Navigation Report

The aim of this project was to implement and train an agent to successfully navigate and collect yellow bananas in a large virtual square world. This is an episodic problem with the goal of maximising its cumulative reward.

The Learning Algorithm

To solve this problem more efficiently, deep Q-Learning network was implemented where the algorithm will improve itself through a cycle of exploration and exploitation activities and learning from experience.

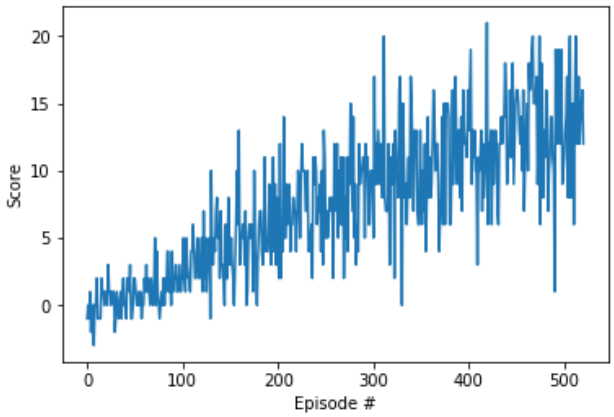
Hyper Parameters

Epsilon-greedy policy is a decaying probability percentage which gives the algorithm a chance of picking a random action. The gradual decrease of this during training allows the system to not be too greedy early on during exploration. As the network as explored more of the environment this epsilon-greedy policy gets smaller. A decay of 0.995 was found to be the best value to achieve a higher result for this problem.

The neural network model is implemented in “model.py” script. It is a 3 layer fully connected network with an input size of 64 using an Adam as the optimiser.

Results

The agent took 521 episodes to average a score of 13. This took less than 6 minutes on GPU.



Future Enhancements

In a reinforcement learning environment the speed to train a system is critical. Or rather the efficiency to search through the search space. This can be achieved by modifying additional hyper parameters such as hidden layers, batch sizes and parameters.

Alternative methods such as prioritised experienced replay, double Q-Learning and Duelling Deep Q-learning improve on the flaws of deep Q-learning. Prioritized experience replay assist in learning more effectively from transitions where more important transitions should

be sampled with higher probability. Double Q-learning has been proven to address the over estimation of the Q-values during the early stages in vanilla deep Q-learning. lastly, the duelling network allows the network to assess the value of each state without having to learn the effect of each action. Deep mind the original developers of DQN combined all these tweaks into the Rainbow network. It was shown to have significant improvement from the vanilla implementation. Thus, by implementing a new improved deep Q-learning algorithm, such as rainbow, it has potential to achieve higher results.