Reinforcement Learning Practical Assignment 2019

Hui Wang

March 2019

1 Introduction

This document contains instructions for the Practical Assignment of Leiden University's Reinforcement Learning Course of 2019. Please read these instructions carefully.

2 Project

Please log into blackboard to download the Assignment.zip. The Assignment contains the skeleton code of some games implementation [1]. You are required to implement 3 game players using Monte Carlo Search, Monte Carlo Tree Search and Q-learning respectively for at least 2 of these games. For this assignment you can choose to either work by yourself or in a team of two. You have to implement and evaluate these Reinforcement Learning methods and report on their performance.

If you have any questions relating to this assignment or the course, please don't hesitate to contact me: h.wang.13@liacs.leidenuniv.nl. You can also use the forum on this course's BlackBoard page.

3 Algorithms

1. Monte Carlo Search.
2. Monte Carlo Tree Search.
3. Q-learning.

These algorithms are quite typical in reinforcement learning, you can find them in our teaching material based on Sutton's book [2]. You are required to understand the their main ideas. If you still do not understand these algorithms very well, please go to google these 3 reinforcement learning algorithms. You can find a lot of descriptions and implementations. You can also ask me during lab sessions, or Dr. Kaifeng Yang about these algorithms.

You are not encouraged to combine Neural Network to finish this assignment. You are required to pay more attention to these 3 reinforcement learning algorithms.

4 General Requirements

1. Implement MCS.py, MCTS.py, Qlearning.py based on corresponding algorithms to play at least 2 kinds of games in the source code.
2. Submit your code implementation and a report.
3. Deadline: 23:59, May 3, 2019. Not accept after May 17. No redo. Penalty: 0.5/week.
4. Do not upload the source code to public internet

5 Evaluation

Now evaluate the behaviour of the different methods (the example method and the methods you implement yourself) , experimenting with different parameter settings. Try, for example, to answer the following questions:

1. Which method converges the fastest? And under which parameter settings?
2. Is this method stable in its convergence w.r.t. different conditions (e.g. different random seeds)?
3. Are other methods maybe slower on average but more stable?
4. How sensitive are the methods to different choices for parameters?(e.g. board size, simulation times...)
5. Which parameter is each method the most sensitive to?
6. . . .

Think about how you can design experiments that help you find answers to those questions. Present the results in such a way that the answers are clear. Make sensible use of figures and tables when appropriate.

In Reinforcement Learning: An Introduction, Sutton & Barto (either edition) [2], you will find many figures that compare different algorithms or parameter settings. Please look through them to get some inspiration for the experiments you could run and present.

When interpreting your results, try to find explanations for your findings. What makes a specific method (in)sensitive to the exact values of your parameters? What makes it converge faster or slower than other methods? For example, parameter c in UCT formular for MCTS.

6 Report

Your report must be written in English and formatted using LATEX, following Springer's LNCS formatting guidelines [1]. It should have a clear structure and about 10 to 15 pages. Please be inspired by this guideline:

Abstract A summary of your work (roughly ten lines). Mention the type of problem you solve, the main methods used, the relevance of your research and your most important findings.

Introduction and Related Work Give an overview of the problem you study and the method you use, and how they can be placed in the field of Reinforcement Learning.

Problem Description Formally define the problem.

Background Introduce any non-trivial terminology, notation, or theorems.

Approach Explain what (novel) techniques you use to solve the problem, and motivate. You can add pseudo code or diagrams to aid your explanation, but beware to not let them stand alone.

Experiments Define the questions you want to answer with your experiments. Explain why your experimental setup is suitable for answering those questions. Specify the hardware you used and relevant implementation and system details. Present your results in tables and/or figures when appropriate. Analyse and discuss your results.

Conclusions Mention and interpret your most important findings, and draw conclusions accordingly. Discuss possible future work.

References Give necessary literature references.

7 Submission requirements

Please upload to BlackBoard the following files:

1. A .pdf file containing your report. If you are working in a team, please name it studentID1 studentID2.pdf (where, of course, you substitute the placeholders by your actual student IDs). Simply name it studentID.pdf if you are working by yourself (substituting the placeholder by your own student ID).

2. A .zip file containing all your code. Make sure to put your best implemented method in MCS.py and my MCTS.py and Qlearning.py. And make sure that we run pit.py, these players can pit against with each other as what I wrote for the sample players. If you use any non-standard libraries, please include them as well. Again, name it studentID1 studentID2.zip if you are working in a team and simply studentID.zip if you did the assignment by yourself.

   Note that the BlackBoard assignment allows for multiple submissions. We will grade the last submission made before the deadline.

[1] https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines

Important!

Note that we will be checking both your written report and your code for plagiarism. You are not allowed to copy work from fellow students or anyone else. If you are basing your algorithm on somebody else's work, be very clear about that and reference them properly. If we do suspect plagiarism or other forms of cheating, measures will be taken.

However: for your experiments you are allowed to test your method against the methods of your fellow students or others. Naturally, you should credit and reference them accordingly.

References

https://github.com/suragnair/alpha-zero-general

Richard S. Sutton and Andrew G. Barto. Reinforcement learning : an introduction. MIT Press, 1998, p. 322.