Project Report: Navigation

A Deep Q Network (DQN) was used to solve the problem. A feedforward neural network consisting of 2 hidden layers learns to detect abstract features given the current state of the environment. The output of the neural network contains one node for every possible action the agent can take; mapping the abstract features into a set of actions, which is the best prediction of the reward for each action.

input (37) -> 64 -> 64 -> output (4)

input here is the current state which is a vector of 37 numbers, while the output size is 4.

Optimization is achieved via gradient descent, with the loss function as below:

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s') \sim \mathrm{U}(D)} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i) \right)^2 \right]$$

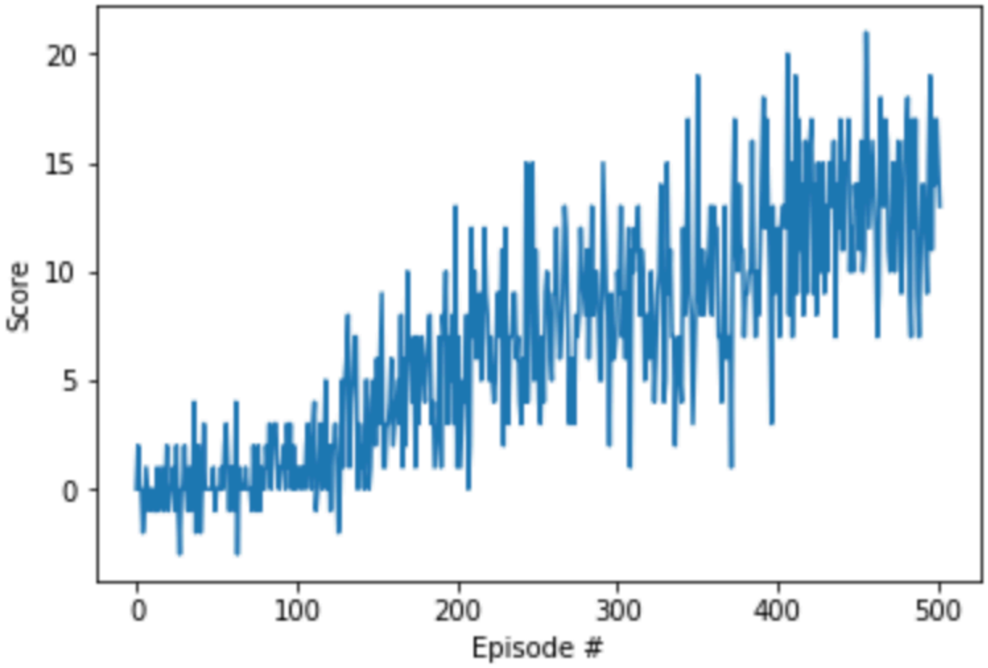
This optimization is essentially the difference between “true values” and our predictions (using a Q-Network). This is where Reinforcement Learning appears to resemble Supervised Learning; the need for a labeled dataset.
However, there really isn’t a label dataset per se, the “true values” are predictions from a copy of a Q-Network whose weights have been frozen in time. This machination, alongside experience replay (training the Q-Network with a batch of previously stored experiences) allows us to mimic Supervised Learning, albeit not completely.
While the Q-Network is trained continuously, the target network’s weights are updated with a copy of the Q-Network’s after a number of steps; for my experiment, this is after every 4 timesteps. And for learning rate, I used 0.0005. Below are other parameters I used.

BUFFER_SIZE = 100000; This is how much experience should be stored.
BATCH_SIZE = 64; How much data points to train the Q-Network with.

GAMMA = 0.99; This determines the importance of future rewards. With numbers approaching 1 favouring long-term rewards, while those close to 0 favours current reward.

Rewards

Using the hyperparameters above, the agent solved the problem in 402 episodes with a mean score over 100 episodes of 13.01. Below is a plot of the rewards.



Ideas for Future Work

While a vanilla DQN was used to solve this problem, a Double DQN will temper the over estimation of q-values the approach used here suffers from.