A Comparison of Some Methods for Evolving Neural Networks

Marko Grönroos
Turku Centre for Computer Science, Åbo Akademi, Lemminkäisenkatu 14 A, FIN-20520 Turku, Finland.
Email: magi@iki.fi, Tel. +358 2 215 4072

This poster paper presents an empirical comparison of four encoding methods for evolving neural networks. We use an evolutionary algorithm to evolve a population of genetically encoded neural networks. Only the network topology is encoded, and the connection weights are trained with a separate neural learning algorithm. The performances of the trained networks are used as fitness values to guide the evolution of the population. After a number of generations, the best network is picked up from the population.

We use a *direct encoding* method derived from the one proposed by Miller, Todd, and Hedge [4]. Each neuron and connection is encoded separately in the genome. The *graph generation grammar* developed by Kitano [3] is an early encoding method based on context-free and deterministic L-systems. Connections are determined from a connection matrix that is generated from a 1×1 symbol matrix using genetically encoded rewriting rules that replace each element with a 2×2 matrix. In *cell space encoding*, introduced by Nolfi and Parisi [5], the neurons are encoded with coordinates in a two-dimensional space. "Axon trees" are grown from the neurons that form connections when their branch tips touch other neurons. The fourth method, which we call here *generative cell space encoding,* by Cangelosi, Parisi and Nolfi [1], is a derivative of the previous one, and employs rewriting of neurons according to production rules to generate the neurons instead of encoding each neuron directly.

Our evaluation criteria in the comparison are classification accuracy and efficiency to use only the relevant input variables. We use three artificial problems to benchmark the input selection ability of the encoding methods. First is the classic 8-x-8 `encoder` problem. The x in the name refers to the fact that the number of hidden neurons is determined by the evolution process, as is the network topology generally. The `additive` data is generated using a function that has a nonlinear additive dependence on the first two variables, a linear dependence on the next three and is independent of the last five (pure noise) variables. The `interaction` dataset is similar to `additive`, except that the first two variables have interactive dependence. We use two real-world problems from the PROBEN1 problem database[6] to benchmark the methods according to classification accuracy: `glass` (classification of glass types) and `heart` (detection of heart disease). These problems are well known in machine learning benchmarking.

We conducted evolution experiments with each method-problem pair. The graph generation grammar performed best with the classification problems, but the direct encoding was not very far behind. The direct encoding and cell space encoding were about equally efficient with the input selection problems, surpassing the two other methods.

Details of the study are given in [2].

References

1. A. Cangelosi, D. Parisi, and S. Nolfi. Cell division and migration in a 'genotype' for neural networks. *Network: Computation in Neural Systems*, 5:497–515, 1993.
2. M.A. Grönroos. Evolutionary design of neural networks. Master's thesis, Computer Science, Department of Mathematical Sciences, University of Turku, Finland, 1998. Available from http://www.iki.fi/magi/opinnot/gradu/.
3. H. Kitano. Designing neural network using genetic algorithm with graph generation system. *Complex Systems*, 4:461–476, 1990.
4. G.F. Miller, P.M. Todd, and S.U. Hegde. Designing neural networks using genetic algorithms. In J.D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 379–384. Morgan Kaufmann, 1989.
5. S. Nolfi and D. Parisi. Growing neural networks. Technical report, Institute of Psychology, CNR, Rome, 1992.
6. L. Prechelt. PROBEN1 - a set of neural network benchmark problems and benchmarking rules. Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe, 76128 Karlsruhe, Germany, 1994.