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

January 4, 2020

Abstract

Hier beschreiben welche Hintergrnde 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 Hintergrnde und Analogien aus der Biologie Reinforcement Learning hat
State of the Art, medizinische Aspekte, Forschungshintergrnde
Was Zielsetzung im Original-Paper, bertragen 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."
Main aspect of the work was to convert the reactivation of so called place cells. These cells are located in the hippocampus and are responsive to the current position of the animal within the environment. O'Keefe and Dostrovsky [1971] postulated that the hippocampus functions as a spatial map, where single hippocampal neurons increased their firing rate whenever a rat traversed a particular region of an environment, as concluded by Nakazawa et al. [2004].
Aubin et al. [2018] set up a experimental task, which was derived and slightly modified from Gupta et al. [2010]. The environment consisted of two succesive T-mazes with lateral return corridors and rewarding food pellets on each side, see Figure 1. A rat was placed in the maze, and trained to make a decision at position T2, with the objective of getting the reward on the left or right hand side based on the task pursued at the moment. The tasks were 1) always turn right, 2) always turn left, 3) alternate between left and right. At reward locations, the rat's hippocampal replays were analyzed. It has been shown that the rats reflected recently experienced trajectories, and, in addition, also those that occurred a longer time ago.

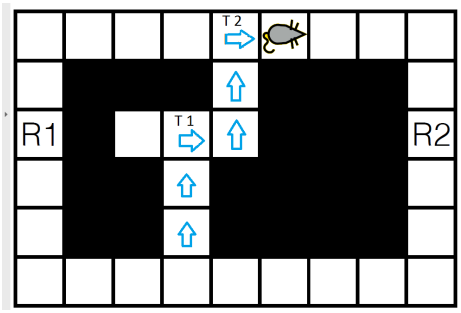

Figure 1: The maze is discretized into32 positions (squares). The agent can use 4 discrete actions (N,E,S,W). The inputstateis the concatenation of 32 location components and two reward memory com-ponents. The location part ofrepresents the activation of 32 place cells co-locatedwith the maze discrete positions, their activityactdepends on the Manhattan dis-tance of the agent to the cell. [Aubin et al., 2018]

To mathematiacally model and explain the hippocampal replay phenomenon, algorithms from the Dyna family of algorithms were used. Dyna is an integrated architecture for learning, planning and reacting, proposed by Sutton [1991], see Figure 2. The Dyna idea is a trivial approach that planning is equivalent to trying out multiple things in the own mind, under the condition that a certain internal model of the problem exists. Ultimately, this architecture was chosen because it is designed to make the best possible use of alternation between online and off-line learning phases [Sutton, 1991]. Aubin et al. [2018] concentrated on the Q-learning version of Dyna (Dyna-Q) extended by prioritized sweeping, by that optimizing the choice of reactivated cells, which will be discussed further in the next section.

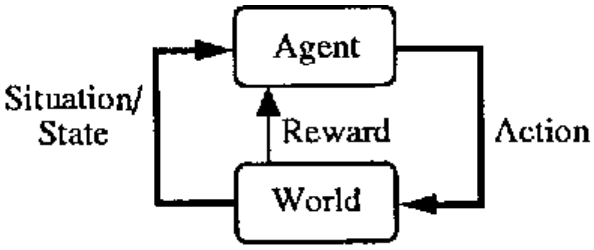

Figure 2: The Problem Formulation Used in Dyna. The agent's objective is to maximize the total reward it receives over time. [Sutton, 1991]

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.

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

- Einfhrung **Markov Decision Problem**
- Erklrung der Notation fr *state*, *action*, *transition function*, *learning rate*, *discount factor*

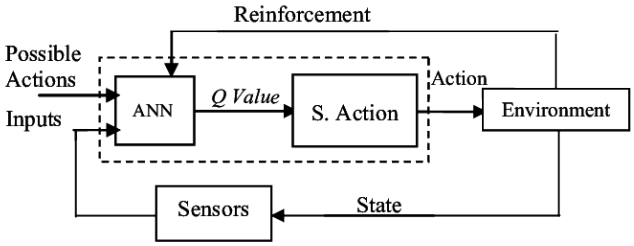

Figure 3: Basic Q-Learning.

- Unterschiede *value iteration*, *policy extraction*, *prioritized sweeping*
- sptestens hier msste die Bellman Gleichung stehen:
- Erklrung Q-Learning, Vorteile, Nachteile

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

- Exploration/Exploitation TradeOff und Techniken
- **HIER:** Idee des Papers: knstliche Erweiterung des State space umd RL fr vergangenheitsabhngige 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

L. Aubin, M. Khamassi, and B. Girard. Prioritized sweeping neural dynaq with multiple predecessors, and hippocampal replays. In *Living Machines 2018*, LNAI, page TBA, Paris, France, 2018. URL https://hal.archives-ouvertes.fr/hal-01709275.

Anoopum S. Gupta, Matthijs A.A. van der Meer, David S. Touretzky, and A. David Redish. Hippocampal replay is not a simple function of experience. *Neuron*, 65(5):695 – 705, 2010. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2010.01.034. URL http://www.sciencedirect.com/science/article/pii/S0896627310000607.

Kazu Nakazawa, Thomas Mchugh, Matthew Wilson, and Susumu Tonegawa. Nmda receptors, place cells and hippocampal spatial memory. *Nature reviews. Neuroscience*, 5:361–72, 06 2004. doi: 10.1038/nrn1385.

J. O'Keefe and J. Dostrovsky. The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171 – 175, 1971. ISSN 0006-8993. doi: https://doi.org/10.1016/0006-8993(71)90358-1. URL http://www.sciencedirect.com/science/article/pii/0006899371903581.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.*, 2(4):160163, July 1991. ISSN 0163-5719. doi: 10.1145/122344.122377. URL https://doi.org/10.1145/122344.122377.