Chad Iverson

Artificial Intelligence Nanodegree Program

December 30, 2017

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

In this research paper, Silver et al. (2016) describe how the Google DeepMind team achieved a major milestone for AI in October of 2015. Their game playing agent, named AlphaGo defeated the European Go champion in 5 of 5 games. The paper outlines how AlphaGo was trained on a novel set of data and leveraged a new search algorithm based on Monte Carlo simulation. Up until this achievement, AI researchers considered Go the final frontier for board games, with search trees typically having a branching factor of 250 and a depth of 150 levels (Allis, 1994). According to Silver et al., some of the best performing Go programs address this complexity with Monte Carlo tree search (MCTS) and policy networks trained using expert player game data.

In developing AlphaGo, the DeepMind team took a creative approach to overcoming known limitations with accuracy and performance of policy networks. In particular, they introduced the idea that reinforcement learning could be used to build upon policy networks trained first with expert player game data. The result was a policy network, used to predict the best next move, capable of competing with state of the art programs. The AlphaGo network achieved an 85% win rate facing off against Pachi, a program reportedly the strongest open source Go player at the time the paper was submitted for review. Interestingly, this was achieved during competition using the reinforcement learning model alone, without relying on search at all. More importantly, this highly accurate policy network could be used to generate gameplay data needed to train the value network.

AlphaGo uses a value network to back game state evaluation functions. These functions play a role in maintaining action values as games are played and simulations are run. Unlike other researchers, the DeepMind team trained the value network with a combination of expert player game data and their own policy-network generated gameplay data. This approach helped avoid the overfitting pitfall associated with training models for game state evaluation. The trained value network achieved a mean standard error (MSE) reasonably consistent between training and test data, at 0.226 and 0.234. According to Silver et al. (2016), this model also “approached the accuracy” (p. 486) of the policy network while demanding 15,000 times less computing resources.

The DeepMind team also introduced another new technique: MCTS enhanced by both policy and value networks. Rather than using randomly selecting moves for evaluation, the AlphaGo search algorithm leverages the policy network to obtain prior probabilities used for initial action selection. Simulations are run using a fast rollout version of policy network. The simulations outcomes are propagated back up the tree, updating action values and visit counts for edges as the tree grows. Finally, these edge properties are used as part of edge selection before subsequent simulations.

The key result coming out of this work is the intelligent agent capable of playing Go at the human level. Beyond this major milestone for AI, other results include the production of networks having unique accuracy and performance characteristics when compared to prior related work. Arguably, the most significant effect of this publication is the influence it has had on the field of AI. With over 1900 citations, according to

Google Scholar, in just under two years, AlphaGo has inspired many researchers and practitioners. The takeaway for emerging AI engineers might be this: to achieve human-level gameplay, explore the potential of general-purpose machine learning techniques over handcrafted evaluation functions.

References

Allis, L. V. (1994). *Searching for solutions in games and artificial intelligence*. PhD thesis, Univ. Limburg, Maastricht, The Netherlands.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*(7587), 484-489.