Genetic Programming For Designing Ad Hoc Neural Network Learning Rules

Javier Segovia
Dep. de Leng. y Sist. Fac. de Informática U.P.M.
Campus de Montegancedo, s/n
Boadilla del Monte 28660 Madrid
fsegovia@fi.upm.es

Pedro Isasi
Departamento de Informática
Universidad Carlos III
C/Butarque,15, Leganes 28911, Madrid.
isasi@gaia.uc3m.es

Abstract

Learning rules have been studied in a purely generic, problem-independent context in an attempt to get universal learning rules. In this paper, we present a method that is capable of automatically producing neural net learning rules depending on the network architecture and the problem to be solved. Genetic programming (GP) techniques have been used to this end, as these techniques show signs of robustness in search processes, as well as being widely applicable.

1 Introduction

Some authors use the idea of adapting some parameters of a neural network, by means of a genetic Algorithm [Miller *et al.*, 1991; Kitano, 1990]

In this paper, we consider the possibility of automatically generating the best learning rule depending on the problem to be solved and the architecture used, taking a set of randomly generated rules and making them evolve as per the principle of natural selection. Genetic programming techniques are used to do this [Koza, 1992]. This idea is not new, although it has not been much exploited. The most important experiment was conducted by David Chalmers [Chalmers, 1990]

His experiments are successful, finding learning rules for linearly separable functions using a single-layer neural net. In this paper, we present the results of work that goes beyond Chalmers approach, as it places no constraints on the learning rule and also operates with non-linear functions.

2 Results Analysis And Conclusions

The problem which we have addressed is to search learning rules for a neural net that is to learn a series of logical functions. The functions are as follows: AND, $\overline{AND}$, OR, $\overline{OR}$, ID1, ID2, $\overline{ID1}$, $\overline{ID2}$, XOR, $\overline{XOR}$. As we can see, not all of them are linearly separable, which means that the net to be used should have more than one layer.

The experiments seem easily to solve one of the problems that has taken up so much work in the field of neural nets, the search for learning rules for multi-layer nets.

The explanation is obvious. The solutions found perfectly fit the **200 sets of initial weights** that constituted the training set.

The experiments show that genetic programming is not a good method for generating generic rules, as it leads to overfitness. However, the method is absolutely efficient in the case of a constrained environment. Hardware implementations of a neural net are one example of such an environment, where methods that require exhaustive information interchange, such as backpropagation, cannot be implemented because of the amount of cabling required.

Another case where non-generic rule generation may be important is when there is no known learning rule, either because of the net architecture or other parameter restrictions. In these cases, and for given experiments, genetic programming would be capable of providing the required learning rule.

Finally, an important applicability of genetic programming methods is to improve existing solutions by evolving already operational solutions.

References

[Chalmers, 1990] D.J. Chalmers. The evolution of learning: An experiment in genetic connectionism. In D.S. Touretzky ..., editor, *Proc. of the 1990 Connectionist Models Summer School.* M. Kaufmann, 1990.

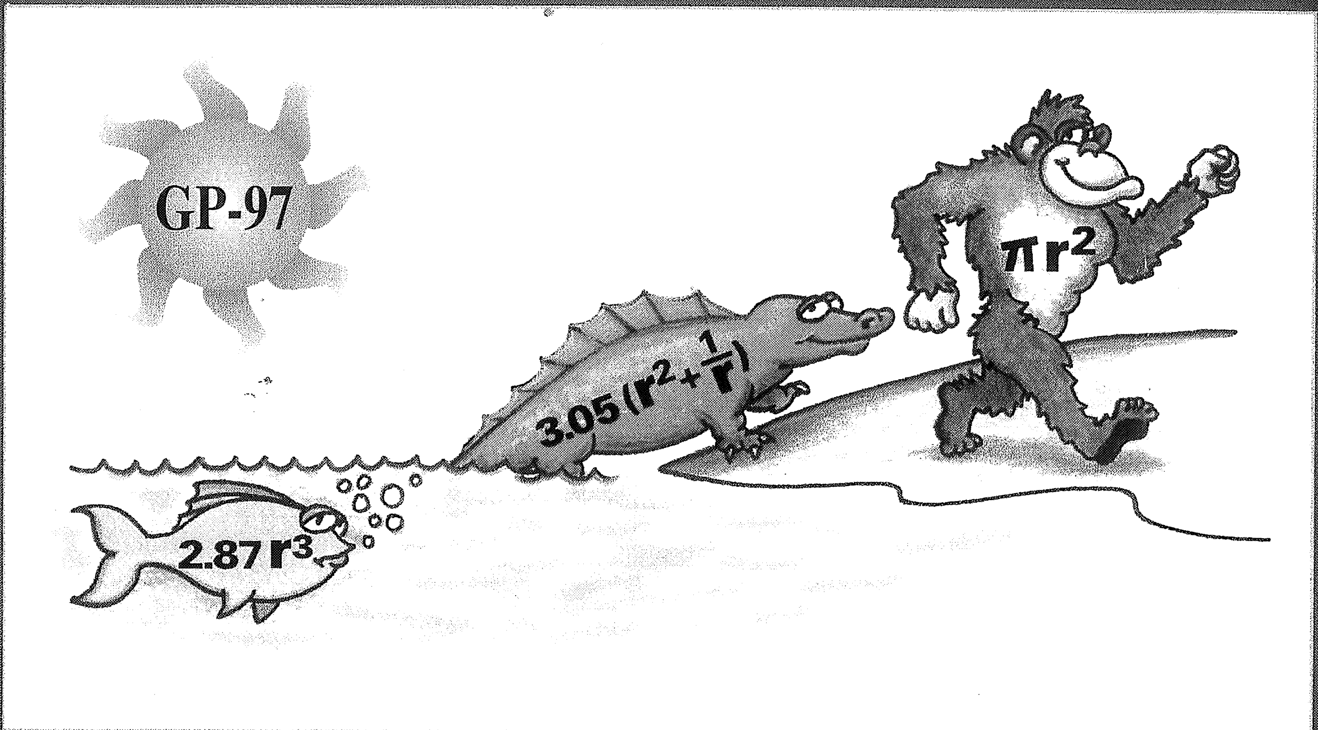
[Kitano, 1990] H. Kitano. Designing neural networks using genetics algorithms with graph generation system. *Complex Systems*, I(4):461–476, 1990.

[Koza, 1992] J. Koza. *Genetic Programming.* MIT press, Cambridge, Mass., 1992.

[Miller *et al.*, 1991] G.F. Miller, P.M. Todd, and S.U. Hedge. Designing neural networks. *Neural Networks*, I(4), 1991.

CONFERENCE PROCEEDINGS

Genetic Programming 1997



Proceedings of the Second Annual Conference
July 13-16, 1997, Stanford University

Edited by
John R. Koza
Kalyanmoy Deb
Marco Dorigo
David B. Fogel
Max Garzon
Hitoshi Iba
Rick L. Riolo