Machine Learning: Artificial Intelligence Decision Making with Minimax

Minimax algorithm state value

When writing the minimax algorithm, each game involves two players and game states can be evaluated as a value. One of the players is called the *maximizer*, because he or she wants to maximize the value of the game and the remaining player is called the *minimizer*.

Minimax algorithm problem specification

Given a game state, the minimax algorithm finds the decision that maximizes the minimum gain. In other words, if you assume your opponent will make decisions that minimize your gain, the algorithm finds the move that will maximize it based on the options your opponent gives you. It is assumed that the game is being played by turns and that the opponent is playing optimally, this is: at each turn a player must make a move, and this move is the best the player can make in that situation.

Minimax algorithm assumption

When running the minimax algorithm, it is assumed that your opponent is playing optimally. This assumption is a worst case scenario, given that if your opponent is not playing optimally the problem is reduced to a simpler one.

codecademy

Minimax algorithm game representation

When writing the minimax algorithm, a game is modeled as a tree. Different elements of the game (as the current state and all possible moves) are represented as different parts of the tree. This visual representation of the game is a great aid in order to implement the minimax algorithm.

Minimax algorithm state evaluation

When running the minimax algorithm, a game state can be evaluated even if it is not a leaf. This game state evaluation is particularly important in some games such as chess, where we have a long sequence of states before reaching a leaf.

Minimax algorithm size restriction

The size of the game tree is a very important restriction in a minimax algorithm, given that it is not possible to visit all states in a reasonable time. If the maximum depth you can consider is reduced, the optimality of your solution is affected. When the size of the game tree is very large, several heuristics can be applied to find a good solution of the minimax algorithm.

Minimax algorithm with alpha-beta pruning

When implementing alpha-beta pruning in the minimax algorithm, its execution time is drastically decreased. For a given unit of time, a minimax algorithm with alpha-beta pruning can go down twice as far as a minimax algorithm without this pruning technique.

codecademy

Alpha-beta pruning variables

When using alpha-beta pruning in a minimax algorithm, it is needed to track the value of two different variables (alpha and beta) in order to decide when to prune a part of the tree. At the beginning of the game, alpha is equal to negative infinity and beta is equal to positive infinity. These values are updated as the game progresses.

Print Share