

# AlphaGo

작동원리

발표자: 문승환

PhD student

Language Technologies Institute, School of Computer Science  
Carnegie Mellon University

# 알파고 vs 유럽챔피언 (판 후이 2단)



2015년 10월 5일 – 9일

## <공식경기>

- 제한시간 1시간, 30초 초읽기 3회
- 5:0 알파고 승리 (불계승 4번)

# 알파고 vs 세계챔피언 (이세돌 9단)



인간과 컴퓨터의 자존심을 건 '세기의 대결'

2016년 3월 9일 – 15일

<공식경기>

- 제한시간 2시간, 1분 초읽기 3회

서울 광화문 포시즌스 호텔

# 이세돌



사진 출처: [매일경제 2013/04](#)



사진 출처: [바둑 TV](#)

“자신이 없어요. 질 자신이요”



"아, 싸울만 해서 싸워요. 수가 보이는데 어쩌란 말이에요."



"불리하다보니 이기자는 생각없이 대충 뒀는데 이겼네요."

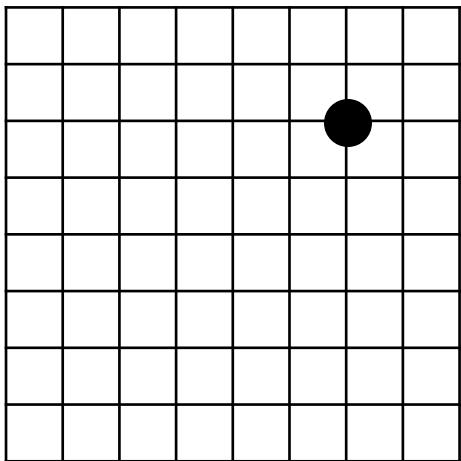
- 구리 九단에게 대역전승을 거둔  
직후의 인터뷰

# 바둑 인공지능?



# 바둑 인공지능? 정의하자면:

$d = 1$

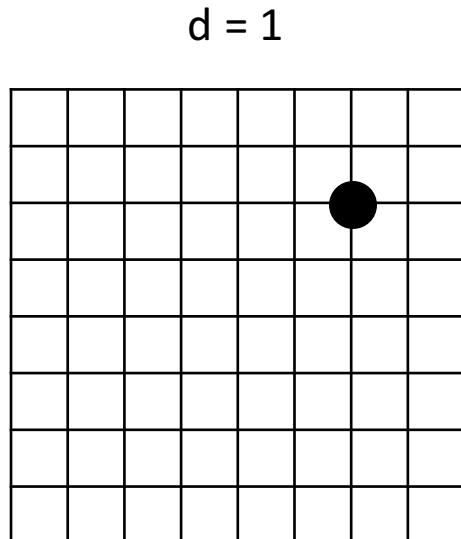


$s$  (state)

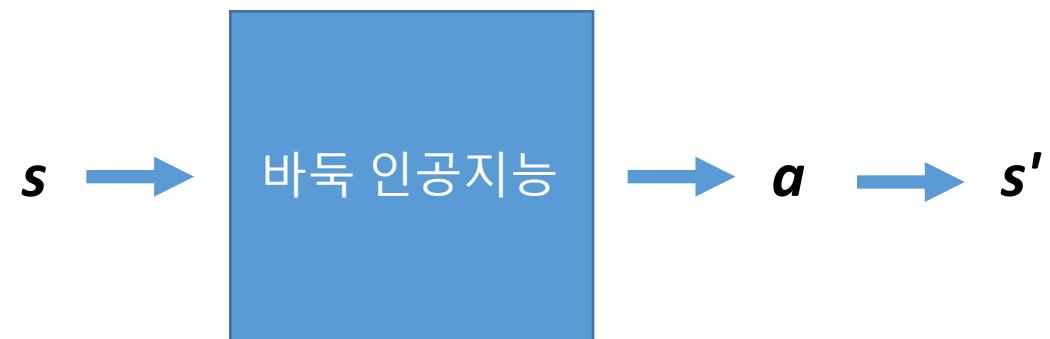
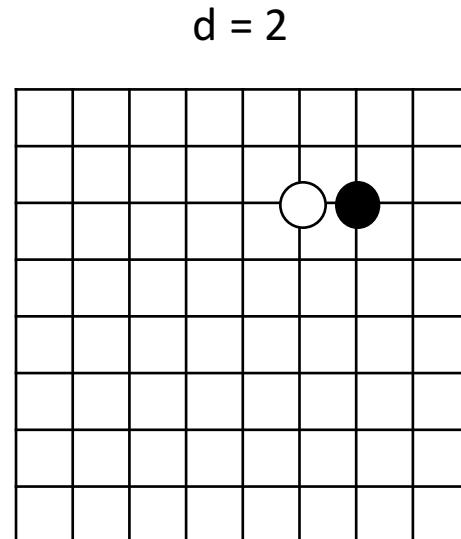
$$= \begin{array}{c} 000000000 \\ 000000000 \\ 000000\textcolor{red}{1}00 \\ 000000000 \\ 000000000 \\ 000000000 \\ 000000000 \\ 000000000 \end{array}$$

(예를 들면 이런 식으로 행렬로 표현)

# 바둑 인공지능? 정의하자면:

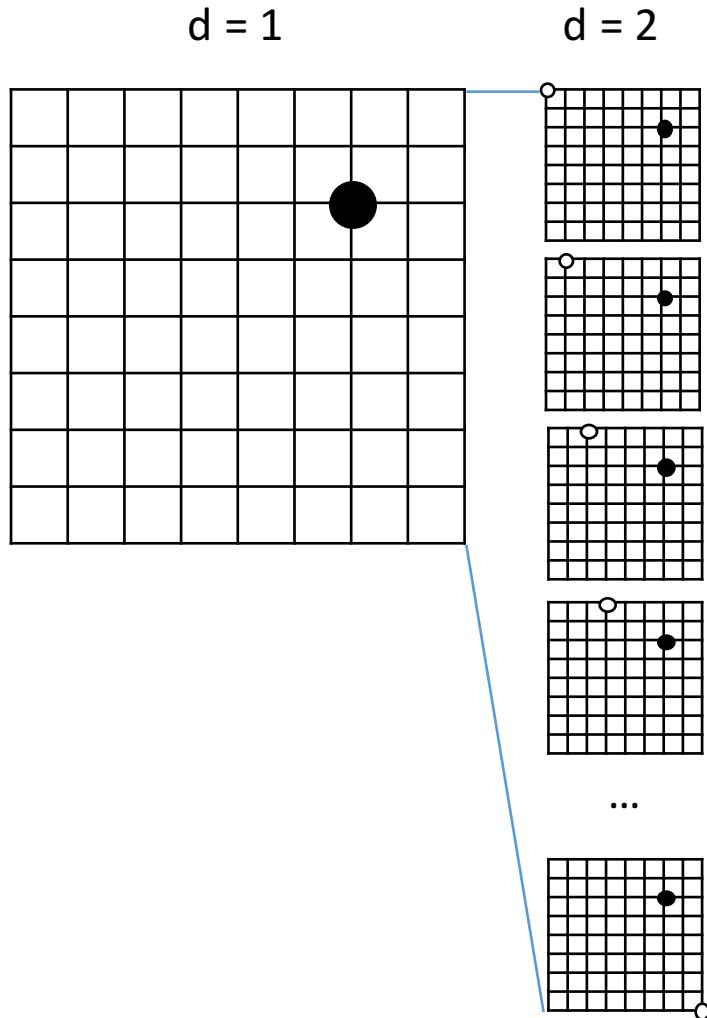


$s$  (state)



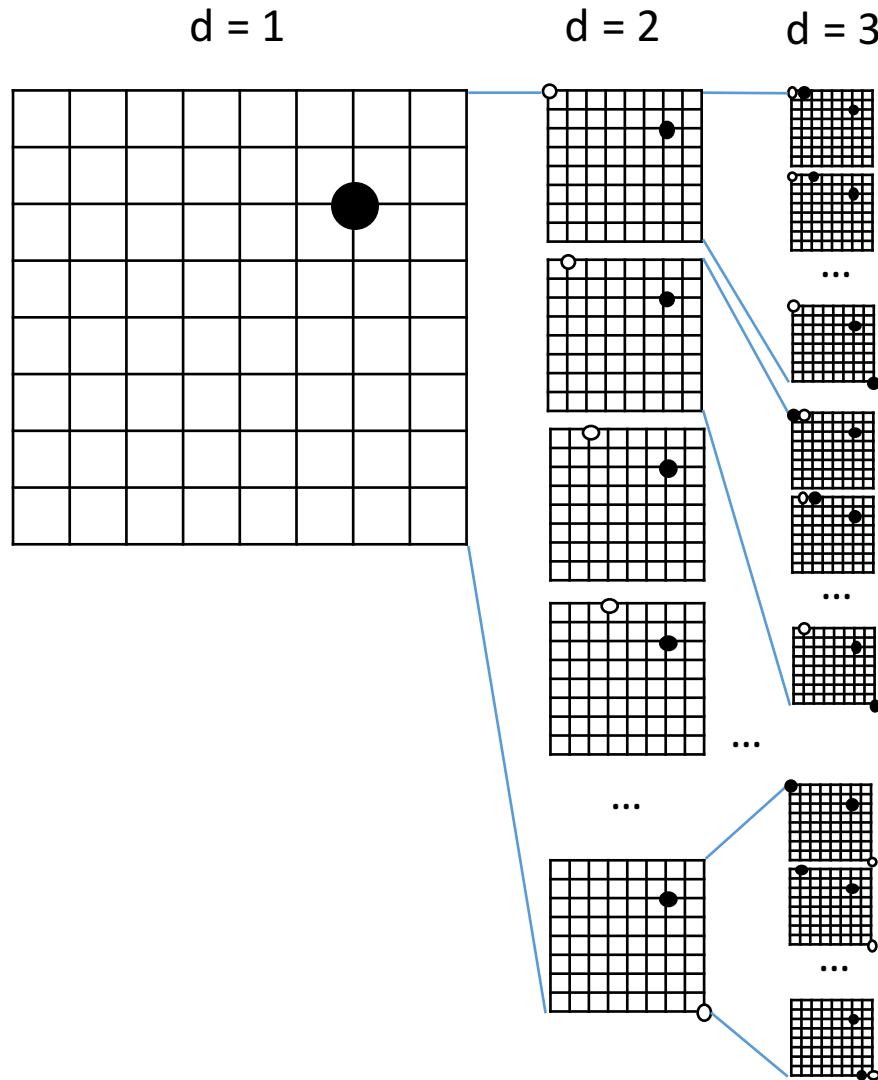
Given  $s$ , pick the best  $a$

# 바둑 인공지능? 이렇게 만들어 보면?

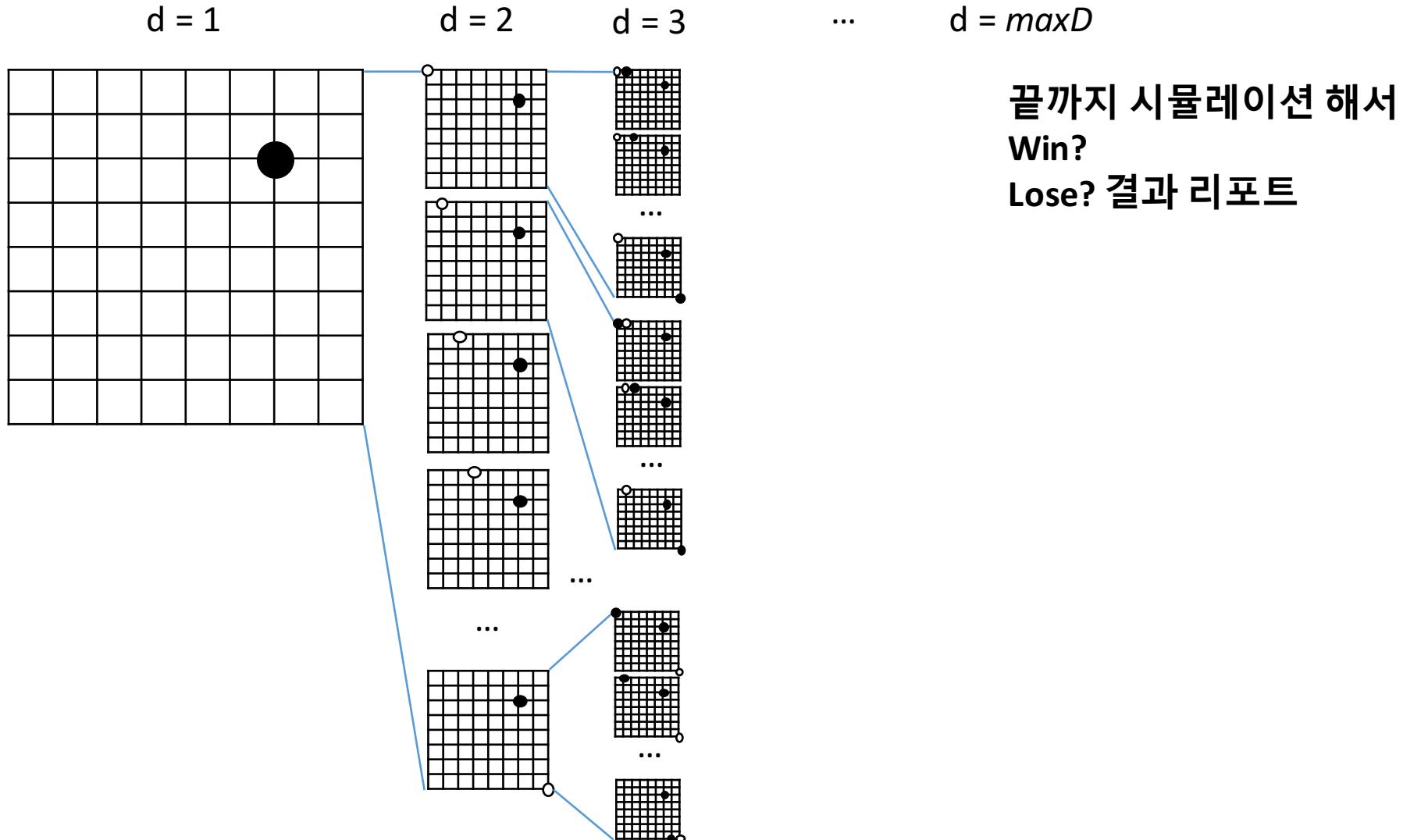


모든 경우의 수를 시뮬레이션

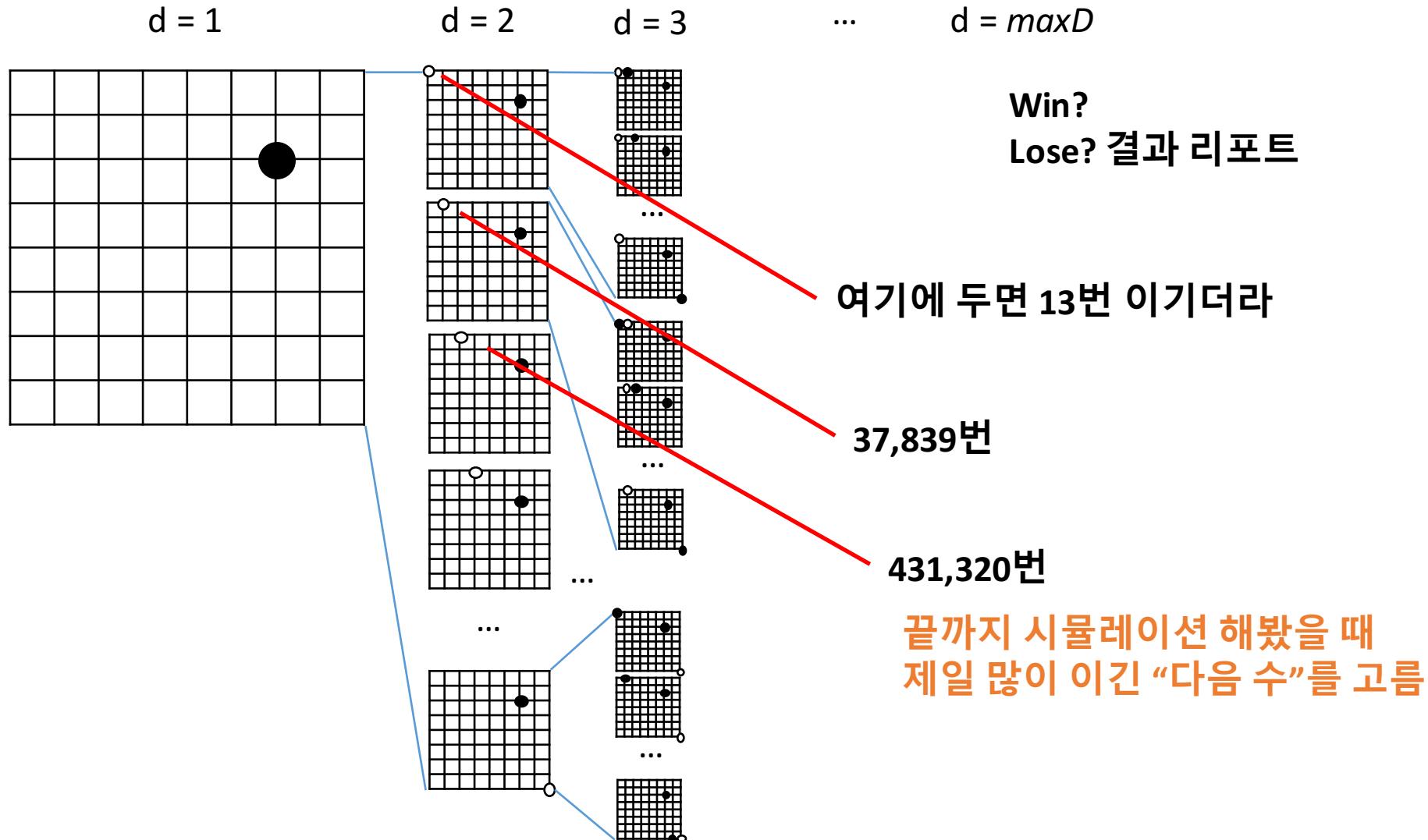
# 바둑 인공지능? 이렇게 만들어 보면?



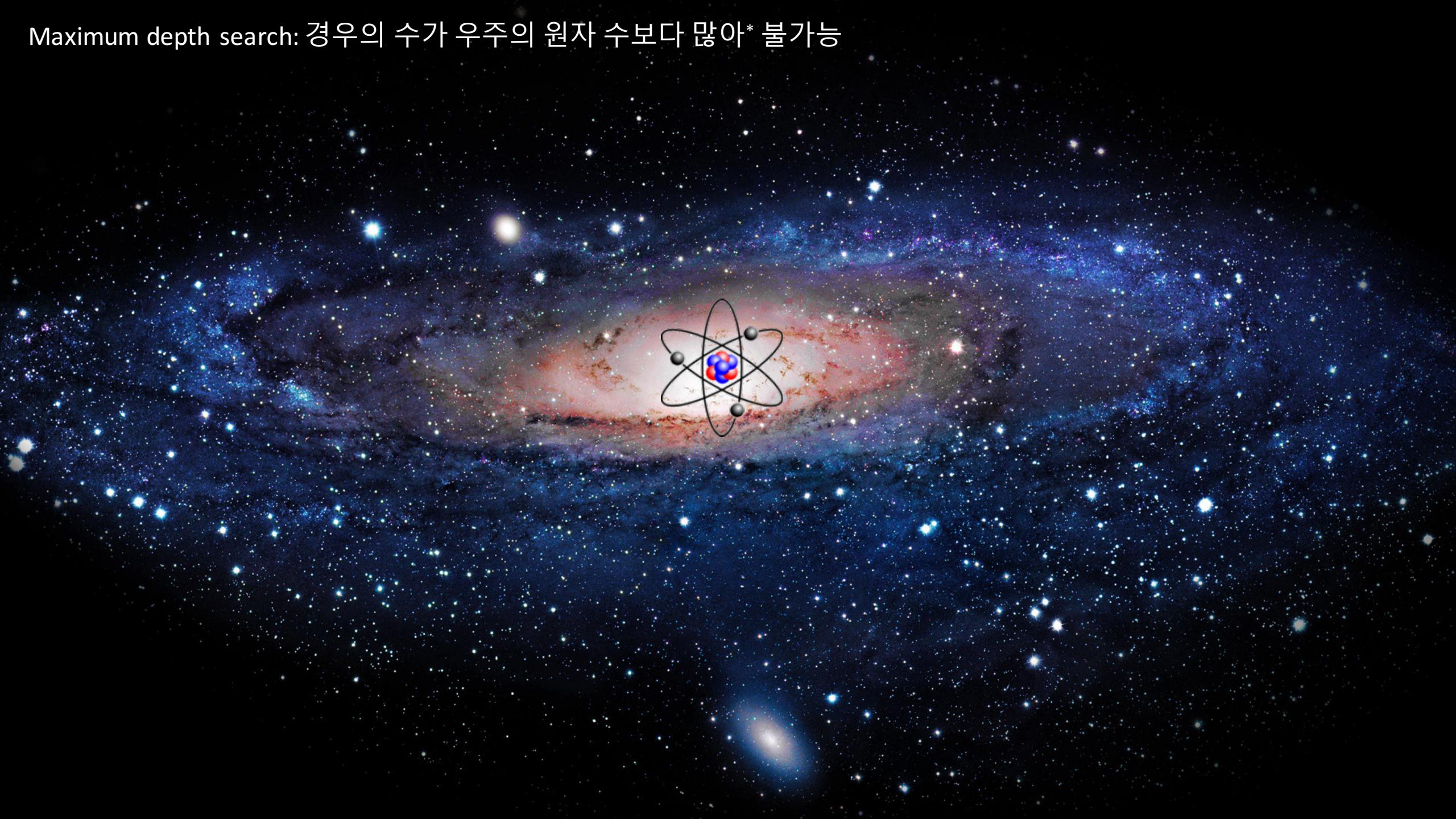
# 바둑 인공지능? 이렇게 만들어 보면?



# 바둑 인공지능? 이렇게 만들어 보면?



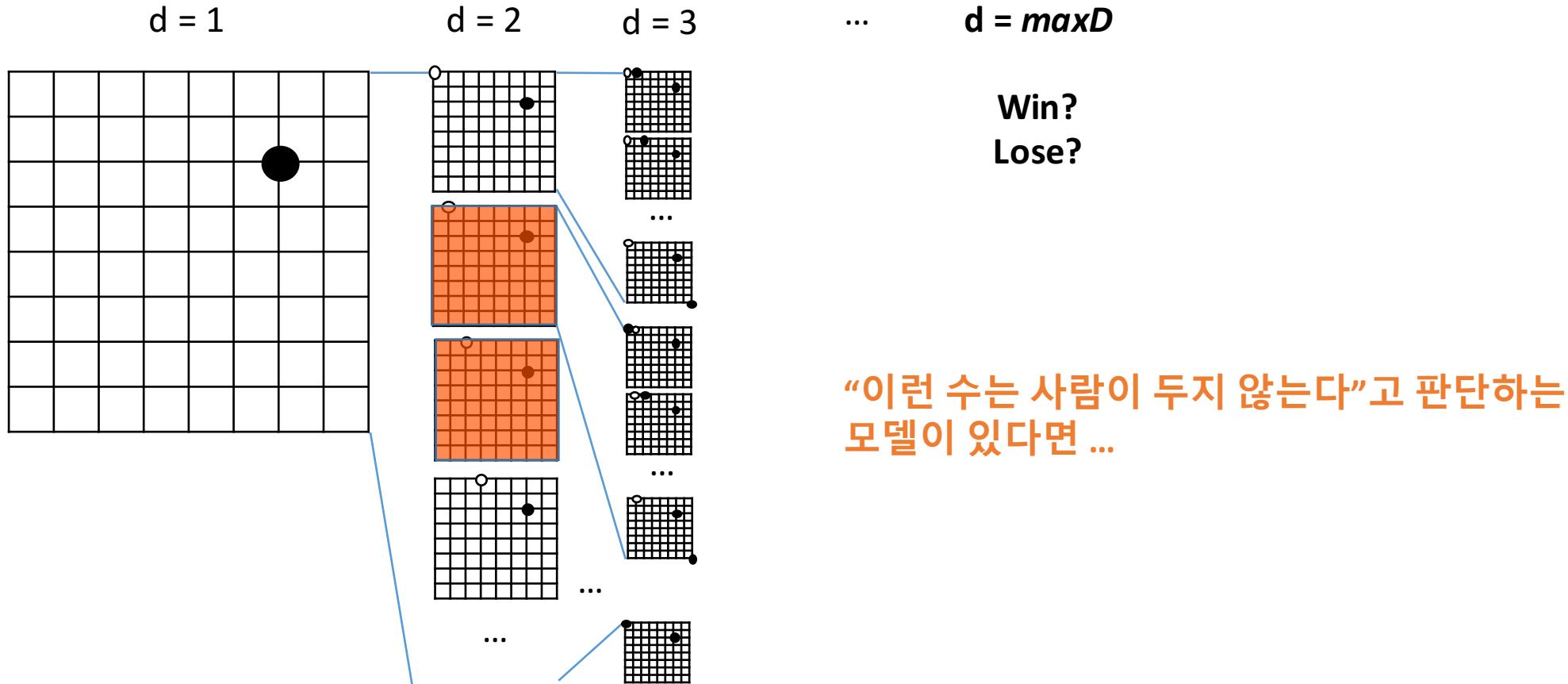
Maximum depth search: 경우의 수가 우주의 원자 수보다 많아\* 불가능



# 핵심: 경우의 수 (Search Space) 줄이기

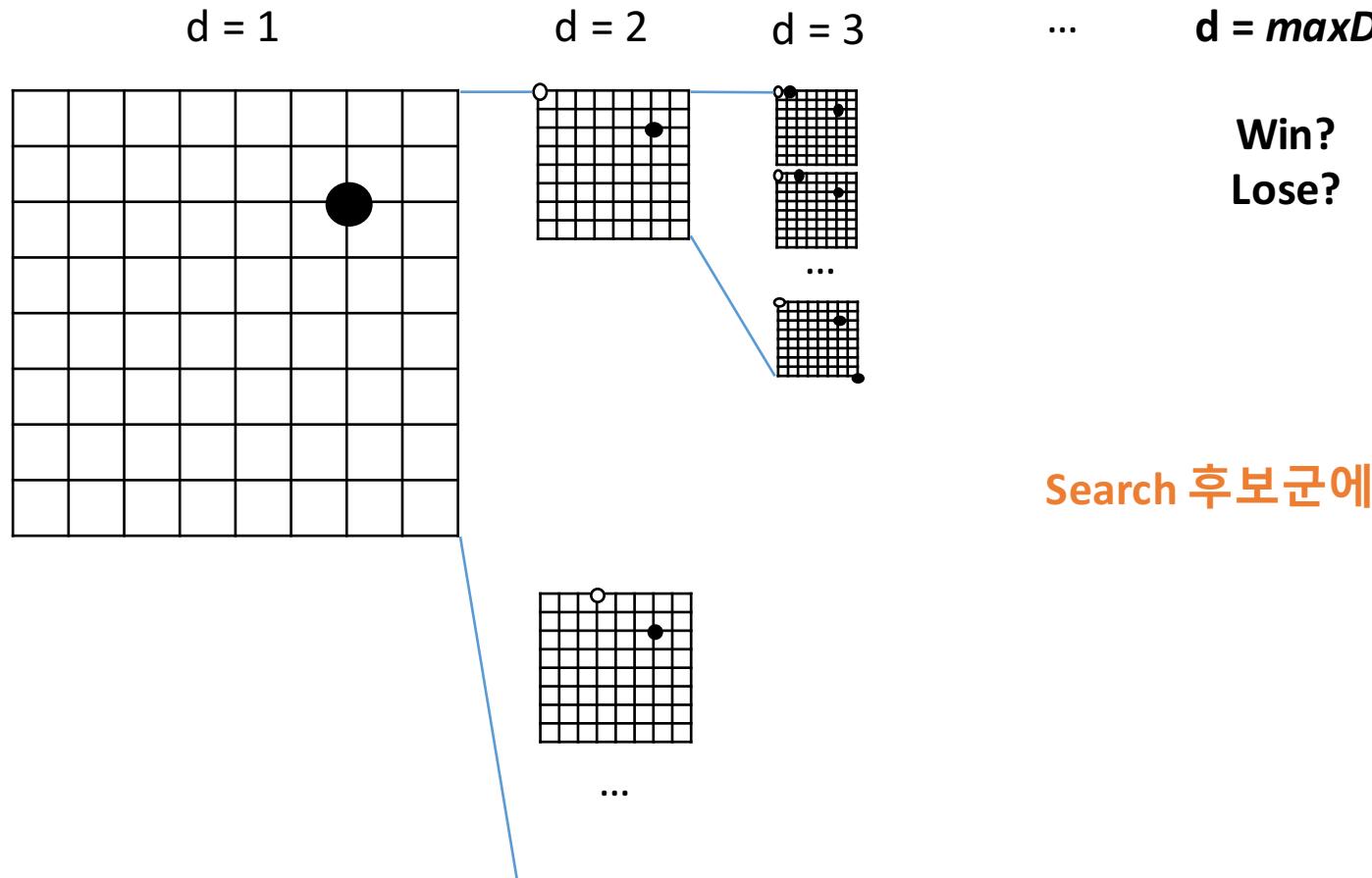
# 경우의 수 (Search Space) 줄이기

## 1. “action” 후보군 줄이기 (Breadth Reduction)



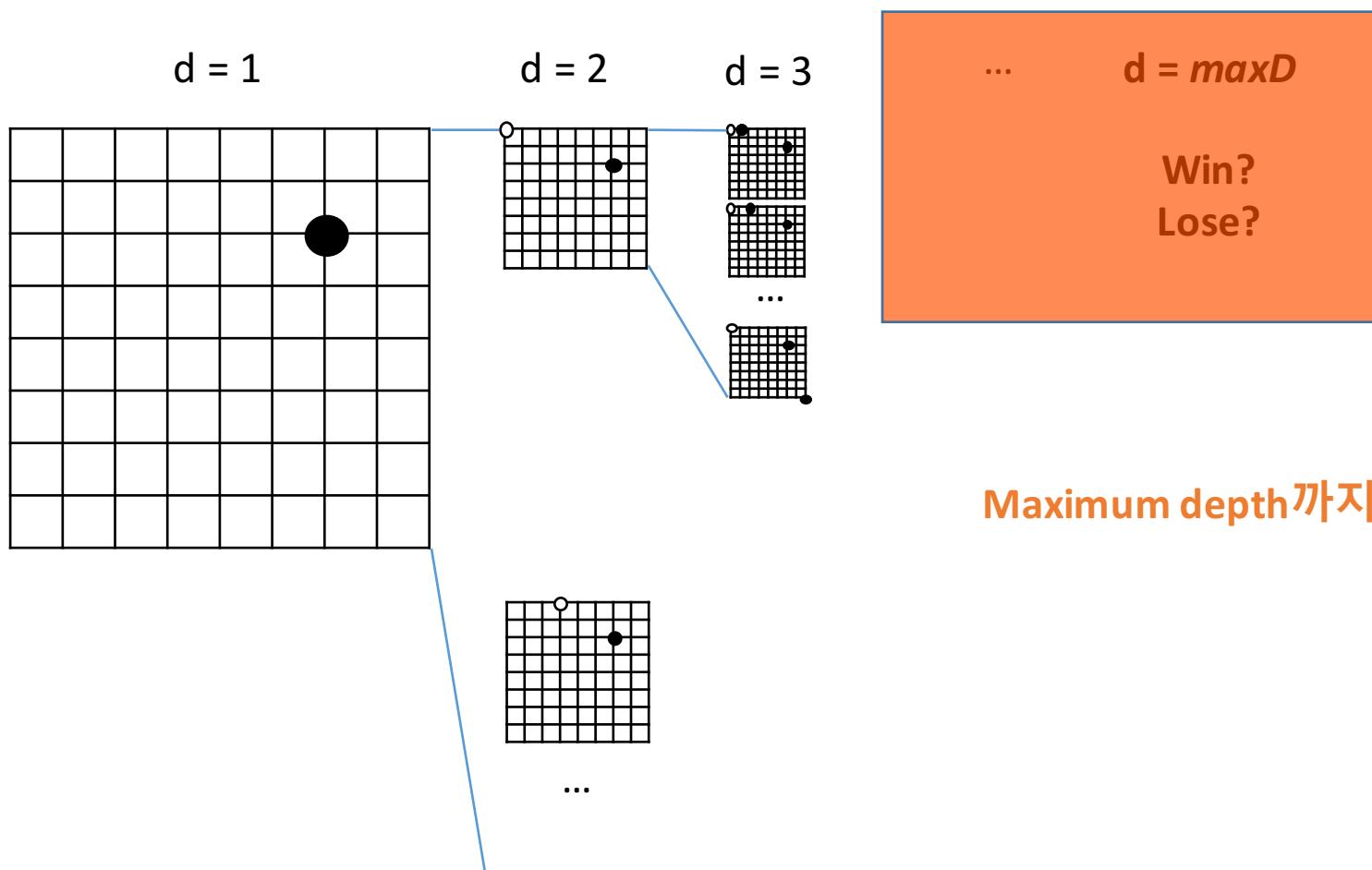
# 경우의 수 (Search Space) 줄이기

## 1. “action” 후보군 줄이기 (Breadth Reduction)



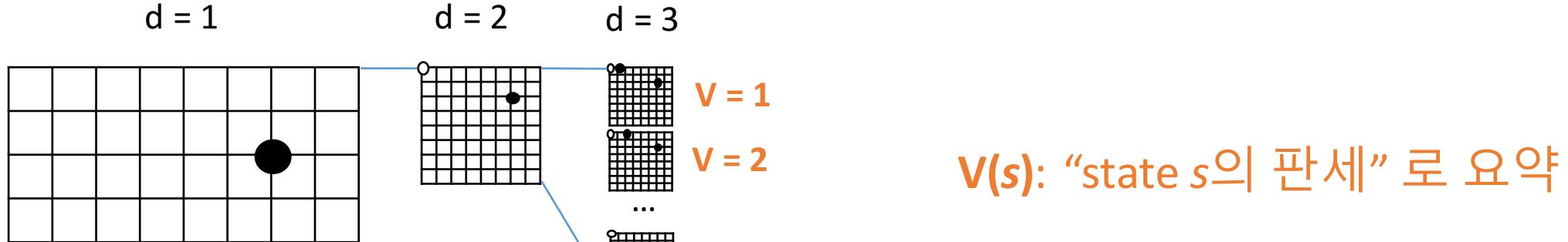
# 경우의 수 (Search Space) 줄이기

## 2. 결과 더 빨리 예측하기 (Depth Reduction)

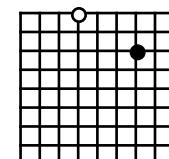


# 경우의 수 (Search Space) 줄이기

## 2. 결과 더 빨리 예측하기 (Depth Reduction)



$V(s)$ : “state  $s$ 의 판세”로 요약



# 경우의 수 (Search Space) 줄이기

1. “action” 후보군 줄이기 (Breadth Reduction)
2. 결과 더 빨리 예측하기 / 판세 평가하기 (Depth Reduction)

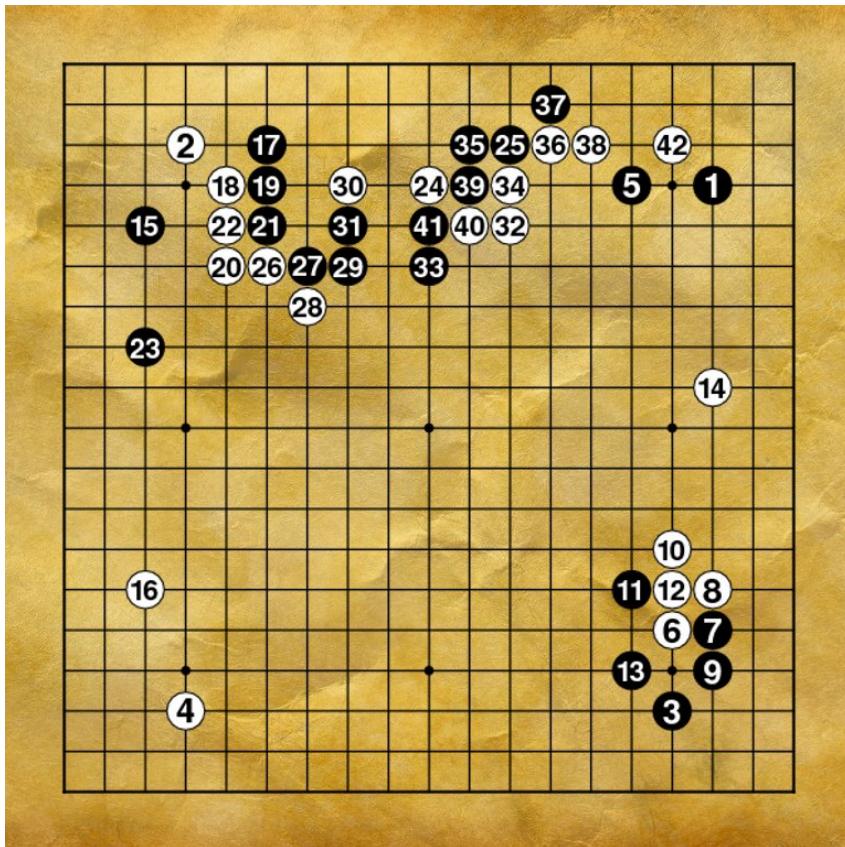
# 1. “action” 후보군 줄이기

Learning:  $P(\text{next action} \mid \text{current state})$

$$= P(a \mid s)$$

# 1. “action” 후보군 줄이기

## (1) 프로 바둑기사 따라하기 (supervised learning)

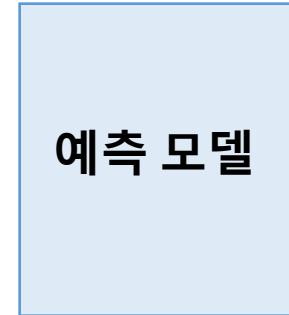


현재 판

s1

s2

s3



다음 판

s2

s3

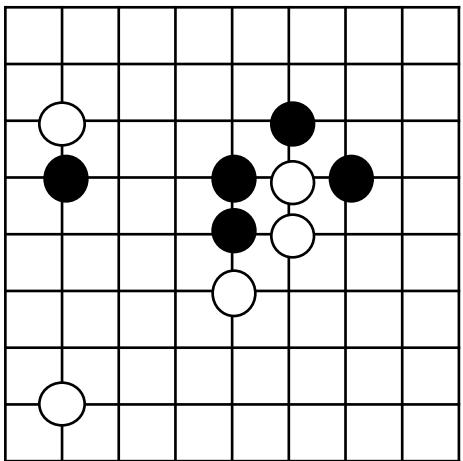
s4

**Data:** 온라인 바둑 고수 (5~9단)  
기보 16만 개, 착점 3000만 개

# 1. “action” 후보군 줄이기

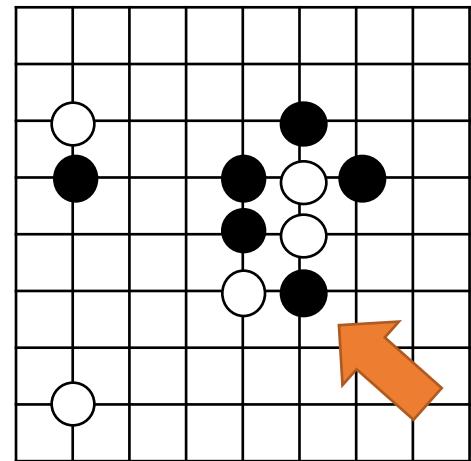
(1) 프로 바둑기사 따라하기 (supervised learning)

현재 판



예측 모델

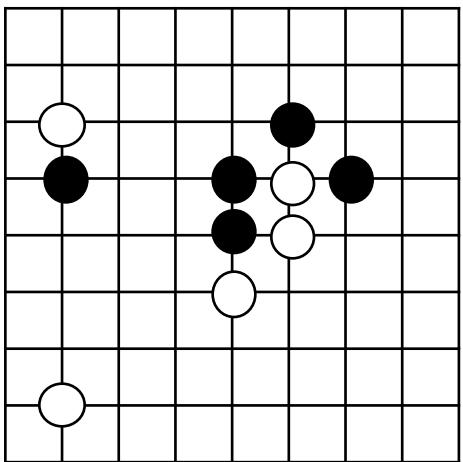
다음 판



# 1. “action” 후보군 줄이기

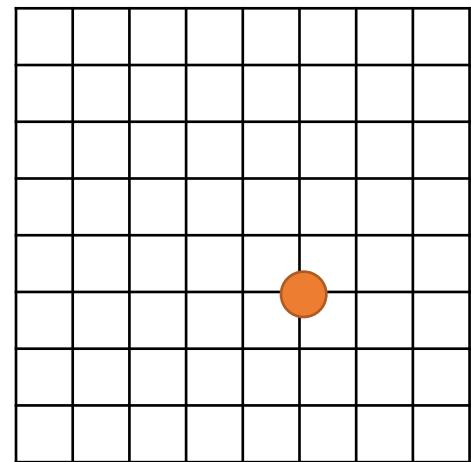
(1) 프로 바둑기사 따라하기 (supervised learning)

현재 판



예측 모델

다음 액션



# 1. “action” 후보군 줄이기

(1) 프로 바둑기사 따라하기 (supervised learning)

현재 판

00 000 0000  
00 000 **1**000  
0 **-1**00 **1**-**1**00  
0 **1** 00 **1**-**1**000  
00 00 **-1**0000  
00 000 0000  
0 **-1**000 0000  
00 000 0000

다음 액션

000000000  
000000000  
000000000  
000000000  
000000000  
00000 **1**000  
000000000  
000000000  
000000000

예측 모델

*s*

$f: s \rightarrow a$

*a*

# 1. “action” 후보군 줄이기

## (1) 프로 바둑기사 따라하기 (supervised learning)

현재 판

00 000 0000  
00 000 **1**000  
**0**-100 **1**-1**1**00  
**0**1 00 **1**-1000  
00 00 **-1**0000  
00 000 0000  
**0**-1000 0000  
00 000 0000

예측 모델

$s$

$g: s \rightarrow p(a|s)$

000000 000  
000000 000  
000000 000  
000000.20.100  
**000000.4** 0.200  
000000.1 000  
000000 000  
000000 000

$p(a|s)$

$\text{argmax}$

$a$

다음 액션

000000000  
000000000  
000000000  
000000000  
000000000  
**000000000**  
**0**00000000  
000000000  
000000000

# 1. “action” 후보군 줄이기

## (1) 프로 바둑기사 따라하기 (supervised learning)

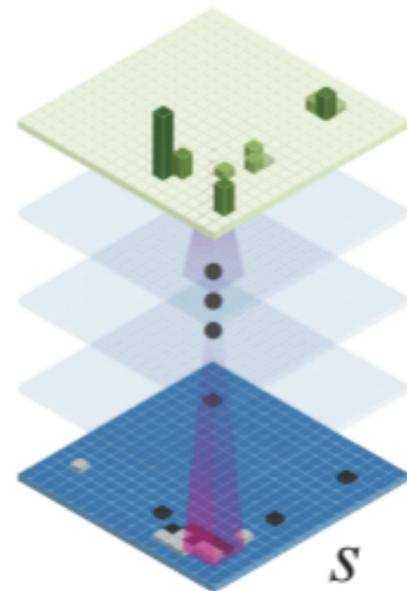
현재 판

00 000 0000  
00 000 **1**000  
0 **-1**00 **1**-**1**000  
0 **1** 00 **1**-**1**000  
00 00 **-1**0000  
00 000 0000  
0 **-1**000 0000  
00 000 0000

예측 모델

$s$

$g: s \rightarrow p(a|s)$



$p(a|s)$

다음 액션

000000000  
000000000  
000000000  
000000000  
00000 **1**000  
000000000  
000000000  
000000000



$\text{argmax}$

$a$

# 1. “action” 후보군 줄이기

(1) 프로 바둑기사 따라하기 (supervised learning)

현재 판

00 000 0000  
00 000 1000  
0 -100 1-1100  
0 1 00 1-1000  
00 00 -10000  
00 000 0000  
0 -1000 0000  
00 000 0000

Deep Learning  
(13 Layer CNN)

$s$

$g: s \rightarrow p(a|s)$

000000 000  
000000 000  
000000 000  
000000.20.100  
000000.4 0.200  
000000.1 000  
000000 000  
000000 000

$p(a|s)$

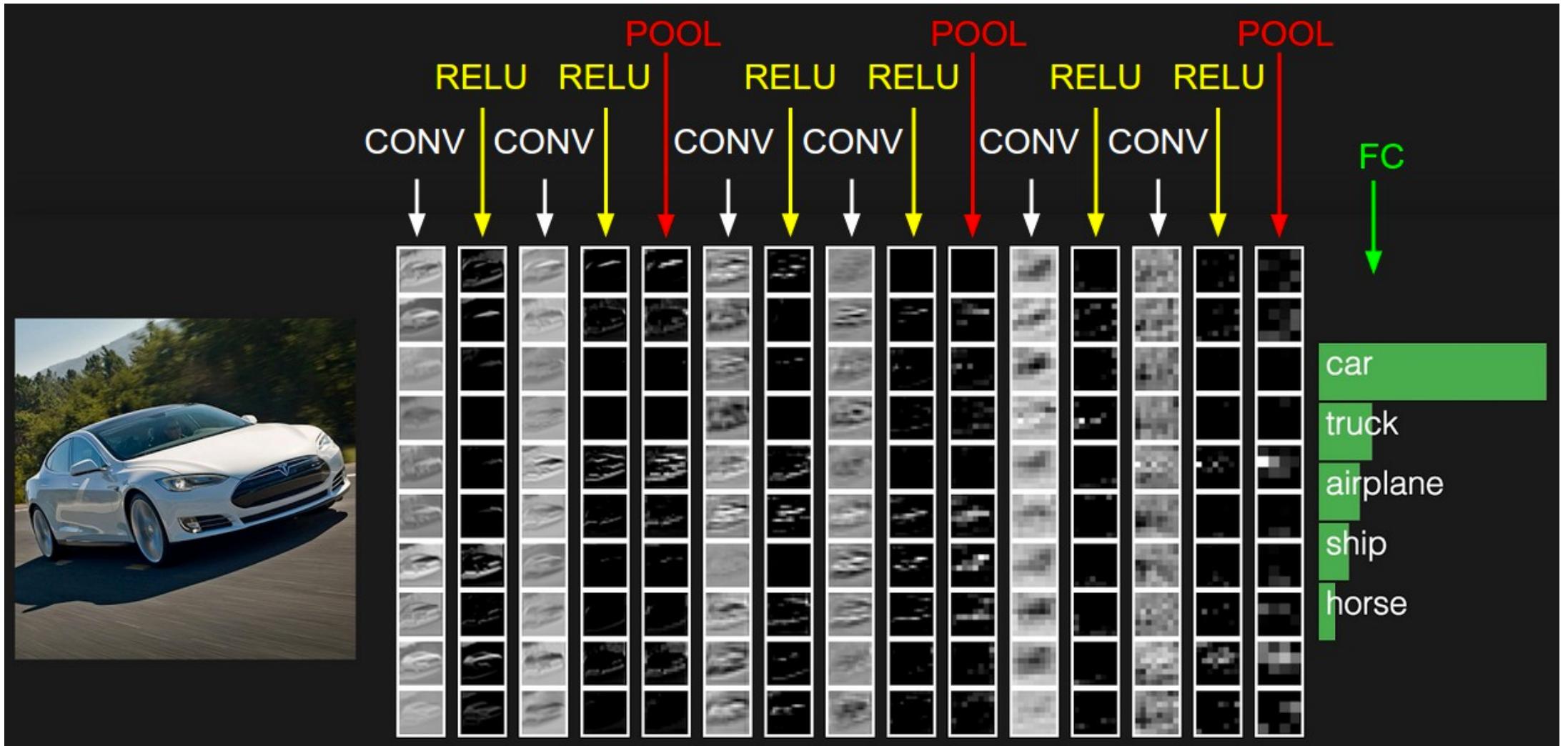
$\text{argmax}$

$a$

다음 액션

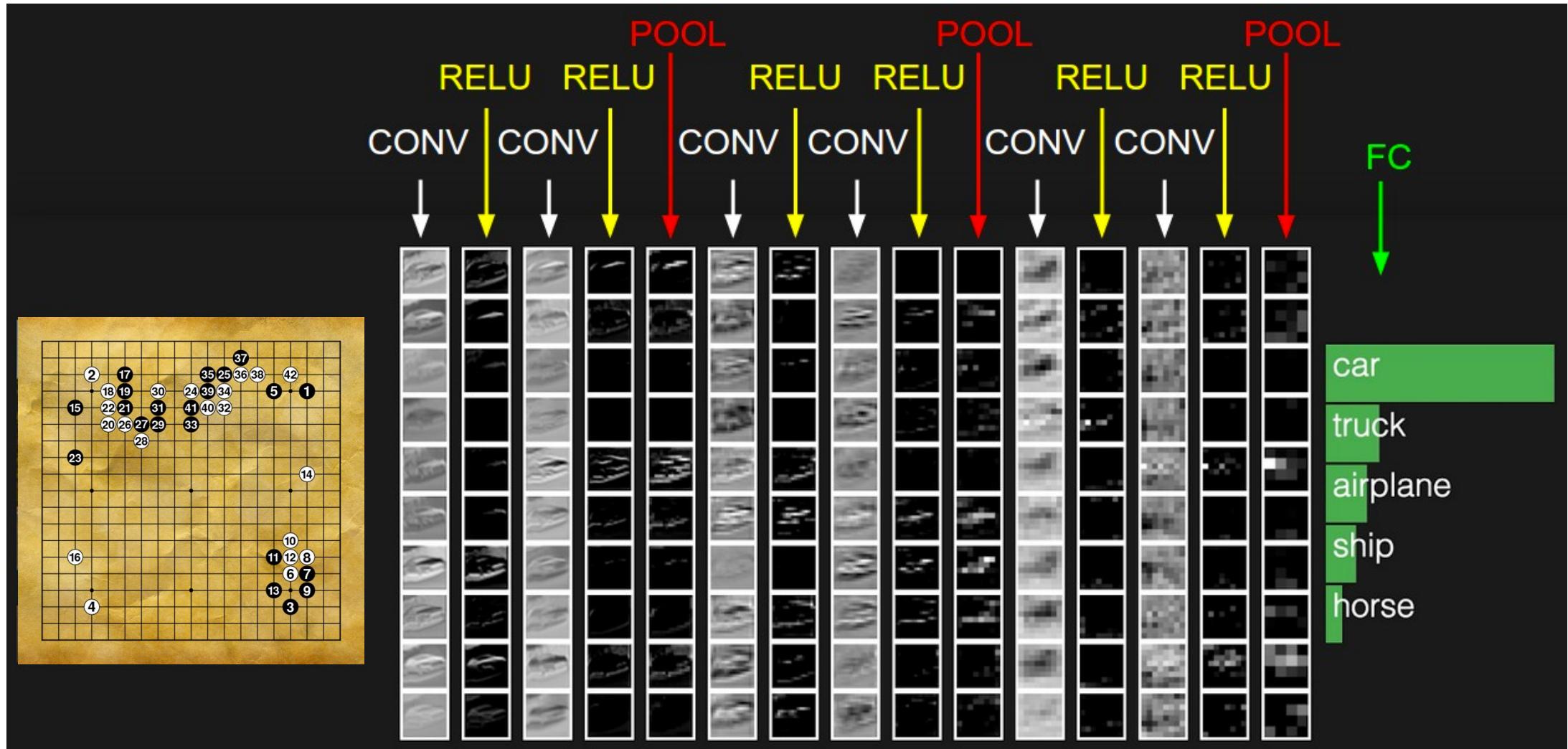
000000000  
000000000  
000000000  
000000000  
000000000  
00000 1000  
000000000  
000000000  
000000000

# Convolutional Neural Network (CNN)



CNN은 레이어별로 input image를 추상화 시켜 Image Recognition을 굉장히 잘함

# Convolutional Neural Network (CNN)



이걸 바둑의 판세를 읽는 데에 사용

**바둑:** 추상화하는 능력이 중요

**CNN:** 추상화하는 능력이 뛰어난 모델

바둑 두는 Task와 CNN의 장점이 맞물린 경우

# Deep Learning

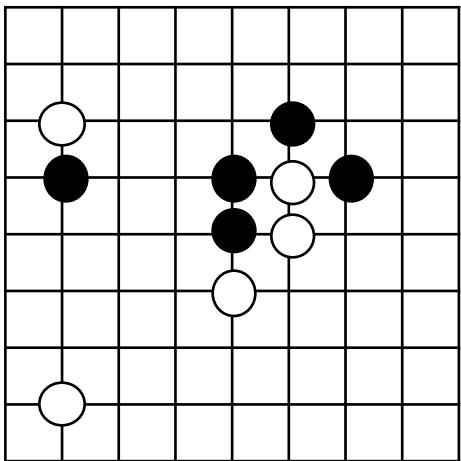
**~= Representation Learning**

→ 단을 쌓아 올라갈수록 추상화된  
feature를 익힘

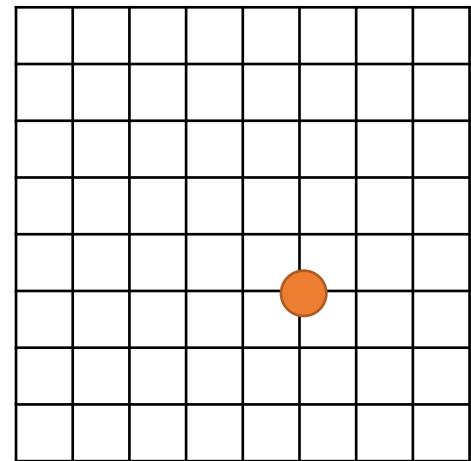
# 1. “action” 후보군 줄이기

(1) 프로 바둑기사 따라하기 (supervised learning)

현재 판



다음 액션



**프로기사 흥내내는 모델  
(w/ CNN)**

**Training:**  $\Delta\sigma \propto \frac{\partial \log p_\sigma(a|s)}{\partial \sigma}$

# 1. “action” 후보군 줄이기

(2) 스스로 발전하기 (reinforcement learning)

프로기사  
흉내내는 모델  
(w/ CNN)

vs

프로기사  
흉내내는 모델  
(w/ CNN)



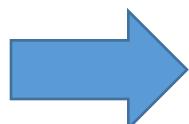
# 1. “action” 후보군 줄이기

(2) 스스로 발전하기 (reinforcement learning)

프로기사  
흉내내는 모델  
(w/ CNN)

vs

프로기사  
흉내내는 모델  
(w/ CNN)

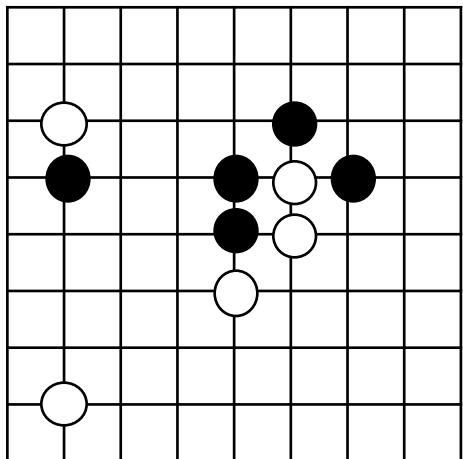


**Return:** 대국 기보, 승/패자 정보

# 1. “action” 후보군 줄이기

(2) 스스로 발전하기 (reinforcement learning)

어느 판



승/패

프로기사 흉내내는 모델  
(w/ CNN)

패

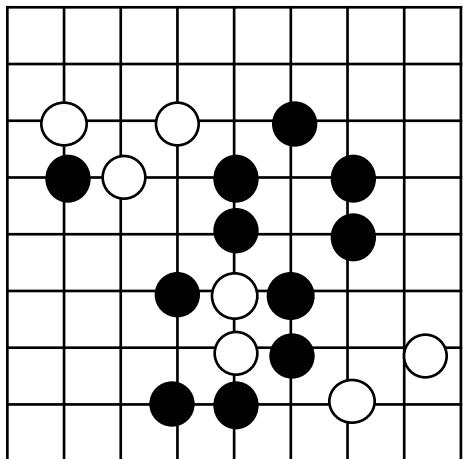
$z = -1$

**Training:**  $\Delta\rho \propto \frac{\partial \log p_\rho(a_t|s_t)}{\partial \rho} z_t$

# 1. “action” 후보군 줄이기

(2) 스스로 발전하기 (reinforcement learning)

어느 판



승/패

승

$z = +1$

프로기사 흉내내는 모델  
(w/ CNN)

**Training:**  $\Delta\rho \propto \frac{\partial \log p_\rho(a_t|s_t)}{\partial \rho} z_t$

# 1. “action” 후보군 줄이기

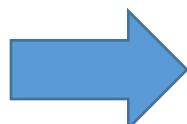
(2) 스스로 발전하기 (reinforcement learning)

업데이트 모델  
ver 1.1

vs

업데이트 모델  
ver 1.3

프로기사 흉내내는 모델과 똑같은 topology, 업데이트 된 parameters를 사용



**Return:** 대국 기보, 승/패자 정보

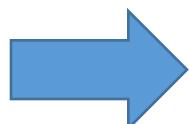
# 1. “action” 후보군 줄이기

(2) 스스로 발전하기 (reinforcement learning)

업데이트 모델  
ver 1.3

vs

업데이트 모델  
ver 1.7



**Return:** 대국 기보, 승/패자 정보

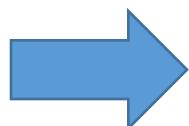
# 1. “action” 후보군 줄이기

(2) 스스로 발전하기 (reinforcement learning)

업데이트 모델  
ver 1.5

vs

업데이트 모델  
ver 2.0



**Return:** 대국 기보, 승/패자 정보

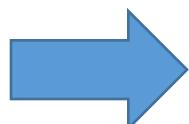
# 1. “action” 후보군 줄이기

(2) 스스로 발전하기 (reinforcement learning)

업데이트 모델  
ver 3204.1

vs

업데이트 모델  
ver 46235.2



**Return:** 대국 기보, 승/패자 정보

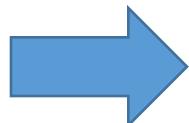
# 1. “action” 후보군 줄이기

(2) 스스로 발전하기 (reinforcement learning) 트레이닝 결과

프로기사  
흉내내는 모델

VS

업데이트 모델  
ver 1,000,000

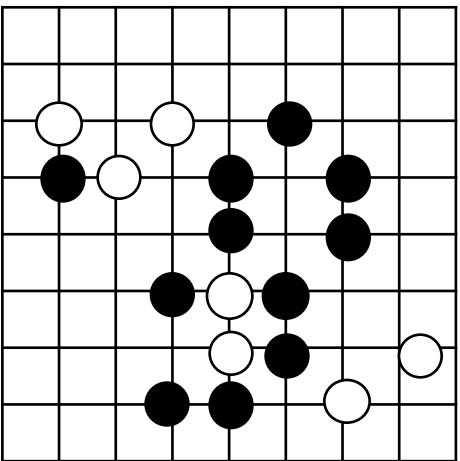


최종 모델이 80% 승리

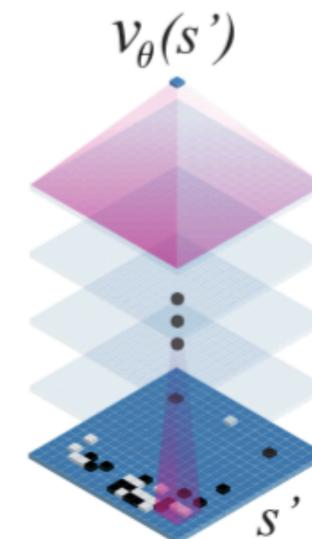
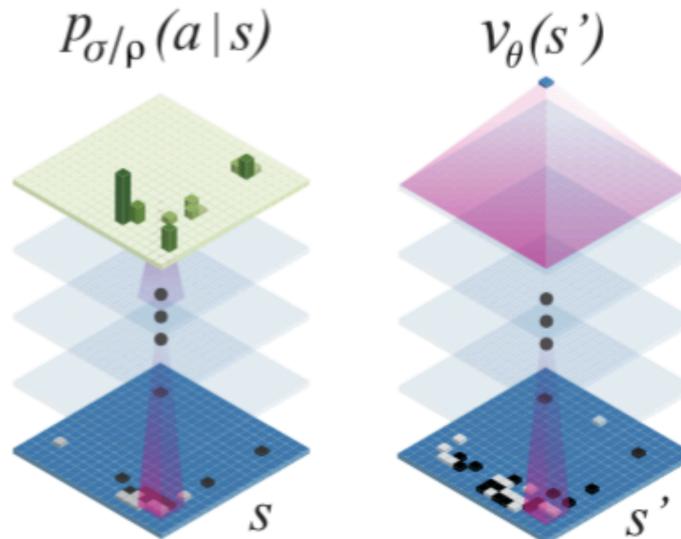
## 2. 판세 평가하기

## 2. 판세 평가하기

어느 판



업데이트 모델  
ver 1,000,000



기준 모델에 regression layer를 더함  
0~1 사이의 값으로 예측  
1에 가까우면 좋은 판세  
0에 가까우면 좋지 않은 판세

승/패

예측 모델  
(Regression)

승  
(0~1)

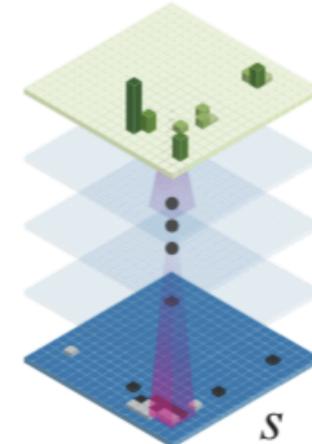
**Training:**  $\Delta\theta \propto \frac{\partial v_{\theta}(s)}{\partial \theta} (z - v_{\theta}(s))$

# 경우의 수 (Search Space) 줄이기

## 1. “action” 후보군 줄이기 (Breadth Reduction)

Policy Network

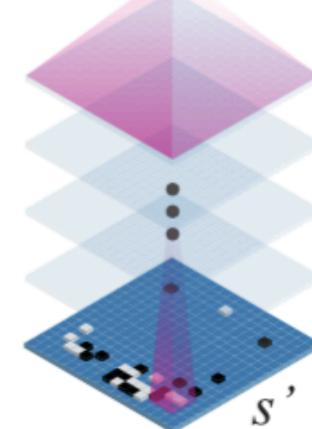
$$p_{\sigma/\rho}(a|s)$$



## 2. 판세 평가하기 (Depth Reduction)

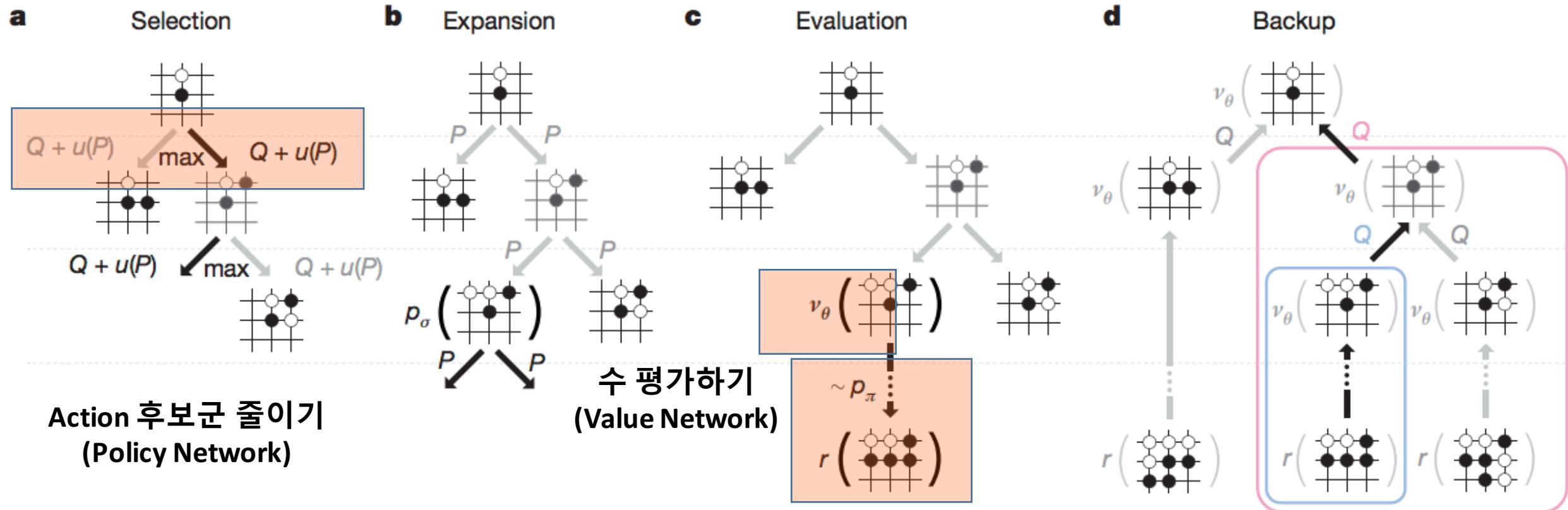
Value Network

$$v_\theta(s')$$



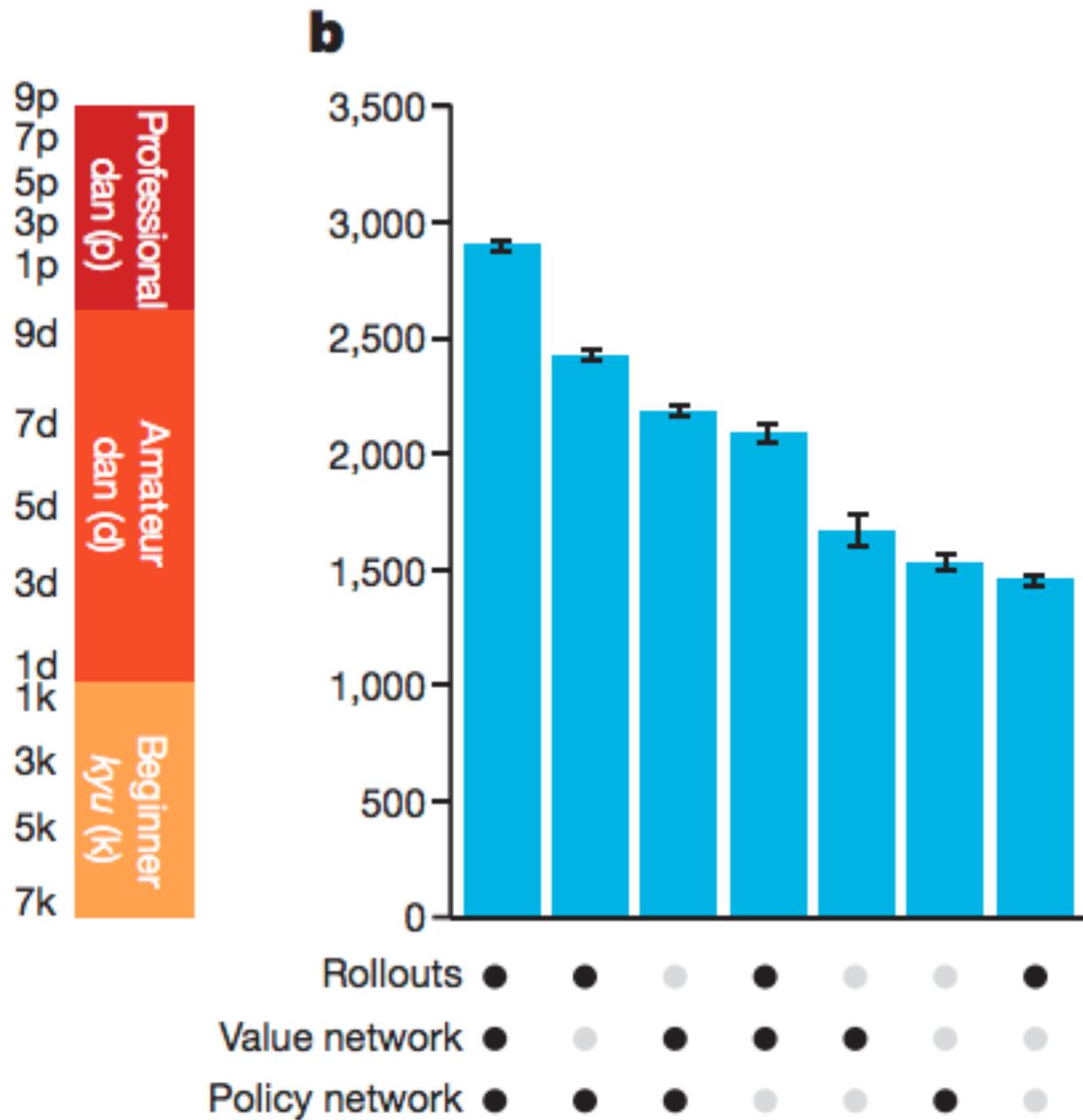
알파고 논문에서 이런 용어를 만들어서 부름

# 수 읽기 (w/ Monte Carlo Search Tree)



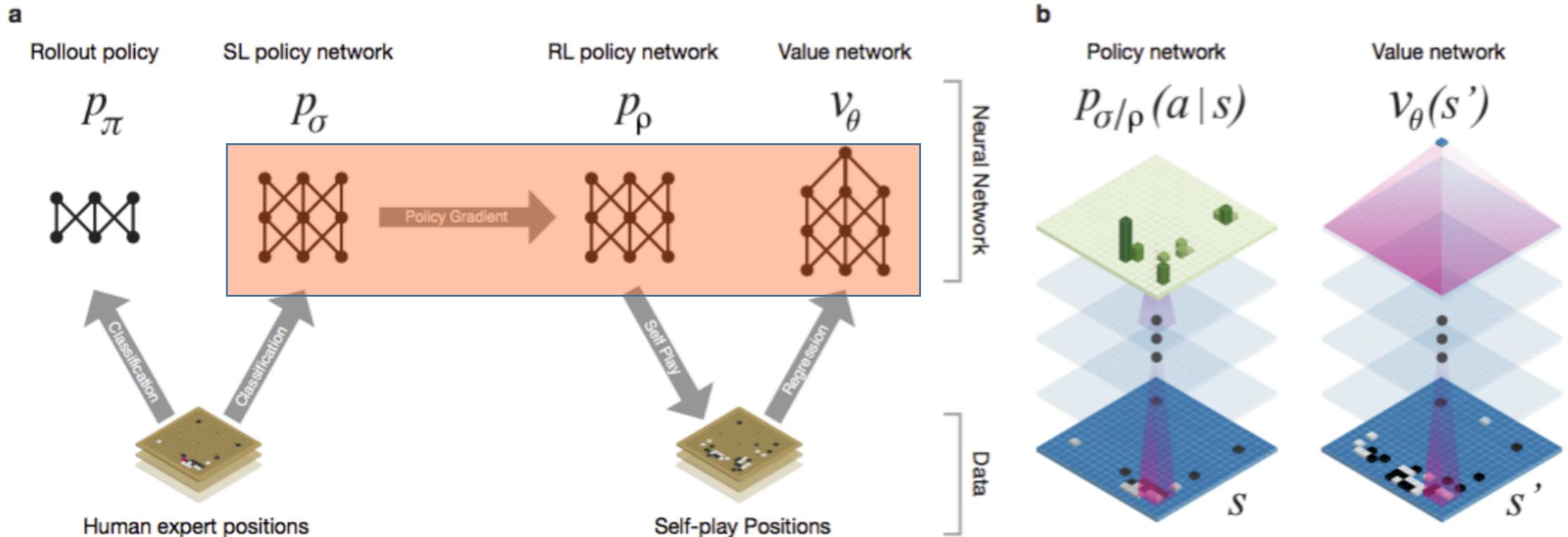
**(Rollout): Faster version of estimating  $p(a|s)$   
; shallow network (3 ms → 2μs)**

# 결과



# Takeaways

임의의 task를 위해 training한 network를 다양하게 활용



# 이세돌 9단 vs 알파고

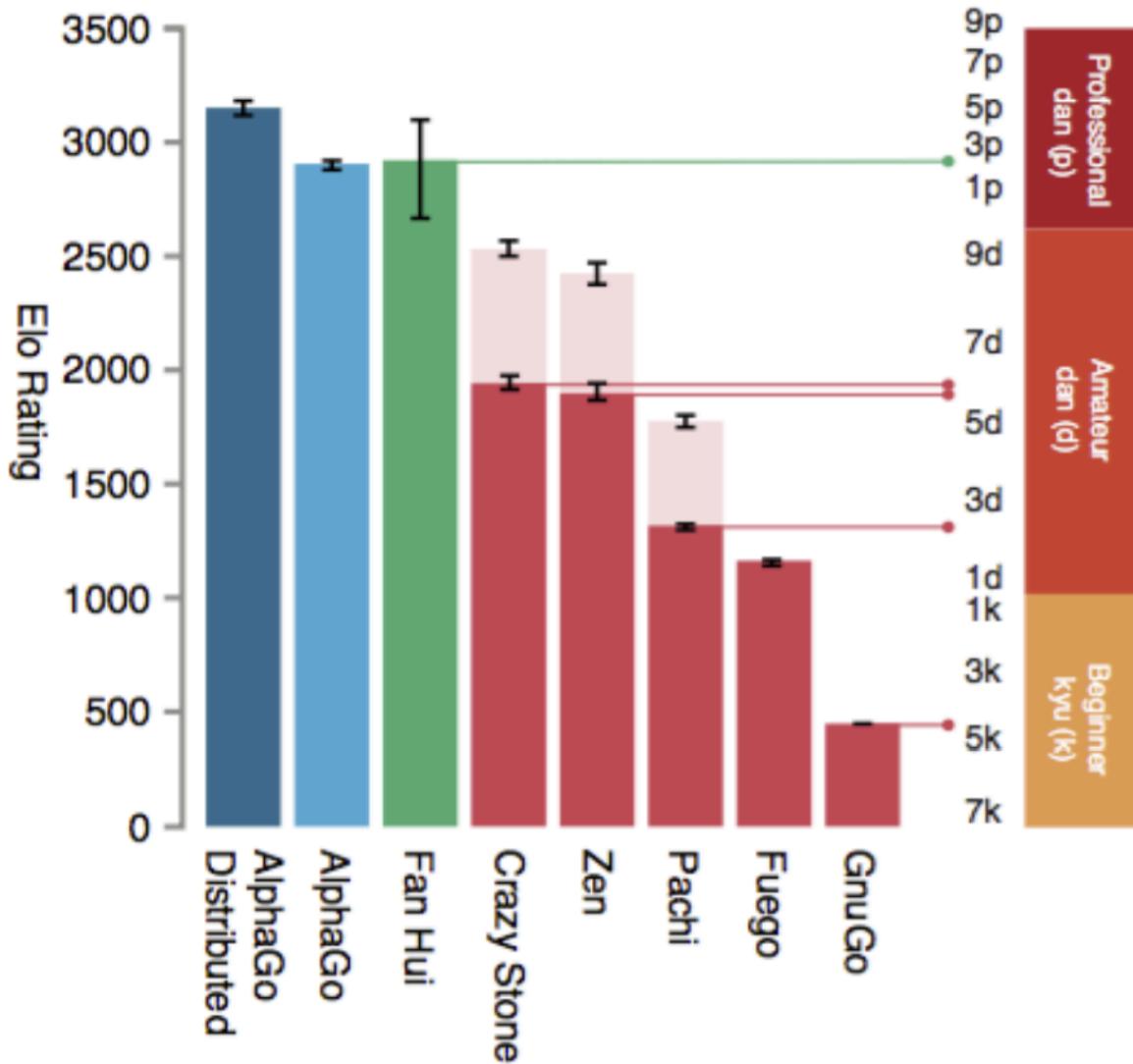


# 이세돌 9단 vs 알파고

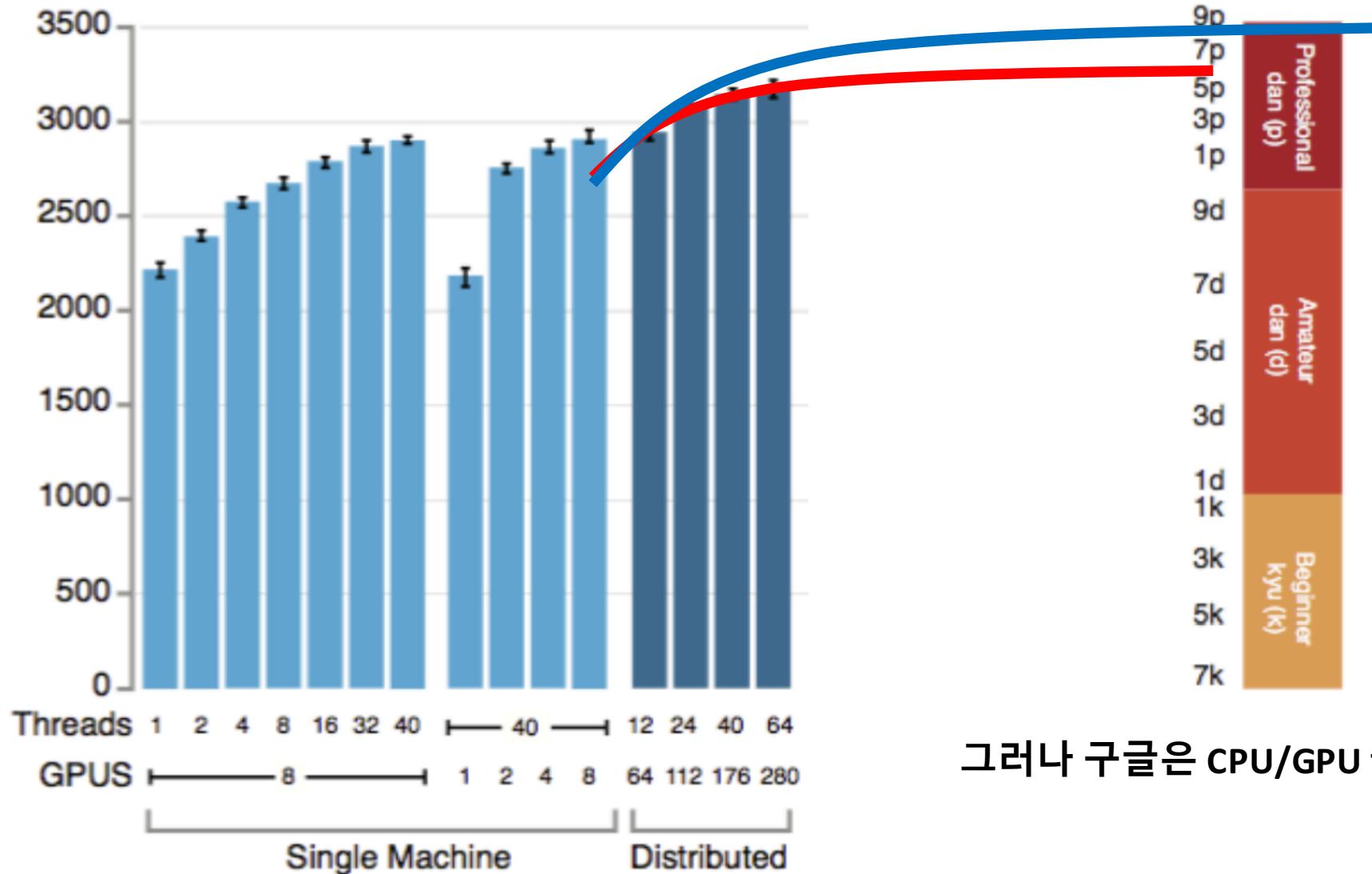
이세돌	알파고
<ul style="list-style-type: none"><li>- 성인 하루 권장 칼로리: ~2,500 kCal</li><li>- 가정: 대국에 하루 필요한 모든 칼로리 소모</li></ul> $2,500 \text{ kCal} * 4,184 \text{ J/kCal}$ $\sim= 10M \text{ [J]}$	<ul style="list-style-type: none"><li>- CPU: ~100 W, GPU: ~300 W</li><li>- CPU <b>1,202개</b>, GPU <b>176개</b></li></ul> $170,000 \text{ J/sec} * 5 \text{ hr} * 3,600 \text{ sec/hr}$ $\sim= 3,000M \text{ [J]}$

정말 대충 계산해서 ...

# 현재 알파고는 프로 5단 수준?



# CPU / GPU를 무한정 늘리면?



어떻게 수렴할지  
의견이 분분

그러나 구글은 CPU/GPU 늘리지 않겠다고 약속

# 매일 3만대국씩 학습한다던데?

결국은 어떤 점으로 수렴하겠지만  
알파고 논문에서 “자가 대결 대국 수 (RL sample #)와 성능 개선”간의 그래프는 공개하지 않음

# 이세돌 9단의 기보를 학습하면?

구글은 일단 이세돌 9단의 기보를 보고 학습하지 않겠다고 약속

학습한다해도 이세돌과의 적은 수 (5판)의 대국 데이터로  
수백만 번의 대국 데이터로 training 된 모델을 tune하는 것은 힘듦  
(prone to over-fitting, etc.)

# 알파고의 약점은?

# AlphaGo 작동원리

발표자: 문승환

PhD student

Language Technologies Institute, School of Computer Science

Carnegie Mellon University

[me@shanemoon.com](mailto:me@shanemoon.com)

3/2/2016