Research review

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

- summary of the paper's results

In my country, South Korea, there was a Go challenge match with AlphaGo and Sedol Lee in last year. Go is a very ancient game and it's one of the most complex game. AlphaGo is Go game program that developed by Google DeepMind. As you know, the result was a victory of AlphaGo. This was a very important milestone of Artificial Intelligence. So, I was interested in this topic and choose it.

According to the paper, AlphaGo was implemented by many machine learning model. Such as Convolutional Neural Network(CNN), Monte Carlo tree search (MCTS) and Reinforcement learning. Actually it was very hard for me, because I am a just a entry level developer. But, it was very interesting. In this paper, AI program won the game against professional human player. It was first time to make victory against human being in Go.

- summary of the paper's goals or techniques

Before AlphaGo, there was two 2 techniques in Go AI program: Reducing searching depth and breadth search by sampling action. By implement those two techniques, we could build a Go machine as amateur level.

Additionally, AlphaGo added two main concepts are that CNN and MCTS. And CNN was implemented by Deep neural network and MCTS was implemented by Monte Carlo rollouts. So, AlphaGo used 2 network: policy network, value network. And then it mixed MCTS. There are some steps for learning algorithm.

Step 1. Supervised Learning (SL) of policy networks - In this step, machined was learned by human player's record. They used CNN for this step. CNN has very good benchmark in classification. According to this paper, after this step, the machine can predict opponent's play at 57% probability.

Step 2. Reinforcement Learning (RL) of policy networks - After step 1, the machine trained itself by trained policy network. Each learning phase, the machine was updated. So, after this step the machine more powerful than step 1. It means predictive

accuracy was also improved. They can reduce breadth by policy network. I means we can't calculate all number of cases.

Step 3. Reinforcement Learning (RL) of value networks - In this step, the machine find best case in current state. The value network can reduce searching depth. And they used MCTS in this step.

In conclusion, the machine can reduce breadth by policy network. And it also can reduce depth in value network. And they can find best case by MCTS.

재귀적인 검색 트리로 체스나 바둑같은 보드 게임 문제를 푸는 경우의 수는 b^d 이다(b는 게임의 폭;각 위치의 유효한 이동 가능 수;판 위에 놓을 수 있는 경우의 수, d는 깊이;게임 길이;경기에서 둬야 할 수의 횟수). 체스의 경우$b \approx 35$, $d \approx 80$, 바둑의 경우 $b \approx 250$, $d \approx 150$로 이 모든 경우의 수를 계산해 수를 두기는 불가능하다.

이를 해결하기 위한 방법으로 첫째, 검색의 깊이는 위치의 평가에 의해 감소될 수 있다. 서브 트리들의 상태 값을 예측하는 함수로 서브 트리들을 가지쳐낼 수 있다(ex. alpha-beta-pruning). 이 방법은 체스, 오델로 등에서는 매우 효과적이지만, 바둑은 훨씬 복잡하기 때문에 잘 작동하지 않는다. 둘째, 검색의 폭은 샘플링 정책 $p(a|s)$에 따라 줄어들 수 있다(위치s 에 따른 이동 가능한 경우의 수 a의 확률분포 p). 예를 들어 Monte Carlo rollouts에서 두 플레이어의 일련된 동작을 샘플링하여 전혀 분기하지 않고 최대 깊이로 검색을 한다. 이 방법을 평균화하면 바둑에서 아마추어 레벨정도의 학습을 할 수 있다.

Monte Carlo tree search (MCTS)는 Monte Carlo rollouts를 사용해 검색 트리에서 각 상태의 값을 측정한다. 현재 가장 강력한 바둑 프로그램도 MCTS를 기반으로 한다.

이전에는 얇은 정책을 쓰거나, 입력이 선형 모델만 가능하는 등의 제약이 있었다. 하지만 최근 CNN으로 이미지 분류, 얼굴 인식, Atari 게임 플레이 등 전례없는 성과를 이뤄내고 있다. 우리도 CNN을 통해 바둑의 유효한 폭과 깊이를 줄였다.

AlphaGo는 정책 및 가치 네트워크를 MCTS와 효율적으로 결합합니다. (MCTS, 강화학습, 딥러닝 다 쓴다는 말인 듯.)

처음은 CNN, 그 뒤로는 강화학습 (MCTS)