Mastering the game of Go with deep neural networks

Петрович Сергей, БПМИ181

Факультет компьютерных наук
Высшая школа экономики

02.03.2021

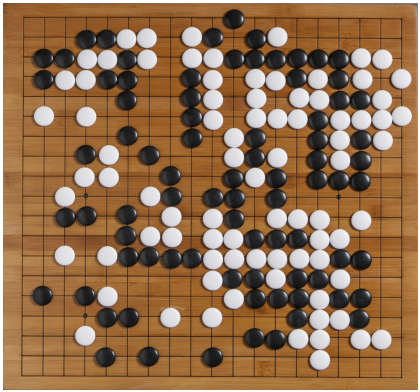
План

1. Что такое игра Go
2. Почему с ней столько шума
3. Архитектура AlphaGo
4. Модификации AlphaGo

Игра Go

Игра Go это:

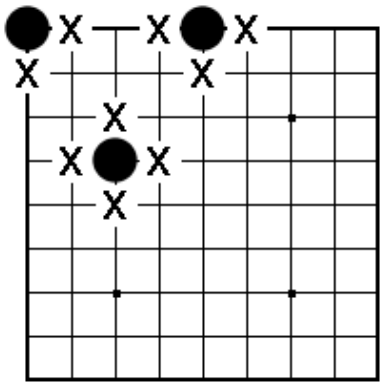
- Стратегическая настольная игра родом из Древнего Китая
- Входит в число пяти базовых дисциплин Всемирных интеллектуальных игр
- Одна из наиболее распространённых настольных игр на Земле



Игра Go

Правила в Go:

- 2 игрока
- Поле размера 19×19
- 361 камень (180 белых и 181 черный)
- Ходы делаются по очереди (первыми ходят черные), но можно пассовать
- Каждый камень должен иметь хотя бы одну степень свободы, иначе он захвачен противником
- Цель игры: отгородить на игровой доске камнями своего цвета большую территорию, чем противник



Исторический лог

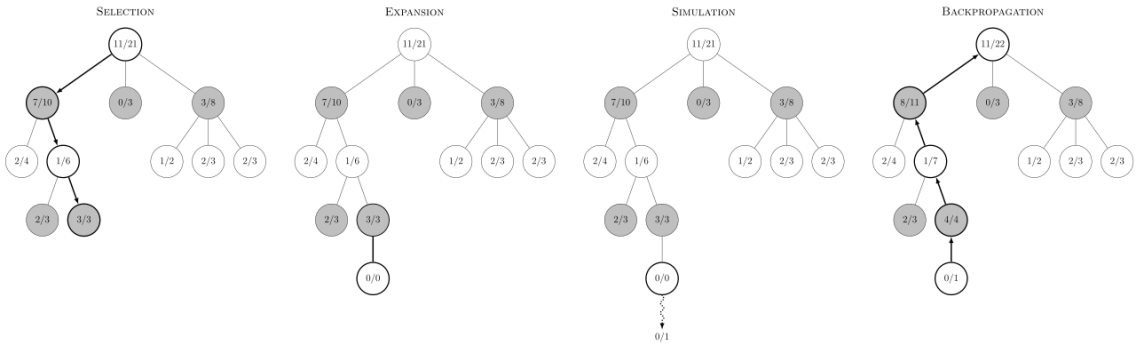
- *1994 год* - чемпион мира по шашкам обыгран программой
- *1997 год* - чемпион мира по шахматам обыгран программой
- ...
- *2006 год* - программа начала выигрывать у юношеских чемпионов на доске 9×9 с несколькими камнями форы
- *2016 год* - чемпион мира по Go обыгран программой **AlphaGo**, созданной в Google Deepmind

Почему столько шума

Почему игра Go считается такой сложной?

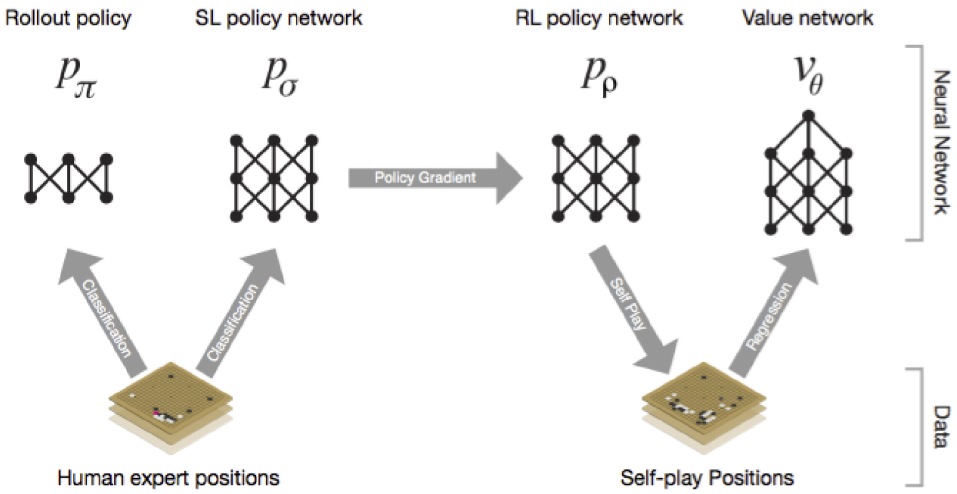
- Большое игровое поле $19 \times 19 = 361$ (в шахматах всего 8×8)
- В среднем около 150 разумных ходов на каждой стадии (в шахматах в среднем около 20 на каждой стадии)
- Длинные партии - в Go партия длится примерно в среднем около 150 ходов (в шахматах около 50 ходов)
- Практический невозможно понять, насколько хороший ход

Monte Carlo tree search



В качестве следующего шага выбирается вершина с максимальным значением выражения: $\frac{w_i}{n_i} + c\sqrt{\frac{\ln N_i}{n_i}}$, где w_i - число побед в i-ом узле, n_i - число симуляций в нем, а N_i - число симуляций в родительском узле.

AlphaGo


Rollout policy
SL policy network
RL policy network
Value network
p_π
p_σ
p_ρ
v_θ
Policy Gradient
Neural Network
Classification
Classification
Self Play
Regression
Data
Human expert positions
Self-play Positions

AlphaGo: Supervised Learning - Архитектура

Цель: научиться предсказывать следующий ход игрока, обучаяся на играх профессионалов

Реализация - две нейронные сети:

- "медленная" **SL policy network (σ)**: сверточные + линейные слои, через softmax выдает распределение вероятностей следующего хода. Обучается с помощью SGD, максимизирую логправдоподобие. Обучалась на всех возможных входных признаках
- "быстрая" **Rollout policy (π)**: несколько линейных слоев с выходным softmax. Обучалась так же, но только на подготовленных шаблонных признаках

AlphaGo: Supervised Learning - Результаты

Результаты:

- **Точность**: accuracy = 57% на отложенной выборке у **SL policy network** и accuracy = 24.2% на отложенной выборке у **Rollout policy**
- **Время**: 2 нс для **SL policy network** и 3 мс для **Rollout policy**!

AlphaGo: Reinforcement Learning - Архитектура

Реализация - тоже две сети:

- **RL policy network (ρ)**: в точности такая же сеть, как **SL policy network (σ)**, и инициализируется ее весами. Обучалась следующим образом:
  - Играет партии со случайно выбранной предыдущей версией себя
  - Обновляет веса по правилу policy gradient:

$$\Delta\rho \propto \nabla_\rho \log p(a_t|s_t, \rho) \cdot z_t$$

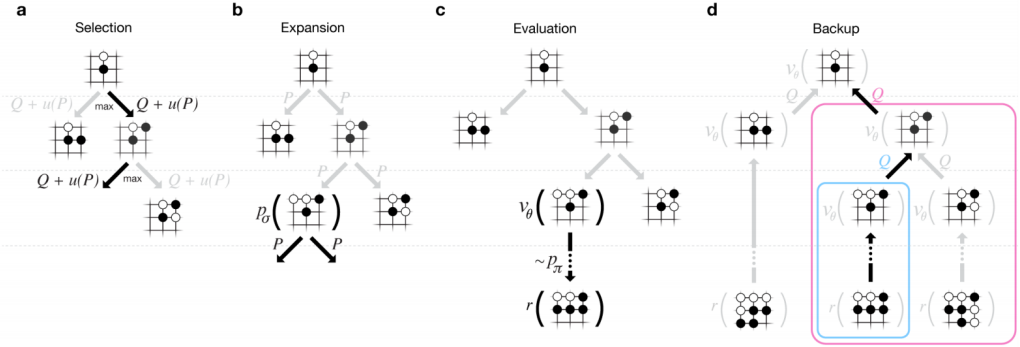
  - $z_t \in \{+1, -1\}$; +1 - если партия выиграна, -1 - если проиграна
- **Value network (θ)**: такая же сеть, но для регрессии. Выдает оценку позиции (число от -1 до +1). Обучалась с помощью SGD на MSE:

$$\Delta\theta \propto \nabla_\theta v_\theta(x) \cdot (z - v_\theta(s))$$

AlphaGo: Reinforcement Learning - Результаты

Выиграш **RL policy network** примерно в 85% случаях в играх с лучшими программами, основанными на MCTS.

AlphaGo: Применение



$$a_t = argmax_a(Q(s_t, a) + u(s_t, a)) \;\; ; \;\; u(s, a) \propto \frac{P_\sigma(s, a)}{1 + N(s, a)}$$

$$V(s_L) = (1 - \lambda)v_\theta(s_L) + \lambda z_L \;\; ; \;\; Q(s, a) = \frac{1}{N(s, a)} \sum_{i=1}^{n} I[s, a, i] \cdot V(s_L^i)$$

AlphaGo: что дальше?

Проблемы и критика:

- Обучение на играх людей
- Искусственно сгенерированные признаки
- Нужны большие вычислительные мощности

AlphaGo Zero

Усовершенствования:

- Обучается исключительно на играх сама с собой, инициализируя веса случайно
- Отсутсвуют синтетические признаки, модель получает исключительно положения камней (но не совсем)
- Все 4 нейронные сети заменяются всего лишь 1, которая умеет предсказывть распределение вероятности на ходах и ценность состояния
- MCTS теперь применяется и во время обучения

Источники

- Go (game): https://en.wikipedia.org/wiki/Go_(game)
- AlphaGo: https://en.wikipedia.org/wiki/AlphaGo
- AlphaGo Zero: https://en.wikipedia.org/wiki/AlphaGo_Zero
- Monte Carlo tree search: https://en.wikipedia.org/wiki/Monte_Carlo_tree_search
- Mastering the game of Go with deep neural networks and tree search: https://www.researchgate.net/publication/292074166_Mastering_the_game_of_Go_with_deep_neural_networks_and_tree_search
- AlphaGo Zero совсем на пальцах: https://habr.com/ru/post/343590/
- Николенко С., Кадурин А., Архангельская Е., «Глубокое обучение. Погружение в мир нейронных сетей», Спб: Питер, 2020