Machine Learning

Practical work 14 - Reinforcement learning

Teachers: A. Perez-Uribe (Email: andres.perez-uribe@heig-vd.ch) & J. Hennebert
Assistant: Yasaman Izadmehr & Shabnam Ataee

0. Reinforcement learning tutorial using OpenAI gym

For this practical work, we ask you to use **Google Colab** as your development environment. Gym is a toolkit developed by OpenAI for training and comparing reinforcement learning algorithms. It supports a large set of Atari-like games as well as other learning tasks like.
To start using a Python library providing these simulation environments and supporting reinforcement learning, we propose you to follow this tutorial:

https://www.learndatasci.com/tutorials/reinforcement-q-learning-scratch-python-openai-gym/

1. The taxi pick-up and drop problem

"There are 4 locations (labeled by different letters), and our job is to pick up the passenger at one location and drop him off at another. We receive +20 points for a successful drop-off and lose 1 point for every time-step it takes. There is also a 10 point penalty for illegal pick-up and drop-off actions."

Use the provided notebooks to compare a "brute force" solution (e.g., random search) and a reinforcement learning approach. Bear in mind that it takes quite some time to train the system. Observe the implementation of the learning approach: state space, action space, the reward function, Q-table, Q-table update using the temporal-difference learning algorithm, and learning parameters (learning rate, discount factor).

2. Black Jack learning

Blackjack or twenty-one is a card game where the player attempts to beat the dealer, by obtaining a sum of card values that is equal to or less than 21 so that his total is higher than the dealer's. The probabilistic nature of the game makes it an interesting testbed problem for learning algorithms, though the problem of learning a good playing strategy is not obvious. Learning with a teacher systems are not very useful since the target outputs for a given stage of the game are not known. Instead, the learning system has to explore different actions and develop a certain strategy by selectively retaining the actions that maximize the player's performance. We have explored the use of blackjack as a test bed for learning strategies in neural networks, and specifically with reinforcement learning techniques (Perez-Uribe and Sanchez, 1998, https://ieeexplore.ieee.org/abstract/document/687170).

- Complete the provided Python notebook and run experiments of an agent learning to play Black Jack. Plot the resulting strategy and its performance.
- The current implementation allows the agent to learn a strategy taking into account the visible card of the dealer. You may test an even simpler strategy where the agent only considers its own cards.

Report

No report is required for this practical work, however, consider putting into practice what you have learned during the theoretical lesson.