Mastering the game of Go with deep neural networks and tree search

Summary by Amir Ziai @amirziai

The game of Go is known as one of the toughest games in the AI community and winning against professional Go players was assumed to be a feat that's at least a decade away before this work at DeepMind. Contrary to what one might think exhaustive search is not possible with even simpler games. For chess the average number of legal moves is 35 (known as breadth or b) and the average length of the game (known as game depth or d) is 80 which yields an astronomical 35^{80} search space. The search space is much bigger for Go with b=250 and d=150. The only solution to this problem is to find clever ways of reducing depth and breadth. For depth reduction the strategy that has led to superhuman performance in chess is to find ways to approximate subtrees without actually expanding them with brute force. This was believed to be intractable with Go previously. For reducing breadth the prevailing strategy to to do a sampling strategy over the policy of available actions.

DeepMind uses a number of strategies to improve the state-of-the-art, which in combination yield a system that defeated the European Go champion 5 to 0. The first component in the pipeline is a supervised learning 13-layer deep neural network with convolutional layers trained over a 30 million position dataset from the KGS Go Server. The reported accuracy on the held-out set is 55.7%, which may not sound very impressive, but is a massive improvement over the 44.4% state-of-the-art system and according to the paper small improvements in this accuracy lead to larger improvements in player strength. The second component of the pipeline is a Reinforcement Learning (RL) network, identical in structure to its Supervised Learning (SL) counterpart, that is able to beat SL in more than 80% of head-to-head games. The RL piece also won 85%+ of games against the best open-source Go program called Pachi. The final component of the pipeline is a position evaluation reinforcement learning network that learns a value function given the state.

By using more intelligent value and policy networks AlphaGo evaluates 3 order of magnitude less positions per move compared to Deep Blue in the match against Kasparov. Two other facts stand out about AlphaGo vs. older systems like DeepBlue. First AlphaGo does not rely on a handcrafted evaluation function and the networks are trained directly from game play. Second, the radically less number of evaluations per position more closely mimics how humans play. If history is any indication breakthroughs like AlphaGo will not be isolated to game playing and have implications in broader AI applications such as planning, scheduling and constraint satisfaction.