PART 6

강화학습

딥러닝 & 강화학습 담당
이재화 강사

Part 6. 강화학습

규칙이 변하지 않는 아주 오래된 게임

1992년
바둑 서버가 개발되어 바둑 애호가들은
온라인에서 바둑 대전을 할 수 있게 됨

2016년
'Master' 라는 이름의 플레이어가
바둑 서버에 등장!

패턴이 짐작가지 않는군.. 누구지..

1968 1985 2016

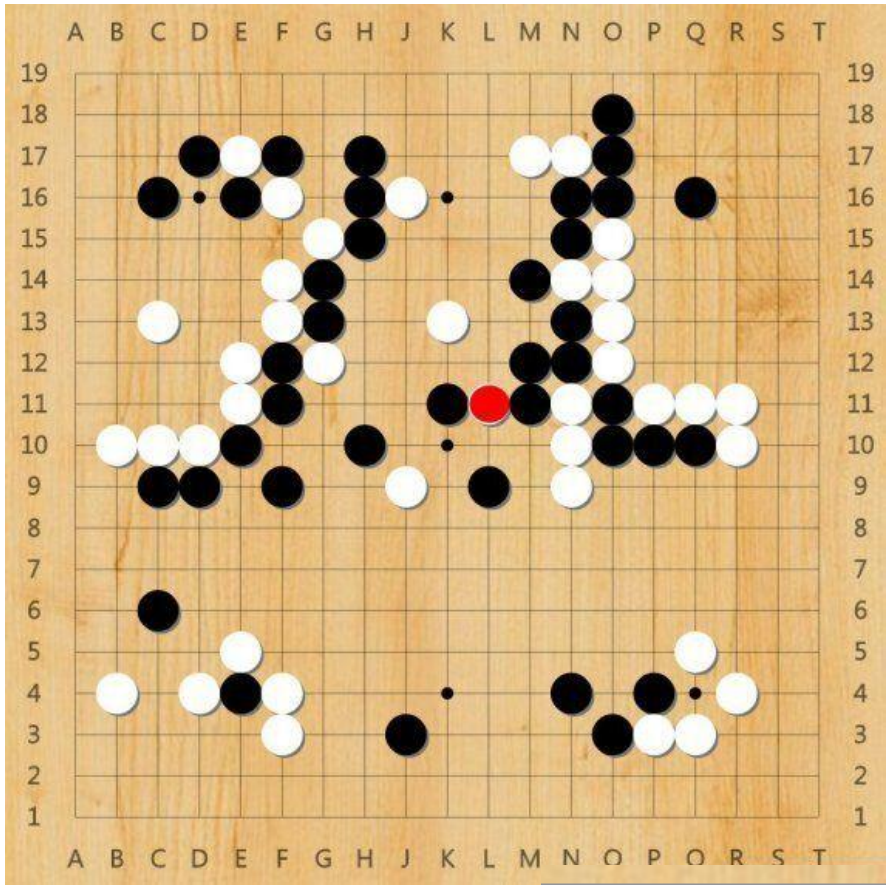
컴퓨터 바둑을 장려하고자
프로 바둑기사를 이길 수 있는 알고리즘을
만드는 사람에게 상금 40만 달러 수여

구글 딥마인드가 바둑을 두기 위해 만든
알파고의 온라인 비밀계정 'Master'

역시 바둑은 어렵구나...

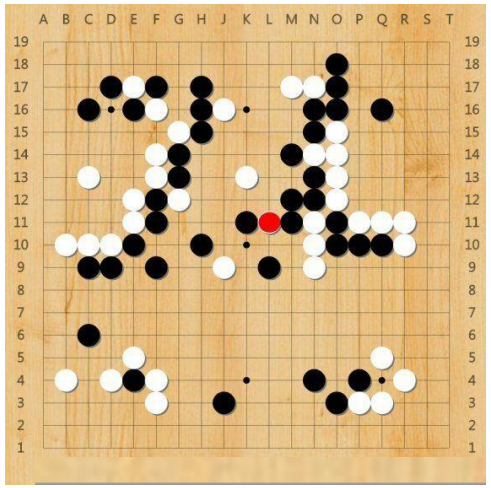
IBM Watson

Part 6. 강화학습



단 3차례 만에
10,000,000개 이상의 경우의 수

- ✓ 250여개의 수 중에서 하나를 선택
- ✓ 평범한 대국에서는 평균 150수

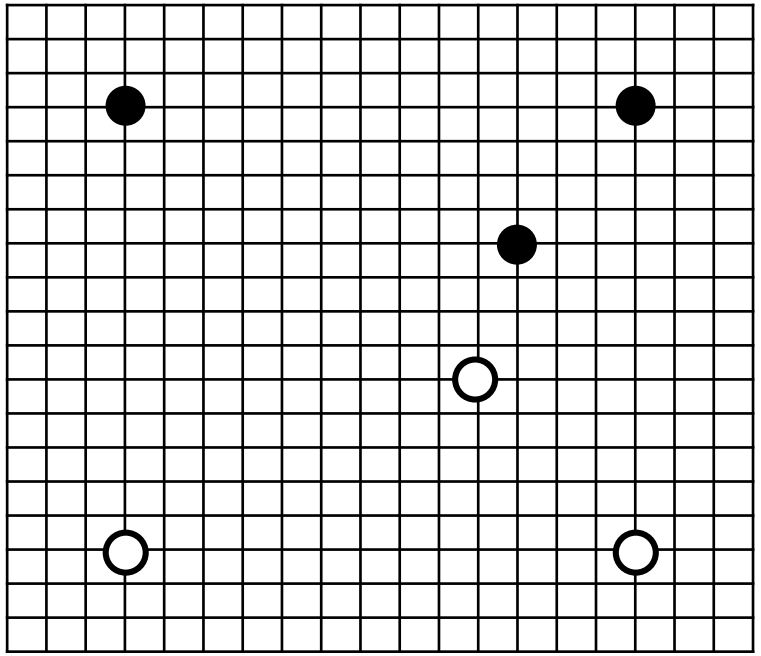


- ✓ 각자 흰 돌 아니면 검은 돌로 시작
- ✓ 플레이어는 19 X 19칸으로된 격자선 위에 자신의 돌을 놓습니다
- ✓ 바둑의 목적은 시합이 끝났을 때 자신의 돌로 지은 집이 바둑판에서 최대한 넓은 영역을 차지하는 것

“인류는 수천년 동안 바둑 전술을 진화시켰는데
컴퓨터는 우리에게 인류가 완전히 틀렸다고 말한다.
그 누구도 바둑의 진리에 닿는 것조차 하지 못했다.”

– 柯洁 –

바둑이 어려운 이유



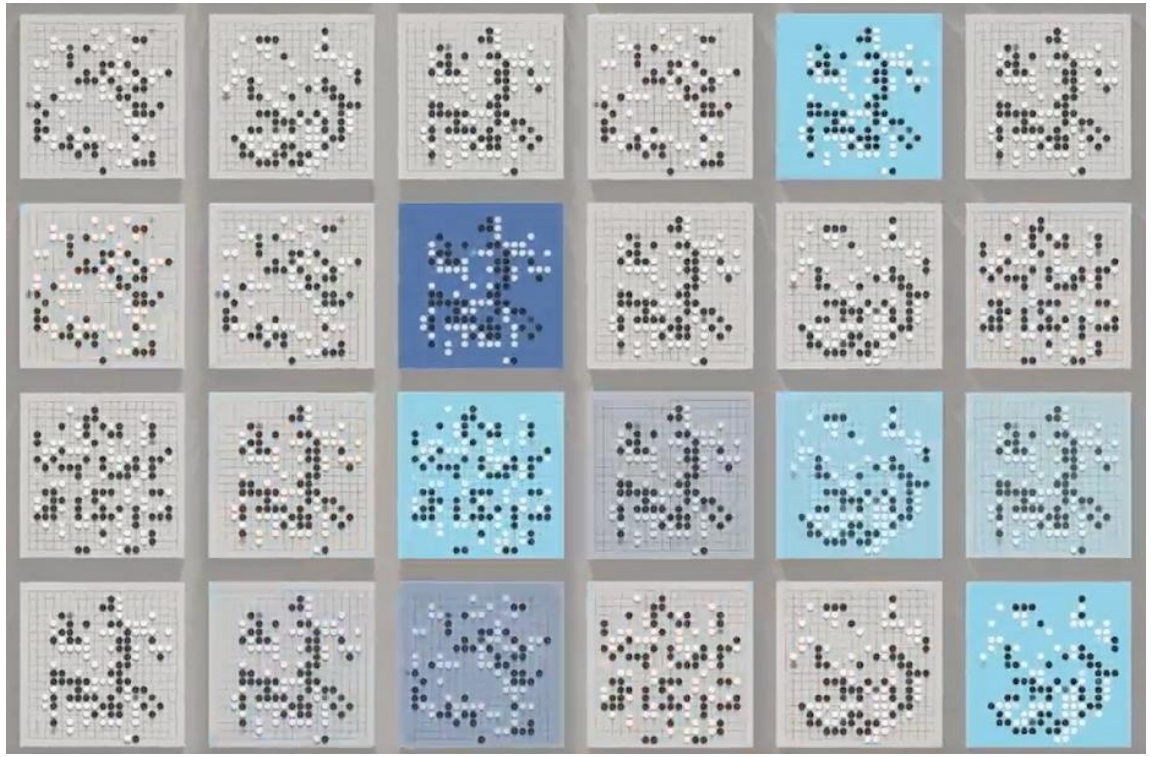
Vs.

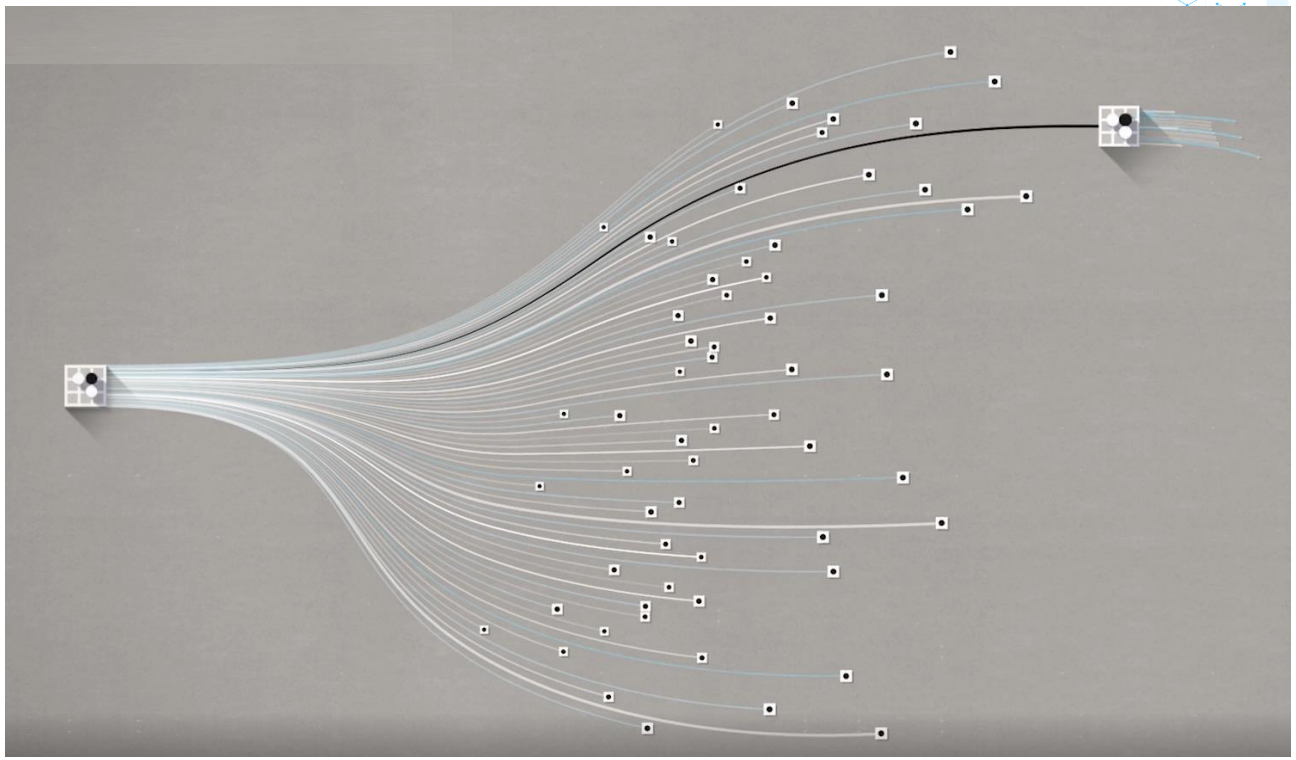
- ✓ 모든 돌의 가치가 똑같다.
- ✓ 이것이 체스와 다른점.
  체스의 평가함수는 각각의 말이 지닌 가치에 크게 의존
- ✓ 바둑에서 평가함수는 바둑판 위에 있는 돌에서
  중요한 패턴이 있는지를 식별

+

대국의 형세가 빠르게 변화하는 점

Part 6. 강화학습





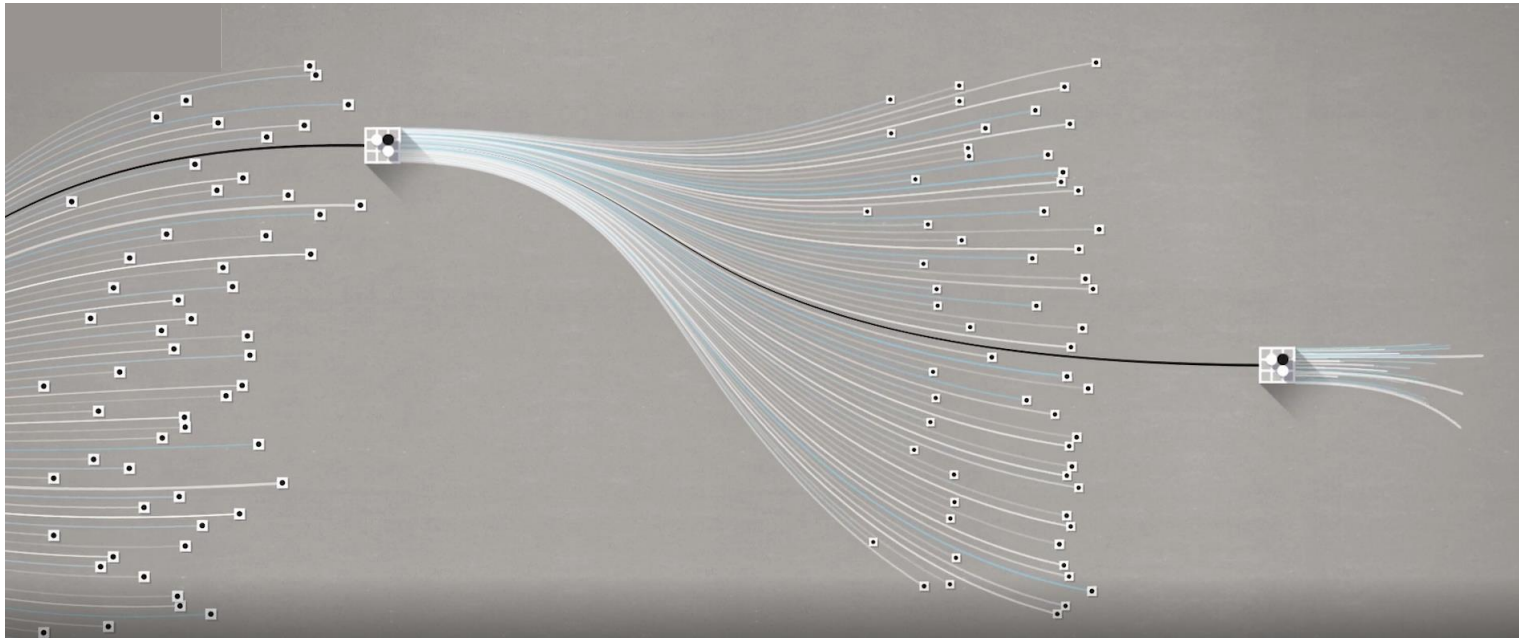
✓ 알파고는 대국에서 수를 둘 때마다 현재의 바둑판 상태로부터 수많은 가상 대국을 만들어 냅니다.

✓ 알파고는 메모리에 생성된 가상 대국을 마주하면서 이 대국이 끝날 때까지 검색 트리에서 하나의 경로를 계속 파고듭니다.

✓ 가상 개국을 치르고 나면 프로그램은 자신이 이겼는지 졌는지를 확인.

✓ 대국이 실제로 일어날 법한 대국이 아니라 할지라도, 중요한 것은 알파고가 이를 수천번 반복하여 어떤 수를 둬야 하는지 그 직관을 깨우침.

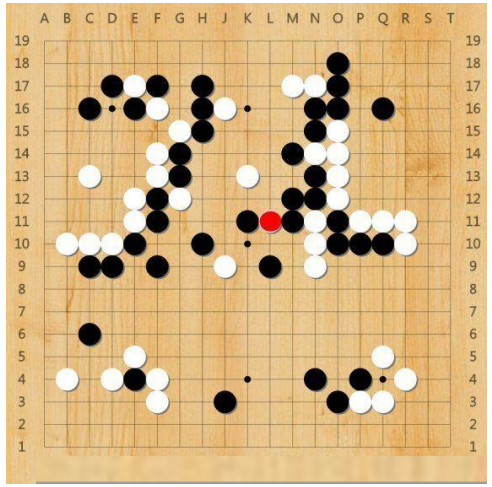
가상대국을 통해 누가 승리했는지를 파악



이 정보를 다시 검색 트리의 상단으로 보내 승리 대 패배 횟수를 기록

Part 6. 강화학습

AlphaGo



※ 알파고에서 까다로운 부분은 현실적인 대국을 재연하는 것

✓ 알파고는 차례가 돌아올 때마다 자신이 어떤 수를 둘지와 상대받이 어떤 수를 둘지 예측

✓ 실제 대국의 흐름을 예측할 때 무작위 수를 둔 대국에서 얻은 통계데이터는 큰 도움이 되지 않는다.

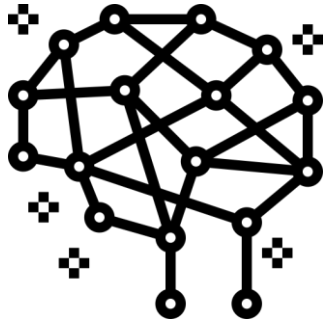
✓ 알파고에게 필요한 것은 프로기사가 둘 것 같은 수를 예측하는 방법

➢ 대국을 시뮬레이션해야 할 때마다 알파고는 바둑판에 가상의 돌을 놓아가며 대국이 어떻게 흘러갈지, 각자 어떤 수를 둘지 차례대로 예측.

➢ 가상 대국이 펼쳐지는 동안 알파고는 신경망을 사용해 바둑판에 놓인 가상의 돌로 다음 수를 결정.

영화『알파고』에서 알파고가 가상대국을 통한 수 예측을 수행하는 장면

Part 6. 강화학습



알파고의 수 예측 신경망
(move-prediction network)

여러개의 합성곱 신경망을 사용

딥마인드가 아타리 에이전트를 만들 때 사용했던 신경망과 매우 유사

✓ 오로지 바둑을 두기 위해 설계되었습니다

✓ 바둑에 최적화된 logic을 많이 보유
이러한 논리의 대부분은 딥마인드가 플레이어의 조작을 데이터로 요약하기 위해서 만들었던 특성과 비슷한 형식.

- 이러한 특성 평면중 몇몇은 바둑의 형세를 요약,
- 어떤 평면은 각 위치에 검은 돌이 있는지를 표시.
- 어떤 평면은 각 위치에 흰 돌이 있는지를 표시
- 이 외의 평면들은 해당 위치에 대한 전략적인 특성에 할애,

▶ 이 특성 평면들은 보통 좋은 수와 관련된 간단한 직관을 반영

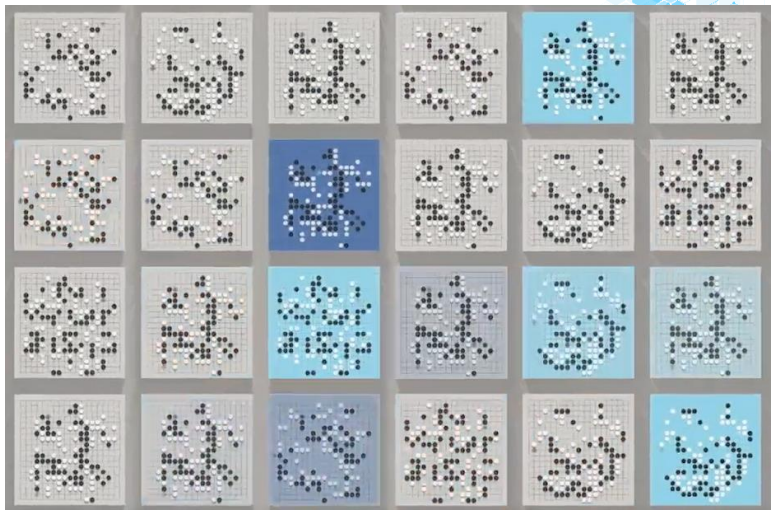
예를 들면

"이 돌 주면을 따라 얼마나 많은 빈칸이 있는가?"

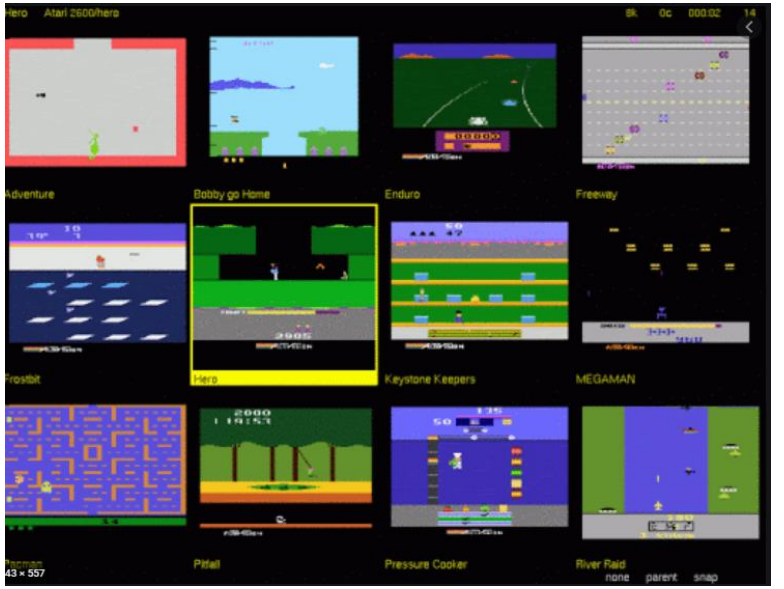
"이 돌이 놓이고 나서 자기 차례가 몇 번 지나갔는가?"

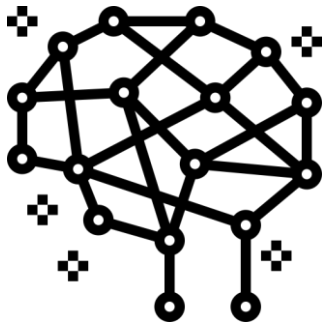
✓ 아타리 게임 신경망은 범용성을 염두

#바둑



#아타리 게임





알파고의 수 예측 신경망
(move-prediction network)

13개의 레이어
(합성곱 레이어)

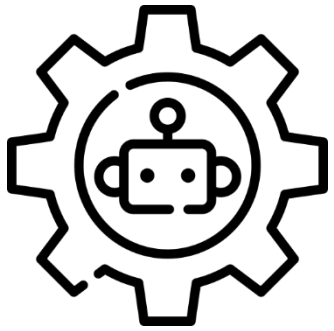
첫번째 합성곱 레이어는
200여개의 5 x 5 필터를 사용
(고유한 패턴 200개를 탐색)

n번째 레이어
(합성곱 레이어)

바둑돌의 복잡한 패턴을
더 많이 찾아내게 된다.

출력 레이어를 제외한
완전연결 레이어 X

- ✓ 모든 경우의 수에 대한 확률분포를 생성.
- ✓ 알파고는 이 신경망의 출력을 사용해 다음 수를 선택
- ✓ 수 예측 신경망이 확률이 높다고 판단한 수가 선택될 확률이 높아지게 됨



아타리 신경망

- ✓ 에이전트가 선택하는 행동에 따라 미래에 제공된 보상을 예측하는 업무
- ✓ 아타리 에이전트는 단순히 가장 높은 보상이 기대되는 행동을 선택합니다.

Part 6. 강화학습

AlphaGo

3,000만개의 수

✓ 한 대국에서 플레이어는 보통 250여 개의 수 중에서 하나를 선택해야 하는데, 알파고의 수 예측 신경망은 이러한 플레이어의 수를 57%라는 매우 높은 정확도로 예측.

✓ 알파고는 여전히 상대방이 어떤 수를 둘지 모른다는 불확실성을 안고 있음.

✓ 알파고가 가상 대국을 펼치는 대신 **실제 바둑 기사가 두는 수를 샘플링** 하면 알파고는 분명 좀 더 현실적으로 상대방의 수를 예측할 수 있을 것이고, **이러한 방식이 알파고가 좀 더 강해지는 길이라고 생각**

알파고의 예측이 정확해질 수록 느려지는 것을 확인

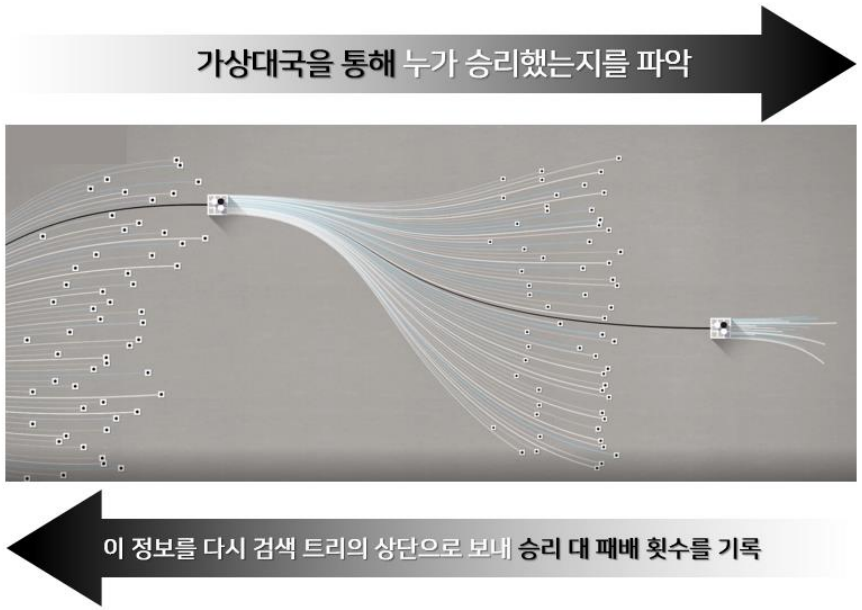
신경망이 전체를 평가하는데 **3 millisecond**

일반 대국을 시뮬레이션 하는 데 **평균 150수**

하나의 대국을 시뮬레이션 하는데 **0.5초**

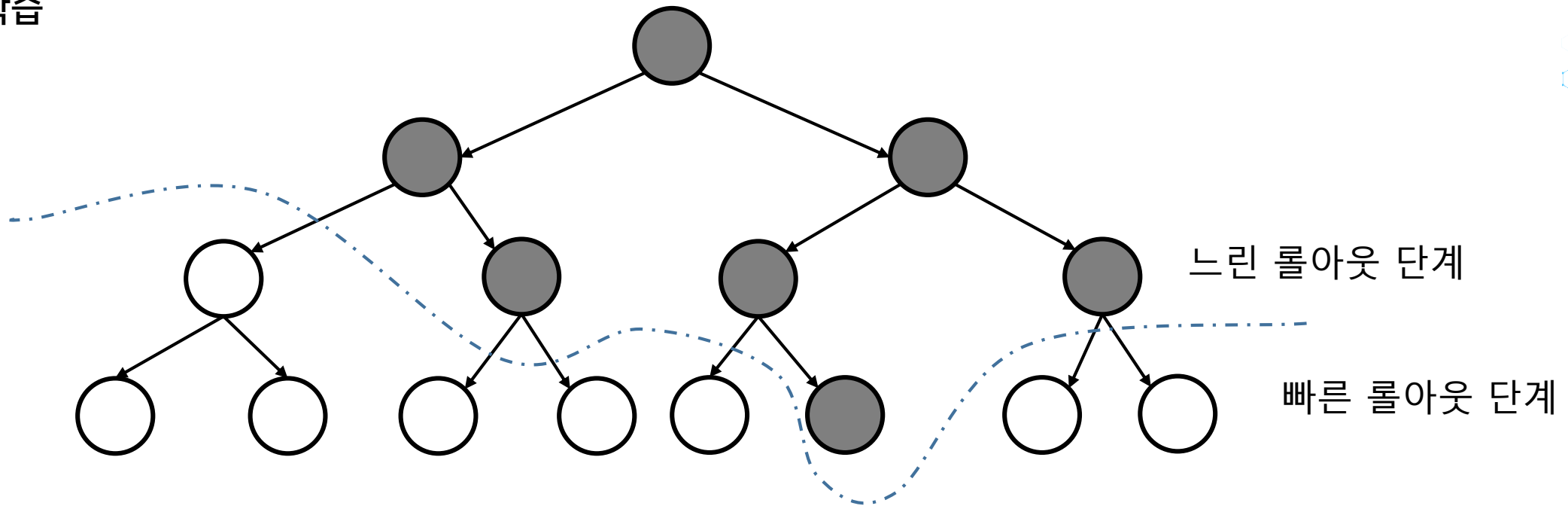
수 천 번의 대국이 필요하므로 이건
몇 시간을 아득히 넘는 수치

Part 6. 강화학습



수 예측 신경망이 불완전한 상태인 한 알파고가 수집하여 검색 트리의 최상단에 저장한

승패 통계로부터 알아낸 수가 정말로 최선의 수인지 보장할 수 없었다는 것.



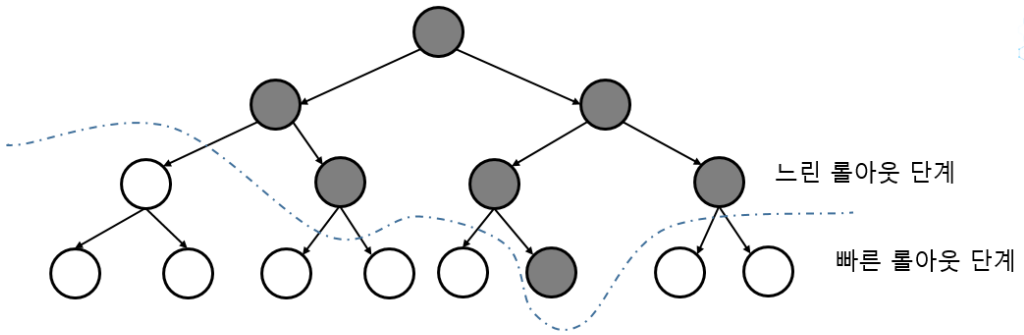
MCTS

(몬테카를로 트리 탐색기법)이 등장

이 기법은 알파고의 느린 수 예측문제와 악수를 두는 문제를 해결하는 방책.

이 방식은 게임을 시뮬레이션 할 때마다 서로 분리된 두 단계를 거침

MCTS



첫 번째 단계 '느린 롤아웃 단계'

알파고는 검색 트리의 상단에서 부터 분기를 따라 하행 즉 아래로 내려오며, '느린 수 예측 신경망'을 실행하여 알파고 또는 상대방이 미래에 둘 수의 확률을 찾은 뒤 자신은 어떠한 수를 둘지를 선택.

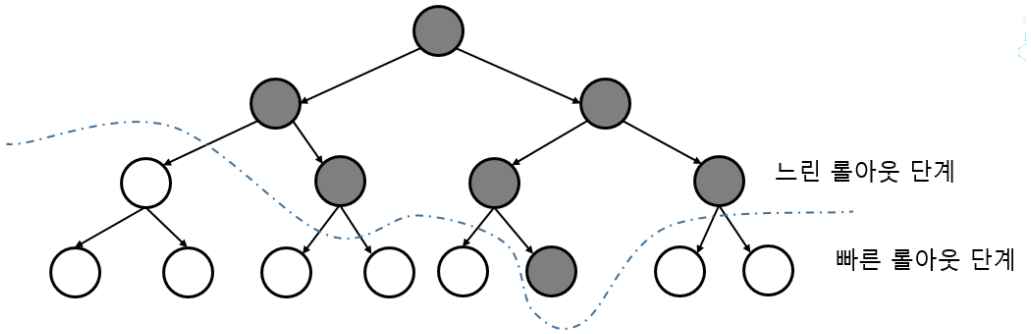
두번째 단계 '빠른 롤아웃 단계'

현재 판의 상태를 두가지 방식으로 평가

먼저 알파고가 현재 판의 진행 상황에서 승리할 확률을 예측하는 신경망의 평가함수로 현재 판을 평가.

이와 동시에 알파고는 '빠른 롤아웃 단계'를 통해 이후 대국이 어떻게 흘러갈지 별도로 시뮬레이션

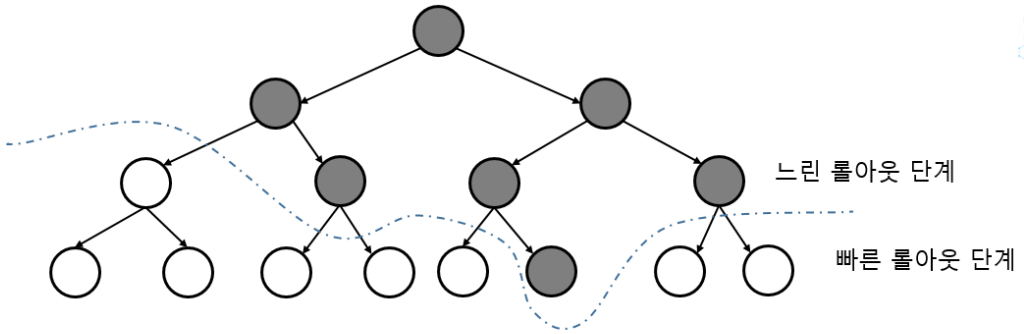
MCTS



첫 번째 단계 '느린 롤아웃 단계'

- ✓ 빠른 수 예측 신경망은 느린 수 예측 신경망과 같은 구조로 되어 있으나, 계산하는데 시간이 걸리는 몇가지 입력 특성을 생략.
- ✓ 신경망은 200만분의 1초 이내에 다음 수를 예측할 수 있었으나 정확도가 절반정도 떨어지게 됨.
- ✓ 이러한 평가함수의 두 부분으로 분할하는 방식을 통해 알파고의 속도 문제를 해결

MCTS



MCTS의 상단에서 다음수를 선택하는 방식으로서 해결

앞서 이야기 했듯 알파고의 수 예측 신경망이 특정한 상황속에서 계속 악수를 둔다고 하더라도

결국 알파고는 이 방식으로 최적의 수를 배울수 있게 된다는 것.

최종적으로 알파고는 시뮬레이션 결과에서 어떤 수가 최선인지를 학습