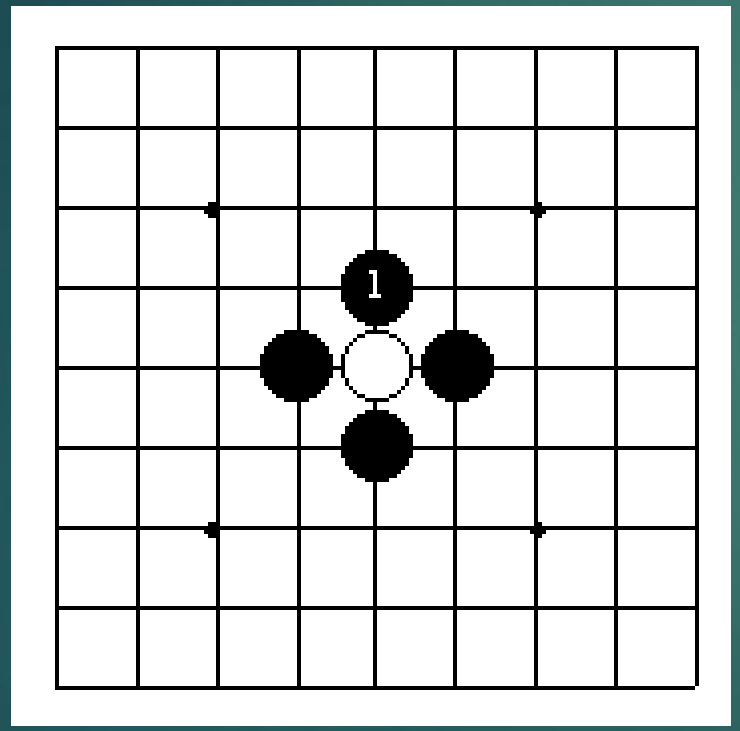
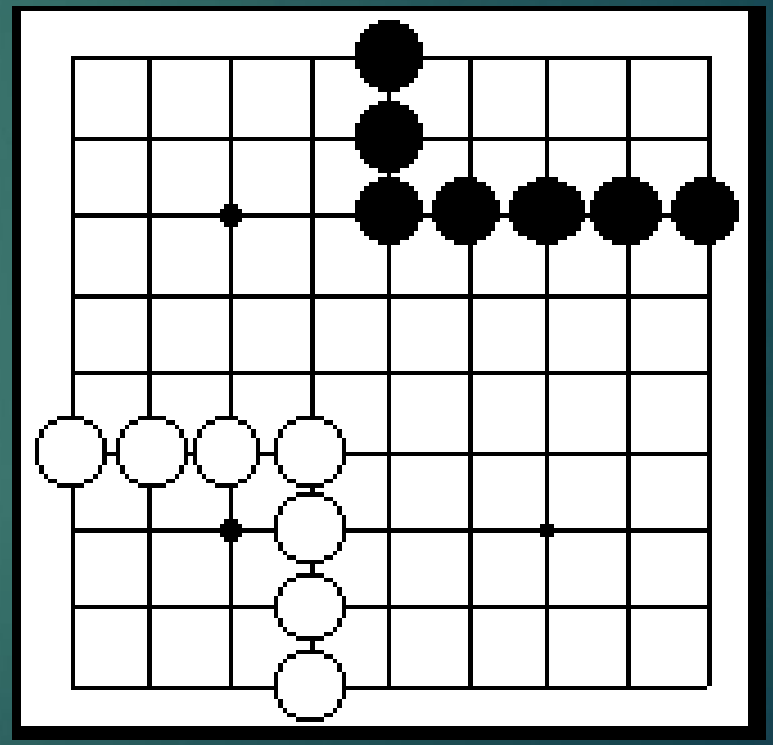
Mastering the game of Go with deep neural networks

АРИНА КОСОВСКАЯ

Правила игры в Go

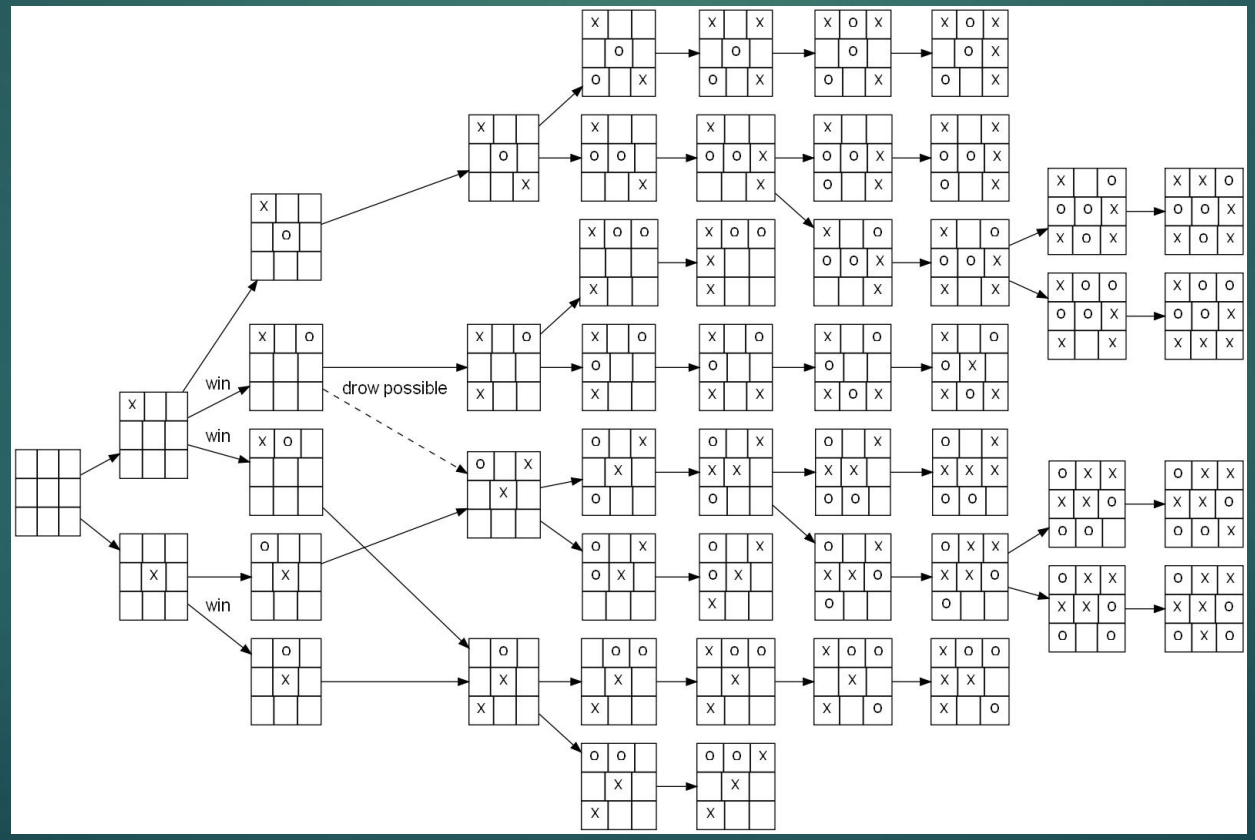
Главной целью игры является окружить как можно больше территории на поле 19x19



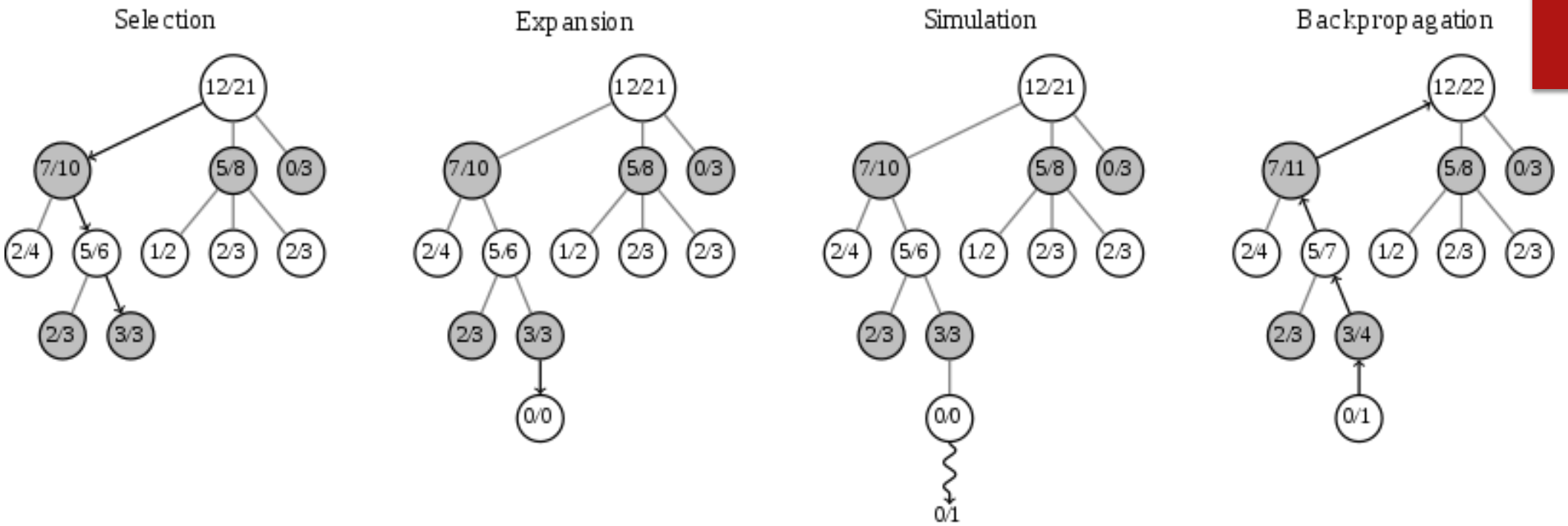


Проблема GO

- В отличие, например, от шахмат или шашек, GO имеет много вариантов для хода, и если в других играх можно построить дерево (Tree Search) действий, то в данной игре оно получится гигантским.



Пример дерева в игре крестики-нолики



Monte Carlo Tree Search

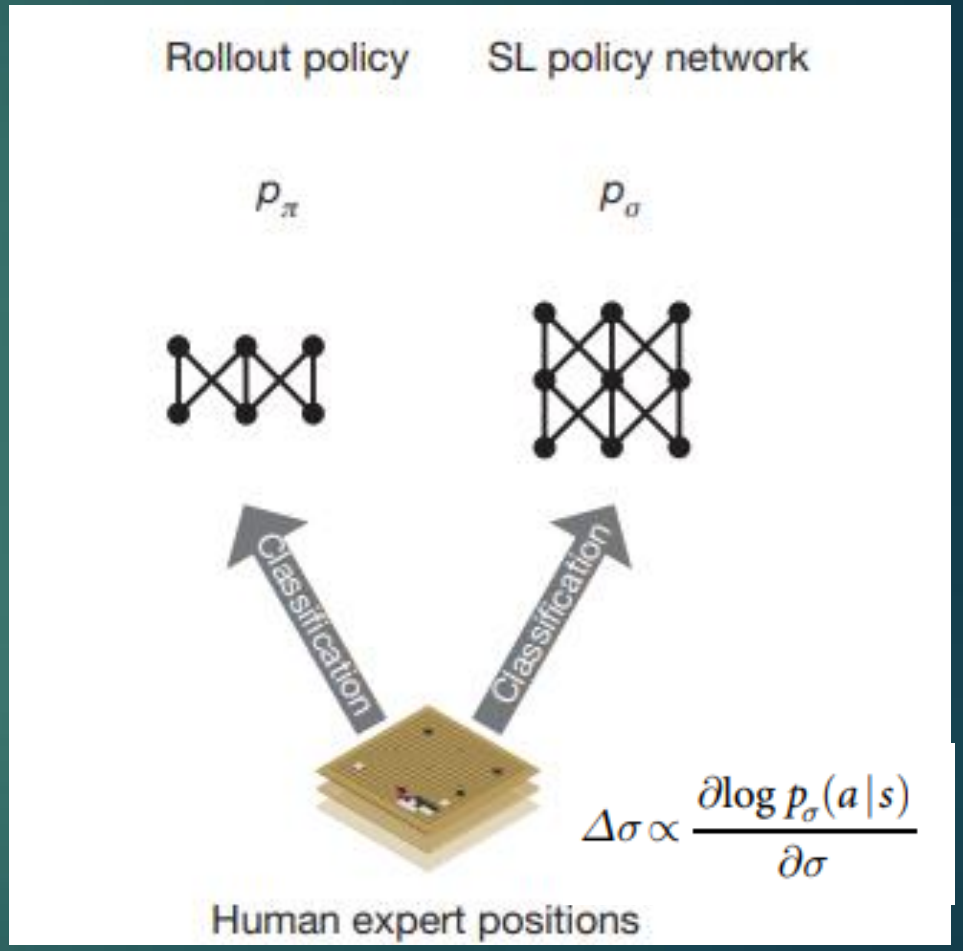
Архитектура Alpha Go

- SL policy network:

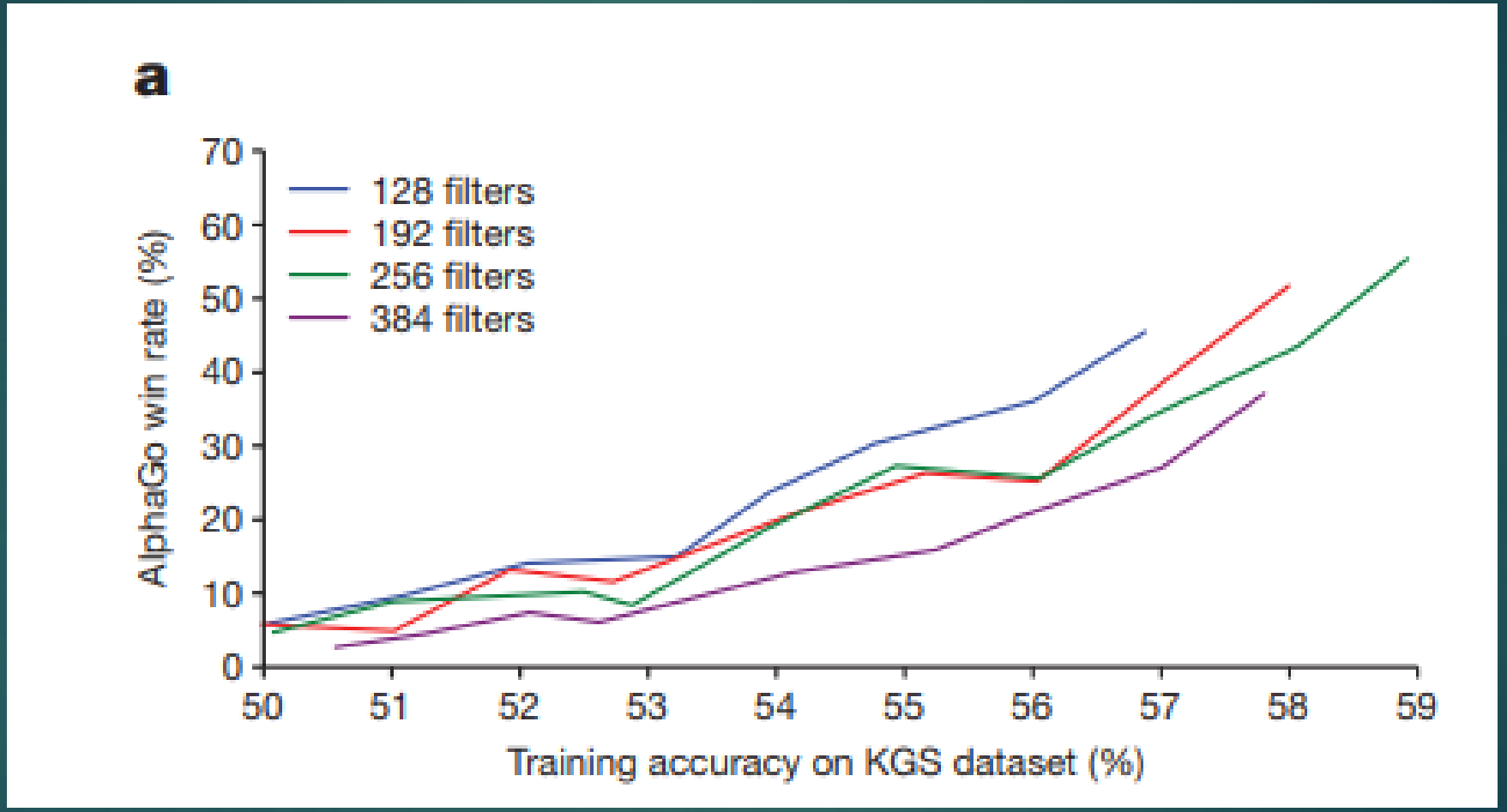
Нейронная сеть обучается на экспертных-человеческих играх с помощью максимизации правдоподобия, состоит из сверточных слоев. На вход подаются табличные признаки и картинки.

- Rollout policy:

Имеет небольшое количество линейных слоев делает то же самое, но работает быстрее. Кроме того подаются только табличные признаки.



Оптимальное кол-во фильтров для conv в SL policy network



- Входные данные для SL policy network:

Extended Data Table 2 | Input features for neural networks

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

- Входные данные для Rollout policy:

Feature	# of patterns	Description
Response	1	Whether move matches one or more response features
Save atari	1	Move saves stone(s) from capture
Neighbour	8	Move is 8-connected to previous move
Nakade	8192	Move matches a *nakade* pattern at captured stone
Response pattern	32207	Move matches 12-point diamond pattern near previous move
Non-response pattern	69338	Move matches 3×3 pattern around move

В итоге после обучения точность SL policy network составляла 57%, а точность Rollout policy 24.2%.
Скорость SL policy network равняется 3мс, а Rollout policy 2 нс.

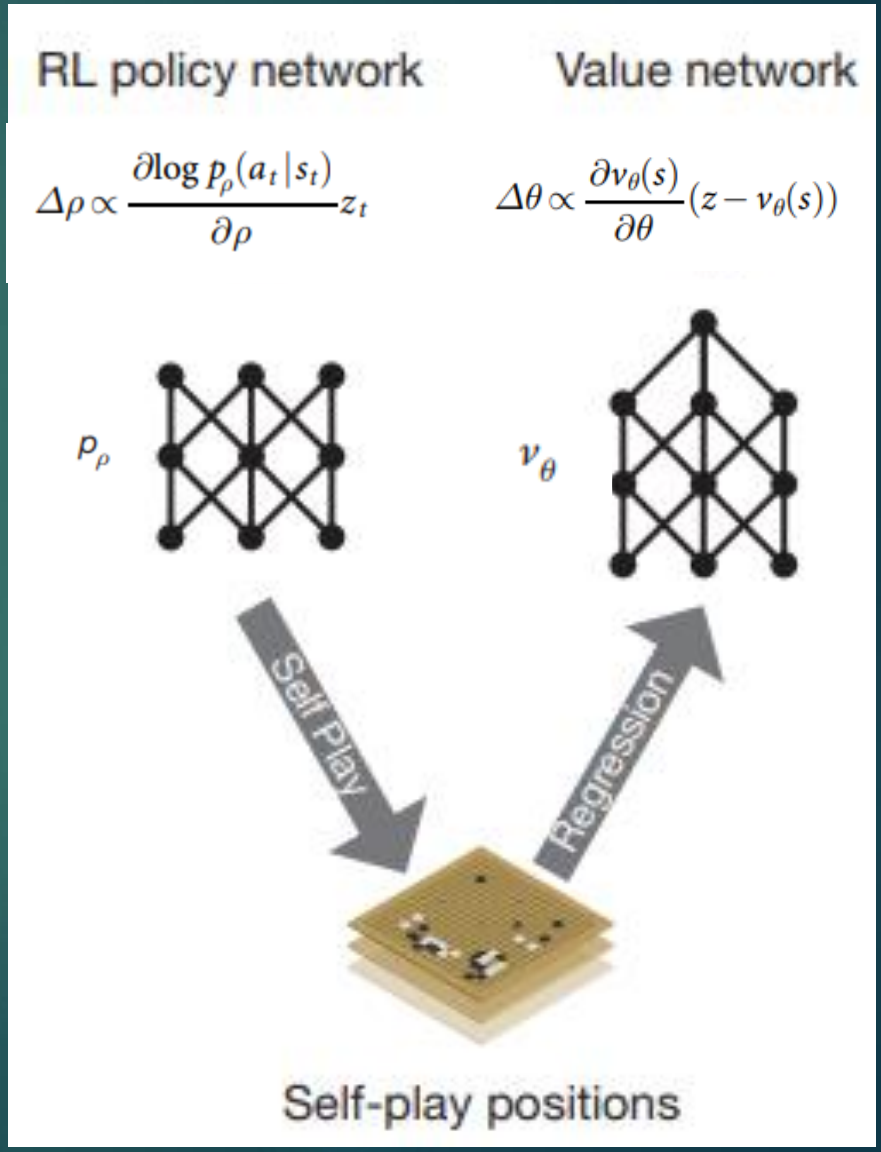
Архитектура Alpha Go

- RL policy network

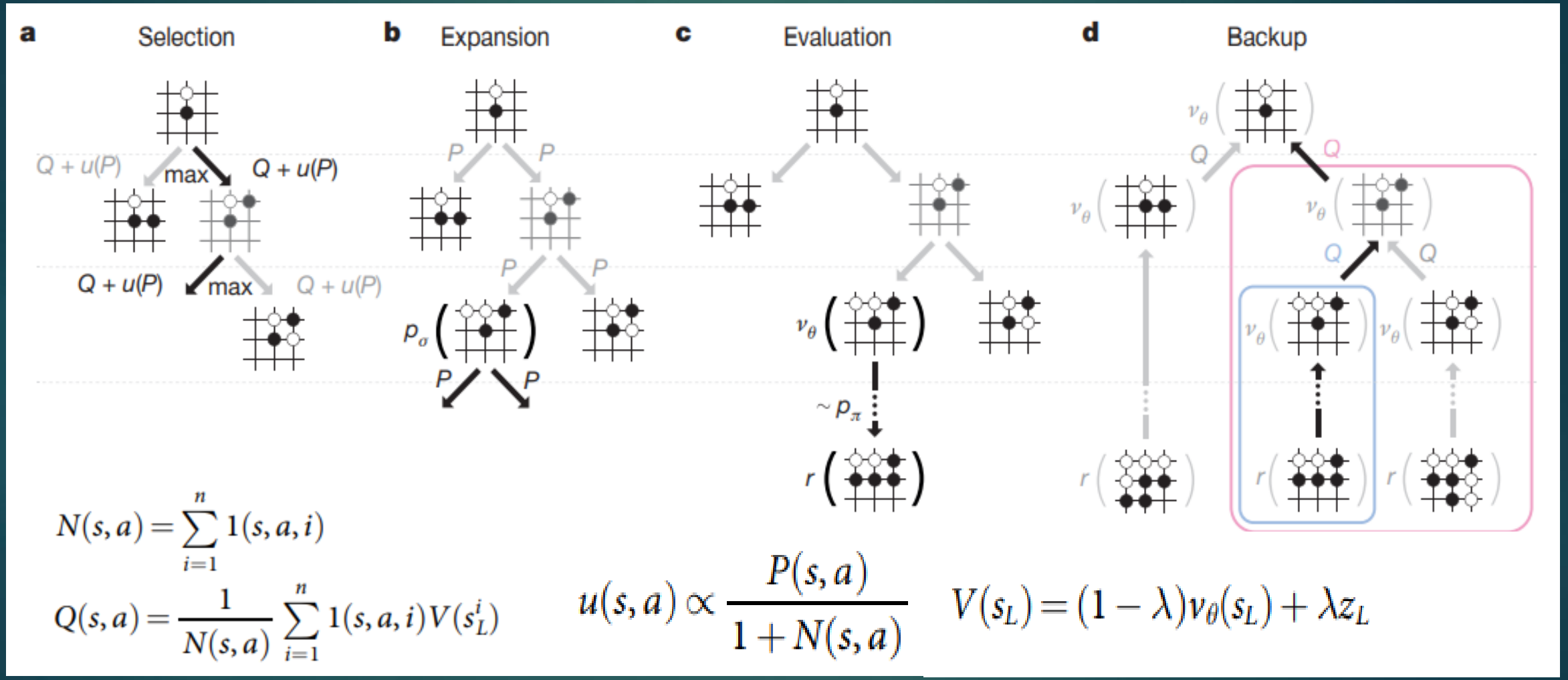
Инициализируем весами SL, сеть учится, играя сама с собой. (z = 1 если случилась победа, -1 если проигрыш). Веса обновляются по правилу policy gradient.

- Value network

Архитектура выдает одно число, учится на регрессии. Минимизируя mse (на парах состояние и результат), предсказывает число от -1 до 1 (где -1- проигрыш, 1 - выигрыш). Обучение происходит на данных, сгенерированных RL policy network.



Monte Carlo tree search in AlphaGo



$$N(s,a) = \sum_{i=1}^{n} 1(s,a,i)$$

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

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

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

Особенности Alpha Go Zero

- Тренируется не на партиях, которые играли эксперты
- Tree Search теперь используется во время тренировки модели
- Получает на вход только положение камней