Abstract

Many reinforcement-learning researchers treat the reward function as a part of the environment, meaning that the agent can only know the reward of a state if it encounters that state in a trial run. However, we argue that in many cases this is an unnecessary limitation and instead, the reward function should be provided to the learning algorithm. The advantage is that the algorithm can then use the reward function to check the reward for states that the agent has not yet encountered. In addition, the algorithm can simultaneously learn policies for multiple reward functions. For each state, the algorithm would calculate the reward using each of the reward functions and add the rewards to its experience replay dataset. The Hindsight Experience Replay algorithm [1] does just this, and learns to generalize across a distribution of sparse, goal-based rewards. We extend this algorithm to linearly-weighted, multi-objective rewards and learn a single policy that can generalize across all linear combinations of the multi-objective reward. We confirmed that our algorithm works by testing on the double-integrator control task, where the task was modified so that the algorithm must trade off between quickly reaching its destination and minimizing fuel use. Whereas other multi-objective algorithms are limited to environments with discrete actions, our algorithm can be also used with continuous actions.