Homework-1 Report

Furkan Arslan

S00371

In this homework, I have used dynamic programming for learning to play checker. For this purpose, I have developed a class named ‘Value_Iteration_AI’. This class is a child of ‘player’ class.

This class takes an opponent model, player id, discount factor and board as arguments. After the initialization, it calls value iteration function. This function is the main element of this project. That functions implement value iteration algorithm to find an optimal policy for the checker.

The value iteration function runs over all states. For this purpose, it firstly starts from the initial state then finds possible states that we can go from the current state. The founded next states are put in a state array. After the current state, the founded states will be processed one by one through all possible states.

Step by step explanation for value iteration. The value iteration loops until it converges. In this infinite loop, there is another loop that runs over all states. It means that value iteration algorithm keeps updating the value of states until finding optimal values. Inside the state loop, I calculated possible values of the state and take their maximum. Then update the state’s value as the founded maximum value.

The logic of calculation of possible state values is that loop over all possible actions that apply in this state and calculate their values. For this calculation, I used the formula of value iteration which combines the value of the next state, reward of action and probability of taking this action. For iteration in possible actions, firstly the action is made; secondly, the opponent move is made then the next state is obtained and also reward is calculated with the information of the next state. After that, calculation of the formula is made.

The reward is one of a key part of reinforcement learning. So in order to calculate the reward, I awarded some actions and also I defined some punishments. If the model is winning state, it gains 100 points. On the other hand, if the model is losing state, it punished with -100 points. The value of draw state is 0. Beside results of the game, the model can gain reward by eating opponent's pieces. The model can gain 5 points for each piece, 10 points for each king. Again the model can lose points if opponent eats model’s pieces. The model loses 5 points for each piece lose and 10 points for each piece king.

After the value iteration is convergent, the policy will be founded. For the creating policy array, the function loops over all states again. Then calculates the values of the possible actions of the state with same way as described above. After that, the index of the action which has maximum value will be 1 in policy array other actions will be 0.

To sum up briefly, the function passes over all states until it converges. During this process, it calculates the value of the state with the formulation of value iteration. In order to calculate the value of the state, it processes over all possible actions which can be taken in the state. During this process, it finds next state after that action happens, the reward of that action, the probability of that action then combines them together to calculate the value of the action. The value of the state will be the maximum value of possible calculated actions. After the value iteration process, the policy array will be created.

Note-1: In my test, my implementation of value iteration has never convergence. It is created too many states so this process of convergence can take hours, maybe days.