Applications of High-Resolution Self-Organizing Maps to Retrosynthetic and QSAR Analysis

Bruno Bienfait

Laboratoire de Chimie Organique, Université Catholique de Louvain, 1, Place Louis Pasteur, B-1348 Louvain-la-Neuve, Belgium

Received December 8, 1992*

Kohonen's self-organizing map (SOM) is a neural network model of the unsupervised class and, by some aspects, is analogous to other clustering algorithms. The high-resolution maps are characterized by a far higher number of neurons than the number of learning patterns to deal with. This paper describes explorations of two sets of organic compounds with this technique. The first set includes 32 carbonyl derivatives encoded by molecular structural subunits and classified according to four disconnection pathways. The self-organizing map is able to separate the 32 derivatives into 4 clusters without a prior knowledge of class membership. The results obtained are compared with those of a supervised multilayer perceptron. The second experiment tests the self-organization of 64 hallucinogenic phenylalkylamines encoded with a list of substituent descriptors (hydrophobicity, volume, and ^{13}C NMR relative aromatic shift). The network organizes the hallucinogenic compounds on the map according to their structural similarities; they are also globally positioned in relation to their biological activities.

INTRODUCTION

Artificial neural networks are computer developed algorithms which simulate some characteristics of the brain and allow their study. Their capacity to learn to associate and generalize makes them of interest in chemistry especially where the relation between cause and effect is not explicity known.[1-3] Over 40 different neural network models exist today. In spite of this great diversity, four common characteristics recur in each model: (1) a variable number of elementary processors (neurons) working in parallel to each other; (2) a connection topology describing the places of the neurons and their connections (synaptic weights); (3) a learning algorithm setting and modifying the connections between the neurons; (4) an exploitation phase during which the information acquired during learning is tested and exploited. The data are fed to the networks as fixed length vectors consisting of binary or continuous valued numbers. These feature patterns are normally divided into two sets depending on whether they are used during the learning phase or during the test or exploitation phase.

There are several ways of presenting a classification of artificial neural networks. Considering the flow of information as a first criterion, neural networks can be divided into two classes: the feedback models and the feedforward models.

In the feedback models, the synaptic weights are normally built in a noniterative operation, starting from the patterns of the learning set. During the exploitation phase, a new pattern is presented at the network input and the output of every neuron is calculated and then recycled as a new input. This process is repeated until the network converges toward a stable state, in which the outputs no longer change. Models belonging to this class are also characterized by the complete interconnectivity of the neurons and by the binary character of the input patterns. A typical example is the Hopfield network.[4] These models are used as associative memory and optimization problem solvers. Applications in chemistry concern protein tertiary structure prediction,[5,6] infrared spectroscopy,[7,8] and chromatography.[9]

The feedforward model category is mainly made up of two subclasses according to whether the learning algorithm is supervised or not. In the supervised case, the network learns to transform an input pattern into an output pattern. The synaptic weights are modified in order to minimize the error, i.e. the difference between the sought after output and that calculated by the network. Generally, many learning cycles (presentation of an input pattern, calculation of the outputs, and corrections of the weights) are necessary for the network to converge toward a solution. The advantage of this technique is its generalization and its predictability. When applying a new input pattern not previously included in the learning set, the network is capable, under certain conditions, of calculating a new valid output. The most representative model of this class is the multilayer perceptron, in which the neurons are placed in several successive layers.[10] This network is also called the "back propagation" model due to how the error is fed back through the network. Over 40 articles describe the applications of this model in chemistry and this will not be discussed further here.[1-3]

When the learning algorithm is unsupervised, or in other words, when the patterns are applied without specifying the desired output, the network must build its own internal representation based on the similarities among the different input patterns. Unsupervised learning is also called competitive learning because the neurons compete to elect and adapt one of them, which is closest to the input pattern being treated. The model from this class nearest to biological systems is probably Kohonen's self-organizing map (SOM). Biological maps are spatial arrangements of neurons that can be located in the nervous system at the visual, auditory, or somatosensory cortex.[11] An essential characteristic of both natural and simulated maps is their ability to project high dimensional data in a two-dimensional topological representation, thereby preserving the most significant information. Kohonen's algorithm achieves this goal by the association of two processes: (1) identification and stimulation on the map of the neuron which is the most sensitive to the current input; (2) spreading of its activity among other spatially close cells.

The SOM has cluster analysis characteristics similar to those of more classical algorithms such as k-means[11,12] or nonlinear mapping.[13,14]

In spite of its unsupervised character, Kohonen's algorithm can be improved and optimized by a supervised method when the number of sought after classes is known. This procedure is called "learning vector quantization" (LVQ) and has been successfully applied to speech recognition (phoneme classi-

* Abstract published in *Advance ACS Abstracts*, March 1, 1994.

0095-2338/94/1634-0890$04.50/0

fication).[11] It performs better than other more conventional methods, such as *k*-nearest neighbor and Bayesian classifiers.

As opposed to the multilayer perceptron, the SOM has little been used in chemistry. Five research groups have investigated it: two in analytical chemistry,[15,16] two in molecular biology,[17-20] and one in medicinal chemistry.[21,22] Gasteiger *et al.* have studied how structure is related to mass and IR spectral data.[15] The clustering of a large database of IR spectra has been investigated by Melssen *et al.*[16] Ferràn and Ferrara have clustered proteins according to their sequence homologies and shown how the learning phase of a SOM roughly mimics the scenario of biological evolution.[17-19] Arrigo *et al.* have developed a program capable of revealing unusual areas among the sequence data of nucleic acids.[20] Rose *et al.* carried out a quantitative structure–activity relationship (QSAR) analysis on a series of 31 antimycin analogues containing structural outliers.[21,22] Their results showed that the SOM compared favorably with principal component analysis, nonlinear mapping, and hierarchical cluster analysis.[21]

The aim of this work is to explore the possibilities of high-resolution SOMs, which are characterized by a higher number of neurons than the number of patterns belonging to the training set. To test this concept, two different problems are dealt with: (1) cluster analysis of 32 carbonyl compounds encoded by a sequence of molecular subunits and selected in relation with four disconnection pathways; (2) a structure–activity analysis of 64 hallucinogenic phenylalkylamines represented by substituent descriptors. Since the applications treated in this paper are based on Kohonen's SOM, we will first briefly describe the basis of this algorithm.

METHODS AND SYSTEM

A vector quantizer is a classical method used to reduce and approximate a large set of vectors into a normally much smaller set of representative vectors. The SOM is a vector quantization algorithm, which also orders spatially the representative vectors on a map, mostly two-dimensional. This ordering process tries to preserve at best the high dimensional topological relations among the original vectors. The mathematical formulations of these two concepts, (1) competitive learning (which leads to vector quantization) and (2) self-organization (which is the result of spatial ordering), will be described successively.

Competitive Learning. As mentioned above, two different sets of vectors are considered. The first set consists of p learning patterns which are data vectors X made up of m real values x_i. These vectors will be used as input and will not be modified during learning. Using neural network terminology, the second set of vectors consists of n synaptic weight vectors W_j made up of m real values w_{ij}. Each weight vector W_j is associated to a neuron j placed on a map (Figure 1). For the time being, the relative positions of the n neurons on the map are ignored. Initially, all w_{ij} are different random numbers. A vector X of the learning set is presented at the network input. The Euclidean distances d_j between X and each vector W_j are calculated as

$$d_j = \left(\sum_{i=1}^{m}(x_i - w_{ij})^2\right)^{1/2} \qquad (1)$$

The neuron with the weight vector W_j closest to the input pattern X has the smallest distance d_j. This j^* neuron is said to be the winner of the competition among all neurons. It is updated, in order to bring its synaptic vector W_{j^*} even closer

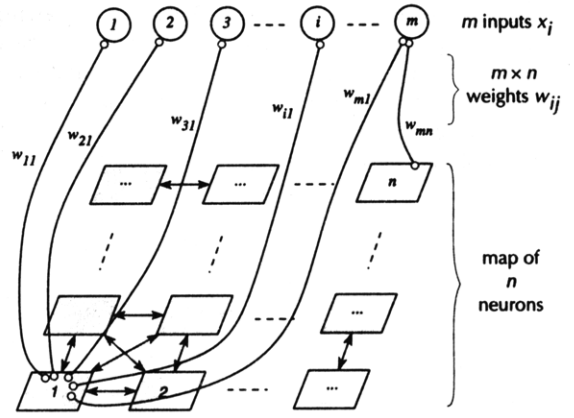


Figure 1. Schematic representation of a self-organizing map. For clarity, only a few weights are shown.

to the current input vector X

$$w_{ij^*}(t+1) = w_{ij^*}(t) + \alpha(t)\,[x_i - w_{ij^*}(t)]$$

$$0 < \alpha(t) < 1 \qquad (2)$$

Successive discrete instants of time are symbolized by t and $t+1$. After each updating, the time variable t is incremented, whereas α, the learning speed, is decreased. The whole process is called a step. It is repeated for each pattern of the learning set. A learning epoch is made up of p steps, where p is the size of the learning set. This learning procedure is iterative and a large number of steps (or epochs) are normally required for the weight vectors to converge toward a good approximation of the more numerous input vectors.

Self-Organization. The iterative process described above constitutes a competitive learning process leading to the quantization of vectors without any topological organization. The map neurons are indeed updated in an isolated fashion regardless of their relative position on the map. In biological maps, the most stimulated neuron diffuses its activation toward its neighboring cells via lateral axons. Kohonen modeled this effect by defining a neighboring set N_{j^*} as a set of neurons topologically close to the winning cell j^*. The learning algorithm is modified so that the j^* cell is updated, as well as all the cells belonging to the neighborhood N_{j^*} (eq 3)

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t)\gamma(t)\,[x_i - w_{ij}(t)]$$

$$\gamma(t) = 1 \qquad \forall j \in N_{j^*}(t)$$

$$\gamma(t) = 0 \qquad \forall j \notin N_{j^*}(t) \qquad (3)$$

A set of neighboring neurons may be defined for instance, as the cell subset belonging to a circular area centered around the winning cell j^*. At first, this area is wide. It may include over half of the map neurons. Like the learning speed α, its radius decreases as the time variable t is incremented. The $\gamma(t)$ parameter may be defined more generally by introducing a Gaussian function depending on the relative distance r_{jj^*} which is the distance between the winning cell j^* and the cell j being updated (eq 4)

$$\gamma(t) = e^{-(1/2)(r_{jj^*}/\sigma(t))^2} \qquad (4)$$

The neighborhood radius, $\sigma(t)$, is also a function decreasing with time.

Once the learning phase is over, labeling must be carried out in order to determine which cells correspond to which

kind of input pattern. Labeling is done by presenting an input pattern, calculating the Euclidian distances, finding the closest weight vector and designating with a label its associated neuron. Visual inspection of the labeled map allows the recognition of areas, where groups of similar input patterns were gathered.

High-Resolution Self-Organizing Maps. Most of the SOM applications are connected with vector quantization problems. As a result, the number of patterns to deal with is much higher than the number of weight vectors. For a high-resolution SOM, this situation is reversed and leads to a better visualization of the intermediate and final labeled maps. The inconvenience, however, is a proportionally longer calculation time.

System. The program used to simulate Kohonen's neural network model is the "SOM_PAK" 1.2, available through the Internet and originally written in ANSI C.[23] This program has been recompiled on a NeXT workstation (~3 Mflops) and modified to monitor learning evolution and to simulate toroidal, edgeless maps (cf. Appendix). Data treatment and results visualization were carried out with the Mathematica program.[24]

Learning Parameters. The values of the learning parameters, such as the speed α, the number of epochs, and the neighborhood radius σ, were chosen empirically starting from the base values of the examples given by SOM_PAK. Monitoring the learning evolution is useful in order to adjust the learning parameters. In SOM_PAK, decreases of the neighborhood radius $\sigma(t)$ and of the speed $\alpha(t)$ are linear. Learning is done in two phases; global spatial ordering occurs first while the second period is used for local and fine adjustments.

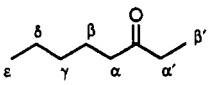
APPLICATIONS

1. Clustering of Carbonyl Compounds. This first experiment uses a data set published by Luce and Govind.[25] These authors trained a multilayer perceptron, a neural network of the supervised class, to recognize the disconnection of a molecule encoded by a list of structural subunits. Their network was conceived to be incorporated in a hybrid expert system of retrosynthetic analysis, in order to obtain more flexibility than traditional rule-based approaches.

Generation of the Input Patterns. A series of 32 carbonyl compounds is shown in Table 1 according to 4 disconnections in α and β of carbonyl group: the reversed aldol, Claisen, Michael, and enamine condensations. Each molecule is encoded according to an ordered list of eight parameters, selected from the entries in Table 2. Each of these parameters represents an on-path atom and its off-path substituents and has been chosen according to their position relative to the carbonyl carbon of the generalized structure shown in Table 1.[25] For instance, the compound corresponding to the first entry of Table 1 has no atom in ϵ and δ (index 0 in Table 2), a CH_3 (index 5) in γ, a CHOH (index 18) in β, a CH_2 (index 4) in α, a hydrogen (index 1) in α', and no atom in the β' position. The entries of Table 1 constitute a set of input patterns suitable to be treated by a neural network. As opposed to Luce and Govind's neural network, a SOM needs no information about the disconnection class membership.

Results and Discussions. The SOM used to cluster these carbonyl compounds is a two-dimensional 9 × 9 map which has about two and half times more neurons than the number of carbonyl compound patterns. The learning evolution is shown in Figure 2. The integer above each map indicates the number of learning epochs accomplished, and the labels a, c, m, and e show the position of minimum Euclidian distances (eq 1) corresponding to the disconnection class (aldol, Claisen, Michael, or enamine) the respective input patterns belong to. The first map of the first row shows the label positions before learning. Only six different labels are visible because of superpositions. As the weight values w_{ij} are initialized with random numbers, the labels are placed randomly on the map. The second map shows the result after one learning epoch or, in other words, 32 steps, as there are 32 patterns in the learning set. We can see three clusters, which tend to move away from

Table 1. Learning Set Input Vectors[25]



	index	C_ϵ	C_δ	C_γ	C_β	C_α	$C_{\alpha'}$	$C_{\beta'}$
Aldol-type disconnection	1	0	0	0.1	0.36	0.08	0.02	0
	2	0	0	0.1	0.38	0.08	0.1	0
	3	0	0.1	0.08	0.36	0.06	0.02	0
	4	0	0.1	0.08	0.38	0.06	0.1	0
Claisen-type disconnection	5	0	0	0.02	0.3	0.08	0.18	0.1
	6	0	0	0.02	0.3	0.08	0.18	0.08
	7	0	0	0.1	0.3	0.06	0.18	0.1
	8	0	0	0.1	0.3	0.06	0.18	0.08
Michael-type disconnection	9	0.02	0.3	0.9	0.08	0.08	0.02	0
	10	0.02	0.3	0.9	0.06	0.08	0.02	0
	11	0.02	0.3	0.9	0.08	0.06	0.02	0
	12	0.02	0.3	0.9	0.04	0.08	0.02	0
	13	0.02	0.3	0.9	0.06	0.06	0.02	0
	14	0.02	0.3	0.9	0.04	0.06	0.02	0
enamine-type disconnection	15	0.02	0.3	0.08	0.08	0.08	0.02	0
	16	0.02	0.3	0.08	0.08	0.06	0.02	0
	17	0.02	0.3	0.08	0.06	0.08	0.02	0
	18	0.02	0.3	0.06	0.08	0.08	0.02	0
	19	0.02	0.3	0.08	0.06	0.06	0.02	0
	20	0.02	0.3	0.08	0.04	0.08	0.02	0
	21	0.02	0.3	0.06	0.06	0.08	0.02	0
	22	0.02	0.3	0.04	0.08	0.08	0.02	0
	23	0.02	0.3	0.06	0.08	0.06	0.02	0
	24	0.02	0.3	0.06	0.04	0.08	0.02	0
	25	0.02	0.3	0.06	0.06	0.06	0.02	0
	26	0.02	0.3	0.04	0.06	0.08	0.02	0
	27	0.02	0.3	0.08	0.04	0.06	0.02	0
	28	0.02	0.3	0.04	0.08	0.06	0.02	0
	29	0.02	0.3	0.06	0.04	0.06	0.02	0
	30	0.02	0.3	0.04	0.06	0.06	0.02	0
	31	0.02	0.3	0.04	0.04	0.08	0.02	0
	32	0.02	0.3	0.04	0.04	0.06	0.02	0

Table 2. Numerical Values for Molecular Subunits in Linked-List Representation[25]

index	feature	value	index	feature	value
0	no atom	0	25	H–(C═)–OR	0.5
1	–H	0.02	26	R–(C═)–OR	0.52
2	R–(CR_2)–R	0.04	27	R–(C═N)–R	0.54
3	R–(CHR)–R	0.06	28	H–(C═N)–R	0.56
4	R–(CH_2)–R	0.08	29	R–(CH_2)–NH_2	0.58
5	R–(CH_2)–H	0.1	30	R–(CHR)–NH_2	0.6
6	R–(C═)–R	0.12	31	R–(CR_2)–NH_2	0.62
7	R–(C═)–H	0.14	32	H–C(═)–NH_2	0.64
8	H–(C═)–H	0.16	33	R–(C═)–NH_2	0.66
9	R–O–R	0.18	34	R–(CH_2)–NHR	0.68
10	R–(NR)–R	0.2	35	R–(CHR)–NHR	0.7
11	R–(+N═)–R	0.22	36	R–(CR_2)–NHR	0.72
12	R–S–R	0.24	37	H–(C═)–NHR	0.74
13	R–F	0.26	38	R–(C═)–NHR	0.76
14	R–Cl,R–Br,R–I	0.28	39	R–(CH_2)–NR_2	0.78
15	R–(C═O)–R	0.3	40	R–(CHR)–NR_2	0.8
16	R–(C═O)–H	0.32	41	R–(CR_2)–NR_2	0.82
17	R–(CH_2)–OH	0.34	42	H–(C═)–NR_2	0.84
18	R–(CHR)–OH	0.36	43	R–(C═)–NR_2	0.86
19	R–(CR_2)–OH	0.38	44	R–(C(C═O)$_2$)–R	0.88
20	H–(C═)–OH	0.4	45	R–(C(C═O)$_2$)–H	0.9
21	R–(C═)–OH	0.42	46	(O═C)–(C═)–(C═O)	0.92
22	R–(CH_2)–OR	0.44	47	R–C≡N	0.94
23	R–(CHR)–OR	0.46	48	R–(CR_2)–Ph	0.96

Table 3. Learning Parameters for the Two Learning Phases of the Clustering of Carbonyl Compounds

learning phase	map dimensions	learning epochs	$\alpha(t)$ start	$\alpha(t)$ end	$\sigma(t)$ start	$\sigma(t)$ end	calculation time, s
1	9 × 9	3	0.15	0.0	7.0	1.0	7
2		20	0.1	0.0	1.0	1.0	

Table 4. Simulation Parameters for the Learning Phase of the Clustering of Carbonyl Compounds with the Modified Encoding Scheme

learning phase	map dimensions	learning epochs	$\alpha(t)$ start	$\alpha(t)$ end	$\sigma(t)$ start	$\sigma(t)$ end	calculation time, s
1	9 × 9	6	0.15	0.0	7.0	1.0	3.5

each other toward the borders and corners of the map. The aldol and Claisen compounds remain gathered in the same cluster. The maps of the first row illustrate the first learning phase. The learning parameters are collected in Table 3. The second row shows the evolution of the second phase, which includes 20 epochs. At the end of learning, the map is clearly divided into four distinct areas. The largest enamine class occupies the left part of the map, whereas the three others divide the right part into three portions.

While the above example uses an unsupervised algorithm, the maps shown in Figure 2 are also useful for a supervised application. Certain characteristics of Luce and Govind's method of encoding molecules can be highlighted by an analysis of the learning evolution. That is, the compounds labeled by the letters a and c remain in one cluster during the first learning phase, whereas the two other groups already move away from each other after the first epoch. This phenomenon can be explained by examining the numeric values of the β position column of Table 1. The Aldol disconnection requires a hydroxyl group in β, whereas the Claisen disconnection necessitates a carbonyl substituent. This important difference in functionality is not reflected by the relatively small interval between the two numeric parameters in Table 2, 0.3 and a mean of 0.37 respectively. If all the occurrences of 0.3 are replaced by –0.3 for all the input patterns (Table 1), the separation in four different clusters occurs after two epochs only (Figure 3). This phenomenon can be reproduced and is independent of the random values of the initial weights. The choice of –0.3 is arbitrary and disrupts partially the order and the scale of the original encoding parameters (Table 2). The analysis of the weights of the four neurons in each corner of the final map of Figure 3 shows that they are close to the input pattern means of the class they respectively belong to (Table

Table 5. Comparison between the Input Pattern Means (Modified Encoding Scheme) and Selected Weight Vectors on the Last Map of Figure 3

		C_ϵ	C_δ	C_γ	C_β	C_α	$C_{\alpha'}$	$C_{\beta'}$
aldol	mean	0.07	0.37	0.09	0.05	0.0	0.06	0.0
	weights at {1,9}	0.07	0.23	0.17	–0.09	0.01	0.05	0.0
Claisen	mean	0.07	–0.3	0.06	0.0	0.0	0.18	0.09
	weights at {9,1}	0.07	–0.15	0.14	–0.12	0.01	0.12	0.05
Michael	mean	0.07	0.06	0.9	–0.3	0.02	0.02	0.0
	weights at {1,1}	0.07	0.05	0.73	–0.26	0.02	0.03	0.01
enamine	mean	0.07	0.06	0.06	–0.3	0.02	0.02	0.0
	weights at {9,9}	0.07	0.06	0.06	–0.29	0.02	0.02	0.0

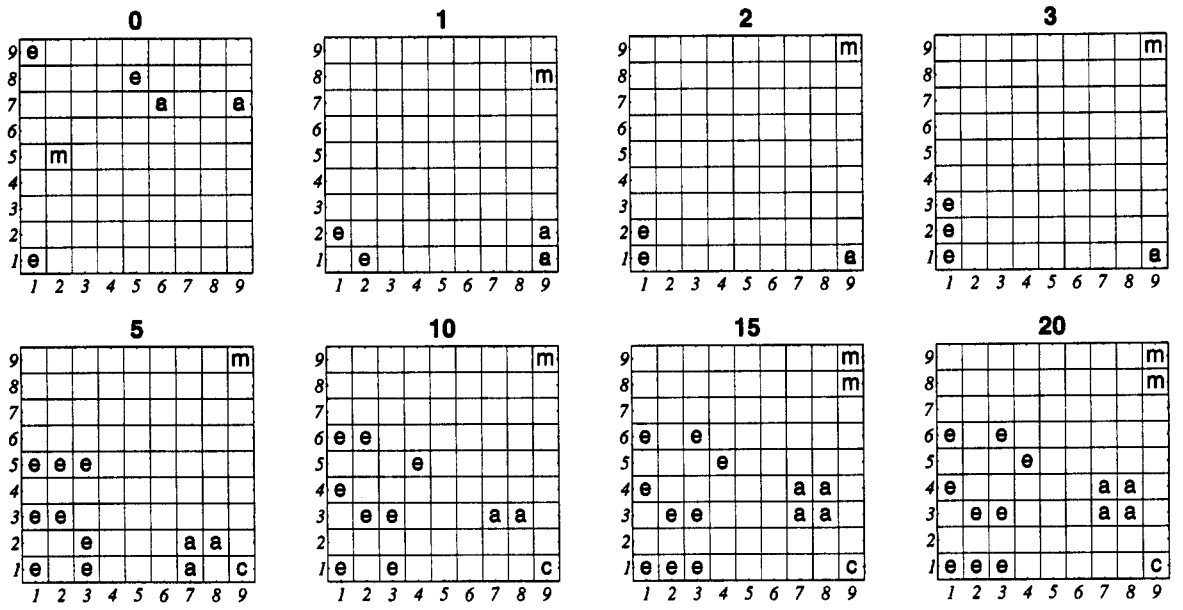


Figure 2. Evolution of the map during learning. The labels a, c, m, and e reveal the position of minimum Euclidian distance for each input pattern according to the disconnection class respectively: aldol, Claisen, Michael, and enamine. The integers above each map indicate the number of learning epochs.

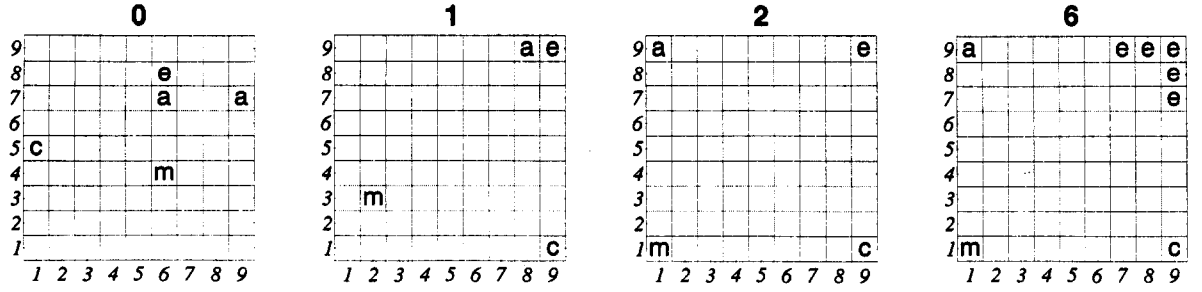
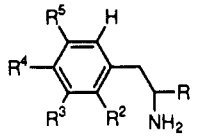


Figure 3. Evolution of the map during learning after modification of the input patterns.

Table 6. Phenylalkylamines Hallucinogens: Identity, Activity (MU = mescaline units),[a] Map Position, and Minimal Euclidian Distance



index	R	R2	R3	R4	R5	MU	position	Eucl dist
1	H	H	H	MeO	H	<1	7, 28	0.83
2	H	H	MeO	MeO	H	<0.2	4, 31	0.54
3	H	H	MeO	MeO	MeO	1	33, 3	0.42
4	H	MeO	MeO	MeO	H	<1	13, 32	0.86
5	H	MeO	H	MeO	MeO	<1	13, 28	0.72
6	H	H	EtO	MeO	MeO	1	32, 30	0.19
7	H	H	MeO	EtO	MeO	7	31, 6	0.33
8	H	H	MeO	PrO	MeO	6	29, 9	0.26
9	H	H	MeO	BuO	MeO	2	27, 10	0.47
10	H	H	EtO	EtO	MeO	1.5	29, 32	0.33
11	H	H	EtO	MeO	EtO	<1	33, 24	0.41
12	H	H	EtO	EtO	EtO	<1	28, 20	0.55
13	H	H	PrO	MeO	MeO	<1	29, 28	0.68
14	H	H	MeO	–OCH$_2$O–		1	1, 1	0.61
15	H	H	–OCH$_2$O–		H	1	6, 31	0.53
16	H	MeO	H	Et	MeO	18	19, 26	0.22
17	H	MeO	H	Me	MeO	20	14, 25	0.31
18	H	H	MeO	MeO	MeS	6	33, 13	0.41
19	H	H	MeO	MeS	MeO	12	29, 4	0.46
20	H	H	MeO	MeO	EtS	6	32, 16	1.28
21	H	H	EtO	MeS	MeO	4	24, 32	0.36
22	H	H	EtO	MeO	MeS	0.5	1, 20	0.38
23	H	H	MeO	EtO	MeS	6	30, 13	0.28
24	H	H	MeO	EtS	MeO	20	26, 5	0.28
25	H	H	EtS	EtO	MeO	2	26, 28	0.73
26	H	H	EtO	EtS	MeO	4	23, 1	0.40
27	H	H	EtS	MeO	EtO	<1	26, 24	0.67
28	H	H	EtO	MeS	EtO	<1	26, 19	0.46
29	H	H	EtO	EtO	MeS	2	31, 20	0.53
30	H	H	EtS	EtO	EtO	<1	26, 24	0.57
31	H	H	EtO	EtS	EtO	<1	25, 18	0.61
32	H	H	MeO	PrS	MeO	16	24, 6	0.15
33	H	H	MeO	BuS	MeO	3	23, 8	0.62
34	Me	H	H	MeO	H	5	9, 4	1.22
35	Me	MeO	H	H	MeO	8	9, 19	0.52
36	Me	MeO	H	MeO	H	5	13, 7	0.89
37	Me	H	MeO	MeO	H	0.5	6, 4	0.69
38	Me	H	MeO	MeO	MeO	2	2, 8	0.66
39	Me	MeO	H	MeO	MeO	20	12, 12	0.48
40	Me	MeO	MeO	MeO	H	2	14, 3	0.65
41	Me	MeO	MeO	H	MeO	4	6, 17	0.79
42	Me	MeO	H	EtO	MeO	20	15, 11	0.26
43	Me	MeO	H	PrO	MeO	20	17, 11	0.41
44	Me	H	Me	BzlO	MeO	2	23, 12	1.39
45	Me	MeO	MeO	MeO	MeO	6	5, 13	0.57
46	Me	H	–OCH$_2$O–		H	3	7, 4	0.67
47	Me	–OCH$_2$O–		MeO	H	3	16, 5	1.19
48	Me	H	MeO	–OCH$_2$O–		2.7	4, 7	0.63
49	Me	MeO	H	–OCH$_2$O–		10	11, 10	0.65
50	Me	MeO	–OCH$_2$O–		H	10	13, 4	0.61
51	Me	MeO	–OCH$_2$O–		MeO	12	7, 13	0.61
52	Me	MeO	MeO	–OCH$_2$O–		5	7, 11	0.74
53	Me	MeO	H	Me	MeO	80	11, 16	0.32
54	Me	MeO	H	Et	MeO	100	14, 18	0.12
55	Me	MeO	H	Pr	MeO	80	16, 17	0.20
56	Me	MeO	H	Bu	MeO	40	19, 16	0.17
57	Me	MeO	H	iBu	MeO	20	18, 16	0.07
58	Me	MeO	H	Pent	MeO	10	20, 15	0.81
59	Me	MeO	H	MeS	MeO	40	13, 15	0.19
60	Me	MeO	H	iPrS	MeO	40	17, 15	0.43
61	Me	MeO	H	Br	MeO	400	13, 19	0.13
62	H	MeO	H	Br	MeO	35	17, 25	0.19
63	H	MeO	H	I	MeO	44	19, 24	0.34
64	Me	MeO	H	NO_2	MeO	70	9, 21	1.14

[a] From ref 29 except compounds 61, 62, and 63 from ref 30.

5). In this case, a map composed of four weights vectors built in one operation from the respective mean vector of each class

Table 7. Hydrophobicity,[28,29] van der Waals Volume,[29] and ^{13}C Delta Shift[32]

symbol	hydrophobicity	volume, Å^3	^{13}C shift
H	0.0	7.2[a]	0.0
Me	0.6	18.6	–3.0
Et	1.1	37.3	–2.7
Pr	1.6	56.0	–2.7
iPr	1.4	56.0	–2.8
Bu	2.1	74.8	–2.8
iBu	1.9	74.8	–2.95[b]
Pent	2.6	93.5	–2.8[b]
MeO	–0.02	28.7	–7.7
EtO	0.5	47.5	–7.7[b]
PrO	1.0	66.2	–7.9[b]
BuO	1.5	84.9	–7.9
OCH_2O[c]	–0.04	13.8	–7.7[b]
BzlO	2.7	112.6	–7.7[b]
MeS	0.6	39.9	–3.6
EtS	1.1	58.6	–2.6
PrS	1.6	77.3	–2.6[b]
iPrS	1.4	77.3	–2.6[b]
BuS	2.1	96.1	–2.6[b]
NO_2	–0.3	17.7	6.1
F	0.1	10.3[a]	–4.4
Cl	0.7	24.4[a]	–1.9
Br	0.9	31.1[a]	–1.6
I	1.1	41.6[a]	–1.1

[a] Calculated from radius of ref 31. [b] Estimated from similar group. [c] To each attached ring atom.

Table 8. Simulation Parameters for the Two Learning Phases of the Mapping of Hallucinogenic Phenylalkylamines

learning phase	map dimensions	learning cycles	$\alpha(t)$ start	$\alpha(t)$ end	$\sigma(t)$ start	$\sigma(t)$ end	calculation time, min
1	33 × 33	20	0.1	0.0	10.0	1.0	7
2	(no borders)	60	0.05	0.0	3.0	1.0	

could advantageously replace a supervised neural network. Such a system works also with the nonmodified set of learning vectors. Luce and Govind's supervised multilayer perceptron required 550 epochs to reach a correct classification. A multilayer perceptron is a powerful classifier which enables the association of several different clusters into one class.[26] Since their data are not of high complexity, a perceptron without hidden layers is sufficient to obtain the same result.[27]

2. Mapping of Hallucinogenic Phenylalkylamines. A set of 64 hallucinogenic phenylalkylamines was collected from the literature (Table 6). Biological activity is measured in mescaline units (MU), which is the relative dose of mescaline necessary to produce the same hallucinogenic effect as the considered phenylalkylamine dose. To avoid complications due to the symmetry of the aromatic nuclei,[29] three compounds of low activity (~10 MU), with substituents other than H at position 6 of the aromatic ring, are not included in this study.

Generation of the Input Patterns. Molecules are encoded using three descriptors for each variable substituent from the aromatic ring and from the alkylamine chain: hydrophobicity, volume, and an electronic parameter (Table 7). The electronic parameter corresponds to substituent perturbations in monosubstituted benzenes observed at C4 in ^{13}C NMR.[32]

Building a phenylalkylamine input vector consists of substituting every occurrence of a symbol like Me or H, by its three substituent descriptors selected from the corresponding entry in Table 7. For instance, mescaline or (3,4,5-trimethoxyphenyl)ethylamine has the fragment symbol list {H, H, MeO, MeO, MeO} (index 3 of Table 6) and is encoded as the {0.0, 7.2, 0.0, 0.0, 7.2, 0.0, –0.02, 28.7, –7.7, –0.02, 28.7, –7.7, –0.02, 28.7, –7.7} vector of size 15. If all 64 vectors built from the molecules of Table 6 are piled up horizontally to

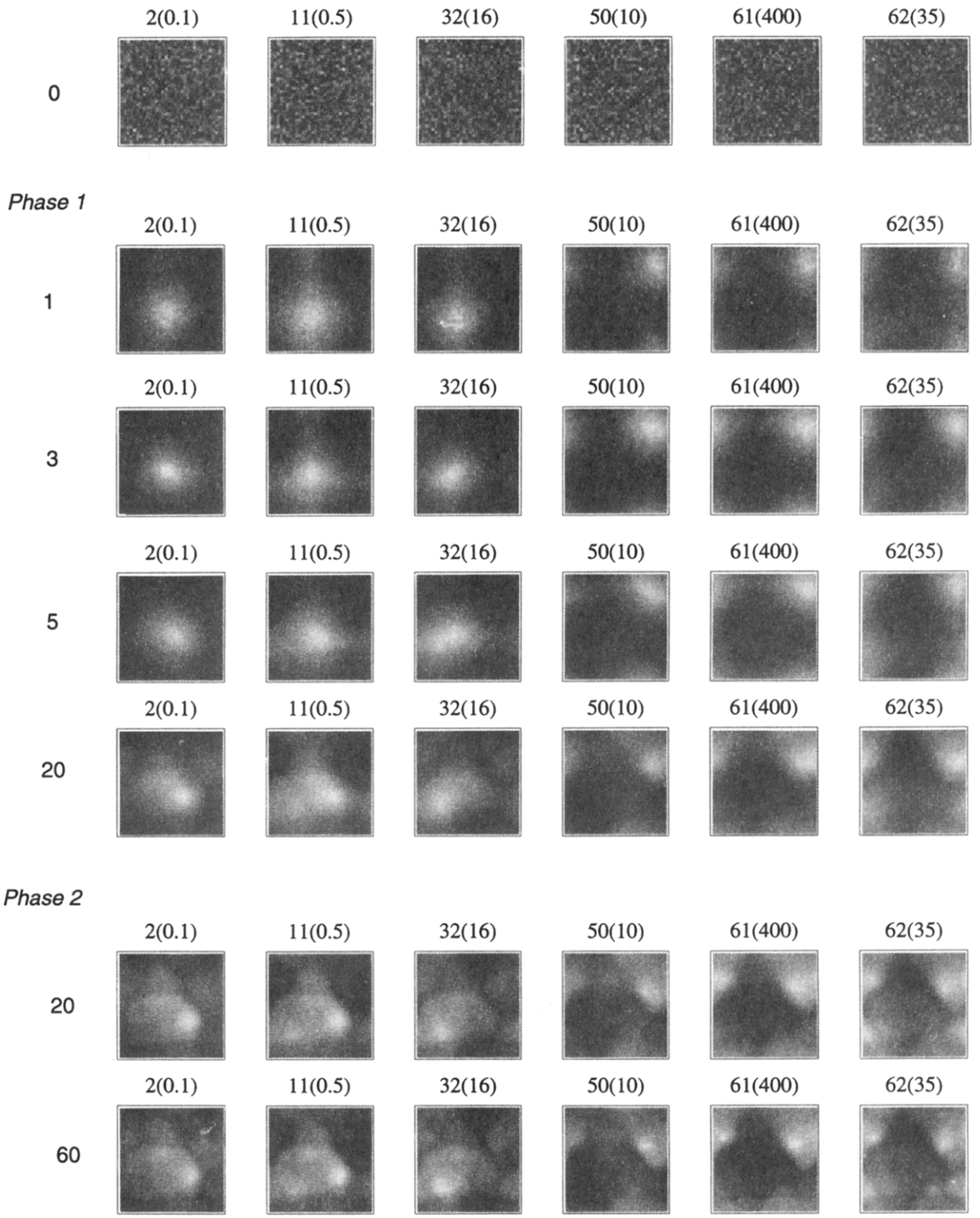


Figure 4. Evolution of distance maps of compounds 2, 11, 32, 50, 61, and 62. The labels on the left side indicate the number of learning epochs.

form a 15 × 64 matrix, it can easily be seen that the 15 columns of this matrix have very different scale values. In order to give the same importance to each column, a preprocessing is necessary. Therefore, autoscaling is applied so that all of them have the same mean and the same variance.[14]

Results and Discussion. The map used for this experiment is composed of 33 × 33 neurons, which is more than 17 times the number of phenylalkylamines. With this data set, it is necessary to use a toroidal map (where the indices run cyclically and so simulate the effect of an infinite edgeless map), otherwise large areas inside the final calculated map will be unused.

As for the previous example, learning takes place in two phases. The learning parameters are shown in Table 3 while the learning evolution is displayed in Figure 4. Each row shows the arrays of Euclidian distances d_j (eq 1) calculated for compounds 2, 11, 32, 50, 61, and 62. The minimum distance point is colored in black, the maximum distance point in white, and intermediary values in gray levels. Before learning, the weights W_j are initialized with random numbers; these are the first row maps. The second row displays the result after only one learning epoch; a global organization already has taken shape. The five following rows show the results after 3, 5, and 20 epochs (first phase) followed by

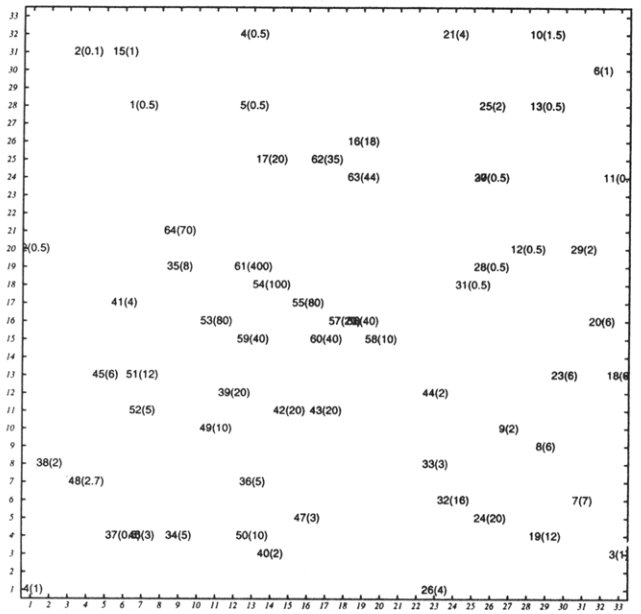


Figure 5. Final positions of the 64 phenylalkylamines.

another 20 and 60 epochs (second phase). During these learning cycles, the dark areas move and shrink progressively while the neighborhood radius decreases.

The final positions of the minimal Euclidian distance for each phenylalkylamine are shown in Table 6 and in Figure 5. It appears clearly that the map is globally organized according to hallucinogenic activity. Resuming learning with different sets of random weights gives consistently very similar maps when looking at relative positions of the labels. The map on Figure 5 shows however numerous local imperfections, essentially because of the very wide range of activity. Small structural changes have in some cases a very strong effect on the activity. This is illustrated by the phenylalkylamine pairs 35(8) and 53(80). They fall near one another as expected because of their molecular similarity (the only change is the replacement of a hydrogen by a methyl group in para) but exhibit an activity ratio of 10.

With distance maps such as those displayed in the last row of Figure 4, it is possible to build an average Euclidian distance map (ρ_j) weighted by the hallucinogenic activity a_k (MU) from each phenylalkylamine k (eq 5).

$$\rho_j = \sum_{k=1}^{64} a_k d_{jk} \qquad (5)$$

The Euclidian distance between the autoscaled input vector X_k of the molecule k and the weights vector W_j of the cell j is defined as d_{jk}. Figures 6 and 7 show respectively a tridimensional view and a contour plot of the ρ_j values thus obtained. In the first case, the distance values ρ_j are negative in order to turn the map, thereby visualizing easily the highest activity zone. The contour plot of Figure 7 is useful as a visual tool to appreciate the activity of new compounds. In principle, a set of test molecules is not necessary as the values of hallucinogenic activity have been excluded from learning. Table 9 shows the results for five new compounds. The first entry (65) is very similar to its bromine equivalent (61). It falls on a cell close to the two most active phenylalkylamines; therefore a strong activity is predicted. The last entry (69) differs considerably from all the previous molecules because of the presence of an ethyl group on the amine chain. Its minimal Euclidian distance (3.89) is much larger than 0.53,

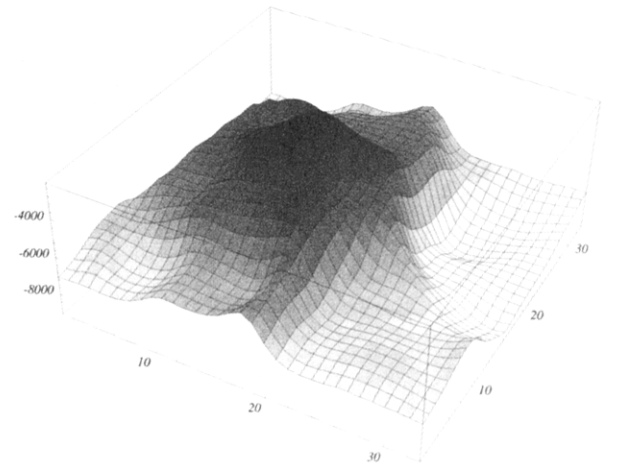


Figure 6. Three-dimensional view of the weighted activity Euclidian distances calculated from (eq 5).

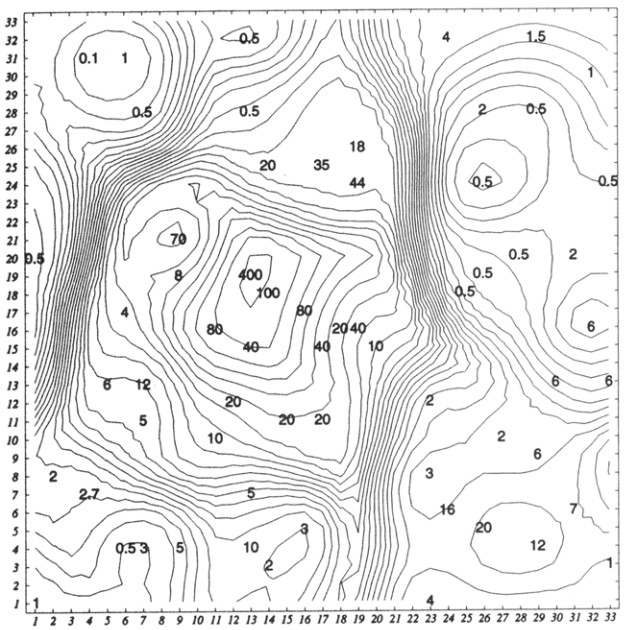


Figure 7. Superposition of the activity values and the contour plot of the three-dimensional view of Figure 6.

Table 9. Test of Compounds Not Included in the Learning Set

index	R	R2	R3	R4	R5	MU	position	Eucl dist
65	Me	MeO	H	Cl	MeO		12, 18	0.20
66	H	MeO	–OCH$_2$O–		H	<5[a]	13, 32	1.19
67	Me	MeO	H	MeO	EtO	<7[a]	12, 13	2.79
68	H	H	H	H	H	0[b]	7, 26	2.70
69	Et	H	MeO	MeO	MeO	<2[a]	4, 8	3.89

[a] Reference 30. [b] Reference 29.

the mean of all compounds in the learning set (Table 6). This compound falls on a cell close to its methyl-substituted equivalent (38) and luckily has the same activity.

Among the molecules belonging to the learning set, the nitro derivative (64) has one of the greatest minimal Euclidian distances. When we consider its input pattern and its closest weight vector on the map, we see that it is the electronic parameter of the para substituent (R4) which brings the major contribution to the difference between these two vectors (Table 10). The next two entries concern compounds 54 and 61, whose minimal Euclidian distances are much smaller. The last entry shows the weight vector associated to a nonlabeled cell situated in {12, 18}, in the neighborhood of the cells

Table 10. Comparison between Some Input Patterns and Their Closest Weight Vectors on the Map

index	R			R2			R3			R4			R5		
{position}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Without Autoscaling															
input pattern 64	0.56	18.6	-3.04	-0.02	28.7	-7.71	0.0	7.24	0.0	-0.28	17.7	6.1	-0.02	28.7	-7.71
weights at {9,21}	0.55	18.4	-2.98	-0.02	28.5	-7.65	0.0	8.28	-0.36	-0.06	15.5	2.9	-0.02	28.7	-7.67
input pattern 54	0.56	18.6	-3.04	-0.02	28.7	-7.71	0.0	7.24	0.0	1.06	37.3	-2.65	-0.02	28.7	-7.71
weights at{14,18}	0.56	19.0	-3.0	-0.02	29.0	-7.70	0.0	7.30	0.0	1.0	39.0	-2.4	-0.02	28.7	-7.71
input pattern 61	0.56	18.6	-3.04	-0.02	28.7	-7.71	0.0	7.24	0.0	0.86	31.06	-1.6	-0.02	28.7	-7.71
weights at {13,19}	0.55	18.3	-3.01	-0.02	28.7	-7.71	0.0	7.31	-0.03	0.85	32.9	-1.88	-0.02	28.7	-7.71
weights at {12,18}	0.56	18.6	-3.04	-0.02	28.7	-7.71	0.0	7.45	-0.08	0.71	28.9	-1.93	-0.02	28.7	-7.71
With Autoscaling															
input pattern 64	1.09	1.09	-1.09	-1.0	1.12	-1.09	-0.46	-1.02	1.25	-1.14	-0.95	3.92	-0.44	0.14	-0.55
weights at {9,21}	1.05	1.05	-1.05	-0.98	1.11	-1.08	-0.45	-0.96	1.16	-0.83	-1.04	2.84	-0.43	0.14	-0.54
input pattern 54	1.09	1.09	-1.09	-1.0	1.12	-1.09	-0.46	-1.02	1.25	0.74	-0.1	0.99	-0.44	0.14	-0.55
weights at{14,18}	1.08	1.08	-1.08	-1.0	1.12	-1.09	-0.46	-1.02	1.25	0.68	-0.03	1.07	-0.44	0.14	-0.55
input pattern 61	1.09	1.09	-1.09	-1.0	1.12	-1.09	-0.46	-1.02	1.25	0.46	-0.37	1.34	-0.44	0.14	-0.55
weights at {13,19}	1.07	1.07	-1.07	-1.0	1.12	-1.09	-0.46	-1.01	1.25	0.45	-0.29	1.25	-0.44	0.14	-0.55

occupied by the molecules 54 and 61. Such empty cells are particularly numerous as the map used in this application has a very high resolution. The weight vectors of these empty neurons represent intermediary states between the occupied surrounding cells. They could also represent new potential compounds. For instance, the synaptic vector associated to the empty cell {12, 18} is very close to the input pattern of the (2,5-dimethoxy-4-chlorophenyl)isopropylamine 65 (Table 9).

It is probably not feasible to discover new derivatives with a greater activity than the most active one of this learning set. Unlike other QSAR studies, this method does not give any information on the drug mode of action.[30] Hallucinogenic effect prediction is much less precise than Clare's work based on regression analysis.[33] Like other clustering methods, such a network could prove useful when two or more different measurements or biological effects (e.g. desired activity and toxicity) have to be considered simultaneously for a target structure.

CONCLUSIONS

This work shows two explorations of Kohonen's self-organizing map algorithm in chemistry. The two maps used are unusually overdimensioned in comparison with the number of patterns to deal with. The first experiment shows the importance of the intermediate phases during the self-organizing process. The second network deals with a much more complex data set, not divided into a finite number of classes. Its prediction ability can be explained by the intuitive concept that similar molecules should have similar properties. Nevertheless, the results obtained for very dissimilar compounds should be cautiously considered. High-resolution maps are characterized by numerous empty cells which represent intermediary states between the occupied surrounding cells.

ACKNOWLEDGMENT

The helpful criticism and comments of the reviewers are gratefully acknowledged.

APPENDIX

A new function has been added to the SOM_PAK 1.2 program[23] to simulate toroidal edgeless maps (Chart 1).

Chart 1

```
#ifdef TOROIDAL /* modification to simulate an infinite, edgeless map */
#ifndef min
  #define min(x,y) ((x)>(y) ? (y):(x))
#endif
#ifndef abs
  #define abs(a) ((a)<0 ? (-a):(a))
#endif
/* new function for rectangular topology */
float toroidal_rect_dist(int bx, int by, int tx, int ty, int dimx, int dimy) /* rij */
{
  float ret, diff ;

  diff = bx - tx;
  diff = abs( diff ) ;
  diff = min( diff, abs(dimx - diff)) ;
  ret = diff * diff;

  diff = by - ty;
  diff = abs( diff ) ;
  diff = min( diff, abs(dimy - diff)) ;


  ret += diff * diff;
  ret = (float) sqrt((double) ret);

  return(ret);
}
#endif
```

REFERENCES AND NOTES

(1) Lacy, M. E. Neural Network Technology and its Application in Chemical Research. *Tetrahedron Comput. Methodol.* **1990**, *3*, 119–128.

(2) Zupan, J.; Gasteiger, J. Neural Networks—A New Method for Solving Chemical Problems or Just a Passing Phase? *Anal. Chim. Acta* **1991**, *248*, 1–30.

(3) Brown, S. D.; Bear, R. S.; Blank, T. B. Chemometrics. *Anal. Chem.* **1992**, *64*, R22–R49.

(4) Hopfield, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. Natl. Acad. Sci. U.S.A.* **1982**, *79*, 2554–2558.

(5) Bryngelson, J. D.; Hopfield, J. J.; Southard, S. N., Jr. A Protein Structure Predictor Based on an Energy Model with Learned Parameters. *Tetrahedron Comput. Methodol.* **1990**, *3*, 129–141.

(6) Friedrichs, M. S.; Wolynes, P. G. Molecular Dynamics of Associative Memory Hamiltonians for Protein Tertiary Structure Recognition. *Tetrahedron Comput. Methodol.* **1990**, *3*, 175–179.

(7) Tusar, M.; Zupan, J. In *Software Development in Chemistry 4*; Gasteiger, J., Ed.; Springer: Berlin, 1990; pp 363–376.

(8) Bruchmann, A.; Gotze, H. J.; Zinn, P. Application of Hamming Networks for IR Spectral Search. *Chemom. Intell. Lab. Syst.* **1993**, *18*, 59–69.

(9) Francelin, R. A.; Gomide, F. A. C.; Lancas, F. M. Use of Artificial Neural Networks for the Classification of Vegetable Oils After GC Analysis. *Chromatographia* **1993**, *35*, 160–166.

(10) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. In *Parallel Distributed Processing. Explorations in the Microstructure of Cognition. Volume 1: Foundations*; The M.I.T. Press: Cambridge, MA, 1986; Chapter 8, p 318.
(11) Kohonen, T. The Self-Organizing Map. *Proc. IEEE* **1990**, *78*, 1464–1480.
(12) Lippmann, R. P. An Introduction to Computing with Neural Nets. *IEEE ASSP Mag.* **1987**, April, 4–22.
(13) Sammon, J. W. A non linear mapping for data structure analysis. *IEEE Trans.* **1969**, *C18*, 401–409.
(14) Kowalski, B. R.; Bender, C. F. Pattern Recognition. A Powerful Approach to Interpreting Chemical Data. *J. Am. Chem. Soc.* **1972**, *94*, 5632–5639.
(15) Gasteiger, J.; Li, X.; Simon, V.; Novic, M.; Zupan, J. Neural Nets for Mass and Vibrational Spectra. *J. Mol. Struct.* **1993**, *292*, 141–159.
(16) Melssen, W. J.; Smits, J. R. M.; Rolf, G. H.; Kateman, G. 2-Dimensional Mapping of IR Spectra Using a Parallel Implemented Self-Organizing Feature Map. *Chemom. Intell. Lab. Syst.* **1993**, *18*, 195–204.
(17) Ferràn, E. A.; Ferrara, P. Topological Maps of Protein Sequences. *Biological Cybernetics* **1991**, *65*, 451–458.
(18) Ferràn, E. A.; Ferrara, P. Clustering Proteins into Families Using Artificial Neural Networks. *Comput. Appl. Biosci.* **1992**, *8*, 39–44.
(19) Ferràn, E. A.; Ferrara, P. A Neural Network Dynamics That Resembles Protein Evolution. *Physica A* **1992**, *185*, 395–401.
(20) Arrigo, P.; Giuliano, F.; Scalia, F.; Rapallo, A.; Damiani, G. Identification of a New Motif on Nucleic Acid Sequence Data Using Kohonen's Self-Organizing Map. *Comput. Appl. Biosci.* **1991**, *7*, 353–357.
(21) Rose, V. S.; Croall, I. F.; Macfie, H. J. H. An Application of Unsupervised Neural Network Methodology (Kohonen Topology-Preserving Mapping) to QSAR Analysis. *Quant. Struct.-Act. Relat.* **1991**, *10* (1), 6–15.
(22) Rose, V. S.; Croall, I. F.; Macfie, H. J. H. Kohonen Topology-Preserving Mapping: an Unsupervised Artificial Neural Network Method for Use in QSAR Analysis. *Pharmacochem. Libr., 16(QSAR: Ration. Approaches Des Bioact. Compd.)* **1991**, 213–216.
(23) SOM_PAK, the Self-Organizing Map Program Package was prepared by the SOM Programming Team of the Helsinki University of Technology, Laboratory of Computer and Information Science, Rakentajanaukio 2 C, SF-02150, Espoo, Finland. It is available free of charge by anonymous FTP connection on the Internet (cochlea.hut.fi or 130.233.168.48). Both UNIX and MS-DOS versions are available.
(24) Wolfram Research, Inc. *Mathematica*, Version 2.0 ed.; Wolfram Research, Inc.: Champaign, Illinois, 1991.
(25) Luce, H. H.; Govind, R. Neural Network Applications in Synthetic Organic Chemistry: I. A Hybrid system Which Performs Retrosynthetic Analysis. *Tetrahedron Comput. Methodol.* **1990**, *3*, 143–161.
(26) Curry, B.; Rumelhart, D. E. MSNet: A Neural Network which Classifies Mass Spectra. *Tetrahedron Comput. Methodol.* **1990**, *3*, 213–237.
(27) This fact can be checked with the "BP" program published by: McClelland, J. L.; Rumelhart, D. E. *Explorations in Parallel distributed Processing: A Handbook of Programs and Exercises*; MIT Press: Cambridge, MA, 1988. A small learning rate must be used in order to avoid local minima.
(28) Hansch, C.; Leo, A.; Unger, S. H.; Kim, K. H.; Nikaitani, D.; Lien, E. J. "Aromatic" Substituent Constants for Structure–Activity Correlations. *J. Med. Chem.* **1973**, *16*, 1207–1216.
(29) Clare, B. W. Structure–Activity Correlations for Psychotomimetics. 1. Phenylalkylamines: Electronic, Volume and Hydrophobicity Parameters. *J. Med. Chem.* **1990**, *33*, 687–702, and references cited therein.
(30) Gupta, S. P.; Singh, P.; Bindal, M. C. QSAR Studies on Hallucinogens. *Chem. Rev.* **1983**, *83*, 633–649, and references cited therein.
(31) Pauling, L. In *The Nature of the Chemical Bond*, 3 ed.; Cornell University Press: Ithaca, NY, 1960; p 260.
(32) Values are considered for $CDCl_3$ solutions. Ewing, D. F. ^{13}C Substituent Effects in Monosubstituted Benzenes. *Org. Magn. Reson.* **1979**, *12*, 499–524.
(33) Clare, B. W. Structure Activity Correlations for Psychotomimetics. 2. Phenylalkylamines—A Treatment of Nonlinearity Using the Alternating Conditional Expectations Technique. *Chemom. Intell. Lab. Syst.* **1993**, *18*, 71–92.