Section 1

AlphaGo trained by combo of:

1. Human Expert Games (Supervised Learning)
2. Self Play (Reinforcement Learning)

Overview:

- Legal sequences of moves: b^d where:
  - b: legal number of moves per position
  - d: game length (number of moves)
- For chess, b ≈ 35 & d ≈ 80; For Go, b ≈ 250 & d ≈ 150
- Optimal value function for a game of chess or go is therefore approximated to $v(s) \approx v^*(s)$ from the state of the game s

(Two) General Principles for search space reduction:

- Take approximate optimal value function $v(s) \approx v^*(s)$ from the state of the game s
  - However, above is still difficult for go, due to its high complexity(compared to chess, checkers, etc.)
- Sampling actions from probability distribution $p(a|s)$ for possible moves a given position s

Previous Go Models:

- Monte Carlo Rollouts Search randomizes moves and plays games out, and then uses results for weighting nodes; averaging over them produces amateur-level Go play
- Running a very large number of simulations leads to asymptotic convergence to high-level Go play
- Even enhanced with predictions of expert moves, which narrows the decision process to high probability moves, only strong amateur level play is achieved due to aforementioned constraints

Monte-Carlo Decision Trees:

- Traversal: Traverse from root node to a leaf node. The following formula balances finding child nodes with high win rates, with finding potentially less explored moves
- Upper Bound Confidence Trees(UCT) Formula:

- $UCB_1(s) = v_i + c\sqrt{\frac{\ln(N)}{n_i}}$
- v_i : wins over simulations, c : exploration parameter, N : Number of Simulations for the Parent Node, n_i : number of simulations for child i

- Node Expansion: Upon reaching a leaf node, generate at least one child node by playing a move(If the game would not yet have concluded)
- Rollout: Simulate a game by choosing, either random moves, or moves from some determined heuristic until the end of a game, and record the result. This is the stage where a policy network to determine the moves may be useful
- Backpropagation: Update the statistics of the nodes from the leaf to the root based on the result. This includes the number of visits to the node, and the win count

AlphaGo's approach:

- Develop Deep Convolutional Neural Network(CNN) approach, such as those employed in Atari Game AIs/Facial Recognition Software
- This network is used to reduce depth and breadth of the search tree
  - Value network is used to evaluate positions
  - Policy network is used for sampling action

Training Policy Network:

- SL policy network p_σ is trained using expert human moves
- Fast learning policy network p_π that samples actions during rollouts
- RL policy network p_ρ corrects SL policy network to redirect it toward the right goal, which should be to win games instead of getting better at predicting moves