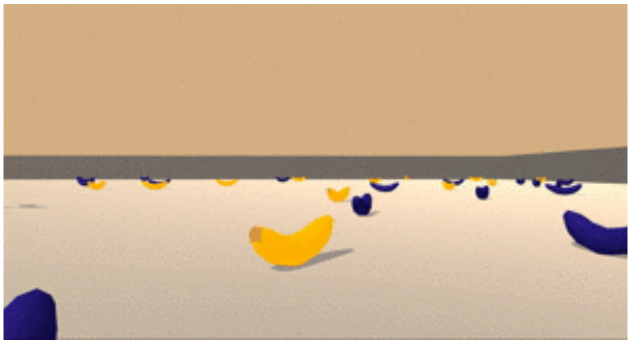
Report: Project1

Deep Reinforced Learning Nanodegree

Introduction

This project is to train an agent to navigate and collect yellow bananas and not the blue ones in a large, square world by using Deep Q Networks.



State and Action Space

A reward of +1 is provided for collecting a yellow banana, and a reward of -1 is provided for collecting a blue banana. Thus, the goal of your agent is to collect as many yellow bananas as possible while avoiding blue bananas.
The state space has 37 dimensions and contains the agent's velocity, along with ray-based perception of objects around agent's forward direction. Given this information, the agent has to learn how to best select actions. Four discrete actions are available, corresponding to:

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

Req: The task is episodic, and in order to solve the environment, the agent must get an average score of +13 over 100 consecutive episodes.

Learning algorithm

The agent uses **Deep Q Network** to learn the navigation environment. The agent uses two networks a local and target network. The local network learns at every step and also stores the experiences for replay. At periodic intervals the target network learns these updates from the local network with a learning rate tau. The agent is run until we reach n episode (fail) or achieves an average score of min +13 in 100 consecutive steps (pass). Each episode also has a max step timeout called max_t in case it does not reach done = True.

Hyper-parameters Used

Agent Specific

BUFFER_SIZE = 100000 # Number of steps of experience that is in memory

```
BATCH_SIZE = 64
GAMMA = 0.99            # Discount rate on future rewards
TAU = 0.001             # Softener - target network update
LR = 0.0005             # Learning rate
UPDATE_EVERY = 4        # Local network -> Target network
```

Learning Specific

```
n_episodes = 2000        # Max number of episodes
max_t = 1000             # Max allowed steps per episode
eps_start = 1.0          # Start with exploration
eps_end = 0.01           # End with exploitation
eps_decay = 0.995        # Transition from exploration to exploitation
```

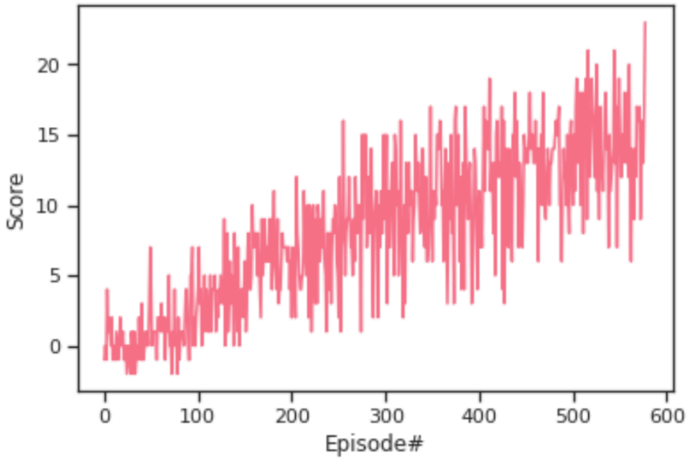
Might as well add it here

```
state_size = 37
action_size = 4
```

NN

Input = state_size -> 2 fully connected layers(64) with Relu activation -> Output = fully connected # action_size

Plot of Rewards



```
Episode 100   Average Score: 0.78
Episode 200   Average Score: 4.79
Episode 300   Average Score: 7.69
Episode 400   Average Score: 10.21
Episode 500   Average Score: 12.48
Episode 578   Average Score: 14.04
Environment solved in 478 episodes!         Average Score: 14.04
```

(I used 14 as the pass mean score)

GIFs on my setup, learning process, trained agent in action

Setup: https://media.giphy.com/media/l4A4Sd3hJpQVYqQxD1/giphy.gif
Learning: https://media.giphy.com/media/XeLZWu5XCDvdrIzy3f/giphy.gif

Trained Agent: https://media.giphy.com/media/RitEEBDSVfgdOJH7XJ/giphy.gif

Future work

1. Try the pixel based training method with convolutional networks
2. Play with Double DQN, Prioritized experience replay, Dueling DQN, multi-step bootstrap target, Distributional DQN, Noisy DQN
3. This might be a stretch - I do want to get to the Rainbow

Setting Up

1. Clone the DRL Nanodegree learning repo :https://github.com/udacity/deep-reinforcement-learning#dependencies
2. Download the environment from one of the links below.
Linux: https://s3-us-west-1.amazonaws.com/udacity-drlnd/P1/Banana/Banana_Linux.zip
Mac OSX: https://s3-us-west-1.amazonaws.com/udacity-drlnd/P1/Banana/Banana.app.zip
Windows (32-bit): https://s3-us-west-1.amazonaws.com/udacity-drlnd/P1/Banana/Banana_Windows_x86.zip
Windows (64-bit): https://s3-us-west-1.amazonaws.com/udacity-drlnd/P1/Banana/Banana_Windows_x86_64.zip
3. Place the file in the DRLND GitHub repository, in the `p1_navigation/` folder, and unzip (or decompress) the file.