Alpha Go Summary

Goals and Techniques

The goal for this project was to invent alternative ways to evaluate and choose game playing moves artificially. Up until this project, conventional and even more modern game playing agents rely on searching game states recursively while evaluating potential future moves by all players. This approach would use techniques to compare 'best' moves by the players of the game and even forecasting the game until winner prediction (full search exhaustion). This technique proved very successful in games such as checkers and more primitive games, however in more complex games such Go the search space was just too huge to use the same searching techniques. This project was aimed to overcome the complexity ceiling of older techniques and prove the case by advancing the AI agent for Go.

Methods Used

Since the existing searching methods used would be way too costly (or even maybe infeasible), the Alpha go team used a deep learning technique that would use board recognition rather than trying a full exhaustive search. The technique would use a system of convolutional layers that would reduce the search size of the game state. This enable the AI agent to evaluate a reduced representation of the board indirectly vs the raw board state.

Some layers in the training pipeline:

Using supervised learning based on actual human player moves and 30 million positions of simulated moves from the KGS Go Server, a policy network is trained to make a probabilistic model os creating a pool of next moves most probable to being beneficial compared against the entire superset of moves. This drastically reduces the size further by evaluating only a subset of possible next moves. To further speed up the training this network is trained using SGD so only a random sample of data is used to train weights of this first layer.

The next layer in the pipeline uses a reinforcement learning approach with random opponents to prevent overfitting and a rewards system of +/- 1 for winning and losing states respectively.

The final layer was to evaluate the the position of a prediction by memorizing game outcomes using the KGS data. However this ultimately led to some overfitting by using sequential data from common games. To help get around this, an faux dataset was generated and sampled from hereby mimicking sampling from many different games.

These policy and value networks are then combined into a MCTS (Monte Carlo Tree Search)

Results & Conclusion

The efforts put into training the networks proved to be successful as to beating Dan Hui, a professional 2 dan Go player and winner of the 2013, 2014 and 2015 European Go championships. the Alpha Go program beat Hui 5 - 0. The program also beat other Go programs by considerable winning percentages against programs like Pachi, Zen and Crazy Stone.