TablutAI

Team: Supernova

Safoura Banihashemi,

Lyudmil Stamenov,

Hesam Sheikh Hassani

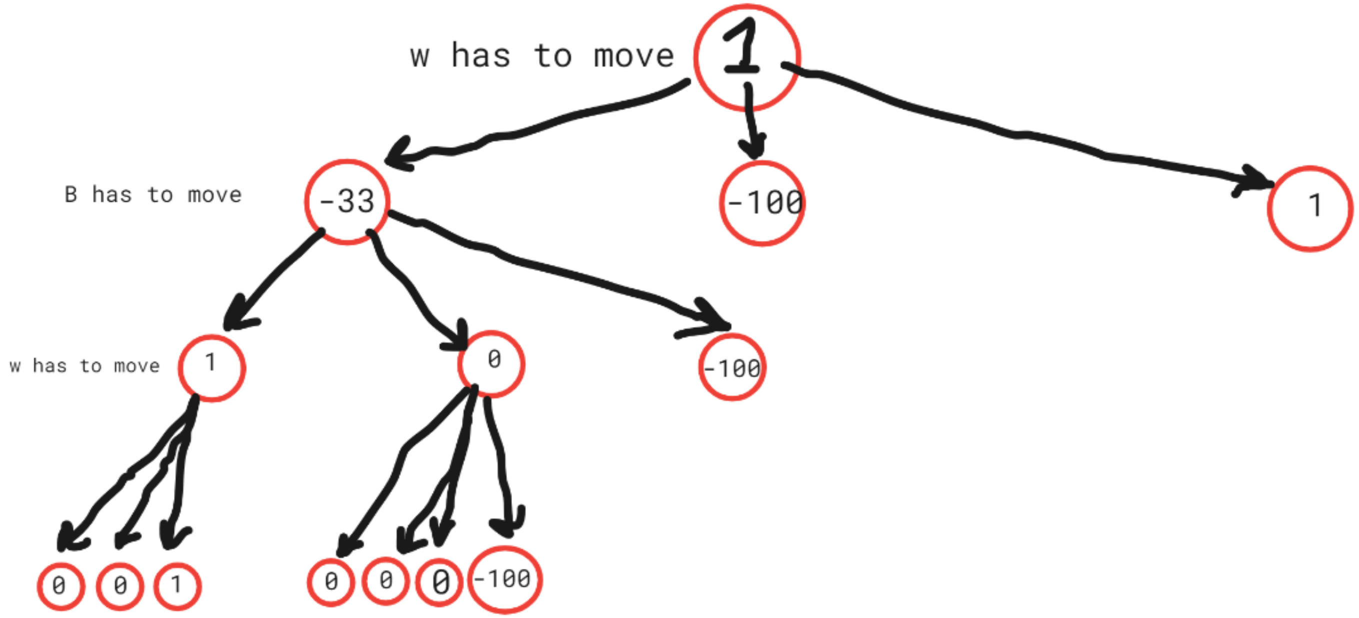
Our Strategy

- What is our strategy and why did we choose it?

- Possible alternatives :
  - Genetic Algorithms
  - A* Search
  - MinMax & Alpha-Beta

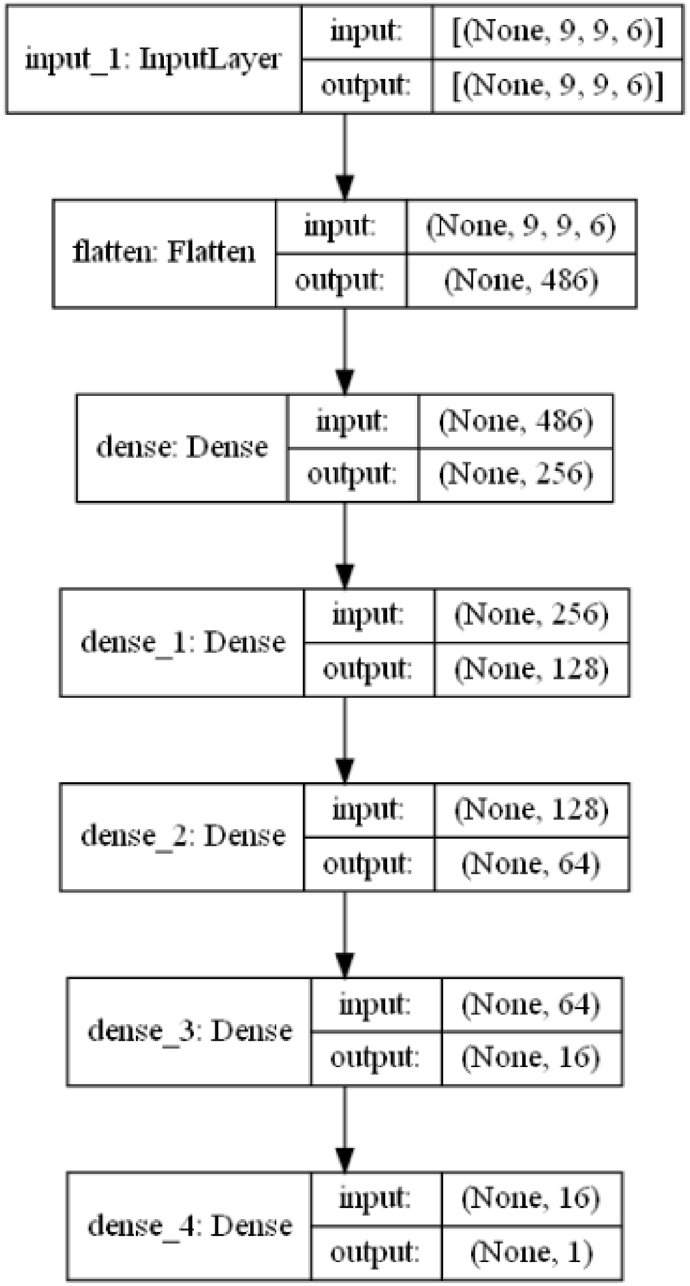
Mean-Max Tree

- Drawbacks of traditional Min-Max algorithm and alpha beta pruning in this problem
- Our approach



Neural-Net

- The neural net assigns a score to a game state in range [-1,1]. If a game state is more favorable to white, the score will be positive, if it's more favourable to black, the score is negative.



Encoding of states

Each game state is encoded into **six** 9x9 matrices, representing:

1. White pieces
2. Black pieces
3. King
4. Castle
5. Escape tiles
6. Camp tiles

Optimizations

- Why optimization ?
  - A significant challenge of the competition is the **60 seconds limit** to choose a move.
  - The average time for the tree to select a state (playing as White):
    - No optimization: 64 seconds (likely exceeding the 60 seconds limit)
    - With tree optimizations: 2.8 seconds

Optimizations

- **Siblings Birth Control:** If a winning state is encountered while exploring children nodes, further exploration halts, reducing computational load.
- **Limit Last Moves:** If white is going to win in 2 moves, the last move Must be done by the King. Similarly, if black is going to win, its last move Must by targeted to a square close to the king to capture it. This way we limit the number of possible moves to check in the tree.
- **Prioritizing Moves:** Moves crucial to securing wins are prioritized, potentially achieving winning within the initial depth.
- **TFLite:** The NeuralNet is optimized further using a tflite mode, significantly enhancing its speed.

Thank you for your Attention!