

An Evolutionary Approach to Rat Exploratory Behaviour in the Elevated Plus Maze

Clarissa Cremona¹, Helmi Fraser²

Group 5: ¹ cc531@hw.ac.uk, ² hmf30@hw.ac.uk

1. Introduction

The elevated plus maze is a widely used experimental apparatus for measuring anxiety in rodents. Historically, it was first used in the 1950s in order to analyse the approach/avoidance conflict in rats exposed to a new environment. More recently, it has been used to test the effects of behaviour modification drugs by observing the movements of the rat before and after exposure.

In this report, a method of developing a computational model to simulate the behaviour of a rodent in this maze is described. This was achieved by using a genetic algorithm to evolve a feed-forward multi-layer perceptron controlling the movements of the robot.

2. Background

2.1 Elevated Plus Maze

Initially conceived in 1955 by K. C. Montgomery, the elevated plus maze was a way to study the simultaneous feelings of curiosity and fear that rats experience when exposed to a new environment. [1]

The elevated plus maze is a four-arm, raised platform in the shape of a plus sign. The maze consists of two enclosed arms and two open arms, perpendicular to the enclosed. [2] It is one of the most widely used and validated tests for measuring the anxiety responses of rats under variety of conditions. [3]

As a result of early experiments, it was found that the rodent tends to spend more time in the enclosed arms. The behaviour observed within these experiments is as a result of the approach/avoidance conflict when a rat is within a new environment. [4] It was common to record the behaviour of the animal for 5 minutes, as it was found that after this time, rats would acclimatise to their environment and its behaviour would change. [5]

2.2 E-puck

Developed by Mondada et al. at the Autonomous Systems Lab of École Polytechnique Fédérale de Lausanne (EPFL), the E-puck (see Fig 1) is an open source, two wheeled, differential robot measuring roughly 7cm in diameter. [6]



Figure 1. E-puck mobile robot

The E-puck was originally developed to allow students to address a wide range of engineering fields.

The platform features a wide range of sensing capabilities as standard, with the additional option to expand through the use of top and bottom extension slots. These physical extensions can range from extra sensors, to additional controllers for more computational capabilities (such as an Arduino board).

The E-puck has a multitude of sensors, including eight infrared proximity sensors, an accelerometer, microphones and a colour CMOS camera in front. Its actuators consist of two stepper motors which control the movement of the two wheels.

2.3 Webots

Webots is a commercial simulation software which supports the e-puck and includes libraries of sensors and actuators for creating and tailoring robot models. It allows for the creation of complex environments for the robot simulation and supports three-dimensional physics through the Open Dynamics Engine library.

It is quite advantageous to perform simulations prior to investigations with real robots. This is because simulations are faster, easier to set up and cheaper to use.

For example, a real robot is susceptible to damage from the environment, such as drops, liquid spills or is hit by an object. Repairing such damage could be costly, both in time

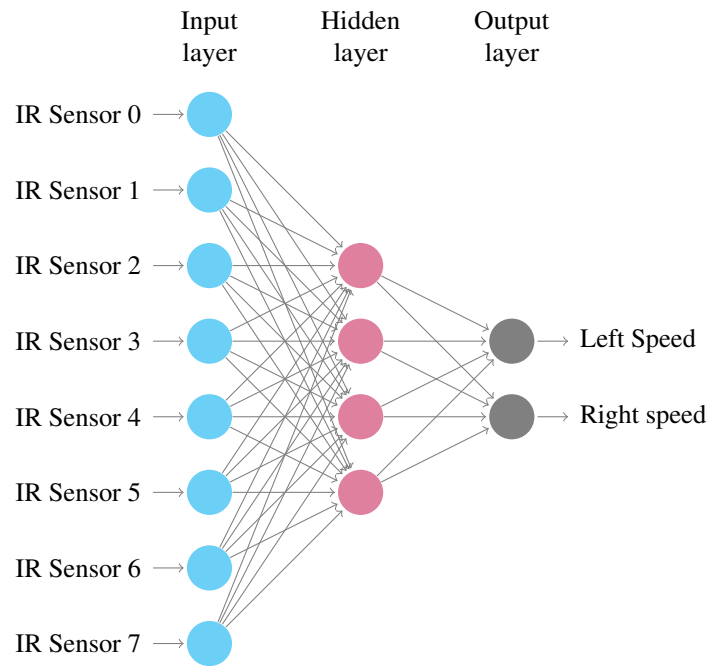


Figure 2. Neural network

and cost. However, within a simulation, this does not cost anything.

In this experiment's example, if a real robot was used, it would have to be fixed or replaced each time a controller failed. In addition to this, a genetic algorithm was used, and this would require a significant investment of time using a real robot microcontroller. [7]

Despite this, the simulation results are transferable onto real robots. Therefore, once the best controller has evolved, this could be run on a physical e-puck.

The elevated plus maze was modelled in Webots. Our model is shown in Fig. 3, along with the E-puck model.

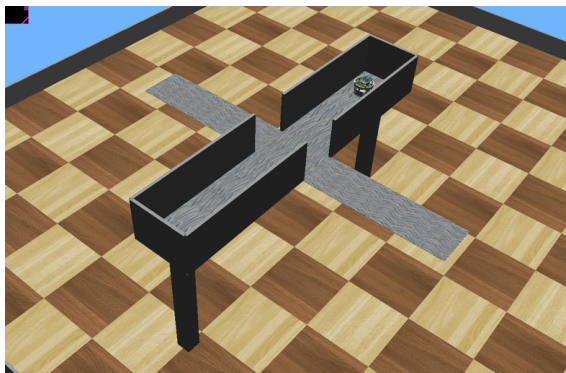


Figure 3. Simulation arena

2.4 Neural Network

An Artificial Neural Network (ANN) is a system that is modelled on the biological brain, particularly on the hypothesis that mental activity consists primarily of electrochemical activity in networks of brain cells called neurons.

The signal output of a neuron can either cause excitation or inhibition in the neuron it is connected to. When a neuron sends an excitatory signal to another neuron, this signal is added to all of the other inputs of that neuron. If it exceeds a given threshold then it will cause the target neuron to fire an action potential, if it is below the threshold then no action potential occurs. [8]

An ANN is comprised of nodes (neurons) connected by directed links. Each neuron receives input values either from the environment or from other neurons. Based on these inputs, the neuron calculates an output value by way of an activation function. Each link also has a numeric weight, known as a synaptic weight, associated with it, which determines the strength and sign of the connection. In effect, it is a distributed and parallel processor.

The network acquires knowledge from its environment through a learning process; a learning algorithm. The objective is to modify the network's synaptic weights for the network to perform its desired function. The ability for synapses to modify is known as synaptic plasticity. [9]

Shown in Fig. 2 is the utilised network.

2.5 Genetic Algorithm

A genetic algorithm is an iterative optimisation method based on natural selection that attempts to emulate the process of biological evolution. [10]

The algorithm begins by first creating a random set of possible solutions to the problem.

Each solution is called an *individual* with its properties being termed *chromosomes*. These chromosomes contain *genes* and the complete set of possible solutions is the *population*.

Each individual in the population is evaluated based on

how well it performs as a solution to the problem according to a fitness function.

The result of this fitness function is a performance metric for the individual, and a selection operator utilises this metric to select individuals to create the next generation, a method termed *reproduction*.

This is achieved through a genetic operator called a crossover, which is the mixing of a certain number of traits (genes) from each parent to produce an offspring. After this, random mutation of individuals in the population may occur. The algorithm terminates when a certain number of generations has been produced or when a solution with an acceptable level of fitness has been found.

3. Implementation

3.1 Arena setup

In the model implemented, an ANN with fixed architecture was used, see Fig. 2. A genetic algorithm was used to optimise the weights of the ANN, evolving the desired rat behaviour.

The simulated elevated plus maze was discretised into 15 positions as shown in Fig. 4. This was inspired by a similar, proven solution detailed within Costa et al. and Costa and Tinós. [11] [12] GPS ranges from the E-puck were assigned EPM positions in order to provide the robot's position to the supervisor.

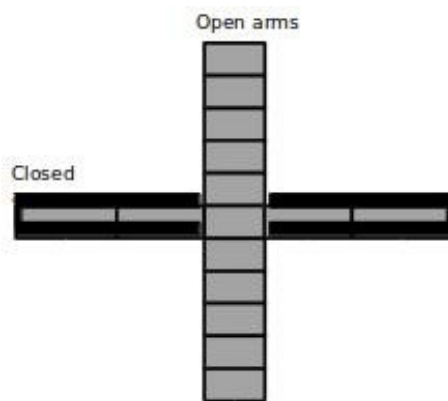


Figure 4. Arena boundaries

3.2 Robot controller

The e-puck is controlled by a multilayer perceptron (MLP), shown in Fig. 2 with arrows representing weighted links.

The network's inputs are data from the circumferential IR proximity sensors. The outputs of the network specify wheel speed values.

In terms of activation functions, every neuron uses a hyperbolic tangent function. This suited this application as this function's output is between -1 and 1, which is ideal because wheel speeds can be negative. The output values are scaled by 1000, to match the possible range of values in wheel speeds.

Each layer of synaptic weights is represented by an N-by-N matrix, where each row is a neuron's connection's weights to the next layer. [13] These individual sets of weights are

encoded into another data structure, which contain the weights of each layer.

3.3 Evolution

The network's weights are optimised by a genetic algorithm.

Each individual of the population is evaluated by the GA. As stated, each individual is a separate robot controller (ANN), with different parameters (weights).

The population is initialised via a function that fills a matrix with uniformly distributed random numbers in the range of -1 to 1, which are the initial weights. The size of the initial population is dictated by the variable *popsize*.

The fitness of each individual is evaluated with a fitness function. At each time step, this takes in as input the position of the robot and the proximity sensor values. The fitness value assigned to a controller under evaluation is increased or decreased depending on whether it exhibited desirable or undesirable behaviour. The algorithm awards "rat-like" behaviour such as predominantly staying within the confines of the enclosure (anxious exploratory behaviour) and close to walls, and punishes falling off the arena, exploring the outer boundaries of the arm and wall collisions.

Each individual is evaluated after navigating through the virtual EPM for 30 seconds.

Successive generations are created through selection and reproduction of the previous generation.

Based on the fitness of these individuals, the best individuals are selected by elitism and the remaining individuals are selected through tournament selection.

In elitism, a proportion of the best individuals are copied directly to the new generation, and the percentage for which this is done is dictated by the variable *elitism*.

In tournament selection, a number of individuals are chosen randomly, according to the variable *tournamentSize*. These individuals are sorted by their fitness and the best two are chosen as parents.

A new individual is reproduced using uniform crossover. In uniform crossover, there is a 50% probability that the current gene is selected from one parent and not the other.

Gaussian mutation is also applied with a probability dictated by the variable *muteRate* (i.e. the number of individuals in a generation to be mutated) and with severity dictated by the variable *severity* (i.e. the number of genes in the individual to be mutated). The variables mentioned above are all user definable.

When the best controller is evolved, the final generation of individuals is sorted by fitness and saved to an .xml file for analysis and reuse.

3.4 Simulation supervisor

The supervisor acts as a tool to guide and control the simulation. The user first sets the desired mode of operation (demo or evolution). Then the supervisor sets the time step of the current simulation as well as the duration of time each controller should run for, and resets the robot to its initial position.

4. Results

A controller which mimicked the approach-avoidance behaviour of a rat in the EPM was successfully evolved.

With this controller, the virtual rat moves along the enclosed walls as it explores the maze, peeks onto the unenclosed positions but slowly turns away back into the enclosed space.

Shown in Table 1 are the algorithm parameters used during evolution.

Variables	
popsize	50
generations	50
tournamentSize	3
elitism	0.2
muteRate	0.02
severity	0.4

Table 1. Genetic Algorithm Parameters

5. Discussion

One of the first challenges faced was how to keep track of the position of the robot. Since the robot's position is required to evaluate the fitness of the controller, this was important.

Due to its accuracy and ease of implementation, a GPS unit was decided upon. However an alternative would be to change the colour of the maze floor and use the ground sensors to detect the change in colour. In this way, the maze could still be discretised into zones.

Using the method implemented, however, dividing the map into more positions could evolve a better controller.

Possibly, this method would be cheaper to implement in a real robot, since indoor GPS equipment is expensive. In addition, the ground sensors could be used to detect the absence of a floor at the edge of the unenclosed spaces, which would be analogous to a rat's whiskers.

The neural network topology if fixed, however different architectures were considered.

A recurrent network was contemplated; however, this would have increased the complexity of both neural network computation and behavioural evolution. In addition to this, the running speed could be effected. However, more hidden layers and recurrence would represent more memory and information processing capability and would possibly result in evolving a good solution quicker. In addition to this, simultaneously evolving the weights, the architecture of the ANN, and the activation function would yield better results, at the cost of time and complexity.

Having fewer input nodes in the ANN and using less IR sensors on the robot was also considered, as rats use mostly the whiskers and eyes to gauge their environment.

Different numbers of outputs were considered too, such as four outputs, each representing a particular action for the robot to take, such as forward, left, right and stay in the same place.

The final behaviour may have been more robotic than the one in our case, but the controller may have evolved quicker.

Initially, the ANN was running on the robot, and the GA was running separately, through the supervisor. A two-way communication network was implemented within the simulation, with the supervisor transmitting the weights to be tested to the robot and the robot transmitting its GPS co-ordinates to the supervisor. Unfortunately, this resulted in an extremely slow running speed, even using the simulation's "fast mode". Therefore, it was decided that the GA should run on the robot too. This increased the simulation speed ten fold. It was thought this was probably due to the removal of the emitter and receiver previously needed when transmitting data between the robot and the supervisor.

For the fitness function, an extra component was implemented, which recorded the number of times the e-puck visited each position, and punished the individual if it visited one position frequently and never visited others. However, this seemed to have no effect on the evolution of the controller, and it was decided that assigning more points to positions further away from the starting position was simpler and more effective.

For more robust evolution, the robot's initial starting position could also be altered.

6. Conclusion

This experiment aimed to examine an application of evolutionary robotics techniques. A computational model of a rat exhibiting the approach/avoidance conflict was obtained. In particular, a genetic algorithm was used to evolve the controller of an E-puck robot within simulation.

The behaviour exhibited by the robot is similar to that of a real rat, though some areas where improvement could be were highlighted.

Further work could be done within the genetic algorithm by modifying the type of selection and crossover used to generate a new population, as well as changing the population and generation size and allowing evolution to progress for longer. In addition, the hardware used by the robot to operate could be evolved. This could be achieved by giving the algorithm access to all available sensors and letting the evolution process select which sensors are best suited. This could lead to efficiency increases.

References

- [1] K. C. Montgomery. The relation between fear induced by novel stimulation and exploratory drive. *Journal of Comparative and Physiological Psychology*, 48 (4):254–260, 1955. ISSN 0021-9940. doi: 10.1037/h0043788. URL <http://content.apa.org/journals/com/48/4/254>.
- [2] Matt Carter and Jennifer C. Shieh. Animal Behavior. In *Guide to Research Techniques in Neuroscience*, pages 39–71. Elsevier, 2010. doi: 10.1016/B978-0-12-374849-2.

- 00002-1. URL <http://linkinghub.elsevier.com/retrieve/pii/B9780123748492000021>.
- [3] Alicia A Walf and Cheryl A Frye. The use of the elevated plus maze as an assay of anxiety-related behavior in rodents. *Nature protocols*, 2(2):322–8, 2007. ISSN 1750-2799. doi: 10.1038/nprot.2007.44. URL <http://dx.doi.org/10.1038/nprot.2007.44>.
- [4] Richard G Lister. The use of a plus-maze to measure anxiety in the mouse. *Psychopharmacology*, 92(2):180–185, 1987. ISSN 0033-3158. doi: 10.1007/BF00177912.
- [5] Munekazu Komada, Keizo Takao, and Tsuyoshi Miyakawa. Elevated plus maze for mice. *Journal of visualized experiments : JoVE*, (22):1–4, 2008. ISSN 1940-087X. doi: 10.3791/1088. URL <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2762911{%&}tool=pmcentrez{%&}rendertype=abstract>.
- [6] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, vol. 1, num. 1, p. 59-65, 2009.
- [7] Olivier Michel. Webots TM : Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42, 2004. ISSN 1729-8806. doi: 10.1.1.86.1278.
- [8] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*, 3rd edition. 2009. ISBN 9780136042594. doi: 10.1017/S0269888900007724. URL <http://portal.acm.org/citation.cfm?id=1671238{%&}coll=DL{%&}dl=GUIDE{%&}CFID=190864501{%&}CFTOKEN=29051579{%&}5Cnpapers2://publication/uuid/4B787E16-89F6-4FF7-A5E5-E59F3CFEFE88>.
- [9] Simon Haykin. *Neural Networks and Learning Machines*, volume 3. 2008. ISBN 9780131471399. doi: 978-0131471399.
- [10] A Fraser and D Burnell. *COMPUTER MODELS IN GENETICS*. New York: McGraw-Hill, 1970. ISBN 0070219044.
- [11] Ariadne A. Costa, Silvio Morato, Antonio C. Roque, and Renato Tinós. A computational model for exploratory activity of rats with different anxiety levels in elevated plus-maze. *Journal of Neuroscience Methods*, 236:44 – 50, 2014. ISSN 0165-0270. doi: <http://dx.doi.org/10.1016/j.jneumeth.2014.08.006>. URL <http://www.sciencedirect.com/science/article/pii/S0165027014002908>.
- [12] Ariadne A. Costa and Renato Tinós. Investigation of rat exploratory behavior via evolving artificial neural networks. *Journal of Neuroscience Methods*, 270:102 – 110, 2016. ISSN 0165-0270. doi: <http://dx.doi.org/10.1016/j.jneumeth.2016.06.010>. URL <http://www.sciencedirect.com/science/article/pii/S0165027016301315>.
- [13] Udo Seiffert. Multiple Layer Perceptron training using genetic algorithms. *European symposium on artificial neural networks*, 9:159–164, 2001. URL <https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2001-460.pdf>.