

# 알파고1 논문 리뷰

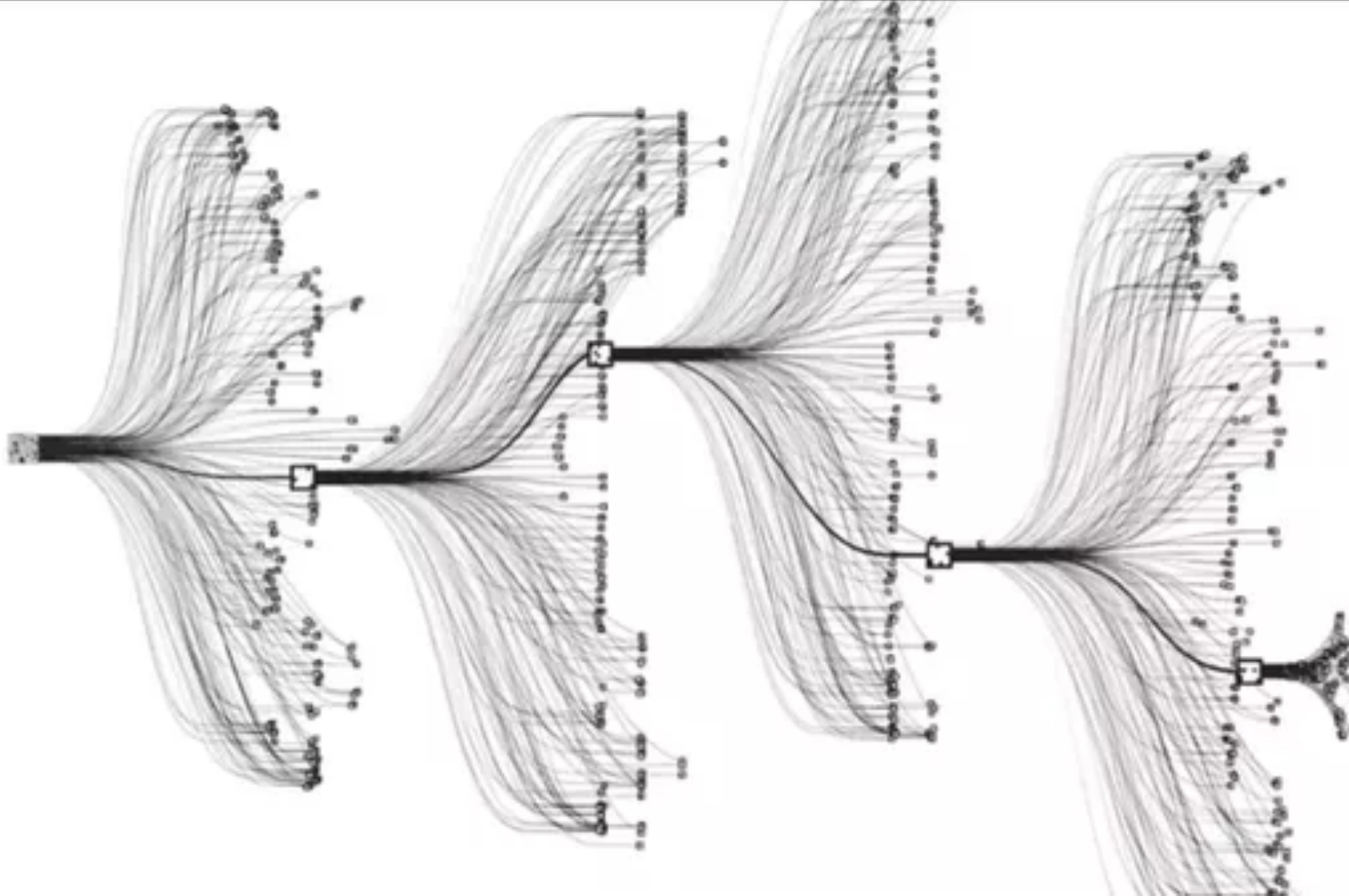
Mastering the game of Go with deep  
neural networks and tree search

(D. Silver, A. Huang et al. 네이처, 2016)

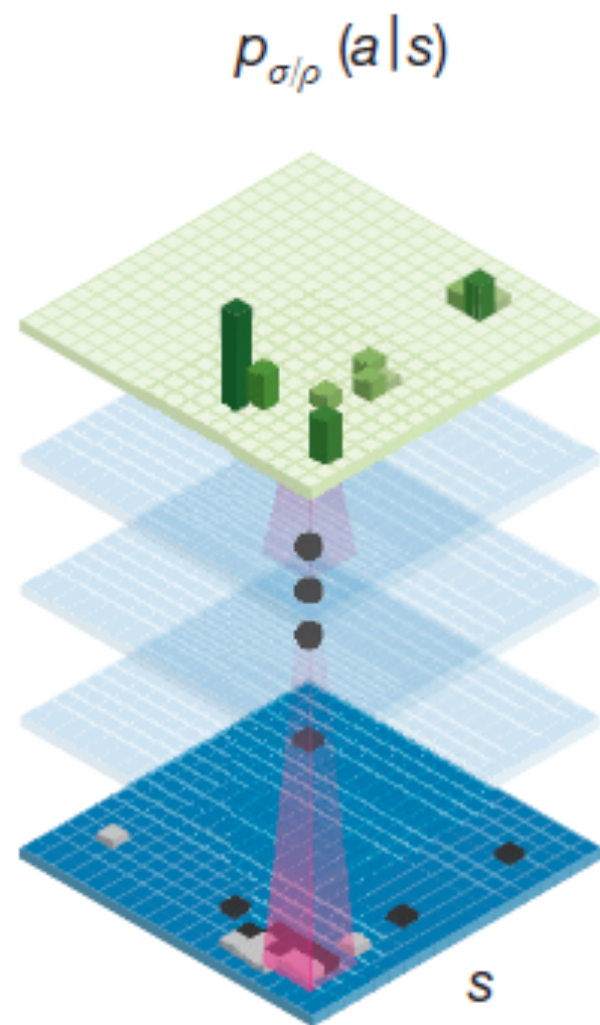


# Monte Carlo Tree Search

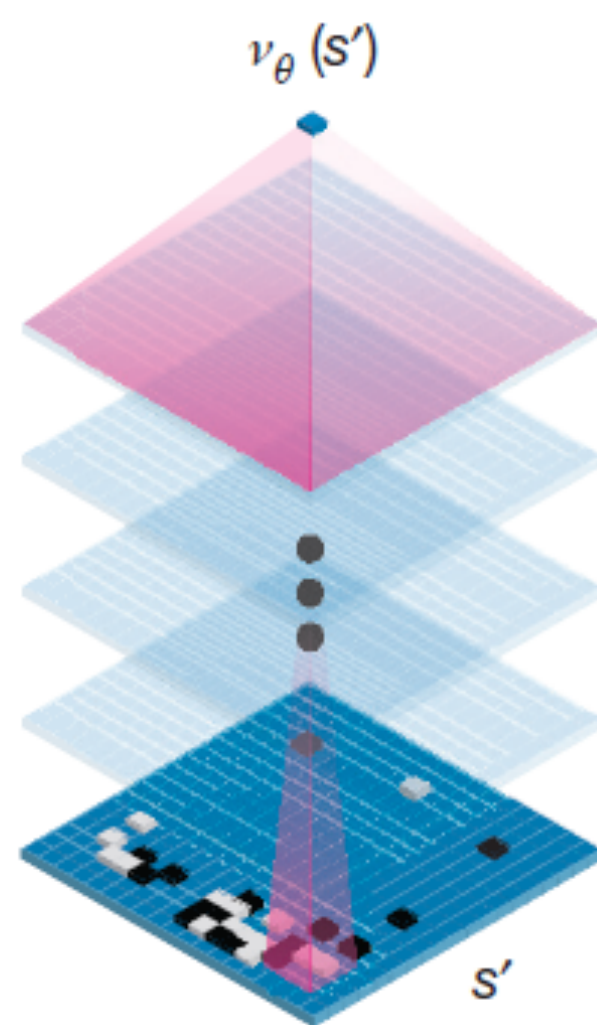
---

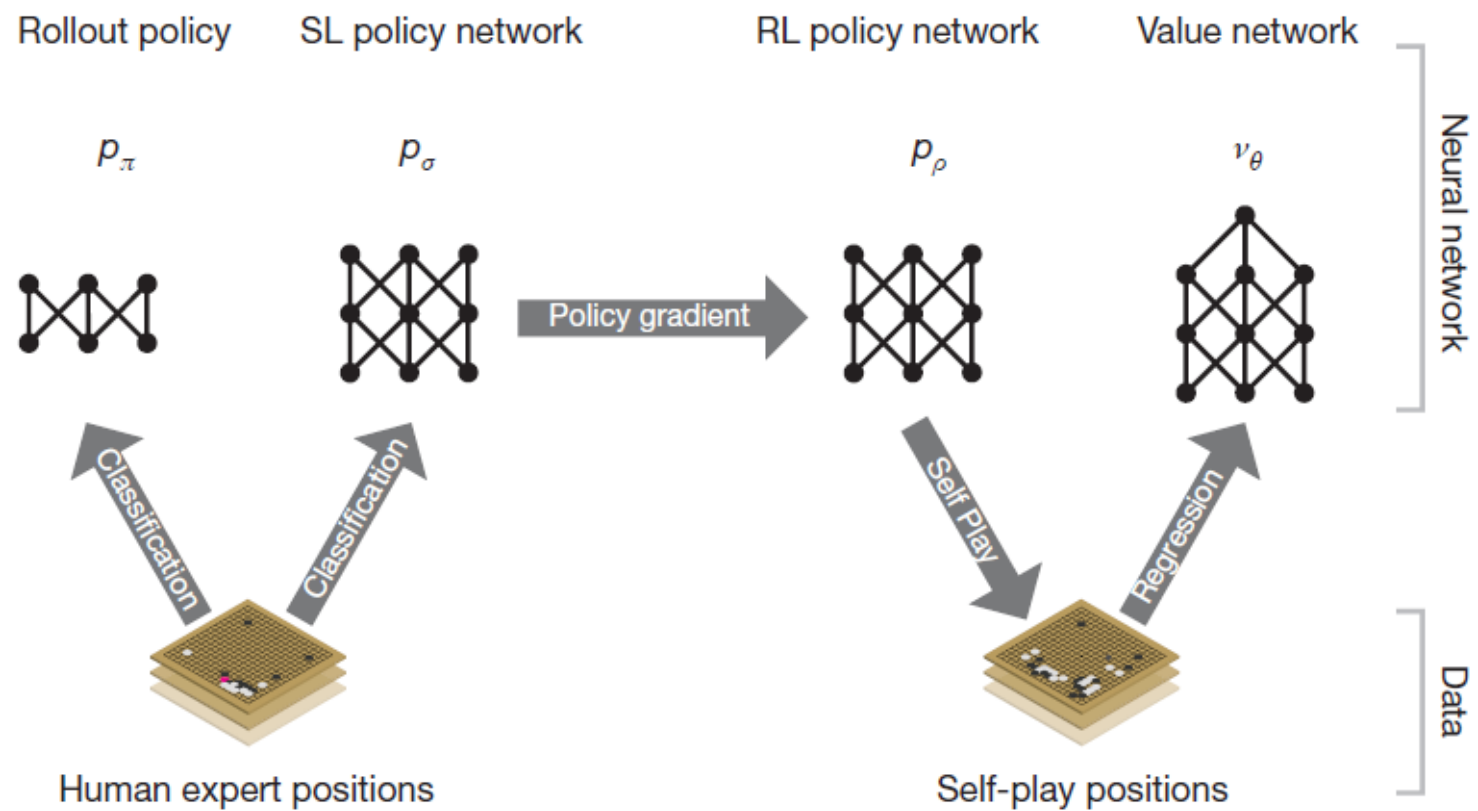


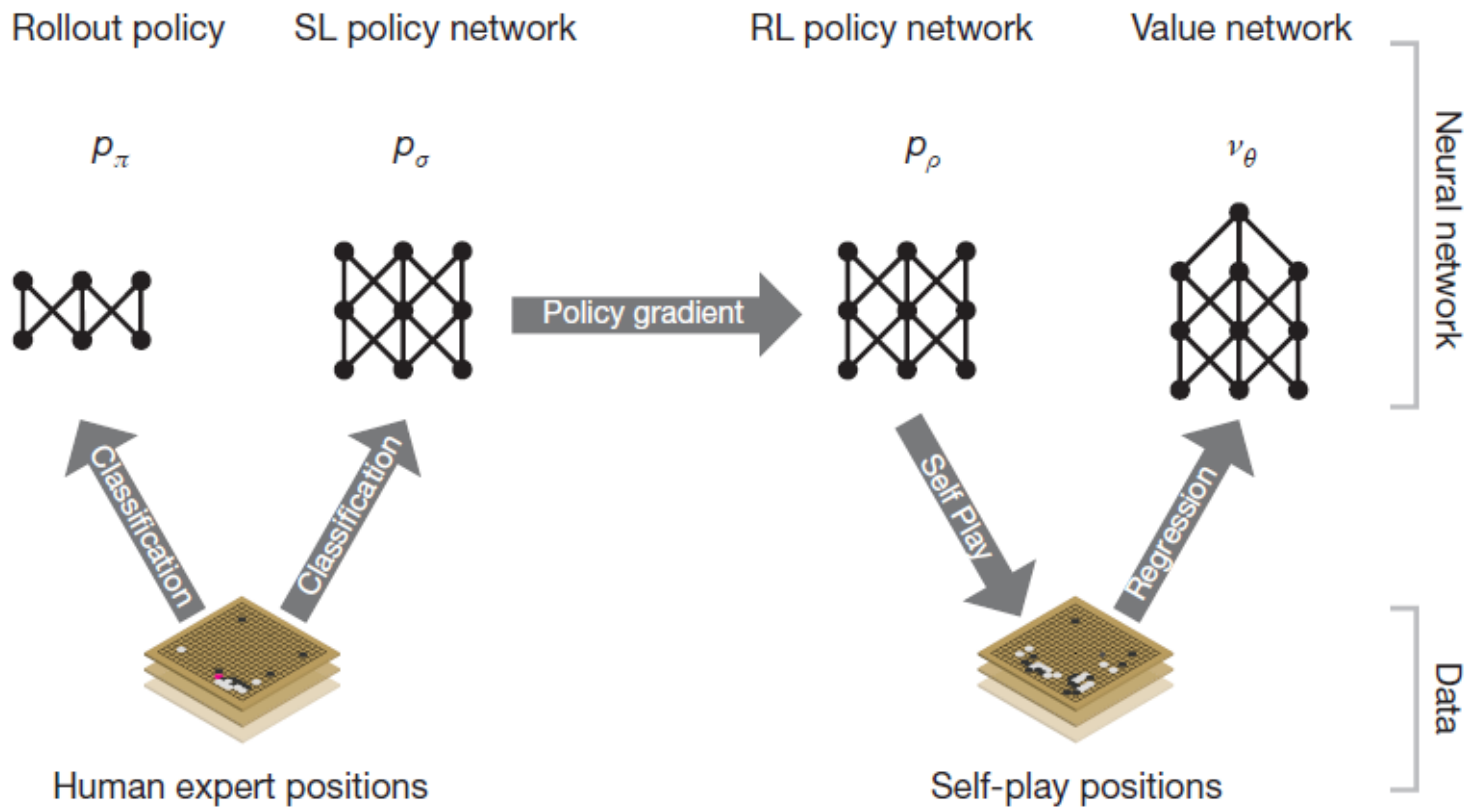
Policy network



Value network



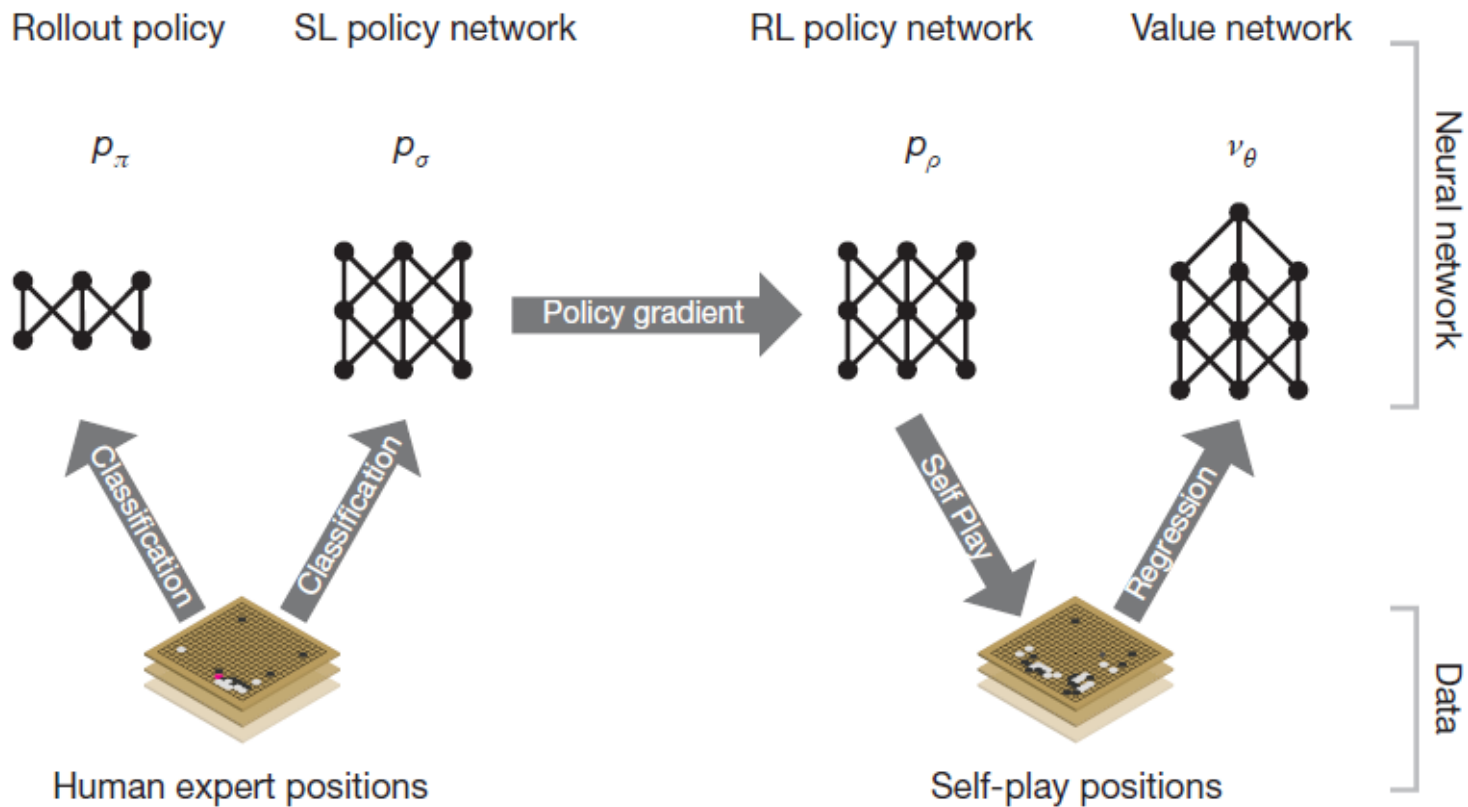




## 1. SL Policy

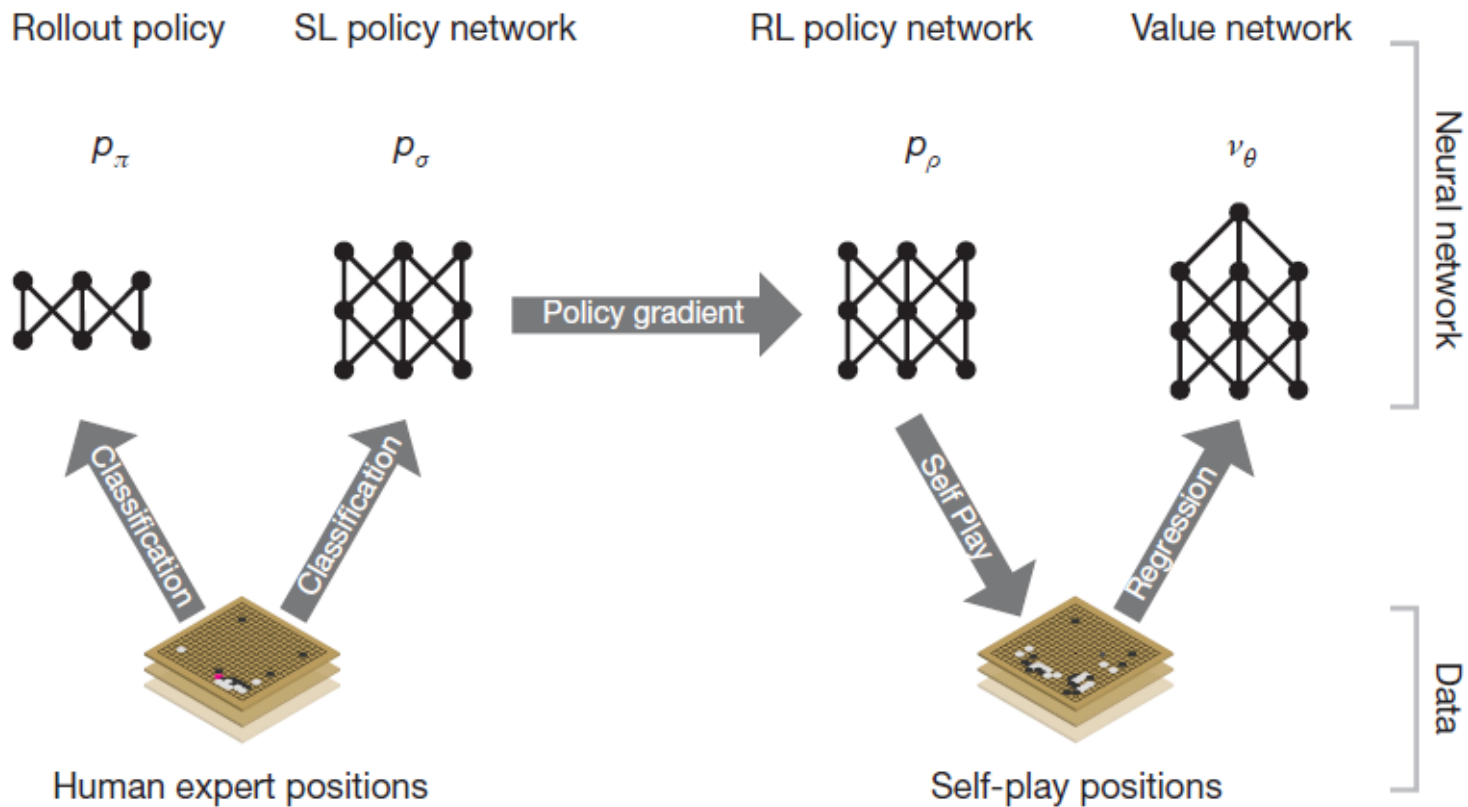
- Classification
- 13 레이어의 CNN 학습
- 인풋은  $19 \times 19 \times 48$
- 데이터 : 3천만수 (KGS 바둑 서버)
- 57% 정확도 달성 (바둑판 정보만 인풋으로 쓸 때는 55.7% 정확도)
- GPU 50개로 3주간 학습
- 3억 4천만 step





## 2. Rollout Policy

- 빠른 시뮬레이션을 위한 작은 네트워크
- 인풋은 Hand-crafted features
- 네트워크는 linear softmax
- 정확도는 24.2%
- 대신 action을 한번 선택하는데 필요한 시간은  $2\mu s$  (SL Policy는 3 ms)



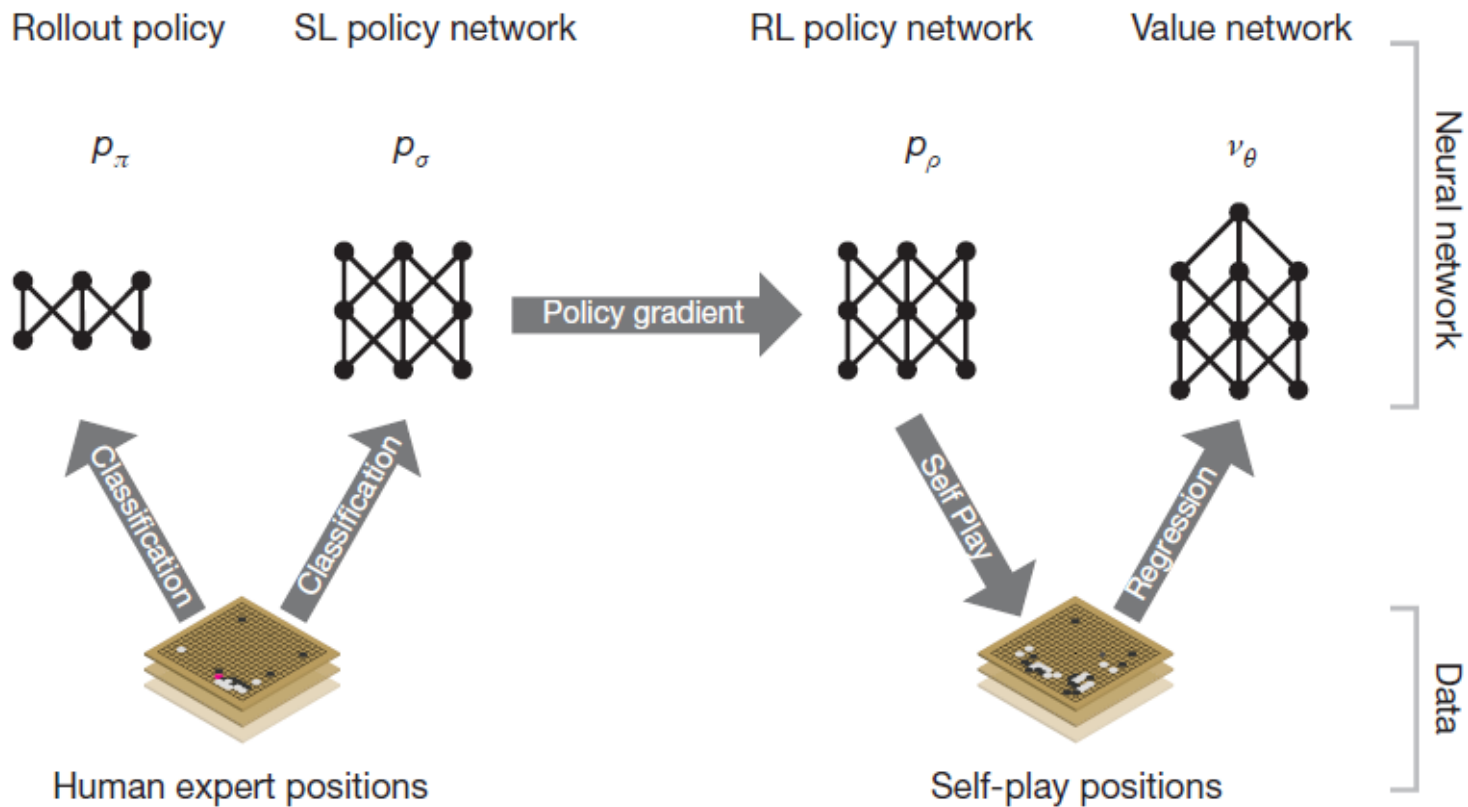
### 3. RL Policy

- SL Policy 와 동일한 형태의 네트워크
- 먼저 SL Policy의 weight로 초기화
- 리워드는 게임이 끝나는 시점에서만 주워지며, 이기면 +1, 지면 -1
- 알고리즘은 REINFORCE 알고리즘 사용

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

- RL vs SL ?  
RL 80% win
- GPU 50개로 1일간 학습

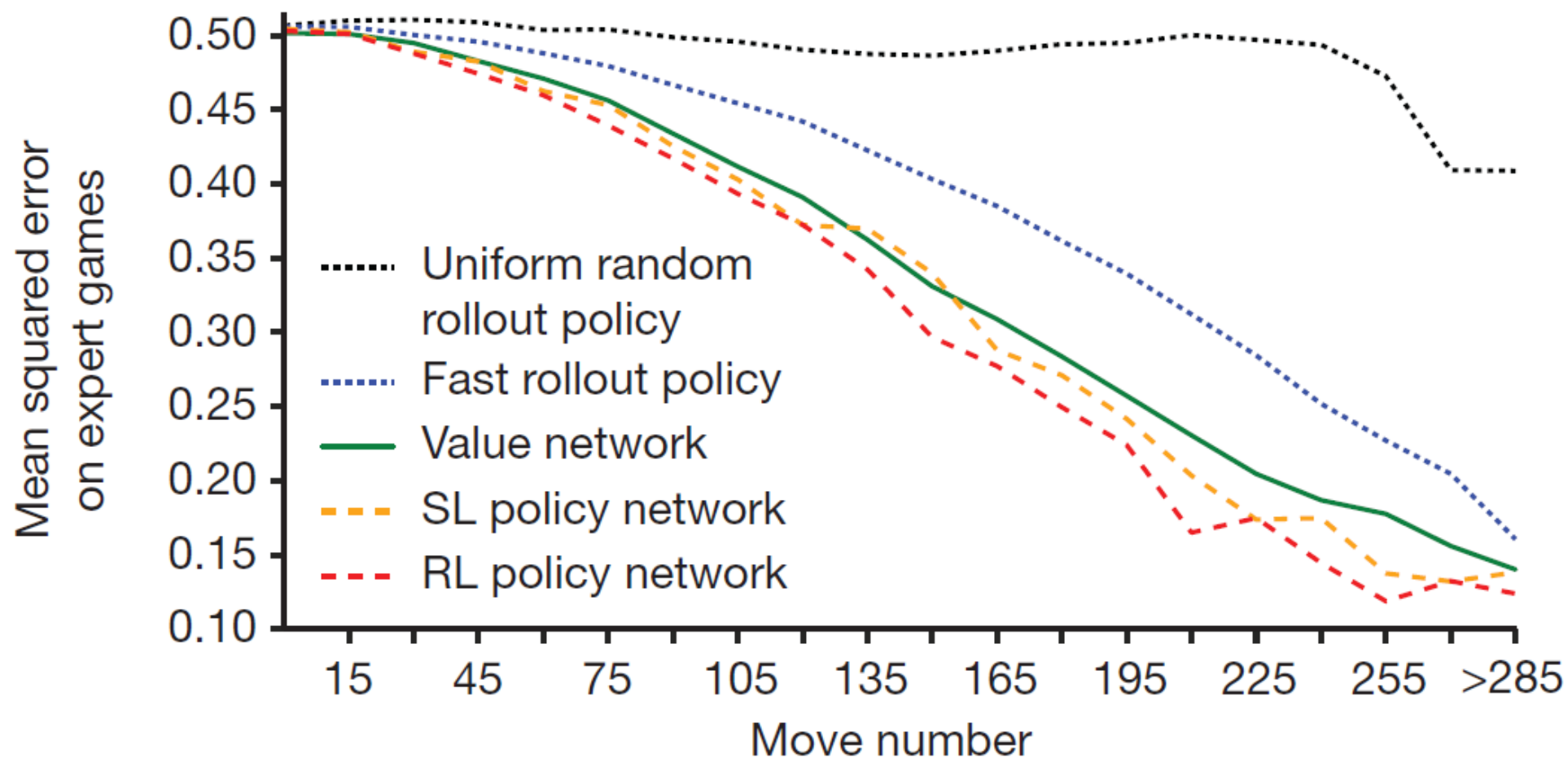




#### 4. Value Network

- RL policy를 따랐을 때에 누가 이길지 예측하는 네트워크  

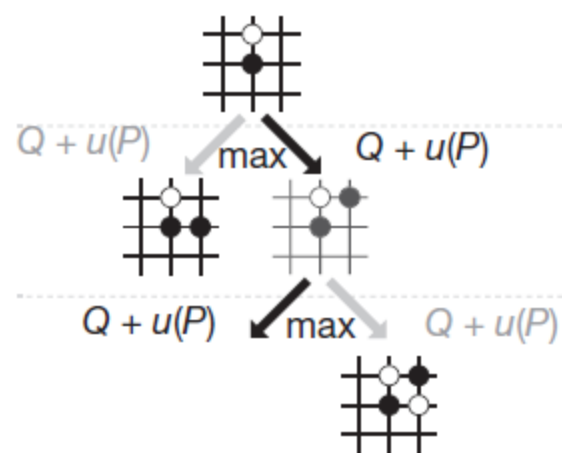
$$v^p(s) = \mathbb{E}[z_t | s_t = s, a_{t...T} \sim p]$$
- 아랫단은 SL 네트워크와 동일하나 윗단이 single output.
- 3천만 개의 상황으로부터 학습 (각각 다른 게임)
- GPU 50개로 1주일간 학습
- 5천만 mini-batch



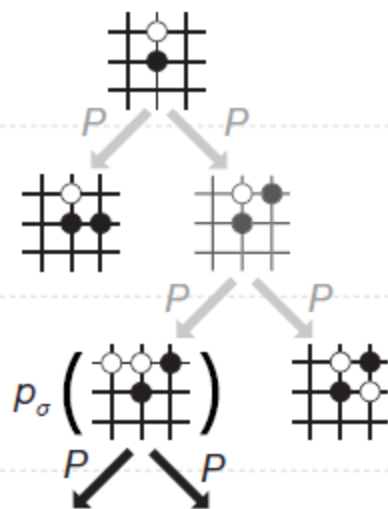
- RL policy를 이용한 rollout 보다 1만5천배 빠르는데, 비슷한 정확도를 보임

**a**

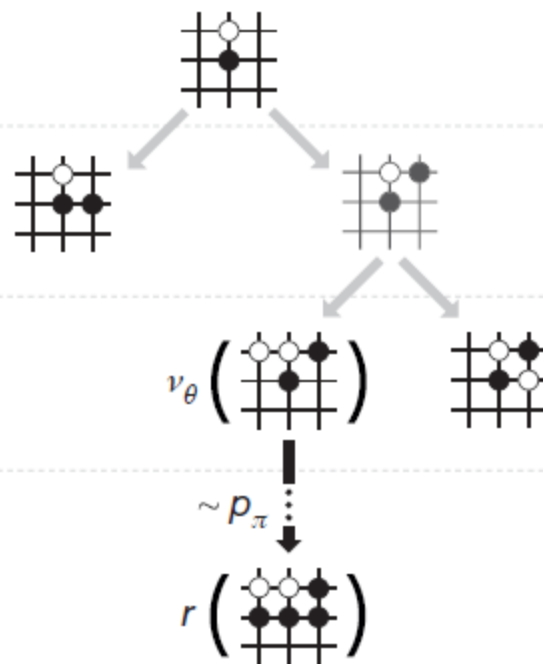
Selection

**b**

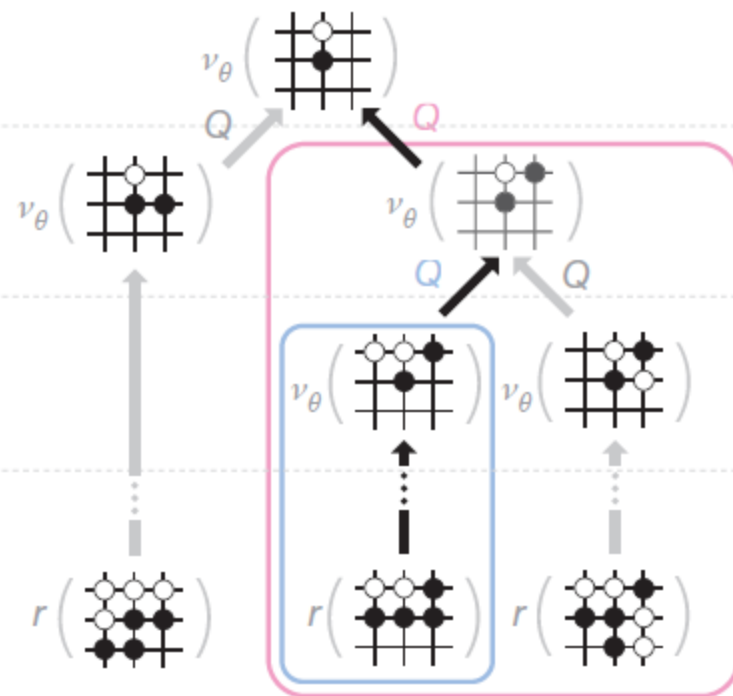
Expansion

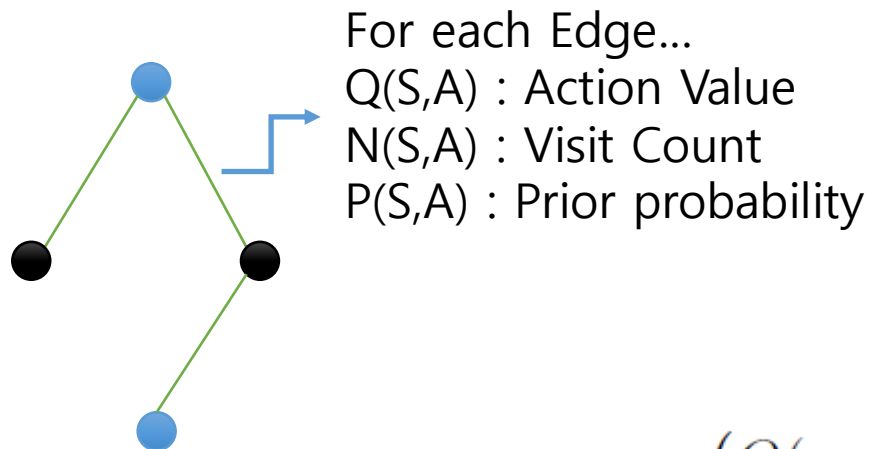
**c**

Evaluation

**d**

Backup





$$a_t = \underset{a}{\operatorname{argmax}} \left( \underbrace{Q(s_t, a)} + \underbrace{u(s_t, a)} \right)$$

$$Q(s, a) = \frac{1}{N(s, a)} \sum_{i=1}^n \mathbf{1}(s, a, i) \underbrace{V(s_L^i)}$$

$$u(s, a) \propto \frac{P(s, a)}{1 + N(s, a)}$$

$$V(s_L) = (1 - \lambda) \underbrace{v_{\theta}(s_L)} + \lambda \underbrace{z_L}$$

Value  
Network

롤아웃으로 게임을 끝까지 플레이 했을  
때의 결과