Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Multi-Layer Self-Organizing Networks based on Habituation

Riccardo Rizzo, Antonio Chella

Rapporto Tecnico N.:
RT-ICAR-PA-03-07 **dicembre 2003**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Multi-Layer Self-Organizing Networks based on Habituation

Riccardo Rizzo[1], Antonio Chella[2]

Rapporto Tecnico N.:
RT-ICAR-PA-03-07

Data:
dicembre 2003

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sezione di Palermo Viale delle Scienze edificio 11 90128 Palermo
[2] Università degli Studi di Palermo Dipartimento di Ingegneria Informatica Viale delle Scienze 90128 Palermo

Multi–Layer Self–Organizing Networks based on Habituation

R.Rizzo [a] and A.Chella [b]

[a] *ICAR - Italian National Research Council*
Viale delle Scienze 90128 Palermo, Italy
ricrizzo@pa.icar.cnr.it
Telephone: +39 091 238 262
fax: +39 091 652 91 24

[b] *DINFO - University of Palermo*
Viale delle Scienze 90128 Palermo, Italy

Abstract

Self–Organizing networks are used to develop a representation of an input manifold, to this purpose different flavors of networks are available. For many applications it is necessary to develop a simple and general approximation of the input manifold and this can be obtained using layered neural architectures where each layer reports a simplification of the precedent layer. To obtain a multi–layer or hierarchical self–organizing networks it is usually necessary to modify the chosen architecture in order to fit personal needs. In this paper it is proposed a general mechanism, obtained from habituation, useful to develop a multi–layer structure from a variety of self–organizing networks. The mechanism is based on a generalization of the neuron model obtained adding a simple habituation mechanism, the result is a simple but powerful method to obtain a multi–layer structure from any self–organizing networks. In this paper this habituation mechanism is analyzed and the applications with SOM and GNG networks are reported

Key words: self–organization, multilayer networks, SOM, Growing Neural Gas

---

1 Introduction

Generalization is an important characteristic of biological systems, a way to hide unnecessary or confusing details. In many applications, from pattern classification to system control a simple description of an input manifold can be used to take quick decisions for a fast response, and a more detailed repre-

Preprint submitted to Elsevier Science *5 December 2003*

sentation is useful when it is necessary to have a good approximation of the original input pattern or to refine an input command.

Actions or decisions related to an input pattern (for example from a set of sensors) are often taken after a matching against an internal model or representation. In neural networks models the comparison of the input with all available internal models it is not time consuming (all the comparisons should be made simultaneously) but it is probably unnatural: it seem to be more acceptable a mechanism that process the input patterns from general characteristics to small details. To accomplish that internal representations should be organized and accessed in a sort of hierarchy and the matching should be performed in a top–down fashion: from general model to details. At the beginning of the recognizing procedure the input pattern is compared to few very general prototypes in order to select the most important characteristics of the model and then continue the comparisons in order to match as many details as possible, or to select the best matching pattern. This approach is useful not only for classification purposes but also for planning. In navigation problems it is natural to plan a trajectory from a starting point to an end point, but details are useful only in the last phase of the movement, when it is necessary to reach precisely the end point of the movement.

A possible way to obtain a multi–level mapping of a sensory input space is to use many overlapped self–organizing neural fields with different characteristics, and make the inputs to "flow" from a neural field to another: the neural fields should learn and generalize in different way the input distribution and their statistics. To this purposes it is necessary to add to neurons a new feature that take into account the activity of the neuron during the learning phase. Neural models can have many time related features, one of these features is habituation: a reversible decrement of neural response, that makes the neuron to ignore an input if it is too frequent. In this paper the habituation mechanism is used to control the flow of the input thought the different layers of a structure made of many overlapped neural fields. Habituation is a local mechanism that is added to each neuron: it is the single neuron that habituate to the local input, not the whole network, and can be used in a variety of self–organizing neural networks to build multi–layer structures.

The paper is organized as follows: the next section contains a review of the previous approach to hierarchical self organizing systems, an analysis of the habituation mechanism is in section 3 and the application of the habituation to obtain a multi–layer system is reported in section 4; section 5 contains an analysis of the results for a multi–layer SOM, a multi-layer GNG network and a mixed structure. The conclusions and future works are in section 6.

2 Previous Works

Hierarchical neural network architectures, obtained from a self–organizing network are not new, but were often developed as modification of a single architecture, and usually inspire by a specific problem. There are many modification of the SOM algorithm obtained using a multi–layer structure as in (Koikkalainen and Oja, 1990) and in (Miikkulainen, 1990) where the hierarchical structure is defined in advance with a fixed number of layers and a fixed size of the maps for each layer. Another hierarchical structure called HOSOM (from Hierarchical Overlapped SOM) was proposed by Suganthan (1999); in this architecture it is possible to obtain on a single layer more overlapping SOM and the final classification is obtained using a voting scheme. In (Lampinen and Oja, 1992) it is shown that a hierarchical SOM structure, were SOM outputs are simply fed into another SOM as inputs, is capable to form arbitrarily complex clusters, resulting in a more flexible classifier.

Ritter, Martinetz, Schulten (1990) developed a hierarchical SOM structure focused to control a robotic manipulator. The structure was composed by a set of two–dimensional SOM each of them is an element of a three–dimensional SOM lattice that maps the space accessible by the robotic manipulator. Each two–dimensional small SOM maps the orientation of the gripper to the orientation of the target, where the arm configuration is obtained using the three dimensional SOM. During the learning phase the small SOMs are not independent but coupled using the same neighborhood mechanism that is used to train the three dimensional SOM framework. This architecture, although effective, is focused on the specific topic, because the two problems, robot arm position and gripper orientation, are not tightly coupled.

Recently, the Growing Hierarchical SOM was developed by Rauber, Merkl and Dittenbach (2002), a growing multi–layer network that uses a data–driven algorithm in order to add units and layers following the complexity of the input manifold. This network was used for classification purposes on a large set of documents.

A hierarchical algorithm obtained from the Growing Cell Structure, introduced by Fritzke (1994), is described by Burzevski and Mohan (1996). This architecture was studied in order to obtain a hierarchical classification structure where each unit on the top layer can have a sub–network of the bottom layer. The insertion of the new sub–network occurs when the units of one layer have the same value of the signal counter, a variable that is incremented each time the neural unit is the best matching unit ($b.m.u$).

Ersoy and Hong (1990) proposed a supevised, hierarchical, multi–layer architecture driven by classification error. The neural network adds a new neural layer, and further process the input data, if it is not able to correctly classify all the input samples. Besides the data processing between each layer this "flows" of the input data from a layer to the next one is also present in the architecture proposed in this work.

Compared to the precedent works the approach presented is different because the multi–layer structure does not came from a pre–existent neural network model, it is obtained from an extension of the model of the neural unit, generalized including a simple habituation mechanism. This mechanism can be added to neurons used in a set of self–organizing networks and activated by a suitable threshold. If the mechanism is activated the multi–layer architecture is obtained coupling many self–organizing networks even of different nature.

3 The Habituation Mechanism

The habituation is a reversible decrement of the neural response to a repetitive stimulus. The response recovers only after a period in which there is no activity, and the longer the pre–synaptic neuron is active the slower it recovers. This is the only aspect of the habituation mechanism that was taken into account in the present model; many other characteristics, for example modification of the habituation speed related to the frequency of habituation–recovery sequence, are not taken into account.

The habituation mechanism is not commonly used in artificial neural networks models, but some architectures were proposed in the past. In (DeSieno, 1988) the conscience mechanism, a mechanism similar to habituation was used; conscience is used to prevent that a unit wins too often, the unit "feels guilty" and prevents itself from winning excessively. The purposes of the algorithm is to obtain a better approximation of the input manifold. Habituation was also used to implement a novelty detection variant of the SOM by Marsland, Nehmzow and Shapiro (2000), where an output unit with habituable synapses is used to detect the new patterns: the output of the network is diminished if the same neural unit responds to many input patterns. In Marshland, Shapiro, Nehmzow (2002) habituation was used to build a growing neural network: new units are added to the network when the same neural unit answer to many input patterns.

We take into account the simplified habituation model that is described in the following equation (Stanley, 1976):

$$\tau \frac{dh_s(t)}{dt} = \alpha \left[h_0 - h_s(t) \right] - S(t), \tag{1}$$

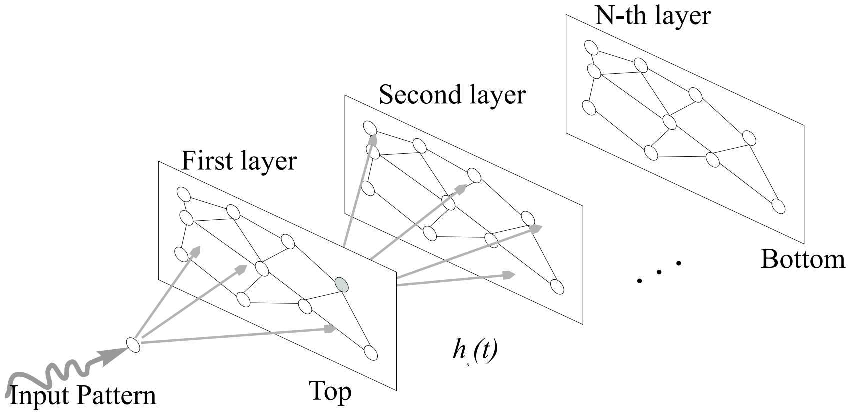

Fig. 1. A representation of the organization of the multi–layer structure and the "flow" of the input pattern

where:

- $h_s(t)$ is the strength of the input synapse
- $S(t)$ is the stimulus

Assuming that a constant non–zero input S is applied, the solution that we will take into account is given by:

$$h_s(t) = h_0 - \frac{S}{\alpha}\left(1 - e^{-\frac{\alpha t}{\tau}}\right), \tag{2}$$

where h_0 is the initial value of the weight.

The proposed architecture is represented in fig 1. A synapses connect each neuron of the first layer to all the neurons of the second layer and so on from the top–layer to the bottom–layer: if the winning neuron habituates to the input stimulus the input "flows" to the next layer. This one–to–all connection is the same exploited by Lampinen and Oja (1992) but the habituation of the synapse modifies the properties of the hierarchical structure. This sort of "weak" connection, that connects a unit to all the units of the next layer allows to mix different neural networks in the same architecture: for example it is possible to use a growing neural network on the first layer and a SOM (fixed topology) on the second layer, in order to exploit the different advantages of the networks.

4 Using the Habituation Mechanism in a Multilayer System

The proposed mechanism exploits the habituation to allow the self–organization of many neural networks organized in layers as shown in fig.1. The basic idea is that the neuron that is habituated to a frequent input pattern will not learn but allows the input to flow to the next layers, from the top layer to a bottom layer. If the neurons on the top layer quickly habituate to the inputs, the network of the layer will learn only the shape of the input manifold, other details will “flow” through and will affect the bottom layers. As an example if the inputs are more concentrated in an area of the manifold the neuron of the first layer became habituated to this kind of input and will “pass” the inputs to the second layer. If the habituation threshold of the second layer is high enough, this layer will “hold” the input and learn more producing a more detailed representation of the input manifold. This mechanism makes the top layer neurons less sensitive to input pattern very close in input space as the right side of fig.3 shows. These similar patterns are sent to the neurons in bottom layers so that details are buried in these “low” layers of the neural system, and top layers present a high–level description of the input manifold.

This mechanism is obtained adding to the neuron model an habituation variable that is updated each time the neuron is the *b.m.u.*, and setting an habituation threshold t_r that is a characteristic of the whole neural network.

The amount of the increment of the habituation variable is a function of the activation a of the neuron, as defined in (Marshland, Shapiro, Nehmzow, 2002):

$$a = e^{\frac{-\|\xi - w\|}{\tau}} \tag{3}$$

where ξ is the input pattern and $\mathbf{w}$ is the weight of the winning neuron (best matching unit or *b.m.u.*); τ parameter decrease the activation value when the input is far from the *b.m.u.* unit. The value of the habituation variable over the time is given by:

$$Ab(t) = Ab(t-1) + Ab_0 * a \tag{4}$$

and the habituation variable saturates at Ab_{MAX}.

In the present model the habituation is strictly local: only the synapse of the b.m.u. unit habituates, not the one of the neighboring neurons.

Habituation variables are decreased during the time for all the units in the network even if it does not catch the input pattern: at the end of each learning step, all the habituation variables are multiplied by $\beta < 1$.

The strength of the input synapse for the top–layer of the network is:

$$h_s(t) = \begin{cases} 0 \text{ if } Ab(t) \geq t_r \\ 1 \text{ if } Ab(t) < t_r \end{cases} \tag{5}$$

if $h_S(t) = 0$ the input flows to the bottom layers, if $h_S(t) = 1$ the first layer catches the input pattern and learn, so the neuron will learn only if the habituation is below a threshold tr, of course is $tr < Ab_{MAX}$, and using $tr > Ab_{MAX}$ disable the habituation mechanism.

During the test stage of the neural multi–layer system for each input it is possible to have one answer for each layer at different levels of detail.

5 Experimental Results

The habituation mechanisms was tested using the SOM (Kohonen, 1982) and the GNG (Fritzke, 1994) networks: the SOM is the most famous and used self–organizing network and it is a reference for many new architectures. The properties of SOM are useful in this case to highlight some characteristics of the proposed mechanism and to indicate the direction for further studies. On the other side the GNG has some features that make the multi-layer structure very interesting: for example using GNG it is possible to build a very simple multilayer self–organizing structure with a flexible topology.

5.1 Multi–layer SOM Architecture

A two –layer SOM structure is shown in left side of fig.2, where the unit **a** is the *b.m.u.* for an input pattern, units **b** and **c** are the neighborhood units, and the set of thin lines indicate the connections between the first layer and the second one. The connections between the units of the same layer are the one that propagates the update of the neural weights, this effect is at the core of the self–organizing principle and was well studied in the literature. The connections from a unit of the first layer (the **a** unit for example) to all the units of the second layer are the one that transmit the input pattern when the unit became habituated. This is implemented in the modification of the SOM learning algorithm reported below (a similar modification is required for the GNG network); assuming that **a** is the *b.m.u.*:

1. **if** ($A_b(t+1) < t_r$)

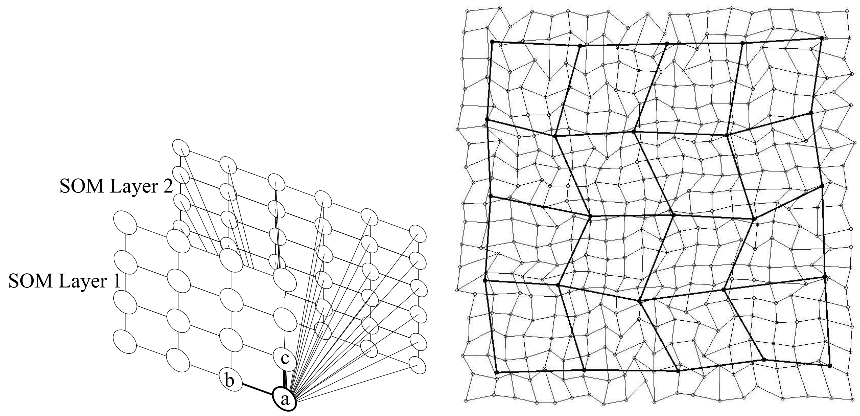

Fig. 2. (left) The double layer SOM structure; (right)a uniform two–dimensional distribution approximated by the two layer SOM

1.1 $w_a(t+1) = w_a(t) + h(t) * (v(t) - w_a(t))$
1.2 *update all the* **a** *neighborhoods*

1.3 $A_b(t+1) = A_b(t) + A_{b0} \cdot e^{-\frac{\|v(t)-w_a(t)\|}{\tau}}$
1.4 **if** $(A_b(t+1) > A_{bMAX})\ A_b(t) = A_{bMAX}$

else *pass the input to the next layer*

2. Decrement the habituation of all the N units:
$A_{bi}(t+1) = A_{bi}(t+1) * \beta \qquad i = 1, 2, ..., N$

Notice that the learning steps 1.1 and 1.2 are executed if the habituation is below the threshold t_r and the update of the habituation variable follows, otherwise the input is passed to the next layer. The relaxation of the network is made in both cases.

A two layer SOM trained using a two dimension uniform distribution is shown in fig.2 (right). In the learning stage τ parameter is set to have the same number of input pattern that train the first layer and the second layer, the meaning of the τ parameter and the discussion about the influence of habituation parameters on the learning phase is left to the next sections. The two network are uniformly distributed on the surface and it is possible to see the two different mapping at different levels of detail: upper level SOM segment the input pattern space in a lesser number of pieces than the second layer network. The first thing to notice about this architecture is that there is not a topological link between the two SOM on the two layer; this because, as already said, the units of the first layer transmit the input pattern to all the units of the next layer and the topological information is completely lost. So the symmetry that is noticed in fig.2 it is not maintained if the two SOM have a different

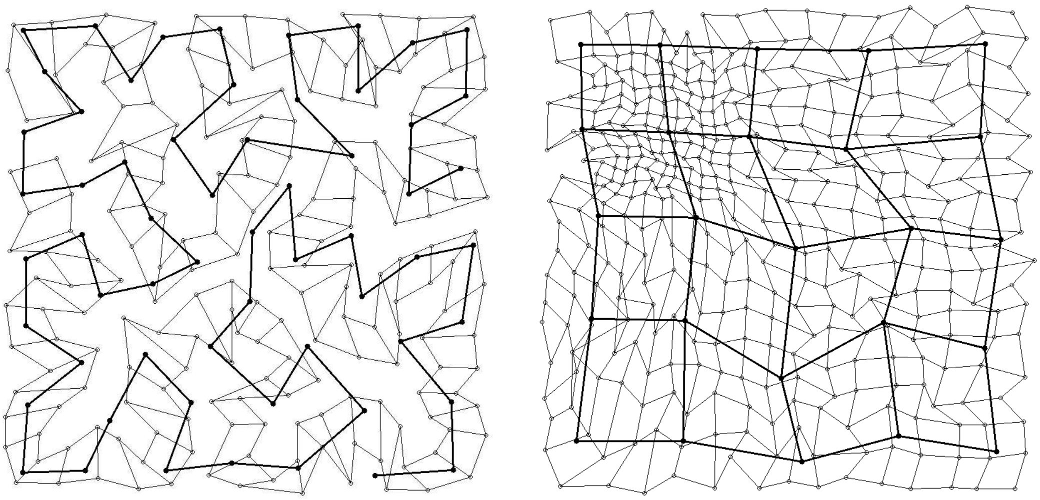

Fig. 3. (left) The configuration of a multilayer SOM ribbon shaped; (right) A non uniform distribution applied to the two layer SOM.The parameter are in table 1.

topology. In fig.3 (left) the input distribution is the same but the topology of the SOM is very different: the network on the the first layer is made by 2×40 units and the one on the second layer is made by 3×80 units. Besides of the irregular shape of the first layer it is possible to notice that the SOM ribbon on the first layer is not following the path of the one on the second layer. Habituation only controls the quantity of the inputs that flows from the first layer to the second layer, in the present architecture the topological information does't flow thought the layers.

The mechanism based on habituation reveals an interesting property if the input pattern are not uniformly distributed over the input space. Fig. 3 (right) shows the results of the training of a two–layer network using a not uniform input distribution. While the SOM on the second layer has not habituation mechanism and distribute more neural units where the input patterns have an higher density, an effect that is known in the literature as magnification factor, the first layer network pass the input pattern if they are more frequent and is not effected by the density of the input patterns.

Table 1
SOM network parameters

Network	Topology	ϵ	σ	β	Ab_0	t_r	τ	h_{MAX}
fig.3 (left)	1^{st} layer 1×60	0.002	30	0.995	2.0	0.01	0.0045	2.0
	2^{nd} layer 2×40	0.003	40	0.99	1.0	3	0.0045	2.0
fig.3 (right)	1^{st} layer 5×5	0.002	5	0.995	2.0	0.01	0.0045	2.0
	2^{nd} layer 20×20	0.003	20	0.99	1.0	3	0.0045	2.0

Fig. 4 shows a two–layer classifier easily obtained using a multi–layer SOM with habituation. Neural units on the first layer can be interpreted as the

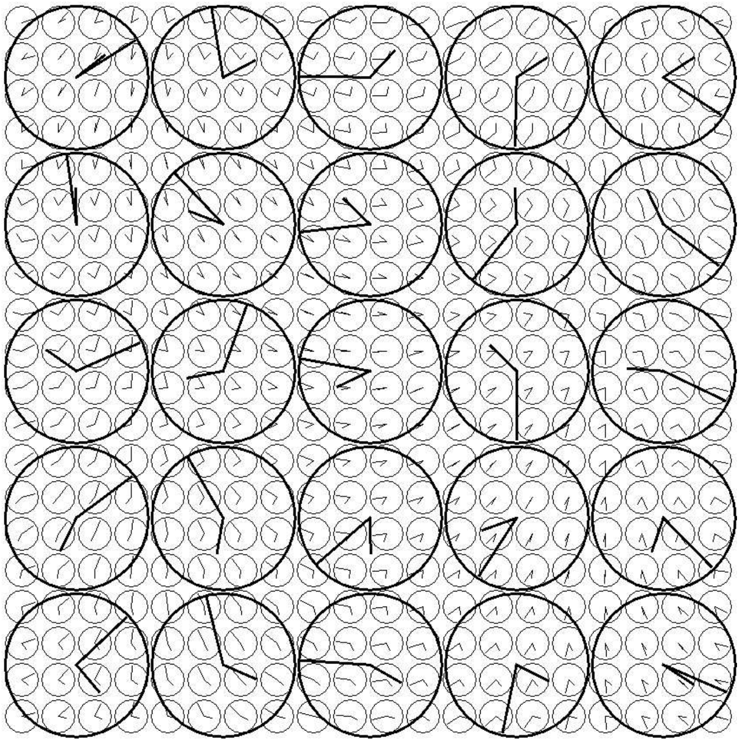

Fig. 4. A uniform two–dimensional distribution approximated by the two layer SOM. The clocks in bold represent the units on the first layer.

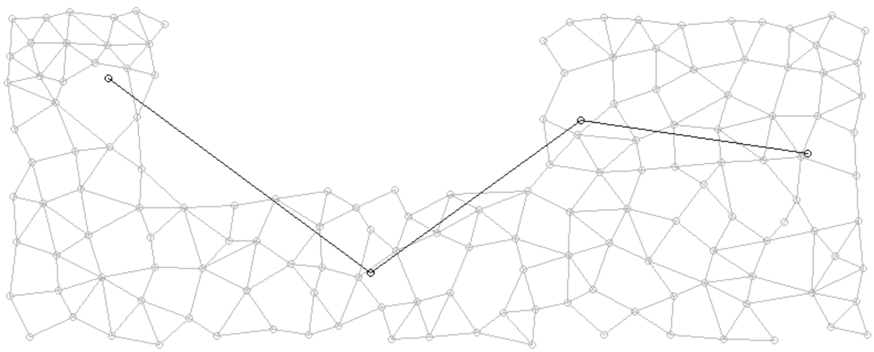

Fig. 5. First Experiment: the top–layer GNG network has 4 units and the bottom–layer network has 150 units (the maximum value).

cluster prototypes or a coarse–grained segmentation of the input space, while second layer represent a finer segmentation.

5.2 Multi–layer GNG Architecture

A multi–layer GNG structure developed using habituation shows other interesting behaviors.

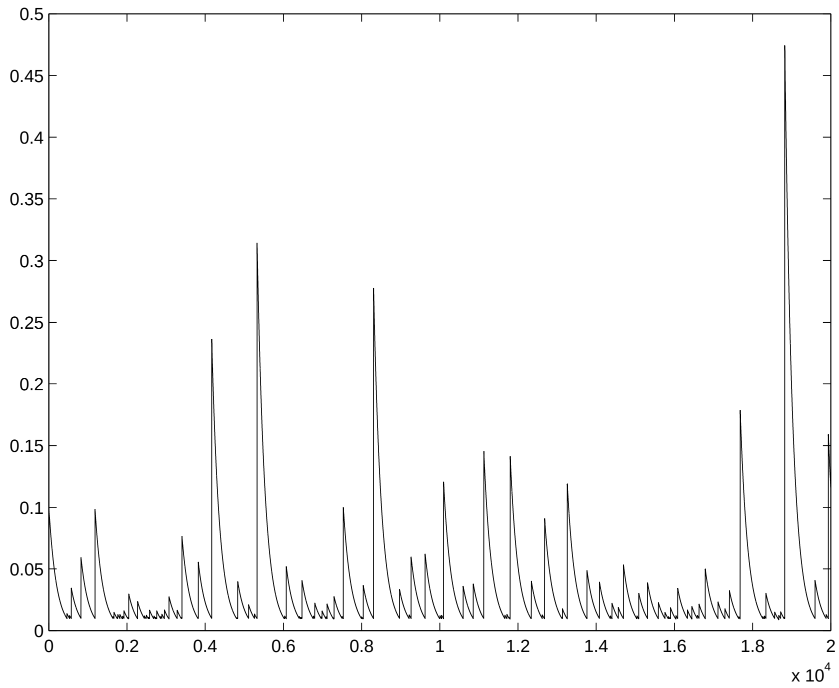

Fig. 6. Habituation variable value for the experiment in fig.5

The GNG network is growing at fixed pace, adding one new unit every λ learning steps. Using the habituation mechanism it is possible to slow down the grow of the top–layer network in order to obtain a simplified representation of the input manifold. In all the reported experiment both the GNG networks have the same networks parameter and the same maximum number of neural units (150 units), the differences are in the parameters related to the habituation mechanism: $Ab_0, Ab_{MAX}, \tau, \beta$ and the threshold tr. Habituation parameters are used to control the growing mechanism by counting a learning step only if the network does not "pass" the input to the next layer (i.e. the $b.m.u.$ has $A_b(t) < t_r$). As in the SOM experiments, the bottom–layer GNG network parameters are fixed in order to cancel the habituation effect, while the top–layer habituation parameters are tuned to shown different aspects of the proposed mechanism.

The number of top–layer neural units can be tuned by varying the threshold tr or the τ parameter. Varying the τ it is possible to change the increase of the habituation variable due to the amount of activation of the neuron: a sharp exponential gives to the habituation value a little increase if the match between the input and the neural unit is small, resulting in a less frequent "release" of the input pattern to the bottom layer. The effect is an increase of the number of neurons in the top–layer, because more learning steps are made on the neural net on this layer. The figs 5, 7 and 8 show some learning results, and the corresponding parameters are in Table 2. In the figures the black GNG network is the top–layer network, and the gray one is the bottom–layer, The network parameters for both the GNG networks are: $\epsilon_b = 0.2, \epsilon_n =$

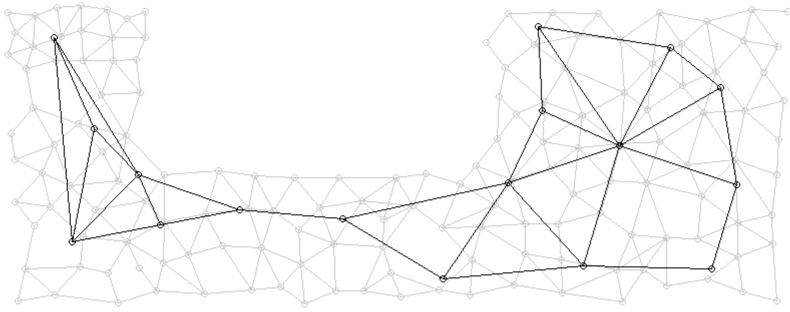

Fig. 7. Second experiment: the top–layer GNG network has 17 units and the bottom–layer network has 150 units (the maximum value).

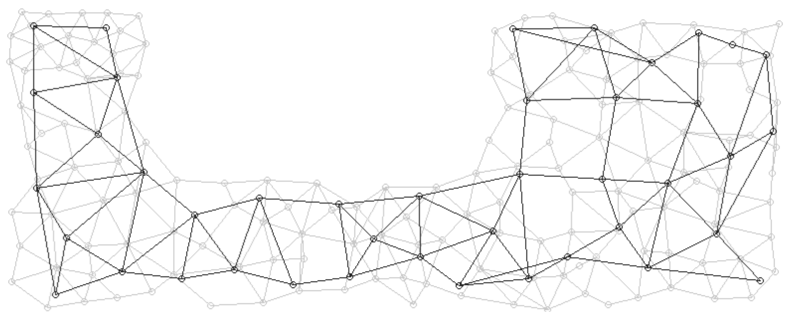

Fig. 8. Third experiment: the top–layer GNG network has 42 units and the bottom–layer network has 150 units.

$0.006, age_{MAX} = 50, \lambda = 100$. Fig.6 reports the habituation variable value of one of the units of the top–layer network during the learning stage.

Table 2
Network parameters

Top–layer GNG	β	Ab_{MAX}	Ab_0	tr	τ
fig5	0.995	2.0	1.0	0.01	0.02
fig7	0.995	2.0	1.0	0.01	0.01
fig8	0.995	2.0	1.0	0.01	0.005

It is interesting to evaluate the total number of neural units on both layers versus the τ value: fig.9 shows how the total number of neuron decrease with the τ value. The experiments shown that around $\tau = 0.0045$ and $\tau = 0.004$ the bottom layer neurons decrease from 150 units, reaching a value around fifty when $\tau = 0.001$.

If we consider the discrete time instants in which the neurons of the top–layer are "available" to the training (i.e. they are not habituated to the input, so they are ready to learn) it can be shown that there is the same effect: increasing

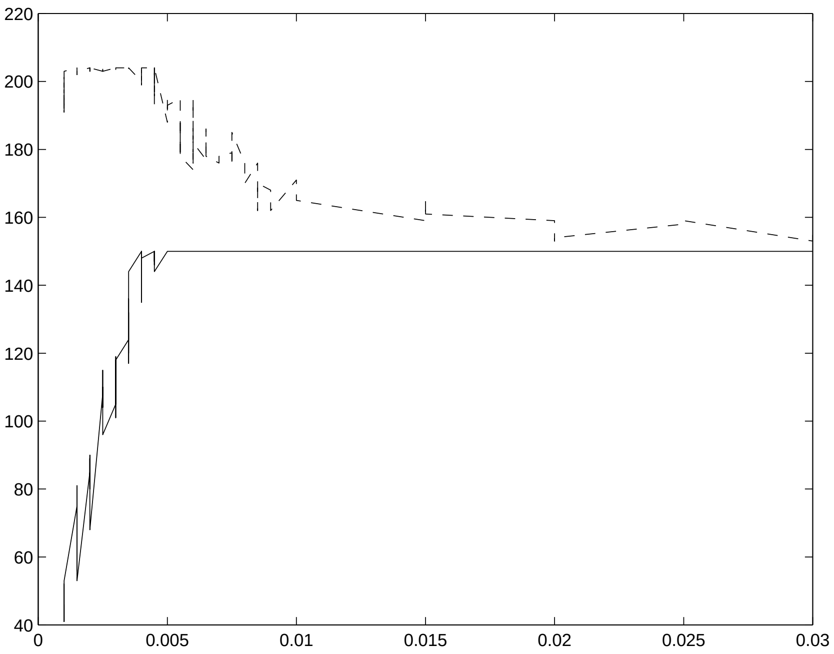

Fig. 9. The number of total neural units versus the value of τ: the upper dashed line is the total number of neurons of the multi–layer system; the lower line is the number of the bottom–layer network.

the τ value the number of "available" learning steps decrease, as shown in fig.10. This means that the top layer network is less trained as a consequence it also develops less neural units. This will need more investigation in the future: at the present the synapse that connects the different layer is an ON–OFF synapse, the learning is obtained in the top–layer or in the bottom–layer because the input pattern can be captured by just one layer. Probably it should be better to make this mechanism more flexible, allowing the network to learn "a little" from the habituated input, not just pass the "whole" input.

Fig.11 shows that the hierarchical GNG structure is capable to follow very complex two–dimensional input distributions. The input manifold is chosen in order to simulate a complex navigation path: the dark areas represent the obstacles and the white ones the available free path. As it is possible to see the bottom layer of the GNG network fills all the available space, and the top–layer GNG (the black one) has a simpler shape but is capable to follow the input distribution.

5.3 A Mixed Architecture

Fig. 12 shows a structure where first layer is a GNG network and the second layer is a SOM with a square topology made of 40×40. The multi–layer structure is trained on a manifold that highlights the advantages of GNG on

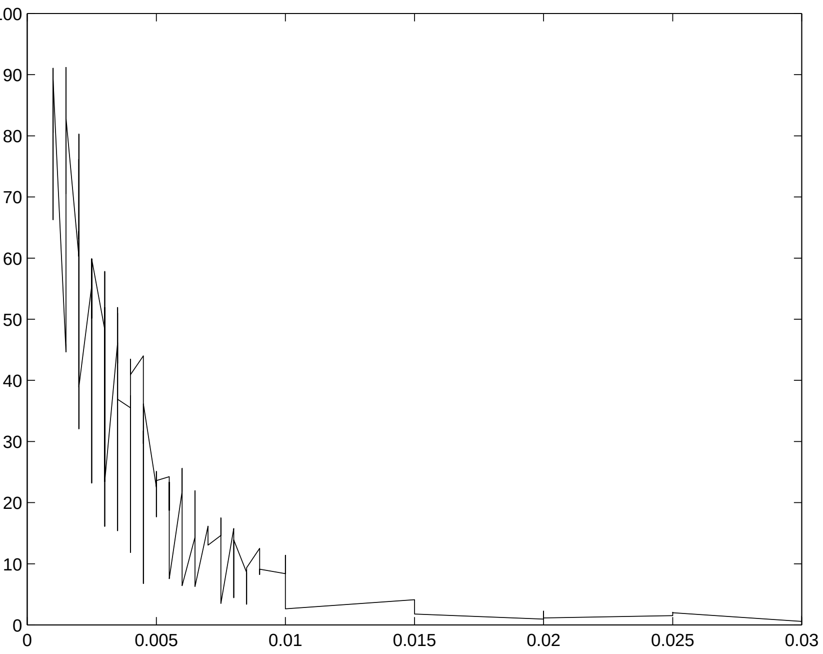

Fig. 10. The availability of a sample unit of the top layer network (percentile of the total learning time) versus τ values.

SOM architecture because the SOM net cannot follow the input distribution and leave many units unused. The GNG instead grows "inside" the double–T input manifold and better covers the input manifold. The parameters of the networks are in table 3: the t_r value was not critical and the main effect is to drive the number of neural units of GNG network.

Table 3
GNG and SOM network parameters

	ϵ	ϵ_n	σ	a_{MAX}	λ	β	Ab_0	t_r	τ	h_{MAX}
GNG	0.2	0.006	–	50	100	0.995	1.0	0.01	0.04	2.0
SOM	0.003	–	40	–	–	0.99	1.0	$> A_{bMAX}$	0.04	2.0

6 Conclusions

In this paper we have proposed a new method to develop a multi–layer self–organizing system using a simplified habituation mechanism. This method is useful when it is necessary to obtain a representation of a complex manifold and, on the same time, it is convenient to hide unnecessary details.

The habituation mechanism proposed allows to connect many self–organizing networks in a multi–layer structure and to train the whole structure. Due to the fact that the mechanism proposed is not linked to a particular neural

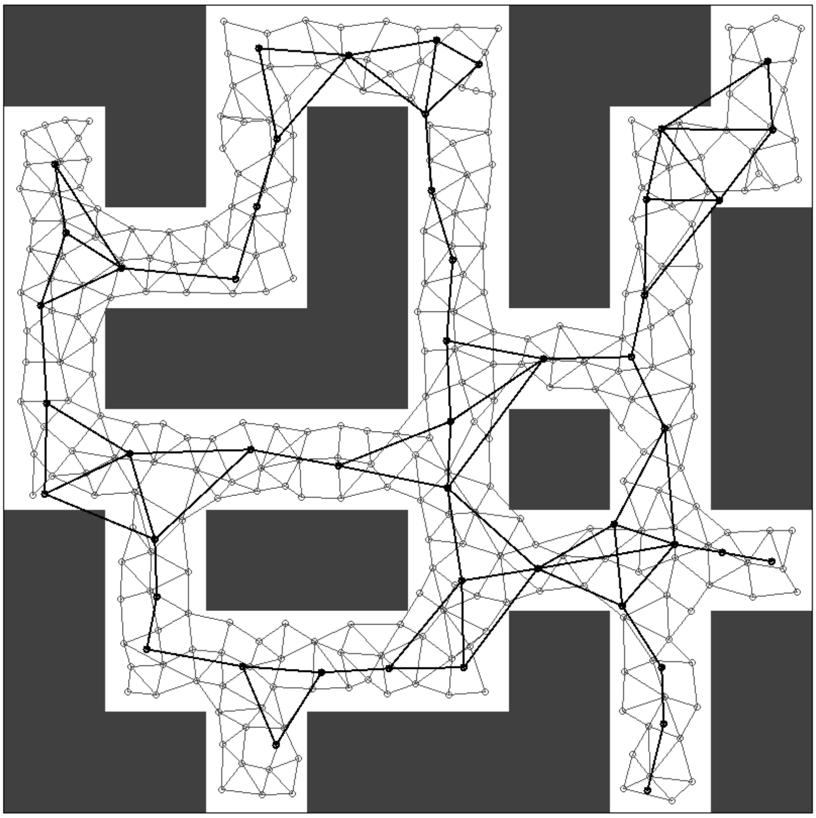

Fig. 11. The approximation of a complex two dimensions distribution, obtained using the same network parameters in fig.7, but using 300 units as a maximum dimension of the networks.

network model it is possible to obtain a multi–layer system using a different model for each layer in order to obtain a very flexible, multi–layer structure.

Moreover some interesting problems are still open: for example a better coupling between neurons of different layers should increase learning speed but probably will need a more specialized architecture; another problem regards the stability–plasticity trade–off: the GNG network never stop the learning phase, but it should be more plausible that the top layer of the neural system should be the more stable (because is the more general) while the bottom layers, carrying all the details should be more plastic. Other open issues are related to the study of the influence of the habituation on the magnification factor of the SOM network (Bauer et al., 1996), (Ritter and Schulten, 1986), to the transmission of the topological information from one layer to another (this means that the top layer network should pass the input only to a subset of the units of the network on the bottom layer), and to a more detailed implementation of the habituation mechanism.

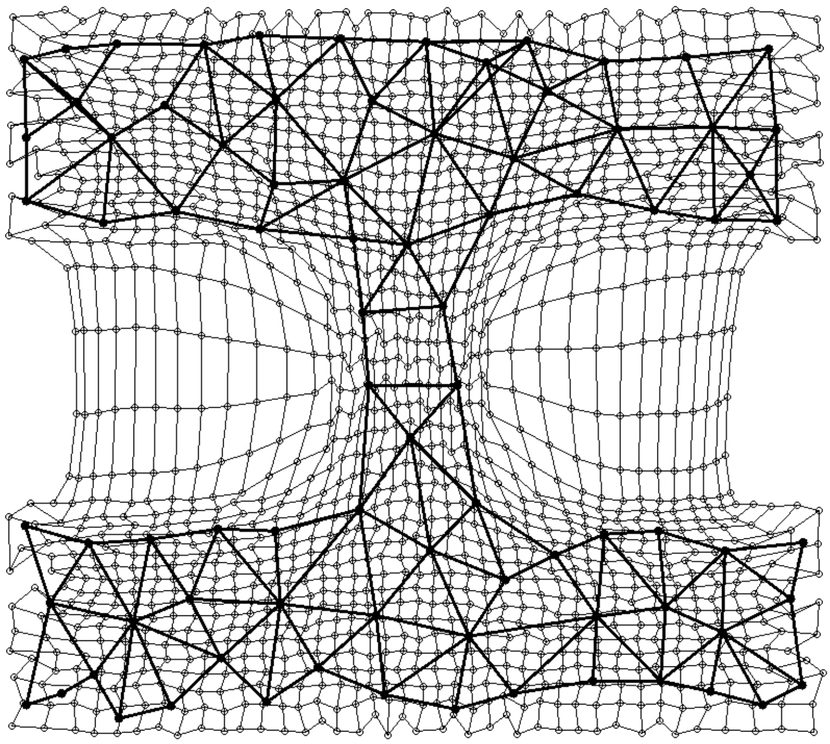

Fig. 12. A multi–layer mixed structure with GNG (first layer) and SOM (second layer); the parameters of the two networks are in the table 3.

References

Bauer, H.U., Der, R. and Herrmann, M., "Controlling the magnification factor of self–organizing feature maps", *Neural Computation*, 8(4):757-771, 1996.

Burzevski, V. and Mohan, C. K., "Hierarchical Growing Cell Structures", ICNN96: Proc. Int'l. Conf. Neural Networks, 1996.

DeSieno, D., "Adding a conscience to competitive learning", IEEE International Conference on Neural Networks,, 24-27 July 1988, pp. 117–124, vol.1

Ersoy, O. K. and Hong, D., "Parallel, Self–Organizing, Hierarchical Neural Networks", *IEEE Transactions on Neural Networks*, vol.1 n. 2, June 1990.

Fritzke, B., "A Growing Neural Gas that Learns Topologies", in G. Tesauro, D.S. Touretzky, & T. K. Leen (Eds.), *Advances in Neural Information Processing Systems 7 (NIPS'94)*, Cambridge: MIT Press.

Fritzke, B., "Growing Cell Structures–A Self-Organizing Network for Supervised and Unsupervised Learning",*Neural Networks*, vol.7, n.9, pp.1441–1460, 1994.

Lampinen, J., and Oja, E., "Clustering Properties of Hierarchical Self–Organizing Maps", *J. Math. Imaging Vision*, vol. 2, n.3, 1992, pp. 261–272.

Kohonen, T., "Self-Organized formation of topologically correct feature maps", *Biological Cybernetics* , Vol. 43, pp. 59-69, 1982.

Koikkalainen, P. and Oja, E., "Self–Organizing hierarchical feature maps", in Proc. Int. Joint Conf. Neural Networks, vol. 2, San Diego, CA, 1990, pp.

279–284.
Marshland, S., Shapiro, J., Nehmzow, U., “A Self–Organizing Network that Grows When Required”, *Neural Networks* 15(2002), pp. 1041–1058
Marsland, S., Nehmzow, U. and Shapiro, J., “A Real-Time Novelty Detector for a Mobile Robot”, EUREL European Advanced Robotics Systems Conference, Salford 2000.
Ritter H., Martinetz T. and Schulten K., Neural Computation and Self-Organizing Maps, Addison-Wesley, Reading, 1992
Miikkulainen, R., "Script Recognition with Hierarchical Feature Maps”, *Connection Science*, 2(1990), pp. 83–101
Rauber, A., Merkl, D. and Dittenbach, M., "‘The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High-Dimensional Data"’, *IEEE Trans. Neural Networks*, vol. 13, pp. 1331–1341, November 2002.
Ritter, H. and Schulten, K., "On the Stationary State of Kohonen’s Self–Organizing Sensory Mapping”, *Biological Cybernetics*, 54, 99-106, (1986)
Stanley, J.C.,“Computer Simulation of a Model of Habituation”, *Nature*, 261, pp. 146–148, 1976.
Suganthan, P.N., “Hierarchical Overlapped SOM’s for Pattern Classification”, *IEEE Transactions on Neural Networks*, vol. 10, n.1, January 1999, pp. 193–196.