



HAL
open science

A modified Self Organizing Map for semantic impairment

Sebastián Basterrech, Jorge Graneri, Eduardo Mizraji, Gerardo Rubino

► **To cite this version:**

Sebastián Basterrech, Jorge Graneri, Eduardo Mizraji, Gerardo Rubino. A modified Self Organizing Map for semantic impairment. MITA 2023 - 19th International Conference on Multimedia Information Technology and Applications, Technical University of Ostrava, Czechia, Jul 2023, Ostrava, Czech Republic. pp.1-9. hal-04358017

HAL Id: hal-04358017

<https://inria.hal.science/hal-04358017>

Submitted on 21 Dec 2023

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.

Public Domain

A modified Self Organizing Map for semantic impairment

Sebastián Basterrech

VŠB–Technical University of Ostrava, Czechia

sebastian.basterrech@vsb.cz

Jorge Graneri

Faculty of Engineering, University of the Republic, Montevideo, Uruguay

jgraneri@fing.edu.uy

Eduardo Mizraji

Faculty of Sciences, University of the Republic, Montevideo, Uruguay

mizraj@fcien.edu.uy

Gerardo Rubino

Ermine Team, INRIA, Rennes, France

gerardo.rubino@inria.fr

Abstract

In 1982 Teuvo Kohonen proposed an algorithm called Self Organizing Map (SOM) that is basically a stochastic model of the establishment of topology preserving connections between the retina and the cortex in our brain. That model has proved an efficient classification tool, a source of inspiration for other biological models and also for the understanding of the principles of self-organization. In this paper, we present a modification of the initial SOM to address an important problem in semantic cognition: the representation of human concepts in the brain. Much of the knowledge acquired in the last decades in this field comes from the study of patients with acquired difficulties in language. The modification presented here is inspired in semantic dementia, a devastating pathology that introduces the deterioration of semantic knowledge into the brain. The paper discusses the model and describes some preliminary results that show its interest as the basis of possible new tools to help to understand the considered type of diseases.

Index Terms

Self Organizing Maps, unsupervised learning, biological applications, concepts representations, semantic dementia.

This paper has been partially supported by ECOS Project U17E03 “MASC” and by the GACR-Czech Science Foundation project No. 21-33574K “Lifelong Machine Learning on Data Streams”.

Paper presented at The 19th International Conference on Multimedia Information Technology and Applications, at Ostrava, Czech Republic, July 11 to 14, 2023, where it received a Best Paper Award.

I. INTRODUCTION

This paper describes some results of a project where we intend to design models able to represent concepts, in a simple way, using a special family of neural networks. These models allow to explore how degrading the networks allows mimicking the effects of a cognitive disease. We chose to focus our work around semantic dementia [7], [14], [12], [4], a particularly hard evolving pathology for which little is known, and where semantic information [13] in the brain degrades until almost disappearing. The model is extremely simple, and it consists of a modification of Self Organizing Maps (SOMs), also called Kohonen networks. We first introduce SOMs, and then, after illustrating their interest for our goals, we show how to modify them with a model of degradation inspired by the previously mentioned disease.

So, we start by introducing Kohonen’s model, a specific type of neural network (among the many references available, see for instance [5], [6]). The input to the network is a vector $v \in \mathbb{R}^M$ for some dimension M , which we will denote $v = (v_1, v_2, \dots, v_M)$. Different instances of v are broadcast to a set of N neurons organized in a pretty regular way, say, to fix ideas, in a 2-dimensional grid (other structures are possible, keeping that regularity spirit). We will identify the neuron with its position r in the grid, for instance given by its coordinates in some referential. Associated with neuron r we have a vector of *neuron weights* $w_r \in \mathbb{R}^M$, with the notation $w_r = (w_{r,1}, w_{r,2}, \dots, w_{r,M})$. We say that the state of the network is given by the values of the N vector weights collectively denoted by W .

The network receives a sequence of vector inputs. It reacts to each one by modifying the weights of its neurons. Assume then that a given point in time an input v is received by the network, whose state is W . We say that the *maximal reaction neuron* or *winner neuron* in those circumstances is the neuron indexed by r^* , defined by

$$r^* = \operatorname{argmin}_r \|w_r - v\|, \quad (1)$$

where $\|\cdot\|$ is a vector norm, typically the Euclidean norm: $\|x\| = \sqrt{x_1^2 + \dots + x_M^2}$.

As briefly stated above, the algorithm proposed by Kohonen consists of sequentially showing to the network many possible input vectors taken from the set of all possible ones, and executing for each received input a specific modification of the network’s weights (see below). At the end of the process, the network is frozen (W doesn’t change anymore). The objective of this *learning* process is that if we now present to the network two different inputs v' and v'' , with corresponding maximal reaction neurons r' and r'' , then if v' is “close” to v'' (in \mathbb{R}^M) we must have that r' is “close” to r'' in the neurons’ grid. In mathematics, this is called a *topology preserving* property. In other words, the learning process will divide the grid in regions composed of contiguous neurons, which de facto classifies the inputs in families. We say that the algorithm performs an unsupervised classification task.

The heart of the process is then the weights’ update rule. To achieve the before mentioned main objective, we must first define a reasonable (not too small, nor too large) neighborhood of each neuron. Let us denote by U_r the neighborhood of neuron r (r is called its *center*). Typically, we consider the set of neurons that are close enough to r in the grid, say at a short Manhattan distance. Then, if r^* is the maximal reaction neuron when input v is received, the weights of every neuron $r \in U_{r^*}$ are updated in the following way:

$$w_r^{\text{new}} = w_r^{\text{old}} + \varepsilon h(\|v - w_r^{\text{old}}\|), \quad (2)$$

where $h(\cdot)$ is a function modulating the distance between v and w_r^{old} . A typical example is $h(x) = \exp(-x^2/2\sigma^2)$. The parameter ε controls somehow the effective transformation between old and new

values. The parameter σ controls the decay to zero of the function $h(\cdot)$, that is, how fast $h(x) \rightarrow 0$ as $x \rightarrow 0$. They are called *hyperparameters* of the algorithm (as the definition of neighborhood used, for instance).

Of course, there are more details to specify, some of them mentioned in the more algorithmic description provided in Algorithm 1. Mathematically, the main issue is, of course, how to guarantee that the desired properties of the final weights hold after the learning process. This can be proved for specific cases (see for instance [1], [10], [2]) and observed in practice; the conditions to have them are basically to show to the network “enough” and “varied” (or “representative”) input vectors, and to have not too bad hyperparameters’ values (adjusted by some trial and error procedure, to stay simple).

Algorithm 1: high-level description of a basic SOM

Data: a function $s(\cdot)$ selecting input vectors from some pool, grid dimensions and some indexing system, definition of neighbourhoods, values for the hyperparameters, a condition to stop the iterations

Result: a trained network with the topology preserving property

```

1 Initialize the weights of the network;
2 stopping condition := False;
3 while stopping condition = False do
4   | send new input vector  $v$  to the network using  $s(\cdot)$ ;
5   | compute the maximal reaction neuron  $r^*$  using (1);
6   | for every neuron  $r$  in the neighbourhood of  $r^*$  do
7     | | update the weights of neuron  $r$  using (2);
8   | end
9   | stopping condition :=  $g(W^{\text{old}}, W^{\text{new}})$ ;
10  |  $W^{\text{old}} := W^{\text{new}}$ ;
11 end

```

Comments. The initialization of the weights W (line 1) can be done sorting them with some uniform distribution, or even deterministically. The selection of the input vector (line 4) is typically done using some probability distribution on the set of all possible inputs, but other procedures in specific applications are possible. The stopping condition (line 9) is, in this version, some measure of how different are W^{old} and W^{new} provided by a function $g(\cdot)$. For instance, denote by $W^{(k)}$ the state of the network (the weights) at the end of the k th iteration after this initial bunch of executions of the main block of the algorithm. Then the stopping condition is set to True when the relative difference between $W^{(k+\Delta)}$ and $W^{(k)}$ is small enough, for a fixed Δ (another hyperparameter when this stopping procedure is used). It could also be just a test on the number of iterations already done (that is, the number of input vectors shown to the network), just the indicator $1(k \geq K)$. In this case, the previous “small enough” translates into another parameter of the technique.

In some implementations, it’s useful (even necessary) to use dynamic version of parameters such as ε or σ . Typically, as the number of iterations, say k , increases, it can be useful to decrease ε , that is, to use some decreasing function $\varepsilon(k)$ (starting rather large). It can also be useful to also make σ a function

of k also decreasing.

Kohonen's networks have another interest, sometimes explicitly used. After the learning procedure, we will have not only a classification of the input vectors population in classes, but *de facto* also what is called a *dimensionality reduction*. This refers to the fact that the input data living in a space of dimension M can be now represented by just two numbers (the coordinates of the neurons), something potentially interesting to reduce computational costs.

II. A SEMANTIC EXAMPLE OF APPLICATION OF SOMS

The following example, taken from an article by Ritter and Kohonen [8], [9], presents a semantic problem: the two-dimensional ordering of a collection of concepts represented by vectors. On the one hand, the considerations made by the authors deal with the problem of encoding semantic information; on the other hand, they demonstrate the possibilities of self-organizing maps as classification tools. We start with a set of animals described by a set of properties. Each animal is represented by a binary vector, where the ones indicate the presence of certain properties and the zeros indicate the absence of others.

		dove	hen	duck	goose	owl	hawk	eagle	fox	dog	wolf	cat	tiger	lion	horse	zebra	cow
is	small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
likes to	hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	fly	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 1. Animal names and their attributes

Fig. 1. Animals and their properties. Original figure from the article by Ritter and Kohonen [8] (was also Fig. 1 in the original paper)

To better visualize the simulation results, we assign a colour to each animal, as shown in Figure 2. Using this data, a 20x20 neuron map is trained using the Kohonen algorithm. Weights are initialized with an Uniform(0,1) law. We executed a fixed number $T = 500$ of iterations. The hyperparameters ε and σ are chosen as functions of the iteration counter k as follows: $\varepsilon(k) = 10^{-k/T}$ and $\sigma(k) = 10^{1-k/T}$. Figure 3 shows the final map, using the colors assigned to each animal.



Fig. 2. Colours code for the list of animals.

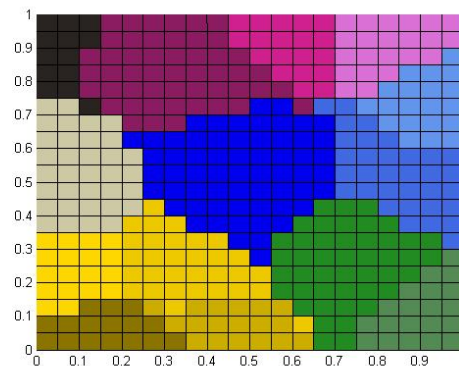


Fig. 3. A realization of the resulting network after 500 iterations.

III. FAMILIARITY, SOMs AND SEMANTIC COGNITION

It is known that the familiarity with a concept is an important element of semantic cognition [11], a fact that could be extremely important in some diseases that affect the semantic system, e.g. in semantic dementia [3]. Patients with semantic dementia make errors in naming tests (especially for animals) when they are shown pictures of them, naming the most familiar element of the corresponding category or simply naming animals that have higher familiarity than others. Many of the responses in these extreme cases are “dog”, “cat” and “horse” only, whatever the animal shown to the patient. Without intending to explain the true underlying process behind this phenomenon, we aim to emulate this behavior with a Kohonen map. To do this, we will modify the example from the previous section.

A. The modified algorithm

Let’s suppose that in the previous algorithm, we focus only on four groups of animals: the privileged three – dog, cat, and horse – and a fourth category consisting of the rest of the animals. Figure 4 represents a particular realization of the algorithm at epoch $T = 500$. The concepts of “dog”, “cat” and “horse” are represented in red, green, and blue, respectively, while the other animals are represented in orange.

We can now introduce the concepts of familiarity and deterioration into the discussion. We will gradually modify the probability of presenting stimuli to the Kohonen algorithm, making little by little more probable the vectors corresponding to the concepts “dog”, “cat” and “horse”. They will be presented in epoch k with

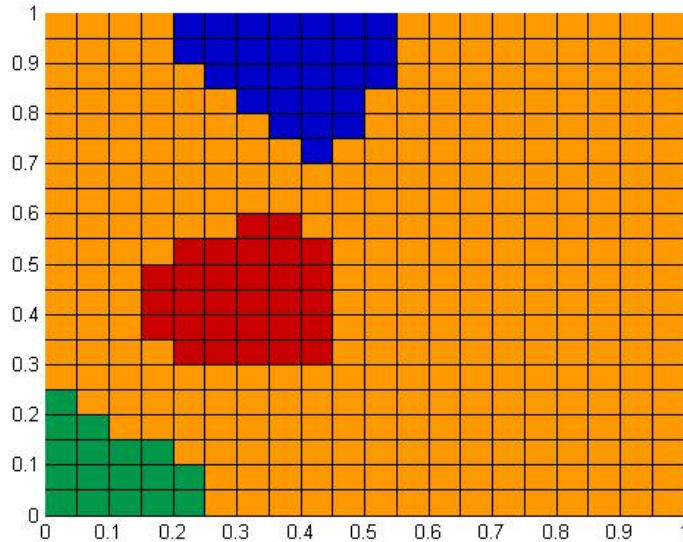


Fig. 4. A realization of a Kohonen map with 500 replications for the animal set described earlier.

probability $(1 - \mu)/3$, and the rest of the stimuli (from Figure 1) will be presented with probability $\mu/13$. The parameter μ is initially set to $13/16$ (ensuring equiprobability among the 16 concepts) and it is from time to time decreased as shown in the description of the procedure. On the other hand, we introduce the concept of aging into the model: starting from epoch $k = 400$ in the experiments shown here, atrophy is introduced with a probability p . This consists of randomly and uniformly erasing neurons from the network, along with some of their neighbors. With probability p , it is decided whether atrophy occurs, and if it does, a neuron is uniformly chosen on a set of close neighbors and then removed. The resulting procedure is described in Algorithm 2. Observe that for clarity reasons, we simply make a fixed number of iterations in this version, but we can use any stopping criteria as discussed for Algorithm 1.

Let us call $f(\cdot)$ the discrete probability mass function parameterized by μ defined over the 16 animals in our list by $f(\text{“dog”}) = f(\text{“cat”}) = f(\text{“horse”}) = (1 - \mu)/3$, and for every other animal a , $f(a) = \mu/13$. Figure 5 shows the resulting map after $T = 500$ replications of the modified algorithm. We start the atrophy simulation at epoch k_0 here set to 400.

Based on these modifications, we can consider the classification that an impaired cortex (where the word *cortex* stands for *Kohonen’s cortex*, represented by the network) could make for a set of new animals examples (under more realistic representations, this type of disturbance could represent the existing diversity within each animal species). These are obtained by perturbing the original ones. Given a vector v belonging to the initial list, we generate vector v' defined by $v' = v + X$, where X is a vector of independent Gaussian variables, each with mean 0 and standard deviation used as another parameter of the model to be adjusted.

Consider the classification $c^{(1)}(v')$ of vector v' using the original examples:

$$c^{(1)}(v') = \operatorname{argmin}_{v \in V} \| v' - v \| .$$

Consider now the classification $c^{(2)}(v')$ of the same v' made by the impaired cortex, that is, by the set of

Algorithm 2: our proposed modification of SOM

Data: function $s(\cdot)$ selecting input vectors, grid dimensions and indexing, definition of neighbourhoods, hyperparameters, etc.

Result: a trained network with the topology preserving property

```

1 Initialize the weights of the network;
2  $\mu := 13/16$ ;
3 for  $k = 1, 2, \dots, T$  do
4   | sample a new input vector  $v$  using  $f(\cdot)$ ;
5   | compute the maximal reaction neuron  $r^*$  using (1);
6   | for every neuron  $r$  in the neighbourhood of  $r^*$  do
7   |   | update the weights of neuron  $r$  using (2);
8   | end
9   | if  $k \bmod 10 = 0$  then //  $k$  is a multiple of 10
10  |   |  $\mu := \mu / \ln(\ln(k + 1))$ ;
11  |   | end
12  |   | if  $k > k_0$  then
13  |   |   | randomly kill neurons;
14  |   | end
15  |   |  $W^{\text{old}} := W^{\text{new}}$ ;
16 end

```

weights resulting from the evolutionary process, given by

$$c^{(2)}(v') = \operatorname{argmin}_{v \in V} \|v - w_{r'}\|,$$

where $r' = \operatorname{argmin}_r \|v' - w_r\|$. See that $c^{(2)}(v') = c^{(1)}(w_{r'})$.

Looking at $c^{(1)}$ as the correct one, we may be interested in seeing how many examples from a new set V' are correctly classified by the demented cortex (resulting from a number k of training epochs with the modified Kohonen algorithm). The number of correct classifications NA by the impaired cortex is then given by $NA = \sum_{v' \in V'} 1_{\{c^{(2)}(v') = c^{(1)}(v')\}}$. Figure 6 shows the mean variation of NA from 50 frozen demented cortex realizations at epoch 500.

IV. CONCLUSIONS

This preliminary text shows the potential of SOMs for building some simple models of an extremely complex system that includes the representation of semantic concepts in the brain and the development of a devastating cognitive disease. This paper shows a way of representing semantic information in the brain and its degradation, mimicking the evolution of a specific cognitive disease. The main next step is to make this representation more complex, with the goal of making the cognitive loss of the subject a consequence of the degradation.

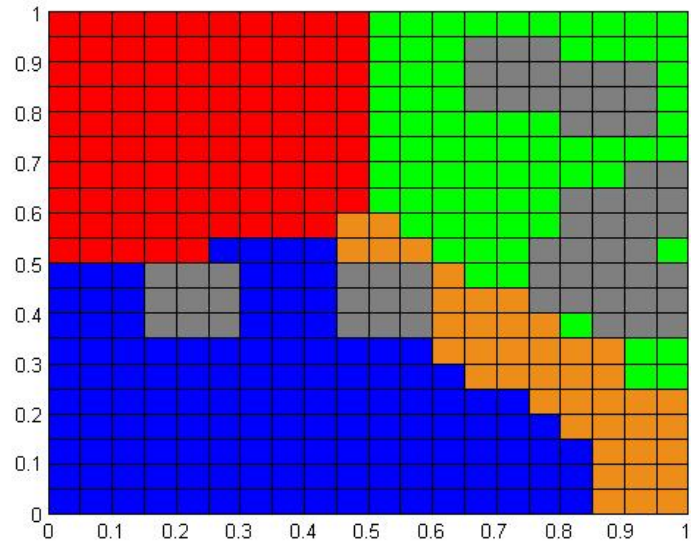


Fig. 5. A realization of a Kohonen map with 500 replications for the animal set of Section 1. Atrophies are represented in gray. Also, note the proportions of the resulting network areas representing the concepts “dog”, “cat” and “horse”.

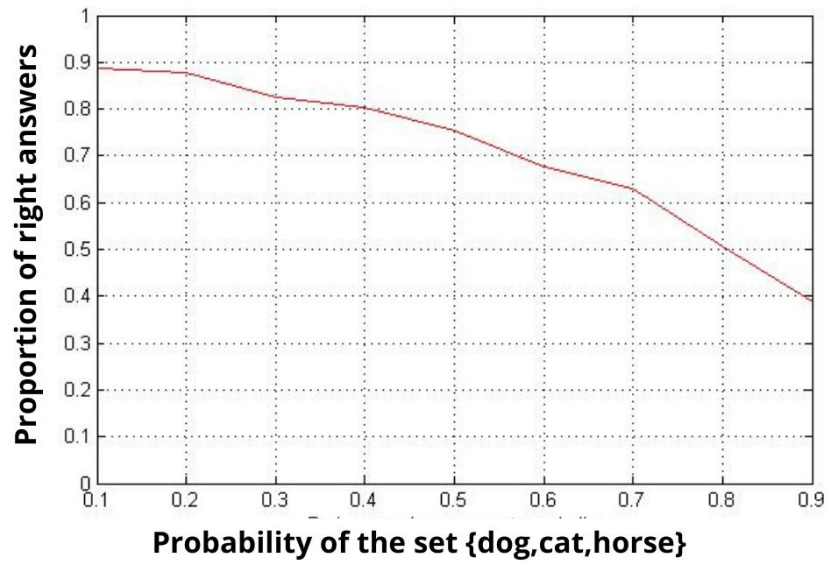


Fig. 6. The average variation of the number of correct classifications from 50 frozen impaired cortex realizations at epoch 500 is shown. The x-axis represents the sum of probabilities of the privileged examples: dog, cat, and horse. In other words, a simulation is conducted for each of the mentioned initial probabilities.

REFERENCES

- [1] Cottrell, M., Fort, J.C. and Pagès, G., “Theoretical aspects of the SOM algorithm”, in Proceedings of the WSOM 97, workshop on self-organizing maps, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, pp. 246—267, 1997.
- [2] Heskes, T., On the convergence properties of the self-organizing map. *Journal of Artificial Intelligence Research* 1, pp. 205-227, 1993.
- [3] Hodges J., Graham N., Patterson K., “Charting the progression in semantic dementia”, *Implications for the organisation of semantic memory*. *Memory* 3, pp. 463–495, 1995.
- [4] Hodges J.R., Patterson K. 2007. “Semantic dementia: a unique clinicopathological syndrome.” *The Lancet Neurology*, Elsevier; 6, pp. 1004–1014.
- [5] Kohonen T., “Self-Organized Formation of Topologically Correct Feature Maps”, *Bio. Cybern.*, 43, pp. 59–69, 1982.
- [6] Kohonen, Teuvo, “Essentials of the self-organizing map”, *Neural networks*, Vol.37, pp. 52–65, 2013
- [7] Mesulam M.M., “Slowly progressive aphasia without generalized dementia”, *Ann. Neurol.* 11, pp. 592–598, 1982.
- [8] Ritter, H., Kohonen, T., “Self-organizing semantic maps”, *Biological cybernetics*, 61(4), pp. 241-254, 1989.
- [9] Ritter H., Martinetz T., Schulten K., “Neural Computation and Self-Organizing Maps - An Introduction”, Addison-Wesley, New York, 1992.
- [10] Ritter, H., Schulten, K. “Convergence properties of Kohonen’s topology conserving maps: fluctuations, stability, and dimension selection”, *Biological Cybernetics*, 60(1), pp. 59–71, 1988.
- [11] Rogers T., Patterson K. Jeffereis E. Lambon Ralph M., “Disorders of representation and control in semantic cognition: Effects of familiarity, typicality, and specificity”, *Neuropsychologia* 76, pp. 220–239, 2015.
- [12] Snowden J.S., Goulding P.J., Neary D., “Semantic dementia: a form of circumscribed atrophy”, *Behav. Neurol.* 2, pp. 167–182, 1989.
- [13] Tulving E. “Episodic and semantic distinction.” In : Tulving E, Donaldson W, eds. “Organization of memory”, Academic Press, New York, 1972, pp. 381–403, 1972.
- [14] Warrington, E. K., “The selective impairment of semantic memory”, *Quarterly Journal of Experimental Psychology*, 27, pp. 635—657, 1975.