Prioritized Sweeping Neural DynaQ: Boost Learning by Dreaming

Alexander Osiik
alexander.osiik@student.uni-luebeck.de
Seminar Cyber-Physical Systems (WS 2019/20)
Institute of Computer Engineering, University of Lübeck

December 24, 2019

Abstract

Hier beschreiben welche Hintergründe und Analogien aus der Biologie Reinforcement Learning hat.
State of the Art.
Was Zielsetzung im Original-Paper war, wie sie erreicht wurde und wie es im Projekt umgesetzt wurde.

1 Introduction

- Hier beschreiben welche Hintergründe und Analogien aus der Biologie Reinforcement Learning hat
- State of the Art, medizinische Aspekte, Forschungshintergründe
- Was Zielsetzung im Original-Paper, Übertragen der **hippocambal replays** auf RL.
  "... replay refers to the re-occurrence of a sequence of cell activations that also occurred during activity, but the replay has a much faster time scale."
- Aufbau des Experiments

2 Reinforcement Learning

2.1 Markov Decision Problem

A Markov Decision Problem (MDP) is a model for problems, where an agent tries to interact with the environment in such way that the utmost reward is achieved. Specifically, the robot is moving (*transition*) through various *states*, having chosen a specific *action* in each state. Each *reward* is determined by initial state, the action and the following state. All transitions are not deterministic, but rather probabilistic, where each probability only depends on the current state and the current action, see **Markov Assumption**. That way there has to be one initial state, but multiple end states are possible.

The main goal is to find a reward maximizing **policiy**, by which the agent selects actions where the maximum reward can be expected.

- S : Set of states $\{s_1, s_2, \ldots, s_n\}$
- A : Set of actions $\{a_1, a_2, \ldots, a_n\}$
- $T : S \times A \times S$: Transition function, which is the probability of going from state s to state s' via action a
- $R : S \times A \times S \to \mathbb{R}$: Reward function
- Π : Policy, where an optimal action is assigned to each state
- $\gamma \in [0, 1]$ Discount factor. This factor determines the degree of exploration for the agent. For $\gamma = 0$, the agent will stick to the policy, and explot the current (possibly) optimal policy. For $\gamma = 1$, the agent will take into account the next states reward, leading to exploration behaviour. A value ¡1 is a contraint, which limits the maximal obtainable reward, leading to a reduction of cycles.

The **V-Values** and **Q-Values** are certain grading schemes used to solve MDPs. It is the total reward the agent can expect, if it performs the optimal, or maximally benefitting, action a in state s, and continues to act optimally.

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} T(s, a, s')[R(s, a, s') + \gamma V^*(s')] \tag{1}$$

As the values from equation 1 is hard do compute, a technique called **Value Iteration** is used. It is used do discretize the equation:

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} T(s, a, s')[R(s, a, s') + \gamma V^*(s')] \tag{2}$$

where k is the radius from the goal state to the agent. For example, regarding the manhattan norm in a 2D plane, the amount of steps the agent has left until it reaches an end state. After that, **Policy Extraction** is performed. It is the assignment of an action to each state, maximizing the expected total reward.

$$\Pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} T(s, a, s')[R(s, a, s') + \gamma V^*(s')] \tag{3}$$

- Einführung **Markov Decision Problem**
- Erklärung der Notation für *state*, *action*, *transition function*, *learning rate*, *discount factor*
- Unterschiede *value iteration*, *policy extraction*, *prioritized sweeping*
- spätestens hier müsste die Bellman Gleichung stehen:
- Erklärung Q-Learning, Vorteile, Nachteile

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a' \in A} Q(s', a') - Q(s, a)]$$

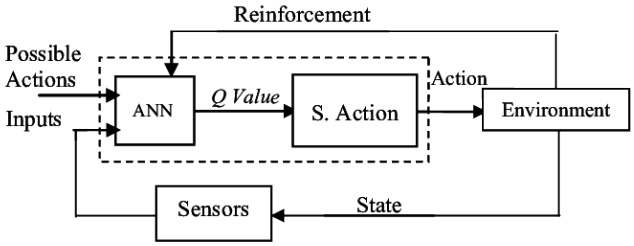

Figure 1: Basic Q-Learning.

- Exploration/Exploitation TradeOff und Techniken
- **HIER:** Idee des Papers: künstliche Erweiterung des State space umd RL für vergangenheitsabhängige Probleme anwendbar zu machen.

3 GALMO

- Vorstellung der Ergebnisse des Original Paper (GALMO)

4 Project

- Umsetzung von Q-Learning in Python
- DQN
- Implementierung GALMO?

5 Results

6 Conclusion

References