Timothy Ryan

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

I will start off by saying that a lot of the terminology of this paper was a little above my head. Nonetheless, I hope to provide a satisfying overview of what was accomplished by the DeepMind team in terms that AI novices like myself can understand.

There are 3 major architectures at play here, so I'll take them each in turn: policy networks, value (or evaluation) networks and Monte Carlo tree searches.

Policy networks try to figure out the best next move. These networks are based around deep convolutional neural networks, which have been successfully used in the past to identify the content of images. These networks take in the current board and output a probability map for the next move with the greatest probability of winning. They were first trained using supervised training on using games publicly available online. Using each board position, the network was trained to reproduce the next move on the board. Once the network was trained in this way, it was again trained using reinforcement learning on games against itself. Moves that eventually led to a win for the system were weighted more heavily over time.

The next network is harder to understand conceptually. Evaluation functions try to give a numeric value to the position of a board. This is run on the resulting board moves to try and predict the value of a board after a move as been made, i.e. to assess whether a move puts the computer player in a stronger position. This network was not trained using supervised learning, as this would've required a standardized set of values assigned to boards by impartial observers. Instead, this was trained solely by reinforcement learning: boards that eventually led to victories were given higher values, and the factors that led to better boards were learned from scratch by the computer over time.

The last network is a Monte Carlo tree search (MCTS). This search uses the values from the previous two networks to determine which trees to go down in a way that I'm going to be honest about not understanding. That said, the previous two networks allow the MCTS to search only the most likely moves and then choose the best among those.

The results of these search algorithms have been very impressive. AlphaGo easily beat all other existing Go playing algorithms. Moreover, it was the first system to beat a professional Go player. Later iterations would go on to beat the two best go players in the world. AlphaGo has really led to the end of fully observable, deterministic, discrete tasks, adversarial tasks from being the test of AI.