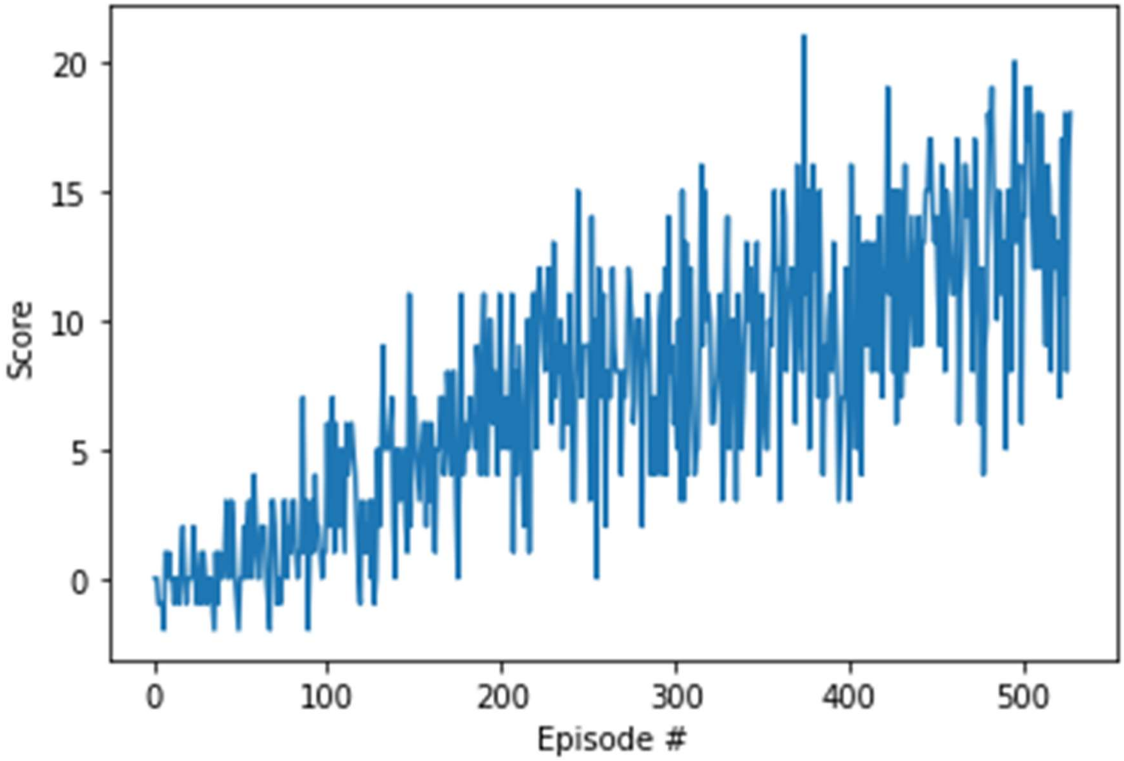
Project 1: Navigation with Unity Banana Environment

Algorithm description

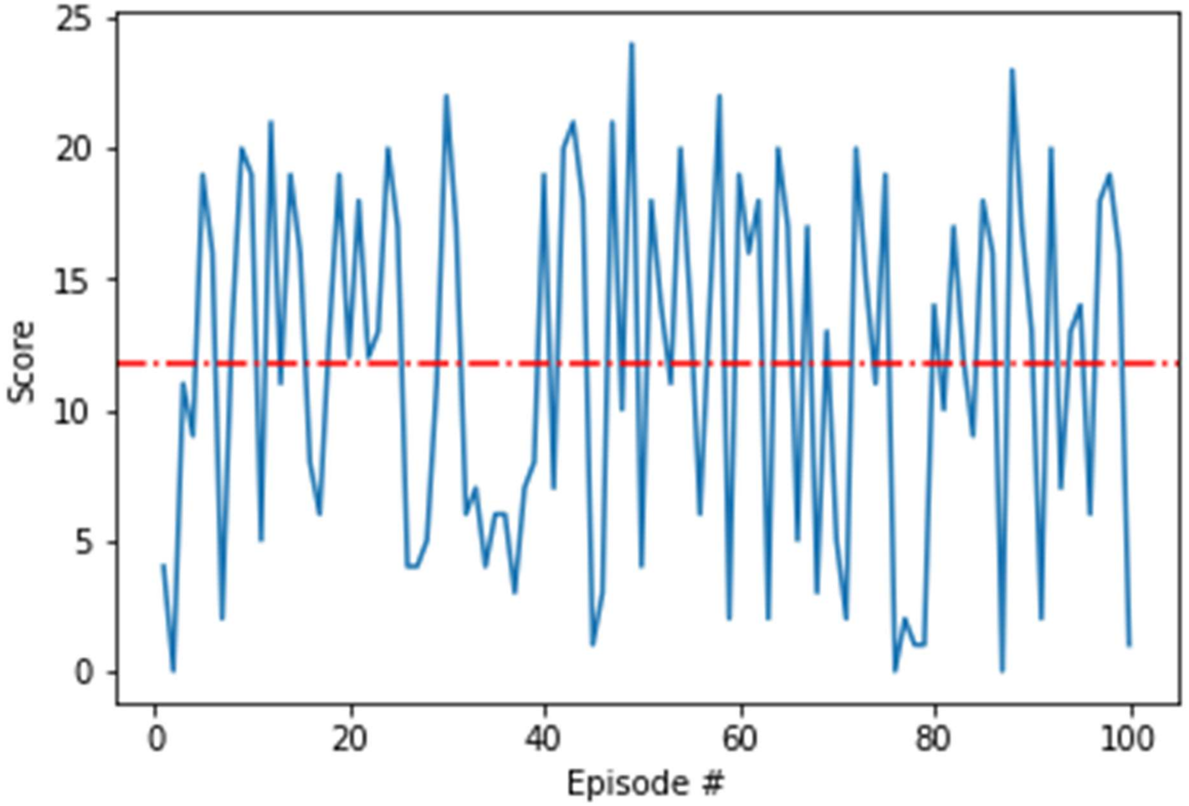
The Deep Q Network (DQN) algorithm was used to solve this environment. DQN is a value-based algorithm that optimizes a neural network that approximates the state-action value function [1]. Most of this code was provided in the Udacity nanodegree repository, and changes were made to allow the agent to work with the provided environment.

Training and Testing Results

The environment was solved by the DQN agent in 527 episodes, obtaining an average score over the previous 100 episodes of +13. Learning shows steady improvement, indicating overall stability, which is ideal in RL applications.



The trained agent was run for an additional 100 episodes to prove its capability to generalize to new states. These results are included here.



Opportunities for Future Work

One thing noted during training was that, in some cases where the nearest yellow bananas were further away, and relatively equidistant from the agent, it would oscillate back and forth, never committing to a direction of travel. This caused episodes to end when they otherwise might not have, bringing down the score for the episode. This could be resolved by a term in the reward function that penalizes inaction.

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller, “Playing Atari with Deep Reinforcement Learning”, https://arxiv.org/abs/1312.5602