AlphaGo, etc.
Lab 4

- Due Feb. 29 (you have two weeks ... 1.5 remaining)

- new game0.py with show_values for debugging
Exam on Tuesday in lab

- I sent out a topics list last night.
- On Monday in lecture, we’ll be doing review problems, plus Q&A.
  - We’ll also do Q&A at the end today if there’s time.
  - I plan to send out review problems over the weekend.

What sorts of questions will be on the exam?

- selecting an appropriate algorithm for various problems
  - state space search vs. local search; BFS vs. A*; minimax vs. MCTS...
- setting up an appropriate model for the problem and algorithm
  - generating neighbors; identifying a goal; describing utilities; choosing a heuristic...
- stepping through algorithms
  - identify the next state; list the order nodes are expanded; eliminate dominated strategies...
AlphaGo

Selection

Expansion

Evaluation

Backup

neural networks

normal MCTS
AlphaGo neural networks

- Rollout policy: \( p_\pi \)
- SL policy network: \( p_\sigma \)
- RL policy network: \( p_\rho \)
- Value network: \( v_\theta \)

- Evaluation
- Selection
- Evaluation
Step 1: learn to predict human moves

- used a large database of online expert games
- learned two versions of the neural network
  - a fast network $P_{\pi}$ for use in evaluation
  - an accurate network $P_{\sigma}$ for use in selection
Step 2: improve the accurate network

- run large numbers of self-play games

- update the network using reinforcement learning
  - weights updated by stochastic gradient ascent

\[ p_\sigma \xrightarrow{\text{Policy Gradient}} p_\rho \]
Step 3: learn a board evaluation network, $V_\theta$

- use random samples from the self-play database
- prediction target: probability that black wins from a given board
AlphaGo tree policy

select nodes randomly according to weight: $\text{value} + \frac{\text{prior}}{1 + \text{visits}}$

prior is determined by the improved policy network $P_p$
AlphaGo default policy

When expanding a node, its initial value combines:

- an evaluation from value network $V_\theta$
- a rollout using fast policy $P_\pi$

$$v(s) = \lambda \ast \text{outcome} + (1 - \lambda) \ast \text{eval}$$

A rollout according to $P_\pi$ selects random moves with the estimated probability a human would select them instead of uniformly randomly.
AlphaGo results

- Beat a low-rank professional player (Fan Hui) 5 games to 0.
- Will take on a top professional player (Lee Sedol) March 8-15 in Seoul.
- There are good reasons to think AlphaGo may lose:
  - AlphaGo’s estimated ELO rating is lower than Lee’s.
  - Professionals who analyzed AlphaGo’s moves don’t think it can win.
  - Deep Blue lost to Kasparov on its first attempt after beating lower-ranked grandmasters.
Transforming normal to extensive form

Key idea: represent simultaneous moves with information sets.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5,5</td>
<td>2,8</td>
</tr>
<tr>
<td>B</td>
<td>1,3</td>
<td>3,0</td>
</tr>
</tbody>
</table>

Diagram: Extensive Form Representation
Transforming extensive to normal form

Key idea: strategies are complete policies, specifying an action for every information set.
**Design Dimensions**
- modularity
- representation scheme
- discreteness
- planning horizon
- uncertainty
- dynamic environment
- number of agents
- learning
- computational limitations

**State Space Search**
- state space modeling
- completeness
- optimality
- time/space complexity

Uninformed Search
- depth-first
- breadth-first
- uniform cost

Informed Search
- greedy
- A*
- heuristics, admissibility

**Local Search**
- state spaces
- cost functions
- neighbor generation

**Hill-Climbing**
- random restarts
- random moves
- simulated annealing
- temperature, decay rate

**Population Search**
- (stochastic) beam search
- gibbs sampling
- genetic algorithms
- select/crossover/mutate
- state representation
- satisfiability
- gradient ascent

**Game Theory**

**Utility**
- preferences
- expected utility maximizing

**Extensive-Form Games**
- game tree representation
- backwards induction
- minimax
- alpha-beta pruning
- heuristic evaluation

**Normal Form Games**
- payoff matrix repr.
- removing dominated strats
- pure-strategy Nash eq.
- find one
- mixed strategy Nash eq.
- verify one
- matrix/tree equivalence

**Monte Carlo Search**
- random sampling evaluation
- explore/exploit tradeoff

Monte Carlo Tree Search
- tree policy
- default policy
- UCT/UCB