An introduction to Reinforcement Learning

Oana Florescu, Tudor Berariu
oana.florescu28@gmail.com, tudor.berariu@gmail.com

Introduction

The goal of reinforcement learning is for an agent to learn to solve a given task by maximizing some notion of external reward. Instead of receiving explicit instructions, the agent learns how to choose actions by exploring and interacting with the environment. The reward signal serves as a way to encode whether the actions taken by the agent were successful.

Approach: We'll be exploring different algorithms(DQN, A2C) and some of their variations in trying to solve Open AI Gym environments CartPole-v0 and CarRacing-v0.

Off Policy Algorithms

On-policy algorithms - i.e. algorithms that learn from data that were generated with the current policy. Off-policy algorithms, on the other hand, are able to learn from experiences (e.g. transitions of the form (s,a,r,s*)) collected from previous policies. Because off-policy methods are able to reuse old data, they tend to be more sample-efficient than on-policy methods.

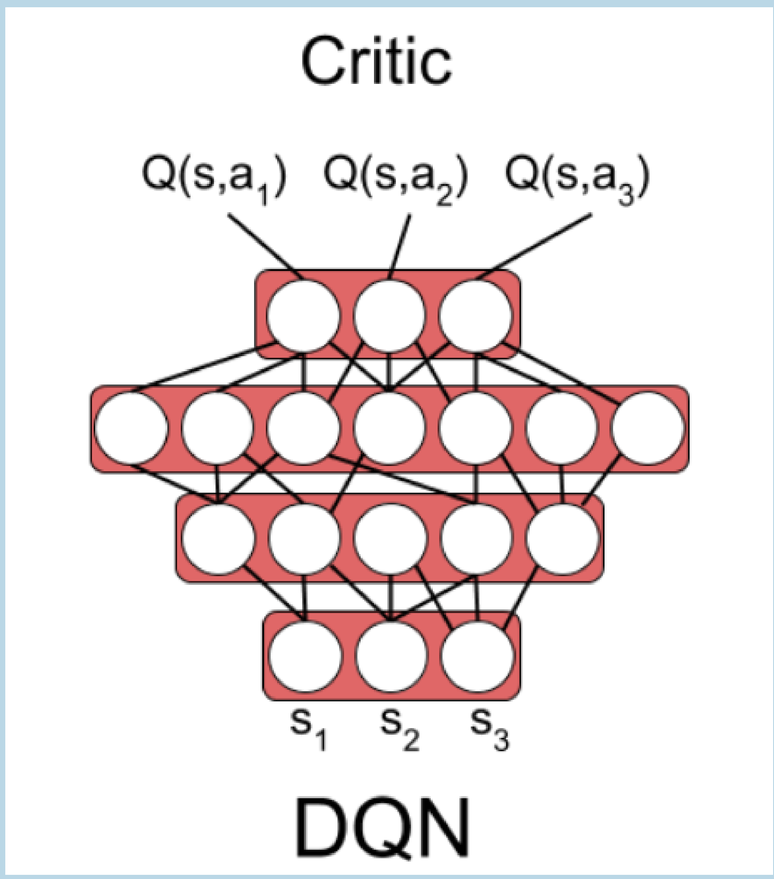
So how does Deep Q-learning work? The core of the algorithm involves the computation of the **temporal difference** (TD) error for transitions (s_i, a_i, r_i, s_{i+1}) sampled from taking actions in the environment:

$$\delta_i = y_i - Q_\phi(s_i, a_i) \tag{1}$$

where $y_i = r_i + \gamma amax Q_{\phi'}(s_{i+1}, a)$ is a bootstrapped estimate of the Q function. Similar to supervised learning, we minimize the squared loss between the **target values**y_i and the outputs of the network $Q_\phi(s_i, a_i)$

$$L = \frac{1}{N}\sum_i (r_i + \gamma amax Q_{\phi'}(s_{i+1}, a) - Q_\phi(s_i, a_i))^2. \tag{2}$$

Tricks: **Stable Target Network, Replay Memory, Stacked Frames**



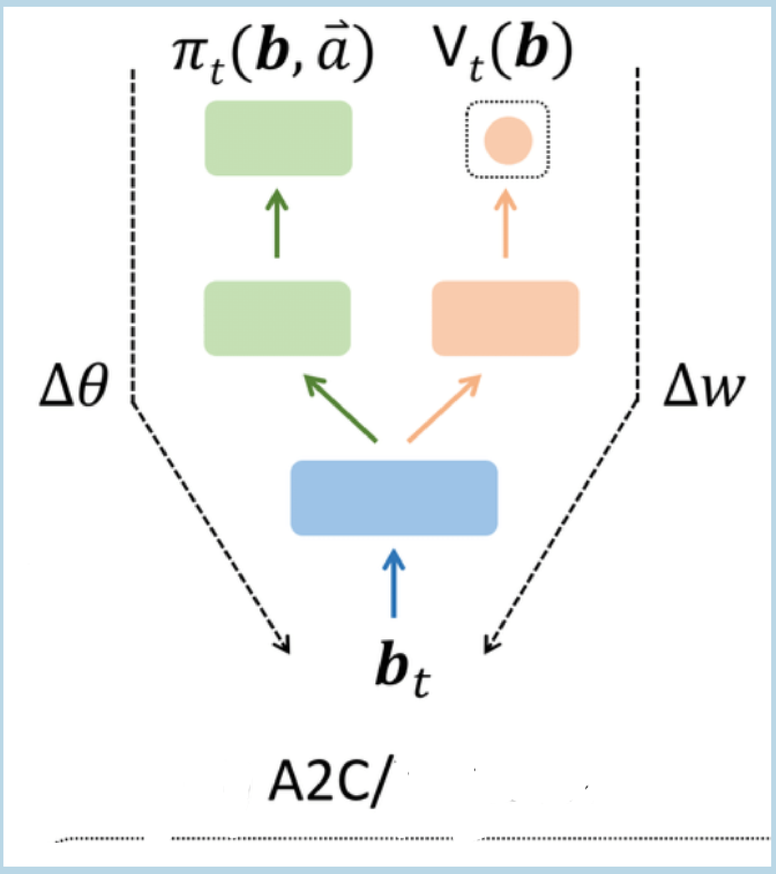
On Policy Algorithm

Advantage Actor-Critic Algorithm (A2C) A2C uses something called an advantage function to estimate the policy gradient

$$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t). \tag{3}$$

The advantage tells us, on average, how much better it is to choose action a_t . In other words, if a_t is better than average then we should increase its probability. Likewise, if it is worse than average then we should decrease its probability.
Here, we are approximating the advantage function A_t by using the critic to bootstrap an estimate of Q^π:

$$A^\pi(s_t, a_t) \approx r(s_t, a_t) + \gamma V^\pi_\phi(s_{t+1}) - V^\pi_\phi(s_t). \tag{4}$$



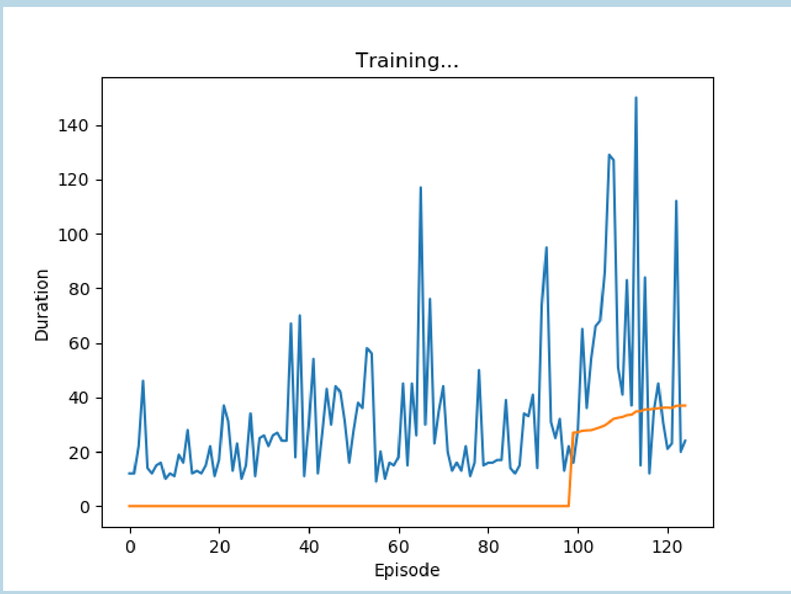
Data

The input data to the DQN model are raw pixel values (game screen) provided by the OpenAI Gym CartPole-v0/ CarRacing-v0 Emulator
The input data provied is pre-processed rescaling it to have values in [0,1] and cropping it to have size 40x90x3
The input data for the A2C model are observations return from the environment: tuple of 4 floats

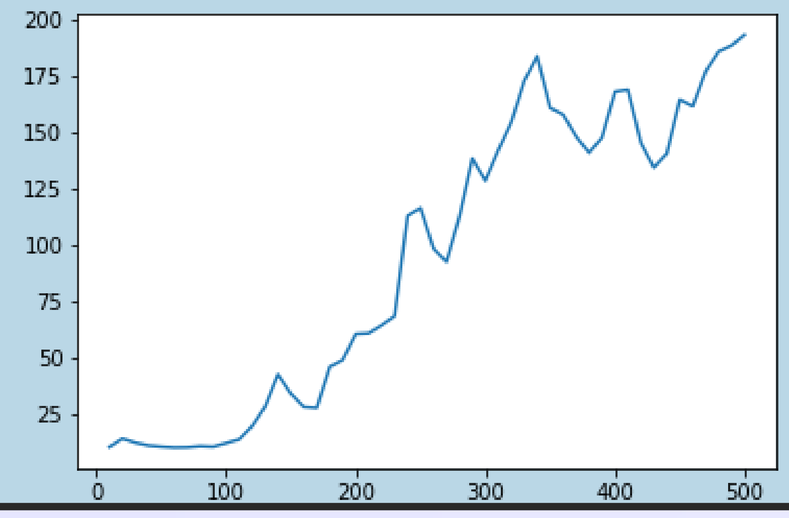
Observations

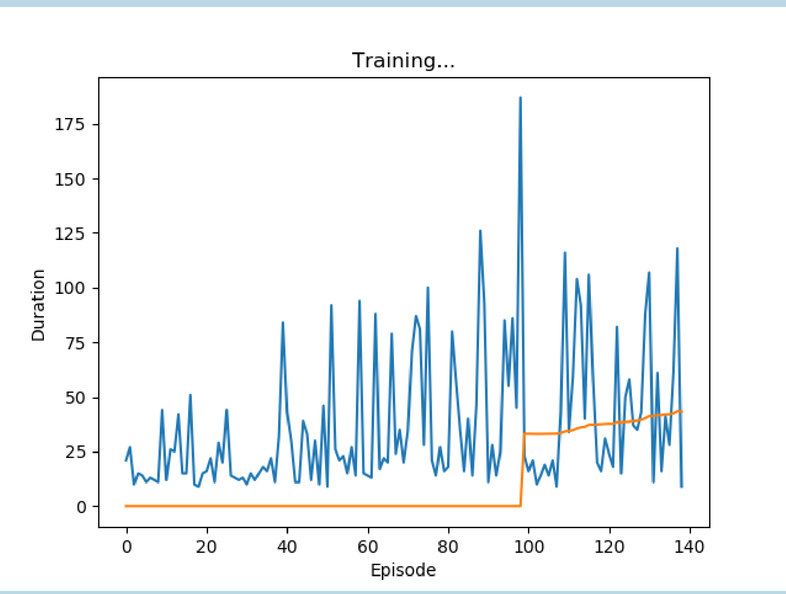
*One major drawback of Deep Q Networks is that they can only handle low-dimensional, discrete action spaces. CarRacing-v0 is a continuous task but easy to discretize.
*Normalizing rewards, returns, and/or advantages is always worth trying
* Substracting a baseline in A2C reduces variance.

Results

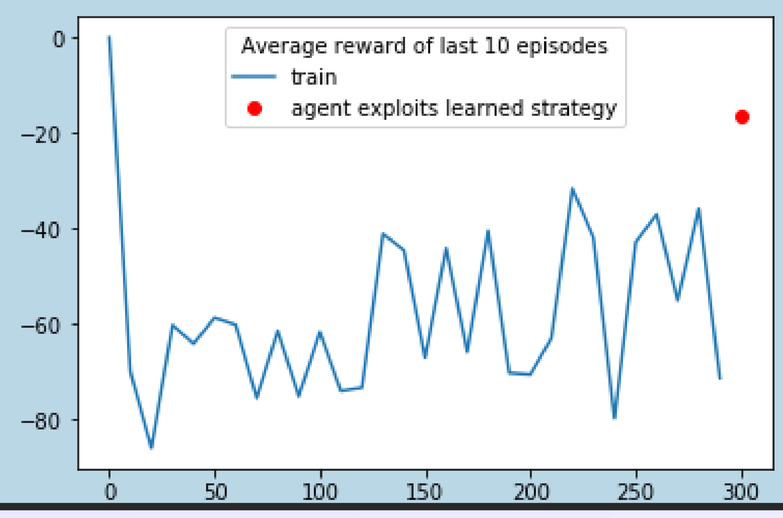


DQN on CartPole-v0
Actor Critic on CartPole-v0





DoubleDQN on CartPole-v0
DQN on CarRacing-v0



References

[1] Sutton Barto: *Reinforcement Learning: An introduction* 2014
[2] Demis Hassabis et al*Human-level control through deep reinforcement learning*