Optimization of GA Parameters to Train Recurrent ANN through Weight Adjustment and Selection of Activation Functions

Alejandro Pazos
Computer Science Dep.
University of A Coruña
15071 A Coruña – Spain

Julián Dorado
Computer Science Dep.
University of A Coruña
15071 A Coruña – Spain

Antonino Santos
Computer Science Dep.
University of A Coruña
15071 A Coruña – Spain

Juan Ramón Rabuñal
Computer Science Dep.
University of A Coruña
15071 A Coruña – Spain

1 INTRODUCTION

Nowadays, recurrent ANN are the most appropriate tool to facing pattern recognition or forecast problems in complex domains or with a temporal component. The use of feedforward ANN in these cases means to force it into a task for which it has not been designed. However, RANN present some problems, due to their slow training and difficult convergence. For these reasons, the use of RANN is not very common and it has been proposed to substitute training algorithms based on gradient descent for others such as GA. Nowadays, the development of GA that train ANN is not too hard. However, the huge number of GA parameters is becoming a problem itself, due to the lack of theoretic studies that clarify the situations in which each of them may be used. We have implemented a system which trains RANN by using GA. This training is carried out by adjusting the weights among the connections of the ANN´s neurons and by selecting the activation functions for each neuron in the network. The system has been proved with success at predicting at short term real-time series. The aim of this research is to carry out tests with the GA´s parameters and to check how these changes affect the functioning of the resulting ANN.

2 GENETIC ALGORITHM

First step, codification. Each ANN has been codified with two arrays. The network connection weights (weight's array) can be seen as a n*n matrix, where n stands for the total number of neurons in the network and the array of the activation functions for each neuron with the type of activation function and its parameters. The GA uses an elitist strategy and the codification of the individuals is made through float point numbers. The second step is to decide the size of the population. In the tests carried out, we started with a population of 100 individuals, which were gradually increased to 5000, where the training time becomes excessive. Three common types of selection of individuals have been selected: random selection between better individuals and whole population, Montecarlo and tournament. The mutation operator used does not differ from the one proposed by Holland. The crossover operator must be modified, since it must be applied to both parts of the individual: the weight part and the activation functions part. The crossover operator has been designed as if both parts were two independent individuals and two crossover operators were applied. Apart from the usual crossover operator with one crossover point, tests have been carried out with a two-point crossover operator and with the uniform operator. For individual substitution, apart from the usual Darwinist technique, which eliminates the worst adapted ones, two additional techniques have been tested: substitution of parents only if the offspring adapts better than the parents and substitution according to similarity of error level.

3 CONCLUSIONS

For the RANN training, the GA parameters that most affect the results are the population size, the crossover number and the mutation number. All three parameters must have a size not too large and always limited by the physical memory capacities of the machine that carries out the simulation. The non-linearity of the solutions search space causes the random selection of individuals to be the best option. It is also the fastest one, since it doesn't need to consult the values of adjustment of the individuals. Relating to crossover types, one or two-point crossover behave in a similar way, and so it's possible to use anyone of them. Nevertheless, uniform crossover behaves clearly worst, showing a great dependency between genes. About individual substitution, the Darwinist substitution of the worst individuals of the population is also the fastest solution and the one that offers best results, due also to the non-linearity of the solutions space. Finally, due to the increasing of the GA performance, systems like this one we present, can be used in personal computes of low cost to develop temporal series forecast tasks, which, not a long time away, need much more expensive workstations.

In further works, it will be developed a hill climbing operator in such a way that an exhaustive search will be done in the individual neighborhood and the dynamic variation of the crossover ratio and mutation ratio during the GA simulation. Relating to ANN, it will be raised the problem of the design of the architecture with the use of an architecture evaluator GA that test them by means of ANN training proves.