REPORT

Application of Genetic Algorithm (GA)
Cartpole Task

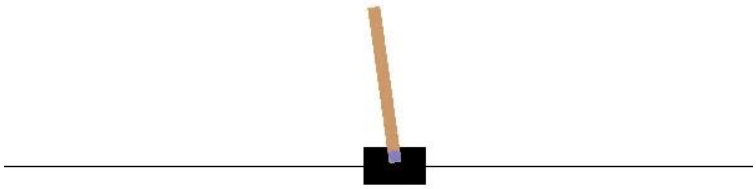
Abstract

In the world of Computer Science, there are many problems which demand massive computing along with powerful computing systems. Unfortunately, powerful computing systems could also take years to solve a particular problem. In such a situation, Genetic Algorithms can ease our task by providing usable near-optimal solutions in a short amount of time. Genetic Algorithms can be applied to variety of real-world problems in order to optimize, infer and efficiently perform searching far better than the brute force search.

Introduction

This report focuses on using a genetic algorithm in order to optimize cartpole task.

What is a Cartpole Task?



As the name suggests, in a cartpole, a pole is attached to a movable cart, which moves along a frictionless surface. The cart is moved by applying a force of +1 or -1. Initially, the pole is at upright position, and it starts falling over as the system starts. A reward of +1 is provided for every timestep for which the pole remains upright.

Problem Statement

Our goal is to balance the pole in an upright position by moving the cart left or right using a genetic algorithm.

Methodology

Environment and Libraries

For this report, 'Open AI Gym' environment was used in Python to stimulate a simple game known as 'CartPole-v0' and then Genetic Algorithm was used to automate the playing of the game. The following Python Libraries and Packages were used to accomplish this task:

- Numpy
- Gym
- Random
- Juypter Notebook
- Matplotlib

Flow of Work

Here is the summary of required steps:

1. Create some random game data.
2. Train our Genetic Algorithm model on it.
3. Run the game environment
4. Evaluate the scores
5. Plotting results.

Solution

The following fitness function was utilized for this task:

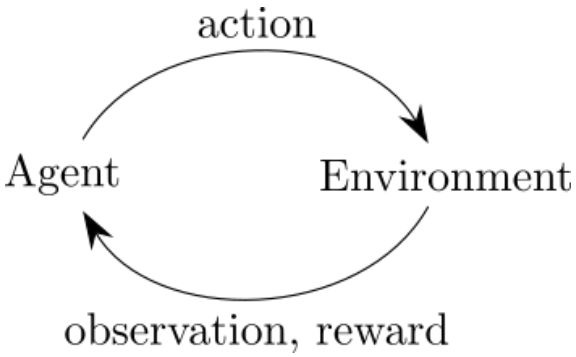
$$fitness = \sum_{i=1}^{n} w_i x_i$$

where,

$$\mathrm{w} = \text{weights}$$

$$\mathrm{x} = \text{observations}$$

The Cartpole-v0 environment in, 'Open AI Gym' consist of 4 observations and 2 actions. At first, the genetic algorithm outputs a random action and then the environment returns an observation and a reward.



Observation Table

Num	Observation	Min	Max
0	Cart Position	-2.4	2.4
1	Cart Velocity	-Inf	Inf
2	Pole Angle	-41.8º	41.8º
3	Pole Velocity at Tip	-Inf	Inf

Action Table

Num	Action
0	Push Cart to the Left
1	Push Cart to the Right

From the above table it is obvious that the action is a discrete value. However, our genetic algorithm outputs continuous value. In order to convert continuous value into discrete value, sigmoid function was used to convert it in a value between 0 and 1, and then the predict function returns a value of 0 if the

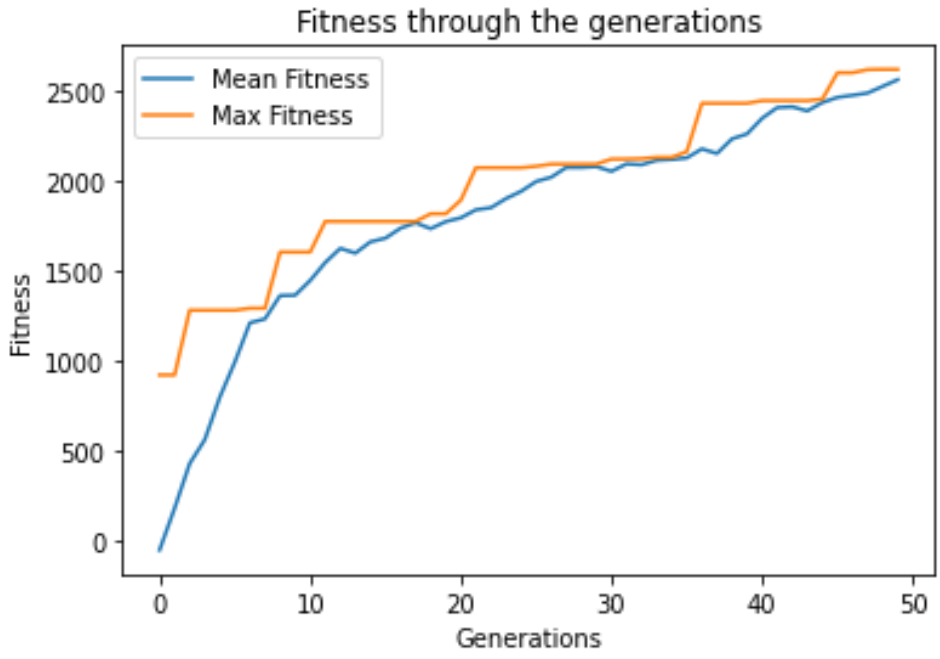
output of sigmoid function is greater than or equal to 0.5, else it returns a value of 1 if the output of sigmoid function is less than 0.5.

The training data is collected by running the game environment 10000 times for random moves. The minimum score requirement is set to 50 and if the score is greater than or equal to 50, then we are going to store every move we made thereafter. The training data is then passed into the genetic algorithm which is run for 50 generations with 8 solutions per population and 4 genes in an individual solution. An array is created to store previous observation as the current action which is stored, is made on previous observation.

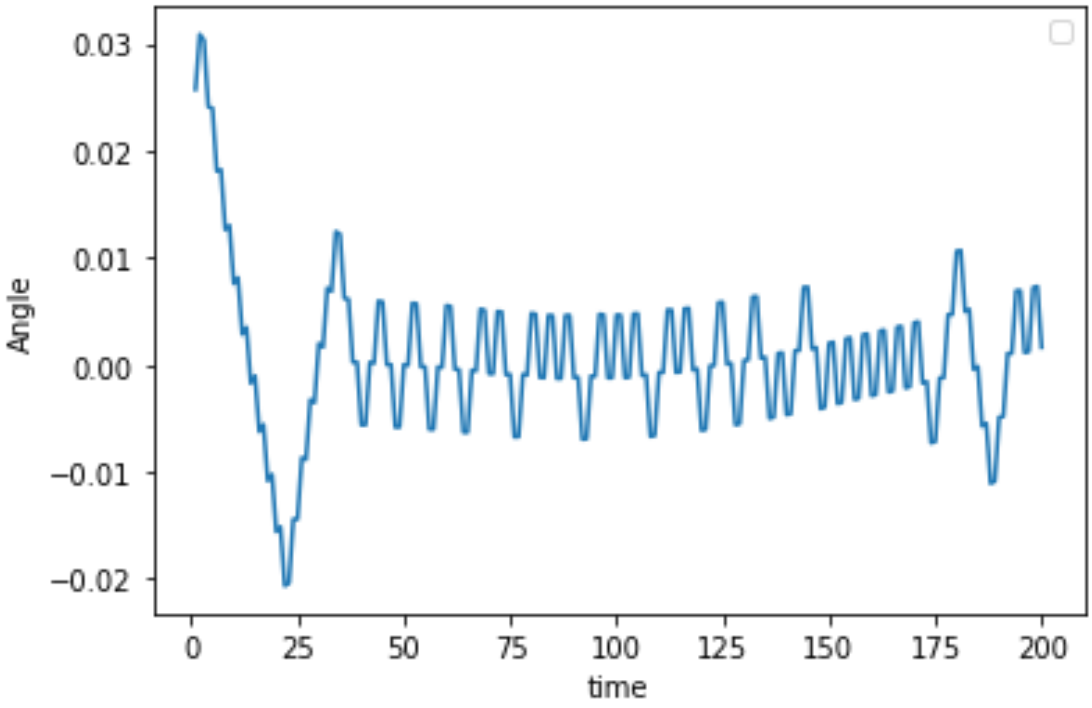
Finally, the game environment is executed for 50 times. The first move is a random move and based on that our model will play the game.

Results

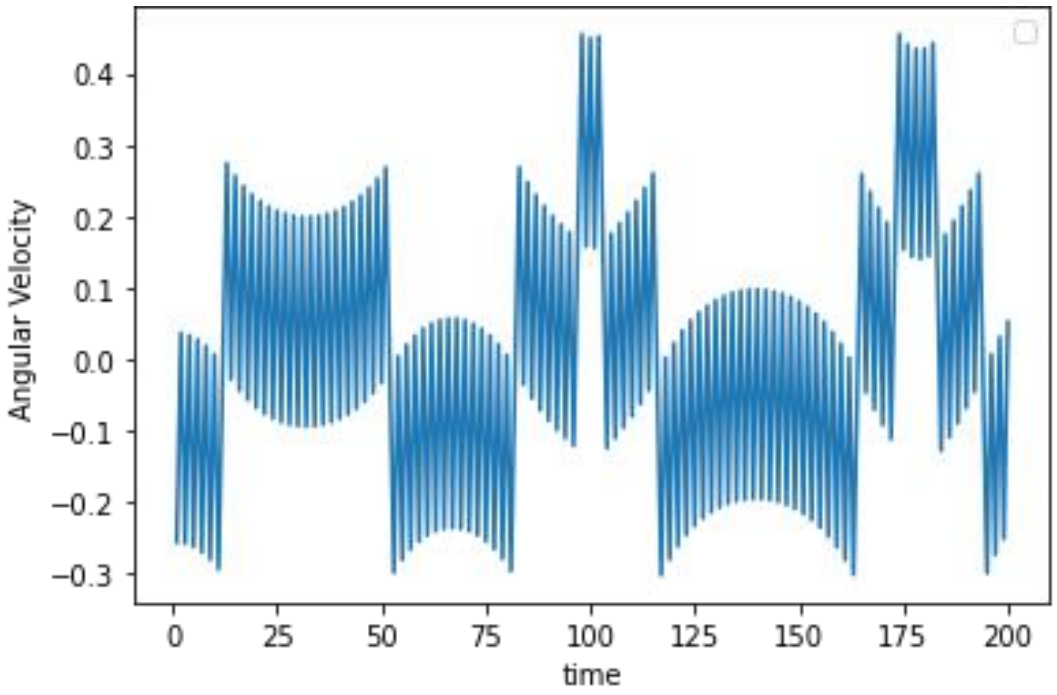
Plot of best Fitness versus Tournaments for several runs of algorithm.



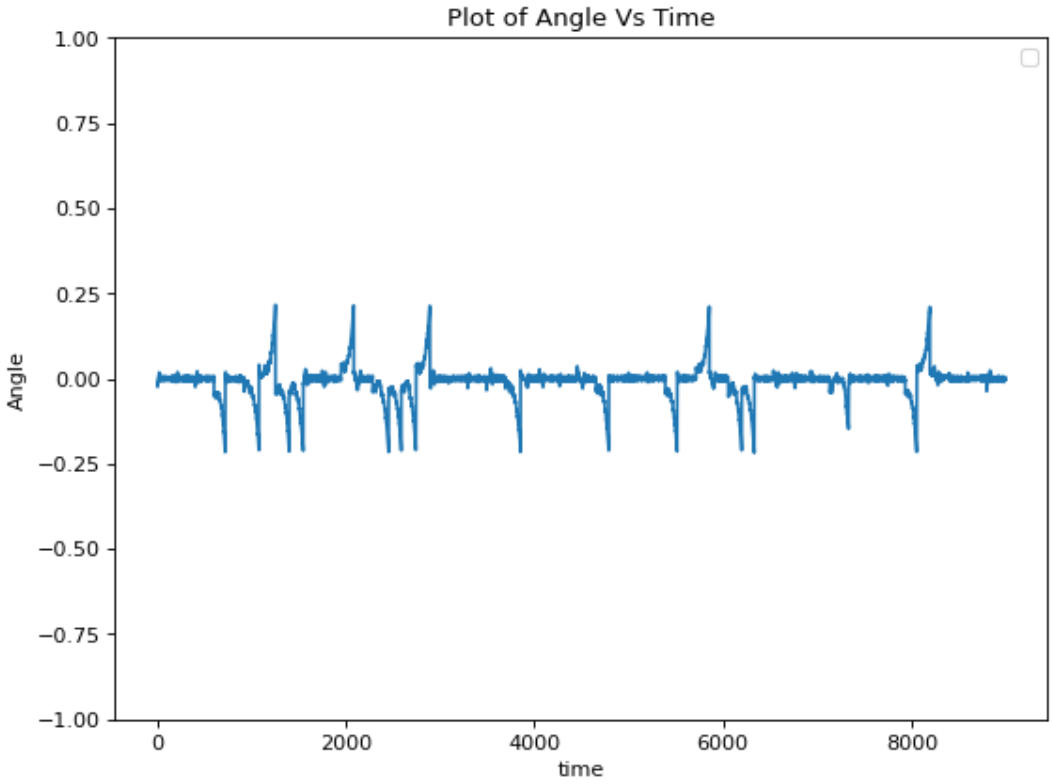
Plot of Angle vs Time for One Episode



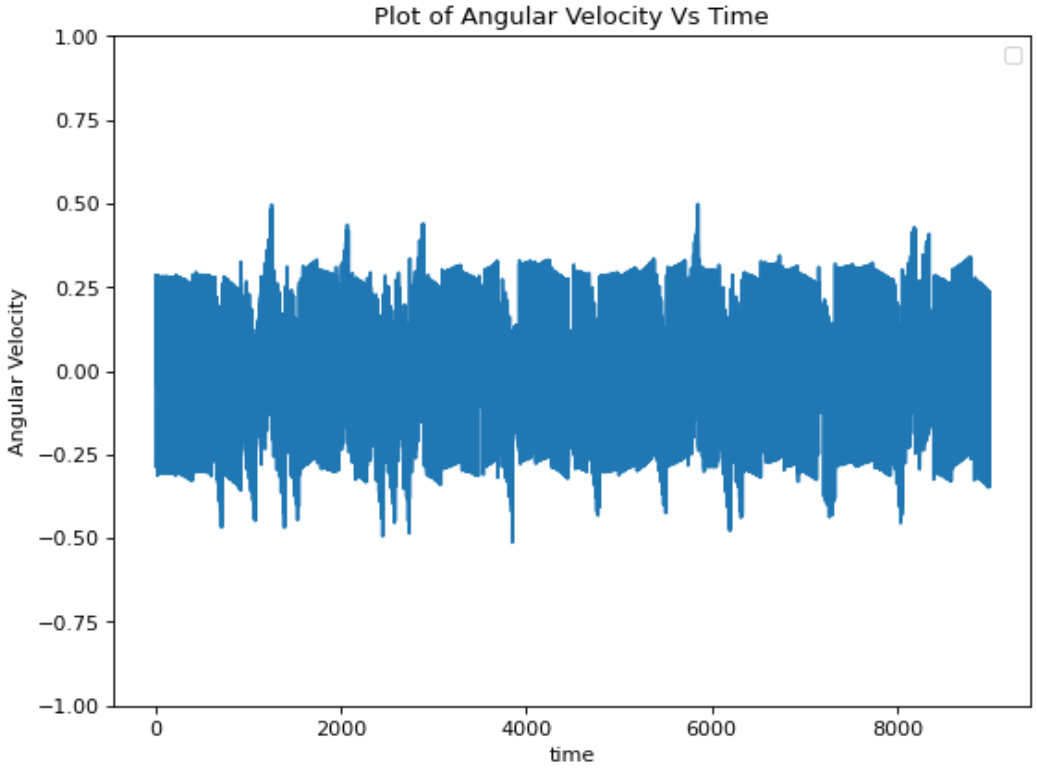
Plot of Angular Velocity vs Time for One Episode



Plot of Angle vs Time for All Episodes



Plot of Angular Velocity vs Time for All Episodes



Example of Best Controller for 50 Tournaments/Episodes

At Tournament 0, Reward = 10

At Tournament 25, Reward = 193

At Tournament 49, Reward = 200

```
Tournament: 0, total_reward: 10.00
Tournament: 1, total_reward: 10.00
Tournament: 2, total_reward: 9.00
Tournament: 3, total_reward: 10.00
Tournament: 4, total_reward: 10.00
Tournament: 5, total_reward: 10.00
Tournament: 6, total_reward: 12.00
Tournament: 7, total_reward: 10.00
Tournament: 8, total_reward: 10.00
Tournament: 9, total_reward: 200.00
Tournament: 10, total_reward: 200.00
Tournament: 11, total_reward: 200.00
Tournament: 12, total_reward: 200.00
Tournament: 13, total_reward: 200.00
Tournament: 14, total_reward: 200.00
Tournament: 15, total_reward: 200.00
Tournament: 16, total_reward: 200.00
Tournament: 17, total_reward: 200.00
Tournament: 18, total_reward: 200.00
Tournament: 19, total_reward: 200.00
Tournament: 20, total_reward: 200.00
Tournament: 21, total_reward: 200.00
Tournament: 22, total_reward: 200.00
Tournament: 23, total_reward: 200.00
Tournament: 24, total_reward: 200.00
Tournament: 25, total_reward: 193.00
```

```
Tournament: 25, total_reward: 193.00
Tournament: 26, total_reward: 185.00
Tournament: 27, total_reward: 200.00
Tournament: 28, total_reward: 200.00
Tournament: 29, total_reward: 200.00
Tournament: 30, total_reward: 200.00
Tournament: 31, total_reward: 200.00
Tournament: 32, total_reward: 200.00
Tournament: 33, total_reward: 200.00
Tournament: 34, total_reward: 200.00
Tournament: 35, total_reward: 200.00
Tournament: 36, total_reward: 200.00
Tournament: 37, total_reward: 200.00
Tournament: 38, total_reward: 200.00
Tournament: 39, total_reward: 200.00
Tournament: 40, total_reward: 200.00
Tournament: 41, total_reward: 200.00
Tournament: 42, total_reward: 200.00
Tournament: 43, total_reward: 200.00
Tournament: 44, total_reward: 200.00
Tournament: 45, total_reward: 200.00
Tournament: 46, total_reward: 200.00
Tournament: 47, total_reward: 200.00
Tournament: 48, total_reward: 200.00
Tournament: 49, total_reward: 200.00
```

Discussion

For this experiment the mutation rate was set to 0.4 and crossover probability set to 0.8. A single episode terminates if any of the following condition fulfils:

1. Pole angle is more than 12 degrees.
2. The center of the cart reaches the edge of the display.
3. Timer for episode end which is of 200 timesteps

The results show that in the starting episodes, the reward value is less and it gradually increases and reaches to its max value which is 200. This is because when the game starts the controller takes a random action as there are no previous observations. As the observations starts arriving, the data is fed to the genetic algorithm model which tries to maximize the score and due to this reason, the reward gets better and better after each episode.

The plots of angle and angular velocity also depict the same thing. The spikes show that the accuracy is falling while maximum accuracy is achieved when the graph is oscillating at the x-axis.

Conclusion

The Genetic Algorithm is well enough to calculate the best values to balance the cartpole. This problem can also be solved using algorithms such as population of hillclimbers or Deep Q-Learning or any other reinforcement algorithm. It would be interesting to see the effect of different algorithms and compare them to understand which one is the best for this specific problem.

References

- Tutorialspoint.com. n.d. *Genetic Algorithms - Introduction - Tutorialspoint*. [online] Available at: <https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_introduction.htm> [Accessed 20 May 2021].
- Medium. 2018. *Understanding OpenAI Gym*. [online] Available at: <https://medium.com/@ashish_fagna/understanding-openai-gym-25c79c06eccb> [Accessed 20 May 2021].
- Tiwari, S., 2019. *Genetic Algorithm: Part 1 -Intuition*. [online] Medium. Available at: <https://medium.com/koderunners/genetic-algorithm-part-1-intuition-fde1b75bd3f9> [Accessed 20 May 2021].
- Tiwari, S., 2019. *Genetic Algorithm: Part 4 -CartPole-v0*. [online] Medium. Available at: <https://medium.com/koderunners/genetic-algorithm-part-4-cartpole-v0-cfb500190cc4> [Accessed 20 May 2021].
- GitHub. 2020. *openai/gym*. [online] Available at: <https://github.com/openai/gym/wiki/CartPole-v0> [Accessed 20 May 2021].