Report

Description of the implementation

This implementation uses a deep Q network (DQN) as a surrogate for a conventional Q table to train an agent to perform the required task, collection of yellow bananas from the Banana environment which is part of the UnityEnvironments tool set. It is based on a previous exercise (Lunar Lander) with some minor changes to accommodate the new state model.

Approach

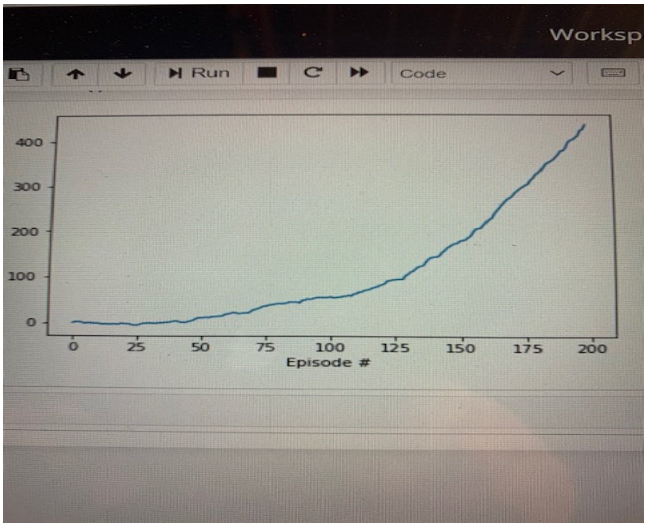
The model was run over 2000 episodes with a maximum of 1000 intervals within each episode. The behavior policy during training was e-greedy with epsilon annealed linearly from 1.0 to 0.01 with decay 0.995. Gamma was set to 0.99

The DQN consisted of an input layer of 37 states, fully connected to a first hidden layer (fc1) of 64 units then a second layer again of 64 units. The output layer consisted of 4 units corresponding to the action space. The first two layers used a ReLU activation and the final layer fc3 left as logits. The loss function for gradient descent was the Mean Squared Error (MSE).

Learning was carried out every 4 iterations using randomly sampled experiences stored in e Replaybuffer. Model parameters were updated using algorithm:

> Theta_target = Tau * Theta_local + (1 – tau) * Theta_target,
> where theta is the set of model weights

Plot of rewards



Ideas for future work

The model was very much a first iteration using essentially the same hyper-parameters as the previous LunarLander exercise, so clearly there are opportunities to change the key hyperparameters (Epsilon, epsilon decay, gamma and learning update step) to achieve better/faster results.

Further development would include a different neural network configuration and potentially a different optimiser.