Project 1: Navigation Report

Deep Reinforcement Learning Nanodegree

Aris Markogiannakis

February 10, 2026

1 Introduction

The goal of this project was to train an agent to navigate and collect yellow bananas while avoiding blue bananas in a large, square world. The environment provides a reward of $+1$ for yellow bananas and -1 for blue bananas. The state space has 37 dimensions (velocity and ray-based perception), and the action space consists of 4 discrete actions: move forward, backward, turn left, and turn right. The environment is considered solved when the agent achieves an average score of $+13$ over 100 consecutive episodes.

2 Learning Algorithm

2.1 Dueling Double DQN (D3QN)

The implemented agent utilizes a **Dueling Q-Network** architecture. Unlike standard DQN, which estimates the Q-value for each action directly, the Dueling architecture decomposes the Q-value into two streams:

1. **Value Stream** $V(s)$: Estimates the value of being in a certain state.
2. **Advantage Stream** $A(s, a)$: Estimates the relative advantage of taking a specific action compared to others.

The final Q-value is combined using the following formula to ensure identifiability:

$$Q(s, a) = V(s) + \left(A(s, a) - \frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}} A(s, a') \right)$$

2.2 Hyperparameters

The following hyperparameters were used during the training process:

2.3 Neural Network Architecture

The model consists of a shared input layer followed by two separate streams:

- **Input Layer**: Linear(37, 32) $\rightarrow$ ReLU
- **Value Stream**: Linear(32, 32) $\rightarrow$ ReLU $\rightarrow$ Linear(32, 1)
- **Advantage Stream**: Linear(32, 32) $\rightarrow$ ReLU $\rightarrow$ Linear(32, 4)

Hyperparameter	Value
Replay Buffer Size	1×10^5
Batch Size	64
γ (Discount Factor)	0.99
τ (Soft Update Rate)	1×10^{-3}
Learning Rate	5×10^{-4}
Update Every	4 steps
FC1 Units	32
FC2 Units	32
Epsilon Start	1.0
Epsilon End	0.01
Epsilon Decay	0.995

Table 1: Agent Hyperparameters

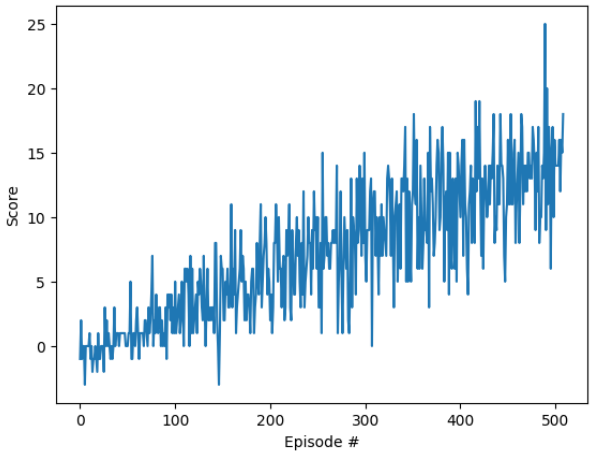

Figure 1: Training progress showing score per episode.

3 Results

The agent successfully solved the environment by reaching an average score of +13 over 100 episodes.

- **Episodes to Solve**: 409 episodes.

4 Future Work

To further improve the agent's performance and stability, the following enhancements could be implemented:

- **Prioritized Experience Replay**: Instead of sampling uniformly, transitions with higher temporal-difference (TD) error could be sampled more frequently.
- **Rainbow DQN**: Combining multiple improvements such as Noisy Nets, Distributional Reinforcement Learning, and Multi-step learning.

- **Hyperparameter Tuning**: Systematic grid search for optimal hidden layer sizes and learning rates.