Analysis Document

“AI Plays Flappy Bird Using Neural Networks and Genetic Algorithms”

The system presented in this project involves a goal based agent playing the popular video game “Flappy Bird” with the target of achieving the highest possible score. Implementation involves neural networks for each of the units, i.e. birds, consisting of three layers namely input layer, hidden layer and output layer. These artificial neural networks are trained using genetic algorithm, by initially using a random neural network to create a new population of 10 units (birds), which then play the game simultaneously using their own neural networks. Afterwards, fitness of each bird is calculated. When all the units are killed, the current population is converted to the next one using genetic algorithm operators.

Now, to create an artificial brain for this system we need to simulate neurons and connect them to form a neural network.

A generic artificial neural network consists of an input layer, one or more hidden layers and an output layers. Each layer has a number of neurons. Input and output neurons are connected directly to an external environment. Finally, hidden neurons are connected between them.

In this project, each unit (bird) has its own neural network used as its AI brain for playing the game. It consists of 3 layers as follows:

1. An input layer with 2 neurons representing what a bird sees:
    - horizontal distance to the closest gap
    - height difference to the closest gap
2. A hidden layer with 6 neurons.
3. An output layer with 1 neuron which provides an action as follows:
    - If output > 0.5 then flap else do nothing

Scope

The scope of the system is limited to generating a population of birds capable of achieving the highest possible scores in the game. Although, this is just the scope in the bluntest of senses.

As far as AI is concerned, the scope for “Genetic Algorithm” to be implemented in this system is really wide as it can be modified for implementation in the following fields:

1. Automotive Design
2. Engineering Design
3. Robotics
4. Evolvable Hardware
5. Encryption and Code Breaking
6. Computer - Aided Molecular Design
7. Gene Expression Profiling
8. Optimizing Chemical Kinetic Analysis
9. Finance and Investment Strategies
10. Marketing and Merchandising

Feasibility Report

1. **Technical Feasibility: -**

   The system requires the following technical inputs:

   a. 3-4 weeks of programming work
   b. Hardware (Laptop/PC) with decent specifications (RAM 4GB or more, Graphics Card 2GB or more)

   Since, both of the major requirements are at our disposal, the project is technically feasible.

2. **Operational Feasibility: -**

   This system can:

   a. Play the game "Flappy Bird" using genetic algorithm, with the goal of generating a population of birds capable of achieving the highest score, i.e. surviving the longest in the game.
   b. Tabulate the fitness scores of birds of a particular generation.

   This system cannot:

   a. Play any other game apart from "Flappy Bird", although, the same game playing approach can be used to play a number of other games.

3. **Schedule Feasibility: -**

   The projected time to be taken for the development of this system should be well within the deadline date as the project timetable is fairly reasonable, in spite of the technical naivety of the programmers and the oncoming mid semester examination.

Literature Study

1. **Reverse Engineering Financial Markets with Majority and Minority Games Using Genetic Algorithms**

   Authors – J Weisinger, D Sornette, J Satinover from ETH Zurich, Zurich, Switzerland

   Link:- https://rdcu.be/7Ky6

   Similar to the implementation in the ongoing project, these researchers, used genetic algorithm in a game playing environment to reverse engineer real world financial time series. They further used their outcomes to analyze stock markets.

2. **Genetic algorithm learning and evolutionary games**

   Author – Thomas Reichmann

   Link:- https://doi.org/10.1016/S0165-1889(00)00066-X

This paper links the theory of genetic algorithm (GA) learning to evolutionary game theory. In this paper it is shown that economic learning via genetic algorithms can be described as a specific form of an evolutionary game. Through this paper a concept of evolutionary superiority and evolutionary stability of genetic populations is developed, which allows for a comprehensive analysis of the evolutionary dynamics of the standard GA learning processes.

3. **Playing to learn: case-injected genetic algorithms for learning to play computer games**

   Author – S.J. Louis, C. Miles from University of Nevada

   Link:- https://ieeexplore.ieee.org/document/1545942

   In this article, the researchers use case-injected genetic algorithms (CIGARs) to learn to competently play computer strategy games. CIGARs periodically inject individuals that were successful in past games into the population of the genetic algorithm (GA) working on the current game, biasing search toward known successful strategies. Hence, making GA more efficient.

4. **Infinite Mario Bros. AI using Genetic Algorithm**

   Authors – Ng Chee Hou, Niew Soon Hong, Chin Kim On, Jason Teo from School of Engineering and Information Technology, Universiti Malayasia Sabah

   Link: - https://ieeexplore.ieee.org/document/6089330

   In this paper, a finite state machine which is suitable to be used for Infinite Mario Bros game is proposed. The Genetic Algorithm (GA) is used along with the proposed finite state machine to evolve an AI agent that is capable to passing some levels of the game. The experimentation results showed that the finite state machine evolved with GA is able to create a competitive game bot that can pass through at least 3 levels of different game maps. The generated AI controller can guarantee to accomplish the tasks for some levels.
   (This paper is the most relatable to the system presented in this document)