Md Majid Jahangir | SBU ID - 112077145
AMS 560 - Big Data Systems, Algorithms and Networks

Assignment 2 (Paper Reading)
Mastering the game of Go without human knowledge

Motivation : There has been a lot of progress in the field of artificial intelligence using supervised learning systems which are trained to replicate human experts' decisions.However, most of the time, these expert data sets are often not available, or unreliable or very expensive. Even though the expert data sets are available, the performance of such supervised learning systems will be constrained by the human experts' data and the system's performance will be upper limited by the performance of the human expert. Hence, reinforcement learning systems are the best alternative here, which are trained from their own self-play experience, without any supervision or use of human data, and enables them to explore domains where human expertises lack and hence allowing them to exceed human capabilities. Therefore, this paper proposes AlphaGo Zero, a self-play reinforcement learning system, that achieves super-human performance without human domain knowledge.

Prior Solutions: Prior solutions to AlphaGo Zero are the previous versions of AlphaGo namely AlphaGo Fan, AlphaGo Lee and AlphaGo Master. AlphaGo Fan was the very first published version of the program that defeated European Go Champion Fan Hui in October 2015. Fan used two deep neural networks, a policy network that was trained by supervised learning to predict human expert moves and a value network which was trained to predict winner of the games played by policy network against itself. Both of these networks were combined after being trained with a Monte Carlo tree search (MCTS) to provide a lookahead search, and using the policy network to narrow the search to high-probability moves and using value network to evaluate positions in the tree. AlphaGo Lee defeated Lee Sedol in March 2016 used similar approach as Fan. However, it differed from Fan in the value network that was trained from outcomes of fast games of self play by AlphaGo, rather than games of self play by the policy network. Also, the policy and value networks were larger than Fan to train it for more iterations and the player was distributed over TPUs than GPUs, enabling it to evaluate neural networks faster during search than Fan.
AlphaGo Master uses the same neural architecture, MCTS algorithm and reinforcement learning algorithm as stated in this proposed paper [1], however the training was started by supervised learning from human data and uses the same features as AlphaGo Lee.

Key Ideas of Proposed Solution: The key ideas of the proposed solution are -

1. **Self-play reinforcement learning -** The proposed solution does not incorporate any supervision from any human data. The program uses a deep neural network. This neural network takes the raw board representation of the position and its history as an input, and it gives an output of move probabilities and a scalar value.The vector of move probabilities represents the probability of selecting each move and the scalar value is an estimation of probability of current player winning from that position. This method uses a single neural network that combines both policy and value network that has been discussed above in "Prior solutions" section. This neural network is trained from games of self-play than any human experts' data by a reinforcement learning algorithm. The reinforcement learning algorithm used here is an approximate of policy iteration scheme that generates a sequence of improving policies by alternating between policy evaluation, estimating the value of the current policy and then using the current value of the policy to generate a better policy. The neural network is trained by a self-play reinforcement learning algorithm that uses a Monte Carlo tree search (MCTS) to play each move. The neural network is initialized with random weights and at each subsequent iteration, games of self play are generated. Also for each step an MCTS

search is executed using the previous iteration of neural network and then a move is played by sampling the search probabilities returned by MCTS.

2. **Domain knowledge -** The performance of AlphaGo Zero is achieved without any human experts' data or human domain knowledge. Some of the key domain knowledge that AlphaGo Zero uses are perfect knowledge about the game rules which is used during MCTS to simulate positions resulting from a series of moves. It uses Tromp-Taylor scoring during MCTS simulations and self-play training.
3. **Self-play training pipeline -** The self-play training consists of three main components - neural networks parameters are continually optimized from recent self play data; players are continually evaluated; and the best performing player so far will be used to generate new self-play data.
4. **Evaluator -** Each neural network checkpoint is evaluated against the current best network, before data generation. The evaluation is done by the performance of an MCTS search and a best move and the corresponding best player is decided.
5. **Self-play -** The current best player decided by the evaluator is used now to generate self-play new data.

Performance : During the training phase, AlphaGo Zero started to train from a random behavior state without any human experts' data set for 3 days and surprisingly in just 36 hours Zero outperformed AlphaGo Lee, where Lee was trained over several months. After 3 days of training of Zero, it was evaluated against AlphaGo Lee that defeated Lee Sedol, under the same match condition that was used during the match. Zero defeated AlphaGo Lee by 100 games to 0 by just using a single machine with 4 TPUs, on the contrary AlphaGo Lee was distributed over several machines and used 48 TPUs.
In a second instance of AlphaGo Zero, larger neural network was used and the reinforcement learning pipeline was applied for a longer duration and training was done over a period of 40 days. The fully trained AlphaGo Zero was evaluated in an internal tournament against AlphaGo Lee, AlphaGo Fan, AlphaGo Master and other variants of Go programs. AlphaGo Zero and Master were run on a single machine with 4 TPUs while AlphaGo Lee and Fan were distributed over 48 TPUs and 176 TPUs respectively. The performance of each program was evaluated on an Elo rating scale. AlphaGo Zero achieved an Elo rating of 5185 compared to 3739 for AlphaGo Lee, 4858 for AlphaGo Master and 3144 for AlphaGo Fan in the internal tournament.
Also, in a head to head 100 game match against AlphaGo Master, Zero won by 89 games to 11.

Difference between AlphaGo Zero and AlphaGo Lee? [Key Question]:

1. AlphaGo Zero is trained just by self-play reinforcement learning, starting from a random play, and without any knowledge of any human domain data; whereas AlphaGo Lee was trained using supervised learning from human experts' move data sets.
2. AlphaGo Zero just uses the rules about the game and the black and white stones from the board as input features; whereas Lee uses the above mentioned features as well as the human experts' data set as input features to train on.
3. AlphaGo Zero uses a single neural network to evaluate positions and to sample the moves without performing any Monte Carlo rollouts; whereas AlphaGo Lee uses two networks namely a policy network and a value network. Both of these networks were combined after being trained with a Monte Carlo tree search (MCTS) to provide a lookahead search, and using the policy network to narrow the search to high-probability moves and using value network to evaluate positions in the tree.
4. AlphaGo Lee is distributed over many machines using 48 TPUs whereas AlphaGo Zero was run on a single machine with just 4 TPUs.

How does AlphaGo Zero learn by playing with itself? [Key Question]:

This section and the below section explains the methodology of how AlphaGo Zero learns by playing with itself -

The program uses a deep neural network. This neural network takes the raw board representation of the position and its history as an input, and it gives an output of move probabilities and a scalar value.The vector of move probabilities represents the probability of selecting each move and the scalar value is an estimation of probability of current player winning from that position. This method uses a single neural network that combines both policy and value network that has been discussed above in “Prior solutions” section. This neural network is trained from games of self-play than any human experts’ data by a reinforcement learning algorithm. The reinforcement learning algorithm used here is an approximate of policy iteration scheme that generates a sequence of improving policies by alternating between policy evaluation, estimating the value of the current policy and then using the current value of the policy to generate a better policy. The neural network is trained by a self-play reinforcement learning algorithm that uses a Monte Carlo tree search (MCTS) to play each move. The neural network is initialized with random weights and at each subsequent iteration, games of self play are generated. In each iteration, Zero plays 25000 games of self play using 1600 simulations of MCTS to select each move. For each step an MCTS search is executed using the previous iteration of neural network and then a move is played by sampling the search probabilities returned by MCTS.

How does the Monte-Carlo tree search work in AlphaGo Zero? [Key Question]:

For each iteration, in each position of the board, a Monte Carlo tree search (MCTS) is executed which is guided by the neural network. The MCTS search gives the probabilities of playing each move as an output. This search operation is performed in policy iteration procedure where the neural network’s parameters are updated to match the scalar value and move probabilities returned by the neural network closely with the improved search probabilities and self play winner. These new parameters are then used in the next iteration of self-play to make the search stronger. Each edge in the search tree stores a prior probability of previous iteration , a visit count and a scalar value. Each simulation of MCTS starts from root and iteratively selects moves that maximizes the probability until a leaf node is encountered. In a simulation, each edge traversed is updated to increment the visit count and to update the scalar value. This vector of search probabilities returned by MCTS is proportional to the exponentiated visit count for each move.

Strength :

1. It uses self-play reinforcement learning without any intervention of human data or human domain knowledge, thus overcoming the limitation where the program’s performance could be limited by the human expert’s domain knowledge. This enables it to achieve a super human performance by exploring domains of the game that has not been explored by a human before, thus exceeding human capabilities.
2. It just uses one single neural network that can be run on a single machine with just 4 TPUs compared to AlphaGo Fan or Lee which needs 176 TPUs and 48 TPUs respectively distributed over several machines.
3. By using the policy iteration scheme explained in above sections, the probabilities vector returned at each iteration by MCTS is stronger than the previous iterations. This helps in selecting the best and stronger move possible for each player’s move.

Weakness :

1. Self-play reinforcement learning algorithms or any neural networks algorithms are a black-box i.e. we have no idea as to how the program learns, which path it takes to decide the best moves, or what are the search spaces it explored before deciding on a move.
2. Go is a game which is deterministic, discrete and easy to simulate and the only challenging aspect of it is the huge branching factor. Therefore, even though the success achieved by AlphaGo program in the past future is huge in such a challenging domain of artificial intelligence, it cannot be used as a sign of imminent Human-level AI [2] because the real world is more complex than the game of Go.

3. AlphaGo Zero is an example of Weak AI or Narrow AI agent i.e. it is capable of performing only one “narrow” task at a time. Even though AlphaGo Zero could play 3 different board games, it does so separately per game. Therefore, even though the achievements of it is impressive, the fact that it can be used as a sign of Human-level AI only if the techniques behind AlphaGo Zero could take us beyond playing the game of Go.

References -

1. Mastering the game of Go without human knowledge - David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel & Demis Hassabis
2. AlphaGo Zero Is Not A Sign of Imminent Human-Level AI - Skynet - https://www.skynettoday.com/editorials/is-alphago-zero-overrated