TD3: MCTS

Exercise 1:

Question 1 • Why is it necessary to balance exploration and exploitation in MCTS?

Question 2 • What is the main difference between the MCTS algorithm and the Minimax algorithm in zero-sum games?

Question 3 • What happens if we apply excessive exploration in MCTS by increasing the parameter c too much?

Question 4 • In MCTS, what does "backpropagation" mean?

Question 5 • How can we improve MCTS performance in an environment with a high branching factor?

 ${\bf Question}~{\bf 6}~$ \bullet What happens if the number of MCTS iterations is smaller than the branching factor?

Exercise 2:

In this exercise, we will revisit the example from the previous sheet and apply two different algorithms to analyze choices and decisions in a game tree. The objective is to compare the results obtained with the Minimax algorithm and the MCTS (Monte Carlo Tree Search) algorithm.



Question 1 • Apply the Minimax algorithm to determine the optimal value of the root and the action to choose between Left and Right.

Question 2 • Now apply the MCTS algorithm with (N = 5, c = 2) and compare the values at the nodes and the final decision. What do you observe?

Question 3 • To reduce computation time, we want to limit the search depth in MiniMax to 3. What modifications would be necessary to implement this restriction? What are the advantages and disadvantages of this approach compared to using MCTS with an equivalent number of calculations (around 5 to 7 iterations)?

Question 4 • Re-examine the result of question 2 by performing 5 additional iterations. Analyze the new result and compare it with the one from question 2 in terms of the final decision made by MCTS.