Empirical Study of the Influences of Genetic Parameters in the Training of a Neural Network

Paulo Gomes*
Instituto Superior
de Engenharia de Coimbra
pgomes@sun.isec.pt

Francisco Pereira
Instituto Superior
de Engenharia de Coimbra
xico@sun.isec.pt

Arlindo Silva
Instituto de Sistemas
e Robótica
arlindo@isr.uc.pt

Abstract

This paper presents the empirical results achieved in the computer system ROBOTS. This program simulates a virtual world, where agents, called robots, interact with an environment. Each robot is controlled by a neural network. The evolution of the robot behaviour (which is determined by the variation of the weights in the neural network) is done using a genetic algorithm. We describe the conceptual model used in ROBOTS. We also show how the genetic parameters and the environment itself, influence the robot's adaptation to the environment.

1 Introduction

In this paper we will study the influence of the parameters of a genetic algorithm on the training of a neural network. In particular, we are interested in the way the behaviour of an artificial agent can evolve using a genetic approach. The agent's behaviour is modelled through the use of a neural network. Development of a neural network is a difficult task. One of the difficulties is training the neural network in order to achieve the desired configuration of weights. This can be done using various algorithms. One of the well known algorithms developed for training neural networks is back propagation [Hertz et al., 1991]. However this algorithm is dependent on the training examples and is difficult to use in a application where the evaluation function is non-stationary (the fitness landscape of a non-stationary function changes over time) [Hertz at al., 1991]. This problem is due to the local search level of back propagation.

Genetic Algorithms are an alternative to conventional algorithms (like back propagation). As a probabilistic global search procedure [Goldberg, 1989] they avoid the traps of local minima and are able to converge to the region where the optimal solution is. Good results in training neural networks with a genetic algorithm have been reported (see e.g. [Porto et al., 1995] [Belew et al., 1991]).

In order to study the influence of genetic parameters in the training of a neural network, we implemented a computer system called ROBOTS. In the next section we describe this system. Section three presents the results obtained in preliminary experiences and shows how the robots behaviour evolved in the virtual world. Finally we present our conclusions and future work in section four.

2 The Robots System

This section presents the conceptual model used in ROBOTS. We start by describing the virtual world used for simulation. Then we present the neural network topology and genetic algorithm.

2.1 Environment

Robots live in a bidimensional virtual environment containing randomly distributed pieces of water, and food. Each position in the world may

*This research was partially supported by a MSc. grant from JNICT (PRAXIS XXI BM 6563 95).

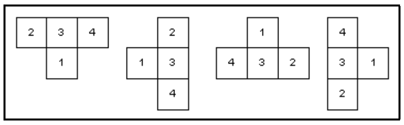

Figure 1: Board positions seen (by the robot) depending on the robot's orientation.

be empty or have one of the following items: water, food, robot, robot and water, or robot and food. Positions with food or water have an associated quantity. Quantities range from 1 to 9.

2.1.1 Robots

The robot behaviour is controlled by a neural network. All robots have the same neural network architecture (only the network weights differ from robot to robot).

A robot has two major properties : energy and mobility. Energy is used by the robot to fight and to survive when is not moving. Mobility is used for moving around the arena. Energy and mobility are measured in units raging from 0 to 7 for energy, and from 0 to 15 for mobility. Any robot can "see" the contents of the position where it stands, and three contiguous positions in front of it. Board positions seen by a robot standing in position 1 are presented in Figure 1.

2.1.2 Interactions between entities

Relations between entities are ruled in the following way :

- Robot/Food: if a robot is in a position with food, the quantity of food decreases by one and the robot's energy increases by one;
- Robot/Water: if a robot is in a position with water, the quantity of water decreases by one and the robot's mobility increases by one;
- Robot/Robot: if two robots move into the same position, the robot with less energy looses as much energy as half of the opponents energy. If the energy level drops bellow zero then the robot dies.

2.1.3 Other rules

Other rules for life in the arena are:

- Each time that a robot moves it looses one unit of mobility;
- If the robot's mobility is zero then it can't move any more;
- Each time that a robot doesn't move, it looses one level of energy, except if it is in a position with food or water;
- If the robot level of energy drops bellow zero then it dies.

2.2 Neural Network

Robots behaviour is modelled by a feed forward neural network with two layers. The input layer has 18 neurons each one corresponding to an input. Inputs correspond to values about the world, which are: perception of the positions in front of the robot (4 positions) and level of energy and mobility that the robot has.

The output layer has nine neurons corresponding to the possible robot moves. The output with the higher value corresponds to the action to be executed.

Since all robots have the same neural network architecture, different behaviours are due to the weights associated with the connections between neurons.

2.3 Genetic Algorithms

The evolution of the robot's behaviour is done using a genetic algorithm. A generation is composed by individuals, where each individual is a robot and is defined by a set of genes. Each gene is a weight of the neural network. Genes are represented by real values. Individuals of the initial population are randomly created (genes are initialised with real values between 0 and 1). The genetic algorithm will apply two genetic operators: selection and mutation. Following [Porto et al., 1995] we will not apply any kind of crossover operator.

2.3.1 Selection

Selection is performed in a natural way. Individuals of a generation are placed in the artificial world and compete for existing resources, trying to survive. When the number of individuals falls under a given threshold, established by the user, the generation is completed. The surviving robots will be the parents of a new generation.

2.3.2 Mutation

Individuals of the next generation are obtained by making copies of the surviving robots and then mutating some of their genes. The number of individuals in the new generation is defined by the user as the population size. There is always one non-mutated copy of each surviving robot in the new generation. The other copies have mutations in their neural network weights by adding a random value to the weights between -1 and +1. Not every weight is mutated, only a percentage of them. This percentage is defined by the user as the mutation rate. Selection of weights is done by a random process.

3 Experimental Results

In this section we present results achieved with the ROBOTS system. Two types of experiments were done. One is the study of the influence of the population size in the evolution of the robot's behaviour. In the second experiment we will compare the behaviour of individuals from a random population with the behaviour of individuals which have evolved over 1000 generations.

3.1 Influence of the Genetic Parameters

In order to study the influence of the population size in the evolution of the robots, three experiments with different population sizes were performed.

All experiments run for 1000 iterations (each iteration corresponds to a new generation). The arena is a 30x30 grid. Experiment one had the following conditions: population size of 20, a mutation rate of 5%, a percentage of food in the arena of 5%, and a percentage of water of 5%. Experiment two and three had a population size of 30 and 40 individuals, respectively. Each generation is completed when the population size reaches 10 live robots.

Data gathered in the experiments comprises: the average life of surviving robots, percentage of used food and water, number of natural deaths, and number of deaths due to robot competition.

In table 1 we present the statistical values.

	Exp. 1	Exp. 2	Exp. 3
AMRLD	13.584	13.898	16.201
APFU	5.411	10.922	30.414
APWU	5.122	7.713	27.807
ANND	9.114	18.685	24.95
ANDRC	0.886	1.315	5.05

Table 1: Statistical values obtained in the three experiments.

AMRLD - Average life duration of the surviving robots
APFU - Average percentage of food used
APWU - Average percentage of water used
ANND - Average number of natural deaths
ANDRC - Average number of deaths by robot competition

The results show that the increase in the population size improves the robot's evolution. Robots live more time, and because of that they encounter each other more times increasing the number of deaths caused by other robots. With the increase of the population size it was expected that the usage of water and food would increase too, and that's what happened.

3.2 Evolution of the Robots Behaviour

We evolved ten different populations over 1000 generations. Individuals of these populations were then compared against individuals of a random population. The process was the following: we created ten new populations by merging the individuals of each evolved population and the individuals of the random one. Evolution of these ten merged populations was then simulated in the virtual world. The simulation runs for 100 iterations. Finally we calculated the average number of survival robots from each population. The results are shown in table 2.

Merged Population Number	Random Population	Evolved Population
1	7.5	12.5
2	6	14
3	7.5	12.5
4	8	12
5	6	14
6	3.5	16.5
7	1	19
8	6	14
9	4.5	15.5
10	0.5	19.5

Table 2 - Statistical values of the population comparison experiments.

The conclusion taken from table 2 is that populations which evolved over 1000 generations, are better adapted for survival than the random population. The evolution of the neural network performed by the genetic algorithm works, thus improving the robots' survival capacity.

4 Conclusions and Future Work

This work presented a hybrid system using a genetic algorithm to train a neural network. We showed that a genetic algorithm can train a neural network in order to improve its performance, despite the non-stationary evaluation function. The results presented are preliminary, and by the time this paper is written, experiments with the mutation rate and environment parameters are being done.

References

[1] Belew, R.K., McInerney, J., and Schraudolph, N.N. (1991). Evolving Networks: Using the Genetic Algorithm with Connectionist Learning. In Proceedings of the Second Artificial Life Conference, 511-547, Addison-Wesley.

[2] Goldberg (1989). Genetic Algorithms in Search, Optimisation and Machine Learning. Addison-Wesley, Reading, MA.

[3] Hertz, J., Krogh, A., and Palmer, R. (1991). Introduction to the Theory of Neural Computation. Addison-Wesley.

[4] Porto, V., Fogel, D., and Fogel, L. (1995). Alternative Neural Network Training Methods. IEEE Expert, June 1995, 16-22.