**Objective**:

Train an agent to collect Yellow Bananas while avoiding Blue Bananas

**Environment:**

The environment has a state space that 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.

We can take four actions, namely:

1. 0: move forward
2. 1: move backward
3. 2: turn left
4. 3: turn right

**Implementation:**

1. We are using DQN for the Reinforcement Learning task
2. Idea being used is from the paper "Human-level control through deep reinforcement learning" (https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf)
3. We are using Replay buffer to train the network
4. We also tried prioritized experience replay code(https://github.com/rlcode/per.git) but did not get the expected result. Also tried using heap for prioritized memory.

**Model:**

1. Input Size: 37
2. Number of hidden layers: 2
3. Hidden layers sizes: [37*4, 37*4]
4. Output Layer size = 4
5. Activation function: Relu
6. Optimizer: Adam

**Hyperparameters:**

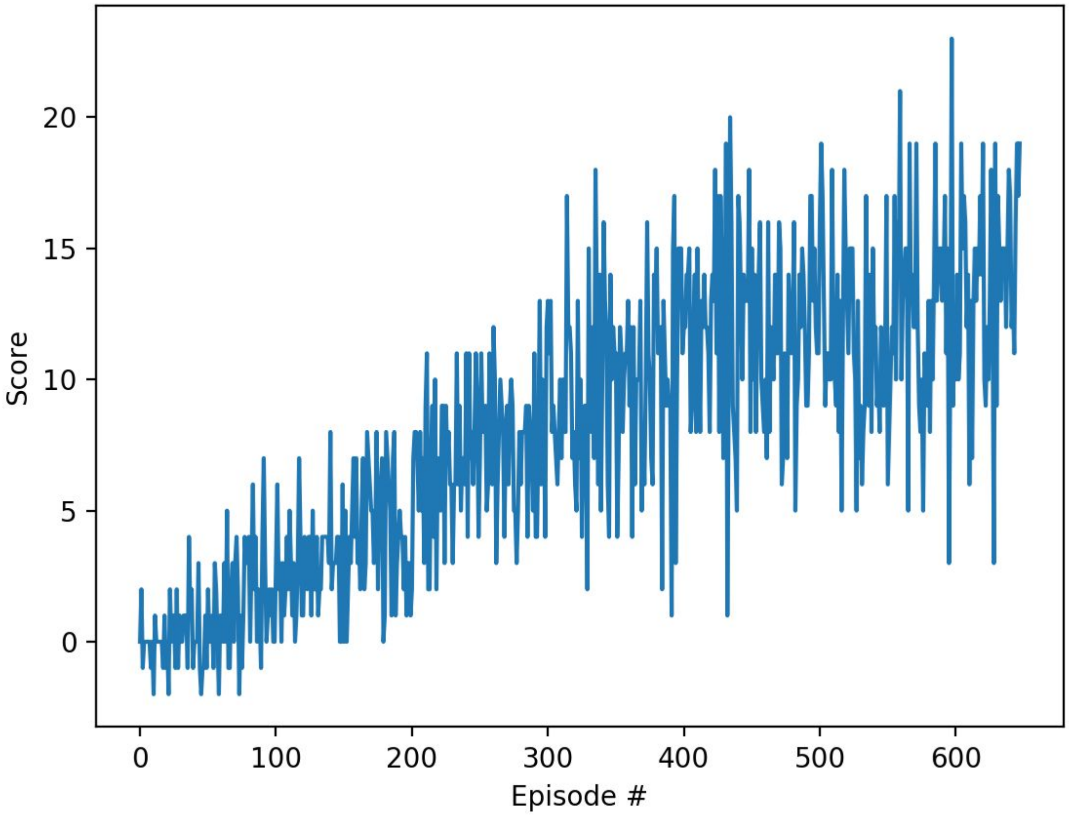
1. Buffer Size:
    a. Description: size of replay memory

    b. Value: int(1e5)

2. Batch size:
    a. Description: number of samples being used in one iteration
    b. Value: 64
3. Gamma:
    a. Description: discount factor
    b. Value: 0.99
4. Tau:
    a. Description: factor for soft update of target model
    b. Value: 1e-3
5. LR:
    a. Description: learning rate
    b. Value: 5e-4
6. Update_every:
    a. Description: After how many samples we need to learn
    b. Value: 4

**Output**:

1. Agent took around 560 episodes to reach an average reward of 13 over 100 episodes.
2. Model is saved in the checkpoint.pth file.



**Ideas for future work:**

1. Need to look at why prioritized experience replay is not work.
2. Work on double DQN and Dueling DQN.