PROJECT REPORT

Banana world navigation project

**Summary of implementation –**

The implementation uses Fixed Q scores and Replay buffer techniques to achieve Deep Q learning. The code has been structured in 2 parts

1. Train-Test code:

This part includes the actual training/testing functions. This python notebook is the primary code to train the model and check model performance. Apart from primary train-test functions, the code has functions to save model weights, plot the scores

1. Classes declaration:

This project uses 3 primary classes

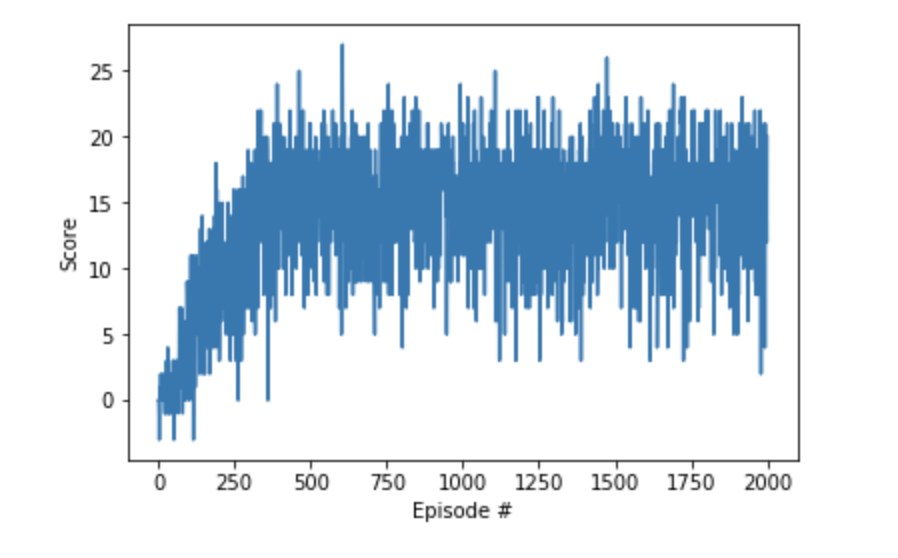
QAgent, QNetwork, ReplayBuffer

**QAgent** contains the agent’s functions related to learning, performing steps, etc. This class interfaces directly with the other 2 classes.

**QNetwork** contains the declaration of Neural network layers

**ReplayBuffer** is the class that stores all action-state-reward-nextState tuples to be sampled randomly during training

**Scores achieved by the model**

****

The above plot shows the performance of the model. The model achieves >13 average score within first 500 learning steps. The average score at end of 2000 epochs is ~17

**Next steps**

The next steps for this project involve trying different network architectures to get better (higher) average scores for the model.

Next steps also involve using different techniques mentioned in Deep Q networks paper (DeepMind) to experiment with agent performance and learning.