**AGENT OPTIMIZATION USING GENETIC ALGORITHM**

**ABSTRACT:**

In computer science and operations research, a genetic algorithm (GA) is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA).[1] The objective of this project is to create a dynamic system to improve agents in a given environment to perform a certain set of tasks using the genetic algorithm approach to optimize the neural networks to find suitable architecture and optimal agents. This Program is constructed where we train a Neural Network for it to play Flappy Bird the Game. It’s inspired by the natural processes that drive biological evolution and although it sounds complicated, the principles behind it are actually quite simple.

**INTRODUCTION:**

A neural network is a series of algorithms that endeavours to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so, the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems. A neural network evaluates unearths opportunities for making trade decisions based on the data analysis. The networks can distinguish subtle nonlinear interdependencies and patterns other methods of technical analysis cannot.

Genetic algorithms are stochastic search algorithms which act on a population of possible solutions. They are loosely based on the mechanics of population genetics and selection. They are probabilistic search methods; this means that the states which they explore are not determined solely by the properties of the problems. A random process helps to guide the search. Genetic algorithms are used in artificial intelligence like other search algorithms are used in artificial intelligence — to search a space of potential solutions to find one which solves the problem.

****

**LITERATURE SURVEY:**

1). Deep reinforcement learning using genetic algorithm for parameter optimization

-IEEE (Feb 2019)

In this paper the genetic algorithm was used to speed up the learning agent such fetch-reach, slide, push, pick AND place, and door opening in robotic manipulation tasks.[2]

2). Scalability and optimisation of a committee of agents using genetic algorithm

- EEE Dept Imperial College London

Here a population of committees of agents is used to simulate the stock market. Each committee of agents is viewed as a player in a game and it was seen how the players compete and interact.[3]

3). Genetic algorithm optimisation of an agent-based model for simulating a retail market

- AJ Heppenstall (University of Leeds) (Dec 2007)

This paper explores the parameterisation and verification of the retail petrol market through data analysis by the use of a genetic algorithm.[4]

4). UAV cooperative multiple task assignments using genetic algorithms

- IEEE (June 2005)

here genetic algorithm was used for assigning multiple agents to perform multiple tasks on multiple targets.[5]

5). Playing games with genetic algorithms

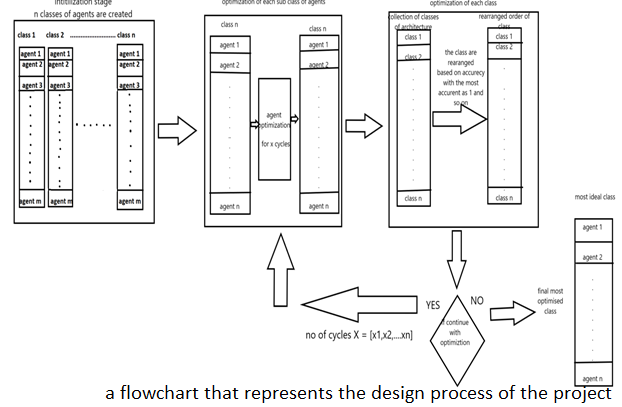
- Robert Marks (Jan 2002)

here Genetic Algorithm was used to seek better strategies in playing the repeated Prisoner's Dilemma.[6]

6). Generating war game strategies using a genetic algorithm

- Robert Mccartney (Feb 2002)

Unlike most games which have fixed rules, the rules for war games can contain uncertainty. This uncertainty makes war games difficult to address with methods typically used for playing games by machine. Thereby the use of genetic algorithms for generating war game strategies was proposed.[7]

****

**PROPOSED SYSTEM:**

There are four stages in total and they are:

* Initialization phase: Here the system randomly creates agents for specific architectures.
* Training phase: This phase takes the agents and architectures and agents and trains each individual architecture a set number of times.
* Evaluation phase: Here each architecture is sorted based on the rate of improvement after the training phase and is then given the next number of times is is allowed to train for example the most improved will be given the ability to train 10 times the next 5 and so on and so forth.
* The control loop: This decides when the maximum number of training cycles are done and exits both the training phase and evaluation phase to give us the final agent which is most optimal for the environment.

**LIMITATIONS:**

1) The downtime between evaluation cycles waste resources.

2) Simple problems would not require such systems as a simple network would work fine.

3) Multiprocessing systems are required else this adds very little benefit.

4) Developers must have an intuitive sense of the problem to create the agents correctly.

**CONCLUSION AND FUTURE ENCHANCEMENT:**

We were successfully able to train the agents but the game had few glitches, all the agents didn't have the same score and was slow .The overall results shown tells us that we are headed in the right direction so our future work would be based on how to rectify these errors and develop a better working model.

**ACKOWNLEDGEMENT:**

Special thanks to Dr. Anupama H S for guiding us in the right direction and also helping us learn a lot during the implementation stage and we also would like to take John Holland for inspiring us to create a project based on genetic algorithms.

**REFERENCES:**

[1].https://en.wikipedia.org/wiki/Genetic\_algorithm#CITEREFMitchell1996

[2].https://ieeexplore.ieee.org/abstract/document/8675632

[3].https://arxiv.org/pdf/0705.1757.pdf

[4].https://www.researchgate.net/publication/23541636\_Genetic\_Algorithm\_Optimisation\_of\_An\_Agent-Based\_Model\_for\_Simulating\_a\_Retail\_Market

[5].https://ieeexplore.ieee.org/abstract/document/1470429

[6].https://www.researchgate.net/publication/228697986\_Playing\_Games\_with\_Genetic\_Algorithms

[7].https://www.researchgate.net/publication/3949330\_Generating\_war\_game\_strategies\_using\_a\_genetic\_algorithm