Report

For the navigation project, we used a DQN model to determine the best action at each step.

This model had a basic architecture of 2 fully connected neural networks with 64 nodes in each layer. The activation function used after each hidden layer is a RELU. The used architecture is the vanilla DQN model.

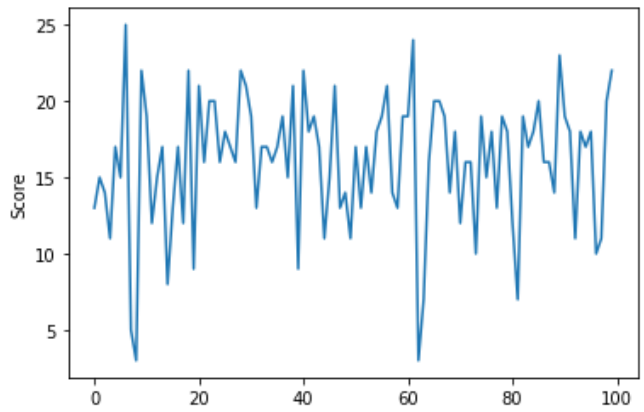
We initialized our variables to:

```
BUFFER_SIZE = 10000     # replay buffer size
BATCH_SIZE = 64         # minibatch size
GAMMA = 0.99            # discount factor
TAU = 1e-3              # for soft update of target parameters
LR = 5e-4               # learning rate
UPDATE_EVERY = 4
eps_start = 1.0         # starting value of epsilon, for epsilon-
greedy action selection
eps_end=0.01            # minimum value of epsilon
eps_decay=0.995         # multiplicative factor (per episode) for decreasing e
psilon
```

The maximum average score to reach before exiting the learning loop and save the checkpoint was initialized to 16.

Training the model with those variables had this output:

```
Episode 100     Average Score: 1.13
Episode 200     Average Score: 4.96
Episode 300     Average Score: 6.94
Episode 400     Average Score: 9.99
Episode 500     Average Score: 13.42
Episode 600     Average Score: 14.44
Episode 700     Average Score: 15.38
Episode 800     Average Score: 14.66
Episode 900     Average Score: 15.53
Episode 1000    Average Score: 15.28
Episode 1100    Average Score: 15.70
Episode 1200    Average Score: 15.73
Episode 1254    Average Score: 16.02
Environment solved in 1154 episodes!    Average Score: 16.02
```



Next steps

After trying the checkpoint, we observed that the agent had a good understanding of locating the nearest yellow banana and how to reach it without passing any blue banana.

Yet, the agent seemed to be confused when no near yellow banana was in sight. He just stayed in place with no deciding actions. This can be explained by the scarcity of this state (no yellow banana in sight) and even when it happened the agent did not try a lot of random action so he learned a bad pattern. This can be solved by training the agent with more episodes and having epsilon converge to the minimum value more slowly.