**Introduction**

Reinforcement learning is a machine learning training method where desired behavior is rewarded (positive reinforcement) while undesirable behavior is penalized (negative reinforcement). A reinforcement learning agent can perceive and interpret its environment, act, and receive a reward or penalty based on the action. In this way, a reinforcement learning agent can learn through trial and error.

Open Ai Gym is an open-source Python library which facilitates the development and comparison of different reinforcement algorithms by providing a standard API to communicate between learning algorithms and environments.

In this paper, we will be focusing on the topic of reinforcement learning using Open Ai Gym. Two environments have been chosen to be the focus of our research which are Cart Pole v0 and Acrobot v1 from the Classic Controls environment. Cart Pole was chosen to showcase positive reinforcement learning while Acrobot was chosen to showcase negative reinforcement learning.

The first task detailed in this paper is to innovate and improve existing reinforcement learning algorithms to improve the performance of the algorithm. The main technique used is to combine different improvements of reinforcement learning algorithms and to compare the results. This allows us to ascertain the best combination which produces the best results.

The second task is to carry out simple research on the reinforcement learning agents. The research topic decided on was to study the effects of different reward functions on the performance of the reinforcement learning algorithms. The base reward function used for Cart Pole and Acrobot are bland and uninformative. Several different reward functions have been proposed and will be tested to determine the best reward function for both problems.

The reinforcement algorithm used in this paper is Q-Learning. The base of the code used in this paper is adapted from medium.com and was created by Matthew Chan. The code has been modified to run the chosen environment at least 30 times and to collect results such as the mean number of episodes for completion and the standard deviation in completion episodes. The code also allows easier change in variables and reward functions to ease the process of result collection.

**Background**

Before going further into the paper, research was done to discover any key papers that may relate to the topic discussed in the paper. Research papers focusing on innovations to reinforcement learning algorithms and reward functions were prioritized.

For innovations in reinforcement learning algorithms, 6 research papers of interest were found. As shown by Jacopo Fior and Luca Cagliero, the granularity of the data influences the results of machine learning-based stock trading. H.R. Tizhoosh has also tried to introduce opposition-based learning as a new scheme for machine intelligence. Research into optimistic exploration value functions has also been done by Michal Gregor and Juraj Spalek. Tom Zahavy, Matan Haroush, et al. have done research into action elimination with deep reinforcement learning. Reseacrh into discouting deep reinforcement learning has also been done by Vincent François-Lavet, Raphael Fonteneau and Damien Ernst. Finally, Hado van Hasselt introduced the Double Q-Learning algorithm which lacks the overestimation bias of the traditional Q-Learning algorithm.

For research into reinforcement learning reward functions, 3 research papers were foud to be relevant. Stefan Šćepanović has tested and compared the training performance of a Double DQN with different reward functions. Reward functions based on the DDQN algorithm have also been designed by Qianhao Xiao, Xin Zhang, et al. Finally, Jiexin Xie, Yue Li, et al. have optimized the reward functions of deep reinforcement learning for robotic trajectory planning.

**Methods**

For both tasks, some parameters have been retained throughout the entire experiment. These parameters are:

1. Seed – 20313854
2. Max Discount Factor – 0.999
3. Min Learning Rate – 0.1
4. Min Explore Rate – 0.01

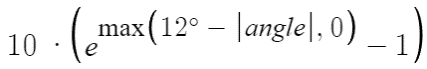
For the first task, 6 improvements were chosen based on the 6 research papers above on innovations in reinforcement learning algorithms. To facilitate easier testing and comparison between the results, improvements that span a range of values have been fixed such as granularity, discount factor and random initialization of initial Q-table values.

The 6 improvements to the reinforcement learning algorithms are:

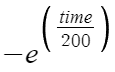
1. N-tuple Q-tables
   * Number of Q-tables can range from 1 to 5
   * Action is selected based on the sum of all Q-tables
   * First Q-table is randomly selected to be updated with reward
   * Second Q-table is randomly selected from remaining tables to get best Q-value
2. Granularity of Q-tables
   * Granularity of Cart Pole is either (1, 1, 6, 3) or (1, 1, 6, 7)
   * Granularity of Acrobot is fixed at (1, 1, 1, 1, 10, 10)
3. Opposition Learning
   * Opposite action of a state is also acted upon, and the opposite reward added to the Q-table
4. Random Initialization of Q-tables
   * Q-tables can start with all 0s or with values drawn from a standard normal distribution
5. Discount Factor Discounting
   * Discount factor can be constant at 0.999 or start from 0.99 and increase by 0.001 each episode until 0.999 is reached
6. Action Limiting
   * Number of actions for Cart Pole is fixed at 2
   * Number of actions for Acrobot is 2 if do nothing action is removed or 3

For the second task of the paper, 5 additional reward functions were chosen for Cart Pole while 3 additional reward functions were chosen for Acrobot.

The 5 reward functions for Cart Pole are:

1. Termination Penalty
   * Actions that lead to termination are given a penalty of -1
2. Time-Based Reward
   * 
   * Reward increases exponentially with time
3. Uniform Angle Reward
   * 
   * Reward varies linearly
4. Exponential Angle Reward
   * 
   * Reward increases more rapidly as angle decreases
5. Logarithmic Angle Reward
   * 
   * Reward decreases more rapidly as angle increases

The 3 reward functions for Acrobot are:

1. Time-Based Penalty
   * 
   * Reward decreases exponentially with time
2. Velocity-Based Penalty
   * Reward decreases when velocity decreases
   * Reward decreases when both velocities are in opposite directions
3. Height-Based Penalty
   * 
   * Reward decreases as height decreases

A reward of 0 is still given if Acrobot reaches threshold and episode terminates.

**Results**