DRL – Homework 1

Group 2

Task 1

Chess as MDP

Since formalizing chess as MDP is far from trivial we will simply list all elements of the formalization with an example and then elaborate on them briefly.

The set of all states S consists of all legal combinations of structural[1] and rule elements[2] necessary to describe a board of chess at a particular timestep. There also exists a subset of terminal states which include boards that are won, lost, or drawn due to the pieces or the rules of drawing. S is large, but finite.

The set of all actions A consists of all moves in chess, including claiming a draw, which are legal in at least one state $s \in S$. One largely compressed embedding for A contains fewer than 3.000 entries and is achieved by encoding all mappings of field to field which are legally playable by at least one piece plus the pawn promotion options[3].

The transitional probability function P for chess poses a problem since chess is a multi-agent setting. If we assume our opponent is a part of the environment, we cannot formulate P, if we assume self-play or fix this issue by some other assumption or a known policy of the opponent, we can formulate P. P maps, in chess, a board and move to a reward and successor board.

The reward function R for chess calculates the expected reward for an action and state pair based on how state values are defined. A terminal board may have a fixed reward, an intermediate board may have a state value function which is based on empirical approximation on how good that board is.

Policy

In a probabilistic policy for chess every state $s \in S$ has a probability distribution over all actions $a \in A$ with probabilities of illegal moves as zero. An optimal policy, assuming optimal counter-play, will try to maximize the expected reward, meaning chance of not losing, preferably winning, for the current state by selecting the appropriate action.

[1] A board representation with all the pieces for every square (see FEN notation).

[2] The castling rights, en-passant square, and player to move (see FEN notation), and the repetition count and moves without progress.

[3] Promotion a Pawn to either Queen, Bishop, Knight, or Rook.

Task 2:

MDP:

The state space S contains the coordinates of the lander, the velocity of the lander, a binarized image of the ground pixels, and the distance to the landing area. Additionally, landing, crashing, and ground-contact with the legs should also be included as a Boolean value. The subset of starting states includes fixed speed and coordinates for the lander, a fixed landing zone, and randomized ground structure.

The subsets of terminal states are states in which the lander has crashed or landed. The action space A includes the four actions "do nothing, fire left orientation engine, fire main engine, fire right orientation engine"[4].

The transitional probability function P contains the mapping of one state and action to a specific reward and successor state, here this means we map a state with the landing in coordinates (20,20) for example to a successor state with different coordinates, different velocity, etc.

The reward function R is a function which calculates the expected reward of a state s and an action a which here includes the cost of traveling away from the landing area, and the reward for the terminal states of landing, including landing quality, and crashing[4].

Policy:

The deterministic policy of the Lunar Lander has, for every state $s \in S$, a specific action $a \in A$ which is selected.
This means, in an optimal policy, in a state in which the lander is too fast it will select the engine that slows it down the most and if it is misdirected it will try to correct its path towards the landing area.

[4] https://gym.openai.com/envs/LunarLander-v2/

Task 3:

1) **Explain what the environment dynamics (i.e., reward function and state transition function) are and give at least two examples.**

The reward of a state s and an action a is the expected future reward $\mathbb{E}[R_{t+1}]$ of that state action pair.

The state transition function describes the probabilities of reaching a state s', with a reward r given a state s and a selected action a in said state.

Examples

In chess the expected future reward could be the expected likelihood of winning. The transitional probabilities are based on expected play by the opponent and the selected own action leading to another state.

For autonomous robotic vacuum cleaners, the reward could, for example, be cleaning an area, not visiting an already cleaned area or not hitting furniture. The expected future reward is thereby based on expected intermediate state rewards until completing the task.

2) **Discuss: Are the environment dynamics generally known and can practically be used to solve a problem with RL?**

Deterministic and Probabilistic transition dynamics

In case we have a full-model of our problem which includes the ground-truth transitional dynamics, we can have full knowledge of the MDP. The probability distribution of potential successor states s_{t+1} for an action a_t in state s_t are all known.

Example: The action of flipping a coin leads in an ideal scenario should be $\approx 50\%$ heads, $\approx 50\%$ tails and a small chance of landing on the side.

Multi-Agent setting

In chess we do not have the opponent's policy to rely on when predicting which move the opponent might play. We have the deterministic transition probability for our own action, but not the opponent's action afterwards. When we play a move a_t in state s_t we do know the state s_{t+1}, however, since it is our opponent's turn afterwards, we do not know which move is selected as a_{t+1}, leading to a state s_{t+2} which is unknown to us; we cannot predict it fully.

This problem may be mitigated in practice by assuming optimal play of our opponent, based on our own approximation of the optimal policy.

Qualification and Ramification Problem

We do not know the environment dynamics outside of our set of states since we do not have a perfect model of the entire world plus laws of physics. Ergo there may be outside influence which prevents us from reaching the successor state s_{t+1} from our current state. Example the self-driving car gets destroyed alongside a large part of the earth by a meteor.

Also, we do not know if our action causes an effect that will, later, prevent us from reaching a certain state which our model predicts we can reach.

This applies mainly to real world problems and means that we can never have full knowledge of environment dynamics but can practically make a model which is a good enough approximation.