Q-Learning with Neural Networks for Robotic Arm Control

Your Name

March 7, 2025

Abstract

This report explores the use of Q-learning with neural networks (NN) for training a robotic arm model in a physics-accurate simulation. The objective is to demonstrate the advantages of using NN-based Q-learning over traditional Q-tables, particularly in high-dimensional robotic control problems. The Mujoco-based model and the Menagie physics engine were employed for accurate physics simulation.

1 Introduction

The intention of this project was to train a robotic arm using Q-learning with neural networks (NN) instead of a conventional Q-table. The complexity of the robot's state-action space necessitates a function approximator, which NN provides efficiently.

2 Methodology

The robotic arm was simulated using Mujoco, with Menagie serving as the physics engine. The model consists of multiple articulated joints, each actuated by motors. The XML model features:

- Six degrees of freedom: Base rotation, shoulder pitch, elbow movement, wrist pitch and roll, and jaw control.
- Defined friction loss and armature parameters for realistic motion.
- Motor constraints with position-based control.
- Contact modeling to simulate object interactions.

The Q-learning algorithm with NN was used to approximate the Q-value function, enabling generalization over large state-action spaces.

3 Mathematical Formulation

The Q-learning update rule is given by:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right] \tag{1}$$

where:

- $Q(s,a)$ is the Q-value for state s and action a.
- α is the learning rate.
- γ is the discount factor.
- r is the received reward.
- s' is the next state.

Instead of storing $Q(s,a)$ in a table, we approximate it using a neural network:

$$Q(s,a;\theta) \approx f(s,a;\theta) \tag{2}$$

where θ represents the NN parameters trained via backpropagation.

4 Why Use NN Instead of Q-table?

A traditional Q-table is infeasible for a high-dimensional state space as it requires an entry for each state-action pair. Given the continuous action space of the robotic arm, NN provides a better alternative by enabling function approximation and generalization, reducing memory requirements and training time.

5 Results and Discussion

Our Mujoco-based robotic arm demonstrated improved convergence rates when using Q-learning with NN compared to discrete Q-tables. The model successfully executed complex motion sequences by leveraging neural network-based value estimation.

6 Conclusion

This report presented an implementation of Q-learning with neural networks for robotic arm control. The Mujoco and Menagie-based setup allowed for realistic physics modeling, and NN-based Q-learning proved effective in handling high-dimensional control tasks. Future work can explore reinforcement learning variants such as DDPG or PPO for further improvements.

7 References

Include references to reinforcement learning and Mujoco documentation.