Intelligent Robotics – Coursework 2

Andreea-Tamara Tanasescu and Deryk Mathew – BSc Computer Science

# Overview

This report aims to describe and analyse the decisions and techniques used in developing a genetic algorithm to evolve an artificial neural network robot controller for the e-puck robot to simulate the behaviour of a rat in an elevated plus-maze.

# Arena

There are various models of the Elevated Plus-Maze (EPM), some of them have only small differences (e.g. less than 5cm difference in length of the arms), others have more notable differences (e.g. having small walls around the “open” arms). Most encountered differences between EPM specifications are in the length of the arms, height of the walls surrounding the arms and the materials used for creating the maze. All of these variations can influence test results and should be taken into consideration (Sandy, 1996).

For our setup, we do not take into consideration the material used for creating the EPM since it is a virtual simulation. The notable dimensions we studied for our EPM were the length of the arms and the walls surrounding them. These are as shown in Table 1 and are based on the dimensions described by File and Pellow.

The only alterations made were to the height of the walls around the arms of the EPM. We made the height of the closed arm walls lower by 10cm. This was done to ease the visualisation of our robot in the maze since we found that 40cm walls would reduce the view angle we could use and still have the robot in view. We also added a 2.5 cm wall around the closed arms to prevent our robot from falling.

One aspect of note is that the elevation does not influence our simulation, but our aim was to create an arena that would be as close to the actual specifications used in real tests.

Table 1 Elevated Plus-Maze Dimensions

|  |  |
| --- | --- |
| Closed Arms | 50 x 10 x 30 cm |
| Open Arms | 50 x 10 x 2.5 cm |
| Central Area | 10 x 10 cm |
| Elevation | 50 cm |

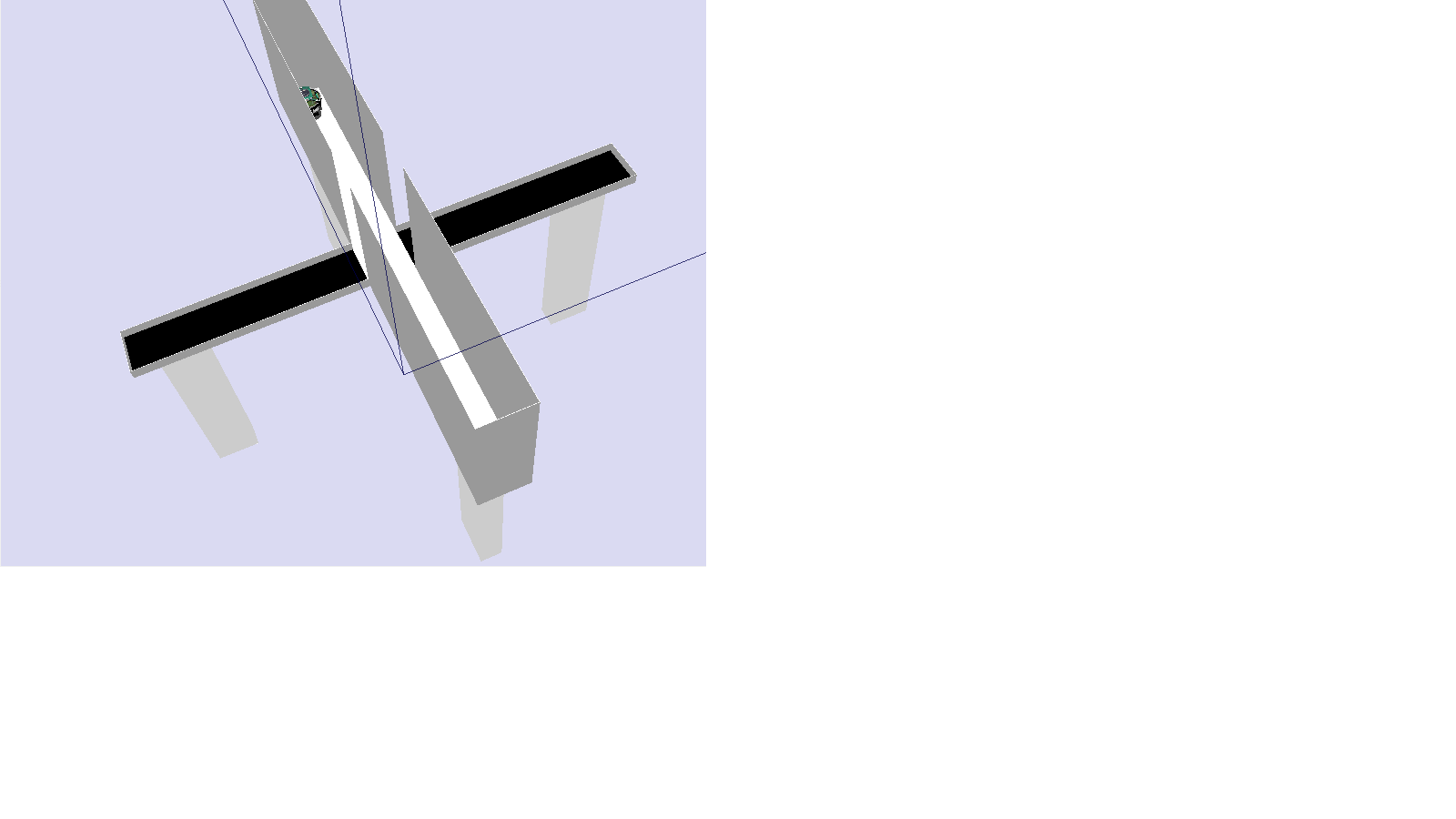
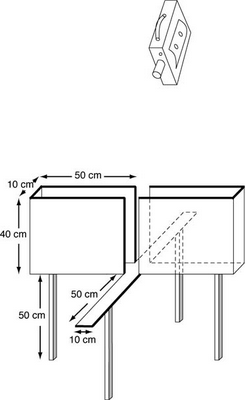
 

Figure 1 Left - The EPM we designed inside the Webots software Right – EPM figure used as reference (File, et al., 2004)

# Artificial Neural Network

The artificial neural network (ANN) used is a recurrent one with one hidden layer and one recurrent context layer modeled as an Elman neural network. (Cheng, et al., 2002)

The input layer has 8 nodes which represent the Infrared (IR) sensors. The hidden and context layers both have a size of 4 and the output layer has a size of 2 representing the speeds of the 2 wheels for our e-puck robot.

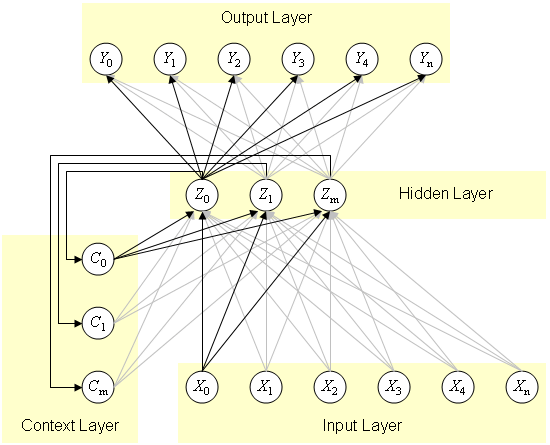


Figure Elman neural network example with one hidden layer and one context layer

Having the context layer connected to the hidden layer as shown in Figure 2, provides recurrence in our ANN and implicitly makes our controller have a memory of the moves it previously made. This is a good choice in our problem space, because the rats we are trying to model have a memory of their previous moves and this influences their movement in terms of having a tendency of moving in the same direction as they did in previous moves and not erratically changing their movement.

The activation function used is the sigmoid function:

This is a good choice in our case since the range of the sigmoid function is [-1, 1] and the range of the speed of our wheels is [-1000, 1000], which makes conversion from the output to the value we are looking for very easy (multiply the result of applying the activation function by 1000).

# Genetic Algorithm

Even though genetic algorithms are very successful there has been no standard workings for a genetic algorithm created. This causes a problem when developing a genetic algorithm as there is no clear best practice when producing it (Goldberg, 1989), due to this there is no way to test if the genetic algorithm created is the optimal genetic algorithm except for trial and error.

For this problem we decided to only evolve the weights of our ANN and not the activation function.

## Genotype

The genotype we used for this problem is an array of weights. These correspond to the weight matrices of our ANN.

## Fitness Function

The fitness function we created for the problem takes into account how much the robot has explored the Elevated Plus-Maze and gives penalties for “fear” when the robot is in the open arm (Costa, et al., 2013).

### Exploration

The exploration element of the fitness function was done my separating the EPM into 21 areas: 5 areas for each arm and the central area. When running the demo to calculate the fitness function of the current ANN with its respective weights, we keep track of what areas the robot has been to and add to the fitness each time it explores a new area.

We also decided to remove areas from “memory” in some cases. If the robot hasn’t been in a certain area for a while, then we remove it and we provide the fitness bonus again. Also if a robot has been to a number of different areas since a specific area, then we remove it from memory. This was done to make the robot try to explore as much of the maze and enforce backtracking to find new areas.

### Fear

The fear element of the fitness is implemented by having a probability of the robot being scared for each area it is in. This is higher in open arms and lower in closed arms. We then generate a random number at each time step and if that number is lower than our fear probability in the current area, then we subtract a fear factor from the total fitness

This implementation of fear also adds randomness to our fitness function values, which may influence evolution, but it models the rat in terms of the fact that it could be scared in an area or it could not and this is random and dependent on other factors (e.g. psychological factors) which are not observable in robots.

## Population size

From previous studies in Biologically Inspired Computing course last semester it was found that a population between 50-100 was optimal for evolving the algorithm(Mitchell, 1998). This was tested with the developed artificial neural network using different population sizes to find the optimum population size which we found was 50.

## Generations

We decided to run the GA for a number of 50 generations. This was done mostly due to time constraints, but we also observed that after that number of generations, our fitness would not increase anymore.

## Elitism

Elitism for the EPM problem was shown to be fundamental to keeping the best individuals in the population (Costa, et al., 2013). We decided to implement elitism in our GA by keeping the individuals of our population with the fitness in the top 10% of the population.

## Parent Selection

The selection scheme used for the genetic algorithm was the tournament selection scheme. This allowed us to modify the selection pressure until the results were optimal and provides a degree of randomness to our selection process, while still trying to have parents that have high fitness.

## Mutation rate

As with the other parts of genetic algorithms there is no standard mutation rate, generally a low mutation rate between 0.5-2% produces a big enough change from generation to generation without making it overly random(Srinivas & Patnaik, 1994). As with all areas of genetic algorithms there is a probabilistic chance of mutating a genotype, the rate of mutation as with elitism should not be too high as this will run the risk of losing the good qualities of the original genotype. Too small on the other hand will inhibit progress as only the elite genotypes will be chosen between generations. Mutation is a secondary system within genetic algorithms as its role is restoring lost genotypes, an example would be where the population that has converged to 0 where the solution is 1, crossover would not regenerate to 1 whereas mutation could (Srinivas & Patnaik, 1994). The mutation rate was found using different rates while testing those rates in generations and the optimal rate of 0.06 was found.

## Crossover

After selection the two parents were selected for crossover(Mitchell, 1998), this is done to pass areas of genotype from two parents to create a child (Figure 3).The selection of the parents is important for the crossover as the chosen genotypes of the parents must fit together to produce a child. This is a single point crossover where the algorithm invokes crossover only if a random number in the range of 0 - 1 is greater than the crossover rate otherwise the parents are unaltered(Srinivas & Patnaik, 1994). The crossover rate is found using trial and error as with most genetic algorithms there is no standard crossover rate.

Parent

Parent

Child

Figure Crossover Illustration

# Conclusion And Future Developments

If we had more time, we would have wanted to include the e-puck ground sensors in our calculation, since we made the floor of the open arms black. Behavior in our ANN may emerge after adding these in our topology and calculations.

We would have also liked to make our fitness function more complex by adding collision detection and punishing the fitness of controllers that collide with walls.

# References

Cheng, Y.-C., Qi, W.-M. & Cai, W.-Y., 2002. Dynamic properties of Elman and modified Elman neural network. *Machine Learning and Cybernetics,* Volume 2, pp. 637-640.

Costa, A., Vargas, P. & Tinos, R., 2013. Using explicit averaging fitness for studying the behaviour of rats in a maze. *Bioinspired Learning and Optimization*, pp. 940-946.

De Jong, K. A., 1975. *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems,* Ann Arbor: University if Michigan.

File, S. E., Lippa, A., Beer, B. & Lippa, M. T., 2004. Animal Tests of Anxiety. *Current protocols in neuroscience.*

File, S. E. & Pellow, S., 1985. The effects of triazolobenzodiazepines in two animal tests of anxiety and in the holeboard. *British Journa of Pharmacology,* 86(3), pp. 729-735.

Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimisation and Machine Learning.* 1st ed. Boston: Addison-Wesley Longman Publishing Co..

Mitchell, M., 1998. *An Introduction to Genetic Algorithms.* Cambridge: MIT Press.

Sandy, H., 1996. A Review of the Validity and Variability of the Elevated Plus-Maze as an Animal Model of Anxiety. *Pharmacology Biochemistry and Behaviour,* Volume 54, pp. 21-30.

Srinivas, M. & Patnaik, L. M., 1994. Genetic algorithms: a survey. *Computer,* 27(6), pp. 17-26.