map (SOM) papers: 1981–1997. *Neural Computing Surveys*, 1(4):102–350. `ftp://ftp.icsi.berkeley.edu/pub/ai/jagota/vol1_4.pdf` (as of 6/4/2004).

Katic, D. and Vukobratovic, M. (2003). Survey of intelligent control techniques for humanoid robots. *J Intelligent & Robotic Systems*, 37(2):117–41.

Kennedy, J., Eberhart, R., and Shi, Y. (2001). *Swarm Intelligence*. Morgan Kaufmann, San Francisco.

Killackey, H., Rhoades, R., and Bennettclarke, C. (1995). The formation of a cortical somatotopic map. *Trends in Neurosciences*, 18(9):402–7.

Kiviluoto, K. (1996). Topology preservation in self-organizing maps. In *Proc Int Conf Neural Networks*, volume 1, pages 294–9. IEEE, New York, NY, USA.

Knowles, J. and Corne, D. (2000). Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–72.

Kohonen, T. (1981). Self-organized formation of generalized topological maps of observations in a physical system. Report TKK-F-A450, Helsinki University of Technology, Espoo, Finland.

Kohonen, T. (1982). Self-organizing formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

Kohonen, T. (1989). *Self-Organization and Associative Memory*. Springer, Berlin, 3rd edition.

Kohonen, T. (1991). The Hypermap architecture. In Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., editors, *Artificial Neural Networks*, volume II, pages 1357–60, Amsterdam, Netherlands. North-Holland.

Kohonen, T. (2001). *Self-Organizing Maps*. Springer, 3rd edition.

Kohonen, T., Mäkisara, K., and Saramäki, T. (1984). Phonotopic maps – insightful representation of phonological features for speech recognition. In *Proc Int Conf Pattern Recognition*, pages 182–5, Los Alamitos, CA. IEEE Computer Society Press.

Kokkonen, M. and Torkkola, K. (1990). Using self-organizing maps and multi-layered feed-forward nets to obtain phonemic transcriptions of spoken utterances. *Speech Communication*, 9(5–6):541–9.

Kopecz, K. (1995). Unsupervised learning of sequences on maps with lateral connectivity. In Fogelman-Soulié, F. and Gallinari, P., editors, *Proc Int Conf Artificial Neural Networks*, volume 1, pages 431–6, Nanterre, France. EC2 & Cie.

Koskela, T., Varsta, M., Heikkonen, J., and Kaski, K. (1998). Recurrent SOM with local linear models in time series prediction. In *Proc European Symposium on Artificial Neural Networks*, pages 167–72, Brussels, Belgium. D-Facto.

Krubitzer, L. (1995). The organization of neocortex in mammals. *Trends in Neuroscience*, 18:408–417.

Krubitzer, L. and Calford, M. (1992). Five topographically organized fields in the somatosensory cortex of the flying fox: microelectrode maps, myeloarchitecture, and cortical modules. *J Comparative Neurology*, 1(317):1–30.

Krubitzer, L., Clarey, J., Tweedale, R., Elston, G., and Calford, M. (1995). A redefinition of somatosensory areas in the lateral sulcus of macaque monkeys. *J Neuroscience*, 5(15):3821–39.

Kun, A. and Miller, W. (1999). Control of variable-speed gaits for a biped robot. *IEEE Robotics & Automation Magazine*, 6(3):19–29.

Lambrinos, D., Scheier, C., and Pfeifer, R. (1995). Unsupervised classification of sensory-motor states in a real world artifact using a temporal Kohonen map. In Fogelman-Soulié, F. and Gallinari, P., editors, *Proc Int Conf Artificial Neural Networks*, volume II, pages 467–72, Nanterre, France. EC2.

Levitt, P. (2000). Molectular determinants of regionalization of the forebrain and cerebral cortex. In Gazzaniga, M., editor, *The New Cognitive Neurosciences*, pages 23–43. MIT Press.

Li, Z. (2002). A saliency map in primary visual cortex. *Trends in Cognitive Sciences*, 6(1):9–16.

Lonai, P. (1996). *Mammalian Development*. Harwood.

Manduca, A. (1996). Multispectral image visualization with nonlinear projections. *IEEE Transactions on Image Processing*, 5(10):1486–90.

Markram, H., Luebke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, 275(5297):213–5.

Marseguerra, M., Zio, E., and Podofillini, L. (2004). A multiobjective genetic algorithm approach to the optimization of the technical specifications of a nuclear safety system. *Reliability Engineering & System Safety*, 84(1):87–99.

Martinetz, T., Ritter, H., and Schulten, K. (1989). Kohonen's self-organizing map for modeling the formation of the auditory cortex of a bat. In Pfeifer, R., Schreter, Z., Fogelman-Soulié, F., and Steels, L., editors, *Connectionism in Perspective*, pages 403–12. North-Holland, Amsterdam, Netherlands.

Merzenich, M., Kaas, J., Sur, M., and Lin, C. (1978). Double representation of the body surface within cytoarchitectonic areas 3b and 1 in "S1" in the owl monkey (aotus trivirgatus). *J Comparative Neurology*, 181(1):41–73.

Miikkulainen, R. (1991). Self-organizing process based on lateral inhibition and synaptic resource redistribution. In Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., editors, *Artificial Neural Networks*, volume I, pages 415–20, Amsterdam, Netherlands. North-Holland.

Miller, K. (2003). Understanding layer 4 of the cortical circuit: A model based on cat V1. *Cerebral Cortex*, 13(1):73–82.

Morasso, P. (1991). Self-organizing feature maps for cursive script recognition. In Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., editors, *Artificial Neural Networks*, volume II, pages 1323–6, Amsterdam, Netherlands. North-Holland.

Morris, R., Rubin, L., and Tirri, H. (1990). Neural network techniques for object orientation detection. Solution by optimal feedforward network and learning vector quantization approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1107–15.

Mountcastle, V. (1998). *The Cerebral Cortex*. Harvard University Press.

Mozer, M. (1989). A focused backpropagation algorithm for temporal pattern recognition. *Complex Systems*, 3(4):349–81.

Mozer, M. (1993). Neural network architectures for temporal sequence processing. In Weigend, A. and Gershenfeld, N., editors, *Time Series Prediction*, pages 243–64. Addison Wesley.

Munoz, A. and Muruzabal, J. (1998). Self-organizing maps for outlier detection. *Neurocomputing*, 18(1–3):33–60.

Nelson, R., Sur, M., Felleman, D., and Kaas, J. (1980). Representations of the body surface in postcentral parietal cortex of macaca fascicularis. *J Comparative Neurology*, 4(192):611–43.

Newsome, W., Maunsell, J., and van Essen, D. (1986). Ventral posterior visual area of the macaque: visual topography and areal boundaries. *J Comparative Neurology*, 2(252):139–53.

Obermayer, K., Blasdel, G., and Schulten, K. (1992a). A statistical mechanical analysis of self-organization and pattern formation during the development of visual maps. *Physical Review A*, 45(10):7568–89.

Obermayer, K., Ritter, H., and Schulten, K. (1990). A principle for the formation of the spatial structure of cortical feature maps. *Proc National Academy of Sciences USA*, 87:8345–9.

Obermayer, K., Schulten, K., and Blasdel, G. (1992b). A comparison of a neural network model for the formation of brain maps with experimental data. In Moody, J., Hanson, S., and Lippmann, R., editors, *Advances in Neural Information Processing Systems 4*, pages 83–90. Morgan Kaufmann, San Mateo, CA.

Ohki, K., Matsuda, Y., Ajima, A., Kim, D., and Tanaka, S. (2000). Arrangement of orientation pinwheel centers around area 17/18 transition zone in cat visual cortex. *Cerebral Cortex*, 10(6):593–601.

Oja, M., Kaski, S., and Kohonen, T. (2003). Bibliography of self-organizing map (SOM) papers: 1998–2001 addendum. *Neural Computing Surveys*, 3(1):1–156. `http://www.soe.ucsc.edu/NCS/VOL3/vol3_1.pdf` (as of 6/4/2004).

Paget, R. (1976). Vowel resonances. In Fry, D., editor, *Acoustic Phonetics*, pages 95–103. Cambridge University Press.

Palakal, M., Murthy, U., Chittajallu, S., and Wong, D. (1995). Tonotopic representation of auditory responses using self-organizing maps. *Mathematical and Computer Modelling*, 22(2):7–21.

Pantev, C., Bertrand, O., Eulitz, C., Verkindt, C., Hampson, S., Schuierer, G., and Elbert, T. (1995). Specific tonotopic organizations of different areas of the human auditory cortex revealed by simultaneous magnetic and electric recordings. *Electroencephalography and Clinical Neurophysiology*, 1(94):26–40. Journal incorporated into *Clinical Neurophysiology*.

Papanicolaou, A., Simos, P., Breier, J., and et al. (2003). Brain mechanisms for reading in children with and without dyslexia. *Dev. Neuropsychol.*, 24:593–612.

Pareto, V. (1896). *Cours d' Economie Politique*. F. Rouge & Cie., Lausanne, Switzerland.

Pearson, J., Finkel, L., and Edelman, G. (1987). Plasticity in the organization of adult cerebral cortical maps: A computer simulation based on neuronal group selection. *J Neuroscience*, 7:4209–23.

Pei, X., Vidyasagar, T., Volgushev, M., and Creutzfeldt, O. (1994). Receptive field analysis and orientation selectivity of postsynaptic potentials of simple cells in car visual cortex. *J Neuroscience*, 11(14):7130–40.

Penfield, W. and Rasmussen, T. (1950). *The Cerebral Cortex of Man: a clinical study of localization of function*. Macmillan, New York.

Pineda, F. (1987). Generalization of back-propagation to recurrent neural networks. *Physical Review Letters*, 59:2229–32.

Pinto, D., Hartings, J., Brumberg, J., and Simons, D. (2003). Cortical damping: Analysis of thalamocortical response transformations in rodent barrel cortex. *Cerebral Cortex*, 13(1):33–44.

Principe, J., Wang, L., and Motter, M. (1998). Local dynamic modeling with self-organizing maps and applications to nonlinear system identification and control. *Proceedings of the IEEE*, 86(11):2240–58.

Rakic, P., Suner, I., and Williams, R. (1991). A novel cytoarchitectonic area induced experimentally within the primate visual cortex. *Proc National Academy of Sciences USA*, 13:2083–7.

Rao, R. and Sejnowski, T. (2000). Predictive learning of temporal sequences in recurrent neocortical circuits. In Solla, S., Leen, T., and Muller, K., editors, *Advances*

in Neural Information Processing Systems 12, pages 164–71, Cambridge, MA. MIT Press.

Rauschecker, J., Tian, B., and Hauser, M. (1995). Processing of complex sounds in the macaque nonprimary auditory cortex. *Science*, 268:111–4.

Reggia, J. (1989). Methods for deriving competitive activation mechanisms. In *Proc Int Joint Conf Neural Networks*, pages 357–63.

Reggia, J., D'Autrechy, C., Sutton, G., and Weinrich, M. (1992). A competitive distribution theory of neo-cortical dynamics. *Neural Computation*, 4:287–317.

Reggia, J., Goodall, S., and Levitan, S. (2001). Cortical map asymmetries in the context of transcallosal excitatory influences. *Neuroreport*, 13(8):1609–14.

Riesenhuber, M. and Poggio, T. (2002). Neural mechanisms of object recognition. *Current Opinion in Neurobiology*, 12(2):162–8.

Ritter, H., Martinetz, T., and Schulten, K., editors (1992). *Neural Computation and Self-Organizing Maps*. Addison Wesley.

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

Ritter, H. and Schulten, K. (1988). Convergence properties of Kohonen's topology preserving maps: fluctuations, stability, and dimension selection. *Biological Cybernetics*, 60(1):59–71.

Roberts, P. (1999). Computational consequences of temporally asymmetric learning rules: I. differential hebbian learning. *J Computational Neuroscience*, 7(3):235–46.

Roe, A. and Ts'o, D. (1995). Visual topography in primate V2: multiple representation across functional stripes. *J Neuroscience*, 15:3689–715.

Royer, S. and Pare, D. (2003). Conservation of total synaptic weight through balanced synaptic depression and potentiation. *Nature*, 422:518–522.

Rumelhart, D., Hinton, G., and Williams, R. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–6.

Sahyoun, C., Floyer-Lea, A., Johansen-Berg, H., and Matthews, P. (2004). Towards an understanding of gait control: brain activation during the anticipation, preparation and execution of foot movements. *Neuroimage*, 21(2):568–75.

Sammon, J. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–9.

Schrater, P., Knill, D., and Simoncelli, E. (2000). Mechanisms of visual motion detection. *Nature Neuroscience*, 3(1):64–8.

Schreibmann, E., Lahanas, M., Xing, L., and Baltas, D. (2004). Multiobjective evolutionary optimization of the number of beams, their orientations and weights for intensity-modulated radiation therapy. *Physics in Medicine and Biology*, 49(5):747–70.

Schuchhardt, J. (1996). Local structural motifs of protein backbones are classified by self-organizing neural networks. *Protein Engineering*, 9(10):833–42.

Sejnowski, T. and Rosenberg, C. (1987). Parallel networks that learn to pronounce English text. *Complex Systems*, 1:145–68.

Sereno, M., Dale, A., Reppas, J., Kwong, K., Belliveau, J., Brady, T., Rosen, B., and Tootell, R. (1995). Borders of multiple visual areas in humans revealed by functional magnetic resonance imaging. *Science*, 268(5212):889–93.

Singh, S. (1976). *Distinctive Features: Theory and Validation*. University Park Press.

Singh, S. and Black, J. (1966). A study of twenty-six intervocalic consonants as spoken and recognized by four language groups. *J Acoustic Society of America*, 39(2):372–87.

Sirosh, J. and Miikkulainen, R. (1992). Self-organization with lateral connections. Technical Report AI92–191, The University of Texas at Austin, Austin, TX.

Sirosh, J. and Miikkulainen, R. (1994). Cooperative self-organization of afferent and lateral connections in cortical maps. *Biological Cybernetics*, 71:65–78.

Snodgrass, J. and Vanderwart, M. (1980). A standardized set of 260 pictures. *J Experimental Psychology: Human Learning and Memory*, 6:174–215.

Snodgrass, J. and Yuditsky, T. (1996). Naming times for the Snodgrass and Vanderwart pictures. *Behavior Research Methods, Instruments and Computers*, 28(4):516–36.

Somervuo, P. (1999). Time topology for the self-organizing map. In *Proc Int Joint Conf Neural Networks*, volume 3, pages 1900–5, Piscataway, NJ. IEEE Service Center.

Somervuo, P. (2003). Speech dimensionality analysis on hypercubical self-organizing maps. *Neural Processing Letters*, 17(2):125–36.

Song, S., Miller, K., and Abbott, L. (2000). Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9):919–926.

Stepniewska, I. and Kaas, J. (1996). Topographic patterns of V2 cortical connections in macaque monkeys. *J Comparative Neurology*, 371:129–52.

Strobel, G. (2000). Genes or environment: what shapes the sensory Homunculus? *Focus – News from Harvard Medical, Dental, & Public Health Schools*. `http://focus.hms.harvard.edu/2000/Apr7_2000/cell_biology.html` (as of 6/10/04).

Stuhlman, O. (1952). *An Introduction to Biophysics*. John Wiley & Sons, New York, 3rd edition.

Sur, M. and Leamey, C. (2001). Development and plasticity of cortical areas and networks. *Nature Reviews Neuroscience*, 2:251–62.

Sur, M., Nelson, R., and Kaas, J. (1982). Representations of the body surface in cortical areas 3b and 1 of squirrel monkeys: comparisons with other primates. *J Comparative Neurology*, 2(211):177–92.

Sutton, G., Reggia, J., Armentrout, S., and D'Autrechy, C. (1994). Cortical map reorganization as a competitive process. *Neural Computation*, 6:1–13.

Takacs, B. and Wechsler, H. (1997). Detection of faces and facial landmarks using iconic filter banks. *Pattern Recognition*, 30(10):1623–36.

Talavage, T., Ledden, P., and Benson, R. (2000). Frequency-dependent responses exhibited by multiple regions in human auditory cortex. *Hearing Research*, 150:225–44.

Tan, K., Lee, T., and Khor, E. (2002). Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons. *Artificial Intelligence Review*, 17(4):253–90.

Temple, E., Deutsch, G., Poldrachk, R., and et al. (2003). Neural deficits in children with dyslexia ameliorated by behavioral remediation. *Proc. Nat. Acad. Sci.*, 100:2860–2865.

Tiao, Y. and Blakemore, C. (1976). Functional organization in the visual cortex of the golden hamster. *J Comparative Neurology*, 4(168):459–81.

Toivanen, P., Ansamaki, J., Parkkinen, J., and Mielikainen, J. (2003). Edge detection in multispectral images using the self-organizing map. *Pattern Recognition Letters*, 24(16):2987–94.

Tootell, R., Silverman, M., Switkes, E., and de Valois, R. (1982). Deoxyglucose analysis of retinotopic organization in primate striate cortex. *Science*, 218(4575):902–4.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 84:327–52.

Tversky, A. and Gati, I. (1978). Studies of similarity. In Rosch, E. and Lloyd, B., editors, *Judgement under Uncertainty: Heuristics and Biases*. Earlbaum.

van Essen, D., Newsome, W., Maunsell, J., and Bixby, J. (1986). The projections from striate cortex (V1) to areas V2 and V3 in the macaque monkey: asymmetries, areal boundaries, and patchy connections. *J Comparative Neurology*, 244:451–80.

Varsta, M., Millan, J., and Heikkonen, J. (1997). A recurrent self-organizing map for temporal sequence processing. In Gerstner, W., Germond, A., Hasler, M., and Nicoud, J., editors, *Proc Int Conf Artificial Neural Networks*, pages 421–6. Springer, Berlin.

Vesanto, J. (1999). SOM-based data visualization methods. *Intelligent Data Analysis*, 3(2):111–26.

Villmann, T. (1999). Topology preservation in self-organizing maps. In Oja, E. and Kaski, S., editors, *Kohonen Maps*, pages 279–92. Elsevier, Amsterdam.

Villmann, T., Der, R., Herrmann, M., and Martinetz, T. (1997). Topology preservation in self-organizing feature maps: exact definition and measurement. *IEEE Transactions on Neural Networks*, 8(2):256–66.

von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85–100.

von der Malsburg, C. and Willshaw, D. (1976). A mechanism for producing continuous neural mappings: ocularity dominance stripes and ordered retino-tectal projections. *Experimental Brain Research*, 1:463–69.

Welker, W. (1990). Why does the cerebral cortex fissure and fold? In Jones, E. and Peters, A., editors, *Cerebral Cortex*, volume 5, pages 3–136. Plenum Press, New York.

Wiemer, J. (2003). The time-organized map algorithm: extending the self-organizing map to spatiotemporal signals. *Neural Computation*, 15(5):1143–71.

Yin, H. (2003). Nonlinear multidimensional data projection and visualisation. In *Intelligent Data Engineering and Automated Learning*, volume 2690 of *Lecture Notes in Computer Science*, pages 377–88. Springer.

Young, M. (1992). Objective analysis of the topological organization of the primate cortical visual system. *Nature*, 358:152–5.

Zhang, L., Tao, H., Holt, C., Harris, W., and Poo, M. (1998). A critical window for cooperation and competition among developing retinotectal synapses. *Nature*, 395:37–44.

Zhou, R. and Black, I. (2000). Development of neural maps. In Gazzaniga, M., editor, *The New Cognitive Neurosciences*, pages 213–36. MIT Press.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–71.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., and da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–32.