Independently Optimised Artificial Neural Network Interactive Searching Algorithm (IOANNIS)

Thomas Lawson, *up780962*, and Yordanka Popova, *up782716*.

(Coursework for ENG621 - Artificial Intelligence)

***Abstract*—Over the past few decades the density of compute power has dramatically increased, whilst at the same time it's cost has dramatically decreased. As a result, much research has been done into the field of artificial intelligence, allowing it to be taken from a concept to a working product thanks to the increased capability and availability of compute power.**

This research will look to propose and analyse an approach to interactive and real time route planning for a robot designed for a capture the flag like task, meaning it will have to locate and retrieve an object returning it to a designated point, whilst all the time avoiding people moving in it's path.

This paper presents and analyses an approach to developing a real time navigational algorithm, named IOANNIS. The paper will present and critically analyse the algorithm's ability to effectively train an artificial neural network using a genetic algorithm, and the time it would take to come up with a viable neural network capable of safely and effectively completing the task.

***Index Terms*—Artificial Intelligence, Path Finding, Artificial Neural Networks, Genetic Optimisation.**

I. Introduction

WITH this paper we hope to introduce an approach to effectively train a tracked robot to successfully navigate a foreign environment in order to retrieve and return an object.

Currently for many tasks carried out by robots a human is required to control and monitor it's operation. This can be expensive and time consuming. However the human touch can be seen as necessary as humans have a better inherent ability to react to unplanned events in the environment.

Autonomous robotic navigation is not a novel concept however, and in recent years with the reduction in price of computing and increase in density of computing power many companies have successfully demonstrated autonomously navigating systems capable of reacting to a real-world environment. For example, Boston Dynamics have created their handle robot, capable of box handling in a warehouse. Or there's Tesla, who have created an autonomous self driving car.

The objective of this research is to present and evaluate an approach for training an Artificial Neural Network quickly and effectively on any platform it is presented with, so long as it's interactions are well defined. We hope this research will help tackle an issue with Neural Networks where they will be able to handle the environment they are trained in, and not work well beyond it. With faster training times

This paper proposes our algorithm IOANNIS and hopes to demonstrate an adaptable approach to the development of a navigation system powered by Artificial Intelligence. It will be meant to be portable, meaning it can be trained on any device, and do so in a fast and efficient manner. The navigation system it produces should be able to avoid changing and moving obstacles in a foreign environment, while searching for an object it is meant to retrieve.

For the purpose of this research a simulation will be created using Unity, with the simulated robot being based on the University of Portsmouth's IRIS robot (figure 1) provided by the University of Portsmouth.

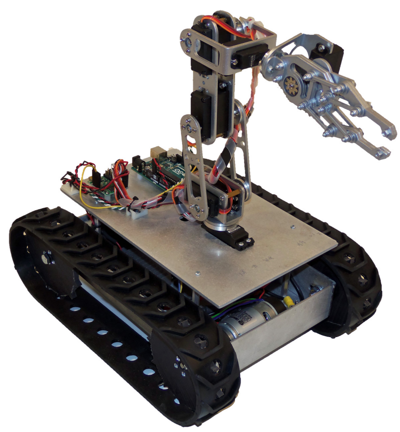

Fig. 1. Iris Robot (Maybe - It's from Google)

The rest of this paper will be organised as follows. Related work will be critically reviewed in section II. Section III will discuss the design of the algorithm IOANNIS. Section IV will discuss the implementation of the algorithm. Experimental results will be presented in section V. Lastly, section VI will present our findings and conclusions, as well as propose areas for future work.

II. Literature Review

III. Design

A. Objectives

The main purpose of this research is to develop an approach to quickly training a robot to navigate successfully. To this end we will define the objectives of our research as follows.

1) The algorithm **shall** be capable of training a neural network to navigate an autonomous robot.
2) The algorithm **shall** be able to successfully train the robot in a quick and efficient manner.
   a) Successfully train **shall** be defined as the to be able to accomplish the retrieval task in a safe manner, meaning it does not collide with anything in the environment.
   b) Successfully train **shall** also be defined as to be able to accomplish it's task in an expedient manner. i.e. complete its task without unnecessary movements or stopping.
   c) Quick and Efficient **shall** be defined as a successful candidate neural network being produced within an acceptable number of generations.
3) The algorithm **shall** be portable. Meaning the algorithm should not just be able to train for one type of robot. For example, the algorithm should be able to train on robots with reversible tracks, four wheels, etc.

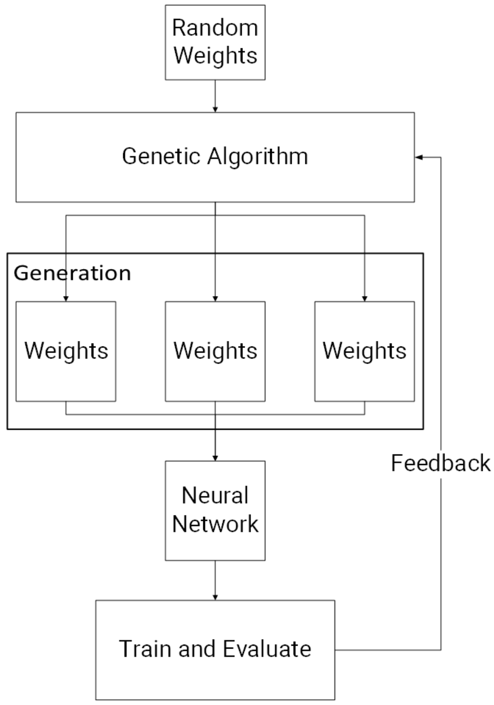

Fig. 2. Stages of the IOANNIS algorithm

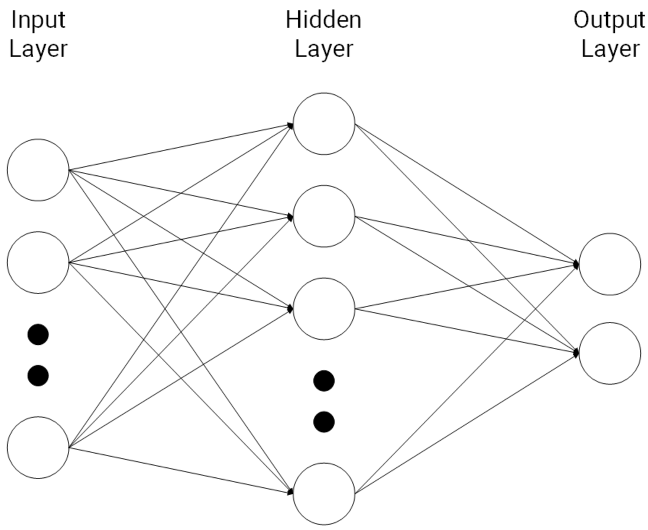

Fig. 3. Representative Neural Network

B. IOANNIS Algorithm structure

For training the neural network, a genetic algorithm was selected. This is because with a task such as navigation, it is hard to use back propagation for training, as that would require for each decision, the actual correct decision to be determined and then re-train the network. When using a genetic algorithm to train a fitness function can be defined, this can evaluate the performance of the algorithm, and then whichever weights work better towards the defined goals will be kept on to further improve the robot.

The stages of the algorithm can be seen in figure 2 and can be summarised as follows.

1) Produce an initial population of weights.
2) For each set of weights in the population initialise a neural network using those weights and use it to operate the simulated robot.
3) Score the simulated robots and provide feedback to the genetic algorithm.
4) Using feedback from the simulation the genetic algorithm will use its fitness function to rank each set of weights.
5) If the end condition is met, return the fittest set of weights.
6) Otherwise crossover or mutate the weights according to the crossover and mutations rates.
7) Return to item 2 and repeat the process with the new population

The following sections will outline the design of the neural network and genetic algorithm that will train it.

C. Neural Network Design

The neural network will use l layers an input layer I and output layer O and x hidden layers H_x. The weights for each node W_{li}, where l represents the layer and i represents the weight's index, will be populated and trained using a genetic algorithm described in section III-D below.

For this task it has been elected to use an input layer consisting of x nodes, one for each sensor input to the robot. We have then selected to use y hidden layers of z nodes, this is because of....

There will be two output nodes. Each output will have applied to it a tan-sigmoid transfer function (figure 4) to smooth the output. We have selected the tan-sigmoid transfer function as one output will represent each track, with a positive output directing the track to run forward, and a negative output directing the track to run backwards. The magnitude of each output will be used to control the speed at which the track runs.

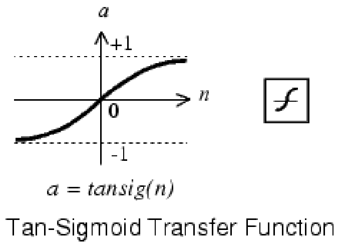

Fig. 4. Tan-Sigmoid Transfer Function

D. Genetic Algorithm Design

The genetic algorithm will be used to hone the weights used by the neural network described above.

1) Genetic Operators: Due to the nature of this task, we have elected to use three genetic operators. We have selected the three operators below, as due to the nature of this task, there is no need to use the operators for coercion where there are restrictions placed on an individuals viability. All candidates produces will be viable.

1) *Crossover Operator:* This will select another individual from the population and split it at a randomly selected point. The first section of genome A will be combined with the second section of genome B, and the first section of genome B will be combined with the second section of genome A. This will produce two children to be passed to the next generation.
2) *Mutation Operator:* This will randomly select a weight in the individual's genome, and randomly assign it a new weight between 0.00 and 1.00. The resulting child will then be passed into the next generation
3) *Pass-through Operator:* This operator will pass the selected individual directly through to the next generation without change. This is the default operator and will be selected if the other two are not.

Each operator will have a chance of being selected proportional to it's selection probability.

2) Fitness Function: The fitness function to evaluate the fitness of an individual $f(I)$ shall be defined as follows.

$$f(I) = c + t \tag{1}$$

Where c represents the number of collisions the robot makes, and t represents the time it takes to complete it's task.

3) End Condition: To determine when the optimal solution has been found, there are two options. Firstly, it is possible to limit to a fixed number of generations, after which the fittest individual will be presented as the optimal solution. The other is to define an end condition, for this algorithm that could be defined as having an adaptable number of generations with at least one genome resulting in no collisions.

For the purpose this algorithm has been produced for, it is pertinent that the selected outcome prioritises safety of the robot and it's surroundings, as such the proposed algorithm will used a goal based end condition, and not be limited by the number of generations. This is because if a genome leads to collisions, this would be undesirable and unfit for purpose.

E. Robot Control

F. Simulator

IV. Implementation

V. Results

VI. Conclusions

Acknowledgement

The authors would like to thank...

Contributors

Thomas Lawson Is a final year student studying for his undergraduate degree in computer engineering at the University of Portsmouth.

He has experience in industry working with the air navigation service provider NATS EnRoute Limited; where he worked on the project delivering the next generation flight data processor 'iTEC' as part of NATS' Deploying SESAR initiative. iTEC is part of NATS' own commitment to EUROCONTOL's larger Single European Skies Advanced Research (SESAR) Joint Undertaking, and is produced in collaboration with the German Deutsche Flugsicherung and Spanish ENAIRE ANSPs.

His final year research project is attempting to propose a method of airspace design using artificial intelligence. He is interested in research concerning air traffic management, and artificial intelligence.

Yordanka Popova Is studying for her...