Temporal Difference Learning in the Game of Havannah

Jeffrey Burkert

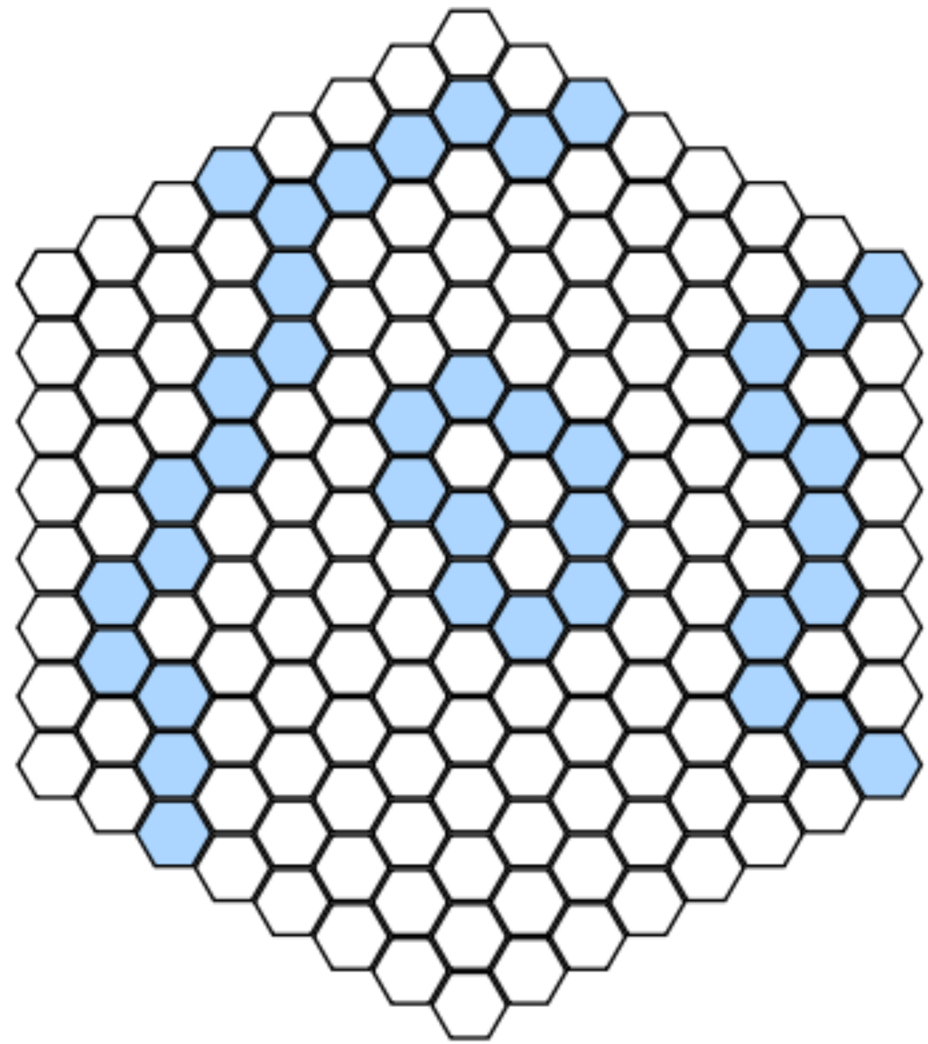
Brief Intro to TD-Learning

- Supervised learning when current value is unknown.
- Uses a neural network evaluation of optimal future state to learn value of current state.
- Used to great success in creating TD-Gammon and other world class backgammon bots including GNU Backgammon, Snowie, and Jellyfish.

Havannah

- Players take turns marking hexes on a Hexagonal grid.

- Three win conditions
  - Connect two corners
  - Connect three sides
  - Surround one hex



Why Havannah?

- Invented by Christian Freeling in the early 90's
- Challenge
    - $1000 to bot that can beat him by 2012
    - No successful takers
- Board is scalable
- Strongest available bots uses Monte Carlo Tree Search

Why Havannah?

ID	Game	State-space	Game-tree
1	Awari	10^{12}	10^{32}
2	Checkers	10^{21}	10^{31}
3	Chess	10^{46}	10^{123}
4	Chinese Chess	10^{48}	10^{150}
5	Connect-Four	10^{14}	10^{21}
6	Dakon-6	10^{15}	10^{33}
7	Domineering (8×8)	10^{15}	10^{27}
8	Draughts	10^{30}	10^{54}
9	Go (19×19)	10^{172}	10^{360}
10	Go-Moku (15×15)	10^{105}	10^{70}
11	**Havannah (19×19)**	$\mathbf{10^{127}}$	$\mathbf{10^{157}}$
12	Hex (11×11)	10^{57}	10^{98}
13	Kalah(6,4)	10^{13}	10^{18}
14	Nine Men's Morris	10^{10}	10^{50}
15	Othello	10^{28}	10^{58}

Network of TD-Havannah

- Inputs
  - One input for each hex, 1 for player 1, -1 for player 2
  - Input set to 1 if player 1 is about to play, -1 otherwise
- One hidden layer with 150 neurons
- Output evaluates the strength of player 1's position

Training the Network

- Enumerate all moves and get network evaluation
- Pick best and backpropagate through the network, possibly multiplying by some factor.
- Naive Approach
    - Train through self play
    - Pick best move
    - Leads to improper state space exploration
- Networks trained this way perform very poorly.

State Space Exploration

- Difficult balance needs to bee struck.

- Initial solution: random moves
  - Tried letting player select random legal move 20% of time
  - Training through 100,000 games on a 2x2 grid could not solve the game

- Experiment:
  - Train selecting random move 20% but make player 2 a fully random player.
  - Solved 2x2 case in 10,000 games.

State Space Exploration

- Clearly we need to ensure fuller state space exploration

- Solution:
  - Early in training, move essentially randomly
  - As networks improve, move randomly less often

- This is better
  - Can achieve a 98% win percentage against random player on 3x3 board.
  - Often falls into a "steady state" and learning slows

State Space Exploration

- Final solution, base exploration on move strength
- Able to achieve ~100% win rate against random bot
- Still very weak by human standards
  - Loses consistently to me!

What it learned

- Learned that the corner win condition is optimal
- Placed stones in close proximity to each other
- Occasionally is able to block an instant win from the other player

Failures

- Pursues win conditions that are blocked
- Often fails to block instant wins
- No sense of strategy on a more than local scale
- Can't force a win on a 3x3 board

DEMO!

Future Work

- Constrain network based on symmetry of hexagon

- Experiment with network architetures
  - Hidden layers
  - Feature Maps

- Modified training
  - Different randomization procedure