

Udacity Deep Reinforcement Learning Nanodegree

Project 1: Navigation

1. Project Overview

In this project, a reinforcement learning agent is trained to navigate in a simulated environment called the **Banana Collector**, provided by Unity ML-Agents. The goal of the agent is to collect yellow bananas (reward +1) and avoid blue bananas (reward -1) by learning from interactions with the environment.

The agent is considered to have solved the environment when it achieves an average score of **+13** over **100 consecutive episodes**.

2. Environment Details

- **State Space:**
The state space has **37 continuous variables** representing environment information such as velocity and spatial orientation of objects around the agent.
 - **Action Space:**
There are **4 discrete actions** available:


```
0 - Move Forward  
1 - Move Backward  
2 - Turn Left  
3 - Turn Right
```
 - **Episode Termination:**
An episode ends when the agent collects a banana or when the time step limit is reached.
-

3. Learning Algorithm

- **Algorithm Used:** Deep Q-Network (DQN)
- **Main Features:**
 - Experience Replay
 - Fixed Q-Targets
 - Epsilon-Greedy Exploration
 - Soft Updates of Target Network Parameters

- Dropout / Dueling DQN
- **Hyperparameters:**

Hyperparameter	Value
Learning Rate (α)	5e-4
Discount Factor (γ)	0.99
Replay Buffer Size	100,000
Batch Size	64
Soft Update Parameter (τ)	1e-3
Update Frequency	Every 4 steps
Initial Epsilon	1.0
Minimum Epsilon	0.01
Epsilon Decay Rate	0.995

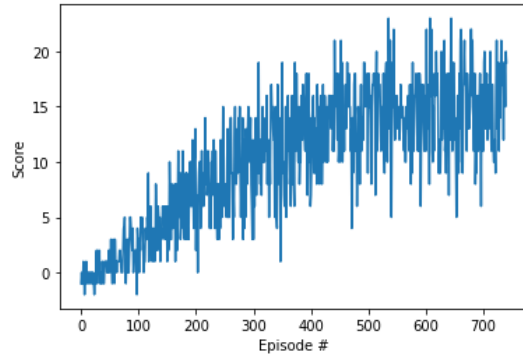
- **Neural Network Architecture:**
 - Input Layer: 37 units (state size)
 - One Fully Connected Hidden Layers: 128 and 128 units
 - Output Layer: 4 units (action size)
 - Activation: ReLU
 - Dropout : 0.25

4. Results

- **Final Performance:**
The agent achieved an average score of 15.03 over 100 consecutive episodes after **741** of training with **dqn** and an average score of 15.02 over 100 consecutive episodes after **785** of training with **dueling dqn**.
- **Score Progression:**
Below is a plot showing the both agent's score per episode and moving average over time:

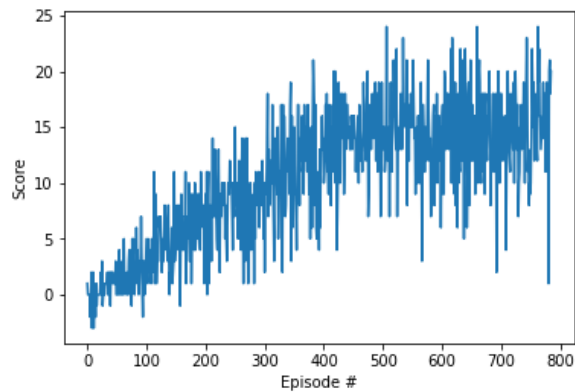
Agent trained with dqn:

```
Episode 100    Average Score: 0.81
Episode 200    Average Score: 4.89
Episode 300    Average Score: 8.51
Episode 400    Average Score: 11.44
Episode 500    Average Score: 13.52
Episode 600    Average Score: 14.40
Episode 700    Average Score: 14.59
Episode 741    Average Score: 15.03
Environment solved in 641 episodes!    Average Score: 15.03
```



Agent trained with dueling dqn:

```
Episode 100    Average Score: 1.13
Episode 200    Average Score: 5.04
Episode 300    Average Score: 8.09
Episode 400    Average Score: 11.49
Episode 500    Average Score: 14.02
Episode 600    Average Score: 14.39
Episode 700    Average Score: 14.69
Episode 785    Average Score: 15.02
Environment solved in 685 episodes!    Average Score: 15.02
```



5. Future Work and Improvements

To further improve agent performance or training stability, the following techniques could be explored:

- Implementing **Double DQN**
 - Applying **Learning Rate Scheduling**
 - Switching to **Rainbow DQN** (a combination of DQN improvements)
-

6. Files Submitted

- **Report:** `Report.pdf`
 - **Trained Model:** `checkpoint_dqn_dropout.pth` and `checkpoint Dueling_dqn_dropout.pth`
 - **Code Files:**
 - `agent.py`
 - `model.py`
 - `dueling_model.py`
 - `Navigation.ipynb`
 - `Navigation_Pixels.ipynb` (Not yet completed)
 - `run_agent.py`
 - `run_dueling_agent.py`
 - `utils.py`
-

7. References

- Udacity Deep Reinforcement Learning Nanodegree Course Materials
- Mnih et al., 2015. "Human-level control through deep reinforcement learning"
- Unity ML-Agents Toolkit Documentation
- OpenAI Spinning Up
- Sutton & Barto. "Reinforcement Learning: An Introduction"