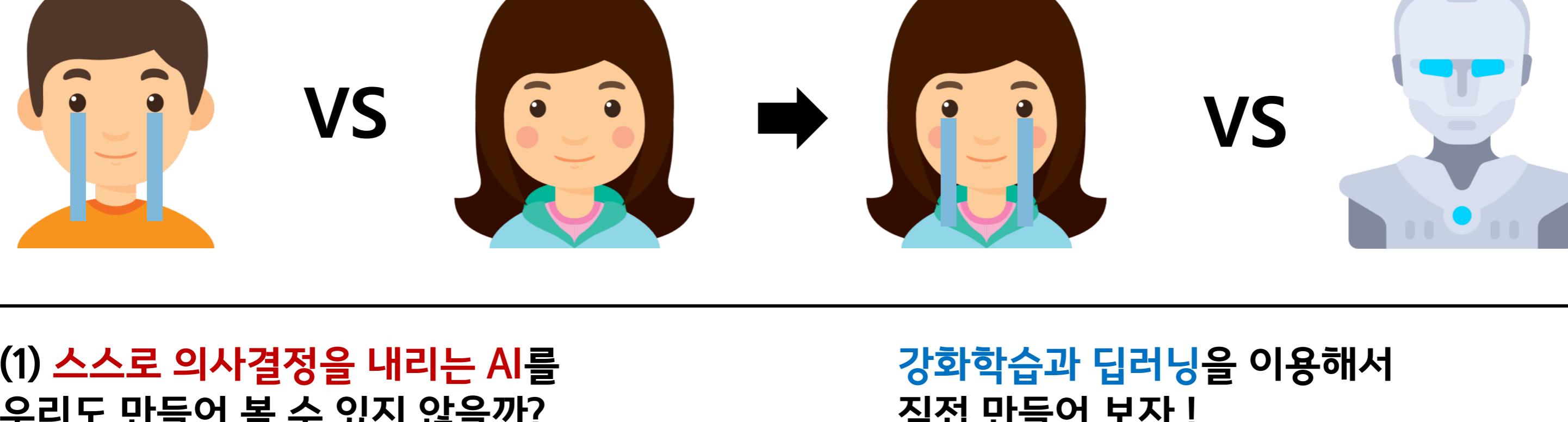


ToBigGo

강화학습과 딥러닝을 기반으로 구축한 오목 AI

김수지 / 서석현 / 안상준

1. 주제 선정 배경



(1) 스스로 의사결정을 내리는 AI를
우리도 만들어 볼 수 있지 않을까?

(2) 오목을 두었을 때 항상 지는 상대에게
나 대신 이겨주는 AI가 있었으면 좋겠다.

강화학습과 딥러닝을 이용해서
직접 만들어 보자!

이를 오목에 적용하여 인간의
간섭 없이 스스로 오목을 두게 해보자!

2. 오목 환경 구축

강화학습이란?
오목판 혹은 백돌 어떤 환경 안에서 정의된 에이전트가 현재의 상태를 인식하여,

선택 가능한 행동들 중
보상을 최대화하는 행동 혹은 행동 순서를 선택하는 방법
승리 베이 있는 자리 다음수

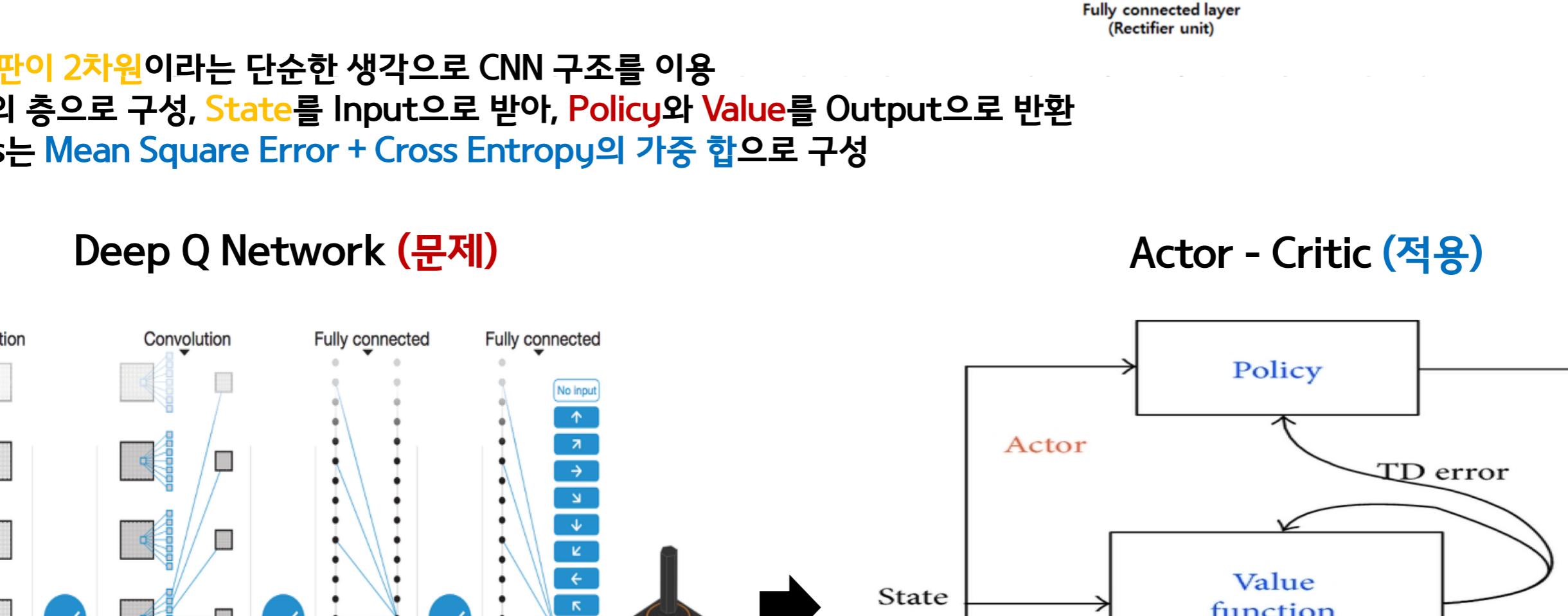


State
현재 오목 판에 돌들이 놓인 상태
[0, 0, 0, 1, -1, 0, -1, 1, 0 ...] → 9 * 9 * N Array

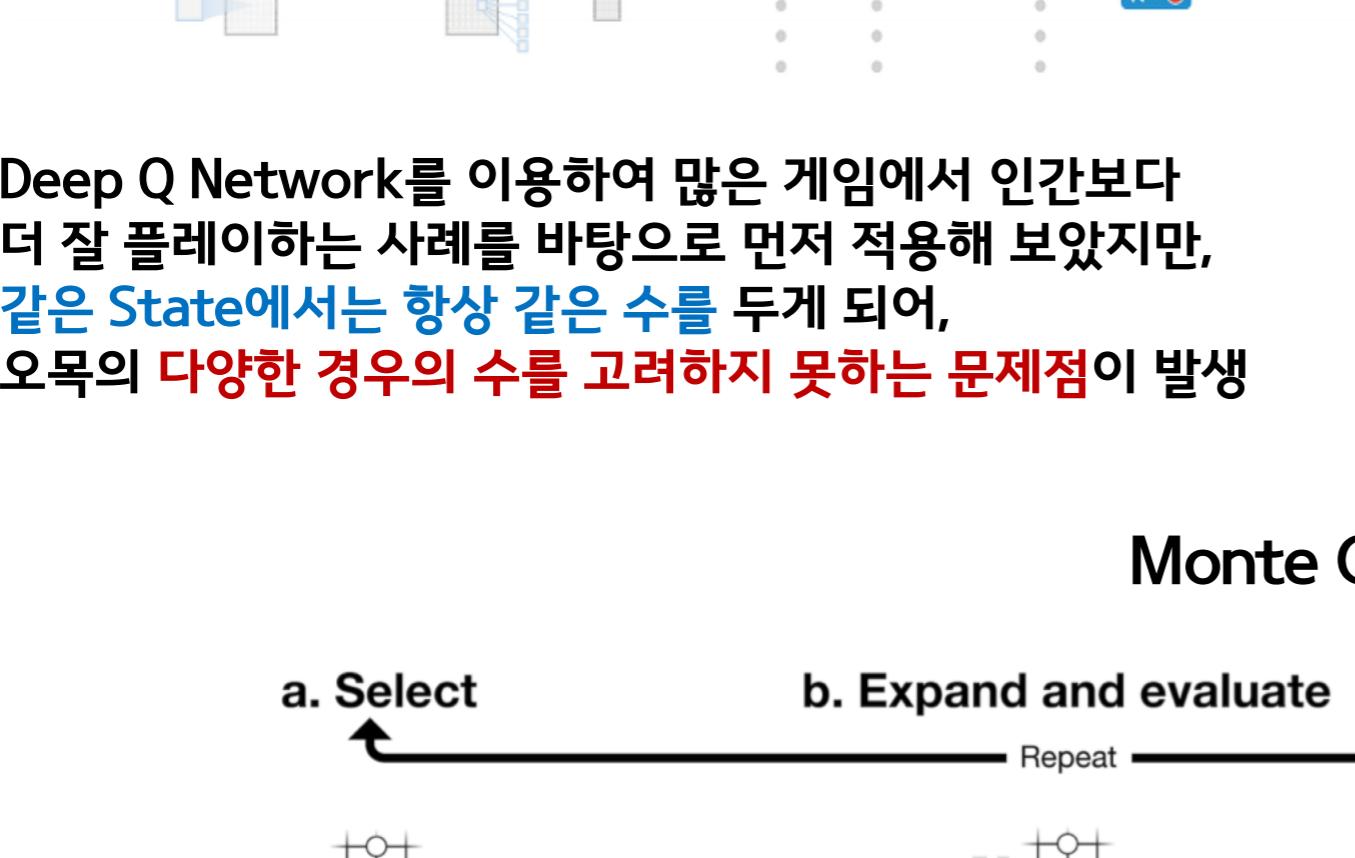
Action
다음 수를 둘 위치
[0, 0, 0, 0, -1, 0, 0, 0, 0, 0] → 9 * 9 Array
다음 수에 대해 판의 위치를 Array 내
1(흑돌) 또는 -1(백돌)로 표기

Reward
Agent에 대해
승리시 +1
패배시 -1
무승부시 0

3. Architecture



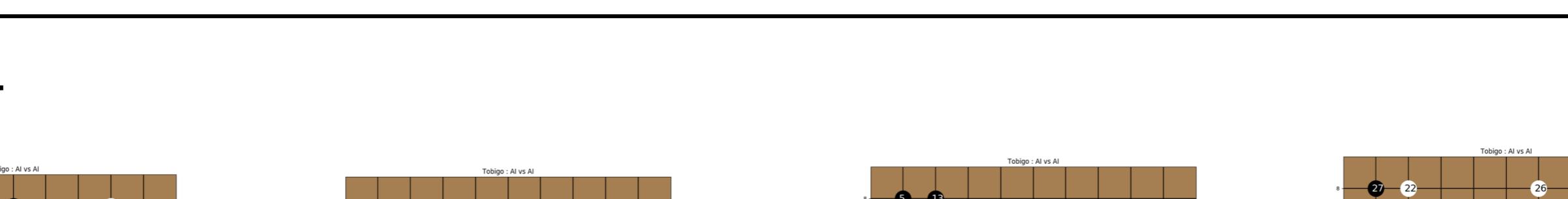
- (1) 오목판이 2차원이라는 단순한 생각으로 CNN 구조를 이용
(2) 3개의 층으로 구성, State를 Input으로 받아, Policy와 Value를 Output으로 반환
(3) Loss는 Mean Square Error + Cross Entropy의 가중 합으로 구성



같은 State에서 여러 Action을 Sampling을 통해
DQN의 문제점을 극복하며, 오목의 판세를 반영

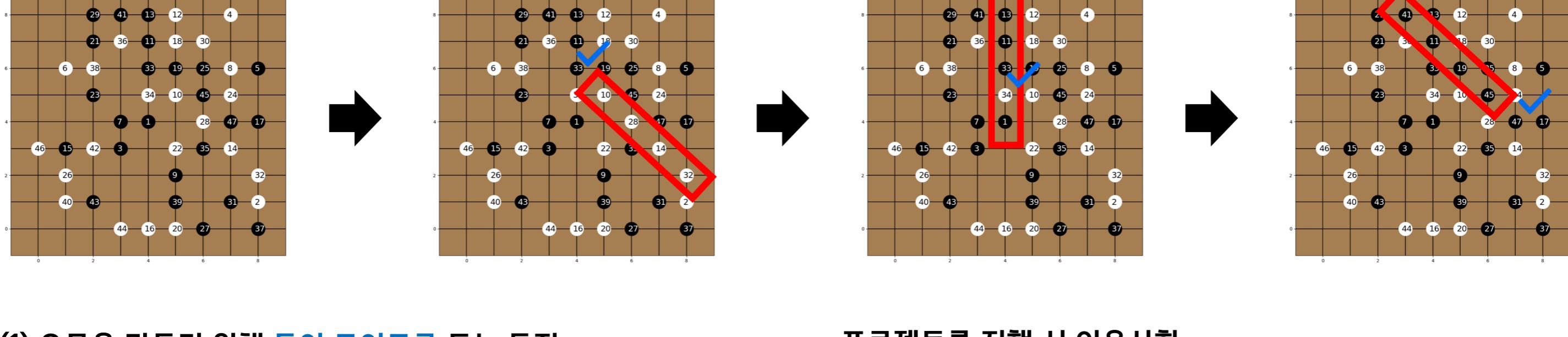
많은 Action 수와 Sampling을 통해 진행되기에 계산량이
너무 많아 학습이 너무 느리게 되었다.

Monte Carlo Tree Search



State를 기반으로 모든 경우의 수를 고려하지 않고, 적절한 시뮬레이션을 통해 근사값을 이용
Actor - Critic의 Policy, Value를 이용하여 Tree 내 Search Space를 축소 → 계산량 감소 효과

4. 결과



프로젝트를 진행 시 이용사항

- Laptop 3ea
- Python with Tensorflow
- Local 환경에서 실험 후 GPU (Tesla K80) Server 이용 (3일 반)
- 소요기간 : 대략 2달

- Laptop 3ea
- Python with Tensorflow
- Local 환경에서 실험 후 GPU (Tesla K80) Server 이용 (3일 반)
- 소요기간 : 대략 2달

- Laptop 3ea
- Python with Tensorflow
- Local 환경에서 실험 후 GPU (Tesla K80) Server 이용 (3일 반)
- 소요기간 : 대략 2달