Article I. Solution Design

Three algorithms were experimented to train an AI Agent to maximize the reward by collecting more of yellow bananas. Three algorithms used are Deep Q Networks, Double Deep Q Networks and Duel Deep Q Networks.

Article II. Algorithms

Section 2.01 Deep Q Networks

(a) Model Architecture

```
self.linear_layers = Sequential(
    Linear(state_size, 64),
    ReLU(inplace=True),
    Linear(64, 64),
    ReLU(inplace=True),
    Linear(64, 128),
    ReLU(inplace=True),
    Linear(128,action_size)
)
```

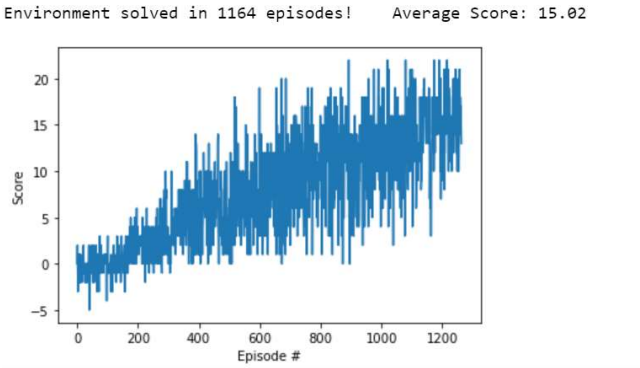
(b) Hyper Parameters

- Replay Buffer Size = 1e6
- Minibatch Size = 64
- Discount Factor (Gamma) = 0.99
- For soft update of target parameters TAU = 1e-4
- Learning Rate= 5e-5
- How often update the network (UPDATE_EVERY) = 2

(c) Result

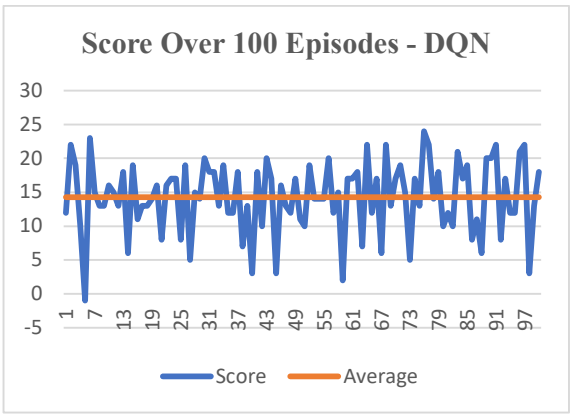
(i) Training

The environment was first solved in 1164 episodes to achieve average score of 15.02



(ii) Test

Average Reward over 100 episodes = 14.26



The values can be found in last cell of Navigation-DQN.ipynb. Computation Details of Graph can be found in graph.xlsx

Section 2.02 Duel Deep Q Networks

(a) Model Architecture

```
self.feature = nn.Sequential(
    Linear(state_size, 64),
    ReLU(inplace=True),
    Linear(64, 64),
    ReLU(inplace=True),
)

self.advantage = nn.Sequential( #Calculate A(s,a)
    Linear(64, 128),
    ReLU(inplace=True),
    Linear(128,action_size)
)

self.value = nn.Sequential( #Calculate V(s)
    Linear(64, 64),
    ReLU(inplace=True),
    Linear(64,1)
)
```

(b) Hyper Parameters

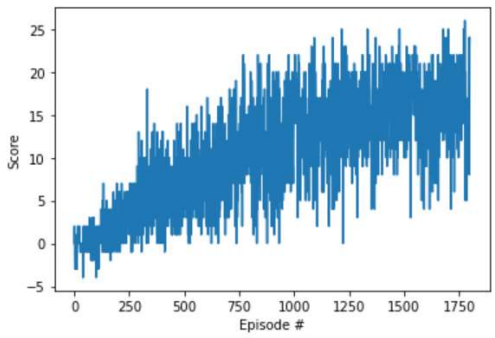
- Replay Buffer Size = 1e6
- Minibatch Size = 64
- Discount Factor (Gamma) = 0.99
- For soft update of target parameters TAU = 1e-4
- Learning Rate= 5e-5
- How often update the network (UPDATE_EVERY) = 2

(c) Result

(i) Training

Overall Average score in 1800 episodes was 16.6

```
Environment solved in 1800 episodes!    Average Score: 16.60
```

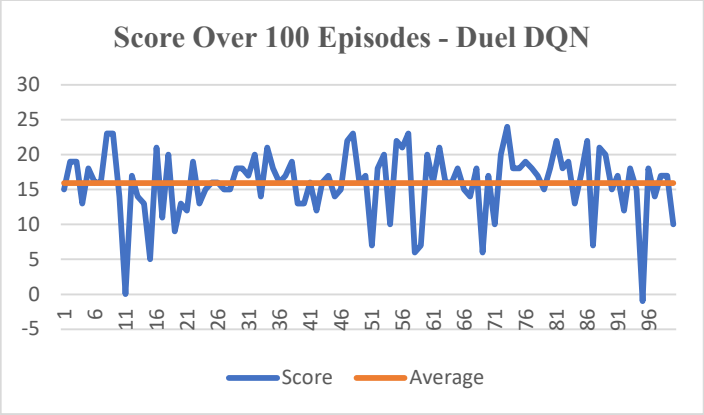


The environment was first solved in 1329 episodes to achieve average score of 15.04.

```
Episode 1200    Average Score: 13.18
Episode 1300    Average Score: 14.57
Episode 1329    Average Score: 15.04
Environment solved in 1229 episodes!    Average Score: 15.04
Episode 1330    Average Score: 15.00
Environment solved in 1230 episodes!    Average Score: 15.00
Episode 1334    Average Score: 15.04
Environment solved in 1234 episodes!    Average Score: 15.04
Episode 1335    Average Score: 15.04
```

(ii) Test

Average Reward over 100 episodes = 15.91



The values can be found in last cell of Navigation-Duel DQN.ipynb.

Computation Details of Graph can be found in graph.xlsx

Section 2.03 Double Deep Q Networks

(a) Model Architecture

```
self.linear_layers = Sequential(
    Linear(state_size, 64),
    ReLU(inplace=True),
    Linear(64, 64),
    ReLU(inplace=True),
    Linear(64, 128),
    ReLU(inplace=True),
    Linear(128,action_size)
)
```

(b) Hyper Parameters

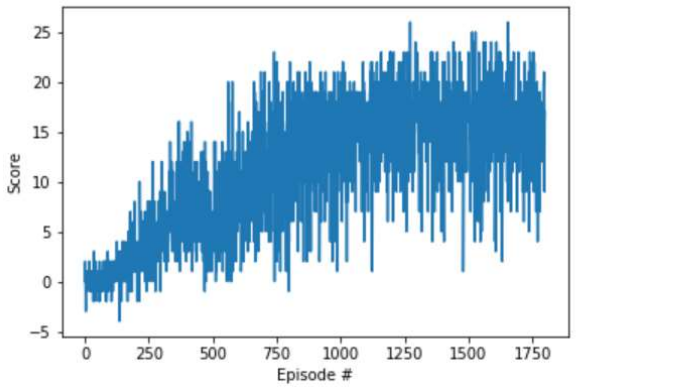
- Replay Buffer Size = 1e6
- Minibatch Size = 64
- Discount Factor (Gamma) = 0.99
- For soft update of target parameters TAU = 1e-4
- Learning Rate= 5e-5
- How often update the network (UPDATE_EVERY) = 2

(c) Result

(i) Training

Overall Average score in 1800 episodes was 15.48.

```
Environment solved in 1800 episodes!    Average Score: 15.48
```

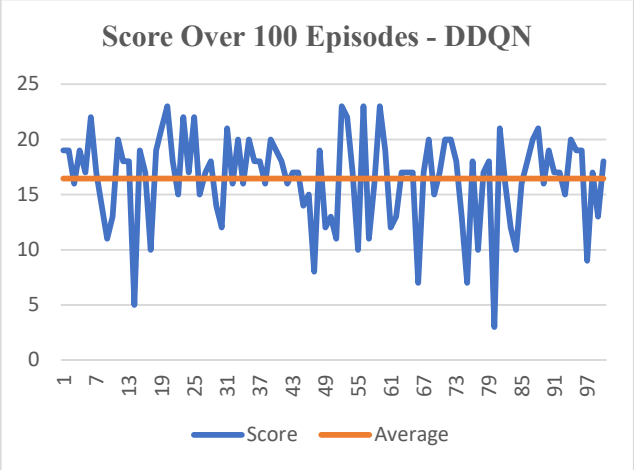


The environment was first solved in 1200 episodes to achieve average score of 15.58.

```
Episode 900     Average Score: 13.57
Episode 1000    Average Score: 13.56
Episode 1100    Average Score: 14.92
Episode 1200    Average Score: 15.58
Episode 1300    Average Score: 15.83
Episode 1400    Average Score: 15.87
```

(i) Test

Average Reward over 100 episodes = 16.44



The values can be found in last cell of Navigation-DDQN-Test.ipynb.

Computation Details of Graph can be found in graph.xlsx

Article III. Conclusion

The best average reward over 100 episodes when training mode is off , was achieved by DDQN Algorithm, it was trained to solved the environment i.e to get average >15 (threshold set) in 1200 episodes .