AlphaGo and AlphaGo Zero

Richard Shen

February 22, 2018
Primer: Go

Basic Rules

- Two-player, deterministic turn-based game (Black and White)
- Goal: place tiles to capture more territory and opponent pieces
- Liberties and chains
- Passing and retiring

Credit: Wikipedia
Primer: Go

Basic Rules
- Two-player, deterministic turn-based game (Black and White)
- Goal: place tiles to capture more territory and opponent pieces
- Liberties and chains
- Passing and retiring

Properties
- Perfect information
- Perfect simulator
- Discrete state and action spaces
- Translational invariance
- Lack of codified board values

Credit: Wikipedia
Primer: Game Trees

- Game tree: tree representation of legal game states and moves
- (Theoretically) possible to compute exact optimal play
  - Minimax algorithm
- Game tree complexity $\approx b^d$
  - Go: $b \approx 250$, $d \approx 150$
  - Completely infeasible

Credit: M. Law (2013)


Monte Carlo Tree Search (MCTS)

- General algorithm for efficiently searching for optimal game tree paths from a given root game state
- 4 stages: leaf selection, child expansion, MC rollout simulation, update counts
- Before AlphaGo, the general approach most successfully applied to solving Go

Credit: Wikipedia
AlphaGo
Policy and Value Networks

- **Separate** 12-layer CNNs with ReLU activations

Credit: Silver (IJCAI 2017)
Policy Network and MCTS Search Breadth

- Approximate leaf values in MCTS using rollouts specified by policy network instead of MC random rollouts
- Reduce the search breadth in MCTS

Credit: Silver (IJCAI 2017)
Value Network and MCTS Search Depth

- Approximate leaf values in MCTS using a value network instead of MC rollouts
- Reduce the search depth in MCTS

Credit: Silver (IJCAI 2017)
**SL-RL Training Pipeline**

- Fast policy network ($p_\pi$) and strong policy network ($p_\sigma$) initially trained to predict expert moves
- $p_\sigma$ later trained through games of self-play to maximize probability of winning ($p_\rho$)
- Value network ($v_\theta$) trained from self-play board positions to predict win probability
MCTS in AlphaGo

- **Selection:** Q-value and prior weight proportional to prior policy network evaluation
- **Expansion:** Expand tree to children of selected node, assign policy network prior
- **Evaluation:** MC rollouts using fast policy network and value network prediction
- **Backup:** Update Q-values and number of visits up to root based on evaluations
Results

- AlphaGo Fan defeated European Go champion Fan Hui 5-0
- AlphaGo Lee defeated world Go champion Lee Sedol 4-1
- AlphaGo Master defeated top Go players worldwide 60-0
AlphaGo Zero
Differences

- Learning to play Go from first principles
- Trained only via self-play
- Joint policy and value network
- Raw board state as input
- ResNet architecture
- MCTS evaluation via value function evaluation (no MC rollouts)
MCTS in AlphaGo Zero

- Self-play via MCTS
- MCTS as search-based policy iteration
Results

- Passes AlphaGo Lee after just 3 days of self-play
- Passes AlphaGo Master after about 21 days
- Strongest Go player in the world after 40 days of self-play starting only from first principles
Extensions

- AlphaZero: “Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm” (Silver et al. 2017c)
- Limited implications beyond perfect information games
- AlphaGo Documentary
  https://www.youtube.com/watch?v=IsAnip3OQ3U
- MiniGo: https://github.com/tensorflow/minigo
References


