SELF-ORGANIZING PROCESS BASED ON LATERAL INHIBITION AND SYNAPTIC RESOURCE REDISTRIBUTION *

Risto Miikkulainen
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188, U.S.A.
risto@cs.utexas.edu

Self-organizing feature maps are usually implemented by abstracting the low-level neural and parallel distributed processes. An external supervisor finds the unit whose weight vector is closest in Euclidian distance to the input vector and determines the neighborhood for weight adaptation. The weights are changed proportional to the Euclidian distance. In a biologically more plausible implementation, similarity is measured by a scalar product, neighborhood is selected through lateral inhibition and weights are changed by redistributing synaptic resources. The resulting self-organizing process is quite similar to the abstract case. However, the process is somewhat hampered by boundary effects and the parameters need to be carefully evolved. It is also necessary to add a redundant dimension to the input vectors.

1 Introduction

The self-organizing feature mapping [5; 7] is an unsupervised learning process where the processing units of e.g. 2-D laminar network become sensitive to specific items of the input space in a topological order which corresponds to the topological order of the input items. The process is both a method for organizing complex empirical knowledge and a model of learning in biological neural networks.

The theory of self-organizing feature maps is fairly well understood and demonstrated [7; 8; 4; 17] and a number of applications and extensions of feature maps have also been developed [9; 10; 11; 12; 13; 15; 16]. These implementations are based on an abstraction of the theory, which relies on global supervision and strong computational capabilities on the processing units. An essential part of the theory is that self-organization is brought about by a parallel and distributed process on a network of neuron-like elements. A plausible implementation must be based on local computations, simple enough to be carried out by neurons.

Kohonen has suggested that lateral inhibition and redistribution of synaptic resources could be responsible for self-organization in biological systems [6]. An implementation of this idea is discussed and simulation results are presented in this paper.

2 Self-organizing feature maps

A 2-D topological feature map consists of an array of processing units, each with n weight parameters. Each unit produces one output value, proportional to the similarity of the map's current input vector and the unit's weight vector. The total response of the map is a localized pattern of activity. In other words, an n-dimensional input vector is mapped onto a location in the 2-D map. The weight vectors are tuned to specific items of the input space so that topological relations are retained. This means roughly that nearby vectors in the input space are mapped onto nearby units in the map.

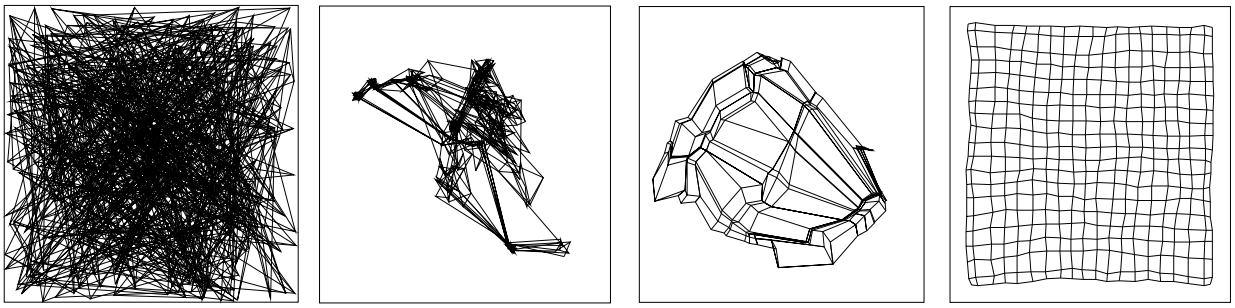
The organization of the map, i.e. the assignment of the weight vectors, is formed in an unsupervised learning process. Input items are randomly drawn from the input distribution and presented to the network one at a time. The weight vector of the maximally responding unit and each unit in its neighborhood are changed towards the input vector. These units now produce an even stronger response to the same input. In the process, weight vectors become better approximations of the input distribution and neighboring vectors become more parallel, which over time results in global order.

Each adaptation step consists of three computational tasks: (1) measuring the similarity of the input vector and the unit's weight vector, (2) determining the adapting neighborhood, and (3) changing the weights in this neighborhood. These tasks can be efficiently implemented by abstracting their neural and parallel distributed nature.

3 Abstract implementation

Self-organization can be efficiently implemented based on Euclidian distance and global supervision. It is not necessary to explicitly model the connections between the units in the network. Every unit computes the distance between its weight vector and the input vector. An external supervisor finds the unit with the smallest distance, looks up the current neighborhood radius from a training schedule, and tells the units within this radius to modify their input weights. The weight adaptations are proportional to the Euclidian difference. The weights of unit (i, j) in a 2-D map are

*To appear in *Proceedings of the International Conference on Artificial Neural Networks (ICANN-91)*, 1991



(a) 0 samples (b) 30 samples (c) 100 samples (d) 10,000 samples

Figure 1: **Abstract implementation of self-organization.** The map consists of 20×20 units in a 2-D array organization. The weight vector of each unit is shown as a point on the unit square $0 \leq x, y \leq 1$. Each vector is connected with a line to the weight vectors of the four neighboring units. In other words, each intersection and end point of lines represents a weight vector, and the grid represents the topological organization of the units. The figures depict organization after 0, 30, 100, and 10,000 input vector presentations. The neighborhood radius was decreased from 7 to 2 in 1000 presentations, and from 2 to 0 during the rest of the simulation. At the same time, the learning rate α was decreased from 0.3 to 0.05, and then to 0.0.

changed according to

$$\Delta\mu_{ij,h} = \begin{cases} \alpha[\xi_h - \mu_{ij,h}] & \text{if } (i,j)\epsilon N_c \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\mu_{ij,h}$ is the hth weight vector component and ξ_h is the hth input vector component, α is the gain parameter and N_c stands for the selected neighborhood. The gain and the neighborhood size gradually decrease over time.

Evolution of the weight vectors in this process is visualized in figure 1. The input data consists of 2-D vectors uniformly distributed on the unit square. Since dimensionality is not reduced in this mapping, the organization of the map can be visualized by plotting the 2-D weight vectors directly on the input space.

The weight vectors are initially uniformly distributed on the unit square (figure 1a). In the beginning of the process, a number of units are clustered together (1b). The clusters are then ordered, and individual weight vectors start to separate from the clusters (1c). As the neighborhood size decreases, the weight vectors gradually spread out to fill the whole input space (1d). In the final map, each unit is responsible for an approximately equal area. The map is slightly contracted at the boundaries, because the units near the boundary do not have complete neighborhoods.

4 "Biological" implementation

Self-organization with the abstract implementation is strong and efficient to simulate, and therefore well-suited for applications. However, it does not explain how self-organization could arise in biological systems. Below, Euclidian distance and global supervision are replaced by local computations compatible with the weighted sum model of the neuron, and the resulting adjustments to the self-organization process are discussed.

4.1 Similarity and neighborhood selection

The weighted sum of the input components, i.e. the scalar product of the input vector and the weight vector, is by itself a simple measure of similarity (with

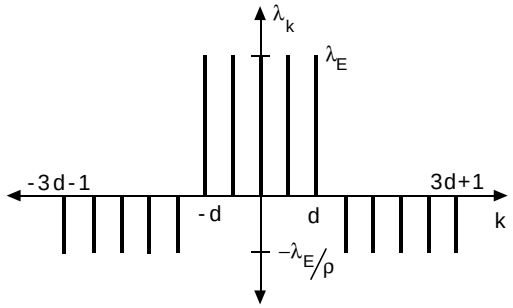


Figure 2: **Lateral inhibition coefficients γ_k (one-dimensional case).** The height of the bar indicates the weight on the connection from unit k to unit 0 (relative indexing). The same connectivity pattern applies to every unit in the network.

certain restrictions; see section 4.3). The initial response η_{ij} of unit (i,j) to an external input vector can be computed as

$$\eta_{ij} = \sigma\left(\sum_h \mu_{ij,h}\xi_h\right) \quad (2)$$

where the function σ is the familiar sigmoid activation function, or its piecewise linear approximation

$$\sigma(x) = \begin{cases} 0 & x \leq \delta \\ (x-\delta)/(\beta-\delta) & \delta < x < \beta \\ 1 & x \geq \beta \end{cases} \quad (3)$$

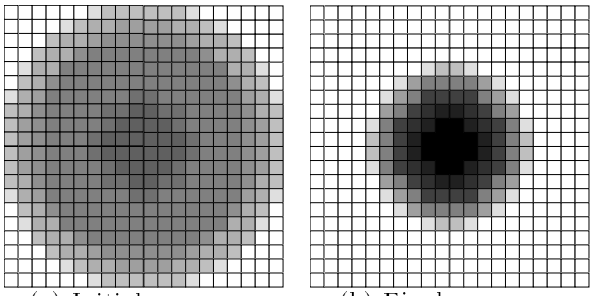
The sigmoid introduces a nonlinearity (a soft threshold between δ and β) into the reponse, and limits its output within the range $[0,1]$.

The neighborhood is selected by focusing the initial response of the map through lateral inhibition. The response evolves over time according to

$$\eta_{ij}(t) = \sigma\left(\sum_h \mu_{ij,h}\xi_h + \sum_{k,l} \gamma_{kl,ij}\,\eta_{kl}(t-\Delta t)\right) \quad (4)$$

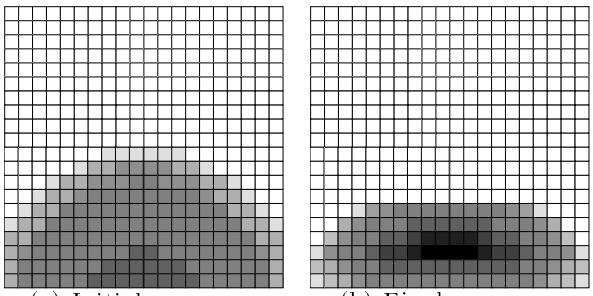
where $\gamma_{kl,ij}$ is the lateral connection weight on the connection from unit (k,l) to unit (i,j), and $\eta_{kl}(t-\Delta t)$ is the activity of unit (k,l) during the previous time step.

The weights on the lateral connections have the form of a "Mexican hat" (figure 2). The connections



(a) Initial response (b) Final response

Figure 3: **Focusing the response through lateral inhibition.** The darkness of each square indicates the activity level of the corresponding unit in the 20×20 map.



(a) Initial response (b) Final response

Figure 4: **Focusing the response at the boundary.**

to the closest units are excitatory and to units further away are inhibitory. In our simulations,

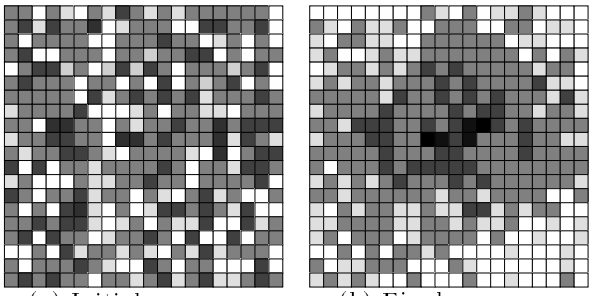
$$\gamma_{kl,ij} = \begin{cases} \gamma_E & |k-i|, |l-j| \leq d \\ -\frac{\gamma_E}{\rho} & (|k-i| > d \text{ or } |l-j| > d) \\ & \text{and } |k-i|, |l-j| \leq 3d+1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The parameter d controls the width of the interaction, γ_E its strength and ρ the ratio of excitation/inhibition.

Lateral inhibition is a biologically plausible form of lateral connectivity, well documented in e.g. low-level vision [1; 14]. The lateral weights are assumed to be preassigned, and their task is to support self-organization of the weights on the external input connections. The self-organizing process does not directly modify the lateral connections.

The primary effect of lateral inhibition is to sharpen the contrast between high and low activity areas. If the width of the lateral inhibition mask is comparable to the initial localized response of the network, the response becomes more focused around the maximally responding unit (or area) in successive iterations of equation 4. The more lateral inhibition there is compared to the lateral excitation, the more focused is the final stable response.

Figures 3, 4 and 5 illustrate the lateral inhibition process. The network consists of 20×20 units with interactions described by equations 3 through 5. In figures 3 and 4, the input weights are perfectly ordered, i.e. they correspond to the unit coordinates. The sigmoid parameters are $\delta = 0.88$, $\beta = 1.25$ and the lateral interaction parameters $d = 2$, $\gamma_E = 0.025$, $\rho = 5.0$. The external input to the network is held constant and the network is allowed to settle until it



(a) Initial response (b) Final response

Figure 5: **Focusing the response of an unordered map.**

reaches a stable state. Usually this takes 10 to 20 iterations. Figure 3 shows how the response to an input vector located at the center of the input space is focused by the network.

Figure 4 demonstrates boundary effects. The lateral inhibition mask for units close to the boundary extends outside the network, which has zero activity. As a result, the strongest activity develops at distance $d+1$ from the boundary. This is because the units at the $d+1$:th row or column receive all the lateral excitation but only about half the inhibition, while the units at the boundary receive half of both.

Figure 5 illustrates focusing the response of a completely unordered map. The initial pattern is highly discontinuous. A mask larger than the network ($d = 4, \gamma_E = 0.0075, \rho = 6.0$) sees the whole network as one large activity cluster and concentrates the response around the center.

4.2 Weight adaptation

According to a well-known hypothesis due to Hebb [2], permanent synaptic efficacy changes require both presynaptic and postsynaptic activity. The connection is significant and deserves to be enforced if the activity causes further activity. This principle can be formulated as:

$$d\mu/dt = \alpha\eta\xi \tag{6}$$

where μ is the efficacy of the synapse, ξ is the presynaptic (input) activity, η is the postsynaptic (output) activity and α is the gain of the change.

The weight changes in the Hebbian model are strictly non-negative. The model can be augmented by requiring that the sum of the synaptic resources of the neuron remains constant in the weight change. In other words, the resources are redistributed proportionally to their use. Assuming that the synaptic efficacies are directly proportional to the amount of resource at the synapse,

$$\mu_{ij,h}(t+\Delta t) = \frac{\mu_{ij,h}(t) + \alpha\eta_{ij}\xi_h}{\sum_h [\mu_{ij,h}(t) + \alpha\eta_{ij}\xi_h]} \tag{7}$$

where η_{ij} stands for the activity of the unit in the final stable state.

Equation 7 implies that only those units which are active do change. The cluster of activity brought about by lateral inhibition selects the area of the maximal response, and the weights of the units within this area change proportionally to their activity. Note that

in general the units most similar to the input vector change the most. This is contrary to the abstract implementation, where the vectors are adapted proportional to the Euclidian difference.

4.3 Self-organization

Bringing the above mechanisms for neuron activation and synaptic weight change together is not sufficient for self-organization. Two additional restrictions must be taken into account:

(1) The scalar product as a similarity measure requires that all weight vectors are of the same length. For example, if the input vector is (0.75 0.25), a unit with exactly the same weight vector responds with $\sigma(0.75^2 + 0.25^2) = \sigma(0.625)$, whereas a unit with e.g. weights (1.0 0.0) outputs $\sigma(0.75)$. The weight vectors must be normalized:

$$\mu_{ij,h}(t + \Delta t) = \frac{\mu_{ij,h}(t) + \alpha\eta_{ij}\xi_h}{\left\{\sum_h \left[\mu_{ij,h}(t) + \alpha\eta_{ij}\xi_h\right]^2\right\}^{1/2}} \quad (8)$$

In effect, the weight vector is rotated towards the input vector. The sum of squares of weight components is constant, i.e. the synaptic efficacy is not directly proportional to the resource being redistributed, but proportional to the square root of it. Doubling the efficacy of a synapse requires quadrupling the amount of resource at the synapse.

2) The direction of the weight vector rotation (equation 8) depends only on the direction of the input vector. The self-organizing process cannot differentiate between inputs in the same direction, and consequently it can only form an $n-1$-dimensional mapping of the n-dimensional input space. This mapping will be distorted, because the magnitude of the weight rotation depends on the length of the input vector: longer vectors cause larger changes. For this reason, the input vectors must be normalized before presenting them to the system. Effectively this means reducing the dimensionality of the input space by one: the input distribution is projected on the surface of a hypersphere.

If the system is to learn the distribution of an n-dimensional input space, a redundant $n+1$:th dimension must be added to the input vectors. The original dimensions are interpreted as angles and the $n+1$:th dimension represents the length of the vector, which is chosen the same for all inputs.

In the simulations reported in this paper, the original distribution was uniformly distributed on the 2-dimensional square $x = (x_1, x_2)$, $-0.5 \leq x_1, x_2 \leq 0.5$). Each input vector was transformed into a 3-dimensional cartesian vector $x = (\xi_1, \xi_2, \xi_3)$ whose coordinates in the spherical system are $(x_1, x_2, 1)$:

$$\begin{cases} \xi_1 = 1 \cdot \cos(x_1)\cos(x_2), \\ \xi_2 = 1 \cdot \sin(x_1)\cos(x_2), \\ \xi_3 = 1 \cdot \sin(x_2) \end{cases} \quad (9)$$

When these 3-dimensional input vectors are presented to the system it organizes itself accordingly, forming a 2-dimensional mapping of the 3-dimensional input space. Since the dimensions are optimally chosen in the self-organizing process [7] and the third input dimension is redundant, the mapping directly represents the original 2-dimensional input space.

Note that the coordinate transformation is unique only when $-\pi/2 < x_2 < \pi/2$, and the rotation direction is correct only when $-\pi/2 \leq x_1 \leq \pi/2$. Numerically the most stable area is a fairly close neighborhood of $(0, 0, 1)_{spherical}$.

5 Experiments

The biological implementation (equations 3 to 5 and 8) was tested with various degrees of initial order in the map and with various parameter settings. Overall, the self-organizing process resembles the abstract case fairly well.

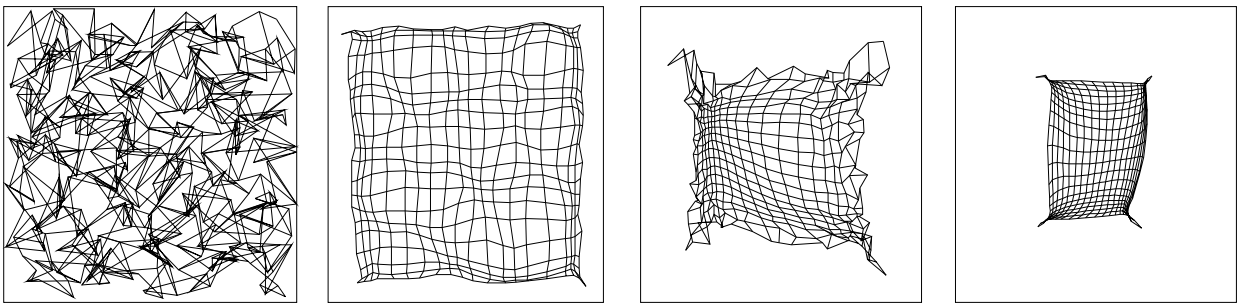
The degree of initial ordering determines how large the activity clusters must be to successfully produce self-organization. This in turn determines the width of the smallest useful lateral inhibition mask. When the initial weight components were randomly distributed within 0.1 radius of their optimal (ordered) values, the network self-organized with $d = 1$ (figures 6a and 6b). For a 0.25 radius, d had to be increased to 2, and for 0.5, to 3. When the network was completely random initially (radius=1.0), total mask diameter of 26 ($d = 4$) was required (figures 7a to 7c).

The fact that lateral inhibition can organize even an initially completely random network is a somewhat surprising result. There are no activity clusters within the initial response that could be focused (figure 5a). However, when the lateral inhibition mask is larger than the network, the whole network forms one initial activity cluster. The differentiation capability of this large a mask is very subtle. The resulting activity clusters are large and tend to form around the center of the network (figure 5b). The subtle differences between them are enough to force the network to organize. One can say that order is brought about by boundary effects: the boundaries mold the mapping in shape, as suggested by Kohonen [6]. Note that the network needs to be almost fully connected to produce order.

As in the abstract implementation, the larger the mask, the stronger is its self-organizing capability, i.e. the faster the convergence to the proper order. This can be seen in figure 6, where a map with 0.1 initial random radius is organized with masks $d = 1$ and $d = 4$. With the smaller mask, 1,600 presentations were required to order the map, whereas the larger mask only needed 400 samples.

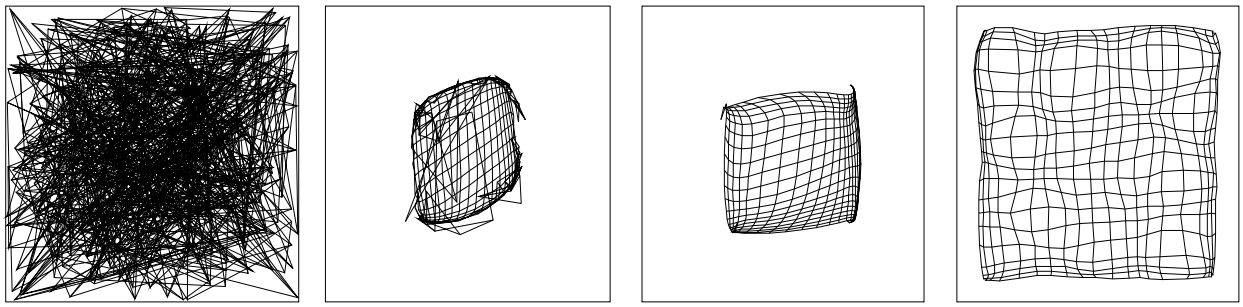
Self-organization begins at the center of the network and spreads towards the boundaries (figures 6c and 7b). This is because the activity clusters tend to form around the center of the network. Weight adaptation is proportional to the activity of the unit, and individual units may be left out of the activity cluster until the map is fairly well ordered (7b). This process differs from the abstract implementation, where the units are first clustered together and the clusters organized (figures 1b and 1c).

The contraction of the final mapping is comparable to the mask size. The mask with $d = 1$ covers the input space fairly well (6b), while $d = 4$ is badly contracted (6d). This is because larger masks generate



(a) 0 samples (b) 1,600 samples, $d = 1$ (c) 100 samples, $d = 4$ (d) 400 samples, $d = 4$

Figure 6: **Self-organizing a map with 0.1 initial randomness.** (a) Initial configuration of the map. Weight components are distributed uniformly within 0.1 of their optimal value. (b) Final configuration of the map organized with the mask $d = 1, \gamma_E = 0.03, \delta = 0.98, \beta = 1.15$. (c) and (d) Configuration after 100 and 400 presentations obtained with the mask $d = 4, \gamma_E = 0.01, \delta = 0.8, \beta = 1.5$. The map is ordered faster, but it is badly contracted. Throughout the simulations, the network was allowed to settle for 10 iterations, learning rate α was constant at 0.1 and $\rho = 8.0$.



(a) 0 samples (b) 350 samples (c) 1,000 samples (d) 5,000 samples

Figure 7: **Self-organizing an initially random map.** The weight components are distributed uniformly between 0 and 1. The figures display organization after 0, 350, 1,000, and 5,000 input vector presentations. Mask $d = 4, \gamma_E = 0.01, \delta = 0.8, \beta = 1.5$ was used during the first 1,000 presentations, $d = 3, \gamma_E = 0.015, \delta = 0.9, \beta = 1.4$ until 2,000, $d = 2, \gamma_E = 0.02, \delta = 0.95, \beta = 1.3$ until 3,000 and $d = 1, \gamma_E = 0.03, \delta = 0.98, \beta = 1.15$ during the rest of the simulation. Throughout the simulation, the settling time was 10, $\alpha = 0.1$ and $\rho = 8.0$.

larger concentrated activity patterns, and weights are changed in larger neighborhoods.

In order to expand the ordered but contracted map, the activity clusters must be made gradually smaller. Increasing inhibition/excitation ratio (decreasing ρ) or increasing the threshold δ of the sigmoid will produce this result. However, the boundary effects still prevent the map from expanding. The clusters never form at the boundary, but at the $d + 1$th layer from the boundary (figure 4b). The weight changes that would expand the map are canceled out by changes in the opposite direction[1].

Currently the only way to form strong clusters close to the boundary is to use masks with smaller excitation width d. Figure 7 displays a simulation where the mask size was decreased from $d = 4$ to $d = 1$ in 4 phases. The final mapping has spread out fairly well and it is fairly uniform. However, the three outermost layers still remain slightly contracted from their optimal position.

[1]If the lateral interaction strength γ_E is gradually decreased, the unit response reflects the external input more, and the response to an input outside the map moves closer to the boundary. However, this will not help expand the map because the magnitude of the response is much smaller. The weight changes that result are small and overridden by stronger changes that move the weights to the opposite direction.

6 Discussion

It is interesting to speculate on the biological meaning of the adjustments that were found necessary for self-organization. How could the transformation of the n-dimensional input space onto a redundant $n + 1$-dimensional hypersphere surface be implemented in biological systems? If the data is represented by coarse coding [3], the input activation to the neuron may be highly redundant. In addition, the input components should be coupled in such a way that an increase in the activity of one component is accompanied by an increasingly larger decrease in the activity of other components.

It appears that gradually decreasing the lateral inhibition mask would require synaptic efficacy changes where excitatory connections become inhibitory. However, feature map units do not necessarily have to represent individual neurons. It is possible to interpret them as neuronal groups connected with multiple excitatory and inhibitory connections. The inhibitory connections between two groups could gradually become stronger than excitatory ones, changing the sign of the lateral interaction. How the inhibition can increase in a geometrically ordered fashion and how a correct schedule for this process can be determined are still open questions.

It is conceivable that biological feature maps would have some initial order. Whatever the task is, some regularities could be expected and the system could

be prewired to deal with them. This would be an advantage in evolution and therefore likely to be produced by evolution. Self-organization would not need to start in a completely random initial state and the lateral connectivity could be narrower, producing less contracted maps without the need for changing the lateral connections.

Currently there is no way to prevent the activity clusters from forming away from the boundary without also disrupting self-organization. The fact that the lateral activity drops to 0 outside the map appears to be important for bootstrapping the mapping. Methods that form clusters at the boundary (virtual extension of activity beyond the boundaries, modifying the masks of the units near the boundary) destroy this property and the map does not self-organize. This phenomenom calls for further research.

7 Conclusion

Lateral inhibition combined with redistribution of synaptic resources produces self-organization with certain restrictions. (1) The input vectors must all have the same length, i.e. an n-dimensional input space must be transformed onto a surface of an $n+1$-dimensional hypersphere before the mapping. (2) The weight vectors must all have the same length, i.e. the synaptic efficacies are proportional to the square root of the resource being redistributed. (3) The lateral interaction, sigmoid and gain parameters must be carefully evolved according to the current degree of order in the network. (4) The outermost layers of the network will become clustered together.

The most important open research question is how to form activity clusters at the boundary of the map without weakening the system's self-organization capability.

References

[1] H. K. Hartline. Inhibition of activity of visual receptors by illuminating nearby retinal areas in the limulus eye. *Federation Proceedings*, 8(1):69–, 1949.

[2] Donald O. Hebb. *The Organization of Behaviour.* John Wiley, New York, 1949.

[3] Geoffrey E. Hinton, James L. McClelland, and David E. Rumelhart. Distributed representations. In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume I: Foundations.* MIT Press, Cambridge, MA, 1986.

[4] Jari Kangas, Teuvo Kohonen, Jorma Laaksonen, Olli Simula, and Olli Ventä. Variants of self-organizing maps. Technical report, Laboratory of Computer and Information Science, Helsinki University of Technology, 1989.

[5] Teuvo Kohonen. Automatic formation of topological maps of patterns in a self-organizing system. In *Proceedings of the 2nd Scandinavian Conference on Image Analysis.* Pattern Recognition Society of Finland, 1981.

[6] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[7] Teuvo Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin; New York, 1984.

[8] Teuvo Kohonen, Gyorgy Barna, and Ronald Chrisley. Statistical pattern recognition with neural networks: Benchmarking studies. In *Proceedings of the IEEE Second Annual International Conference on Neural Networks.* IEEE, 1988.

[9] Teuvo Kohonen, Kai Makisara, and Tapio Saramaki. Phonotopic maps - insightful representation of phonological features for speech recognition. In *Proceedings of the 6th International Conference on Pattern Recognition*, pages 182–185. IEEE Computer Society Press, 1984.

[10] Teuvo Kohonen, Kimmo Raivio, Olli Simula, Olli Ventä, and Jukka Henriksson. An adaptive discrete-signal detector based on self-organizing maps. In *Proceedings of the International Joint Conference on Neural Networks, Washington, DC*, 1990.

[11] Risto Miikkulainen. *DISCERN: A Distributed Artificial Neural Network Model of Script Processing and Memory.* PhD thesis, Computer Science Department, University of California, Los Angeles, 1990.

[12] Risto Miikkulainen. A distributed feature map model of the lexicon. In *Proceedings of the Twelfth Annual Cognitive Science Society Conference*, Hillsdale, NJ, 1990. Lawrence Erlbaum.

[13] Risto Miikkulainen. Script recognition with hierarchical feature maps. *Connection Science*, 2(1&2):83–101, 1990.

[14] F. Ratliff, H. K. Hartline, and D. Lange. The dynamics of lateral inhibition in the compound eye of limulus. In *Proceedings of the International Symposium on the Functional Organization of the Compound Eye*, 1966.

[15] Helge Ritter and Teuvo Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61:241–254, 1989.

[16] Helge J. Ritter, Thomas M. Martinetz, and Klaus J. Schulten. Topology-conserving maps for learning visuomotor-coordination. *Neural Networks*, 2(3), 1989.

[17] Helge J. Ritter and Klaus J. Schulten. Convergency properties of kohonen's topology conserving maps: Fluctuations, stability and dimension selection. *Biological Cybernetics*, 60:59–71, 1988.