

# On Self-Organizing Feature Map (SOFM) Formation by Direct Optimization Through a Genetic Algorithm

José Everardo B. Maia

State Univ. of Ceará, Fortaleza, Ceará, Brazil  
Department of Statistics and Computing  
Email: jmaia@uece.br

Guilherme A. Barreto

Federal Univ. of Ceará, Fortaleza, Ceará, Brazil  
Department of Teleinformatics Engineering  
Email: guilherme@deti.ufc.br

André L. V. Coelho

University of Fortaleza, Fortaleza, Ceará, Brazil  
Graduate Program in Applied Informatics  
Email: acoelho@unifor.br

## Abstract

*This paper examines the formation of self-organizing feature maps (SOFM) by the direct optimization of a cost function through a genetic algorithm (GA). The resulting SOFM is expected to produce simultaneously a topologically correct mapping between input and output spaces and a low quantization error. The proposed approach adopts a cost (fitness) function which is a weighted combination of indices that measure these two aspects of the map quality, specifically, the quantization error and the Pearson correlation coefficient between the corresponding distances in input and output spaces. The resulting maps are compared with those generated by the Kohonen's self-organizing map (SOM) algorithm in terms of the Quantization Error (QE), the Weighted Topological Error (WTE) and the Pearson correlation coefficient (PCC) indices. The experiments show the proposed approach produces better values of the quality indices as well as is more robust to outliers.*

## 1. Introduction

First attempts to build self-organizing neural network models were strongly motivated by the purpose of simulating the formation of topographically ordered cortical maps in the brain [22, 8], with the visual cortex receiving most of the attention [19]. As expected, most of these computational models were quite successful in their original biologically motivated purposes. However, it was Kohonen [12] that pioneered in proposing a computationally efficient algorithm for the training of such maps, commonly referred to as the Self-Organizing Map (SOM) algorithm.

Originally conceived as a simplified version of Malsburg's model, the SOM has become one of the most important self-organizing neural network models with a number of important applications in pattern recognition, specially as a tool for data visualization and dimensionality reduction [5, 13]. Roughly speaking, the SOM is a vector quantization algorithm that tries to map distance (proximity) relations of input data vectors onto an output array of neurons. In other words, data points that are close to each other in the original input space are mapped to neurons which are, in turn, also close to each other in the output array, thereby resulting in a topology-preserving mapping.

Training the SOM algorithm comprises two basic mechanisms. First, a competitive strategy is used for the selection of a single winning neuron for each input vector; for example, the winning neuron is the one whose weight vector is the closest to the input according to the Euclidean distance. Second, a cooperative strategy makes use of a neighborhood function centered on the position of the winning neuron in the output array to update the weight vectors of the neurons around the winner. The consequence of the neighborhood-oriented approach is that the weight vectors of those neurons which are closer to the winning neuron are updated more strongly in the direction of the current input data.

It turns out that after some suitable number of training iterations the joint work of SOM's competitive-cooperative strategies results in a topology-preserving map as a remarkably emergent property. It is worth emphasizing that despite its apparent computational simplicity, the mathematical analysis of the SOM's convergence as well as its optimality as a data representation algorithm are problems difficult to handle [6], still requiring contributions. For example, after more than two decades of its proposal, the competitive-

cooperative mechanism responsible for the map formation in the SOM is not thoroughly understood. There are many theoretic important advances in this regard, but most of the knowledge comes from empirical studies [2].

In what concern the theory of self-organizing feature maps (SOFM) formation, several approaches have already been proposed, among them we can mention those based on Markovian models [18], Information Theory [14, 10], Kernel methods [20], Constrained Optimization techniques [23], Expectation-Maximization algorithm [7, 9], among others. Essentially, these studies view the generation of topographic maps as an optimization problem. Thus, the basic idea is to formulate a cost function that takes its minimum with respect to the parameters to be determined when the desired state of the mapping is reached. Then the minimization of the cost function will automatically lead to the optimal set of parameters.

The problem is that this approach requires a differentiable cost function. However, it has been shown that a cost function for the original SOM algorithm would be highly discontinuous and with many local minima [4]. In this case, a natural alternative would be to develop topographic maps by means of genetic algorithms (GAs) [3], an approach that was not fully explored yet.

Bearing this in mind, in the current paper we tackle the SOFM formation problem by direct optimization through a genetic algorithm. We show by computer simulations that, with the proposed fitness function, the direct optimization process always leads to topology-preserving mappings, which also display robustness to outliers.

The remainder of the paper is organized as follows. In Section 2 we introduce the proposed GA-based approach to SOFM formation, describing in detail all its components. In Section 3 we present the obtained results and discuss them accordingly. The paper is concluded in Section 4.

## 2. The Proposed Approach

We briefly here describe the SOM algorithm. In short, the SOM is a competitive learning network with the neurons geometrically arranged in an output array. Its number of neurons and the dimension of the output array must be defined in advance. Usually, the array is one- or two-dimensional, but three-dimensional cases are easy to find. Each neuron is associated with a weight vector representing the centroid of a particular region of the data space (Voronoi cells). The weight vectors have the same dimension as the input vectors. This paper uses the following notation:  $N$  is the number of neurons,  $P$  is the input space dimension,  $L$  is the number of data samples,  $\mathbf{w}_j$  is the weight vector of neuron  $j$ ,  $\mathbf{w}_i$  is the weight vector of the winning neuron, and  $\mathbf{x}$  is a given input vector. Once chosen the number of neurons and the dimension of the

output array, the SOM training consists of the following steps:

### Pseudo-code for the SOM algorithm

1. Initialize (randomly) the weight vectors:  $\mathbf{w}_j(0)$ ,  $j = 1, \dots, N$ .
2. Sample input data vectors  $\mathbf{x}$  according to a probability distribution  $p(\mathbf{x})$ .
3. Choose the winning neuron  $i(\mathbf{x})$  at iteration  $n$ , using the minimum Euclidean distance criterion:

$$i(\mathbf{x}) = \arg \min_{\forall j} \|\mathbf{x}(n) - \mathbf{w}_j(n)\|, j = 1, \dots, N.$$

4. Update the weight vectors of the neurons as follows:

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta(n)h_{i,j}(n)[\mathbf{x}(n) - \mathbf{w}_j(n)],$$

where  $\eta(n)$  is the learning-rate parameter and  $h_{i,j}(n)$  is the neighborhood function centered around the position of the winning neuron  $i(\mathbf{x})$  in the output array; both  $\eta(n)$  and the width of  $h_{i,j}(n)$  decay in time for the sake of convergence.

5. Repeat Steps 2-5 until no significant changes in the map are observed.

As pointed out in the introduction, our goal is to introduce a GA-based approach to SOFM formation and compare it with the standard SOM in terms of quality of the generated map. Genetic algorithms are stochastic optimization methods grounded on concepts from evolutionary biology and population genetics, that is, on the neo-Darwinian theory of evolution. Since these methods are not based on derivatives of an objective function, they are suitable to deal with discontinuous cost functions and less prone to get trapped on local minima than the gradient based methods.

The issue of map formation using GA has been addressed in some studies [1, 17, 16, 15]. Some of these works are not explicitly concerned with the map formation issue, but rather in using GA to evolve the original SOM with the goal of finding optimal network topology (number of neurons and array dimension) as well as optimal training parameters, such as the learning rate scheduling and annealing of the neighborhood width. The work of Polani [17] is representative of this group.

In [11] map formation is achieved via a two-step procedure. In the first step, they apply the  $K$ -means algorithm for cluster formation, while in the second step, with the weight vectors frozen, they use GA to build a topologically ordered map. Thus, according to this approach weight adaptation is separated from weight ordering.

To the best of our knowledge, Curry and Morgan [1] pioneered in proposing a single-step GA-based direct optimization approach for SOFM formation, tuning the weights simultaneously with the map ordering. The current paper follows Curry and Morgan’s approach, but differs significantly in a number of points, among them the type of fitness function, the genetic operations used, and in the performance parameters used for comparison. Furthermore, we show that the proposed approach is robust to outliers while theirs seems not to be.

## 2.1 Basics of Genetic Algorithms

Genetic algorithms are a family of stochastic search algorithms that imitates Natural Selection (survival of the fittest) in the search for the best solution [3]. They are based on the application of genetic operations (selection, recombination and mutation), which mimic well-known evolutionary phenomena. In this approach one has a population of candidate solutions, coded as chromosomes, and a fitness function that measures the adaptability of the individuals to the environment, that is, the quality of the solutions. For each iteration (generation) the population individuals evolve using genetic operations and some strategy, usually based on the fitness function, that defines which individuals will survive to the next generation. Generally, with the aim to preserve diversity, the strategy is not limited to the selection of the most fittest individuals, although this strategy is adopted frequently. The pseudocode description of a typical GA is given below [3].

### Pseudo-code for the GA procedure

1. Generate the initial population of individuals.
2. Evaluate the fitness of each individual.
3. Repeat
  - Select probabilistically the individuals to reproduce (with a bias to those best-ranked).
  - Breed new offspring through crossover and mutation (genetic operations).
  - Evaluate the fitness values of the offspring.
  - Replace worst ranked part of population with offspring.
4. Until termination

The most significant advantages of using evolutionary algorithms lie in the gain of flexibility and adaptability to the task at hand, in combination with robust performance and global search characteristics. The key to successful implementation of GAs most often lies in the choice of a genetic

representation that mirrors the problem well and the use of ‘intelligent’ genetic operators which again have a natural affinity to the problem of interest [3]. In the next sections we describe the components of the proposed GA-based approach to SOFM formation.

## 2.2 The Proposed Fitness Function

To measure the quality of the formed map, we use the Quantization Error (QE) and the Topographic Error (TE) indices, defined respectively as

$$QE = \frac{1}{L} \sum_{l=1}^L \|\mathbf{w}_i - \mathbf{x}_l\|, \quad (1)$$

and

$$TE = \frac{1}{L} \sum_{l=1}^L \Psi(\mathbf{x}_l, \mathbf{w}_1, \dots, \mathbf{w}_N), \quad (2)$$

where  $\Psi(\mathbf{x}_l, \mathbf{w}_1, \dots, \mathbf{w}_N) \in [0, 1]$ .

The QE index assesses how good is the SOFM as a vector quantizer, whereas the TE index captures the proportion of data points for which the closest and second-closest weight vectors are not adjacent on the map grid, which is determined by  $\Psi$ . The relative importance of these two indices depends on the application and it is very difficult to manage using Kohonen’s SOM algorithm. It would be desirable to have some degree of control of these indices during the formation of the topology-conserving maps, so that the user can choose the most important one for a given application. The GA-based approach provides such a flexibility by the choice of a suitable fitness function.

In principle, any of the above indices of map quality, or their combination, could be used as a fitness function for direct optimization by GA, in view that the fitness function is only used in the stages of mating selection and environmental selection [3]. On the other hand, the fitness function has a significant impact on the map quality (QE and TE) and on the computational demands of the genetic approach.

In this context, Kirk and Zurada [11] used the Weighted Topological Error (WTE) index, defined as

$$WTE = \frac{1}{L} \sum_{l=1}^L \frac{|i_1(l) - i_2(l)| - 1}{N - 1}, \quad (3)$$

where  $i_1(l)$  and  $i_2(l)$  are the indices of the neurons whose weight vectors are, respectively, the first and second closest ones to the  $l$ -th input vector. They compared the WTE values obtained by their GA-based approach with those obtained by Kohonen’s SOM, showing clear advantages for the GA. However, their results were achieved solely for one-dimensional SOMs.

**Table 1. Average values of QE, WTE and PCC (data set 1,  $\alpha = \beta = 1$ )**

Method - Grid size	QE	WTE	PCC
GA - $1 \times 9$	0.154	0.021	0.867
SOM - $1 \times 9$	0.141	0.034	0.823
GA - $3 \times 3$	0.086	0.055	0.891
SOM - $3 \times 3$	0.085	0.044	0.866
GA - $1 \times 25$	0.083	0.062	0.860
SOM - $1 \times 25$	0.081	0.054	0.821
GA - $5 \times 5$	0.084	0.035	0.961
SOM - $5 \times 5$	0.081	0.030	0.911

**Table 2. Average values of QE, WTE and PCC (data set 2,  $\alpha = \beta = 1$ )**

Method - Grid size	QE	WTE	PCC
GA - $1 \times 9$	0.142	0.037	0.793
SOM - $1 \times 9$	0.137	0.077	0.731
GA - $3 \times 3$	0.089	0.050	0.906
SOM - $3 \times 3$	0.091	0.044	0.862
GA - $1 \times 25$	0.085	0.057	0.890
SOM - $1 \times 25$	0.081	0.044	0.871
GA - $5 \times 5$	0.078	0.025	0.965
SOM - $5 \times 5$	0.077	0.030	0.921

Conversely, Curry and Morgan [1] used as fitness function the *locally weighted distortion index* (LWDI):

$$LWDI = E \left[ \sum_{\forall j} h_{ij} \|\mathbf{x} - \mathbf{w}_j\| \right], \quad (4)$$

where  $E[\cdot]$  denotes the expectation operator taken over the whole set of input vectors. A comparison between their GA-based approach and Kohonen's SOM in terms of the obtained LWDI values favored the former.

In this paper we show that a fitness function formed by the simple combination of the Quantization error (QE) and the Pearson correlation coefficient (PCC) is as good as the WTE and LWDI for the purposes of SOFM formation. The proposed fitness function is given by

$$Fitness(\widetilde{\mathbf{W}}) = \alpha \cdot PCC(\widetilde{\mathbf{W}}) - \beta \cdot QE(\widetilde{\mathbf{W}}), \quad (5)$$

where  $\widetilde{\mathbf{W}}$  denotes the whole set of weight vectors, and the parameters  $\alpha, \beta \in [0, 10]$  weigh the relative importance of the indices with respect to each other. The PCC index is computed as the cross-correlation between pairs of distances  $[d(\mathbf{r}_m, \mathbf{r}_n), d(\mathbf{w}_m, \mathbf{w}_n)]$ , where  $(\mathbf{r}_m, \mathbf{r}_n)$  are the coordinates of pairs of neurons in the output array and  $(\mathbf{w}_m, \mathbf{w}_n)$  are the corresponding pairs of weight vectors.

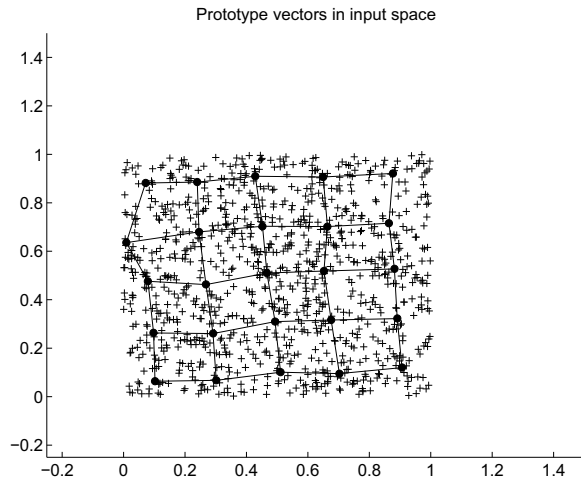
It is worth noting that *PCC* is an index of the type *the larger, the better*, while *QE* is of the type *the lesser, the better*. The direct optimization (maximization) of the fitness function shown in Eq. (5) via GA is the main contribution of this paper. The proposed approach is compared with the Kohonen's SOM in terms of the QE, WTE, and PCC values produced for a given formed map. The robustness to outliers of both approaches is also evaluated.

### 3. Computer Simulations

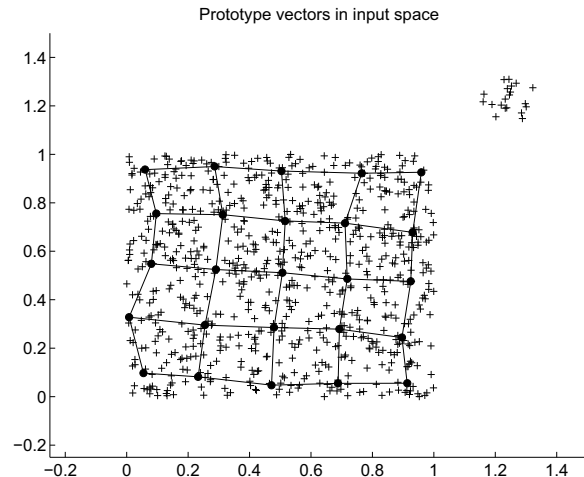
In our GA implementation, the individuals are coded as real numbers in a matrix with  $N$  rows and  $P$  columns rep-

resenting the  $N$  weight vectors of the neurons in the network. Single-point crossover is adopted whereas mutation is implemented by adding a zero-mean Gaussian random variable to the current value of a gene. Each individual of the current population (irrespective of its fitness value) generates a fixed number of descendants, which are evaluated and the most fittest are selected for the next generation. In this paper, we decided for a simple deterministic selection method (both for reproduction and population replacement) to reduce simulation time. Indeed, this choice, together with the high rates of mutation used sometimes, makes our GA to much resemble an Evolution Strategy [3]. Two artificial data distributions were used in the simulations: (i) a uniform distribution on the unit square (data set 1), and (ii) a uniform distribution on the unit square with outliers (data set 2). A total of 1,000 samples were used for input data, and a population of 40 individuals was available in each generation. The tests were performed in MATLAB with the support of SOM toolbox [21].

A series of runs of the GA used mutation rates of 0.01, 0.05, 0.10, 0.20, and 0.50. Following the methodology proposed in [1], initialization was determined by a seed value for the random number generator so that results could be checked subsequently. The tests were performed with the GA algorithm and the Kohonen's SOM for the following network topologies:  $1 \times 9$ ,  $3 \times 3$ ,  $1 \times 16$ ,  $4 \times 4$ ,  $1 \times 25$ , and  $5 \times 5$ . The parameter  $\alpha$  of the fitness function assumed the values of 1, 2, 5, and 10, and the simulations run for 500, 1,000, 2,000, and 10,000 generations. Typically, the GA converges after 500 up to 2,000 generations. The mean values averaged over 20 runs for the indices QE, WTE, and PCC are shown in Tables 1 and 2 for the two simulated distributions. (Due to space limitation, the standard deviation values for the GA were omitted, although we can assert that they were not much high and vary in magnitude across the indices.) Moreover, some variability is found in the runs for all six network topologies and for the two distributions. As expected, there is a predominance of favorable results to the



**Figure 1. Typical 5x5 map formed by the GA-based approach for data set 1 ( $\alpha = \beta = 1$ ).**



**Figure 2. Typical 5x5 map formed by the GA-based approach for data set 2 ( $\alpha = \beta = 1$ ).**

proposed GA-based approach for all indices.

A typical SOFM built by the proposed GA-based approach for the data set 1 is shown in Figure 1, where one can see a topologically correct map. Interesting results are obtained for the data set 2, which contain outliers. Figure 2 shows a typical SOFM formed by the proposed GA-based approach. For the sake of comparison, Figure 3 shows a typical SOFM formed by the Kohonen approach.

By comparing these two figures, one can easily note that the map formed by our GA did not assign a weight vector to the small clouds of points (outliers) in the upper right corner of the figures, while Kohonen's SOM algorithm *always* assigns one weight vector to this cloud. An immediate consequence of this behavior of Kohonen's algorithm is that the final map is considerably distorted. Hence, one can conclude that the proposed approach is relatively more robust to outliers than the Kohonen's algorithm.

As a final remark, we would like to point out that, if required by the user, the proposed GA-based approach can do assign a weight vector to the cloud of outliers. It can be easily done by giving more importance to the *QE* index than to the *PCC* index (i.e.  $\beta \gg \alpha$ ) in the fitness function shown in (5). Figure 4 shows a map at an intermediate generation being evolved by the proposed algorithm with  $\alpha = 1$  and  $\beta = 4$ , where we clearly see the displacement of a weight vector towards the cloud of outliers. This flexibility is not provided by the standard SOM algorithm.

## 4 Conclusions

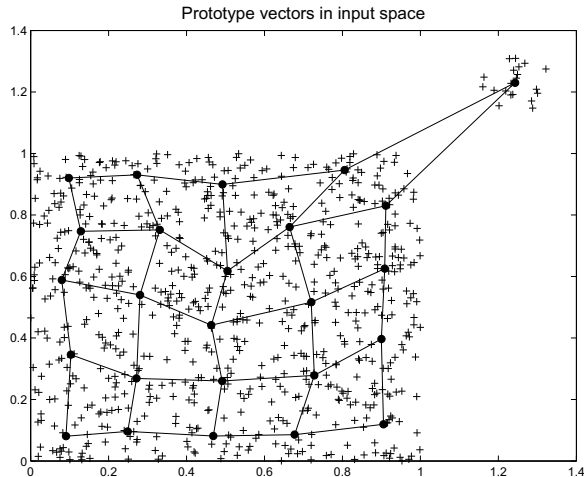
This article introduced a novel fitness function for feature map formation using genetic algorithms. The main

motivation was to obtain better quality maps for the approximation of data distribution in the input space even in the presence of outliers. Maps of different sizes and dimensions obtained by the proposed procedure were compared with those obtained by standard Kohonen's SOM algorithm using three quality criteria, namely: the Quantization Error (*QE*), the Weighted Topological Error (*WTE*), and the Pearson correlation coefficient (*PCC*). The experiments show better values of these indices for the map obtained by the proposed GA-based direct optimization.

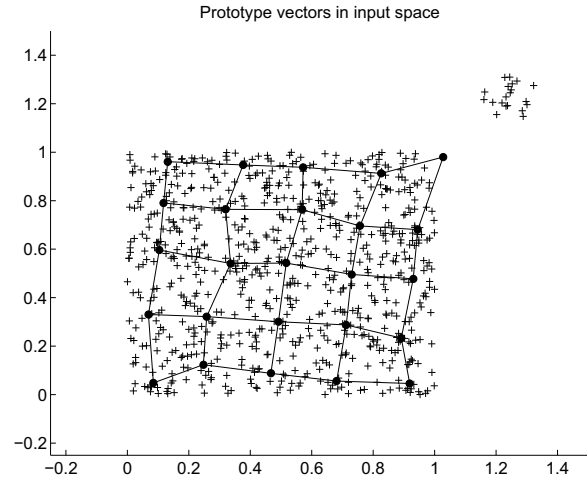
In continuity to this work, some further investigation is currently underway. First, we are evaluating the feasibility of defining a neighborhood kernel for the mutation operation. That is, every time a single gene is randomly selected for being mutated, its immediate neighbors are also mutated. It is expected that this kernel-based mutation accelerates the weight vector ordering and hence the whole map formation algorithm. Second, we are testing the effectiveness of different selection and crossover operators. Finally, as the fitness function is a weighted linear combination of the *QE* with a neighboring preservation measure, we are testing a time-varying procedure, similar to that of the SOM algorithm, which adaptively changes the parameters  $\alpha$  and  $\beta$  in (5), starting with the *QE* heavily weighted with respect to the *PCC* index (or vice versa).

## 5 Acknowledgements

The second author thanks CAPES (PRODOC grant) for the financial support.



**Figure 3. Typical 5x5 map formed by the Kohonen's algorithm for data set 2.**



**Figure 4. A 5x5 map being evolved by the proposed approach for dataset 2 ( $\alpha = 1, \beta = 4$ ).**

## References

- [1] B. Curry and P. H. Morgan. Evaluating Kohonen's learning rule: An approach through genetic algorithms. *European Journal of Operational Research*, 154(1):191–205, 2004.
- [2] E. de Bodt, M. Cottrell, P. Letremy, and M. Verleysen. On the use of self-organizing maps to accelerate vector quantization. *Neurocomputing*, 56:187–203, 2004.
- [3] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.
- [4] E. Erwin, K. Obermayer, and K. Schulten. Self-organizing maps: ordering, convergence properties and energy functions. *Biological Cybernetics*, 67(1):47–55, 1992.
- [5] A. Flexer. On the use of self-organizing maps for clustering and visualization. *Intelligent Data Analysis*, 5:373–384, 2001.
- [6] J. C. Fort. SOM's mathematics. *Neural Networks*, 19(6–7):812–816, 2006.
- [7] T. Graepel, M. M. Burger, and K. Obermayer. Self-organizing maps: Generalizations and new optimization techniques. *Neurocomputing*, 21:173–190, 1998.
- [8] S. Grossberg. On the development of feature detectors in the visual cortex with applications to learning and reaction-diffusion systems. *Biological Cybernetics*, 21:145–159, 1975.
- [9] T. Heskes. Self-organizing maps, vector quantization and mixture modeling. *IEEE Transactions on Neural Networks*, 12(6):1299–1305, 2001.
- [10] R. Kamimura. Cooperative information maximization with gaussian activation functions for self-organizing maps. *IEEE Transactions on Neural Networks*, 17(4):909–918, 2006.
- [11] J. S. Kirk and J. M. Zurada. Motivation for a genetically-trained topography-preserving map. In *Proceedings of the 2002 International Joint Conference on neural Networks (IJCNN'02)*, volume 1, pages 394–399, 2002.
- [12] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.
- [13] T. Kohonen. *Self-Organizing Maps*. Springer, 2001.
- [14] R. Linsker. How to generate ordered maps by maximizing the mutual information between input and output signals. *Neural Computation*, 1(3):402–411, 1989.
- [15] M. McInerney and A. Dhawan. Training the self-organizing feature map using hybrids of genetic and Kohonen methods. In *IEEE World Congress on Computational Intelligence (WCCI'04)*, volume 2, pages 641–644, 1994.
- [16] A. S. Nissinen and H. Hyotyniemi. Evolutionary self-organizing map. In *Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT98)*, volume 3, pages 1596–600, 1998.
- [17] D. Polani. On the optimization of self-organizing maps by genetic algorithms. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 157–169. Elsevier, 1999.
- [18] H. Ritter and K. Schulten. Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability, and dimension selection. *Biological Cybernetics*, 60:59–71, 1988.
- [19] N. V. Swindale. The development of topography in visual cortex: A review of models. *Network: Computation in Neural Systems*, 7(2):161–247, 1996.
- [20] M. M. Van Hulle. Maximum likelihood topographic map formation. *Neural Computation*, 17(3):503–513, 2005.
- [21] J. Vesanto et al. *SOM Toolbox for Matlab 5*. Helsinki University of Technology, 2000.
- [22] C. von der Malsburg. Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85–100, 1973.
- [23] L. Wiskott and T. J. Sejnowski. Constrained optimization for neural map formation: A unifying framework for weight growth and normalization. *Neural Computation*, 10:671–716, 1998.