AlphaGo analysis

Deepmind team had the challenge to create an effective move selection and position evaluation heuristics to play Go at a professional level. This opens a world of possibilities for extending thise methodologies to domains where full depth search is impossible and decision making is unclear.

So here is how they did it:

They used two concepts, one for each challenge:

Policy networks: allow them to select moves based on probability distribution to be later consumed my sampling, looking for the highest.

Value networks: evaluate position, to predict outcome value of a given state to avoid searching to the end game.

Graph Implementation:

Input: The board was represented as a 19x19

Policy Network: Conv. nets over image of a given state S to abstract positions in a representation with a smaller space to reduce depth and breadth of the search tree.

Value Network: Conv. nets over image of a S' state where the policy network placed the selected move that needs evaluation.

Training Pipeline:

1. There is the Policy network was trained in two parts:

Supervised policy network is used for expert knowledge in the player heuristics. So their preferred moves were used to change the selected move probability (P) for sampling later. Internally convs/rect/conv.../softmax=P(legal_moves)

RL policy network: This networks aims to improve Supervised policy playing against previous versions of self. This is done by calculating the P(P(legal_moves)) to maxime wining the game, as oppose to predicting expert moves in Supervised policy

Internally it is equal to the Supervised policy network, but a reward function is added at terminal state for those moves winning the game agains randomly selected version of the Policy network. . For optimization, SGA was used to maximize the probability

2. The Value network was trained using as imput the policy with the strongest probability of winning, produced by RL network. For optimization, SGA was used to maximize the probability

Internally, regression was used to approximate the Expected Value outcome from positions. For optimization SGD was used in order to reduce error to real outcome. Reducing the overall error using

whole games lead them to overfitting but it was resolved generating new samples by self playing.

Evaluation:

After choosing the strongest probabilty for each of the policy networks for a give move the value of the move is estimated using a combination of random montecarlo rollout until terminal state and Value network output (SL).

Conclusion:

Altough more expensive, the use of deep neural networks for policy sampling and value calculation proves that a ML pipeline can combine the best human heuristics with superhuman self-training even in the context of unattainable search space to reach professional level at GO.