



**HAL**  
open science

## Motivated Self-Organization

Nicolas P. Rougier, Yann Boniface

► **To cite this version:**

Nicolas P. Rougier, Yann Boniface. Motivated Self-Organization. 12th International Workshop on Self-Organizing Maps and Learning Vector Quantization, , Jun 2017, Nancy, France. hal-01513519

**HAL Id: hal-01513519**

**<https://inria.hal.science/hal-01513519>**

Submitted on 25 Apr 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Motivated Self-Organization

Nicolas P. Rougier  
Inria Bordeaux Sud-Ouest  
33405 Talence, France

Email: <http://www.labri.fr/perso/nrougier/>

Yann Boniface  
Loria - Campus Scientifique  
54506 Vandoeuvre-ls-Nancy, France  
Email: [yann.boniface@loria.fr](mailto:yann.boniface@loria.fr)

**Abstract**—We present in this paper a variation of the self-organizing map algorithm where the original time-dependent (learning rate and neighborhood) learning function is replaced by a time-invariant one. The resulting self-organization does not fit the magnification law and the final vector density is not directly proportional to the density of the distribution. This lead us to introduce the notion of motivated self-organization where the self-organization is biased toward some data thanks to a supplementary signal. From a behavioral point of view, this signal may be understood as a motivational signal allowing a finer tuning of the final self-organization where needed. We illustrate this behavior through a simple robotic arm setup.

## I. INTRODUCTION

We introduced in [1] the Dynamic Self-Organized Map (DSOM) architecture that is a variation of the self-organizing map algorithm [2] where the original time-dependent (learning rate and neighborhood) learning function has been replaced by a time-invariant learning rule. This modification of the learning rule yields several interesting new properties. First, and because of the time invariance, it is possible for the network to support life-long learning. This means that the network can be fed continuously with new input and the network is able to self-organize itself around the whole set of data (using a set of hypothesis detailed in [1]). This kind of property cannot be easily achieved using SOM-like algorithms because they generally and explicitly depend on a time decreasing learning rate and/or neighborhood function (SOM, NG, GNG) that requires to know beforehand the number of data to be processed. The second property deals with the magnification law as introduced by [3]. Most vector quantization (VQ) algorithms try to match the density through the density of their code book: high density regions of the distribution tend to have more associated prototypes than low density regions. This generally allows to minimize the loss of information (or distortion) as measured by the mean quadratic error. However, in the case of DSOM, the magnification law is not fit and the density of the code book is uncorrelated with the density of the data, hence leading to a regular quantification *a priori* of the underlying probability density function. This second property could be easily considered as a serious drawback for vector quantization if one wants, for example, to estimate data density function. However, from a more behavioral point of view, we argue in this paper that this may be a desirable property provided that we can modulate learning using a dedicated signal, i.e. a motivation to concentrate learning on what is relevant to the task in order to have finely tune representations where necessary.

## II. MODEL

We'll use here the definitions that we first proposed in [1].

### A. Definitions

Let us consider a probability density function  $f(x)$  on a compact manifold  $\Omega \in \mathbb{R}^d$ . A vector quantization (VQ) is a function  $\Phi$  from  $\Omega$  to a finite subset of  $n$  code words  $\{\mathbf{w}_i \in \mathbb{R}^d\}_{1 \leq i \leq n}$  that form the code book. A cluster is defined as  $C_i \stackrel{\text{def}}{=} \{x \in \Omega | \Phi(x) = \mathbf{w}_i\}$ , which forms a partition of  $\Omega$  and the distortion of the VQ is measured by the mean quadratic error

$$\xi = \sum_{i=1}^n \int_{C_i} \|x - \mathbf{w}_i\|^2 f(x) dx. \quad (1)$$

If the function  $f$  is unknown and a finite set  $\{x_i\}$  of  $p$  non biased observations is available, the distortion error may be empirically estimated by

$$\hat{\xi} = \frac{1}{p} \sum_{i=1}^n \sum_{x_j \in C_i} \|x_j - \mathbf{w}_i\|^2. \quad (2)$$

In the following, we will use definitions and notations introduced by [3] where a neural map is defined as the projection from a manifold  $\Omega \subset \mathbb{R}^d$  onto a set  $\mathcal{N}$  of  $n$  neurons which is formally written as  $\Phi : \Omega \rightarrow \mathcal{N}$ . Each neuron  $i$  is associated with a code word  $\mathbf{w}_i \in \mathbb{R}^d$ , all of which established the set  $\{\mathbf{w}_i\}_{i \in \mathcal{N}}$  that is referred as the code book. The mapping from  $\Omega$  to  $\mathcal{N}$  is a closest-neighbor winner-take-all rule such that any vector  $\mathbf{v} \in \Omega$  is mapped to a neuron  $i$  with the code  $\mathbf{w}_i$  being closest to the actual presented stimulus vector  $\mathbf{v}$ ,

$$\Phi : \mathbf{v} \mapsto \arg \min_{i \in \mathcal{N}} (\|\mathbf{v} - \mathbf{w}_i\|). \quad (3)$$

The neuron  $\mathbf{w}_i$  is called the *winning element* and the set  $C_i = \{x \in \Omega | \Phi(x) = \mathbf{w}_i\}$  is called the *receptive field* of the neuron  $i$ . The geometry corresponds to a Voronoï diagram of the space with  $\mathbf{w}_i$  as the center.

### B. Self-Organizing Maps (SOM)

SOM is a neural map equipped with a structure (usually a hypercube or hexagonal lattice) and each element  $i$  is assigned a fixed position  $\mathbf{p}_i$  in  $\mathbb{R}^q$  where  $q$  is the dimension of the lattice (usually 1 or 2). The learning process is an iterative process between time  $t = 0$  and time  $t = t_f \in \mathbb{N}^+$  where vectors  $\mathbf{v} \in \Omega$  are sequentially presented to the map with respect to the probability density function  $f$ . For each presented vector  $\mathbf{v}$  at time  $t$ , a winner  $s \in \mathcal{N}$  is determined according to equation

(3). All codes  $\mathbf{w}_i$  from the code book are shifted towards  $\mathbf{v}$  according to

$$\Delta \mathbf{w}_i = \varepsilon(t) h_\sigma(t, i, s) (\mathbf{v} - \mathbf{w}_i) \quad (4)$$

with  $h_\sigma(t, i, j)$  being a neighborhood function of the form

$$h_\sigma(t, i, j) = e^{-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma(t)^2}}. \quad (5)$$

where  $\varepsilon(t) \in \mathbb{R}$  is the learning rate and  $\sigma(t) \in \mathbb{R}$  is the width of the neighborhood defined as

$$\sigma(t) = \sigma_i \left( \frac{\sigma_f}{\sigma_i} \right)^{t/t_f}, \quad \text{with } \varepsilon(t) = \varepsilon_i \left( \frac{\varepsilon_f}{\varepsilon_i} \right)^{t/t_f}, \quad (6)$$

while  $\sigma_i$  and  $\sigma_f$  are respectively the initial and final neighborhood width and  $\varepsilon_i$  and  $\varepsilon_f$  are respectively the initial and final learning rate. We usually have  $\sigma_f \ll \sigma_i$  and  $\varepsilon_f \ll \varepsilon_i$ .

### C. Dynamic Self-Organizing Map (DSOM)

DSOM is a neural map equipped with a structure (a hypercube or hexagonal lattice) and each neuron  $i$  is assigned a fixed position  $\mathbf{p}_i$  in  $\mathbb{R}^q$  where  $q$  is the dimension of the lattice (usually 1 or 2). The learning process is an iterative process where vectors  $\mathbf{v} \in \Omega$  are sequentially presented to the map with respect to the probability density function  $f$ . For each presented vector  $\mathbf{v}$ , a winner  $s \in \mathcal{N}$  is determined according to equation (3). All codes  $\mathbf{w}_i$  from the code book  $\mathbf{W}$  are shifted towards  $\mathbf{v}$  according to

$$\Delta \mathbf{w}_i = \varepsilon \|\mathbf{v} - \mathbf{w}_i\|_\Omega h_\eta(i, s, \mathbf{v}) (\mathbf{v} - \mathbf{w}_i) \quad (7)$$

with  $\varepsilon$  being a constant learning rate and  $h_\eta(i, s, \mathbf{v})$  being a neighborhood function of the form

$$h_\eta(i, s, \mathbf{v}) = e^{-\frac{1}{\eta^2} \frac{\|\mathbf{p}_i - \mathbf{p}_s\|^2}{\|\mathbf{v} - \mathbf{w}_s\|_\Omega^2}} \quad (8)$$

where  $\eta$  is the *elasticity* or *plasticity* parameter. If  $\mathbf{v} = \mathbf{w}_s$ , then  $h_\eta(i, s, \mathbf{v}) = 0$

### D. Standard distributions

The DSOM algorithm reflects two main ideas:

- If a neuron is close enough to the data, there is no need for others to learn anything: the winner can represent the data.
- If there is no neuron close enough to the data, any neuron learns the data according to its own distance to the data.

The closeness of the winner to the data is controlled using the *elasticity* parameter as illustrated on figure 1. and for uniform distributions, DSOM with proper elasticity is comparable to SOM as illustrated on figure 2 and detailed in [1].

## III. EXPERIMENTAL RESULTS

As we explained in the introduction, the dynamic nature of the DSOM algorithm leads to a regular self-organization, that is, a self organization that does not fit the magnification law and consequently, code-vectors are evenly spread on the distribution support. Even from a behavioral point of view, this may not be satisfactory since we may want to have finer representations in some region of the input space that are of some interest and more generic representations in some other parts of the input space.

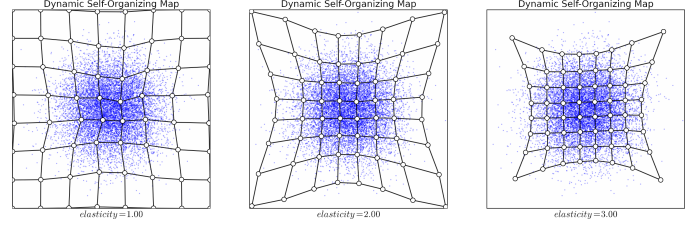


Fig. 1. Three DSOM with respective elasticity equal to 1, 1.5 and 2 have been trained for 20 000 iterations on a normal distribution using a regular grid covering the  $[0, 1]^2$  segment as initialization. Low elasticity leads to loose coupling between neurons while higher elasticity results in a tight coupling between neurons.

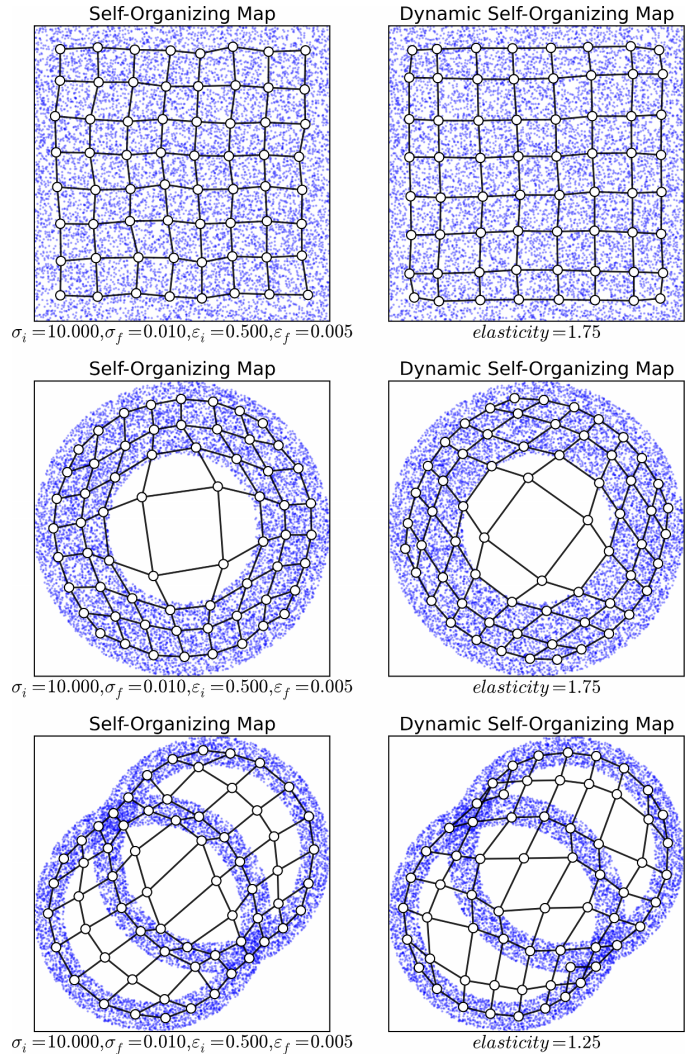


Fig. 2. Side by side comparison of the SOM and DSOM algorithms on very simple and uniform distributions. For each figure, network of size  $8 \times 8$  has been trained for 20 000 epochs using 10 000 samples. Initialization has been done by placing initial code vectors randomly over the  $[0, 1]^2$  area.

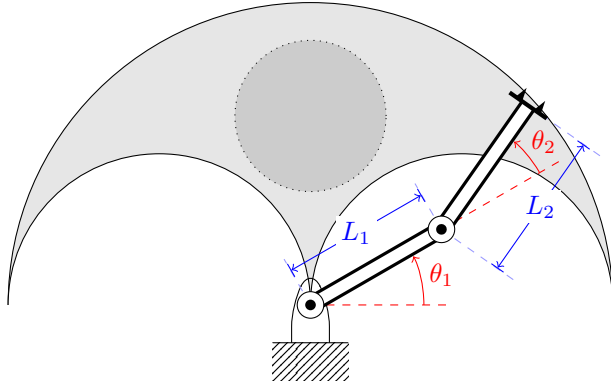


Fig. 3. The robotic arm is made of two segments of respective size  $L_1$  and  $L_2$  whose positions are given by angles  $\theta_1$  and  $\theta_2$ . The gray area represents reachable positions in the case  $L_1 = L_2$  and the dotted disc area corresponding to the region of interest (arbitrary defined).

### A. Experimental setup

Let us consider the case of a simple robotic arm made of two linked segments  $L_1$  and  $L_2$ . Segment  $L_1$  can rotate freely around its base in the range  $[-\pi/2, +\pi/2]$  and segment  $L_2$  can rotate around  $L_1$  endpoint in the range  $[-\pi/2, +\pi/2]$  (see figure 3). The goal of the experiment is to learn the correspondence between  $\{\theta_1, \theta_2\}$  and the Cartesian coordinates  $\{x, y\}$  of the end point of the arm with finer representations in the dotted disc area. Since we do not want to use the inverse model of the arm, the only way to generate data is then to draw  $\{\theta_1, \theta_2\}$  from their respective domain and to compute the corresponding  $\{x, y\}$  end position of the arm.

### B. Results

The mapping from  $\{\theta_1, \theta_2\}$  to  $\{x, y\}$  is not linear and leads to a larger density on the frontier of the domain. If we use such a distribution, we obtain the self-organization drawn on left part of figure 4. This self-organization spreads almost evenly on the whole reachable region with some noticeable and useless representations outside the region. However, as we explained we would like to have finer representations within the dotted gray area and consequently, we built a modulation signal defined as the distance of the end-point to the center of the disc area. This modulation is associated to any sample given to the model and this allow us to modify the original neighborhood equation (8) as follows:

$$h_\eta(i, s, \mathbf{v}) = e^{-\frac{\alpha}{\eta^2} \frac{\|\mathbf{p}_i - \mathbf{p}_s\|_2^2}{\|\mathbf{v} - \mathbf{w}_s\|_2^2}} \quad (9)$$

with  $\alpha$  being the modulation. This  $\alpha$  modulation modifies the overall elasticity on a per sample basis and allows to have a finer tuning of the self-organization. This is illustrated on the left part of the figure 4 where self-organization has concentrated code vectors in the region of interest.

## IV. CONCLUSION

Since the early work of [4], [5], the idea of a critical period in the early years of development, where most sensory

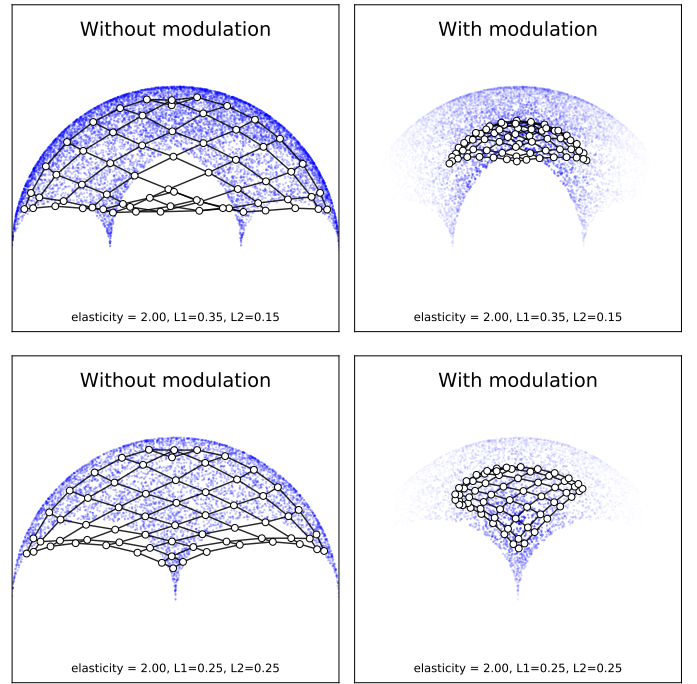


Fig. 4. Sample data are generated by drawing uniformly  $\theta_1$  and  $\theta_2$  in  $[-\pi/2, +\pi/2]$  and computing the end point position. This leads to a non-uniform distribution where higher density is found on the periphery of the distribution support. The left figure displays resulting self-organization of a DSOM with an elasticity of 2.5 and no modulation while the right figure displays the same DSOM algorithm where each sample is modulated according to its distance to the center of the figure.

or motor properties are acquired and stabilized have been widely accepted. In such context, the original SOM algorithm gives a fair account of such development. However, *cortical representations are not fixed entities, but rather, are dynamic and are continuously modified by experience* as explained by [6] and the capacity of the cortex to re-organize itself in face of lesions, deficits or change in the environment [7], [8] cannot be easily explained using SOM-like algorithm since after learning period, the resulting self-organization is *frozen* and cannot be easily changed.

Thanks to its dynamic nature, the DSOM algorithm may help to solve this dilemma by ensuring a tight coupling between representations and the environment with a code book that does not fit data density and cover the whole data support evenly. Since we may need nonetheless to have more representations in some region of the input space, we proposed to modulate learning by providing the algorithm with an explicit signal giving the importance of the data. In a more general framework, we could expect a cerebral structure (e.g. basal ganglia, amygdala) to compute such signal: if some regions of the perceptive space is judged behaviorally relevant, model could develop precise representations in this region. This has been illustrated through a very simple robotic experiment where the motivation signal is computed according to the distance of the arm end-point to the center of the distribution. If learning was to be driven solely by data density (like in most VQ), such modulation would certainly be strongly attenuated or not possible at all.

## APPENDIX

$\Omega$	: a compact manifold of $\mathbb{R}^d$ where $d \in \mathbb{N}^+$
$f(x)$	: a probability density function ( <i>pdf</i> ) $\Omega \rightarrow \mathbb{R}$
$\{x_i\}$	: a set of $p$ non-biased observations of $f$ .
$\mathcal{N}$	: a set of $n$ elements, $n \in \mathbb{N}^+$ .
$\Phi$	: a function defined from $\Omega \rightarrow \mathcal{N}$
$\mathbf{w}_i \in \mathbb{R}^d$	: code word associated to an element $i$ of $\mathcal{N}$
$\{\mathbf{w}_i\}$	: code book associated to $\mathcal{N}$
$C_i$	: cluster associated to element $i$ such that $C_i = \{x \in \Omega   \Phi(x) = \mathbf{w}_i\}$
$\ x\ $	: euclidean norm defined over $\mathbb{R}^d$
$\ x\ _{\Omega}$	: normalized euclidean norm defined over $\Omega$ as $x \mapsto \frac{\ x\ }{\max_{y,z \in \Omega} (\ y-z\ )}$
$\xi$	: distortion error defined as $\sum_{i=1}^n \int_{C_i} \ x - \mathbf{w}_i\ ^2 f(x) dx$
$\hat{\xi}$	: estimated distortion error defined as $\frac{1}{p} \sum_{i=1}^n \sum_{x_j \in C_i} \ x_j - \mathbf{w}_i\ ^2$
$\varepsilon(t)$	: learning rate at time $t$
$\lambda(t)$ or $\sigma(t)$	: neighborhood width at time $t$
$\eta$	: elasticity or plasticity
%	

## REFERENCES

- [1] N. Rougier and Y. Boniface, "Dynamic self-organizing map," *Neurocomputing*, 2011, to appear.
- [2] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.
- [3] T. Villman and J. Claussen, "Magnification control in self-organizing maps and neural gas," *Neural Computation*, vol. 18, pp. 446–449, 2006.
- [4] D. Hubel and T. Wiesel, "The period of susceptibility to the physiological effects of unilateral eye closure in kittens." *Journal of Physiology*, vol. 206, pp. 419–436, 1970.
- [5] N. Daw, "Mechanisms of plasticity in the visual cortex," *Investigative Ophthalmology*, vol. 35, pp. 4168–4179, 1994.
- [6] D. Buonomano and M. Merzenich, "Cortical plasticity: From synapses to maps," *Annual Review of Neuroscience*, vol. 21, pp. 149–186, 1998.
- [7] P. B. y Rita, *Brain Mechanisms in Sensory Substitution*. Academic Press New York, 1972.
- [8] V. Ramachandran, D. Rogers-Ramachandran, and M. Stewart, "Perceptual correlates of massive cortical reorganization," *Science*, vol. 258, pp. 1159–1160, 1992.