# 10 Fast Evolutionary Learning in the SOM

**Teuvo Kohonen**

Above we dealt with vector-valued $\mathbf{x}(t)$ and $\mathbf{m}_i(t)$. The SOM philosophy, however, can be much more general. The structures of the models and data can be different: it will suffice that some *fitness function* is definable between the general inputs $X$ and the general models $M_i$, respectively. Let this function be denoted $f(X, M_i)$. *Notice carefully that we do not need any distance function in the $X$ space, nor in the $M_i$ space.*

Even under the above conditions, the SOM can be computed in an "evolutionary" process (cf. [1], Sec. 5.7). We may initialize the models as random samples from the set of possible models. Next we input the samples of $X$, one at a time, and at each step determine that model $M_c$ for which

$$c = \arg\max_i \{f(X, M_i)\} \; . \tag{77}$$

The next step is some kind of *variation* of the $M_i$ in the neighborhood set $N_c$ of the fittest model $M_c$. This variation usually means random but statistically independent replacement of each $M_i$, $i \in N_c$ by some other possible model $M$ on the condition $f(X, M) > f(X, M_i)$.

The evolutionary learning can be implemented by the batch-type SOM without random probing, whereupon it proceeds fast:

1. Initialize the models $M_i$, e.g., by a random choice of their parameter values from a set of possible values.

2. Input a number of items $X$ and list each of them under the respective winner unit (i.e. that $M_i$ for which some fitness function $f(X, M_i)$ is maximum). In case there is a tie, i.e., two or more $M_i$ have the same fitness to $X$, select one of them randomly for the effective unique "winner" under which the listing is made.

3. Find a new value $M_i'$ for each $M_i$ such that if $U_i$ is the union of lists relating to model $M_i$ in the same way as in the Batch Map algorithm discussed in Sec. 1, the sum of the fitness-function values $f(X, M_i')$, $X \subset U_i$ is increased. If there exist ties, a random choice between the best $M_i'$ is made.

4. Repeat from step 2.

# References

[1] T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin, 1995; 2nd edition, 1997.

[2] T. Kohonen, Fast evolutionary learning with batch-type self-organizing maps. To appear in *Neural Processing Letters*, 1999.