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

Harshita Arya

1. Introduction and Motivation

This work is based on the approach proposed by Sharma, et al. 2020 in [1], with the key difference being that we take a Neural Network based approach for fitting the value function over the state space.We derive from an Online Empirical Value Learning (ONEVaL), a 'quasi-model-free' reinforcement learning algorithm for continuous MDPs. It requires a generative model to sample from but not the full model specification. ONEVaL can compute near-optimal policies and comes with theoretical performance guarantees on sample complexity to achieve a desired level of performance.The algorithm relies on using a fully randomized policy to generate samples that have the β-mixing property. It also uses randomized function approximation in a RKHS to achieve arbitrarily small approximation error. The value function is estimated 'empirically' by taking several samples of the next state using the generative model.

We finally try to contrast our work with the existing [2], where NFQ employs a neural network for the Q-function along with experience replay to enable effective and efficient training that requires fewer interactions with the environment. We will try to contrast both these approaches by evaluating on benchmark problems that exhibit the sample efficiency of these approaches.

2. Background and Problem Statement

When applying reinforcement learning to real-world problems, an important challenge is finding an appropriate representation for the value function. Multi-layer perceptrons are an appealing choice due to their ability to approximate nonlinear functions. However, despite successful applications, multilayer perceptrons also come with difficulties. A key issue is that the representation in multilayer perceptrons is global rather than local. This can lead to problems such as instability in learning, lack of generalization, and difficulty in scaling to large problems. To address the limitations of global value function approximations like neural networks, an alternative is to use more localized representations that can exploit the structure of a problem. Developing reinforcement learning algorithms that can effectively leverage both local and global value function representations remains an active area of research.

In order to exploit only the good sides of the global approximation through a multi layer perceptron we address the issues with using multi-layer perceptrons for value function approximation is to constrain the influence of each update to the neural network. The core idea underlying our proposed method is simple: when making an update at a new datapoint, we also explicitly retain prior knowledge. We implement this concept by storing all previous state-action transition experiences in memory. This experience data is then reused every time the neural network representing the Q-function is updated. By reusing prior experiences, we can regularize the network updates to prevent drastic changes and preserve previously learned representations. This experience replay mechanism allows us to leverage the generalization capability of neural networks while overcoming common challenges like instability and interference arising from their global representation.

We will consider a discounted MDP (X , A, P, r, γ) where X is the state space and A is the action space. The transition probability kernel is given by P (·|x, a), i.e., if action a is executed in state x, the probability that the next state is in a Borel-measurable set B is P (x_{t+1} ∈ B|x_t = x, a_t = a) where x_t and at are the state and action at time t. The reward function is r : X × A → R. We are interested in maximizing the infinite horizon expected discounted reward where the discount parameter is γ. Let Π denote the class of stationary deterministic Markov policies mappings π : X → A which only depend on history through the current state. We only consider such policies since it is well known that there is an optimal MDP policy in this class. When the initial state is given, any policy π determines a probability measure P π . Let the expectation with respect to this measure be E^{π} . We focus on the infinite horizon discounted reward criterion. The expected infinite horizon discounted reward or the value function for a policy π and initial state x is given as

$$v^{\pi}(x) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t}\, r(x_t, a_t) \middle| x_0 = x\right]$$

The optimal value function is given as v^* (x) =sup$_{\pi\in\Pi \, v \, \pi}$ (x) and the policy which maximizes the value function is the optimal policy, π^* .

We will assume that we have interactions (x_1 , x_2 , . . . x_N , x_{N+1} , ...) generated by a randomized policy π_g i.e., x_{t+1} ~ P (·|x_t , a_t) where at ~ π_g (·|x_t). We assume that for any a and x, π_g (a|x) is strictly positive. We keep a window of size N which moves forward one step in every iteration. Now these N samples serve as the states for which we will compute our approximate value function and then use function approximation to generalize.

3. Work to be Performed

As we will try to build upon [1] , we see that it tries to fit the value function over the state space by computing the best fit within F^(θ)$^{1:J}$

$$\begin{aligned}&\min_{\alpha} \frac{1}{N}\sum_{n=1}^{N} |\sum_{j=1}^{J} \alpha_j \phi(x_n;\, \theta_j) - \widehat{v}(x_n)\,|^2 \\ &\text{s.t.} \quad \|\,(\alpha_1, \ldots, \alpha_J)\,\|_{\infty} \le C/J.\end{aligned}$$

Drawing inspiration from [2], instead we try to use a neural network based approach as our function approximator. We finally try to test our approach on standard problems like

1. The Pole Balancing Task
2. The Mountain Car Benchmark
3. The Cartpole Regulator Benchmark

And contrast the results with [2].

4. Results Expected

In control problems, three basic types of task specification might be distinguished.

1. Avoidance control task - keep the system somewhere within the ’valid’ region of state space. Pole balancing is typically defined as such a problem, where the task is to avoid that the pole crashes or the cart hits the boundary of the track.
2. Reaching a goal - the system has to reach a certain area in state space. As soon as it gets there, the task is immediately finished. Mountaincar is typically defined as getting the cart to a certain position up the hill.
3. Regulator problem - the system has to reach a certain region in state space and has to be actively kept there by the controller. This corresponds to the problems typically tackled with methods of classical control theory. The problem types show different levels of difficulty, even when the under-lying plant to be controlled is the same. In the following, we consider three benchmark problems, where each belongs to one of the above categories.

We consider three benchmark problems, where each belongs to one of the above categories.

Evaluating Learning Performance

Each learning experiment consists of a number of episodes. An episode is a sequence of control cycles, that starts with an initial state and ends if the current state fulfills some termination condition (e.g. the system reached its goal state or a failure occurred) or some maximum number of cycles has been reached. Learning time in principle will be measured in many different ways: number of episodes needed, number of cycles needed, number of updates performed, absolute computation time, etc.

Evaluating Controller Performance

Controller performance will be evaluated with respect to some cost-measure, that evaluates the average performance over a certain amount of control episodes. In principle this cost measure can be chosen arbitrarily. Due to its practical relevance, we will use the average time to the goal as a performance measure for the controller. In the regulator problem case, we measure the overall time outside the target region.

5. References.

[1] Hiteshi Sharma and Rahul Jain. *Finite Time Guarantees for Continuous State MDPs with Generative Model*. In 2020 59th IEEE Conference on Decision and Control (CDC).

[2]Martin Riedmiller. *Neural Fitted Q Iteration - First Experiences with a Data Efficient Neural Reinforcement Learning Method.* In ECML 2005.

[3] Hiteshi Sharma, Mehdi Jafarnia-Jahromi and Rahul Jain. *Approximate Relative Value Learning for Average-reward Continuous State MDPs.*

[4] Hiteshi Sharma, Rahul Jain and Abhishek Gupta. *An Empirical Relative Value Learning Algorithm for Non-parametric MDPs with Continuous State Space.*

[5] Boyan and Moore. *Generalization in reinforcement learning: Safely approx-imating the value function*. Advances in Neural Information Processing.