Neural Network based Finite Time Guarantees for Continuous State MDPs with Generative Model

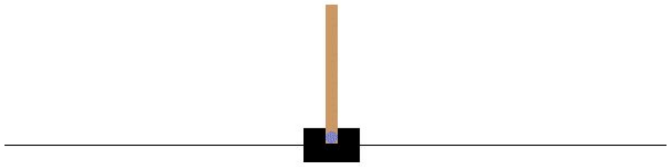
Harshita Arya

Introduction

The Two Approaches:

Finite Time Guarantees for Continuous State MDPs with Generative Model

Neural fitted Q Iteration- first Experience with Data Efficient Neural Reinforcement Learned Method



Introduction

Neural Network based OneVAL

An ‘online’ reinforcement learning algorithm for continuous MDPs that is ‘quasi-model-free’ that can compute nearly-optimal policies and comes with non-asymptotic performance guarantees including prescriptions on required sample complexity for specified performance bounds. The algorithm relies on use of a ‘fully’ randomized policy that will generate a β-mixing sample trajectory.

Introduction

Neural Fitted Q Iteration

Efficient and Effective training of a Q-value function represented by a multi-layer percep-tron. Based on the principle of storing and reusing transition experiences,a model-free, neural network based Reinforcement Learning algorithm.It is shown empirically, that reasonably few interactions with the plant are needed to generate control policies of high quality

Setup

The Experiment is done on Continuous CartPole Environment for both the approaches

The Environment is created using gymnasium Spaces with the following configuration:

```
self.gravity = 9.8
self.masscart = 1.0
self.masspole = 0.1
self.total_mass = self.masspole + self.masscart
self.length = 0.5  # actually half the pole's length
self.polemass_length = self.masspole * self.length
self.force_mag = 10.0
self.tau = 0.02  # seconds between state updates
self.kinematics_integrator = "euler"
```

Approach and Setup

Neural Network based Finite Time Guarantees for Continuous State MDPs with Generative Model:

- Select sample from the interaction using random Beta Sampling which is a similar approach to the replay buffer.
- The Exploration scheme used is epsilon greedy and decaying epsilon.
- Perform Value Iteration using the Bellman Equation
- Function Approximation using Neural network based Approach

Approach and Setup

The approach for Neural fitted Q Iteration Algorithm is as follows:

- Generation of the training set P

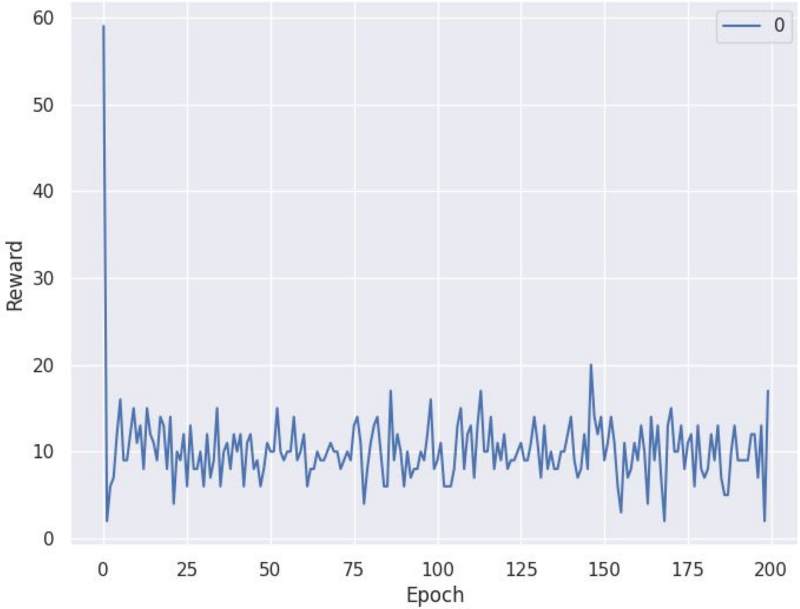
  generate pattern set P = {(input , target), l = 1, . . . , #D} where:

  input= sl , ul ,

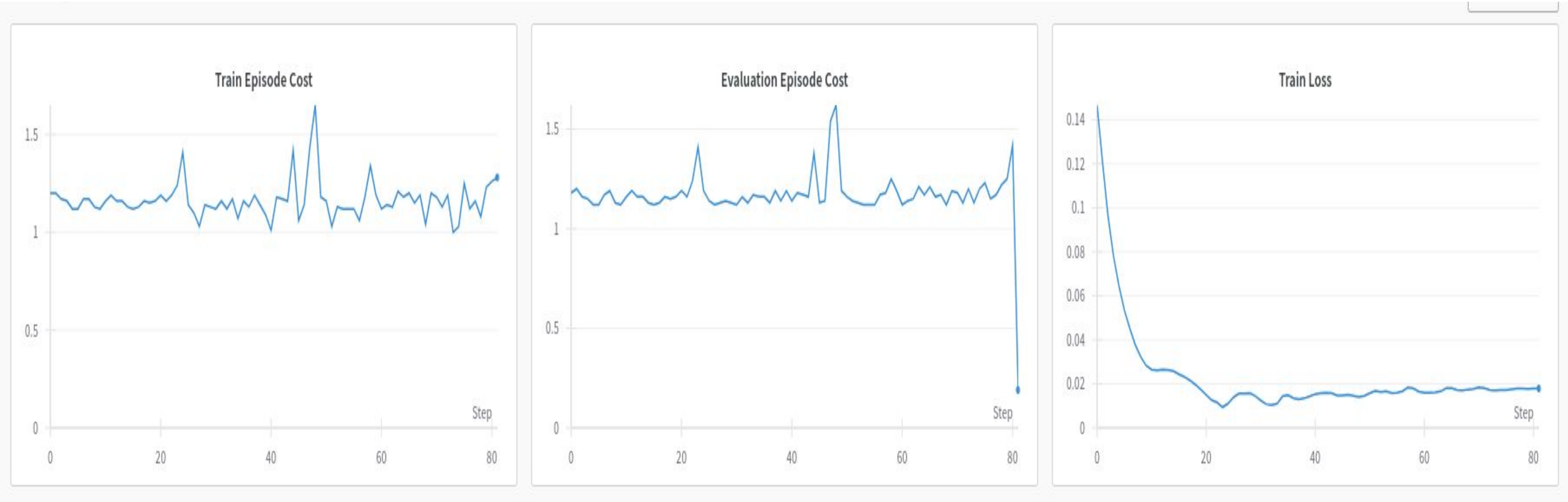
  target = c(sl , ul , sl) + γ minb Qk (sl , b)

- A Supervised Learning Approach Training using RProp

Results



Results



Train Episode Cost
Evaluation Episode Cost
Train Loss
Step

Conclusion

OneVAL is good for Online Learning with Stable Behaviour

NeuroFitted Q Iteration is good for Offline Learning with Fast Convergence.

- Historical Data Analysis: Market Analysis, Customer Behaviour Analysis

Future Work

Use the Continuous MoonLander Environment for both Approaches