

Banana's Everywhere: A Deep Reinforcement Learning Implementation

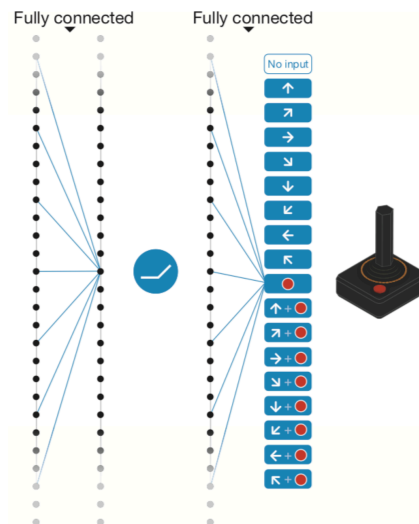
Hunter Kempf

General Overview:

I solved the banana world environment using Deep Q Learning in 105 episodes. My approach for speeding up the learning was to tune the epsilon decay lower from .995 to .985 which reduced the episodes needed for training from 442 to 105!

Technical Details :

My Implementation of Deep Q Learning uses a state input from the unity engine and passes it through a 2 hidden layer Neural Net to approximate the Q Table. The below illustration shows an example of how this network would work.



The training is done until the agent can consistently score above 13 (if the average over 100 episodes is over 13).

Each episode in my training method initializes a new environment and has the agent choose an action to take, computes the next state and reward received and cumulative reward then has the agent take a step.

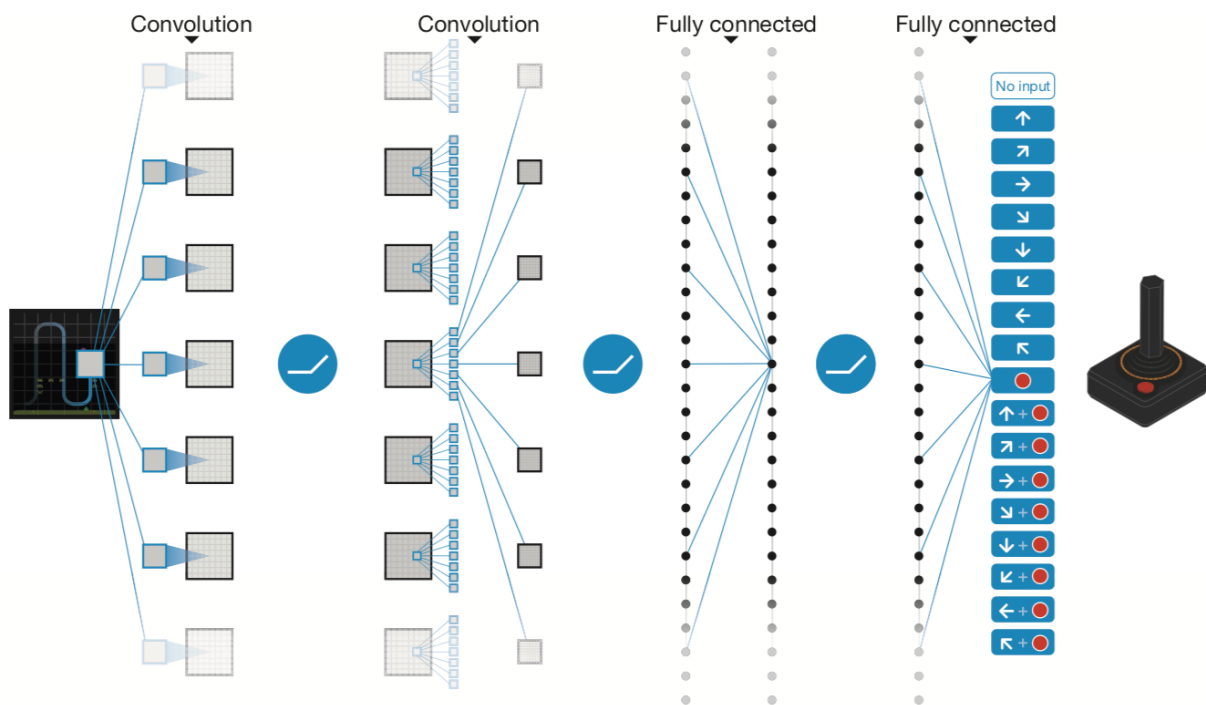
The process of the agent taking a step is relatively simple. The agent adds the state, action, reward tuple to its memory buffer then every 4 steps initiates a learn protocol where it samples 64 steps from its memory. With those steps it learns using the Q Table Approximation Neural Net and a learning rate gamma. Using those 64 sample steps the Q network is optimized to minimize MSE loss. The weights of the Q network are updated using a soft update meaning that

the target weights are mixed with the current weights to smooth out any changes and allow the weights to converge. Then the whole process starts over with a new action chosen.

Once an episode finishes the total reward gained is computed and stored to show the trend of the model improvement.

Further Improvements

Currently I am ingesting the state and passing it through two fully connected layers and outputting an action. A further improvement could be made to take in the pixels instead of the state and use convolutional layers to determine a 'state' as shown in the diagram below.



I could Also implement Double DQN, Prioritized Experience Replay or Dueling DQN which would all help to improve the Deep Q Learning Model.