Artificial Intelligence Nanodegree

Alpha Go Deep Neural Networks And Tree Search

Guillermo Aure
June 28, 2017

Summary

The following summary explains at a high-level detail the content of the "Mastering the game of Go with deep neural networks and tree search." The central topic revolves around how to create an artificially intelligent agent capable of playing the Go game. For years the creation of AI agent capable of playing the Go game against a human player has been one of the most challenging areas of the AI game space. The problem as per the paper is related to the game search space and the definition of an optimal policy that led an AI agent to play as at least equal to its human counterpart. The first of the problems is because the Go game board has 250 possible legal moves at the beginning of the game and it can go up to 150 plies before the match ended; to put this in perspective, the chess game has 35 legal moves, and up to 80 plies before the event ends. The search space of a Go game is not only massive in depth but also in breadth; exploring this space and returning an optimal play in a reasonable time is almost impossible. Traditional technics like depth limit and pruning are still insufficient due to the vast search space and the difficulties of finding an adequate optimal policy. Efforts of creating an agent before AlphaGo (name of the Go AI agent created by the Google team) have partially succeeded, optimizing the traditional tree exploration, by a technic called Monte Carlos Tree Search (MCTS). The technic alleviates the tree search and creation of an optimal policy by using full game simulations at specifics nodes depth to determine which branches have the best likelihood of winning the game and that way help to create an optimal policy. Even with this optimization, the AI agent was only capable of play at an amateur level. The problem faced by the MCTS technic is that it is not possible to go deep enough in a reasonable time to get a value that can produce an optimal policy. The issue was improved but still was the same to big not enough time. The AlphaGo time faced the problem using a different approach, instead of continuing pursuing the optimization of the MCTS technic, they decided to complement is by using supervised learning technology specifically convolutional neural networks. In essence, instead of searching the best move in a graph to come out with and optimal policy, AlphaGo uses a combination of neural networks capable of learning from images and Reinforcement Learning Technics. The images are the representation of Go game boards. AlphaGo uses more than just one Supervise Learning neural network working in tandem; these neural networks are combined with another technic called Reinforcement Learning. The neural network training pipeline as it refers to in the paper works in the following way. A network is trained using images of Go game boards to learn which move a human expert player will take next from a specific state of the board. After the first neural network is trained its internal parameters (weights) are initialized the same on a second neural network, so this second neural network has the knowledge acquired by the first one, so it is capable of select a next move (human-like) from a specific Go board (learn by the first one). Finally, the

second neural network is improved by using Reinforcement Learning Technics; the improvement process consisted in simulating tournaments and injecting in the second neural network weight adjusting function (gradient ascent function) a modifier which is the reward obtained every time the second neural network finishes a game in time and as the winner. After the second neural network training finishes, the AlphaGo agent beat every existing Go AI agent out there without using any tree search. A third neural network was used to calculate the value of state based on the result of following the recommended actions of the second neural network. The result of the used of second and third neural networks and a Monte Carlos Search Tree is that at each state of a search the optimal policy is a combination of the prediction of the best human action, the count of winning vs. simulate games and the utility value of the state. The AlphaGo beat the European world champion 5 to none in a tournament. The conclusion is that the use of the MCTS plus the deep learning neural networks has proof the potential of an artificially intelligent agent to deal with human strategic complex reasoning.