Using Genetic Algorithms to Extract Rules FromTrained Neural Networks

Edward Keedwell, Ajit Narayanan and Dragan Savic

School of Engineering and Computer Science
University of Exeter
Exeter, UK, EX4 4QF

Abstract

This paper describes a novel method of extracting rules from trained artificial neural networks. The system which uses genetic algorithms to find an optimal path through the network for a classification. It is claimed that this system compares favourably to traditional rule extraction algorithms due to its ability to extract rules from very large and complex networks without an exponential increase in computational complexity.

1 INTRODUCTION

The use of artificial neural networks(ANN) as classifier systems is well documented as they often attain higher accuracy rates for a given domain than their symbolic counterparts. In many applications, however, it is not sufficient to classify data accurately, the classification must be explained. ANNs do not offer such an explanation explicitly as the information which has been encoded is distributed around the network. Andrews *et al* (1996) state that this information is coded in (a) the network architecture itself (b) an activation function associated with each unit of the ANN; and (c) a set of numerical parameters (called weights). Any rule extraction system therefore must take one or more of these components as input and create meaningful rules from them, this approach uses genetic algorithms.

2 METHOD

The ANN is represented to the genetic algorithm as a set of integers and the placement of those integers (genes) in the chromosome. Therefore [5,1] represents the weight from the fifth input unit to the first hidden unit multiplied by the weight from that unit to a specified output unit. The fitness function for the gene is therefore a multiplication of the weight values. The fitness of the gene represents the effect that that particular gene has on the output, be it negative or positive. Normal genetic algorithm processes (such as mutation and crossover) are then performed to find the top chromosomes.

This system, once given the required classification for a particular output unit, can find the most fit paths through the network for that classification. These paths through the network each have an associated weighting as to how strongly they effect the output. Rules can then be created which state a direct relationship between the input to the network and the output with this weighting.

Whilst this system does not output definite rules about the ANN (a single input does not guarantee a particular classification) it allows the user to find the important elements in the ANN's decision making process.

Traditional algorithms use complex functions to extract rules and it may seem as though there is little need for the genetic algorithm here as the computation is very simple. However, the problem increases exponentially for large networks; for instance an ANN with 100 input units, and three hidden layers of 30 units each generates a weight search space of 2,700,000 elements.

This system has been successfully tested on three datasets, including the mushroom classification dataset which is available from the UCI repository (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/).

3 CONCLUSIONS

This gives a ranked set of rules can give the ANN user a valuable insight into the process which the ANN has gone through to enable it to correctly classify the data. Therefore, it is anticipated that this system will be of most use to those ANNs which include many hidden layers and would thus be uncomputable using traditional methods.

4 BIBLIOGRAPHY

1. Andrews, R. *et al* (1996) An Evaluation And Comparison of Techniques For Extracting and Refining Rules From Artificial Neural Networks.

http://www.fit.qut.edu.au/NRC/ftpsite/QUTNRC-96-01-04.html