Studying the Capacity of Grammatical Encoding to Generate FNN Architectures

Germán Gutiérrez, Beatriz García, José M. Molina, Araceli Sanchis

SCALAB. Departamento de Informática. Universidad Carlos III de Madrid.
Avda. Universidad 30, 28911-Leganés (Madrid)-SPAIN.
{ggutierr, masm}@inf.uc3m.es,molina@ia.uc3m.es,100027154@alumnos.uc3m.es

Abstract: Many methods to codify Artificial Neural Networks have been developed to avoid the defects of direct encoding schema, improving the search into the solution's space. A method to estimate 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 Feedforward Neural Networks. A first step of this method is considered with two encoding strategies, a direct encoding method and an indirect encoding scheme based on graph grammars: generative capacity, how many different architectures the method is able to generate.

1. 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 optimum net architecture using Genetic Algorithms (GA). One 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, all the search space is covered and the fitness function is able to guide the search.

The goal is to evaluate and compared different encoding algorithms. In this way, we need to estimate 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 Genetic Algorithm 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 generated neural networks architecture space. Thus different encoding methods can be compared.

Our interest is focused in the evolution of architectures of feedforward neural networks (FNN) with one hidden layer, 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 FNN 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 FNN and an IEM based on Graph Grammars with a expansion process to obtain the binary matrix that represent the FNN architecture. 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 FNN architecture space) of both encoding methods proposed 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. In [11], a preliminary study of the representation capacity of another IEM based on Cellular Automata [12] is presented using that objective measure. The results showed that the IEM based on cellular approach is able to cover the search space.

Section 2 and 3 is related with the direct and indirect encoding schemes, respectively, used in this work. Section 4 describes the objective assessment. The generative capacity of the encoding schemes is shown in section 5. Finally, some conclusions are presented in section 6

2. Direct Encoding Method

In direct encoding schema the most usual way to codify the architecture 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 FFN 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 Dim_x x Dim_y, where Dim_x is equal to the number of input neurons plus the number of output neurons and Dim_v corresponds with the maximum number of hidden neurons to be considered (see Fig 1). To relate that matrix with an architecture of a FNN with one hidden layer, the meaning for the grid position (i,j) is defined as follows. Let's n the number of input neurons; if $i \leq n$ 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 Dim_x x Dim_y gives the length of the chromosome.

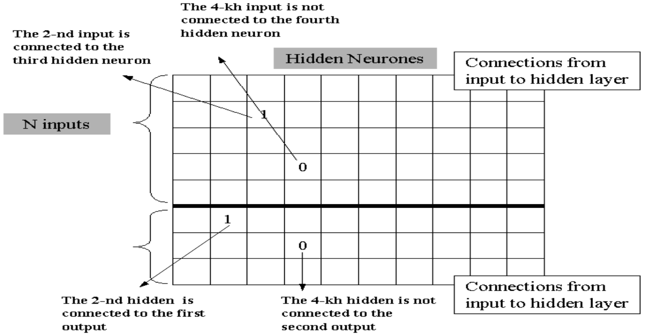

The 4-kh input is not connected to the fourth hidden neuron
The 2-nd input is connected to the third hidden neuron
Hidden Neurones
Connections from input to hidden layer
N inputs
1
0
1
0
The 2-nd hidden is connected to the first output
The 4-kh hidden is not connected to the second output
Connections from input to hidden layer

Fig. 1. Matrix connection for direct encoding method.

3. Method based on Graph Grammar/Indirect Encoding Method

In 1990, Kitano [9] presents a new method for designing neural networks by means of Genetic Algorithms, where neural networks are represented through graph grammars, codified in the chromosomes of the individuals. The selected indirect encoding method is based on an improvement of Graph Grammars used by Kitano.[13]

Whole system is composed of three modules, Genetic Algorithm Module, Grammar Module and Neural Network Module. In [14] a detailed description of Neural Network and Genetic Algorithm Modules can be found. The Grammar Module takes charge of generating FNN architecture from a Graph Grammar indicated into the chromosome. The generated FNN is trained and relevant information about the goodness of FNN is used as the fitness value for the genetic algorithm module. And finally the genetic algorithm module takes charge of generates Graph Grammars of next generation.

Into each chromosome a bi-dimensional Graph Grammar is described and a bi-dimensional word, a binary matrix, is obtained, in a deterministic way, from the bi-dimensional Graph Grammar [13]. The binary matrix obtained represents a connection matrix, where each element a_{xy} represents the existence *($a_{xy}=1$)* or the absence ($a_{xy}=0$) of connection from node labelled x to node labelled y.

Since the goal in this paper is to compare the generative capacity of both methods, the binary matrix, the word, obtained from the Graph Grammar codified into a chromosome has to be translated into the matrix connection described in section 2. Every backward connection or every direct link from a input node to a output node is not considered on the translation process: Only connections corresponding to a link from input nodes to a hidden node or a link from hidden node to output node are hold.

4. Generative Capacity

The goal is to evaluate different encoding algorithms studying the generative capacity (how many different architectures the method is able to generate) over the whole FNN architecture space and the search strategy (how the Genetic Algorithm generates neural networks).

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

The first one is related with the ability of the methods to generate an initial population of chromosome covering the complete FNN space. As the initial population has a finite size, the individuals are a subset of the search space. Clearly, an algorithm that generates random individuals following an uniform distribution over the search space is better than algorithms that introduce a bias in the initial population.

The second one is related with the way the algorithm searches in the search space. The best situation to apply genetic algorithm is that small changes in the genotype produce small changes in the fitness value. This situation avoids the epistasis problem. 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 number of hidden neurons (*nh*) and the number of connections (*nc*). We propose to calculate the histogram of *nh* and *nc* for the initial population (point one), and to evaluate random walks (representing in a 3-D graph the fitness landscape over *nh* and *nc*) for the search (point two).

In this work, results concerning to the first point are shown. The study involves the following steps:

1. Generate a random 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 architecture.
3. Make a histogram of the number of hidden neurons (*nh*) in the binary matrix.
4. Make a histogram of connections (*nc*) in the binary matrix.
5. 3-D histogram considering the bidimensional space *nh x nc*.
6. Represent the percentage of connections for each FNN obtained with the same number of hidden nodes

5. Experimental Results

In order to analyze the generative capacity of both encoding methods, a initial population of 10^4 chromosomes is randomly generated for each encoding method. In both schemes every value for an element of the chromosome has the same probability than the other values. With direct encoding method, in any element of the binary matrix (the chromosome) there is a “0” or a “1” with the same probability (1/2). For IEM based on graph grammars the chromosome codifies a bidimensional grammar. Into the chromosome, the production rules (the right side of the production rules), the expansion method and the level of recursion are codified in binary. Each allele has the same probability (1/2) to be a “0” or a “1”.

In these results a generic domain is used, with 2 inputs and 2outputs ($n_I = n_O = 2$). Then the maximum number of hidden neurons of FNN considered for DEM is 16 ($(n_I+n_O)^2$), [11]. Therefore the length of the chromosome for direct encoding scheme is 64, equal to the maximum number of connection considered.

In case of IEM, the length of the chromosome is fixed and not depends on the problem. The maximum number of hidden nodes and connections depends on the size of the binary matrix immediately obtained from the derivation process for each chromosome. The size of that matrix depends on which rules are codified in the chromosome and especially in the value of two parameters that stop the derivation process [13].

For both encoding schemes if in the binary matrix the elements of a column are 0, it means that the hidden node that this column represents has not any connection from input layer or to output layer. In this case, it could be eliminated from the FNN.

For grammatical approach only the results within range 0 to 16 of hidden nodes and 0 to 64 of connections are exposed to compare with results of DEM, that are limited to these values for the size of FNN obtained. But bigger FNNs are obtained for the IEM.

Fig 2 (a) (b) show FNN with the same number of hidden nodes obtained from the initial population of chromosome. FNN architectures with the same number of connections are shown if Fig 3 (a) (b) for both direct and indirect encoding scheme.

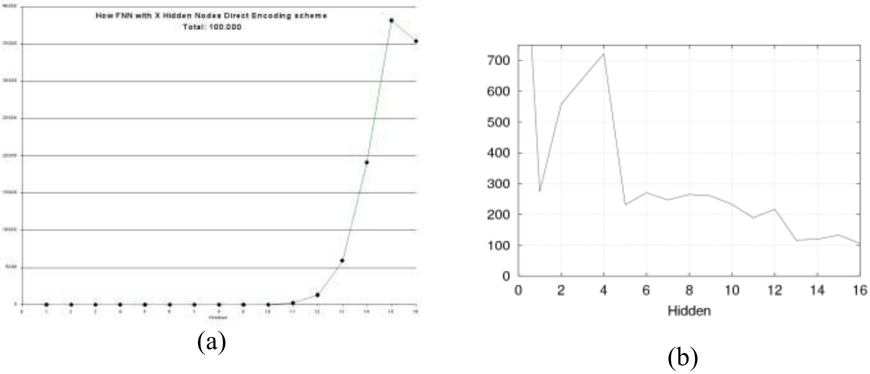
In Fig 4 (a) and Figure 4 (b) the 3-D histograms for both encoding schema is shown. In them, the number of FNN with a same number of hidden nodes and a same number of connections is displayed.

In Fig 5 (a) (b) the percentage of connectivity for FNN with the same number of hidden nodes for both encoding schemes are shown. The maximum number of connections for a given number of hidden nodes (h) is $(h\times(n_I+n_O))$, the minimum number of connection is (n_I+n_O)

In Fig 2 (a) and Fig 3 (a) is observed that for direct encoding scheme the probability density function, corresponding to this discrete distribution, could be obtained exactly from the probability that a column has no element equal to “1” and an element of matrix connection (the chromosome in DEM) is equal to “1”. It is clear that must be a binomial distribution. But for indirect encoding (see Fig 2(b) and 3(b)) it is not feasible, at least too complex, to obtain exactly the discrete probability density function that corresponding to the histogram of *nc* and *nh*.

Besides, in Fig 5 (a) (b) can be observed that the percentages of connectivity of FNN obtained with both encoding schemes. EM are limited to some values (35 - 75 %). By opposite, for IEM (see Fig 5 (b)) FNN with percentages of connectivity ranging from minimum, $\left(100/_{n_I+n_O}\right)$, to maximum are obtained.

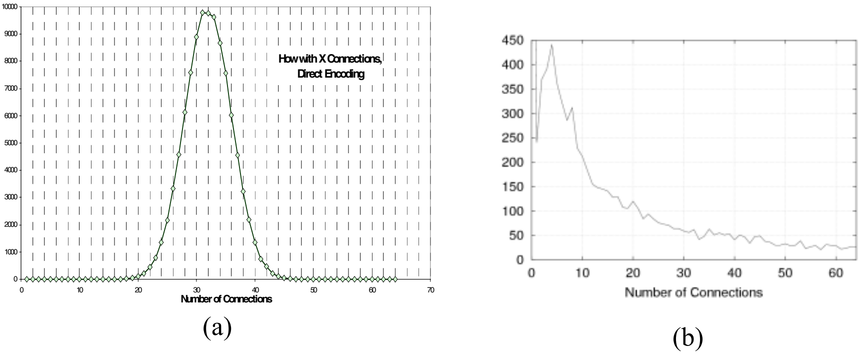
All these advantages for IEM, considering a initial set of chromosomes randomly generated with a uniform distribution, could make the search more feasible for the evolutionary algorithm.



(a)

(b)

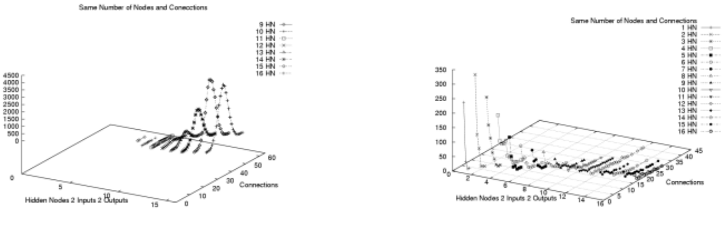
Fig. 2.. FFN have the same number of hidden nodes. DEM (a), IEM (b).



(a)

(b)

Fig. 3. How many FNN have the same number of connections. DEM (a), IEM (b).

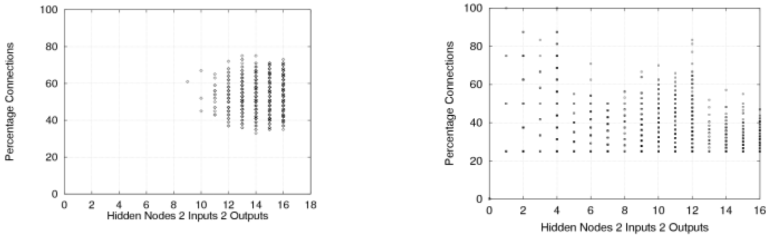


(a) (b)

Fig. 4. FNN have the same number of hidden nodes and connections. DEM (a), IEM (b).

As it is observed in all figures, it is clear that indirect encoding scheme cover the FNN space for a initial random population of chromosomes better than the DEM. With DEM no FNN with less than 10 hidden nodes and only FNN with a number of connections ranging between 18 and 47 are obtained. Therefore only a limited subset of space of FNN is obtained from a uniform distribution of chromosome space.

On the contrary, with IEM and considering only the subset of FNN space with less or equal than 16 hidden nodes and 64 connections, the space of FNN is covered. Even though FNN with few hidden nodes and connections are majority obtained.



(a) (b)

Fig. 5. Percentage of connectivity. DEM (a), IEM (b).

6. Conclusions

Indirect Encoding methods are applied in order to reduce the length of chromosomes. As a Evolutionary Algorithm begins the search with a random initial population, it is important that it covers as completely as possible the search space. The analysis presented in this work shows that direct encoding methods do not carry out this condition, on the contrary than the indirect encoding scheme based on grammars aforesaid.

In future works, estimations of what density distribution functions correspond with the histograms shown will be study. Random walks over the space of chromosomes will be evaluated in order to get to know how the genetic algorithm look for in the search 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, 1990.

[5] F. Gruau. "Neural Network Synthesis Using Cellular Encoding and the Genetic Algorithm". Ph.D. Thesis, Ecole Normale Supèrieure 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. Merril and R.F. Port. Fractally configured Neural Networks. Neural Networks, 4, 53-60, 1991.

[11] G. Gutierrez, J.M. Molina, I. Galván, A. Sanchis. An Objetive Measure to Compare some Automatic Generation Methodsof NN Architectures. IEEE International Conference on Systems, Man and Cybernetics, 2002. Tunisia.

[12]] G. Gutierrez, I. M.Galván, A. Sanchis, J. M. Molina. Generative Capacities of Cellular Auomata Codification for Evolution of NN Codification. International Conference on Artificial Neural Networks 2002 (ICANN 2002). LNCS 2415 Jose R. Dooronsoro (Ed.) Springer.

[13] M. A. Guinea, G. Gutierrez, I. Galván, A. Sanchis, J. M. Molina. Generative Capacities of Grammars Codification for Evolution of NN Architectures. International Conference on Evolutionary Computation, WCCI, 2002. USA.

[14] G. Gutiérrez, P. Isasi, J.M. Molina, A. Sanchís and I. M. Galván. 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.