Research Review: AlphaGo

Problem overview

As we know in many simple games search algorithms like MIN-MAX or MIN-MAX with Alpha-beta pruning can achieve good results compared with best human players. If we have **b** different moves and **d** moves to complete game in average the search space to find optimal move is contains **b in a power of d** variants. For the games with high values for **d** and **b** search of optimal moves become computationally impossible. For example for chess we have (b≈35, d≈80) and for Go we even have more amazing numbers: (b≈250, d≈150). So, to deal with such games we need to find a way to reduce both depth and breadth search spaces. Traditionally, to reduce depth space heuristic evaluation functions are used. And breadth space was reduced using sampling policy based algorithms like Monte Carlo tree search. This worked perfectly for chess but not for Go with its huge search space. So, new ideas were required in this area to proceed.

Ideas and techniques

The main idea was introduced there is to combine Monte Carlo tree search algorithm together with Deep convolutional neural networks for position evaluation and sampling policy. Although idea of using Neural nets was not new by itself – some existing Go programs used NN to learn weights for the features of evaluation functions (for example: W1 * territory secured + W2 * territory influenced + W3 * number of atari + other). The way how Deep Mind team introduced NN usage was innovative.

In recent years, Convolutional neural net archived human-like performance in image recognition and classification, converting image pixel to its internal semantic representation.

The main idea was to use board as 19x19 images passing it into two convolution neural networks – value network for position evaluation (heuristic function) and policy network for sampling actions for Monte Carlo tree search. The intuition behind this is that humans evaluates position and possible moves in a similar way – analyzing images of stone shapes.

The second idea was a training process organization:

- It started from supervised learning by training Policy network on a 30 million of position from KGS Go server
- Then Policy network was improving playing with random previous version of same network (reinforcement learning)
- The final stage is training of value network. The idea there is to use MSE as metric between predicted outcome and actual outcome estimated using Policy network from the previous steps. As the result position evaluation using value network produced same accuracy as simulation using Policy network but 15000 faster.

Another interesting idea which was introduced is to combine both policy and value networks results when evaluating specific move using balancing parameter λ that can switch algorithm from pure Monte Carlo tree search using rollouts (λ=1) to pure heuristic function evaluation of the successors of the move with heuristic function represented by value network (λ=0). However, it achieved best performance with (λ=0.5) using both value and policy network in same degree winning ≥95% of games against other parameters.

The last improvement was that algorithm provided is fully distributed and can be run on multiple computers with CPUs and GPUs to archive even better performance.

Results

The results were amazing.

AlphaGo won 494 out of 495 games (99.8%) against other Go programs. Even using algorithm only with value network estimation without policy network (λ=0) it was able to beat all existing Go programs.

- In October 2015 AlphaGo defeated the European champion Fan Hui
- Won match against Lee Sedol (ranked 9-dan, one of the best players at Go) on March 2016
- And finally, AlphaGo won three-match series against the world's champion Ke Jie

Basically, AlphaGo achieved one of artificial intelligence's "grand challenges"

Conclusion

As a conclusion, we can come up with two stunning facts:

- There is minimal domain knowledge required about the game for the introduced AI to achieve outstanding performance. Although there are some GO specific features were provided to the Network as input, most knowledge were extracted from its own playing experience.
- Second fact is even more amazing - even though initially policy and value convolutional networks were trained using human played games, it is totally possible that this AI can learn game from scratch using only reinforcement learning part - it just can take a little bit longer to archive.