Revisiting State Augmentation methods for Reinforcement Learning with Stochastic Delays

Somjit Nath [1], **Mayank Baranwal**[1,2], **Harshad Khadilkar**[1,2]

[1]TCS Research, [2]IIT Bombay

Introduction

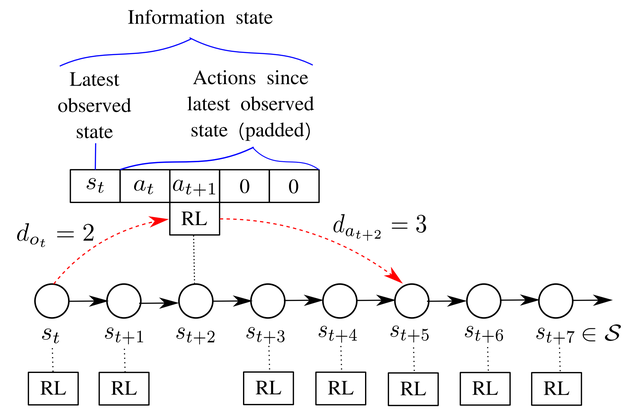
In Reinforcement Learning, we mainly consider the environment and the agent to be in sync with each other. Most Reinforcement Learning(RL) algorithms fail to learn anything substantial due to the presence of delays.
In some domains, waiting for the most recent observation or for the most recent action to get executed is fine, but in most domains, particularly in real world scenarios, the environment does not respond well to such 'waiting' actions. Thus handling delay in RL is crucial aspects to enable RL to be used in a variety of real world problems like congestion control, control of robotic systems, distributed computing, medical domains etc.

State Augmentation

RL algorithms model the environment as a Markov Decision Process (MDP). In the presence of delays, one can equivalently model the underlying problem as partially observable MDP (POMDP). POMPDs are generalizations of MDPs, however solving POMDPs without estimating hidden action states leads to arbitrary sub-optimal policies. In the presence of delays, it becomes imperative to add the delayed observation with the non-implemented actions to make the states Markov. The reformulation allows an MDP with delays to be viewed as an equivalent MDP without delays.
Some Key Points to note:

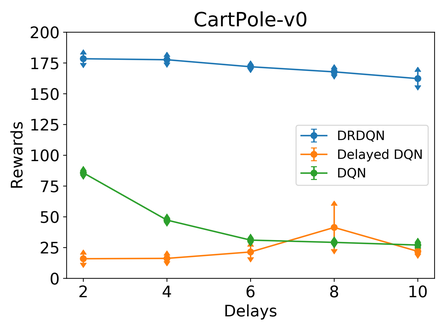
- We assume the delay value at every time step is not known to us, thus the number of action we need to augment is also not known.
- State Augmentation increases the state space exponentially, so we use a neural network to append the actions directly to the state, thus making the state dimension grow linearly with delays.

Delay Resolved Algorithms



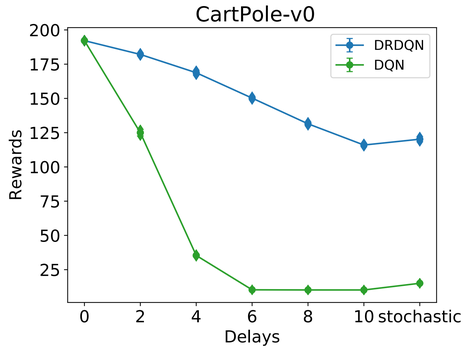
The figure is a graphical illustration of how state augmentation works. An action is computed in every time step, using the latest known state and a vector of all actions taken since that state was observed.

Results with Constant Action Delays



The figure above is a plot of the rewards across varying action delays for the CartPole environment. One of the algorithms compared is Delayed DQN (Dalal et al 2021) which uses forward models to predict the unobserved states coming from the un-implemented actions.
Delayed DQN is very dependent on how good the forward models learn and if it fails to learn, then it can lead to poor performance.

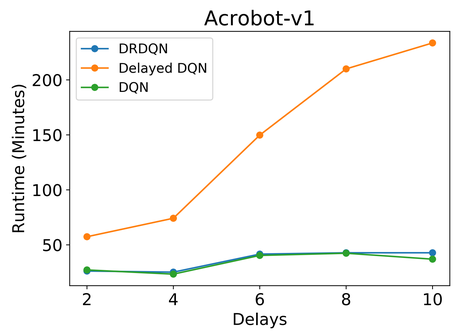
Results with Observation Delays



The figure above shows the comparison with baseline DQN with both constant and stochastic delays. DRDQN reaches optimal performance for constant and stochastic delays.

Compute Comparison on Acrobot

DRDQN requires almost similar time as DQN whereas Delayed DQN with forward models requires more time and the gap increases for higher delays.



Conclusion

In the main paper, we formally define both constant and stochastic delays. We adapt the state augmentation methods to other domains as well.