Honours Math for Machine Learning HW5

Saina Koukpari - McGill University

Problem 1.1

The state is the battery level and the actions are 1) actively searching for a can, 2) remaining stationary and waiting for someone to bring a can, and 3) heading back to home base to recharge the battery. The reward is increased when the robot receives and empty can, and decreased when the robot runs out of battery. Thus the reward depends on the state of the battery and not the robots environment where the probability of doing an action is based on battery level, so the reward is stochastic.

Problem 1.2

TD-Gammon: The state space is the sequence of board positions and the action space is the set of possible ways it could play its dice roll and the corresponding positions that would result. The reward function is defined as zero except for a time where a game is won where it would then be 1 and so the rewards are deterministic. Given an action the next state is stochastic since it depends on the dice roll and so the policies used are stochastic.

Atari: The state space is the 'stacked' four most recent frames of the game and the action space is next possible state, i.e. video frame. The reward function indicates the games score as a function of time steps where it outputs 1 for an increased score, -1 for a decreased score and 0 otherwise. The rewards are deterministic since it depends on the previous states, i.e. the memory of experiences over playing the same game. Given an action the next state is also deterministic and the policies used are deterministic.

Alpha Go: The state space is the sequence of board positions and the action space is the set of go moves, i.e. where to place a stone or to resign. The reward function is 1 if the move wins, -1 if it loses and 0 otherwise where the rewards are stochastic since it depends on the movement made. Given an action the next state is stochastic and so the policies used are stochastic.

Problem 1.3

The optimal strategy for rock, paper, scissors is stochastic. The probabilities for optimal policy π are given by $(R, P, S) = (1/3, 1/3, 1/3)$, and the probabilities for a player who tends to play rock twice as often is given by $(R, P, S) = (1/2, 1/4, 1/4)$.

Problem 1.4

Treating the opposing player as optimal means that each player will optimally respond to the current game state, i.e. opponents move. Treating the opposing player as statistically modelled then implies that we can consider the player as a part of the environment since they are not changing their behavior based on any learning.

Problem 1.5

The other players were represented statistical since we can exploit the non optimal moves. Self-play would lead to exploration of state space regions that aren't typical for real human players since the players know little about the confidence of their opponents moves.

Problem 2.1

There are 198 inputs and 40-80 hidden units in the neural network. The output represents the predicted probability of winning. Given that the output of a hidden unit j is,

$$h(j) = \sigma\left(\sum_i w_{ij} x_i\right)$$

we can define the formula for the final output as,

$$\hat{v}(S_t, w) = \sigma\left(\sum_j \sum_i w_{ij} x_i\right)$$