Genetic algorithms based AI to play Flappy birds

Devishi Kesar
Computer science engineering
IIIT-Delhi
Delhi, India
devishi15024@iiitd.ac.in

Siddharth Arya
Computer science engineering
IIIT-Delhi
Delhi, India
siddharth14103@iiitd.ac.in

***Abstract*— The aim of this project is to implement, compare and improve some algorithms, based on their performance on flappy bird. This would improve our understanding of the algorithms and implementation of algorithms in the field of computer games.**

I. Introduction

Reinforcement learning algorithms are one of the most popular choices when it comes to applying artificial intelligence for games. The main aim of this paper was to implement genetic algorithms for scoring with neural networks and compare the performance with other algorithms. All the algorithms were developed on the available execution of flappy birds in python by sourabhv[1].

II. Motivation

Flappy bird is a side-scroller where the player controls a bird, attempting to fly between columns of green pipes without hitting them. The game became popular because of its addictive nature and extreme difficulty. This prompted many people across the world to come up with AIs for the game. This project is another such attempt where we are implementing two AI algorithms and comparing them. This would improve our understanding of the algorithms and implementation of algorithms in the field of computer games.

III. Objective

- To generate artificially intelligent agents that can play the game of flappy bird better than humans.
  AI algorithms used:
  1. Deep Q-learning
  2. Genetic Algorithm with Neural Network
- To compare the algorithms on the basis of their efficiency and score achieved.

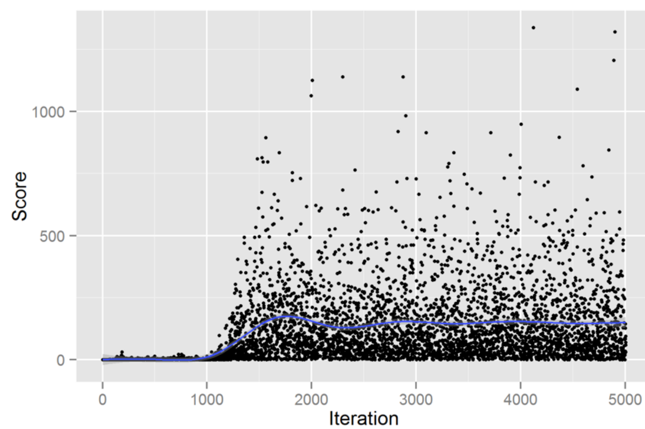

Fig. 1. Performance of q learning over iterations.

IV. Algorithms

- Genetic algorithm: It is a metaheuristic used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection. The algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population ”evolves” toward an optimal solution.
- Deep Q Learning: It is a convulational neural network trained with a variant of Q learning. Through trials-and-errors, Q learning finds a Q-value for each state-action pair. This Q-value represents the desirability of an action given the current state. Over time, if the world is static (i.e. the physics or the cause-and-effects don’t change), the Q- values converge and the optimal policy of a given state is given by the action with the largest Q-value. Update Step:

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[r_t + \gamma max_a(Q(s_{t+1}, a))]$$

V. Results

A. Explanation

1) Genetic Algorithm: The algorithm has been developed over a neural network with three layers. The input layer consists of 2 nodes with the position of pipes and position of bird, the hidden layer consists of 3 nodes and the output has 1 node telling the bird whether to flap or not. The weight of the nodes are updated using the genetic algorithm. The implementation of genetic algorithm was done using tournament selection of 10 birds with 15 in every generation and top 5 selected as elite.The mutation rate was 20%. Our algorithm reached a score of greater than 1043 in just 34 generations.[Fig.2]

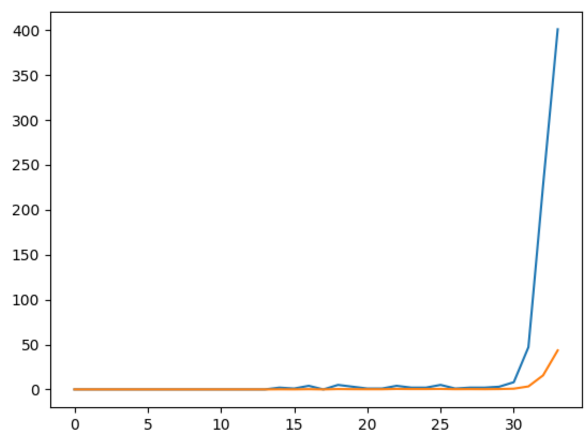

Fig. 2. Performance of genetic algorithm over generations.

2) Deep Q Learning: The Q learning algorithm took 4 days for training , the environment provides the position of pipes and bird and velocity of bird, etc as continous values, these were discretised by dividing the image into multiple frames and applying 1 action per frame. The input contains 3 inputs: pipe position, bird position, velocity of bird in particular direction; and 1 output node for deciding whether to flap or not in the frame. The game screens were preprocessed with the following steps: convert image to grayscale, resize image to 80x80 and stack last 4 frames to produce an 80x80x4 input array for network.The first layer convolves the input image with an 8x8x4x32 kernel at a stride size of 4. The output is then put through a 2x2 max pooling layer. The second layer convolves with a 4x4x32x64 kernel at a stride of 2. We then max pool again. The third layer convolves with a 3x3x64x64 kernel at a stride of 1. We then max pool one more time. The last hidden layer consists of 256 fully connected ReLU nodes. We initialize ϵ = 1 and linearly anneal ϵ from 0.1 to 0.0001 over the course of the next 3,00,000 frames. Since the model was trained for 1 frame per action, when tested the result was a linear increase for every second and pipe received(the line is a litle jagged as there is approximately 1 pipe in 2 seconds). While when tested for 2 frames and 4 frames per action(4fpa), the result was uneven as expected.

B. Inference and Comparison

The comparison of genetic algorithms with neural networks and q learning is difficult because there is no common parameter for comparison. Hence we have stated the advantages and disadvantages observed for both below.

Neural networks with GA:

Advantages:1. It is inherently parallel, so is able to operate very effectually on parallel hardware.

2. It is robust and works well over a broader class of problems with reasonable effectiveness.

3. It has adaptive learning ability.

4. Able to solve the optimization problems.

5. Easy method to understand and conveniently transfer to existing simulations and models.

Disadvantages:1. Time consuming procedure.

2.No guarantee to find best solution, so sometimes has difficulty to find exact global optimum.

3.Needs high processing time once the neural networks are massive.

4. Tough to be analyzed by human being once is trained.

Q Learning:

Advantages: 1. No model is required.

Disadvantages:1.It has shallow knowledge and cannot look ahead.

2. It can restrict ability to learn

3. Takes a lot of time to train.

The algorithm of Q learning was trained on a very weak machine. Powerful machines would obviously yield results faster. The model of Q learning has been trained on 1fpa so it was natural that 4 fpa game i.e. the hard version of the game will not show good performance. The model trained by us is susceptible to fpa. One of our motives was to develop ACO for playing flappy bird but we soon realised that the game doesn't exhibit a swarm mentality as the environment is probabilistic(it is just like applying an AI trained for a grid on another grid).

References

[1] https://github.com/sourabhv/FlapPyBird

[2] http://sciencenetlinks.com/science-news/science-updates/checkers-solved/

[3] http://blogs.discovermagazine.com/crux/2016/01/27/artificial-intelligence-go-game/#.WgcARxOCzBI I. J. Computer Network and Information Security, 2012, 7, 12-18

[4] http://blog.datumbox.com/using-artificial-intelligence-to-solve-the-2048-game-java-code/

[5] https://github.com/yanpanlau/Keras-FlappyBird

[6] https://github.com/yenchenlin/DeepLearningFlappyBird