Complex Systems **6** (1992) 105–123

Kohonen Self-Organizing Maps: Is the Normalization Necessary?

Pierre Demartines*
François Blayo
Laboratoire de Microinformatique, Ecole Polytechnique Fédérale de Lausanne, INF-Ecublens, CH-1015 Lausanne, Switzerland

Abstract. The self-organizing algorithm of Kohonen is well known for its ability to map an input space with a neural network. According to multiple observations, self organization seems to be an essential feature of the brain. In this paper we focus on the distance measure used by the neurons to determine which one is closest to an input stimulus. The distance measures proposed until now are not very satisfactory, from either a biological or computational point of view. Using mathematical considerations and numerical simulations, we show that the original dot product measure is applicable without input normalization when the dimension of the input space is high. When adding a feature of biological neurons (accommodation) to the algorithm, the network converges with normalization as well (in our simulations, for a dimension $n > 12$).

1. Introduction

In many biological systems, and especially in the cerebral cortex, various areas are organized according to different sensory modalities. Some of them perform specialized tasks, such as speech control or analysis of visual or auditory signals. Between these areas, the associative areas reveal a fine structure that corresponds to a topographical order that depends on sensory input. Marshall and Talbot found, for example, that the primary visual cortex contains a map of the retina in which neighborhood relations are preserved [12].

Starting from this observation, Kohonen proposed an original algorithm that realizes a mapping of a high-dimensional input space in an output space whose dimensionality is lower or the same, where neighborhood relationships are preserved.

*Current address: Laboratoire de Traitement d'Images et de Reconnaissance des Formes, Institut National Polytechnique de Grenoble, 46 av. Félix-Viallet, F-38031 Grenoble, France

2. The original self-organizing algorithm

In the initial version of the algorithm [7], Kohonen defined a network of neurons whose interconnectivity is dissociated into two parts: a simple associative memory layer between the input and the neurons, and a lateral layer interconnecting the neurons locally. The purpose of the associative layer consists of coding the synaptic weights in order to build incrementally an associative memory, depending on the succesive presentations of the input stimuli. The second layer realizes a kind of contrast enhancement that creates a cluster centered around the local maximum of the response to an input stimulus. The combination of the two layers, associated with a suitable adaptation rule, leads to a spatial arrangement of the neurons in the weight coordinate system in which neighboring neurons respond to neighboring stimuli.

The degree of lateral coupling between neurons in the second layer is defined by a "Mexican hat" function. In a short-range lateral-coupling distance the function is excitatory, whereas it is inhibitory in a long-range distance. This function reproduces, for instance, the response of the on-center receptive field of the neuron in the retina [9]. The relaxation phase that follows the creation of the neuron activity converges to a stable state with the formation of an activity cluster around the maximum activity neuron. This process is the phenomenon most responsible for self organization.

3. The simplified algorithm

Despite the biological interest of this model, it is most convenient for the purpose of conventional computer simulations to express algorithmically the function of each layer rather than the way to realize it. Then a computational algorithm would consist of two steps: finding the neuron whose activity is maximum with respect to an input stimulus, then defining a subset of neurons in the network around this maximum, corresponding to the cluster. The weight vectors of the neurons in this subset are modified in the direction of the input vector. The repetition of these two steps conducts the network being organized. This simplified version of the algorithm [8] is described below in detail.

We consider a vector $\mathbf{x}$ that is composed of a set of n scalar signals $[x_1, x_2, \ldots, x_n]^T$ and a set of weights $\mathbf{W}_i = [W_{i1}, W_{i2}, \ldots, W_{in}]^T$ that represent the synaptic efficiency between the input and the neuron i ($1 \leq i \leq N$, where N is the total number of neurons). Let us define a similarity criterion, for instance, the dot product between $\mathbf{x}$ and $\mathbf{W}_i$, or any other similarity measure of the distance $\delta(\mathbf{x}, \mathbf{W}_i)$. Then the index k of the neuron presenting the best response is determined by the condition

$$\delta(\mathbf{x}, \mathbf{W}_k) = \min_{1 \leq i \leq N} \delta(\mathbf{x}, \mathbf{W}_i) \tag{1}$$

Next, around this maximally responding neuron k, we choose a topological neighborhood $V_k(t)$ such that all neurons that lie within a defined radius

of neuron k are included in $V_k(t)$. All neurons located in $V_k(t)$ have their weights updated according to the following adaptation rule, expressed in the discrete-time index t:

$$\mathbf{W}_i(t+1) = \mathbf{W}_i(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{W}_i(t)] \Big|_{i \in V_k(t)} \tag{2}$$

The other neurons have their weights unchanged.

It must be pointed out that the neighborhood radius is generally chosen as a decreasing function of time, as well as of the gain parameter α. This last parameter can be chosen using a method inspired by the well-known gradient-descent method [6]. Then the distribution density function of the vectors $\mathbf{W}_i$ converges toward a discretized image of the probability density function $p(x)$ of the input stimuli x [7]. A formal demonstration has been proposed by Cottrell and Fort [2] in the case of one-dimensional networks with one-dimensional input space.

4. The problem of distance measure

There are several commonly used distance measures in the simplified algorithm that determine the "winner" unit, that is, the unit whose weight vector is nearest to the input vector [8]. The three most often used measures (between an input vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ and the weight vector $\mathbf{W}_i = [W_{i1}, W_{i2}, \ldots, W_{in}]^T$ of neuron i) are:

$$\text{Euclidean distance:} \quad \delta(\mathbf{x}, \mathbf{W}_i) = \|\mathbf{x} - \mathbf{W}_i\| = \sqrt{\sum_{j=1}^{n} (x_j - W_{ij})^2} \tag{3}$$

$$\text{Manhattan distanca:} \quad \delta(\mathbf{x}, \mathbf{W}_i) = \sum_{j=1}^{n} |x_j - W_{ij}| \tag{4}$$

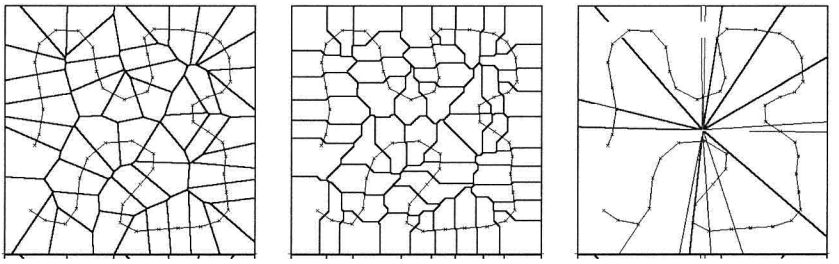
$$\text{Dot product:} \quad \delta(\mathbf{x}, \mathbf{W}_i) = \mathbf{x} \cdot \mathbf{W}_i = \|\mathbf{x}\| \, \|\mathbf{W}_i\| \cos(\mathbf{x}, \mathbf{W}_i) \tag{5}$$

When both weight and input vectors are normalized, $\|\mathbf{x}\| \, \|\mathbf{W}_i\|$ is a constant, and the dot product becomes a valid measure of proximity. In this case, the maximum result for expression (5) gives the winner unit index.

The *domination region* of a particular neuron is the part of the input space in which this neuron wins the competition. The representation of these domination regions for the whole network shows how the input space is divided into subregions (quantization property). Considering this representation, the expected results of the self-organization process are:

1. that the probability of receiving an input vector is the same for each region (first main property of the self-organizing maps [8]); if the input space is two dimensional, and its density function is uniform, the regions should have the same area;

2. that the regions are related (not split) and have a "reasonable" shape (without big outgrowths);



Euclidean distance Manhattan distance Dot product distance

Figure 1: Regions of domination without weight normalization.

3. that exactly one neuron is responsible for one region, with its weight vector *inside* the region.

More generally, the self-organization process should produce a network in which:

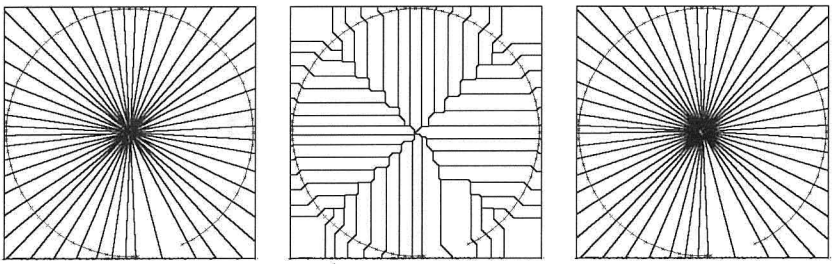
1. there is a correct mapping of the input space (each neuron has the same probability of being excited);

2. there is a topological conservation of the input space (the neighborhood relationships are preserved).

The fact that the units have ordered vector weights—so the input space is mapped by the network with topological conservation (the second property)—is not shown with only the representation of the domination regions. In figures 1 and 2 this ordering is emphasized with the fine lines connecting neighboring neurons (the widely used representation of self-organizing maps in the weight space).

When the distance measure is the euclidean distance, this partitioning in domination regions is a *Voronoi tessellation*. Using another distance measure (but keeping the same network configuration), the shape of these regions changes, as shown in figure 1. In this figure the domination regions obtained using three different distance measures are superposed on the network representation in the weight space. The input space is two dimensional, whereas the network is one dimensional. The weights are not normalized.

Without normalization, the dot product measure gives inconsistent regions (there is more than one neuron in several regions, and no neuron in some other regions). Using this measure in a two-dimensional input space without weight normalization, the network is not able to self organize properly (it does not achieve the expected vector quantization of the input space). On the contrary, when the weight and input vectors are normalized, one dimension of the input space is lost, but the network is able to self organize using the dot product measure.



Euclidean distance Manhattan distance Dot product distance

Figure 2: Regions of domination with weight normalization.

Remarks.

1. *It is not necessary to normalize both weight and input vectors to obtain the self organization with the dot product measure.*

   - *Kohonen [8] argued that the normalization of input vectors alone is sufficient to obtain the convergence of the network because the weights are moved close to the input vectors, which are normalized, and become almost normalized automatically.*
   - *On the other hand, it is easy to see that the normalization of input vectors is not necessary when the weight vectors are normalized: if* $\|\mathbf{W}_i\|$ *is a constant* $\forall i$*, then* $\|\mathbf{x}\|\,\|\mathbf{W}_i\|$ *is also a constant for a given input vector* $\mathbf{x}$*, so the comparison of all distances* $\delta(\mathbf{x}, \mathbf{W}_i)$ *gives the unit for which* $\cos(\mathbf{x}, \mathbf{W}_i)$ *is maximum.*

2. *There is another way to normalize input vectors without losing one dimension [8]: project the input vectors onto a hypersphere of* $n + 1$ *dimensions, where* n *is the dimension of the initial input space. Nevertheless, one should know the range of input variables to ensure that the hypersphere will be large enough (otherwise the projection will be impossible for certain input vectors).*

When the weight vectors are normalized, the euclidean distance and the dot product measure are equivalent because $\|\mathbf{x} - \mathbf{W}_i\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{W}_i\|^2 - 2\,\mathbf{x} \cdot \mathbf{W}_i$, so $\min(\|\mathbf{x} - \mathbf{W}_i\|^2) = \max(\mathbf{x} \cdot \mathbf{W}_i)$ if $\|\mathbf{W}_i\|$ is constant. This can be verified visually by observing the similarity of the domination regions for these two distance measures, as shown in figure 2.

All of these computational considerations are unsatisfactory for many reasons, however. From the VLSI point of view, the cost in operators is against the euclidean distance measure implementation, as shown by Vittoz [14]. Furthermore, from an organizational quality point of view, a network that uses the Manhattan distance shows a predilection to be axis oriented, and does not cover the input distribution as well as other distance measures, as is

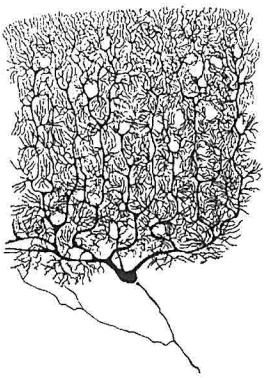


Figure 3: Purkinje cell of the cerebellum cortex. Such a cell has up to 100,000 input connections. [From S. R. Cajal, *Histologie du Systeme Nerveux de l'Homme et des Vertebres*, trans. by L. Azoulay (Paris, Maloine, 1909)].

observed with simulations for a two-dimensional input space. Moreover, formal neurons generally compute neither a euclidean nor Manhattan distance between their inputs and their synaptic weights. On the contrary, the most widely used neuron model performs a dot product between its input vector and its weight vector [10], and seems to be more plausible from the biological point of view. For this reason, the dot product is the generic operation used in VLSI realizations in which several neural networks are implemented [1].

Unfortunately, as we said before, when the dot product is used a normalization operation has seemed until now to be necessary. In VLSI implementations, this normalization operation is time- and area-consuming, and reduces the possible integration of a large number of operators. However, despite some general considerations previously proposed [13], it is not proved that an explicit normalization device exists in biological systems; so how can it work? The following sections show that, under certain conditions, the Kohonen network is able to converge without any normalization operation even using the dot product as the distance measure.

5. Biological systems work on high-dimensional spaces

Considering biological systems, we should be surprised by the large number of connections that converge to only one neuron (see figure 3). Most of the neurons receive between 1000 and 10,000 connections with other cells, and sometimes (as in figure 3 in a Purkinje cell) up to 100,000 connections. Actually, it represents as many dimensions as in the input space.

We have seen that normalization is needed with spaces of two or three dimensions. What happens when the input space has a dimension of 1000? The computing time or the VLSI surface required for the normalization oper-

ation becomes prohibitive, but perhaps it is no longer necessary to normalize inputs or weights.

6. Experimental distribution of the norm of a random vector

The main question is, what is the euclidean norm of a vector of random components when the number of components is large enough? More formally, considering a vector $\mathbf{x}$ composed of n independent random values ($\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$), what are the expectation value $\mu_{\|\mathbf{x}\|} = \mathrm{E}(\|\mathbf{x}\|)$ and the variance $\sigma^2_{\|\mathbf{x}\|} = \mathrm{Var}(\|\mathbf{x}\|)$ as functions of n (with $\|\mathbf{x}\|$ the euclidean norm of $\mathbf{x}$)?

For a vector of dimension n varying from 1 to 1000, and for several x_k distributions, we obtained by simulation the numerical results shown in figure 4. The most interesting fact in figure 4 is that, for any distribution, the standard deviation $\sigma_{\|\mathbf{x}\|}$ of the norm converges asymptotically to a constant, while the mean $\mu_{\|\mathbf{x}\|}$ grows as a power of $1/2$ of n. Because of the Chebychev inequality

$$P\left(\left|\|\mathbf{x}\| - \mu_{\|\mathbf{x}\|}\right| \geq \varepsilon\right) \leq \frac{\sigma^2_{\|\mathbf{x}\|}}{\varepsilon^2} \tag{6}$$

the probability that the norm $\|\mathbf{x}\|$ falls outside a fixed-width interval centered on $\mu_{\|\mathbf{x}\|}$ becomes approximately a constant (as $\sigma_{\|\mathbf{x}\|}$ also becomes approximately a constant). The consequence of this fact is that the relative error committed when taking $\mu_{\|\mathbf{x}\|}$ instead of $\|\mathbf{x}\|$ becomes negligible.

Remembering the Kohonen argument about normalization of input vectors (remark 1, section 4), if all the input vectors $\mathbf{x}$ have a norm very close to $\mu_{\|\mathbf{x}\|}$—as these observations suggest for high-dimensional spaces—the explicit normalization operation should no longer be necessary.

In section 7 we formalize and generalize this observation, and in section 8 we describe some simulations of a Kohonen network that maps a high-dimensional input space.

7. Mathematical results on the norm of a random vector

Let $f(x_k)$ be an arbitrary distribution law for the components x_k, with mean $\mu = \mathrm{E}(x_k)$ and variance $\sigma^2 = \mathrm{Var}(x_k)$; what are the expressions of $\mathrm{E}(\sqrt{\sum_{k=1}^n x_k^2})$ and $\mathrm{Var}(\sqrt{\sum_{k=1}^n x_k^2})$ (the mean and the variance of the vector norm $\|\mathbf{x}\|$)? If the x_k are independent, the central limit theorem states that the variable $\|\mathbf{x}\|^2 = \mathcal{S}^2 = \sum_{k=1}^n x_k^2$ converges to a normal variable when n is statistically "large enough" (i.e., in practice when $n > 30$). We get:

$$\text{Mean:} \qquad \mu_{\mathcal{S}^2} = \mathrm{E}(\mathcal{S}^2) = n(\sigma^2 + \mu^2) \tag{7}$$

$$\text{Variance:} \qquad \sigma^2_{\mathcal{S}^2} = \mathrm{Var}(\mathcal{S}^2) = n(4\mu^2\sigma^2 - \sigma^4 + 4\mu\mu_3 + \mu_4) \tag{8}$$

with μ_k the moment of order k, relative to the origin.

Now the problem is to find $\mu_{\|\mathbf{x}\|} = \mathrm{E}(\sqrt{\mathcal{S}^2})$ and $\sigma^2_{\|\mathbf{x}\|} = \mathrm{Var}(\sqrt{\mathcal{S}^2})$. Equations (7) and (8) show that the exact and general results (if they exist) depend

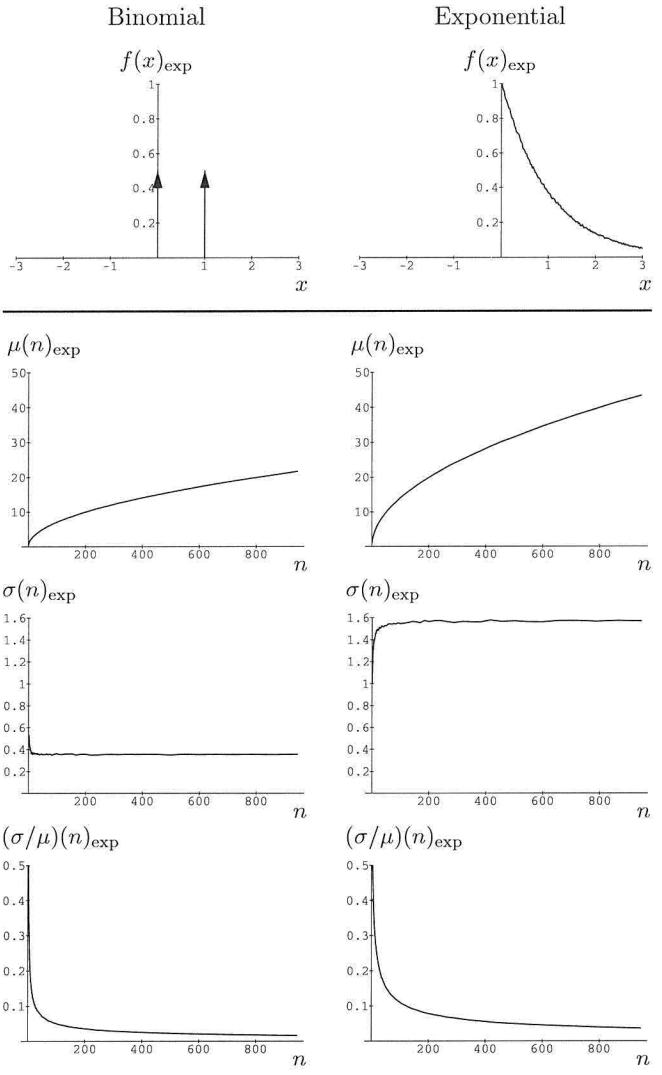


Figure 4: Mean and standard deviation of the norm of random vector $\mathbf{x}$. Several distributions are considered for the components x_k.

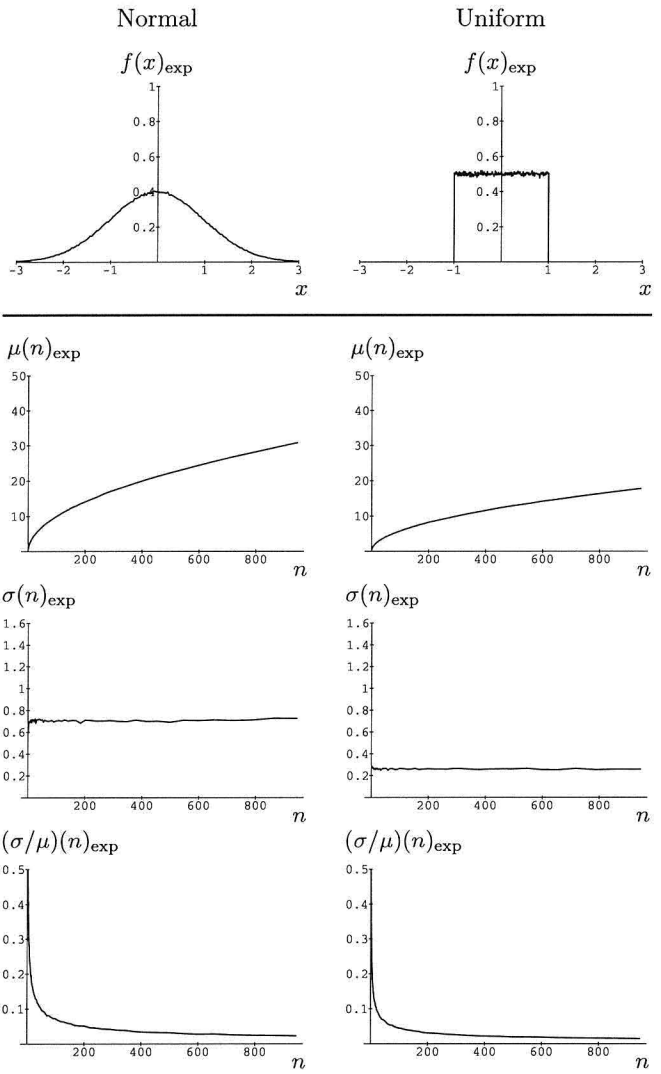


Figure 4: (*Continued*)

at least on the moments of orders 1, 2, 3, and 4 of x_k. J. C. Fort has proved that, if the n x_k are independent (and have a finite moment of order 8),

$$\mathrm{E}(\sqrt{\mathcal{S}^2}) = \sqrt{\mu_{\mathcal{S}^2} - \frac{\sigma^2_{\mathcal{S}^2}}{4\mu_{\mathcal{S}^2}} + O\left(\frac{1}{n}\right)} \tag{9}$$

$$\mathrm{Var}(\sqrt{\mathcal{S}^2}) = \mathrm{E}(\mathcal{S}^2) - \mathrm{E}(\sqrt{\mathcal{S}^2})^2 = \frac{\sigma^2_{\mathcal{S}^2}}{4\mu_{\mathcal{S}^2}} + O\left(\frac{1}{\sqrt{n}}\right) \tag{10}$$

With equations (7), (8), (9), and (10), we obtain the generic formulation

$$\mu_{\|\mathbf{x}\|} = \mathrm{E}(\|\mathbf{x}\|) = \mathrm{E}(\sqrt{\mathcal{S}^2}) \cong \sqrt{a\,n - b} \tag{11}$$

$$\sigma_{\|\mathbf{x}\|} = \mathrm{Var}(\|\mathbf{x}\|) = \mathrm{Var}(\sqrt{\mathcal{S}^2}) \cong b \tag{12}$$

with a and b constants depending only on the x_k distribution law,

$$a = \frac{\mu_{\mathcal{S}^2}}{n} = \sigma^2 + \mu^2 \tag{13}$$

$$b = \frac{\sigma^2_{\mathcal{S}^2}}{4\mu_{\mathcal{S}^2}} = \frac{4\mu^2\sigma^2 - \sigma^4 + 4\mu\mu_3 + \mu_4}{4(\sigma^2 + \mu^2)} \tag{14}$$

For some usual distribution laws, we obtain the parameters a and b shown in table 1. Using the conventional functions

$$\mathbf{1}(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}, \qquad \delta(x) = \begin{cases} 0, & x \neq 0 \\ \infty, & x = 0 \end{cases}, \qquad \int_{-\infty}^{+\infty} \delta(x)dx = 1,$$

these results confirm the observations given in section 6.

8. Simulations of a Kohonen map in multidimensional space

These simulations, made with the software developed in our laboratory [4], compare the self-organization results (using the dot product distance measure) *with* input and weight vector normalization, and *without* any normalization.

The simulated network is a rectangular 30-by-30 grid of neurons. The learning algorithm is the simplified algorithm (winner take all, decreasing alpha and neighborhood) with the dot product distance measure. The input space dimension n varies from 3 to 200, and the maximum number of iterations is 20,000. The random distribution is uniform between -1 and $+1$ for each input component.

When the input space has two dimensions, it is easy to *see* whether the network becomes well organized because it is possible to represent the network units in the weight space (as in figures 1 or 2). However, when the input dimension is 200, qualifying the organization becomes impossible using the same means.

Distribution	$f(x)$	a	b
Uniform $U(\alpha,\beta)$	$\frac{\mathbf{1}(x-\alpha)-\mathbf{1}(x-\beta)}{\beta-\alpha}$	$\frac{\alpha^2+\alpha\beta+\beta^2}{3}$	$\frac{4\alpha^4-\alpha^3\beta-6\alpha^2\beta^2-\alpha\beta^3+4\beta^4}{60(\alpha^2+\alpha\beta+\beta^2)}$
$U(-\beta,+\beta)$	$\frac{1}{2}[\mathbf{1}(x+\beta)-\mathbf{1}(x-\beta)]$	$\beta^2/3$	$\beta^2/15$
$U(-1,+1)$	$\frac{1}{2}[\mathbf{1}(x+1)-\mathbf{1}(x-1)]$	$1/3$	$1/15$
Normal $N(\mu,\sigma)$	$\frac{1}{\sigma\sqrt{2\pi}}\exp\frac{-(x-\mu)^2}{2\sigma^2}$	$\sqrt{\mu^2+\sigma^2}$	$\frac{\sigma^4+2\sigma^2\mu^2}{2(\mu^2+\sigma^2)}$
$N(0,1)$	$\frac{1}{\sqrt{2\pi}}\exp\frac{-x^2}{2}$	1	$1/2$
Exponential $E(\alpha)$	$\alpha\exp(-\alpha x)\,\mathbf{1}(x)$	$\frac{2}{\alpha^2}$	$\frac{5}{2\alpha^2}$
$E(1)$	$\exp(-x)\,\mathbf{1}(x)$	2	$5/2$
Bernoulli $B(\alpha,\beta)$	$\frac{1}{2}[\delta(x-\alpha)+\delta(x-\beta)]$	$\frac{\alpha^2+\beta^2}{2}$	$\frac{(\alpha^2-\beta^2)^2}{8(\alpha^2+\beta^2)}$
$B(-\beta,+\beta)$	$\frac{1}{2}[\delta(x+\beta)+\delta(x-\beta)]$	β^2	0
$B(0,1)$	$\frac{1}{2}[\delta(x)+\delta(x-1)]$	$1/2$	$1/8$

Table 1: Matching parameters a and b of equations (11), (12), (13), and (14) computed for the distribution laws of figure 4.

9. How to measure the organization

When working in two dimensions and with a uniform input distribution, we say that a network is *well* organized if the grid is regular, that is, if the distance between the units is almost constant. We try to extend this concept to n dimensions with a more precise criterion to qualify the organization.

Let us define a "disorder" level based on the statistical distribution of the euclidean distance measure between the weights of consecutive network units (in our simulations, the grid is rectangular, but the criterion is extensible to other neuron dispositions, such as a hexagonal grid):

$$\Theta = \frac{\sigma_\Delta}{\mu_\Delta} \tag{15}$$

with

$$\mu_\Delta = \frac{\sum_{i=1}^{h}\sum_{j=1}^{w-1}\Delta \mathrm{h}_{ij} + \sum_{i=1}^{h-1}\sum_{j=1}^{w}\Delta \mathrm{v}_{ij}}{2wh-w-h} \tag{16}$$

$$\sigma_\Delta = \sqrt{\frac{\sum_{i=1}^{h}\sum_{j=1}^{w-1}(\Delta \mathrm{h}_{ij}-\mu_\Delta)^2 + \sum_{i=1}^{h-1}\sum_{j=1}^{w}(\Delta \mathrm{v}_{ij}-\mu_\Delta)^2}{2wh-w-h}} \tag{17}$$

$$\Delta h_{ij} = \|\mathbf{W}_{ij} - \mathbf{W}_{i,j+1}\| \tag{18}$$

$$\Delta v_{ij} = \|\mathbf{W}_{ij} - \mathbf{W}_{i+1,j}\| \tag{19}$$

In these equations, w is the width and h the height of the network grid (in our simulations, $w = h = 30$). Δh_{ij} is the euclidean distance between the weights of the units ij and $i, j+1$ in the grid (horizontal intervals), and Δv_{ij} is the euclidean distance between the weights of the units ij and $i+1, j$ (vertical intervals). $2wh - w - h$ is the number of intervals in the network grid. μ_Δ represents the mean distance between the weights of two adjacent units in the network, and σ_Δ the standard deviation of this distance.

In a perfectly regular grid (where all intervals are equal), σ_Δ tends to zero, as does the "disorder" level Θ. On the contrary, the more irregular the grid is, the greater is Θ.

To simplify the notation and the explanations of the result curves shown later, we consider Θ as a function of two parameters: the dimension n and the number of learning iterations k ($\Theta(n, k)$).

Several curves for $\Theta(n, k)$ are given in sections 11 and 13. That these curves are all decreasing suggests that Θ could be a Lyapunov function, in which case we would be able to demonstrate the convergence for any dimension. We are now working on this problem in collaboration with M. Cottrell and J. C. Fort, who have demonstrated the convergence in one dimension with the organization criterion defined by Kohonen [8].

10. An alternative representation of the network valid for any dimension

We define also another method for *visualizing* the network organization, the results of which look like those produced with the usual weight-positioning representation method. In contrast to the Θ function method, this method cannot be used to represent the evolution of the organization, but is useful for characterizing the state of the network. In this representation, which can be interpreted as the unfolding of the network grid onto a plane, the units are positioned step by step (starting from the center of the grid) as a function of the preceeding units and the distance between the respective weight vectors. In the upper right quarter of the grid, for example, the position of the unit ij is defined as

$$P_x^{ij} = P_x^{i,j-1} + \|\mathbf{W}_{ij} - \mathbf{W}_{i,j-1}\| \tag{20}$$

$$P_y^{ij} = P_y^{i-1,j} + \|\mathbf{W}_{ij} - \mathbf{W}_{i-1,j}\| \tag{21}$$

with

$$P_x^{(h/2)j} = 0, \tag{22}$$

$$P_y^{i(w/2)} = 0 \tag{23}$$

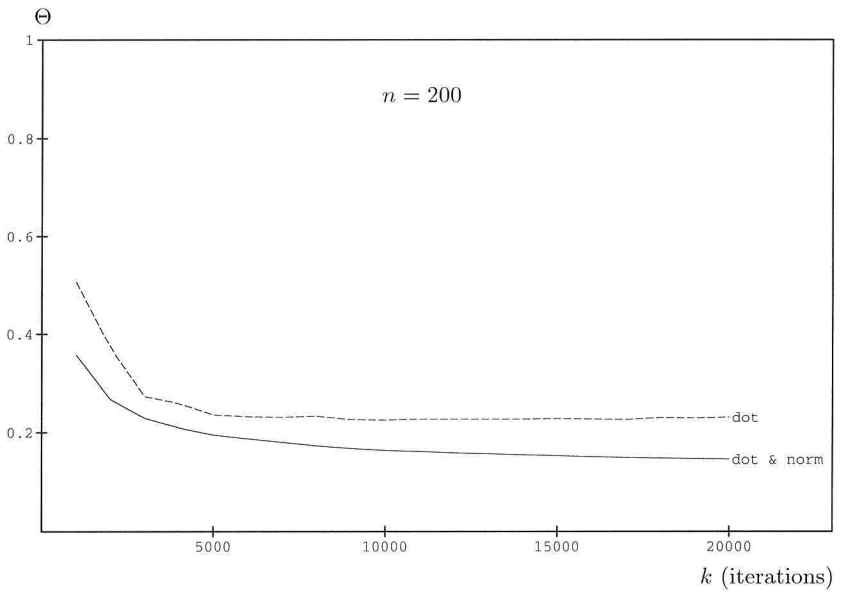


Figure 5: Evolution of $\Theta(n,k)$ with $n = 200$ and $k = 1$ to 20,000. Solid line: with input and weight normalization. Dashed line: no normalization.

The expressions for the three other quarters may be inferred by symmetry. The algorithm implemented to build this representation is recursive and begins with one of the two limit expressions (22) and (23).

Some examples of this representation are given in section 11 (right part of figures 7*a* and 7*b*) to illustrate the network organization more intuitively. We call this representation a *curvilinear representation.*

11. First results and discussion

Figure 5 shows the evolution of the disorder level $\Theta(n,k)$ (as defined in section 9) with $n = 200$ and k varying from 0 to 20,000. This is the temporal evolution of a fixed input dimension network. The solid line corresponds to a network with input and weight vector normalization, while the dashed line corresponds to a network without any normalization. Both networks use the dot product distance measure. In the both cases, the Θ function is a decreasing function of k (the number of iterations). It means that, from our criterion point of view, both corresponding networks self organize.

We performed the same experiment with the input dimension $n = 3$ (not 2, because it makes no sense to map with a two-dimensional network the circle produced by a normalized distribution). This experiment confirms a well-known result: without normalization and in a low-dimensional input space, the network does not self organize at all, as shown in figure 6 with the

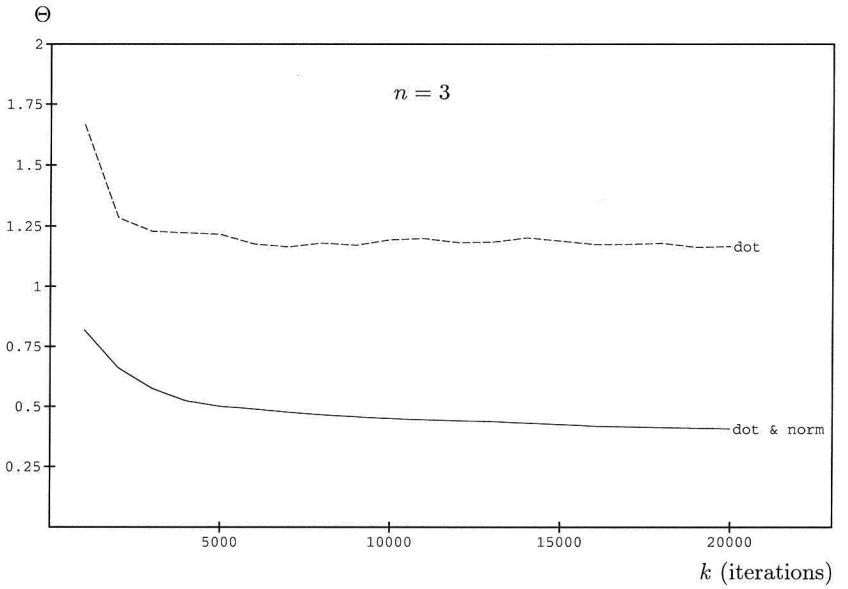


Figure 6: Evolution of $\Theta(n,k)$ with $n = 3$ and $k = 1$ to 20,000. Solid line: with input and weight normalization. Dashed line: no normalization.

non-decreasing behavior of the dashed line, representing $\Theta(n,k)$ with $n = 3$ and without normalization. This figure (figure 6) shows the same curves as in figure 5, but with the input dimension $n = 3$ instead of 200.

These experiments show that the quality of organization, without normalization, is much better in high-dimensional input space than in low-dimensional input space. However, even in 200 dimensions, the result without normalization is not as good as with input and weight vector normalization, as shown in figure 5.

In figure 7 the comparison of the visualization in curvilinear representation (defined in section 10) of the resulting network state with or without normalization is shown more intuitively.

In the following sections, we will show how a particular property of biological neurons, rarely implemented, may improve these results.

12. Biological neurons are not tireless

Consider a network trying to self organize in a two-dimensional input space, with the dot product distance measure but without any normalization; one should observe that only the units with largest weight vector norms are moved (in the weight space). The reason for this is that, in the competition between all the units, the most frequent winners are those with a large weight vector norm (see the dot product expression, equation (5)).

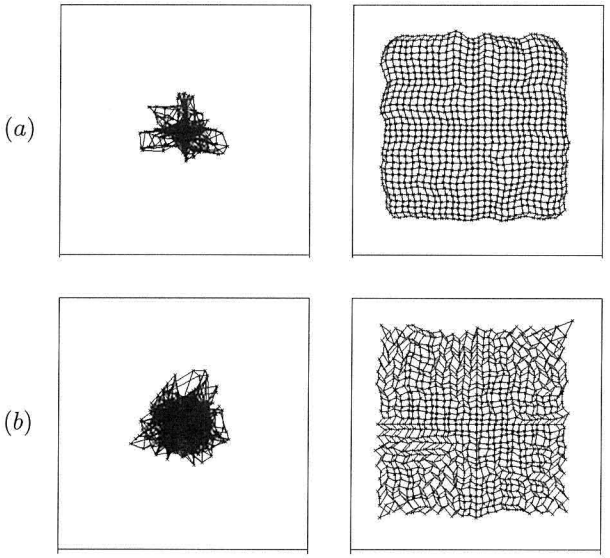


Figure 7: Curvilinear representation of networks after $k = 20{,}000$ iterations, (a) with normalization and (b) without normalization. Left side: projection on 2 axes. Right side: curvilinear representation.

In biological neurons, it seems plausible that the neurons *are not tireless*, that is, a particular neuron is not able to deliver a high level of activity very often. This fact is due to several phenomena, such as *accommodation* and *post-inhibitory rebound*. Because of these phenomena, the threshold required for firing tends to increase as the neuron is accumulatively stimulated at the subthreshold level ([11], pages 20–21, 263, 268). Such a property, called *conscience* by DeSieno [5], gives shorter convergence time and better organization in a network using the euclidean distance measure.

We have implemented this property in our simulations as part of the relaxation algorithm (we call *relaxation* that part of the auto-organization algorithm in which a winner is found in response to a given input vector). A variable (the *potential* p) representing the available amount of neurotransmitter is defined for each neuron. To be eligible, a unit must have a corresponding potential $p \geq p_{\min}$, where $p_{\min}$ represents a threshold of excitability. Then, the potential p of the winner unit is decremented by $p_{\min}$. For each iteration, the potential p of each neuron is incremented by $1/N$, where N is the total number of neurons. The potential p is truncated to 1. The minimum potential (or threshold) $p_{\min}$ is a parameter that varies between 0 and 1. If $p_{\min} = 0$, there is no change compared to the Kohonen algorithm because all units are eligible without restriction. If $p_{\min} = 1$, all the neurons are each elected in turn, without any consideration of the input vectors,

because when a unit is elected, it takes N iterations to recover the potential $p = 1$ and again be eligible. Simulations show that the optimal value for this parameter is about $p_{\min} = 0.75$.

With this accommodation property, the algorithm becomes (with the same notation as in equations (1) and (2)):
Find the winner unit index k:

$$\delta(\mathbf{x}, \mathbf{W}_k) = \min_{\substack{1 \le i \le N \\ p_i \ge p_{\min}}} \delta(\mathbf{x}, \mathbf{W}_i) \tag{24}$$

The weight adaptation remains:

$$\mathbf{W}_i(t+1) = \mathbf{W}_i(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{W}_i(t)] \Big|_{i \in V_k(t)} \tag{25}$$

The potential of every unit is modified according to:

$$p_i(t+1) = \begin{cases} p_i(t) + 1/N, & i \neq k \\ p_i(t) - p_{\min}, & i = k \end{cases} \tag{26}$$

When using accommodation, the comparison made in section 11 now gives quite better results.

13. Second results

Figure 8 shows the evolution of the disorder level $\Theta(n, k)$ (as defined in section 9) with $n = 200$ and k varying from 0 to 20,000. The solid line corresponds to a network with input and weight vector normalization, while the bold line corresponds to a network without any normalization, but using the accommodation algorithm as defined in section 12. All these networks use the dot product distance measure.

With accommodation, the organization quality of a network without any normalization is now comparable to the quality obtained with normalization, as shown in figure 8. To find the dimension of the input space that is large enough to get this result, we have made intensive simulations using varying dimensions. Figure 9 shows the disorder level $\Theta(n, k)$ (as defined in section 9) after $k = 20{,}000$ iterations and for n varying from 3 to 200 (for each curve, the simulations take about 10 hours of computing time on a network of 30 Sun™ SPARCstations). The dashed line corresponds to a network without any normalization, and the solid line corresponds to a network with input and weight vector normalization. The bold line corresponds to a network without any normalization, but using the accommodation algorithm as defined in section 12. All these networks use the dot product distance measure.

Also, considering the curvilinear representation defined in section 10, it seems that the organization obtained without any normalization is now quite good (figure 10; compare with figures 7*a* and 7*b*).

These experiments show that with an input dimension greater than or equal to 12 (with the experimental conditions previously mentioned), the quality of organization measured with our criterion becomes comparable with and without normalization. For large dimensions, the organization becomes even better without normalization, thanks to the accommodation algorithm.

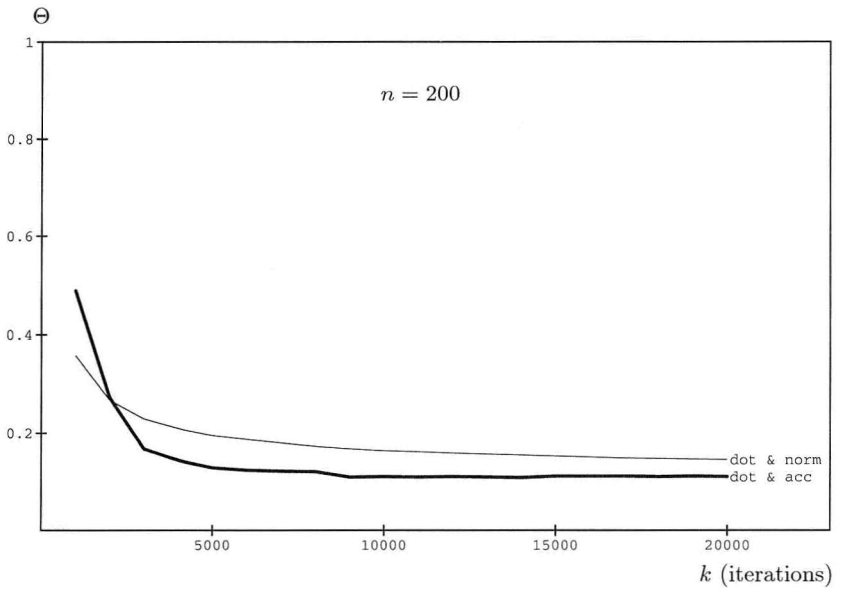


Figure 8: Evolution of $\Theta(n, k)$ with $n = 200$ and $k = 1$ to 20,000. Solid line: with input and weight normalization. Bold line: without any normalization, but using the accommodation property.

14. Conclusions and future work

In this paper we have discussed the necessity of the synaptic weight vector normalization. From a statistical point of view, the simulations show that this normalization is not necessary when the input space has a dimension greater than 12 and a uniform distribution of stimuli. The mathematical results obtained in sections 6 and 7 show that with a non-uniform distribution we should make the same observation. The critical point is that the number of independent input variables has to be large enough to ensure that the vector norms are comparable. This point has to be studied carefully if one wants to apply our results to real application data. For instance, if the degrees of freedom of even a high-dimensional input space are only 2 or 3, one cannot say that the input vector components are independent. Our future work will focus on simulations and on comparisons to known applications with real input data. On the other hand, we are now working on the definition of better criteria to analyze the quality of the organization in high-dimensional spaces. We are also studying the function Θ to determine if it is a Lyapunov function, in order to propose a demonstration of the convergence of the Kohonen algorithm in any dimension.

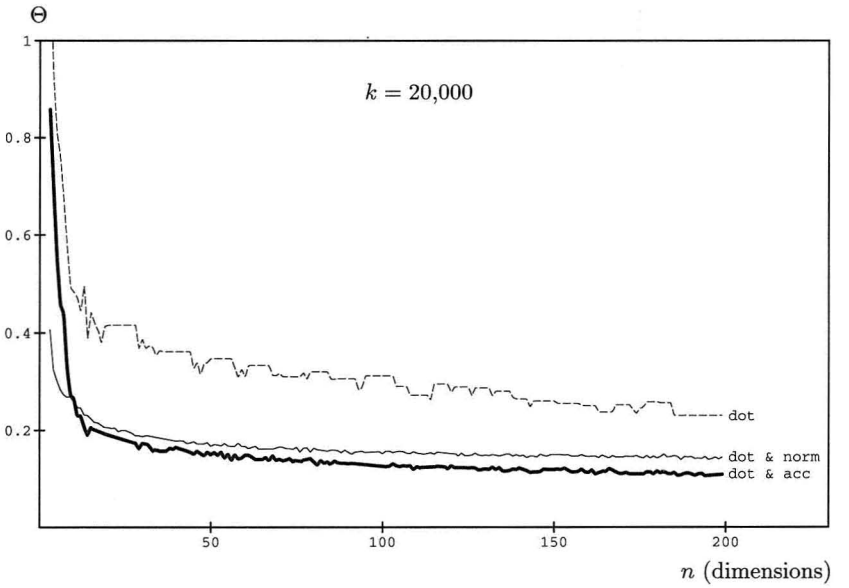


Figure 9: Evolution of $\Theta(n, k)$ with $k = 20{,}000$ and $n = 3$ to 200. Dashed line: without any normalization. Solid line: with input and weight normalization. Bold line: without any normalization but using the accommodation property.

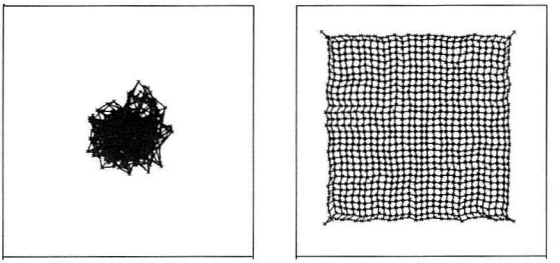


Figure 10: Curvilinear representation of networks after $k = 20{,}000$ iterations, with an input space of $n = 200$ dimensions and without normalization. Left side: projection on 2 axes. Right side: curvilinear representation.

Acknowledgments

We want to thank all the persons whose attentive lectures, constructive remarks, and fruitful suggestions helped us in writing this paper, and especially Prof. M. Cottrell (University of Paris I), Prof. J. C. Fort (University of Paris V), Prof. J. Bretagnolle (University of Paris XI), and Prof. C. Jutten (INPG Grenoble).

References

[1] F. Blayo, "Une Implantation Systolique des Algorithmes Connectionnistes" (PhD No. 904, LAMI EPF-Lausanne, 1990).

[2] M. Cottrell and J. C. Fort, "Etude d'un Processus d'Auto-Organisation," *Annales de l'Institut H. Poincaré*, **23**(1) (1987) 1–20.

[3] M. Cottrell and J. C. Fort, "Aspects Théoriques de l'Algorithme d'Auto-Organisation de Kohonen," *Annales du Groupe CARNAC*, LAMI EPF-Lausanne, **2** (1989) 73–83.

[4] P. Demartines and L. Tettoni, "SOMA: Simulateurs Logiciels de Modèles Neuronaux Adaptatifs," internal report, LAMI EPF-Lausanne (1991).

[5] D. DeSieno, "Adding a Conscience to Competitive Learning," *International Conference on Neural Networks*, **1** (1988) 117–124.

[6] C. Jutten, A. Guérin, and H. L. Nguyen Thi, "Adaptive Optimisation of Neural Algorithms," International Workshop on Artificial Neural Networks, Granada (1991).

[7] T. Kohonen, "Self-Organization of Topologically Correct Feature Maps," *Biological Cybernetics*, **43** (1982) 59–69.

[8] T. Kohonen, *Self-Organization and Associative Memory*, Second Edition (Berlin, Springer-Verlag,1988).

[9] S. W. Kuffler, J. G. Nicholls, and A. R. Martin, *From Neuron to Brain: A Cellular Approach to the Function of the Nervous System* (Sunderland, MA, Sinauer Associates, 1984).

[10] W. S. McCulloch and W. Pitts, "A Logical Calculus of the Ideas Immanent in the Nervous System," *Bulletin of Mathematical Biophysics*, **5** (1943) 115–133.

[11] R. J. MacGregor, *Neural and Brain Modelling* (London, Academic Press, 1987).

[12] W. H. Marshall and S. A. Talbot, "Recent Evidence for Neural Mechanism in Vision Leading to a General Theory of Sensory Acuity," pages 117–164 in *Visual Mechanism*, edited by H. Kluwer (Lancaster, PA, Cattell, 1942).

[13] E. Oja, "A Simplified Neuron Model as a Principal Component Analyzer," *Journal of Mathematical Biology*, **15** (1982) 267–273.

[14] E. Vittoz, "VLSI Implementation of Kohonen Maps," internal report in Esprit-BRA project NERVES No. 3049 (1990).