

DESENVOLVIMENTO DE BOT DO JOGO DA VELHA POR MEIO DO ALGORITMO MINIMAX

O algoritmo minimax é um método para achar o melhor caminho possível para determinado objetivo dentro de uma busca em profundidade em uma árvore de decisões, basicamente é uma heurística que visa indicar o melhor caminho para chegar no objetivo. Ele visa maximizar o ganho mínimo e minimizar a perda máxima de um determinado jogador, por isso o nome minimax.

Uma de suas principais aplicações está no tic-tac-toe, ou jogo da velha em português, onde é possível construir uma árvore contendo todas as possíveis jogadas a partir de um estado inicial. Basicamente, cada profundidade da árvore corresponde a um turno de determinado jogador, se o turno for do bot definimos essa profundidade como maximizadora, caso contrário será minimizadora. Após isso, devemos atribuir pesos para cada um dos nós terminais do jogo (nós onde o jogo finaliza: vitória, derrota ou empate), se o nó estiver em profundidade maximizadora a vitória do bot terá peso 1, empate 0 e derrota -1, em caso de profundidade minimizadora teremos o contrário. Em sequência, uma busca em profundidade é aplicada sobre a árvore e a cada nó percorrido acontecem duas situações: -Nó está em profundidade maximizadora, todos os seus filhos são verificados e o seu peso será igual ao que tiver maior peso.

-Nó está em profundidade minimizadora, todos os seus filhos são verificados e o seu peso será igual ao que tiver menor peso.

Após todos os nós da árvore serem percorridos, o algoritmo minimax finaliza com a árvore totalmente distribuída de acordo com os pesos respectivos, de acordo com a figura 1.

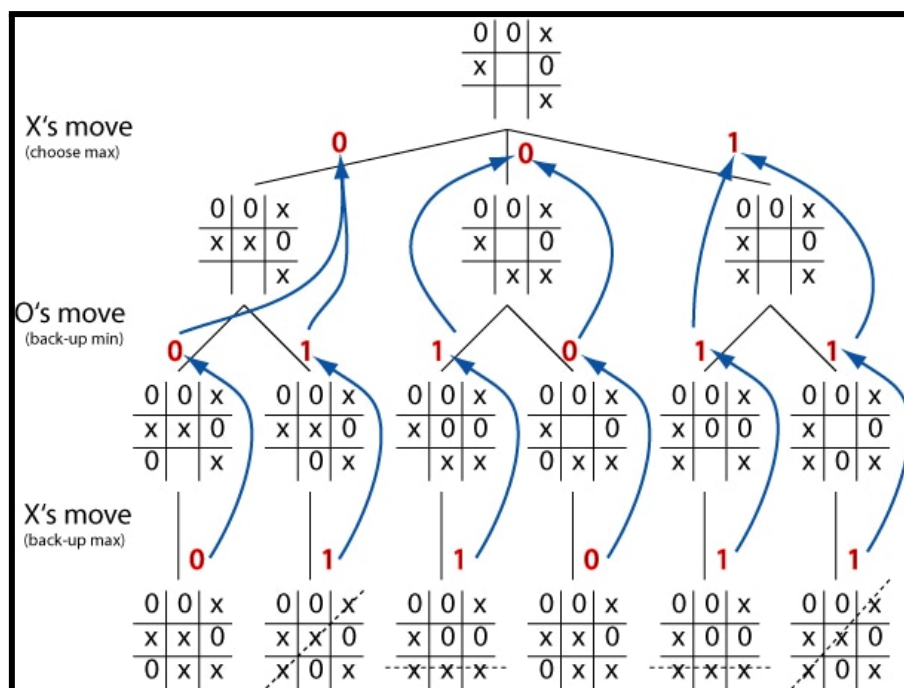


Figura 1 - Exemplo de minimax na árvore de decisões

Com a árvore totalmente construída e com pesos distribuídos fica simples decidir a melhor jogada para o bot, basta apenas escolher sempre a que tiver o maior peso dentre as possíveis jogadas, assim retornando a melhor jogada possível daquele momento.

Na implementação do projeto apresentado, a árvore de decisão é gerada apenas após a primeira jogada, visto que se fosse gerada desde o início ficaria muito grande e geraria uma lentidão no código pois são centenas de milhares de estados possíveis de jogo. Construindo a árvore após o primeiro movimento, ela reduz seu tamanho consideravelmente, resultando em um desempenho muito superior. Caso a primeira jogada seja a do bot, as decisões ainda não foram construídas, portanto é setado que nesse caso a sua jogada será a posição do meio, por ser considerada uma boa jogada inicial (por mais que na primeira jogada todas posições tenham praticamente a mesma probabilidade de derrota).

A decisão da melhor jogada ocorre sempre que for o turno do bot, onde o maior peso será selecionado e a posição da jogada será retornada.