Department of Artificial Intelligence and Machine Learning

AGENT OPTIMIZATION USING GENETIC ALGORITHM

Submitted By

Student Name: Mrinal Durani

USN: 1BY19AI034

Student Name: Puja S.

USN: 1BY19AI040

Student Name: Rtwick George Moses

USN: 1BY19AI043

Student Name: Siddharth Arora

USN: 1BY19AI054

Under the guidance of:

Dr. Anupama H S

Associate Professor of AI&ML

BMSIT&M

2020-21

EVEN Semester

Contents

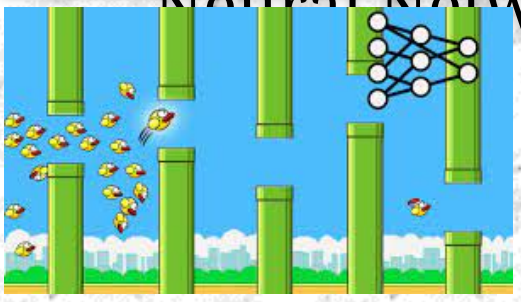
- Introduction
- Objective
- Literature Survey
- Existing Systems
- Limitations of Existing Systems
- Proposed Methodology
- System Elements
- System Architecture

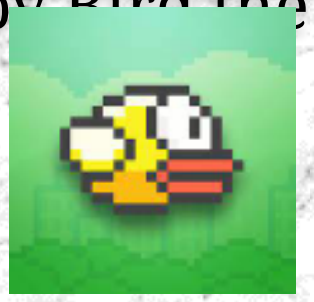
Introduction

- A **Neural Network** is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.
- Genetic Algorithms are probabilistic search methods; i.e. the states which they explore are not determined solely by the properties of the problems rather a random process helps to guide

Objective

- 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





Literature Survey

1. Deep reinforcement learning using genetic algorithm for parameter optimization -IEEE (Feb 2019)
2. Scalability and optimisation of a committee of agents using genetic algorithm - EEE Dept Imperial College London (May 2007)
3. Genetic algorithm optimisation of an agent-based model for simulating a retail market - AJ Heppenstall (University of Leeds) (Dec 2007)
4. UAV cooperative multiple task assignments using genetic algorithms

   - IEEE (June 2005)
5. Playing games with genetic algorithms

   - Robert Marks (Jan 2002)

6. Generating war game strategies using a genetic algorithm

Existing Systems

- In 1987 the first published research appeared which used the Genetic Algorithm[5].
- Current systems use neural networks or other means to create agents for certain agents[2]
- NP problems also use genetic algorithm to estimate answers[4]
- Dynamic environments which have the issue of uncertainty like war games also exploit it and can be successfully navigated by simulating using the genetic approach[6]

Limitation of Existing Systems

The major issue for past systems is deciding the architecture of the neural network and whether to have multiple or a singular unified architecture.

Metric	Dynamic approach	Static Approach
Accuracy	Higher accuracy	lower accuracy
Computational difficulty	Lager computational Load	Lesser computational Load
Size of data stored	Lager amount of data	Lesser amount of data
Dynamic Environment	Are more adapt to these environments	Are Less adapt to these environments
Rate of improvement	Lesser likelihood to plateau	higher likelihood to plateau

Proposed Methodology

- The proposed methodology works by trying to continuously evaluate the agents and classes and allocate resources based on improvement of performance.
- Hench we hope to have the benefits of a static which is low computational load and processing .
- The flexibility offered by a dynamic system along with its accuracy , adaptability and flexibility to changing environment stimula .

Proposed Methodology

- 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 max 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.

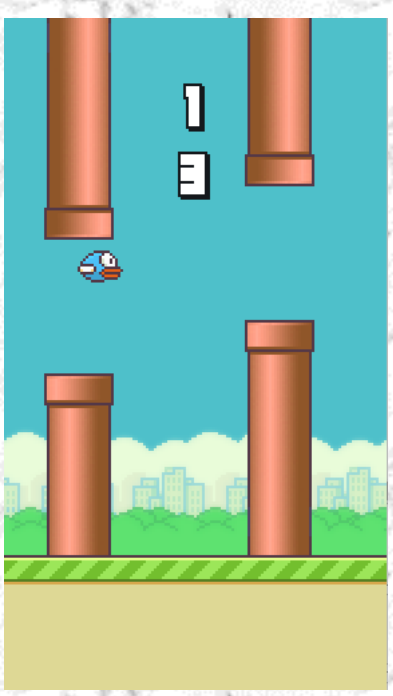
System Elements

Agent

Class of agents with identical architecture

Collection of different classes with various architectures

Gameplay Environment

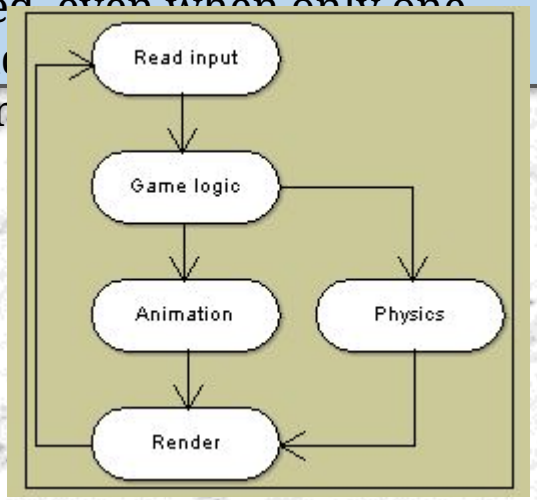


System Elements

Multithreading:

1).In computer architecture, multithreading is the ability of a central processing unit (CPU) (or a single core in a multi-core processor) to provide multiple threads of execution concurrently, supported by the operating system.

Advantages: If a thread gets a lot of cache misses, the other threads can continue taking advantage of the unused computing resources, which may lead to faster overall execution, as these resources would have been idle if only a single thread were executed. ***Disadvantages***: Multiple threads can interfere with each other when sharing hardware resources such as caches or translation lookaside buffers (TLBs). As a result, execution times of a single thread are not improved and can be degraded, even when only one thread is executing, due to lower frequencies or ad
that are necessary to accommodate thread-switchi



System Elements

Pygame:

Pygame was originally written by Pete Shinners to replace PySDL after its development stalled. It is a cross-platform set of Python modules designed for writing video games.

Pygame uses the Simple DirectMedia Layer (SDL) library, with the intention of allowing real-time computer game development without the low-level mechanics of the C programming language and its derivatives.

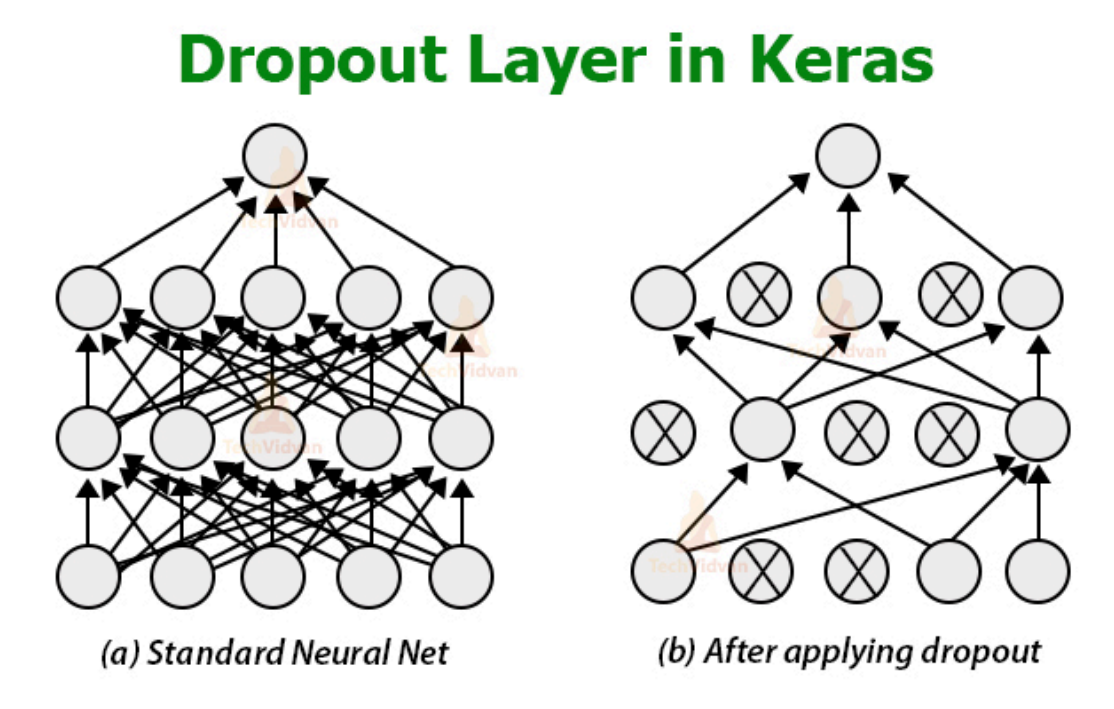
Notable games using Pygame :-

- Flappy Bird
- Snake and Ladders
- Retro Racing Games

System Elements

Keras

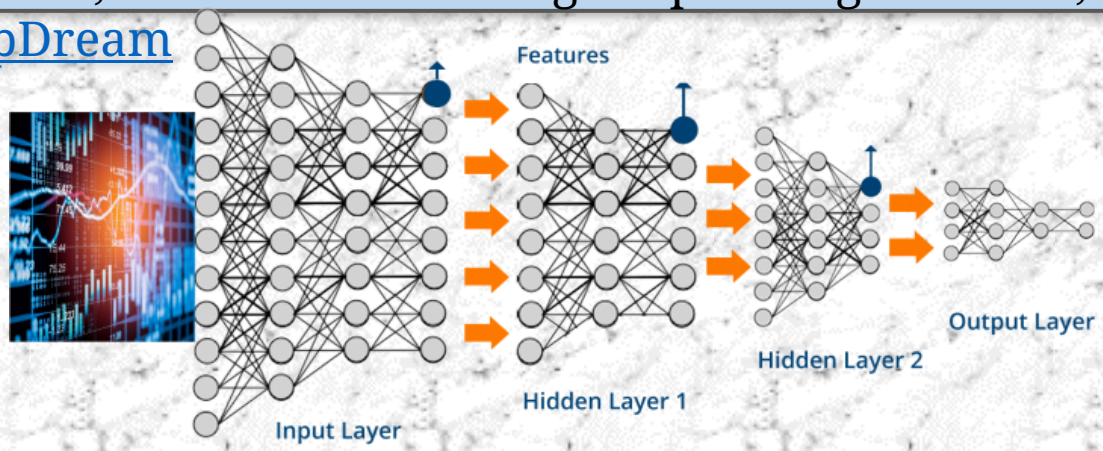
It is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. It helps us to develop and evaluate deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code.



System Elements

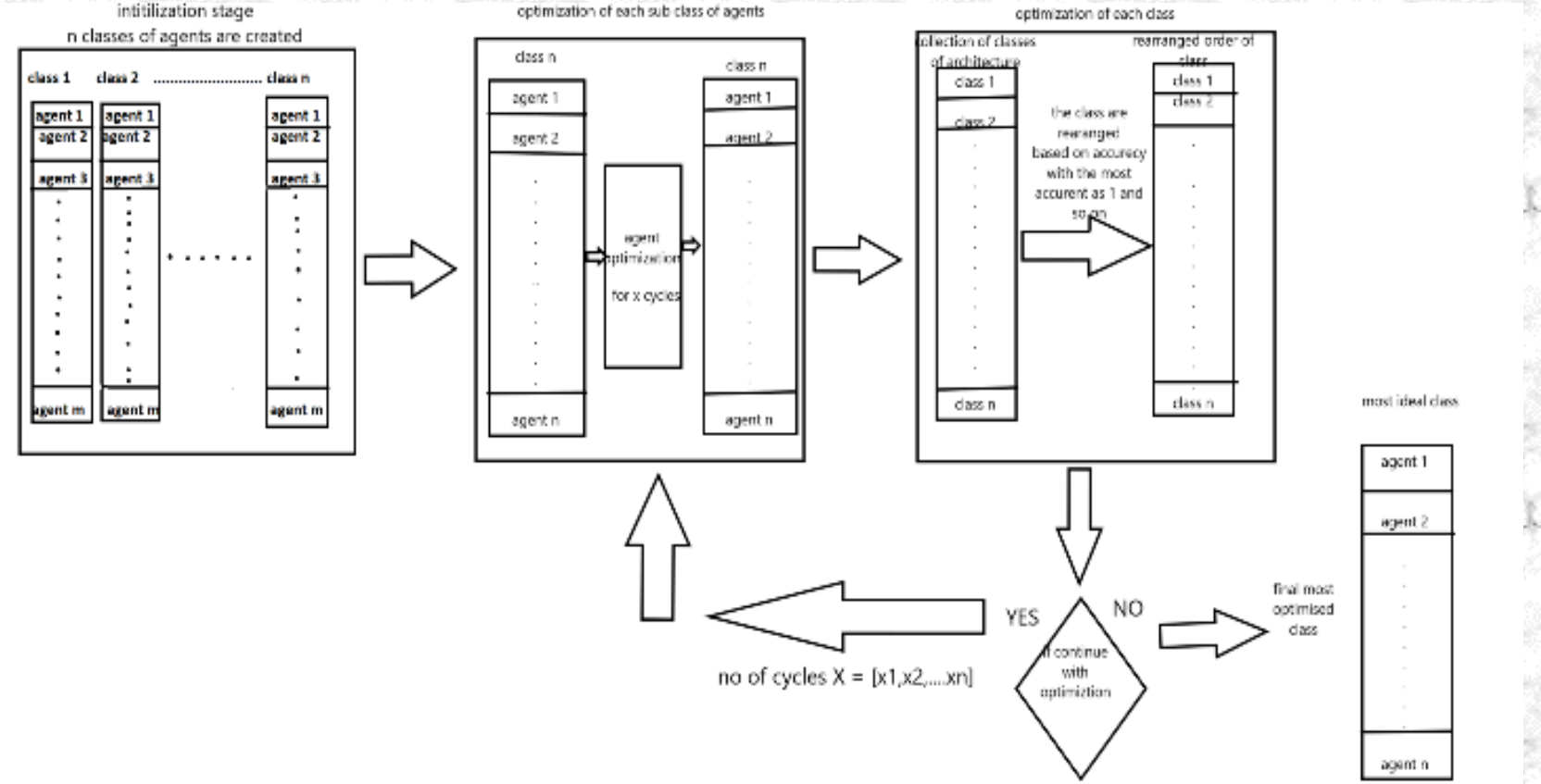
TensorFlow

TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.
TensorFlow is a symbolic math library based
on dataflow and differentiable programming. It is used for both research and production at Google.
It was developed by the Google Brain team for internal Google use and released under the Apache License 2.0 in 2015.
Applications : Among the applications for which TensorFlow is the foundation, are automated image-captioning software, such as DeepDream



Department of Artificial Intelligence and Machine Learning

System Architecture



Thank You