

# Studying the Capacity of Cellular Encoding to generate Feedforward Neural Network Topologies

German Gutierrez<sup>1</sup>, Ines Galvan, J.Molina, Araceli Sanchis Department of Computer Science  
Universidad Carlos III de Madrid  
Leganes, Madrid, Spain 28911  
E-mail: ggutierr@inf.uc3m.es<sup>1</sup>

**Abstract**—Many methods to codify Artificial Neural Networks have been developed to avoid the disadvantages of direct encoding schema, improving the search into the solution's space. A method to analyse how the search space is covered and how are the movements along search process applying genetic operators is needed in order to evaluate the different encoding strategies for Multilayer Perceptrons (MLP). In this paper, the generative capacity, this is how the search space is covered for an indirect scheme based on cellular systems is studied. The capacity of the methods to cover the search space (topologies of MLP space) is compared with the direct encoding scheme.

## I. INTRODUCTION

In the last years, many works have been centered toward automatic resolution of the design of neural networks architecture [1] [2] [3] [4] [5] [6]. Two representation approaches exist to find the optimal net architecture using Genetic Algorithms (GA). One of them based on the complete representation of all the possible nodes and/or connections, Direct Encoding Methods (DEM), where every node and/or connection of the neural network is indicated into the chromosome of the GA [7] [8]. And other based on an indirect and compact representation of the architecture instead of codifying the complete network, called Indirect Encoding Methods (IEM) [1] [9] [10].

IEM are applied in order to reduce the length of the genotype, the search space, and to make the problem more scalable. No exhaustive analysis of those features exists in the literature. The main problem is the definition of an objective measure to evaluate different codifications and search strategies. The measure should be able to evaluate the generative capacity and the efficiency of the search strategy, analyzing the different neural network architectures generated with several methods.

Typically, a good problem representation requires the representation of any possible solution, and a good search strategy requires that similar genotypes produce similar architectures. In this way, the whole search space is covered and the fitness function is able to guide the search.

The main goal of this work is to evaluate and compare different encoding algorithms. In this way, we need to analyze the neural networks generated, to analyze the generative capacity of the method (how many different architectures the method is able to generate) over the whole search space and the search strategy (how the GA generates neural networks).

Hamming distance might be used to show how the neural network space is covered by the encoding scheme. However,

indirect approaches could not be applied because each chromosome and its corresponding binary matrix have a different meaning depending on the expansion method. We proposed an objective measure to evaluate in the same way different methods and the evaluation is made over the space of generated neural networks architecture space. Thus different encoding methods can be compared.

Our interest is focused in the evolution of topologies of multilayer perceptron (MLP) with one hidden layer, i.e., just the number of hidden nodes and connections between different layers. The task of weight training is left to be carried out by the back propagation algorithm. So, only the topology of MLP will be codified into the chromosome.

In this work, the generative capacity of two different encoding methods are studied and compared: a DEM that represents every possible connection of a MLP and an IEM based on Cellular Systems. Using the procedure defined in this paper, we could analyze the generative capacity (representation) and the search space in the domain of genotypes of direct codification scheme and compare with the indirect one. An objective assessment and the results concerning to the generative capacity (ability to cover the MLP architecture space) of both encoding methods are shown. A random initial population of chromosomes will be created for both encoding methods, the expansion process will be applied to each chromosome for the IEM and conclusions obtained from the histograms of neural networks achieve will be exposed.

Section II and III are related with the direct and indirect encoding schemes, respectively, used in this work. Section IV describes the objective assessment. The generative capacity of the encoding schemes is shown in section V. Finally, some conclusions are presented in section VI

## II. DIRECT ENCODING METHOD

In direct encoding schema the most usual way to codify the topology is to place the chromosome (a string) as the concatenation of rows (or columns) of the binary connections matrix, where each cell represents the existence or absence of a set connection between two nodes. Since we are interested only in MLP architectures with one hidden layer, rejecting any other type of neural network, as recurrent neural networks, the direct encoding scheme used in this work only represents useful connections, connections between nodes of one layer and immediately next layer. Thus, the length of the chromosome is reduced.

The direct encoding scheme used in this work is based on a binary matrix of dimension  $DimX \times DimY$ , where  $DimX$  is equal to the number of input neurons plus the number of output neurons and  $DimY$  corresponds with the maximum number of hidden neurons to be considered (see 1). To relate that matrix with an architecture of a MLP with one hidden layer, the meaning of the position  $(i, j)$  is defined as follows. Let's  $n$  the number of input neurons; if  $i = n$  then  $(i, j)$  represents a connection between the  $i - th$  input neuron and the  $j - th$  hidden neuron; if  $i > n$ ,  $(i, j)$  represents a connection between the  $j - th$  hidden neuron and the  $(i - n) - th$  output neuron. That relation is shown in Fig 1. The chromosome is just the concatenation of the matrix rows and  $DimX \times DimY$  gives the length of the chromosome.

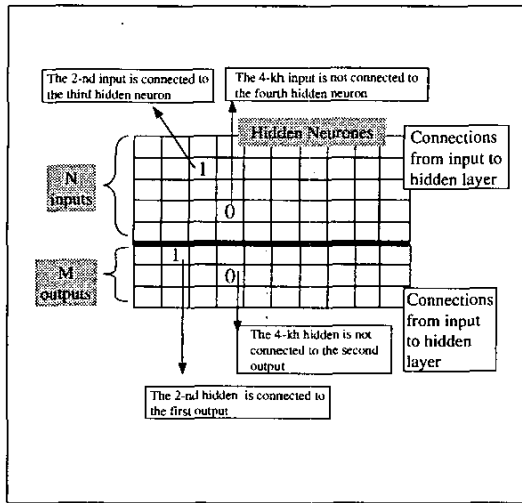


Fig. 1. Matrix of connections

### III. METHOD BASED ON CELLULAR AUTOMATA INDIRECT ENCODING METHOD

DEM of neural network architectures are relatively simple, straightforward to implement and they would be suitable for problems requiring a small number of neurones and connections. They may allow quick generation and optimization of tightly pruned designs. However, the capabilities of direct encoding for solving large problems are limited, because large problems require much larger chromosome. In order to reduce the length of the genotype and to make the problem more scalable, indirect encoding schemes to represent neural network architectures have been proposed in the last years [1] [9] [10]. They basically consist of codifying, not the complete network, but a compact representation of it. That representation (for instance based on grammars, fractals, etc.) is expanded to obtain neural network architecture. The genetic algorithm evolves that representation, instead of connections matrix of neural networks as in the case of direct encoding.

The IEM used in this work is based on cellular automata. The system -shown in figure 2- is composed of three different modules: the Genetic Algorithm Module, the Cellular Module and the Neural Network module. The scheme is cycled, and

each cycle corresponds to a genetic algorithm generation. By means of genetic operators a population of MLP architectures (individuals expressed through cellular automata) is obtained, except the first one, that is randomly generated as is usual. These individuals are evaluated through a fitness function. To calculate the fitness value, the cellular automata are executed over an initial configuration codified in the individual chromosome, and a binary matrix is obtained. In a next step, this matrix is translated into a MLP architecture and is trained. After the MLP is trained, is tested and an error value is obtained. With this error and some other relevant information about the MLP (size, learning cycles, etc.), the fitness value of the considered individual is computed. The process is repeated until all the population is evaluated. The proposed architecture follows the general mechanism of other well-known systems like Kitano's [9] and the GANET system [14].

The cellular module is based on cellular automata philosophy and is composed of two bi-dimensional cellular subsystems. Several seeds give the initial configurations of the two-dimensional cellular subsystems and the rules of these subsystems are applied to generate final configurations. The first cellular subsystem, called *growing cellular system*, is related with the generation of MLP with a large number of connections. The second cellular system, named *decreasing one*, is related with the reduction of this number of connections. The seeds are defined through two co-ordinates and they are codified in the chromosome of the genetic algorithm, which evolves to maximize the performance of MLP architecture. A detailed description of the system can be founded in [13]. The motivation of this approach is based on the idea that only a few seeds for initial configurations of cellular systems can produce a wide variety of MLP architectures.

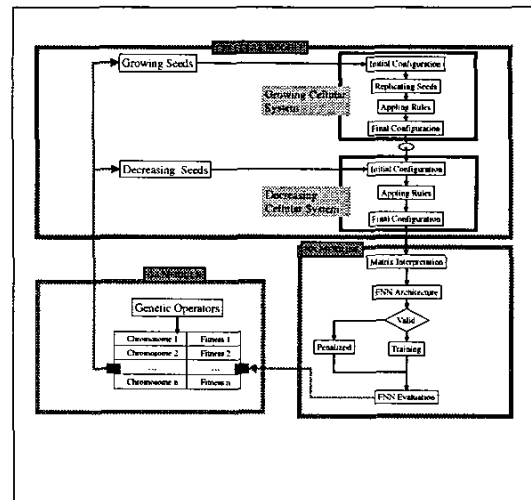


Fig. 2. Modules of IEM based in cellular systems

For cellular encoding method the length of the chromosome is reduced, because for cellular encoding method it depend on the number of growing seeds plus the number of decreasing seeds (eq 1), instead of the maximum number of connections, as for direct encoding method (eq 2)

$$\text{length chrom cellular} = 3 * (gs + ds) \quad (1)$$

$$\text{length chrom direct} = (\text{inputs} + \text{output})^3 \quad (2)$$

#### IV. GENERATIVE CAPACITY

This work concerns with the evaluation of different encoding algorithms studying the generative capacity (how many different architectures the method is able to generate) over the whole MLP topologies space and the search strategy (how the GA generates neural networks).

In order to analyze the behavior of the different encoding methods, two kind of information have to be studied:

1) *Ability to generate an initial population:* (point one). The first one is related with the ability of the methods to generate an initial population of chromosomes covering the complete MLP with one hidden layer neural network space. As the initial population has a finite size, the individuals are a subset of the search space. Clearly, an algorithm that randomly generates chromosomes following an uniform distribution over the search space is better than algorithms that introduce a bias in the initial population of chromosomes.

2) *The way the algorithm searches:* (point two). The second one is related with the way the algorithm searches in the search space. The best situation to apply GA is that small changes in the genotype produces small changes in the fitness value. This situation avoids the epistasis problem [12]. This problem could not be addressed directly (as in other genetic approximations) with the indirect encoding methods because of the particular process introduced to translate the genotype in to the neural network.

We propose to approximate the points one and two analyzing the kind of MLP with one hidden layer obtained, for DEM and IEM based on cellular systems. The experiments made in this work show information relative to the ability to generate an initial population, for each encoding method, that covers the MLP topologies space widely, (point one). The study involves the following steps:

- 1) Randomly generate a initial population of chromosomes with a uniform distribution.
- 2) Apply the translation process for each encoding method and for each chromosome to obtain the correspondent binary matrix or neural network topology.
- 3) Represent the percentage of connections for each MLP obtained with the same number of hidden nodes.

In addition, this initial population could facilitate the search process. If a set of chromosomes, generated by means of a uniform distribution, gives rise to a set of individuals that embrace a great extent of topologies space (phenotype), then it does not generate a restricted initial population of topologies. And any point (any topology) of the space will be reach easily.

#### V. EXPERIMENTAL RESULTS

In these work a very simple but interesting domain is used. This domain, called *minimum interesting coding problem*, is

define by four binary inputs and two binary outputs. Then the maximum number of hidden neurons of MLP considered is  $(\text{inputs} + \text{outputs})^2 = (4 + 2)^2 = 36$ . The function that establish the relation between outputs and inputs is known and it is described in equation 3.

$$X_1, X_2, X_3, X_4 \in \{0, 1\}; Y_1 = X_3, Y_2 = (X_3 \text{ xor } X_4) \quad (3)$$

In order to study the generative capacity of the initial population,  $N$  chromosomes are randomly generated with a uniform distribution. The corresponding phenotype (topologies of MLP with one hidden layer) is obtained for each chromosome. Exactly,  $N$  binary matrixes of connections are obtained. The question that arises in this point is: whatever of them are different, whichever functionally different topologies are obtained? This is not at all a trivial task.

At this point a mechanism of representing the obtained matrixes is required. In [14] is shown a measurement that allows us to establish an injective relation ( $f(a) = f(b) \Rightarrow a = b$ ) between the set of all the possible matrix of connections and  $R$ . Nevertheless, this measure does not give to a clear idea of what type of networks is obtained, that is to say, how many hidden nodes and how many connections have the topology generated from a chromosome.

A measurement that indicates what type of networks are being obtained, i.e. how many hidden nodes have and which is the percentage of connections, is needed.

Therefore, to be able to observe the kind of MLP obtained, all of them are represented in a graph. Each one of the points represents a network, where  $x$ -axis indicates the number of hidden nodes and the  $y$ -axis the percentage of connections of this network.

For DEM, the initial population of MLP obtained from  $10^4$  chromosomes generated with a uniform distribution, is shown in fig 3

One of the parameters of the system of codification based on cellular systems is the number of seeds, growing seeds ( $gs$ ) and decreasing seeds ( $ds$ ). All possible combinations from 1 to 10 of growing and decreasing seeds have been considered, but only a few of them are shown in the figures 4, 5, 6, 7.

$10^4$  chromosomes are generated, by means of a uniform distribution. Each of them codifies the positions, within the grid, of  $X$   $gs$  and  $ds$ , where  $1 < X < 10$  and  $1 < Y < 10$ . Each MLP with one hidden layer is a point into the figure, where  $x$ -axis indicates the number of hidden nodes and the axis and the percentage of connections of this network, as it was previously said.

In the case of DEM, a uniform distribution in the chromosomes space does not produce a distribution that completely covers the MLP (with one hidden layer) space. But in the IEM based on Cellular Systems, besides of reducing the chromosomes size, a uniform distribution in the chromosomes space is translated into a MLP space that widely covers the phenotype space.

#### VI. CONCLUSIONS

As an Evolutionary Algorithm begins the search with a randomly generated initial population, it is important that it

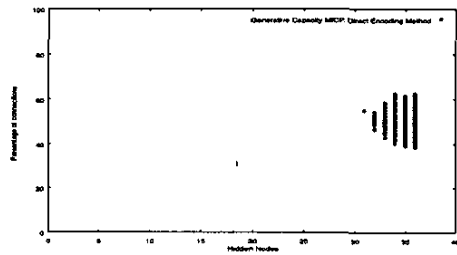


Fig. 3. MLP with one hidden layer. Direct encoding

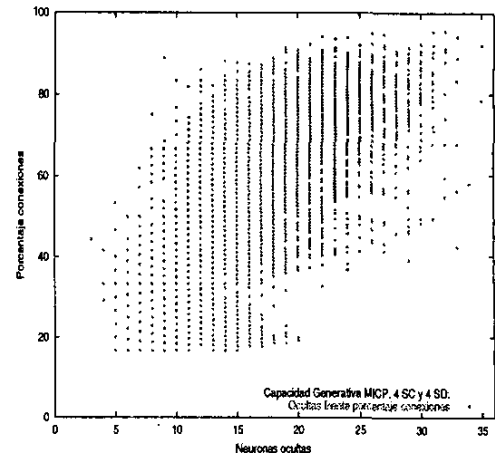


Fig. 6. MLP with one hidden layer. Cellular encoding,  $gs=4, ds=4$

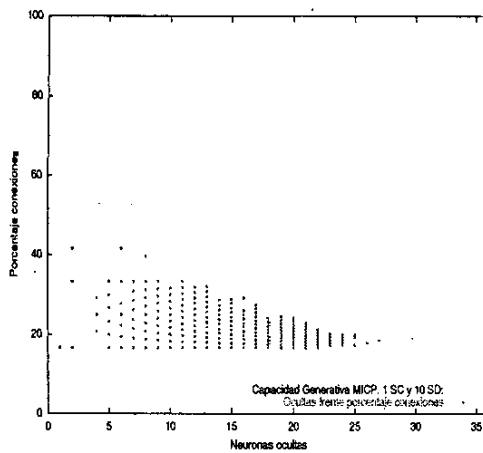


Fig. 4. MLP with one hidden layer. Cellular encoding,  $gs=1, ds=10$

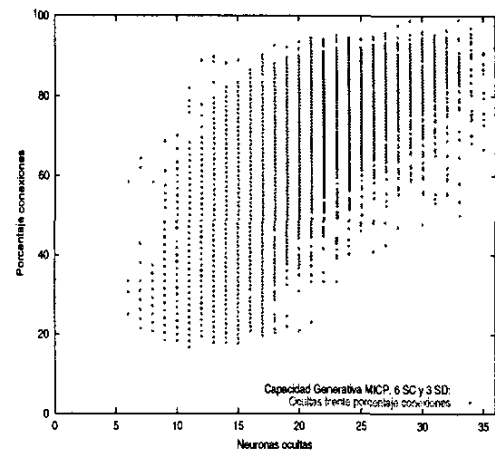


Fig. 7. MLP with one hidden layer. Cellular encoding,  $gs=6, ds=3$

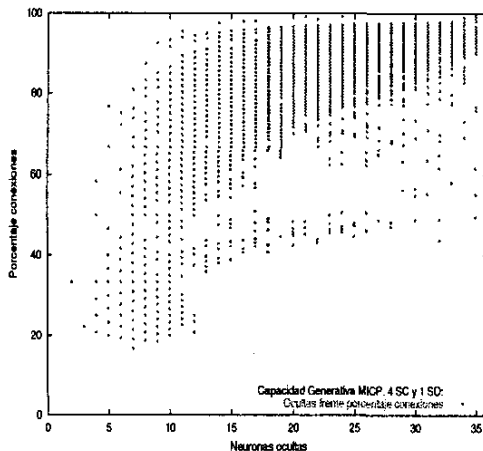


Fig. 5. MLP with one hidden layer. Cellular encoding,  $gs=4, ds=1$

covers as widely as possible the space of topologies. The analysis presented in this work shows that direct encoding methods do not carry out this condition, by the opposite than the indirect encoding scheme based on Cellular Systems aforesaid.

In future works, random walks, applying the genetics operators, over the space of chromosomes will be evaluated in order to get to know how the genetic algorithm looks for in the search space (MLP space) for both encoding methods, direct and indirect.

## REFERENCES

- [1] S. Harp, Samad T. and Guha A. *Towards the Genetic Synthesis of Neural Networks*. Proceedings of the Third International Conference on Genetic Algorithms and their applications, pp. 360-369, San Mateo, CA, USA, 1989.
- [2] G.F. Miller, P.M. Todd and S.U. Hegde. *Designing neural networks using genetic algorithms*. In Proc. of the third international conference on genetic algorithms and their applications, pp 379-384, San Mateo, CA, USA, 1989.
- [3] S. Harp, Samad T. and Guha A. *Designing Application-Specific Neural Networks using the Genetic Algorithm*. Advances in Neural Information Processing Systems, vol2, 447-454, 1990.

- [4] F. Gruau. *Genetic Synthesis of Boolean Neural Networks with a Cell Rewriting Developmental Process*. Proc. of COGANN-92 International Workshop on Combinations of Genetic Algorithms and Neural Networks, pp. 55-74, IEEE Computer Society Press, 1992.
- [5] F. Gruau. *Neural Network Synthesis Using Cellular Encoding and the Genetic Algorithm*. Ph.D. Thesis, Ecole Normale Supérieure de Lyon, (1994).
- [6] F. Gruau. *Automatic Definition of Modular Neural Networks*. Adaptive Behavior, vol. 2, 3, 151-183, 1995.
- [7] T. Ash. *Dynamic Node Creation in Backpropagation Networks* ICS Report 8901, The Institute for Cognitive Science, University of California, San Diego (Saiensu-sh, 1988), 1988.
- [8] D.B. Fogel, Fogel L.J. and Porto V.W. *Evolving Neural Network*, Biological Cybernetics, 63, 487-493, 1990.
- [9] H. Kitano. *Designing Neural Networks using Genetic Algorithms with Graph Generation System*, Complex Systems, 4, 461-476, 1990.
- [10] J.W.L. Merrill and R.F. Port. *Fractally configured Neural Networks*. Neural Networks, 4, 53-60, 1991.
- [11] G. Gutierrez, J.M. Molina, I. Galvn, A. Sanchis. *An Objective Measure to Compare some Automatic Generation Methods of NN Architectures*. IEEE International Conference on Systems, Man and Cybernetics, 2002. Tunisia.
- [12] Colin R. Reeves, Christine C. Wright. *Epistasis in Genetic Algorithms: An Experimental Design Perspective*. ICGA 1995: 217-224
- [13] G. Gutierrez, P. Isasi, J.M. Molina, A. Sanchis and I. M. Galvn. *Evolutionary Cellular Configurations for Designing Feed-Forward Neural Networks Architectures*. Connectionist Models of neurons, Learning Processes and Artificial Intelligence. 6th International Work-Conference on Artificial Neural Networks, IWANN 2001. Proceedings, Part I. LNCS 2084. J. Mira, A. Prieto (Eds.) Springer.
- [14] M. A. Guinea, G. Gutierrez, I. Galvn, A. Sanchis, J. M. Molina. *Generative Capacities of Grammars Codification for Evolution of NN Architectures*. International Conference on Evolutionary Computation, WCCI, 2002. USA.