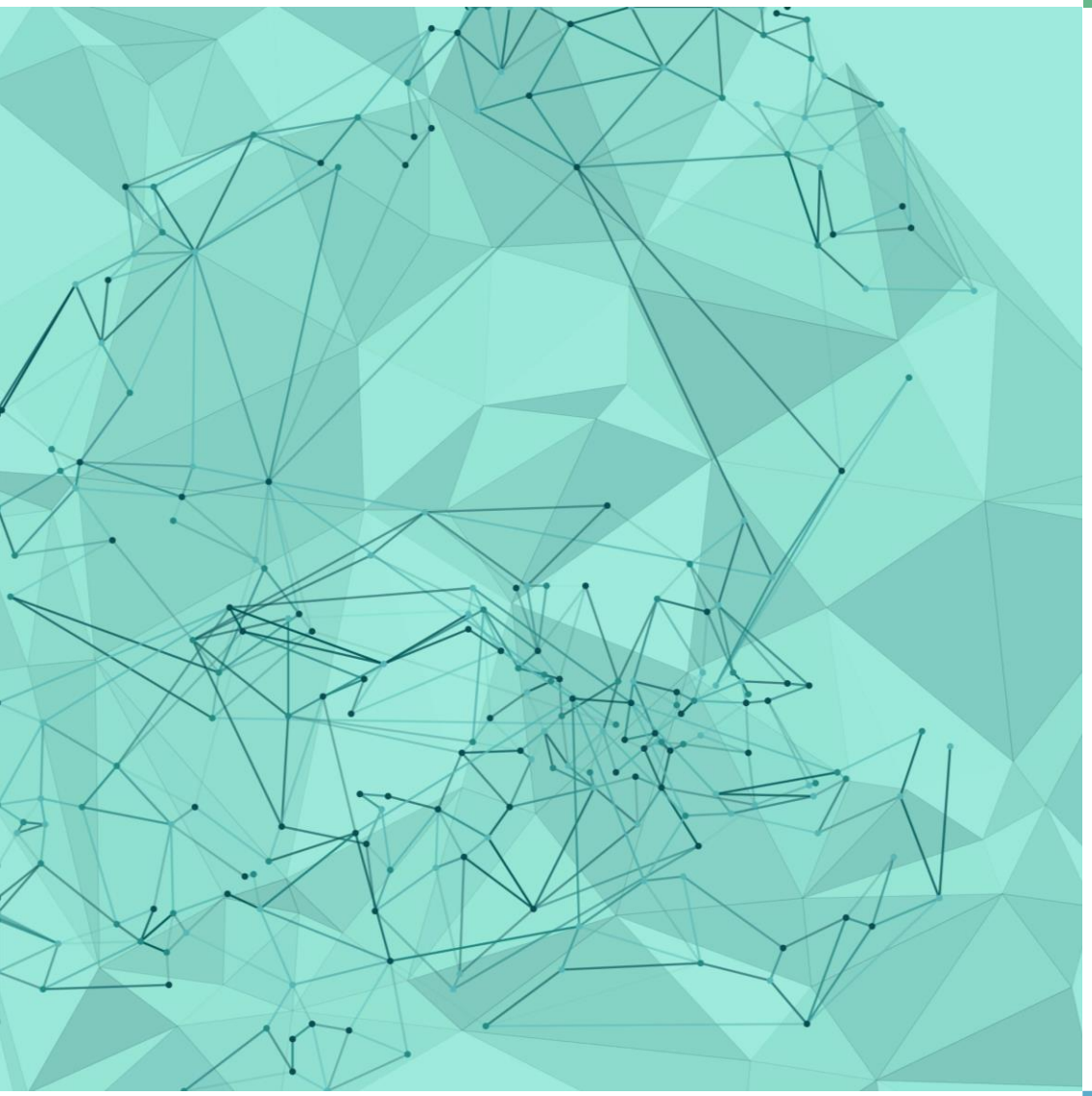


Mastering the game of Go without human knowledge

발표자: 박준영

목차

개요

AlphaGo Zero

: 인간의 지식 없이 랜덤에서 시작하여 바둑 최고의 경지까지 오른 인공지능.

개요

인간의 지식 (Human Knowledge)?

Feature	# of planes	Description
Stone colour	3	Player stone / opponent stone / empty
Ones	1	A constant plane filled with 1
Turns since	8	How many turns since a move was played
Liberties	8	Number of liberties (empty adjacent points)
Capture size	8	How many opponent stones would be captured
Self-atari size	8	How many of own stones would be captured
Liberties after move	8	Number of liberties after this move is played
Ladder capture	1	Whether a move at this point is a successful ladder capture
Ladder escape	1	Whether a move at this point is a successful ladder escape
Sensibleness	1	Whether a move is legal and does not fill its own eyes
Zeros	1	A constant plane filled with 0
Player color	1	Whether current player is black

Feature planes used by the policy network (all but last feature) and value network (all features).

출처: Mastering the game of Go with deep neural networks and tree search (2016)

개요

인간의 지식 (Human Knowledge)?

Feature	# of planes	Description
Stone colour	3	Player stone / opponent stone / empty
Ones	1	A constant plane filled with 1
Turns since	8	How many turns since a move was played
Liberties	8	Number of liberties (empty adjacent points)
Capture size	8	How many opponent stones would be captured
Self-atari size	8	How many of own stones would be captured
Liberties after move	8	Number of liberties after this move is played
Ladder capture	1	Whether a move at this point is a successful ladder capture
Ladder escape	1	Whether a move at this point is a successful ladder escape
Sensibleness	1	Whether a move is legal and does not fill its own eyes
Zeros	1	A constant plane filled with 0
Player color	1	Whether current player is black

Feature planes used by the policy network (all but last feature) and value network (all features).

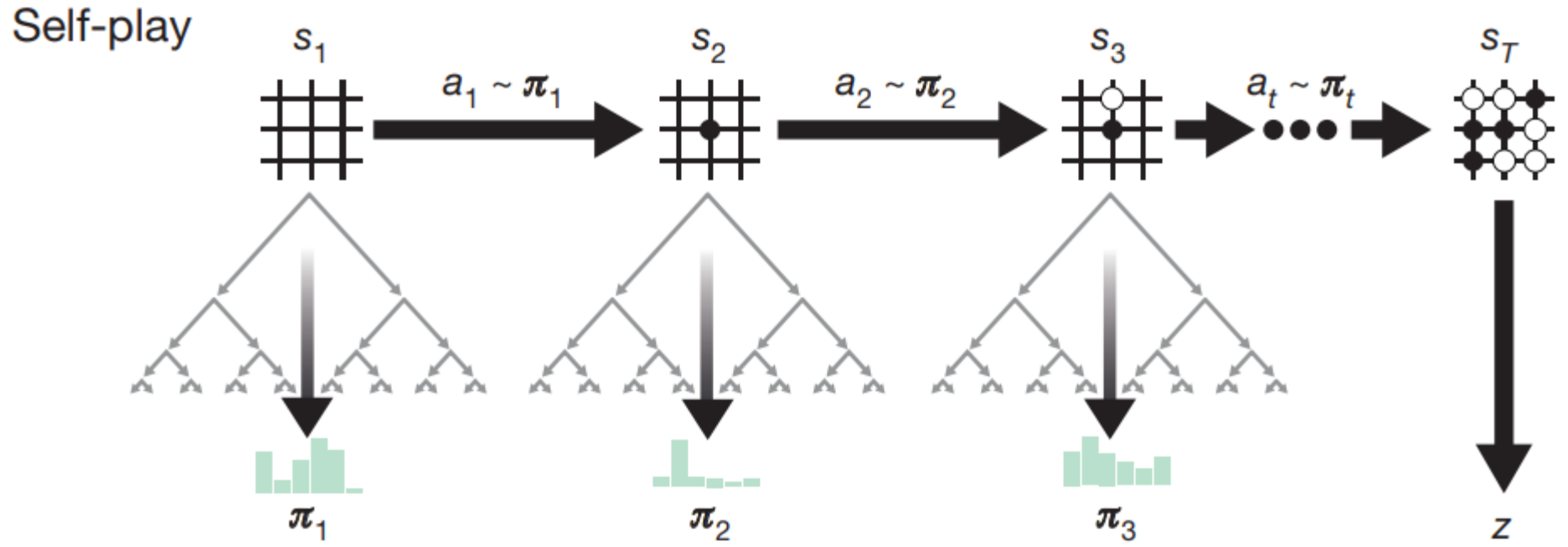
출처: Mastering the game of Go with deep neural networks and tree search (2016)

개요

AlphaGo Zero에 쓰인 domain knowledge

- 완벽한 게임 규칙(perfect knowledge of the game rules)
  - 각 상황마다 할 수 있는 선택지를 알려준다.
  - 722턴이 끝나면 게임 강제 종료.
- Tromp-Taylor scoring 방식으로 점수 산정
  - 사람이 주로 쓰는 규칙(한국룰, 중국룰, 일본룰)은 완벽하지 않아 각종 예외가 존재.
  - 수학적으로 명확한 Tromp Taylor 규칙을 이용해 학습.
- 바둑판은 rotation이나 reflection에 대해 invariant
  - 위와 같은 변환을 이용해 학습 데이터 뻥튀기 가능.

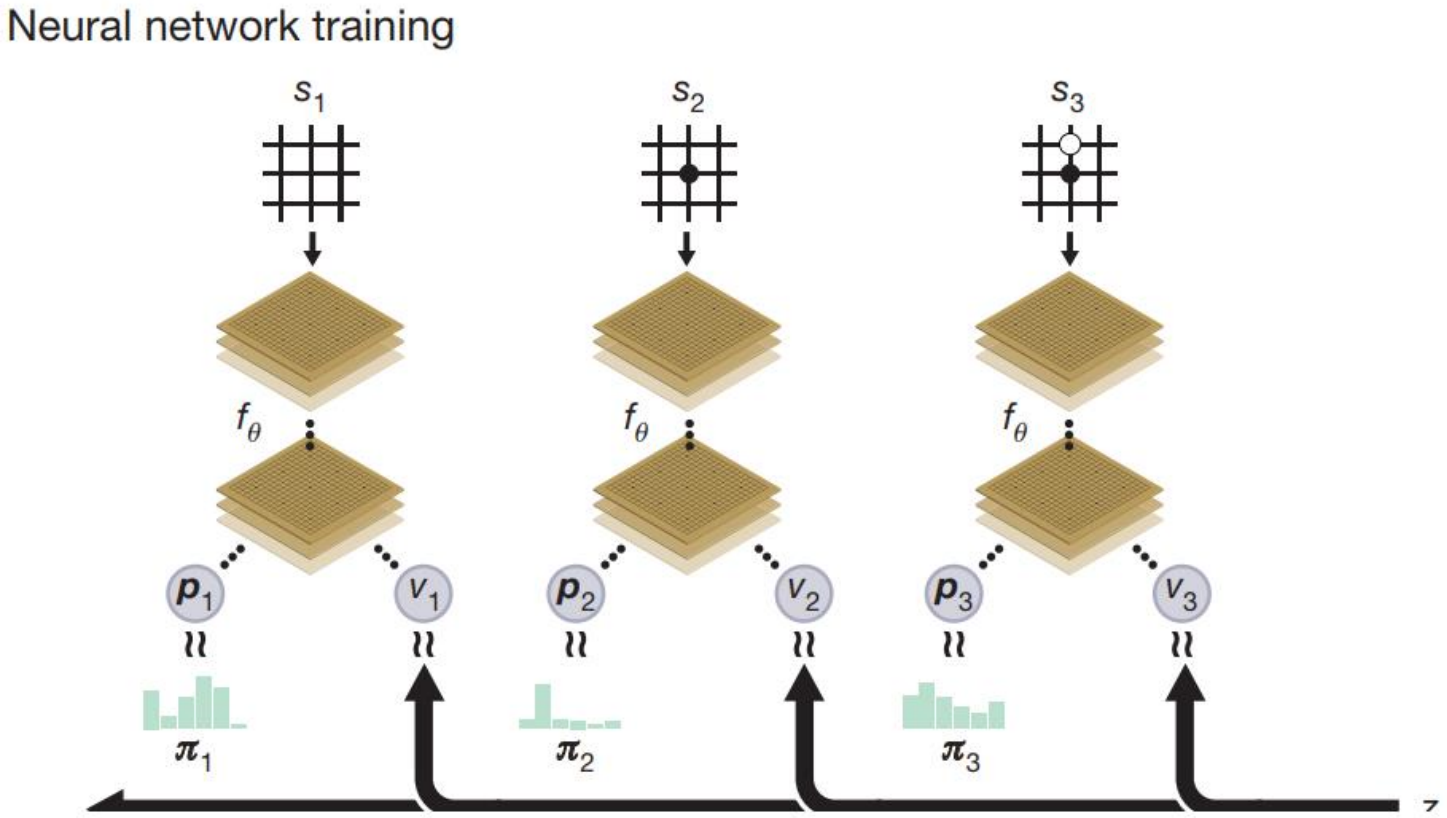
개요



학습데이터를 만드는 과정

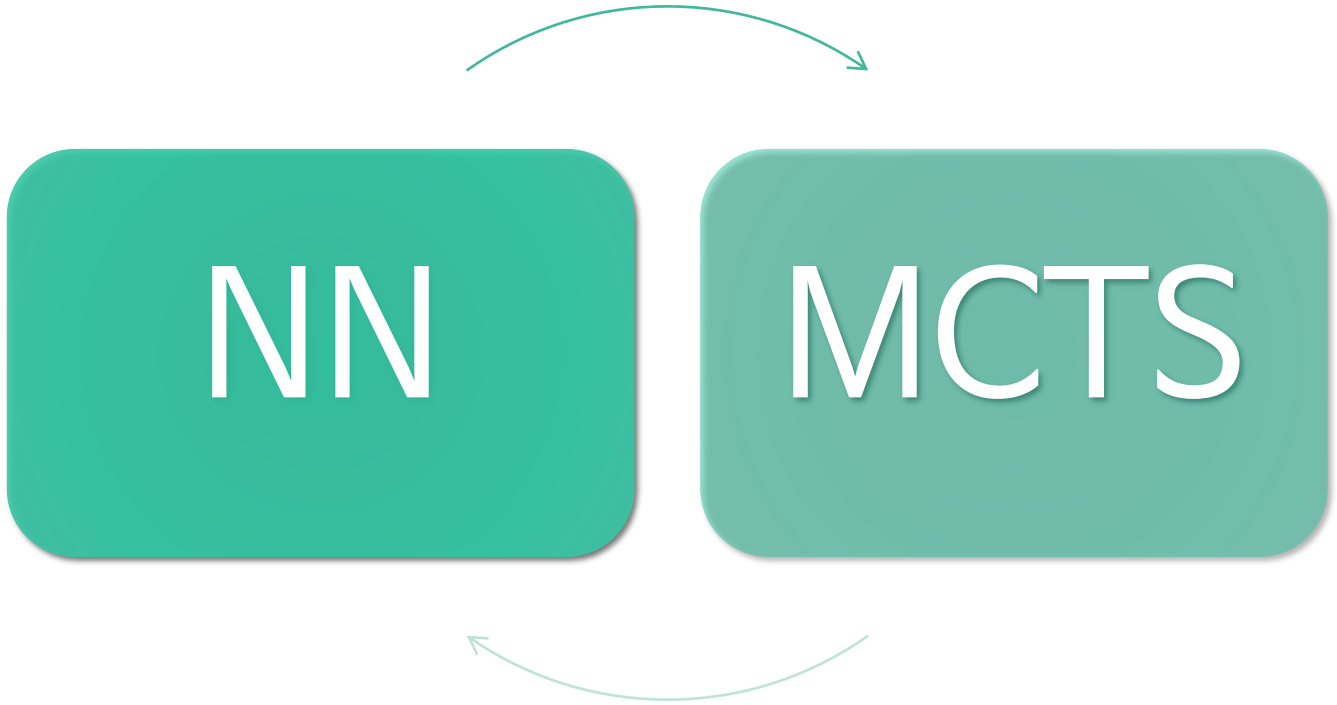
출처: Mastering the game of Go without human knowledge (2017)

개요



출처: Mastering the game of Go without human knowledge (2017)

개요

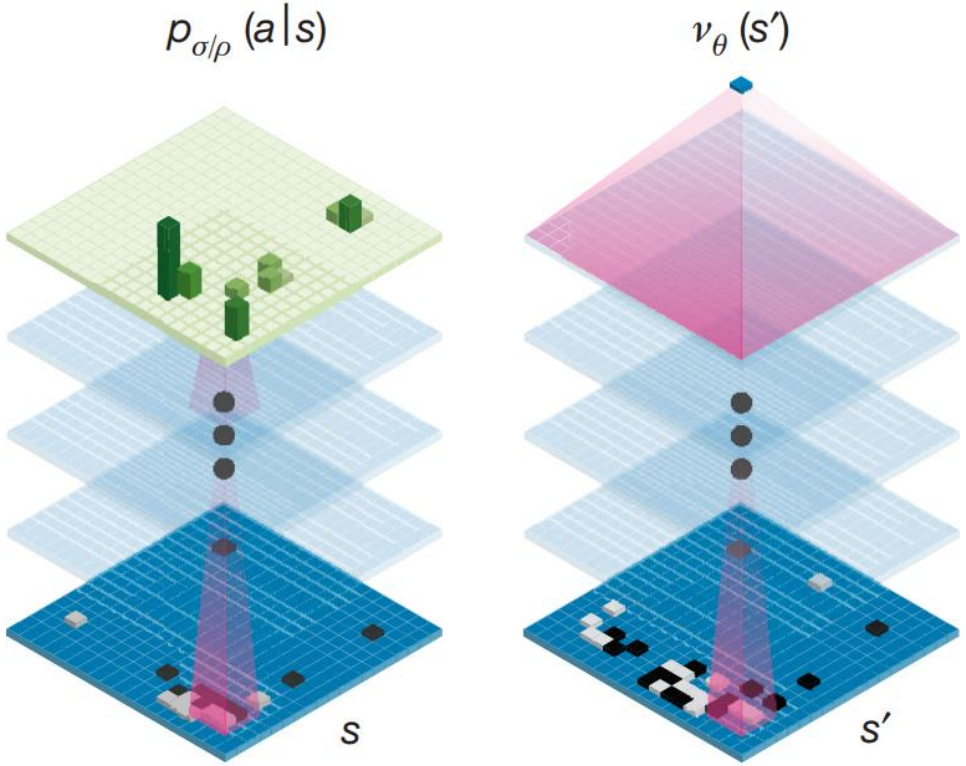


서로가 서로를 보완하며 발전

Dual Residual Network

인간의 직감에 해당하는 구조.

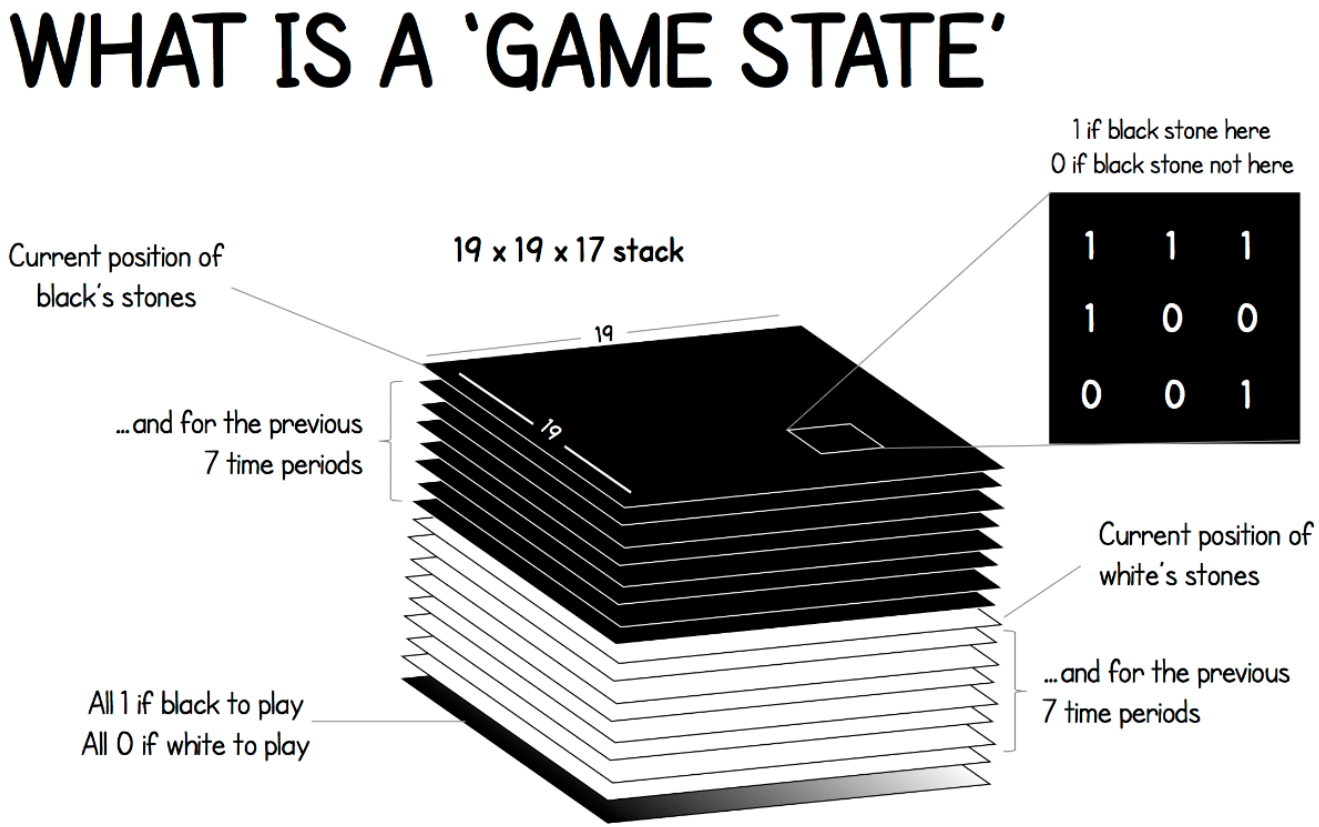
둘 만한 곳은 어디?



지금 대충 얼마나 유리?

출처: Mastering the game of Go with deep neural networks and tree search (2016)

Dual Residual Network



출처: AlphaGo Zero Explained In One Diagram
https://medium.com/applied-data-science/alphago-zero-explained-in-one-diagram-365f5abf67e0

Dual Residual Network

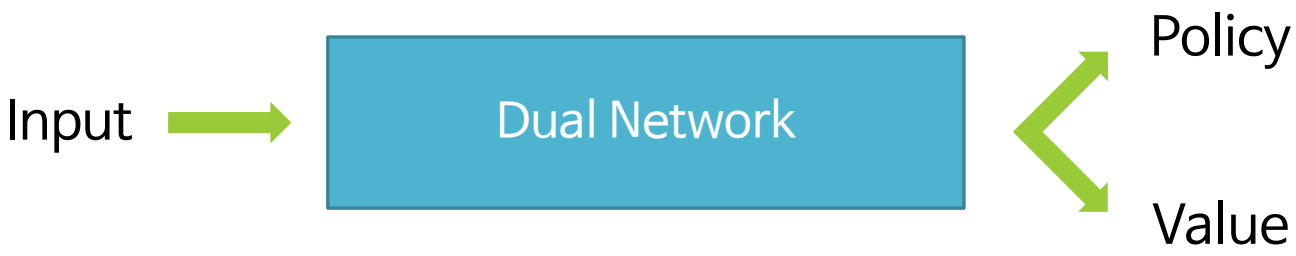
Dual + Residual

Policy와 Value를 동시에 출력

Residual Network를 사용

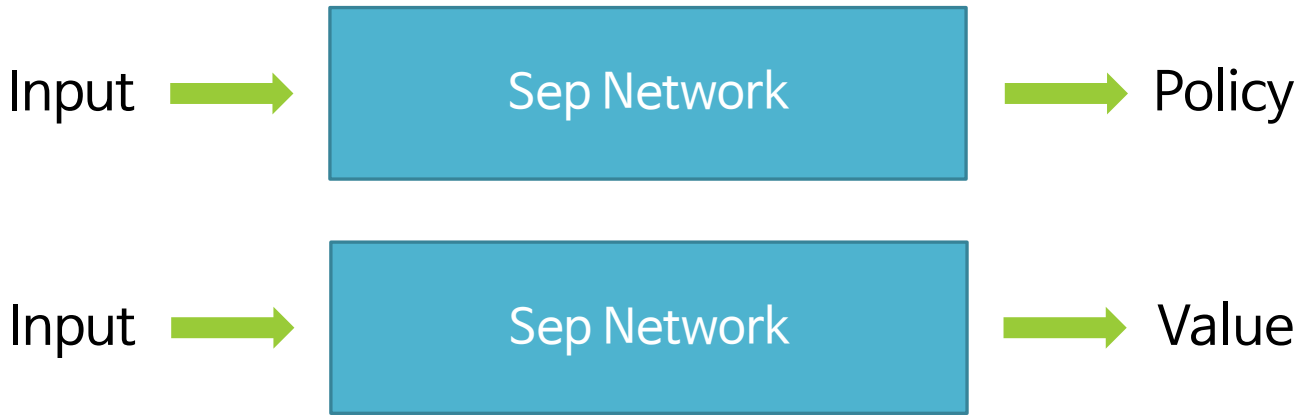
Dual Residual Network

Dual



Dual Residual Network

Separate



Dual Residual Network

Residual Network

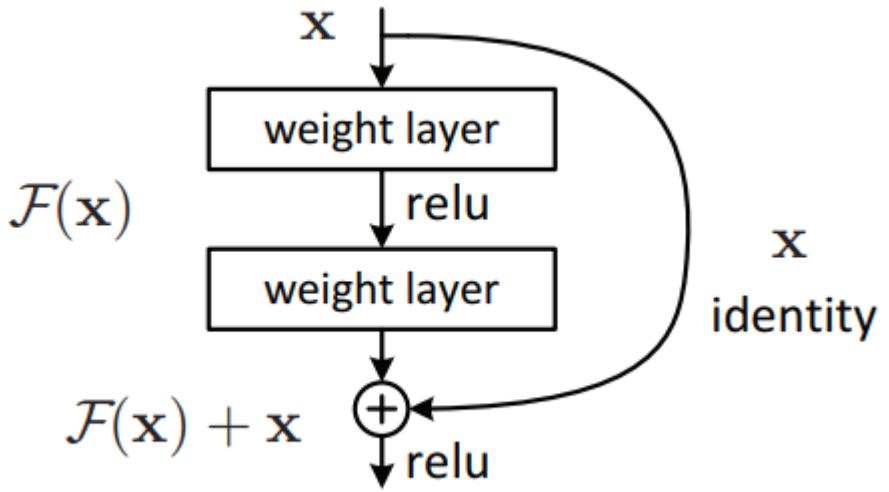
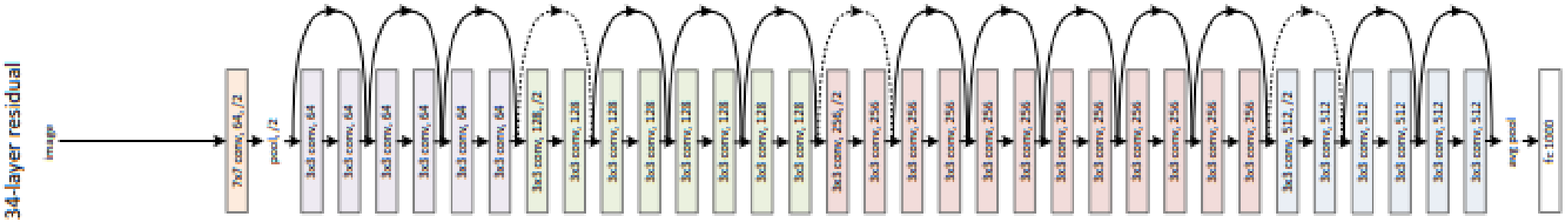


Figure 2. Residual learning: a building block.

Dual Residual Network

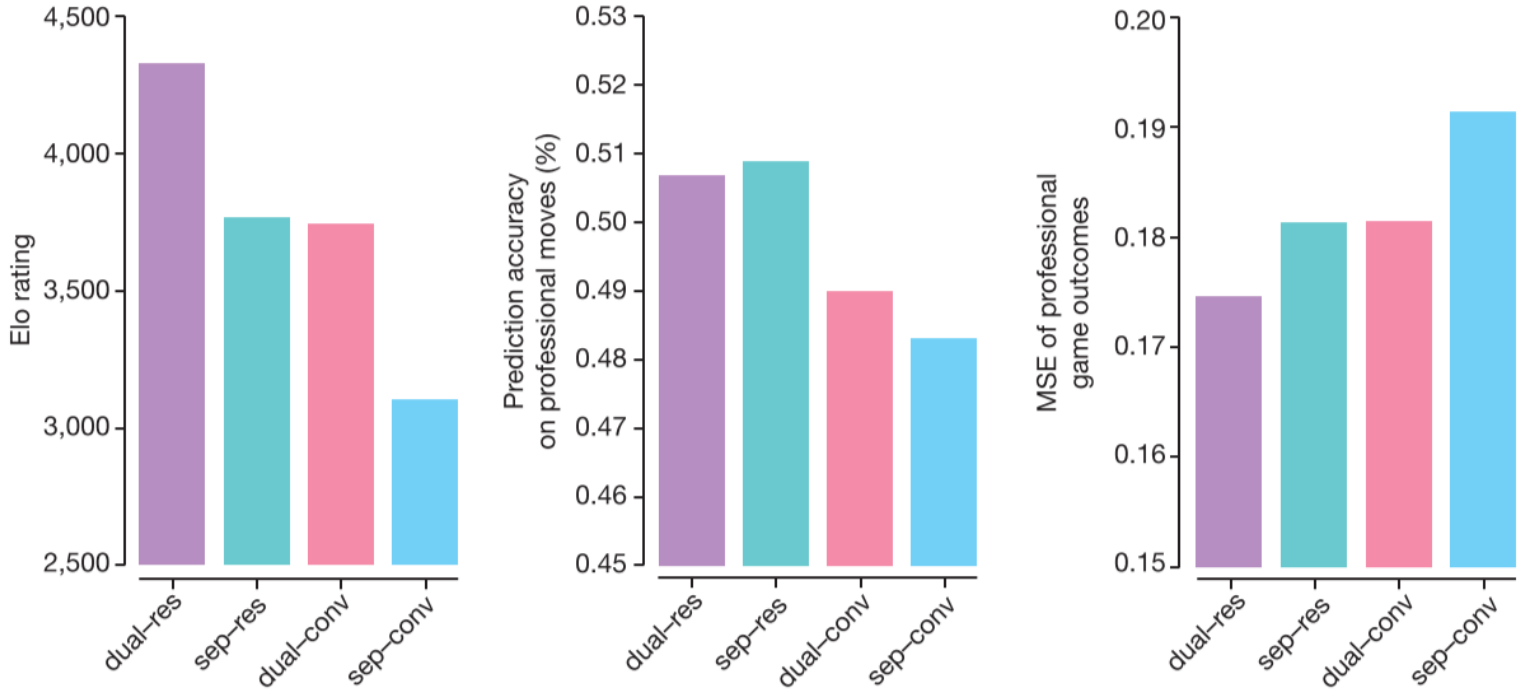
Residual Network

: 앞의 Residual Block을 많이 쌓은 신경망 모델



출처: Deep Residual Learning for Image Recognition (2015)

Dual Residual Network



출처: Mastering the game of Go without human knowledge (2017)

Dual Residual Network

Residual Network

- Convolutional Block
  - 256 filters, 3×3 kernel convolution layer
  - Batch normalization
  - Rectifier nonlinearity

Dual Residual Network

Residual Network

- Residual Block
  - 256 filters, 3×3 kernel convolution layer
  - Batch normalization
  - Rectifier nonlinearity
  - 256 filters, 3×3 kernel convolution layer
  - Batch normalization
  - Skip connection
  - Rectifier nonlinearity

Dual Residual Network

Residual Network

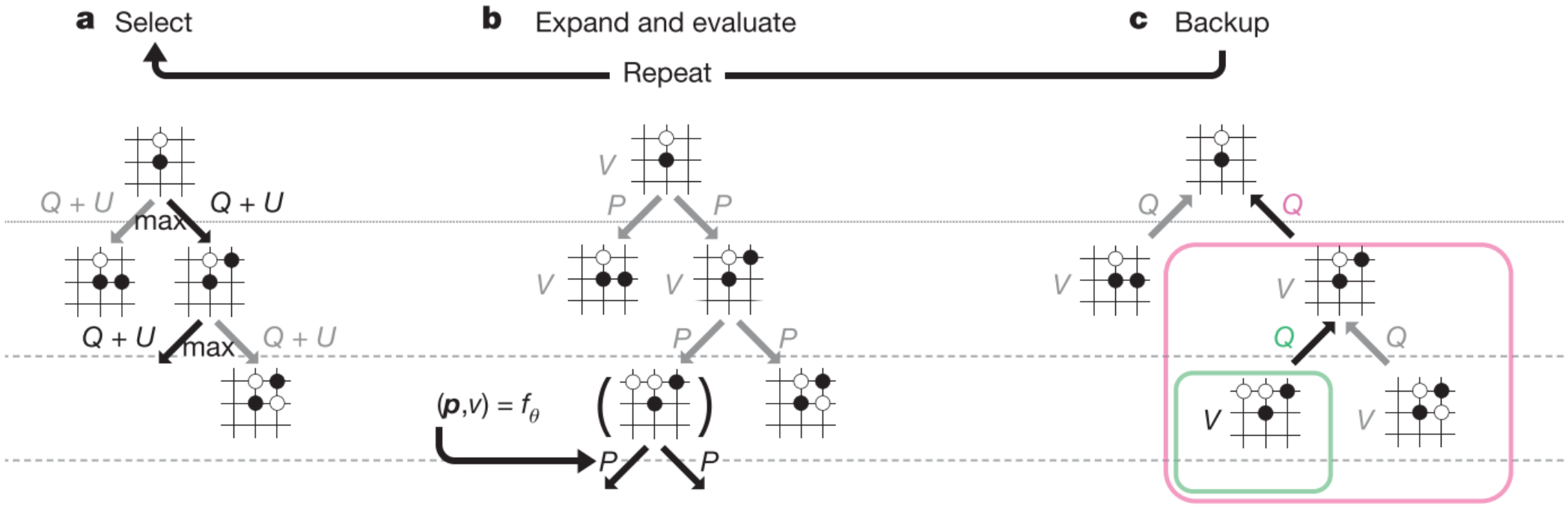
- Policy Head
  - 2 filters, 1×1 kernel convolution layer
  - Batch normalization
  - Rectifier nonlinearity
  - Fully connected layer (output: $\mathbb{R}^{19^2+1}$)

Dual Residual Network

Residual Network

- Value Head
  - 1 filters, 1×1 kernel convolution layer
  - Batch normalization
  - Rectifier nonlinearity
  - Fully connected layer (output: $\mathbb{R}^{256}$)
  - Rectifier nonlinearity
  - Fully connected layer (output: $\mathbb{R}^{1}$)
  - Tanh layer

Monte Carlo Tree Search



Monte Carlo Tree Search

Tree Node

- 각 노드마다 상태(s_t), 행동(a_t) 밑 통계치를 저장.

- $N(s,a)$: 해당 노드에 방문한 횟수 (탐색한 횟수)
- $W(s,a)$: 해당 value의 총합
- $Q(s,a)$: 해당 노드의 승률
- $P(s,a)$: 해당 노드의 policy

Monte Carlo Tree Search

Select

: 탐색할 노드를 찾기 위한 과정

- PUCT 알고리즘을 사용.

$$u(s,a) = c_{\text{puct}} P(s,a) \frac{\sqrt{\sum_b N(s,b)}}{1 + N(s,a)}$$

  - c_{puct}: 트리 탐색의 깊이/너비를 조절하는 상수.

- 너무 한 노드만 탐색하는 것을 막기 위한 알고리즘.

Monte Carlo Tree Search

Select

: 탐색할 노드를 찾기 위한 과정

- 최종적으로 다음의 수식에 따라 탐색할 노드를 선택.

$$a_t = \text{argmax}_a\big(Q(s,a) + u(s,a)\big)$$

- Leaf node에 도달할 때까지 반복.

Monte Carlo Tree Search

Expand and Evaluate

: 선택된 노드를 탐색하고 이후 상태를 예측하는 과정.

- 탐색할 노드를 인공 신경망 연산 Task Queue에 넣고 대기.
- 인공 신경망에서 나온 policy 값을 이용해 다음 상태를 expand.

Monte Carlo Tree Search

Backup

: 인공신경망에서 나온 값을 바탕으로 tree를 업데이트 하는 과정.

- Leaf node에서 root node 방향으로 반대로 value를 업데이트 함.

→ 이 모든 과정(simulation)을 다수의 스레드에서 병렬적으로 수행.

Monte Carlo Tree Search

Play

: 실질적으로 다음 행동을 결정하는 과정.

- 정해진 시간동안 또는 정해진 횟수만큼 simulation을 수행한 뒤 멈춤.
- Root node의 자식 노드 중에서 방문 횟수가 가장 많은 행동을 선택.
  - 많이 방문 = 좋은 행동

Training Pipeline

1. Selfplay
2. Optimization
3. Evaluation

   → 위 세 과정은 병렬적으로 수행된다.

Training Pipeline - Selfplay

: 학습 데이터를 만드는 과정

- 현재 최고의 모델 α_θ를 이용해 학습 데이터를 생성.
- 한 iteration에 25,000 게임을 수행.
- 한 턴당 1,600회 MCTS 시뮬레이션 수행.

- MCTS simulation 결과와 게임의 결과를 바탕으로 학습 데이터 생성.
  - 30번째 턴까지는 $\tau = 1$을 사용, 이후로는 $\tau \to 0$을 사용. (시작 단계에서의 다양성을 보장하기 위해)

$$\pi \approx \frac{N(s,a)^\tau}{\sum_b N(s,b)^\tau}$$

Training Pipeline - Optimization

: 학습 데이터를 바탕으로 새로운 모델을 만드는 과정

- 64개의 GPU worker를 이용해 학습.
- 각 worker마다 32 batch-size 사용.
- 최근 생긴 500,000 게임에서 랜덤하게 골라서 배치를 구성.

$$l = (z - v)^2 - \pi^T \log p + c\|\theta\|^2$$

Training Pipeline - Optimization

Thousands of steps	Reinforcement learning	Supervised learning
0–200	10^{-2}	10^{-1}
200–400	10^{-2}	10^{-2}
400–600	10^{-3}	10^{-3}
600–700	10^{-4}	10^{-4}
700–800	10^{-4}	10^{-5}
>800	10^{-4}	-

Learning rate used during reinforcement learning and supervised learning experiments, measured in thousands of steps (mini-batch updates).

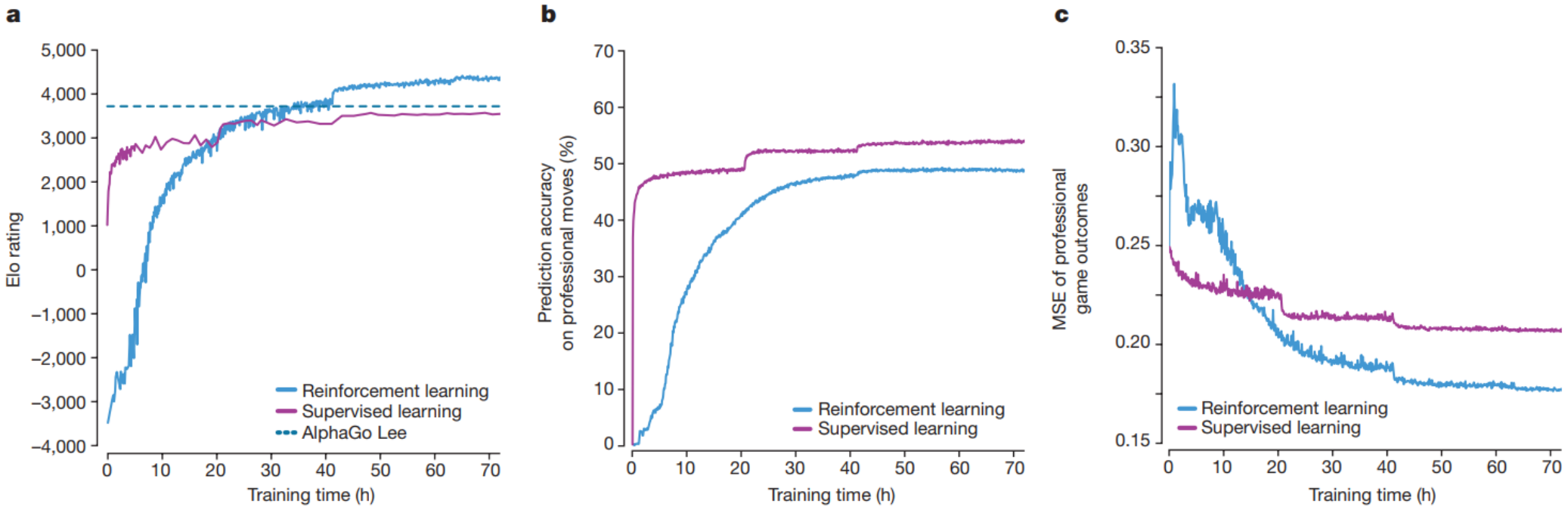
출처: Mastering the game of Go without human knowledge (2017)

Training Pipeline - Evaluation

: 새로운 모델을 평가하는 과정

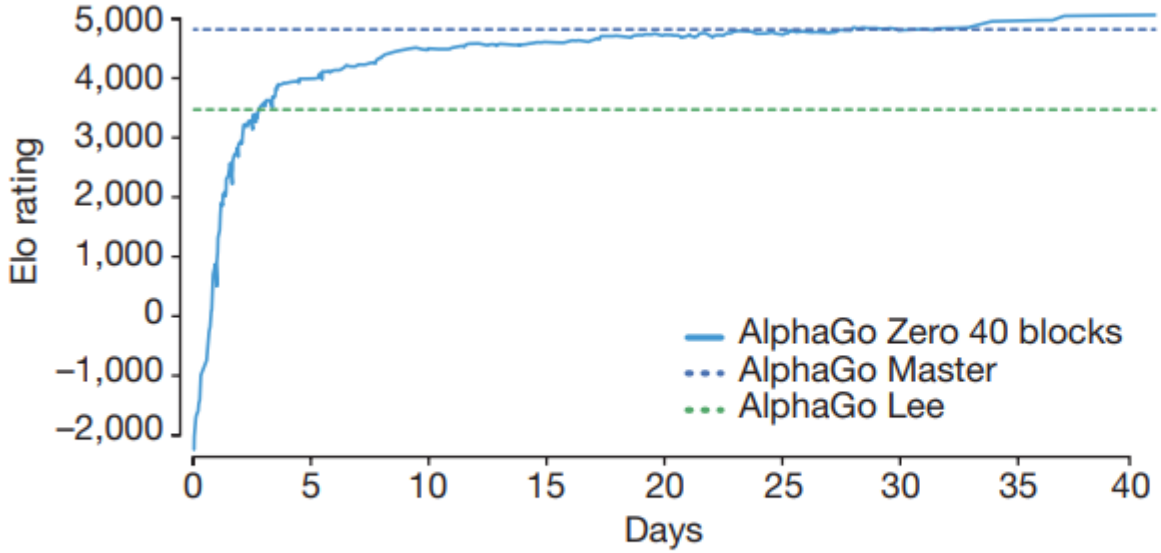
- 좋은 퀄리티의 학습 데이터를 만들기 위해 새로운 모델이 나오면 평가함.
- 현재의 가장 좋은 모델 α_θ와 400판을 붙어 승률이 55%보다 높으면 모델 교체.

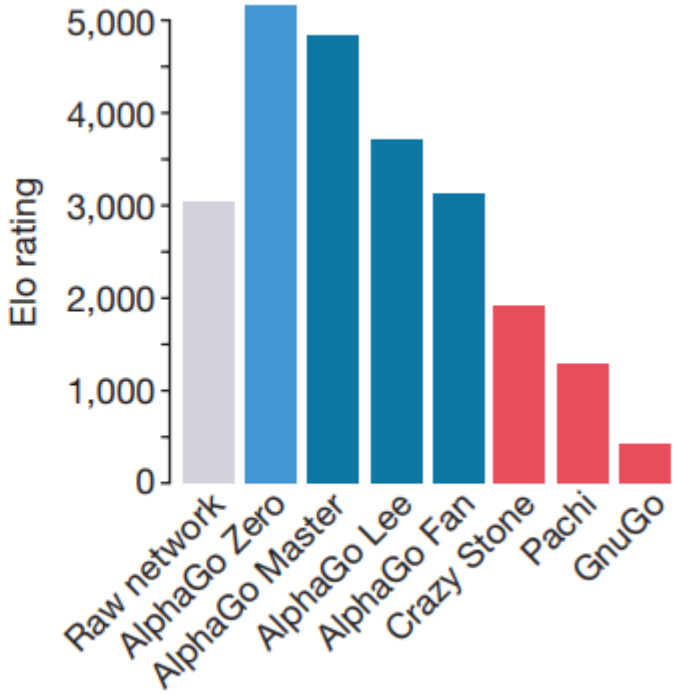
Results



출처: Mastering the game of Go without human knowledge (2017)

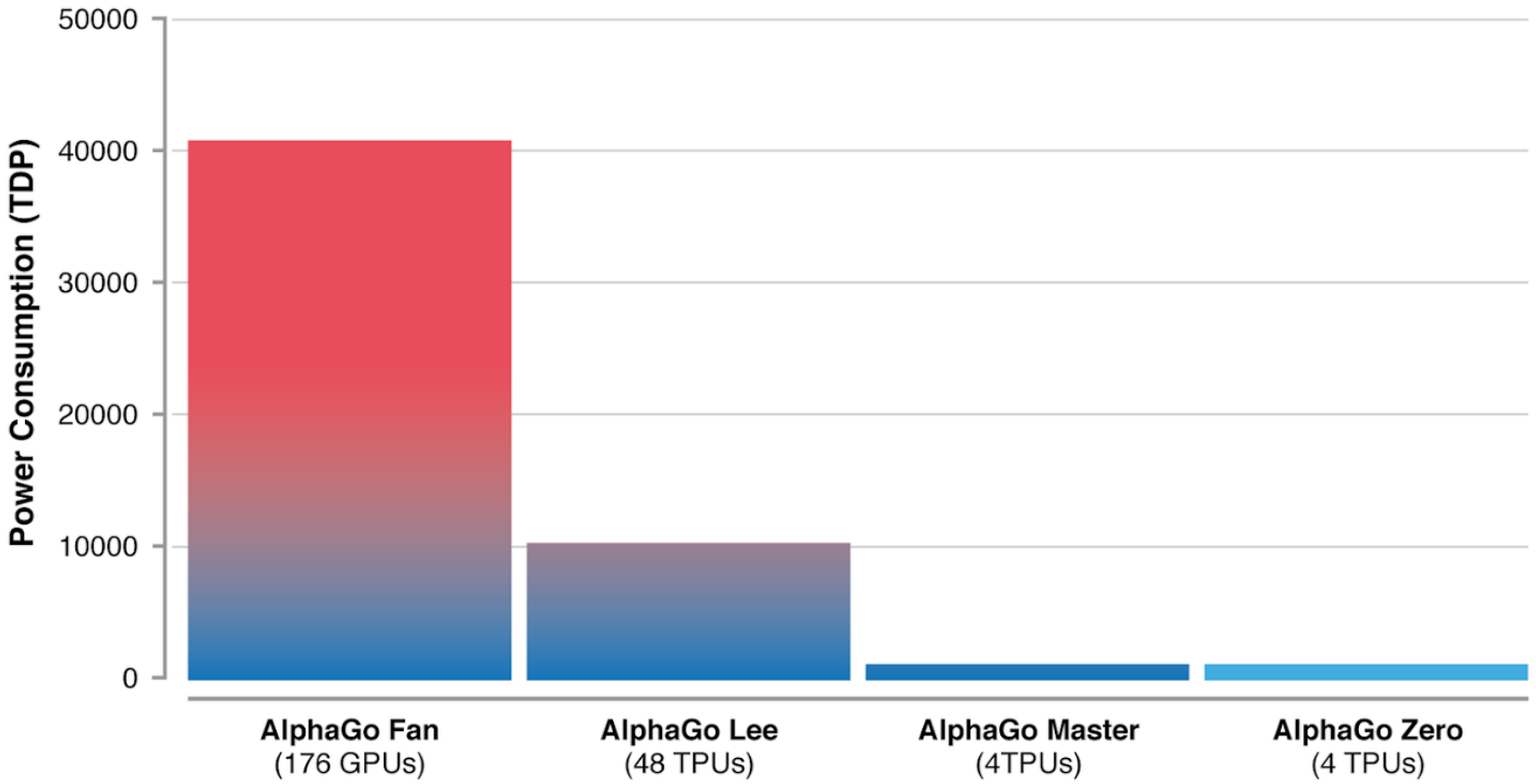
Results





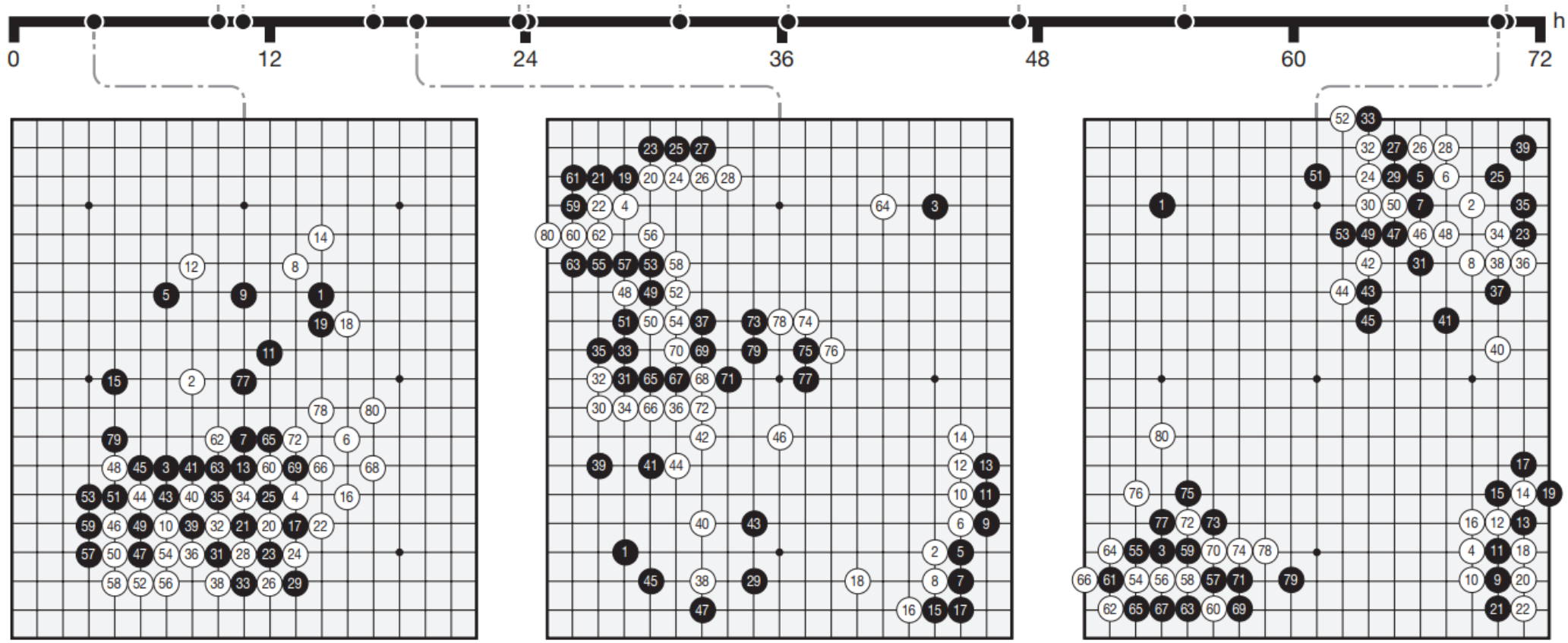
출처: Mastering the game of Go without human knowledge (2017)

Results



출처: AlphaGo Zero: Starting from scratch, DeepMind blog
https://deepmind.com/blog/article/alphago-zero-starting-scratch

Results



References

- Silver, D. *et al*. Mastering the game of Go with deep neural networks and tree search. (2016)
- Silver, D. *et al*. Mastering the game of Go without human knowledge (2017)
- He, K. Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition (2016)
- Foster. AlphaGo Zero Explained In One Diagram. https://medium.com/applied-data-science/alphago-zero-explained-in-one-diagram-365f5abf67e0 (2017)

Thank You