Programming an Adaptive Artificial Intelligence Utilizing Neural Networks and the Monte Carlo Tree Search Method

Lawrence, Jr., Trenton

Hoehne, Rush

Lane III, Larry

The objective of this project was to develop an artificial intelligence for an original strategy game that adapted and learned from its past successes and failures and applied its new knowledge to future games that it plays. We used neural networks for analysation of past moves to efficiently predict enemy moves. Neural networks were used in conjunction with the Monte Carlo Tree Search Method (MCTS). The MCTS is a heuristic search algorithm that has previously been used almost solely in board game scenarios. We used the MCTS outside of a board game demonstrating its capabilities outside of board games, and also demonstrated that it is rather efficient, especially when used in conjunction with neural networks. We wrote the foundation of the strategy game first and foremost, which allowed us to tailor the AI to the game itself. When we had finished the game, we moved onto designing the AI. We decided that we would implement a variation of the MCTS, the leading method used in artificial intelligence for the board game Go. We implemented a simple neural network to analyse old moves in an attempt to help out the decision process of the MCTS. Early on, one of our biggest issues was the sheer amount of turns that the MCTS was trying to analyse and the massive quantity of move paths it had to follow. We resolved this issue by trimming out the weakest move paths, and just following the stronger ones. For each trial, two artificial intelligences played 100 games against each other. We ran 100 trials of 100 games, and observed many trends involving the data, the most important of which was the nearly linear growth in efficiency for each run. We concluded that the artificial intelligence grew more efficient over time, and that the MCTS is viable outside of a board game scenario.