The Impact of Network Topology on Self-Organizing Maps

Fei Jiang[1,2], Hugues Berry[1], Marc Schoenauer[2]

[1] Project-Team Alchemy,
INRIA Saclay – Île-de-France,
Parc Orsay Université
28, rue Jean Rostand
91893 Orsay Cedex, France

[2]Project-Team TAO
INRIA Saclay – Île-de-France &
LRI (UMR CNRS 8623)
Bât 490, Université Paris-Sud
91405 Orsay Cedex, France

Fei.Jiang@inria.fr, Hugues.Berry@inria.fr, Marc.Schoenauer@inria.fr

ABSTRACT

In this paper, we study instances of complex neural networks, i.e. neural networks with complex topologies. We use Self-Organizing Map neural networks whose neighborhood relationships are defined by a complex network, to classify handwritten digits. We show that topology has a small impact on performance and robustness to neuron failures, at least at long learning times. Performance may however be increased (by almost 10%) by evolutionary optimization of the network topology. In our experimental conditions, the evolved networks are more random than their parents, but display a more heterogeneous degree distribution.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE—*Learning*(Connectionism and neural nets); G.1.6 [**Mathematics of Computing**]: NUMERICAL ANALYSIS—*Optimization*(Stochastic programming)

General Terms

Algorithms, Design, Experimentation

Keywords

Self-Organizing Map, Topology, Evolutionary Optimization

1. INTRODUCTION

In the last decade, complex network topologies, e.g. small-world or scale-free ones, have attracted a great amount of interest (for a review, see [2]). According to the famous algorithm of Watts and Strogatz [20], small-world networks are intermediate topologies between regular and random ones. Their properties are most often quantified by two key parameters [18]: the clustering coefficient ($< C >$) and mean-shortest path (MSP). The clustering coefficient quantifies the extent to which the neighbors of a given network node (i.e. the nodes to which it is connected) are, on average, themselves interconnected. It reflects the network capacities of local information transmission. The graph distance between two nodes of the network is the smallest number of links one has to travel to go from one node to the other. The MSP is the average graph distance of the network and indicates the capacities of long distance information transmission.

Figure 1 illustrates an example of small-world network formation. Starting from a regular network (a ring in the figure, where each node is connected to its four nearest neighbors), one re-wires each link with (uniform) probability p. A regular network ($p = 0$) has a high clustering coefficient but its MSP is long. At the other extreme, totally random networks, obtained with $p = 1$, display a small clustering coefficient but a short MSP. For small-to-intermediate values of p (for instance $p \in [0.004, 0.100]$ for rings with 1000 nodes and 10 links per node [20]), the obtained networks preserve a high clustering coefficient while their MSP is already small. Such networks have optimal information transmission capacities, both at the local and global scales [7] and are called small-world networks. Many "real-world" networks, either technological (the Internet, electric power grids), social (collaboration networks) or biological (neural networks, protein or gene networks), have been found to display such a small-world topology [2].

But small-world networks fail to capture some characteristics of some real-world networks (e.g the presence of "hubs" connecting almost disconnected sub-networks), and other types of networks have been studied, like the scale-free networks [1]: in those networks, the degree distribution $P(k)$ (that gives the probability that a node, chosen at random in the network, connects with k other nodes) follows a power law relationship: $P(k) \sim k^{-\gamma}$. The "preferential attachment method" [1] can be used to build such topologies, reflecting also the dynamical aspect of those network, whose size can increase over time. Many "real-world" networks also fall into this category (the Internet, airplane routes, metabolic networks, social collaboration networks...) [2].

Importantly, the connectivity structure of such complex networks (i.e. their topology) is a crucial determinant of their information transfer properties [2]. Hence, the computation made by complex neural networks, i.e. neural networks with complex connectivity structure, could as well be dependent on their topology. For instance, recent studies have shown that introducing a small-world topology in

GEC'09, June 12–14, 2009, Shanghai, China.

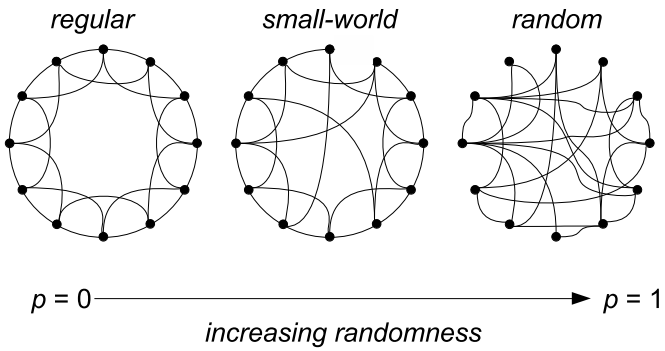

Figure 1: Small-world networks according to Watts and Strogatz. Starting with a regular ring network ($p = 0$, left), each link is rewired to a randomly-chosen destination node with probability p. When $p = 1$ (right), the network is a random one. At small-to-intermediate p values (depending on the network size), the network displays small-world properties (center). Adapted from[20].

a multilayer perceptron increases its performance [15, 3]. However, other studies have inspected the performance of Hopfield [6, 12, 11, 17] or Echo state networks [4] with small-world or scale-free topologies and reported more contrasted results.

Using artificial evolutionary algorithms to modify the topology of neural networks so as to optimize their performance has become widespread in the artificial neural networks community for several years [21, 16, 13]. But, in most cases, the resulting topologies are quite simple and the number of connections/neurons is low (typically a few dozens at most). Furthermore, the evolutionary mechanisms used in most of these studies do not modify the topology in an intensive manner. Hence, the optimization of large, complex neural networks through artificial evolution has hardly been studied. However, some recent results have demonstrated the importance of the topology for networks in related areas, such has 1D-cellular automata [10] or boolean networks [14] – thanks to their optimization using Evolutionary Algorithms.

In the case of Self-Organizing Maps (SOMs), the role of network topology has been studied for several years under the perspective of the relationship between data topology and network topology. In particular, much effort has been devoted to the development of network topologies that preserve that of the data [19]. In this data-driven context, the network topology is thus constrained by the data under study. In the context of complex networks however, a key issue concerns the general performance of complex network classes: considering a given data set, do the different complex network topology classes (regular, small-world, scale-free) yield significant differences with respect to performance or robustness?

This paper investigates this issue through an experimental study on the relationship between complex topology following 2D-Watts and Strogatz models and the performance of the corresponding SOMs on a supervised learning problem (handwritten digit classification). The robustness of the results with respect to noise are also addressed. After introducing the context in Section 2, Section 3 is devoted to the *direct problem*, i.e. observing the performances of networks with different topologies. The *inverse problem* is addressed in Section 4: what topology class emerges from the evolutionary optimization of the classification accuracy of a class of networks?

2. METHOD AND EXPERIMENTS

The target neural networks of this study are Self Organizing Maps (SOMs). This model was first described as an artificial neural network by T. Kohonen, so it is sometimes called Kohonen maps [8]. SOMs are usually used for unsupervised learning tasks and produce low-dimensional representations of high-dimensional data, and are thus useful for visualization purposes. In the present work however, SOMs are used for supervised learning tasks, and this section will detail the supervised learning procedure that is used throughout this work: after the standard SOM unsupervised learning a label must be given to each neuron of the network. The classification of an unknown example is then achieved by finding the best matching neuron of the network.

The task considered in this work is the recognition / classification of handwritten digits, using the well-known MNIST database: The MNIST database of handwritten digits [9] has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. SOMs will hence be used here with partly supervised learning, in order to give an unambiguous performance measure and estimate the corresponding topology/performance relation. It should be noted that the goal here is not to reach the best possible performance for the MNIST problem (and indeed SOMs cannot compete with best-to-date published results) but to compare the relative performances of different topologies on the same problem [9].

Each digit in the data base is described by a $M = 28 \times 28$ matrix of pixels (integer gray level in [0,255]). The N neurons of the SOMs are scattered on a $2d$ space. Each neuron i has an associated M-dimensional weight vector $\mathbf{w_i}$ that is initialized randomly and will be adjusted while learning the training set. The different phases of the learning process go as follows.

2.1 Learning

This phase is the classical unsupervised SOMs learning process (for more details, see, again, [8]). At each learning step t, a sample digit $\mathbf{I}(t)$ is uniformly picked up in the learning dataset. For every neuron i, its distance d_i to $\mathbf{I}(t)$ is computed by:

$$d_i = \sum_{j=1}^{M} (I_j - W_{ij})^2$$

The corresponding *Best Matching Unit* (BMU) is the neuron whose weight vector is the closest (in L^2-norm) to $\mathbf{I}(t)$. The weights of the BMU k are updated:

$$\mathbf{w_k}(t+1) = \mathbf{w_k}(t) + \eta(t) \times (\mathbf{I}(t) - \mathbf{w_k}(t)),$$

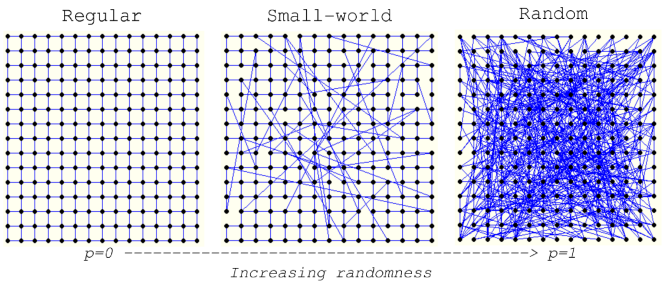

Figure 2: Illustration of 15×15 Kohonen maps with various interconnection topologies, ranging from regular (left), to small-world (center) and regular (right) depending on a rewiring probability p.

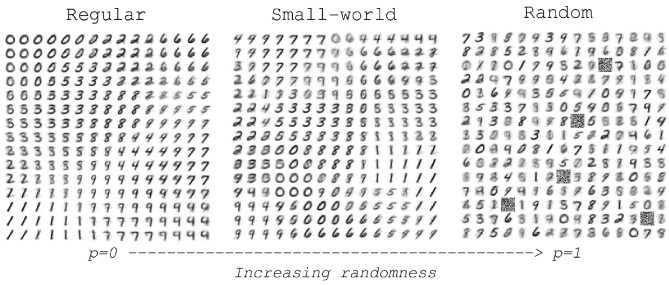

Figure 3: Results of the learning phase using the three topology classes shown in Figure 2. The weight vector $\mathbf{w_i}$ of each neuron i is represented as a 28×28 image, centered on the neuron position in the $2d$ grid. For the random network, some neurons are disconnected from the network, and thus do not learn from the examples during the learning phase. Their weight vectors are thus kept random, yielding the random images on the figure.

where η is a user-defined *learning rate*. The weights of the neighbors of the BMU are updated similarly, but with a learning rate η that decays according to a Gaussian law of the distance with the BMU (the definition of the distance is discussed in next Section). The variance of the Gaussian law is also called the *radius* of the neighborhood.

In the present work, the total number of learning steps was varied between 10^4 and 10^6, and the radius is decreased along learning iterations (see e.g. Figure 4).

2.2 Distance

In the classical SOMs algorithm, the N neurons are regularly scattered on a regular $2d$ grid, that can be quadrangular or hexagonal. Hence the two distances that can be defined between neurons, the Euclidian distance and the graph distance (the minimum number of hops between two neurons following the graph connections), are equivalent. However, when the topology diverts from that of a regular grid (e.g. links are added or suppressed), the situation changes dramatically. Hence, because the goal here is to evaluate the influence of the topology of the network on its learning performance, the distance between two neurons will be their graph distance. In other words, while classical SOMs algorithms use regular grid networks for neuron positions and Euclidean distance for learning, we define the distance between the neurons as the graph distance as given by their complex interconnection network.

Figure 2 illustrates three kinds of interconnection topologies for $2D$ SOMs networks. In analogy with Watts and Strogatz algorithm (fig. 1), neurons are first positioned on a square grid and each neuron is connected to its 4 nearest neighbors on the grid. This defines a regular topology (fig. 2, left). Each link is then rewired with probability p: its destination neuron is changed to a uniformly randomly chosen neuron. Depending on the value of p, the neuron interconnection network thus varies from regular (left) to small-world (center) and totally random (right).

Figure 3 shows the results of the learning phase (described above) using the three topology classes shown in Figure 2. In this figure, the weight vector $\mathbf{w_i}$ of each neuron i is represented as a small 28×28 image, centered on the neuron position in the $2d$ grid. In the regular network, the original 784-dimensional data images of handwritten digits have been projected on the $2d$ map so that the visual proximity between two digits is well rendered by the Euclidean distance between the network nodes. When the rewiring probability p increases, the image representations of the neuron weights become increasingly fuzzy. Furthermore, the visual proximity between two images becomes less and less correlated to the Euclidean distance between the neurons, because it is correlated to their graph distance.

2.3 Labelling

The aim of this phase is to prepare the map obtained from the unsupervised learning phase above for the recognition/classification of handwritten digits, in order to be able to later classify examples without label. This is done by assigning a label to each neuron after the learning phase the following way. The BMU of each example of the training set is computed (see Section 2.1). For each neuron of the network, a probability vector is then computed, describing the different votes of each class for this neuron as BMU. For example, if neuron i is the BMU of 20 (labelled) examples, out of which 16 are labelled "1" and 4 labelled "7", the probabilities attached to this neuron are computed as $p_i(1) = \frac{16}{20}$, and $p_i(7) = \frac{4}{20}$ (and $p_i(l) = 0$ for other values of l). The basic label for the neuron is then defined as the class with higher probability, e.g. "1" in the preceding example. Neurons that never were BMUs are given the basic label of the class from which they are at shortest distance (in L^2 norm).

2.4 Classifying

Using either the probability vector, or the basic label, two strategies for classifying unknown examples can be designed. In both cases, the test example is presented to the network, the distance between the example and each neuron is computed, and the N nearest neurons from the examples are recorded (N is a user-defined parameter).

Majority by numbers: The class given to the unknown example is the basic label most often encountered among the N nearest neurons. Preliminary experiments (for maps of $3,600$ neurons) with N ranging from 1 to 500 showed that the best performances are obtained for $N = 1$.

Majority by probability: Here, the probability vector

p_i attached to each neuron i is used to compute the probability that the test image belongs to class k ($k \in [0, 9]$) using the following equation:

$$P_k = \frac{1}{N} \sum_{i=1}^{N} p_i(k)$$

The test image is given the label of the class with highest probability P_k. For this strategy, preliminary experiments reported that the best performance is obtained with $N \in [1 - 8]$.
The latter strategy is more computationally expensive than the former. Moreover, the same preliminary experiments mentioned above showed that its performance is not significantly better. Hence, all following experiments will use the first strategy ("Majority by numbers") with $N = 1$: the class given to an unknown test example is simply the basic label of its BMU.

The performance (or fitness) F of the network can then be computed as the misclassification error over the whole test set:

$$F = n_{err}/N_{test},$$

where n_{err} is the number of incorrectly classified test examples and N_{test} the size of the test set.

3. DIRECT PROBLEM

The goal of the first experiments is to compare the classification performance of SOMs built on different topologies, namely ranging from regular to random topologies according to the Watts and Strogatz model (see Figure 2). Figure 4-A shows the plots of the classification performance F during the (unsupervised) learning phase for networks of 1024 neurons with regular (rewiring probability $p = 0$, bottom curve) to small-world (intermediate curves) to fully random ($p = 1$, top curve) topologies. The initial learning rate is $\eta(0) = 0.008$ (adjusted after a few preliminary runs) and the total number of learning steps is 10^6. The full MNIST database was used for those experiments, i.e. the size of training set is 60000 and the size of test set is 10000.

3.1 Influence of the radius

First of all, Figure 4-A shows that, at long learning times, the network performance is clearly independent from the topology. This is not surprising since the role of the topology decreases with the radius R. Indeed, the number of neighbors within a radius R of a given neuron increases when the rewiring probability p increases. However, this difference decays as R decreases. Important differences are however obvious at short to intermediate learning times: the more random, the less efficient the network at this time scale. This remark deserves further analysis. Indeed, the performance of these random networks evolves in a piecewise constant fashion. Comparing this evolution to the simultaneous decrease of the neighborhood radius (Figure 4-B) uncovers that performance plateaus are synchronized to radius plateaus.
The more random the network, the lower its Mean Shortest Path. Hence, a possible interpretation is that, for high p values, the influence of a given neuron at short learning times extends over the entire $2d$ space, to almost every other neuron. Thus, at short time scales, almost all neurons are updated each time a new image is presented, which actually forbids any learning in the network. This interpretation is supported by Figure 4-D, where the initial radius is five time smaller than in Figure 4-A, everything else being equal. Here, the differences in short time behaviors observed above have vanished.

3.2 Robustness against noise

Because the long term goal of our studies is to investigate computing architectures involving a large number of computer units, the noise we are interested in studying is noise on the topology, and its impact on the computing performances of the network (here, its classification performance). Noise is hence modelled here by deactivating at each learning step a fraction ν of the neurons (the list of the $N\nu$ deactivated neurons is chosen uniformly for each learning step). All neurons are however considered active for the evaluation phase (Section 2.4).

Figure 4-C shows the performance of the same networks during learning with $\nu = 0.25$ noise level (i.e. 25% of the neurons are insensitive to learning, at each step) and the same large initial radius than in Figure 4-A. The differences between both figures can be explained by looking again at the radius. Clearly, because the deactivated neurons are protected from update, the effect of large radius that is described above is strongly attenuated. In other words, the presence of noise (here random node failures) actually *improves* the performance of these complex random networks at short learning times. That this effect is effectively related to large radius sizes is confirmed by inspection of Figure 4F, which shows that with small initial radius, this 'beneficial' effect of noise is not observed (compare with Figure 4D).

Another result from Figure 4 is that the effects of noise are restricted to short-to-average learning times and almost disappear with long learning times, where the performances of all networks are similar (whatever the topology randomness or initial radius). Hence, for long learning times, the SOMs are robust to neuron failure rates as high as 25%, and this robustness does not seem to depend on their neighborhood topology.

Finally, Figure 5 shows the effects of network size on its performance. Each point is averaged over 11 independent runs. While large SOMs ($N > 2,000$) perform better with regular neighborhood networks, the situation is just the opposite with small ($N < 200$) SOMs, where random networks perform better than regular ones. Small-world topologies are intermediate (not shown). Note however that even for the extreme sizes, the difference of fitness between regular and random topologies, though statistically significant (see caption), remains minute.

4. INVERSE PROBLEM

The inverse problem consists in optimizing the topology in order to minimize the classification error. Evolutionary Algorithms [5] have been chosen for their flexibility and robustness with respect to local minima. However, due to their high computational cost, only SOMs with $N = 100$ neurons could be tested: according to the results of previous section, the best topology among the Watts and Strogatz models for

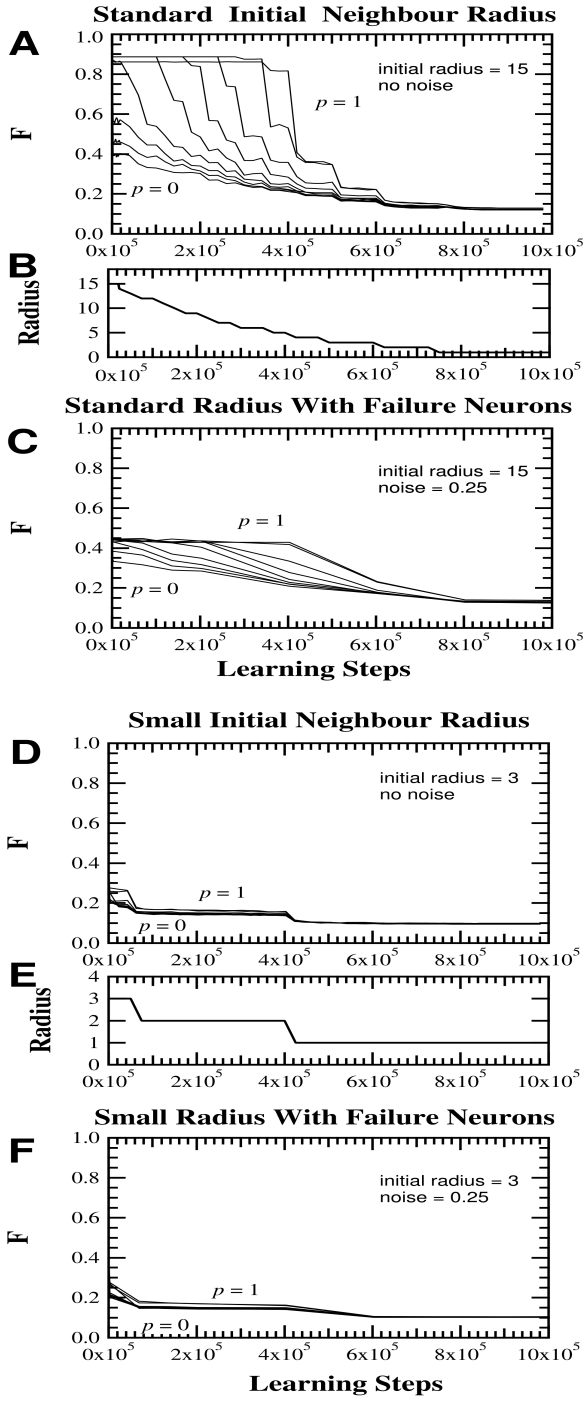

Figure 4: **Evolution of the performance F during learning for SOMs on complex neighborhood networks. Neighborhood networks are constructed positioning neurons on a square grid, and linking each neuron to its 8-nearest neighbors on the grid (Moore neighborhood). Each link is then rewired to a (uniformly) randomly-chosen destination neuron with probability $p = 0, 0.002, 0.004, 0.008, 0.016, 0.032, 0.064, 0.256, 1.000$ (from bottom to top). Panels *A*, *C*, *D* and *F* show the evolution of the fitness F for different values initial radius and noise levels (as indicated on each panels). Panels *B* and *E* display the evolution of the neighborhood radius. Other parameters: map size $N = 1024$ neurons, initial learning rate $\eta(0) = 0.080$, training and test sets of 30,000 and 10,000 examples, respectively.**

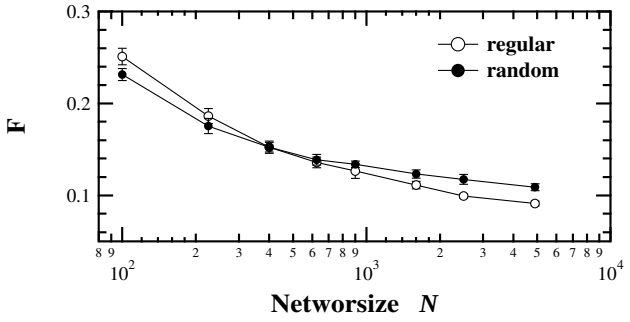

Figure 5: **Performance F vs number of neurons N after 10^6 learning steps for regular (rewiring probability $p = 0$, white circles) or random ($p = 1$, black circles) topologies. Each point is an average over 11 random initial weight and topology realizations. Vertical bars are standard deviations. Stars indicate statistically significant differences (unpaired t-test, $p < 0.010$). Other parameters as in Figure 4-A.**

this size of network and this task is that of a random network (see Figure 5). The purpose of the following experiments is to find out whether optimization will tend to push the topology toward random networks, or if other topologies, outside the Watts and Strogatz model, will appear.

4.1 The Algorithm

Evolutionary Engine: The algorithm used here is a Steady-State Genetic Algorithm with 2-tournament selection and 6-tournament replacement: at each generation, the best of two uniformly drawn networks undergoes variation (see below), and replaces the worse of 6 uniformly drawn networks for the population. The initial population is composed of 100 different small-world networks (obtained with $p = 0.050$).

Variation operators: Evolutionary algorithms typically use two types of variation operators: crossover, involving two or more parents to generate one offspring, and mutation, that uses a single parent to create an offspring. However, because no meaningful crossover operator could be designed here, only mutation was used (no crossover operator is better than a poor crossover operator).

The mutation consists in random rewiring of $C\%$ of uniformly chosen links. C decreases exponentially during evolution ($C(g) = 30\,(102.6)^{-g/g_{max}}$ where g is the generation number and g_{max} is the total number of generations). Here, $g_{max} = 200,000$, and $C(g)$ decreases from 102 ($g = 0$) down to 1 ($g = g_{max}$).

Fitness: The fitness is computed as the average misclassification error F (see Section 2.4) over 5 learning phases, starting from 5 different initial weights.

Each network contains 10×10 neurons, each neuron having 8 neighbors (Moore neighborhood). The initial learning rate is set to $\eta(0) = 0.35$ for a fast learning. However, in order to further decrease the computational time, the learning algorithm is run during only 10000 learning steps (see discussion below), using only 2000 examples from the training

set, and the fitness is computed using 5000 examples from the test set.

4.2 The results

As already mentioned, considering the small size of the SOMs involved, one may expect random networks to perform slightly better than regular ones (Figure 5). The main statistics of the best networks obtained during 9 evolution runs are plotted Figure 6.

The first remark from Figure 6-A is that indeed, the classification error of the best topology in the population decreases along evolution, from 0.355 to ≈ 0.325, i.e. a $> 9\%$ improvement. But the most interesting results can be seen when looking at the characteristics of the best topologies that have emerged during evolution: Figure 6-B shows an important decrease of the Mean Shortest Path, while Figure 6-C demonstrates a clear collapse (more than fourfold reduction) of the Clustering Index. In other words, the topology evolves towards more randomness – as could be expected from Figure 5.

Interestingly, there is another important change in the topology along evolution, concerning the network connectivity distribution. Indeed, the standard deviation σ_k of the connectivity distribution $P(k)$ (where $P(k)$ is the probability that a neuron chosen at random has k neighbors) almost triples during evolution (Figure 6D). This means that the connectivity distribution of the networks broadens (becomes less sharply peaked). In other words, artificial evolution yields more heterogeneous networks. However, it should be kept in mind that this result is highly dependent on the topology of the data themselves (here MNIST database), and could be different with other data.

4.3 Generalization w.r.t. the Learning Process

During the evolution process, the networks were trained during 10000 learning steps at each generation, mainly for computational cost reasons. But how do the evolved networks perform with learning protocols of different lengths (e.g. one million steps)? In order to investigate this generalization ability, the 6 best networks from the initialization phase and the 6 best networks obtained after evolution were trained during respectively 10000 (Figure 7, top) and one million (Figure 7, bottom) learning steps. Note that the results obtained with 10000 learning steps are not a simple zoom-in of the results obtained with one million learning steps, because the radius R decays at different rates in these two cases (as shown in the intermediate plots).

With 10000 learning steps, the fitnesses obtained at the end of the learning phase by the evolved networks are slightly better than those obtained with the initial networks. Surprisingly, this improvement of the fitness is much clearer with one million learning steps. At the end of the learning protocol, the average fitness of the 6 best evolved networks is $> 4\%$ better than that of the 6 best initialized networks (note that this figure is lower than the $> 9\%$ improvement above, because the 12 netwoks were selected form 3 evolution runs only). In the case of one million learning steps, this difference increases up to 8%. Finally, note that, at the end of the learning period, the difference between the two populations is statistically significant ($p < 0.01$, unpaired t-test) for both learning conditions (10000 and one million steps). Hence, the networks selected using 10^4 learning steps also outperform the initial networks for very different learning processes (here 100-times longer). Further investigations are required to better understand this phenomenon.

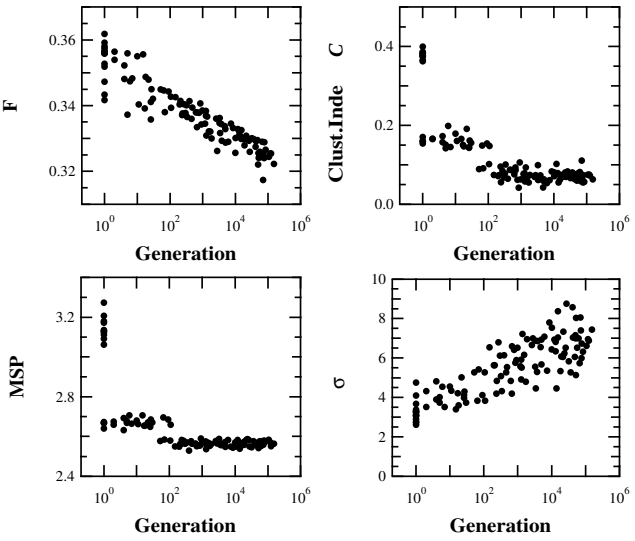

Figure 6: Time courses of the main network statistics during artificial evolution. Each time a mutation gives rise to a topology with a better fitness than the best one in the current population, its fitness (*A*), average mean shortest path (*B*), average clustering index $\langle C \rangle$ (*C*) and the standard deviation of its connectivity distribution σ_k (*D*) are plotted against the current generation number. Each panel groups the results of 9 evolution runs. Parameters: $\eta(0) = 0.35$, fitness estimated as an average over 5 independent runs of 10,000 learning iterations with 2,000 examples from the training set and 5,000 examples from the test set.

5. CONCLUSION

The objective of this paper was to study the influence of topology in a case of neural network defined on a complex topology. On the limited experiments presented here, it seems that the performance of the network is only weakly controlled by its topology. Though only regular, small-world and random topologies, have been presented, similar results have been obtained for scale-free topologies. This suggests that for such learning task, the topology of the network is not crucial.

Interestingly, though, these slight differences can nevertheless be exploited by evolutionary algorithms: after evolution, the networks are more random than the initial small-world topology population. Their connectivity distribution is also more heterogeneous, which may indicate a tendency to evolve toward scale-free topologies. Unfortunately, this assumption can only be tested with large-size networks, for which the shape of the connectivity distribution can unambiguously be determined, but whose artificial evolution, for computation cost reasons, could not be carried out. Similarly, future work will have to address other classical computation problems for neural networks before we are able to draw any general conclusion.

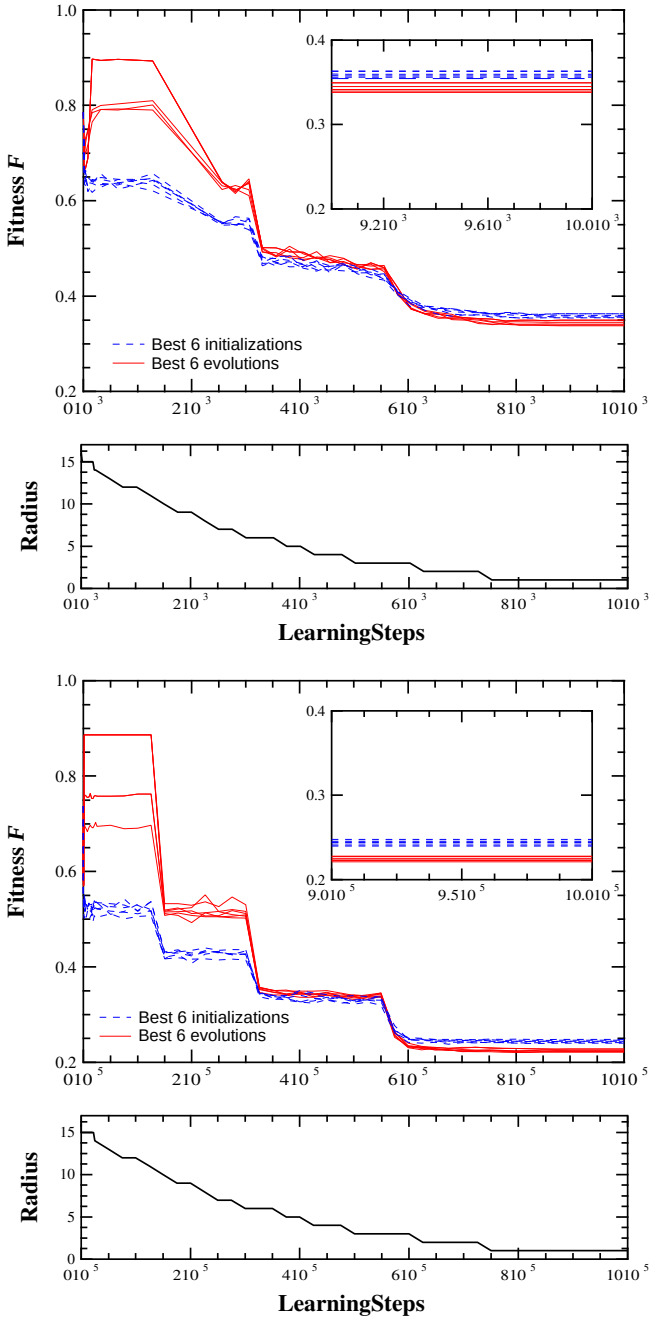

Figure 7: Evolution of the fitness during learning of the 6 best networks from the initialization phase (dashed lines) and the 6 best networks obtained after evolution (full lines). The learning protocol consisted of 10^4 (upper panels) or 10^6 (lower panels) learning steps. The insets show magnified views of the results at the end of the learning phase. The evolution of the neighborhood radius is also given in each case for comparison purposes. Each curve is an average over 11 initial realizations of the neuron weights.

6. REFERENCES

[1] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509, 1999.

[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424:175–308, 2006.

[3] J. Bohland and A. Minai. Efficient associative memory using small-world architecture. *Neurocomputing*, 38–40:489–496, 2001.

[4] Z. Deng and Y. Zhang. Complex systems modeling using scale-free highly-clustered echo state network. In *IJCNN'06*, pages 3128–3135, 2006.

[5] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer Verlag, 2003.

[6] B.J. Kim. Performance of networks of artificial neurons: the role of clustering. *Phys. Rev. E*, 64:045101, 2004.

[7] J. M. Kleinberg. Navigation in a small world. *Nature*, 406(6798), 2000.

[8] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, 1989.

[9] Yann LeCun and Corinna Cortes. The MNIST database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`.

[10] C. Darabos M. Tomassini, M. Giacobini. Evolution and dynamics of small-world cellular automata. *Complex Systems*, 15:261–284, 2005.

[11] P.N. McGraw and M. Menzinger. Topology and computational performance of attractor neural networks. *Phys. Rev. E*, 68:047102, 2003.

[12] L. G. Morelli, G. Abramson, and M. N. Kuperman. Associative memory on a small-world neural network. *Eur. Phys. J. B*, 38:495–500, 2004.

[13] S. Nolfi and D. Parisi. *Handbook of brain theory and neural networks*, chapter Evolution of artificial neural networks, pages 418–421. MIT Press, 2002.

[14] P. Oikonomou and P. Cluzel. Effects of topology on network evolution. *Nature Physics*, 2:532–536, 2006.

[15] D Simard, L Nadeau, and H. Kroger. Fastest learning in small-world neural networks. *Phys. Lett. A*, 336:8–15, 2005.

[16] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[17] D. Stauffer, A. Aharony, L da Fontoura Costa, and J Adler. Efficient Hopfield pattern recognition on a scale-free NN. *Eur. Phys. J. B*, 32:395–399, 2003.

[18] S. H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, 2001.

[19] T. Villmann and E. Merenyi. *Self-Organizing Maps: Recent Advances and Applications*, chapter Extensions and Modifications of the Kohonen-SOM and Applications in Remote Sensing Image Analysis, pages 121–145. Springer-Verlag, 2001.

[20] D.J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[21] X. Yao. Evolving artificial neural networks. *Proc. IEEE*, 87:1423–1447, 1999.