report

Leonardo Uchoa Pedreira

7/28/2021

Contents

Description 1

Algorithm: DQN and Q-Learning 1

Description

This document is a report describing the learning algorithm and details of implementation, along with ideas for future work.

Algorithm: DQN and Q-Learning

The algorithm used is called `DQN`, where a neural network is used is to implement the `Q-Learning` Algorithm. The `Q-Learning` algorithm attempts to estimate action-value pairs in order to maximize the expected total reward and, therefore, to obtain the optimal policy for the given task.

The `Q-Learning` algorithm belongs to class of value-based methods, whose goal is to solve the bellman equation. Solving the bellman equation gives us the optimal policy, given that our environment meets certain criteria in our Markov Decision Process setting. For `Q-Learning` in particular the equation we're trying to solve is as follows

$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\Bigg(\underbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}_{\text{new value (temporal difference target)}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \Bigg)}^{\text{temporal difference}}$$

So in a given time step `t` we search for the action that maximizes the action-value pair $Q(s_{t+1}, a)$ (and hence why this algorithm belongs to the tabular methods class)