Report

Introduction

In this project the agent is navigating in a large square world with a purpose, collecting bananas! The agent gets a reward of +1 each time it gets a yellow banana and -1 if it gets a blue one. So the goal is for the agent to collect as many yellow bananas and to avoid as many blue bananas as possible. The state space has 37 dimensions which contains agent's velocity, as well as perception of objects around its forward direction. The agent is able to take four discrete actions:

- 0 - move forward
- 1 - move backward
- 2 - turn left
- 3 - turn right

The environment is considered solved if the agent can get an average score of +13 over 100 consecutive episodes.

Learning Algorithm

The agent uses DQN for it's perception (analysis) and decisioning. The DQN is fully-connected neural network with ReLUs as its activation functions. The last layer outputs the Q-values for the four different actions as defined above in the introduction. The training of the agent makes use of Experience Replay with a buffer size of 10k. Also, Fix Q-Targets are used where the target network are updated every 4 times the local network is trained. We used soft-update to update the target network with $\tau = 10^{-3}$. For more details of the network architecture and hypterparameters, please see below.

Architecture

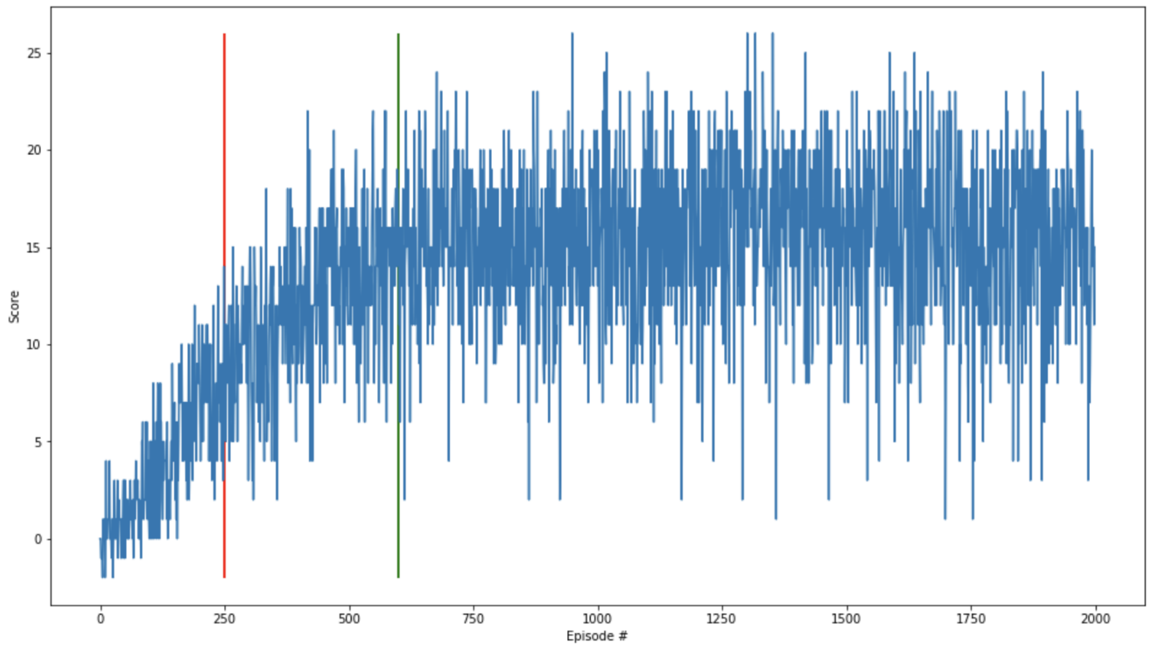
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                   [-1, 64]           2,432
              ReLU-2                   [-1, 64]               0
            Linear-3                   [-1, 64]           4,160
              ReLU-4                   [-1, 64]               0
            Linear-5                    [-1, 4]             260
================================================================
Total params: 6,852
Trainable params: 6,852
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.00
Params size (MB): 0.03
Estimated Total Size (MB): 0.03
----------------------------------------------------------------
```

Hyperparameters

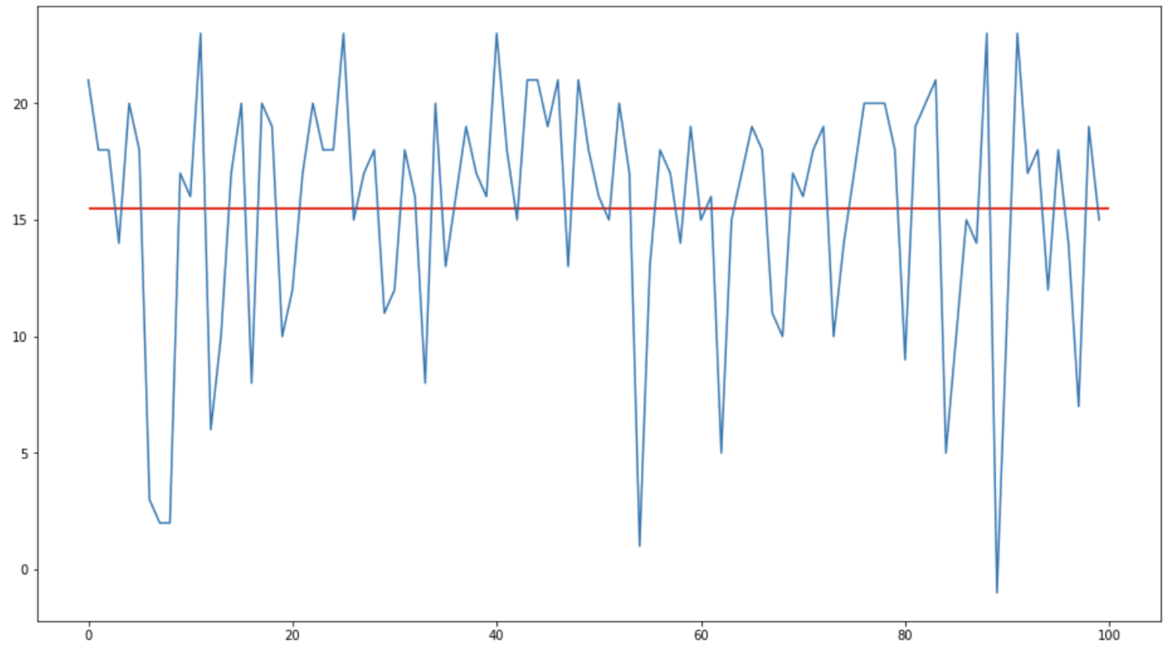
```
BUFFER_SIZE = int(1e5)  # replay buffer size
BATCH_SIZE = 64         # minibatch size
GAMMA = 0.99            # discount factor
TAU = 1e-3              # for soft update of target parameters
LR = 5e-4               # learning rate
UPDATE_EVERY = 4        # how often to update the network
```

Performance

The agent first achieved the target score episode 249 and converged at around episode 600.



The agent is able to receive an average reward (over 100 episodes) of +15.5



Future Improvement

It's quite surprising that the initial implementation works pretty much out of the box. There definitely is a lot of room for improvement for sure. Just to list a few things to try out:

- Learning from pixels
- Double DQN
- Prioritized Experience Replay
- Duel DQN
- Some data augmentation techniques
- A deeper neural network (ResNet, Inception, etc.)