

# Deep Reinforcement Learning Nanodegree

## Project 1 – Navigation Report

Chris Cadonic  
chriscadonic@gmail.com

July 4, 2020

## 1 Introduction

As part of the first project for the deep reinforcement learning nanodegree, I built and trained an agent that learns how to navigate through a world with yellow and blue bananas in an effort to maximize how many yellow bananas it can acquire.

### 1.1 Environment Overview

In the Unity-ML environment provided, the agent navigates through a world filled with yellow and blue bananas. As the agent moves through the world, if it comes into contact with a banana it will pick it up, receiving a reward of +1 if it picks up a yellow banana and -1 if it picks up a blue banana. Its goal is to maximize its reward for a single play session where it navigates and tries to pick up as many yellow bananas while avoiding blue bananas. The game is considered solved if it achieves an average score over 100 games of **+13**.

As the agent interacts, it receives information from the environment of the form: **insert state info**

Interactions available to the agent include moving in one of four directions:

- move forward,
- move backward,
- turn left,
- turn right,

with these four actions forming the action space  $\mathcal{A}$ . The agent solves the game by understanding what the optimal action is to take in each of the possible states that it finds itself in.

## 2 Approach

Initially I had approached this problem by first attempting to use Q-learning as a first benchmark, but given the continuous nature of the state space  $\mathcal{S}$  this would also necessitate discretization or tile coding the space. For this reason, I instead chose to implement a vanilla DQN network and iteratively improve it to achieve better performance.

Figure 1

## 3 Results and Discussion

### 3.1 Learning Performance

Figure 2

### 3.2 Next Steps

## 4 Conclusion

## References