Project: Navigation

Environment

For this project I used a Unity ML-Agents environment of Banana world. The environment is a 3-d square space enclosed by walls, with blue and yellow bananas on the field. A reward of +1 is provided for collecting a yellow banana, and a reward of -1 is provided for collecting a blue banana. Thus, the goal of the agent is to collect as many yellow bananas as possible while avoiding blue bananas.

The state space has 37 dimensions and contains the agent's velocity, along with ray-based perception of objects around the agent's forward direction. Given this information, the agent has to learn how to best select actions. Four discrete actions are available, corresponding to:

- **0** - move forward.
- **1** - move backward.
- **2** - turn left.
- **3** - turn right.

The task is episodic, and in order to solve the environment, the agent must get an average score of +13 over 100 consecutive episodes.

Implementation

I implemented a Duelling Double Deep-Q Network which is an improved version of the Deep-Q Networks. The Deep-Q network uses neural networks as a function approximate with 2 techniques that improve training.

1. Fixed Targets: The target network is not updated for a number of steps to avoid oscillation in the loss function.
2. Experience Replay: Training is done on random samples from memory instead of sequential learning, which helps avoid learning strong correlations between consecutive experiences.

The model architecture has 2 fully connected layers with 64 Neurons and RELU activations, after which we branch off the network to separately calculate value (best value for corresponding state) and advantage function (best action advantage). After which we combine both to calculate the final Q-value. In the loss function, the Q in the target values are calculated by both the networks.

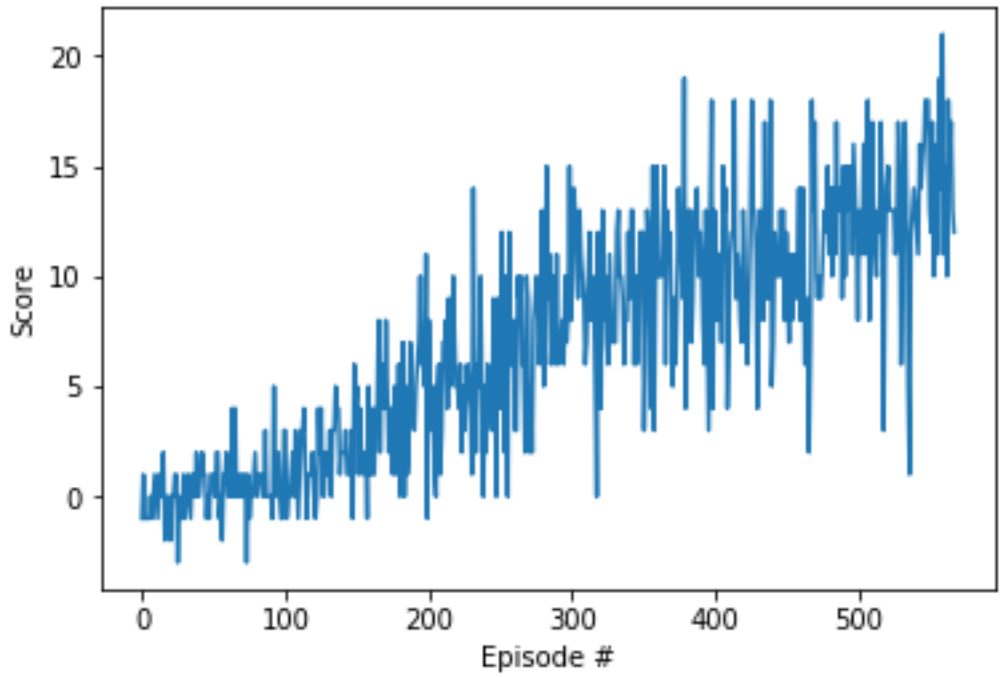
Hyper parameters

Replay Buffer Size	1e5
Mini-Batch Size	0.99
Gamma	1e-3
TAU	1e-3
Learning Rate	5e-3
Update Every	4

Number of Episodes	500
Max time steps per episode	2000
Epsilon Start	1
Epsilon Minimum	0.1
Epsilon Decay	0.995

Results

The environment gets solved in 467 episodes, achieving an average score of 13.03



Improvements

1. Carrying out an exhaustive grid search of different hyper parameters.
2. Implementing other improvements on DQN such as Prioritized Experience Replay, A3C, and Distributional DQN, ultimately leading to Rainbow DQN which combines all these improvements, and promises significant improvements over all these algorithms.