This repository Search Pull requests Issues Marketplace Gist

evizitei / isolation-agent

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

Branch: master isolation-agent / research_review.md Find file Copy path

evizitei include reference to source paper 592b2e6 36 seconds ago

1 contributor

58 lines (44 sloc) 3.09 KB Raw Blame History

Research Review: Alpha Go

Go has long been an elusive target for AI experts, and state-of-the-art AI was previously thought to be many years away from competitive play with leading professional players.

Source Paper: https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf

Goals

The main goal of the AlphaGo initiative was to apply recent advances of convolutional networks in visual domains to the 19x19 "image" of a go board, generating a competitive professional player AI.

This was planned by training 2 networks: a policy network for picking likely moves, and a value network for evaluating the utility of a position.

The policy network is trained on professional and expert human moves from a corpus of recorded games on the KGS go servers. This uses convolutional layers interspersed with ReLUs and capped off with a softmax over all legal moves (361 legal moves at the beginning of the game, strictly less from then on). This policy network is then supported by a Monte Carlo Tree Search algorithm which performs rollouts on chosen moves to better evaluate the resulting board positions.

That base policy is then trained using reinforcement learning (with the supervised learning policy network as a baseline), playing the network against prior versions of itself, to nudge the weighting away from predicting likely next moves and towards the goal of winning games (a subtle but important difference). The reward function is very sparse, awarding 0 for all timesteps before the end of the game, and then applying +1 for the winning player and -1 for the losing player to all positions back through the game.

The Value network is trained by playing the policy network against itself and learning to evaluate positions according to which player is likely to win. The value function was trained using regression on state/outcome pairs (the state of a board position and the eventual outcome of that game) and was optimized by minimizing the mean squared error between the value for the position and the outcome of the game.

Results

The base Supervised Learning policy network achieved 57% prediction accuracy on the held out test set from KGS. That sounds low, but is actually an incredible result given how many options there are for each move, and how close in value some of those options can be. As an advanced amateur go player myself, playing through a recorded professional game, I can only predict with accuracy 20-to-30% of moves (though I can do better if allowed to choose 4 or 5 likely moves at each position). State of the art for other research at the time was 44%.

The reinforcement learning policy network, with no search, won 85% of games against current strongest open source AI player (Pachi), and in tournament play ranked several ELO above the best AI programs available today.

The most significant result was the fully integrated (and distributed) program defeating a professional go player (first Fan Hui, at the time of this paper, then later a former world champion Lee Sedol and then the best human player alive today, Kie Je). This was a feat expected to be at least 10 years off.