Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium, Volume 22, No. 1, ISSN 1726-9679
ISBN 978-3-901509-83-4, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2011
Make Harmony between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals & Proceedings of DAAAM International 2011

CELLULAR GRAPH GRAMMAR ENCODING OF NEURAL NETWORKS

FEKIAC, J[ozef]; KLIMKOVA, E[va] & PAVLECH, M[ichal]

Abstract: *This paper presents a novel approach to neural network evolution. Genomes used in this evolution are based on a well-described cellular encoding and a general mathematical idea of graph grammar. These genomes are then optimized with genetic programming. This approach has been tested on the Iris flower data set to classify three Iris flower species according to four measured quantities of each plant. Disadvantages and possible future improvements of this approach are listed in the end of the paper.*

Key words: *neural network, network encoding, evolutionary computation, genetic programming, topology optimization*

1. INTRODUCTION

Neural networks are important tools in data processing, automated control, classification, function approximation and more advanced artificial intelligence tasks.

There are quite a lot of neural network topologies designed for specific family of tasks, e.g. feedforward and RBF networks for approximation and classification, recurrent architectures for associative memory and prediction, network ensembles that combine various tasks into one large network. It is quite clear the task of designing such a network is a very intense task and the final network will be probably redundant.

An evolutionary algorithm can be used to create an untypical network topology optimized to solve one particular complex task. When deciding which evolutionary technique to use, the first task is the selection of an appropriate encoding scheme as described in (Fekiač et al., 2011). Then the choice of a heuristic optimization algorithm results from the structure of the encoded network genotype.

2. NETWORK TOPOLOGY OPTIMIZATION WITH EVOLUTIONARY ALGORITHMS

The most elementary group consists of the direct encoding methods, which make a direct relation between network components (neurons, synapses) and the genotype, i.e. genotype is a full mathematical representation of a graph (Koza & Rice, 1991; Schiffmann, 2000; Stanley & Miikkulainen 2002).

Another encoding method is parametric encoding, which is a higher abstraction of a network, but its use is more easy than direct encoding. Its basic principle is to use a well-known neural network topology and use the evolutionary algorithm only to optimize the network parameters. Mostly the number of layers and number of neurons per layer in a multilayer feed-forward network is optimized this way (Mandischer, 1993).

The most complicated type of encoding is indirect (usually growth) encoding, which enables the creation of complex modular networks with self-similar parts, that are bigger and more complex than the genotype in which they are encoded. In these types of encoding, the herangr of the genotype is usually repeatedly applied to a temporal network representation until the full network is unfolded (Kitano, 1990; Gruau, 1994; Luerssen, 2009).

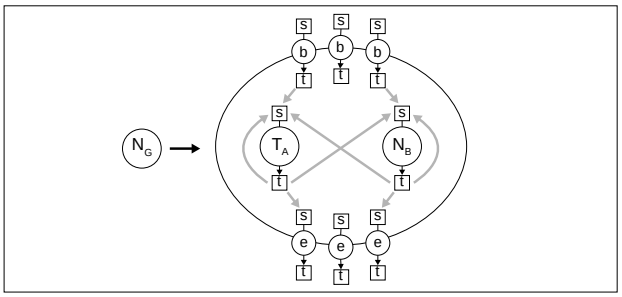

Fig. 1. Cellular graph grammar production rule

3. CELLULAR GRAPH GRAMMAR TREE ENCODING

In this research we employed an encoding based on cellular graph grammars and combined with genetic programming. To enable the use of genetic programming paradigms (Koza, 1990), the graph grammar in the genome has been stored in a form of grammar tree (Fig. 2) instead of production rules (Fig. 1). This approach also avoids the competing conventions problem cerated by the independent order of production rules.

The network is created by starting with the rood production rule connected to the inputs and outputs according to soft label matching and then is repeatedly rewritten by descendant production rules until it contains only terminals. Details of the network derivation are described in (Luerssen, 2009).

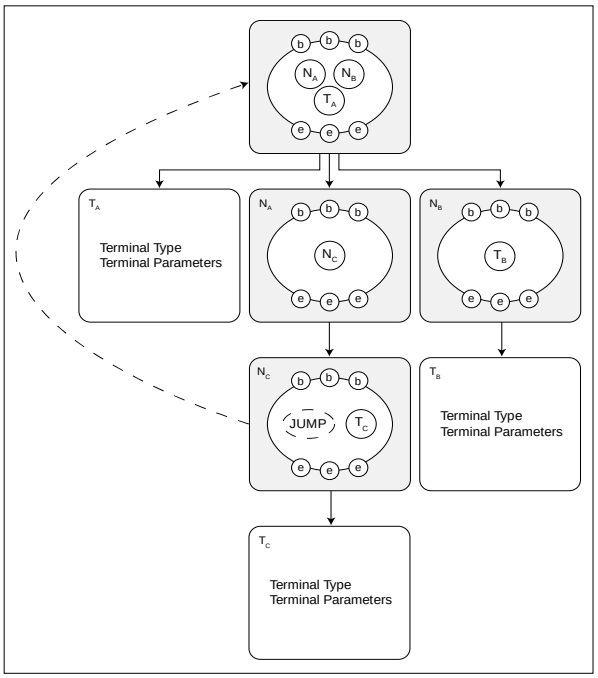

Fig. 2. Graph grammar of a network geotype shown as a grammar tree with recursion

Number of runs	10
Iterations per run	100
Population size	200
Training (approximation) data set size	75
Training (Winner-Takes-All) data set size	36
Test Winner-Takes-All data set size	37
Attribute (input neurons) count	4
Categories (output neurons) count	3

Tab. 1. Simulation settings

4. EXPERIMENT

The proof-of-concept experiment has been run with parameters specified in the table Tab. 1. The Iris flower dataset as collected by (Fisher, 1936) has been used as a testing problem. This dataset consists of 150 records, which of them containing four measured attributes of corresponding Iris flower and its assignment to three different species.

The task of the experiment was to find a neural network capable of the classification into these three groups according to four given input attributes.

There were seven different neuron activation functions available to choose by the algorithm: linear, sigmoid, threshold, tanh, signum, gaussian and sombrero. Output neurons were always linear. Weights of the connections between these neurons were determined by the distance of labels that were responsible for the connection.

The results of the experiment can be seen in Tab. 2. As can be seen in the table, the best of the runs reached a 5% error on the test dataset. The network topology found by this run of the experiment is shown in Fig. 3 with appropriate neuron activation functions.

Time progressions of the partial errors of the best run are shown in Fig. 4. As can be seen, they have a tendency to decrease to zero. Cost function is computed as an average of the approximation error on the training set and the error gained with the winner-takes-all evaluation of the validation set. As can be seen in the figure, it is possible for the approximation error to rise for a short period of time, when the winner-takes-all method decreases appropriately. Error on the test set is obtained from the winner-takes-all evaluation of the test set.

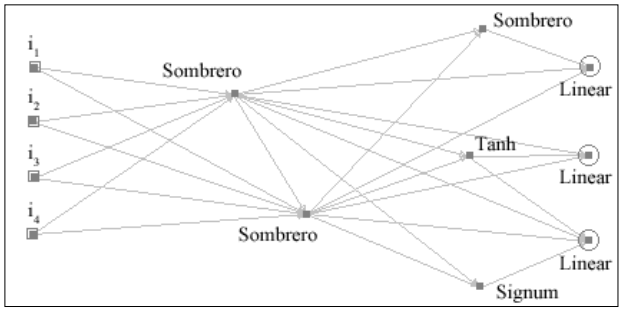

Fig. 3. The best network topology found by the experiment

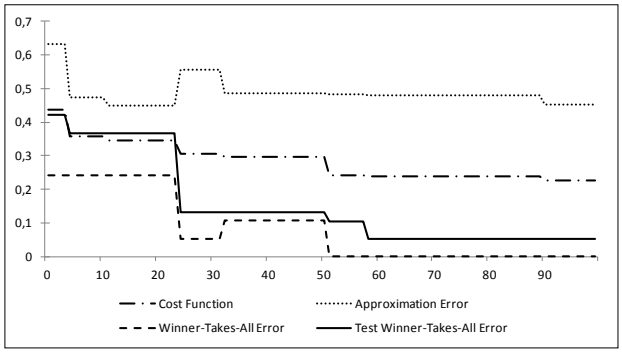

Fig. 4. Best run of the algorithm for the Iris flower data set

Average cost function evaluations per run	86144
Test data set error RMS (average)	0.2658
Test data set error RMS (best)	**0.0526**
Test data set error RMS (standard deviation)	0.1547

Tab. 2. Simulation results

5. CONCLUSION

As can be seen in the results in Tab. 2, the algorithm was able to find a neural network for classification of three linearly unseparable classes with one mistakenly classified sample of the test data set. Although this seems very promising, the average error and its standard deviation reached by multiple runs shows an expected limitation of this algorithm and ecoding – the fine-tuning of the network weights.

As mentioned before, there is a future need to find an efficient method for weight optimization in this algorithm. The method used in this experiment is quite straightforward optimization that does not show the required qualities.

Another alarming fact is the average number of evaluations of the cost function (i.e. how many times the whole dataset has been passed through the network). A big part of this number is created by the initialization phase. That means, that the generation of a random genotype creating meaningful network has to be optimized in future research.

6. ACKNOWLEDGEMENTS

This research is supported by the Internal Grant Agency of Tomas Bata University under the projects IGA/28/FAI/10/D, IGA/40/FAI/11/D and by the European Regional Development Fund under project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

7. REFERENCES

Fekiač, J.; Zelinka, I.; Burguillo, J. C. (2011). A Review of Methods for Encoding Neural Network Topologies in Evolutionary Computation, in *Proceedings of 25th European Conference on Modeling and Simulation ECMS 2011*. ISBN 978-0-9564944-2-9, pp. 410-416, Germany : Digitaldruck Pirrot GmbH

Fisher, R.A. (1936). UCI Machine Learning Repository: Iris Data Set. *Available from*: http://archive.ics.uci.edu/ml/datasets/Iris

Gruau, F. (1994). Neural network synthesis using cellular encoding and the genetic algorithm. Dissertation, l'Ecole Normale Superieure de Lyon , France. 159 p.

Kitano, H. (1990). Designing neural networks using genetic algorithms with graph generation systems, in *Complex Systems*, vol. 4, issue 4. pp. 461-476

Koza, J. R (1990). Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. 131p. Stanford : Stanford University

Koza, J. R.; Rice, J. P. (1991). Genetic generation of both the weights and architecture for a neural network, in *Seattle International Joint Conference on Neural Networks*, vol. 2, pp. 397-404

Luerssen, M. (2009). Experimental Investigations into Graph Grammar Evolution : A Novel Approach to Evolutionary Design. 204p. Saarbrucken : VDM Verlag Dr. Müller

Mandischer, M. (1993). Representation and Evolution of Neural Networks, in *Proceedings of the International Joint Conference on Neural Networks and Genetic Algorithms*. pp. 643-694

Schiffmann, W. (2000). Encoding Feedforward Networks for Topology Optimization by Simulated Evolution, in *Fourth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies*. pp. 361-364

Stanley, K. O.; Miikkulainen R. (2002). Evolving neural networks through augmenting topologies, in *Evolutionary Computation, MIT Press*, vol. 10, number 2. pp. 99-127