Introduction

Reinforcement Learning involves a software agent to learn in an environment the various decisions/actions it should take to optimize a well-defined objective. The traditional algorithms involved working with finite State-Action space, thereby constraining the applications of these techniques. The novel techniques use a Deep Neural Network to approximate the function corresponding to the State-Action mapping, thereby removing the above constraints.

Deep Q Network

We employ a Deep Q Network with the following two important features:

- Experience Replay - Using a Replay Buffer and Random Sample State, Action, next State/Action, Reward Tuple
- Fixed Q Targets - Use of an extra Neural Net to prevent the correlation of weights and gradients.

Software

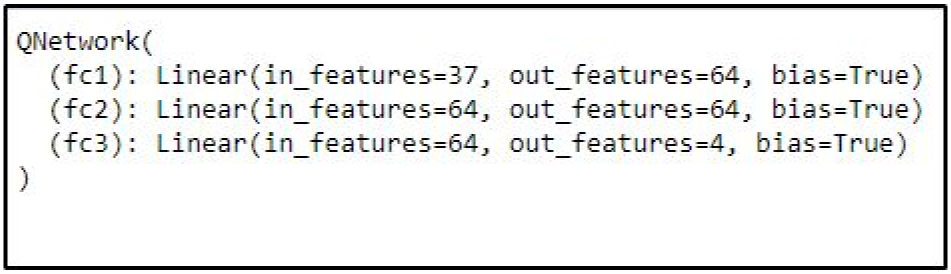
We use **Unity** Environment to train our smart software agents.

Problem Statement

The environment has both blue and yellow bananas in a square world. The objective is to find yellow bananas(a reward of +1 associated). The blue bananas have a reward of -1 associated. The agent can take four actions of Moving Forward, Left, Right, and Backward. The state space is thirty-seven dimensional. This is an episodic task and we must get an average score of +13 over 100 consecutive episodes.

Implemented Solution

We implemented a Deep Q Network with Experience Replay and Fixed Q Targets.
The architecture of Deep Q Network:



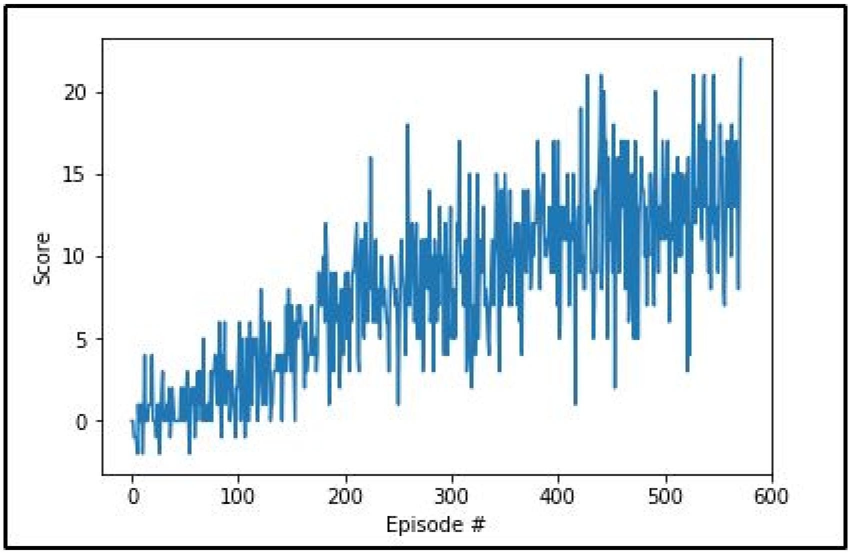
```
QNetwork(
  (fc1): Linear(in_features=37, out_features=64, bias=True)
  (fc2): Linear(in_features=64, out_features=64, bias=True)
  (fc3): Linear(in_features=64, out_features=4, bias=True)
)
```

The Q Networks are implemented using Pytorch Library. We use a Dequeue to serve as the Replay Buffer. Q Networks approximate the function representing the State-Action space and are topped with an epsilon-greedy selection strategy.

The various hyperparameters are listed as below

Buffer Size(replay buffer size)	1e5
Batch Size(Mini Batch)	64
Gamma(Discount Factor)	0.99
Tau(Soft Update of Target Params)	1e-3
Learning Rate	5e-4
Update Target Q Network Interval	4
Number Of Episodes	2000
Epsilon Decay	0.995

We get a requisite score of +13 over 100 episodes in just 472 episodes.
The episode to scores plot is shown below:



Future Work

To implement Double DQN, Prioritized Experience Replay.