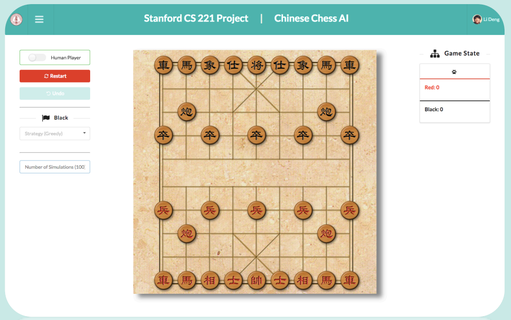
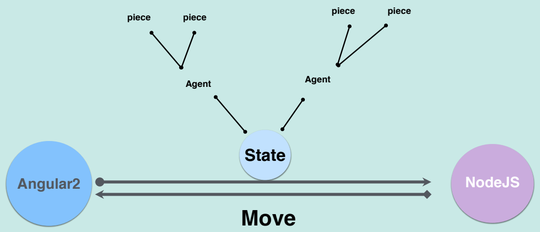
AI Agent for Chinese Chess

Simulator



- Human Mode + Simulations Mode
- Live Analysis
- *Frontend: Angular2 + SemanticUI*
- *Backend: Node.js*
- *OOP Design in Typescript*



Motivation

- Popularity in China
- Scarcity of AI-powered Chinese Chess Engine

Challenge

- Complex Game Rules
- Large branching factor (38)

Evaluation Performance

Materials Value
- Piece Value
- Piece Position

Attacking Power
- Number of Threatening
- Number of Captures
- Number of Center Cannons
- Number of Aligned Cannons

Mobility
- Mobility of Rook
- Mobility of Cannon
- Mobility of Horse
- Mobility of Elephant

TD Learning

$$w^{t+1} := w^t + \eta \cdot r^t \cdot (A \cdot \overline{\phi^t} + B \cdot \phi^{t*} - w^t)$$

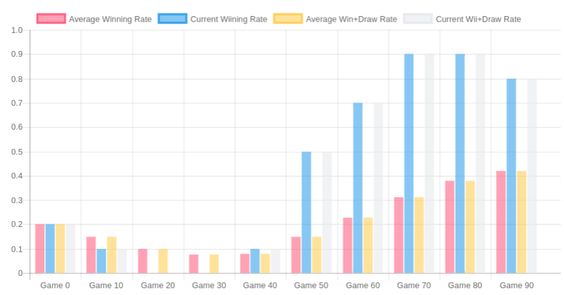
w^t : weigh at game t

$\overline{\phi^t}$: average feature vector across all states in game t

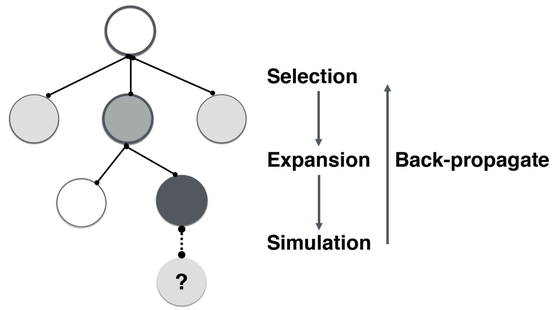
ϕ^{t*} : last (win/lose)feature vector in game t

r^t : reward in game t $\quad r = \begin{cases} Win : 1 \\ Lose : -1 \\ Draw : 0 \end{cases}$

Learning Curve



Search Efficiency



Search Effenciency Comparison

Strategy	Search Depth	Average Runtime for Each Move(ms)
Alpha-Beta Pruning	2	76
Greedy	1	3
Alpha-Beta Pruning	3	600
Alpha-Beta Pruning	4	7307
Alpha-Beta Pruning with Move Reorder (Type A)	2	72
Alpha-Beta Pruning with Move Reorder (Type A)	3	239
Alpha-Beta Pruning with Move Reorder (Type A)	4	3175
Temporal Difference Learning	2	393
Temporal Difference Learning	3	1176
Temporal Difference Learning	4	9568
Monte Carlo Tree Search	2	43
Monte Carlo Tree Search	3	105
Monte Carlo Tree Search	4	315

Strategy

- *Greedy*
- *MiniMax*
- *Alpha-beta Pruning*
- *Pruning with Move Reorder*
- *Monte Carlo Tree Search*
- *Temporal Difference Learning*

Github

Li Deng
dengl11@stanford.edu
Stanford CS221 2016 Autumn