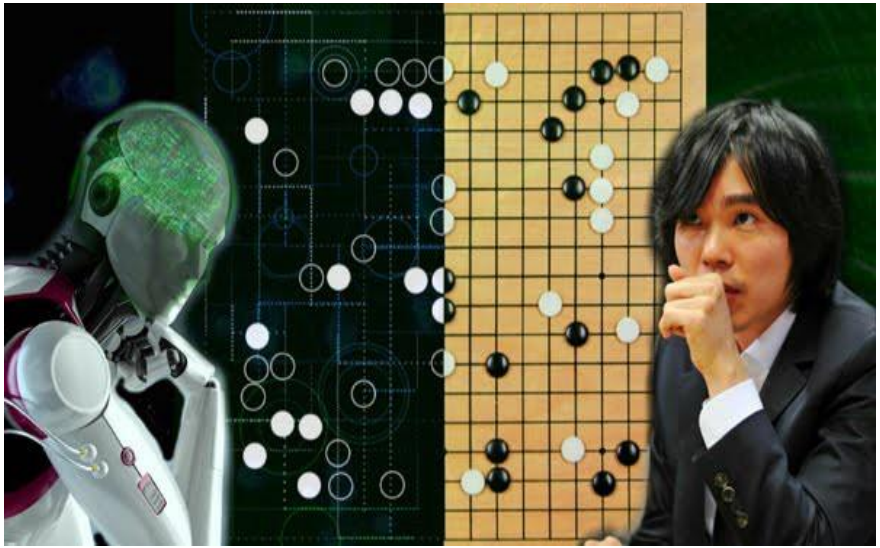
# От Го до AlphaGo



**Мизгирева Юлия**

# Го (вэйцы, бадук)

- Другие названия игры:
дзаин («сидящий отшельник») и ран-ка («сгнившее топорище»)

- Возникла 2500-5000 лет назад в Китае (?)
- ~ VII век н.э. -- была завезена в Японию
- 2 половина XIX века -- из Японии завезена в Европу (получила популярность во 2 половине XX века, в России -- с1965 года)

# Правила

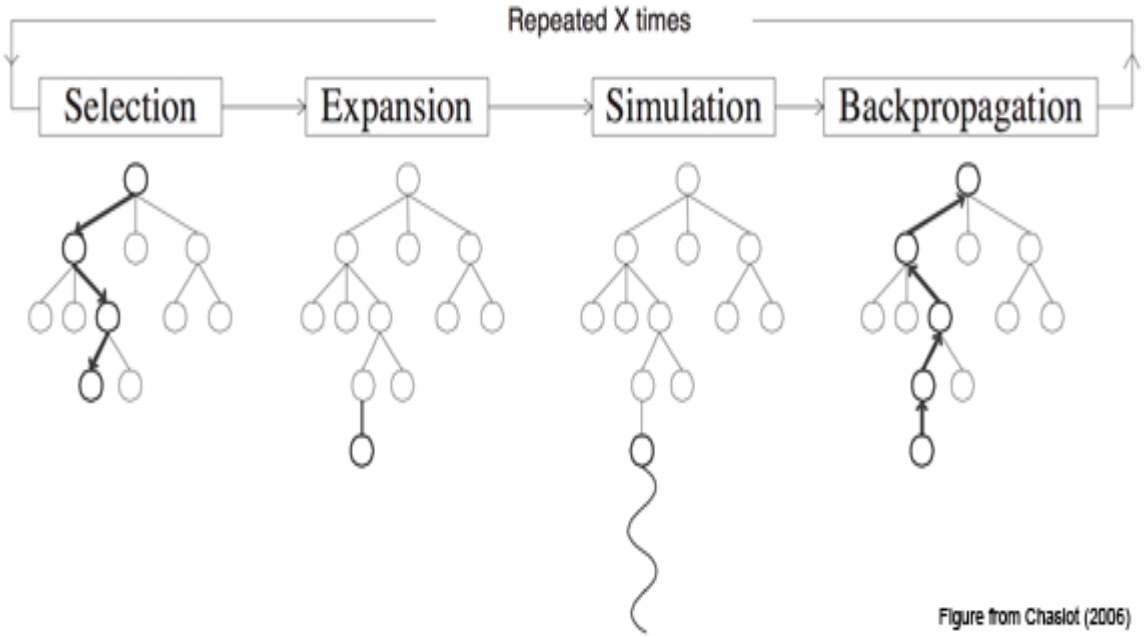
- Доска 19x19 (181 черный камень + 180 белых)
- Камни ставятся на пересечения
- Черные ходят первыми, затем по очереди (можно пасовать – пас считается ходом)
- Цель – захват территории, большей, чем у противника
- Захваченные камни снимаются с доски
- Правило ко – запрет на повторение позиции
- Самоубийственные ходы запрещены
- Территория считается по количеству свободных пересечений внутри живой группы игрока + пленные камни
- Игра заканчивается после двух пасов подряд

# Сложность компьютерной реализации

- Количество вариантов дальнейшего хода исчисляется сотнями
- Камни образуют группы (не всегда очевидно, как отдельный ход повлияет на исход игры)

# Monte Carlo Tree Search (MCTS)

- В каждой ноде сохраняется value (~вероятность победы) и количество раз посещения ноды



# AlphaGo

- MCTS + нейросетки
- Нейросетки используются для оценки value каждой ноды дерева
- 2 типа нейросеток:
    - Policy networks (2 нейросетки -- “медленная” и “быстрая”): состояние доски -> какой бы ход сделал человек?  (supervised learning + последующий reinforcement learning)
    - Value network: состояние доски -> вероятность выиграть/проиграть (число от -1 до 1)
- “Медленная” -- для поиска следующей ноды в MCTS; “быстрая” – для симуляций, чтобы оценить value следующей ноды.
- Value network без симуляций оценивает value следующей ноды. Итоговое value – взвешенная сумма этих двух посчитанных ранее value.

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

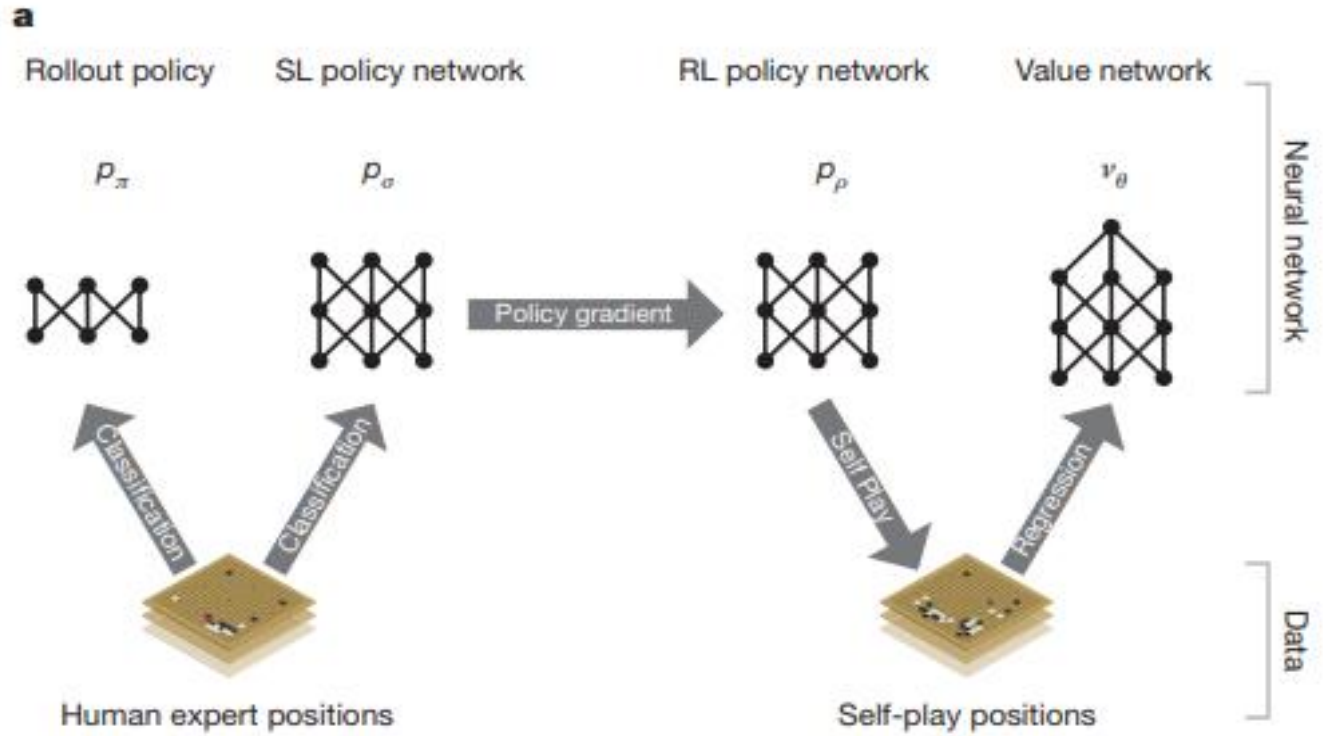
# Features

**Extended Data Table 2 | Input features for neural networks**

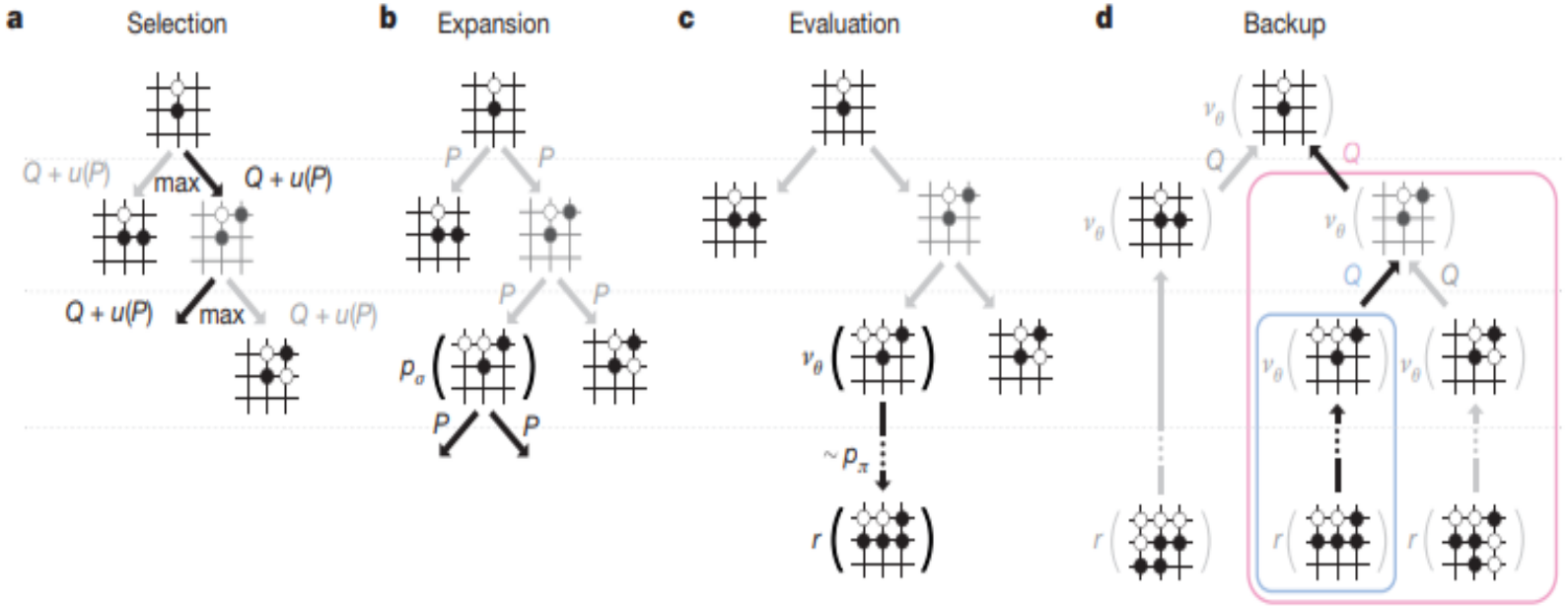
| Feature | # of planes | Description |
|---|---|---|
| Stone colour | 3 | Player stone / opponent stone / empty |
| Ones | 1 | A constant plane filled with 1 |
| Turns since | 8 | How many turns since a move was played |
| Liberties | 8 | Number of liberties (empty adjacent points) |
| Capture size | 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 |

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

# Схематическая архитектура AlphaGo



# Monte Carlo Tree Search в AlphaGo



# AlphaGo vs Ли Седоль (9 дан)

- Март 2016
- 5 партий: 4:1
- "I, Lee Sedol, lost, but mankind did not"

# AlphaGo Zero

- Отличия от AlphaGo:
  - Для обучения не использовались партии профессиональных игроков (только self-play);
  - Не использовались заранее рассчитанные фичи;
  - MCTS использовался в фазе обучения;
  - Policy network и Value network объединены в одну нейросеть;
  - Подход легко обобщается и для других игр с полной информацией (шахматы, сёги…).

Mastering the game of Go without human knowledge (2017)