AlphaGo

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

Prepared by Nikita Sergeev,
DCAM, MIPT, 2019

Agenda

1. Problem setting
2. Policy and value networks
3. Features and architectures
4. Search algorithm

Problem setting

Game of perfect information:

- State space
- Action space
- A state transition function
- Policy
- Value function

$$v^p(s) = \mathbb{E}[z_t | s_t = s, a_{t \ldots T} \sim p]$$

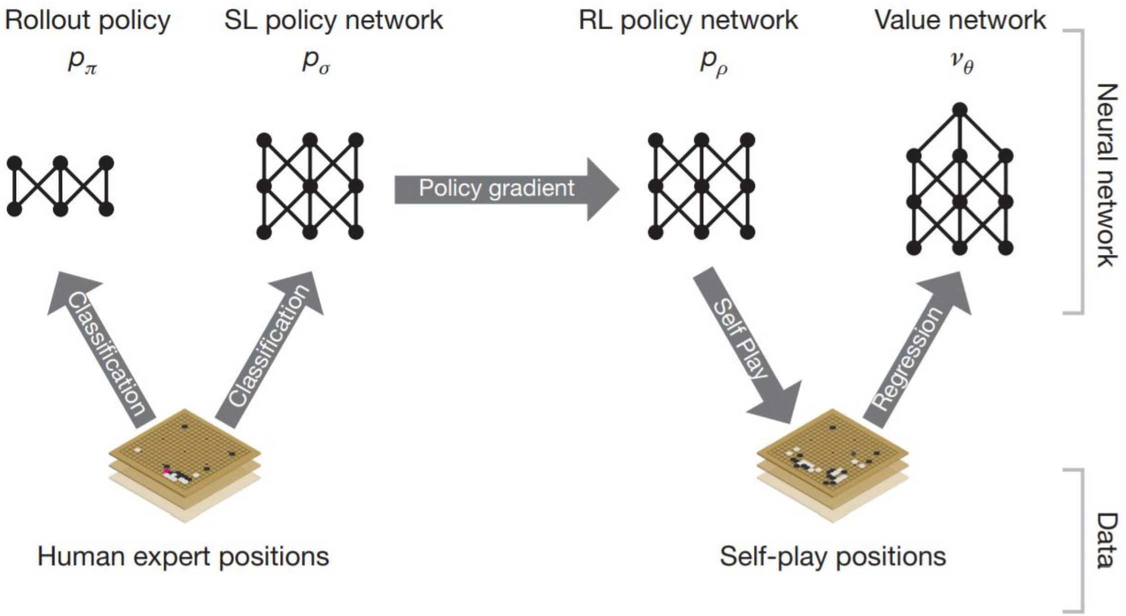
$$v^*(s) = \begin{cases} z_T & \text{if } s = s_T, \\ \max\limits_a - v^*(f(s,a)) & \text{otherwise} \end{cases}$$

Policy network: classification

- Train-test split - 1mil: 28 mil;
- Position - state and human action;
- Mini-batch gradient descent;
- 3 weeks of training

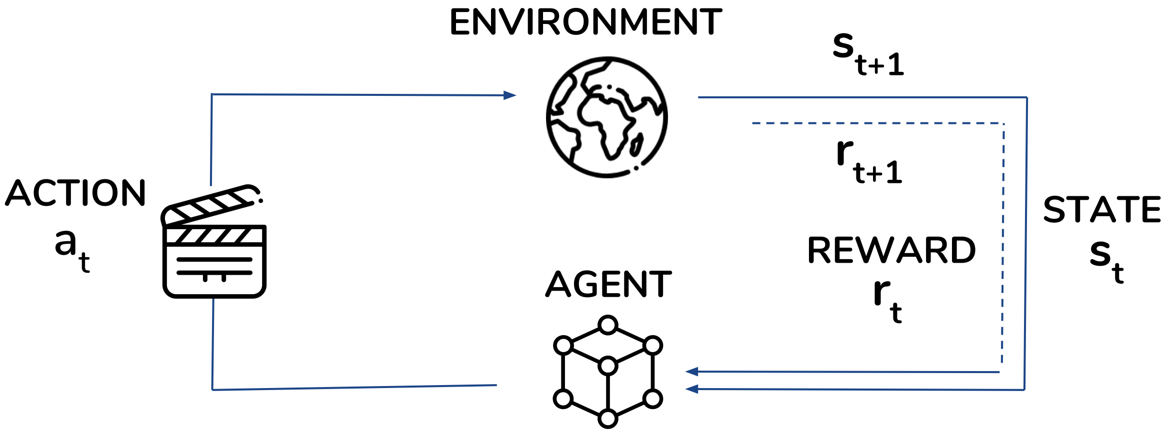
$$\Delta\sigma = \frac{\alpha}{m}\sum_{k=1}^{m}\frac{\partial \log p_\sigma(a^k|s^k)}{\partial\sigma}$$

Rollout policy



- Linear softmax policy;
- Much more features than in SL policy network

Policy network: reinforcement learning



$$\Delta\rho = \frac{\alpha}{n}\sum_{i=1}^{n}\sum_{t=1}^{T^i}\frac{\partial \log p_\rho(a_t^i|s_t^i)}{\partial\rho}(z_t^i - v(s_t^i))$$

Value network

- Artificial datasets to prevent overfitting
- Learning only using single training example
- 1 extra feature

$$\Delta\theta = \frac{\alpha}{m}\sum_{k=1}^{m}\left(z^k - v_\theta(s^k)\right)\frac{\partial v_\theta(s^k)}{\partial\theta}$$

$$v^{p_\rho}(s_{U+1}) = \mathbb{E}[z_{U+1} | s_{U+1}, a_{U+1,\ldots T} \sim p_\rho]$$

Features

Input features for neural networks

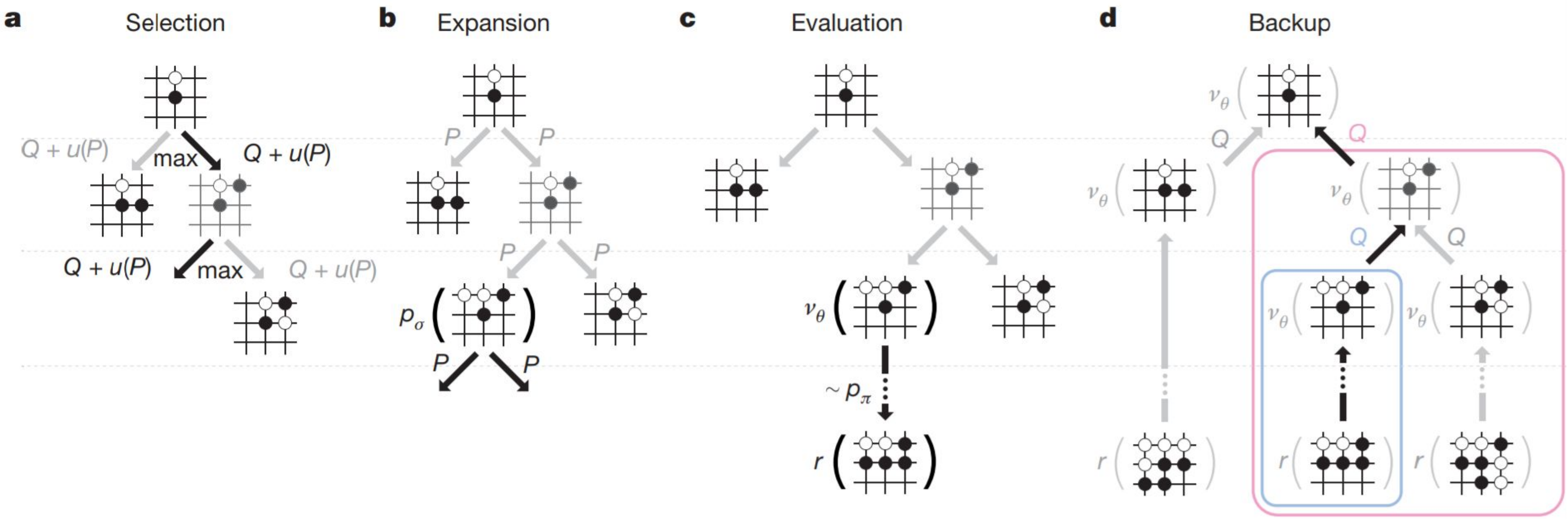
Feature	# of patterns	Description
Stone colour	3	Player stone / opponent stone / empty
Ones	1	A constant plane filled with 1
Turns science	8	How many turns since a move was played
Liberties	8	Number of liberties (empty adjacent points)
Capture size	8	How many opponent stones would be captured
Self-atari 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

Features

Input features for rollout and tree policy

Feature	# of patterns	Description
Response	1	Whether move matches one or more response pattern 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
Self-atari	1	Move allows stones to be captured
Last move distance	34	Manhattan distance to previous two moves
Non-response pattern	32207	Move matches 12-point diamond pattern centred around move

Search algorithm



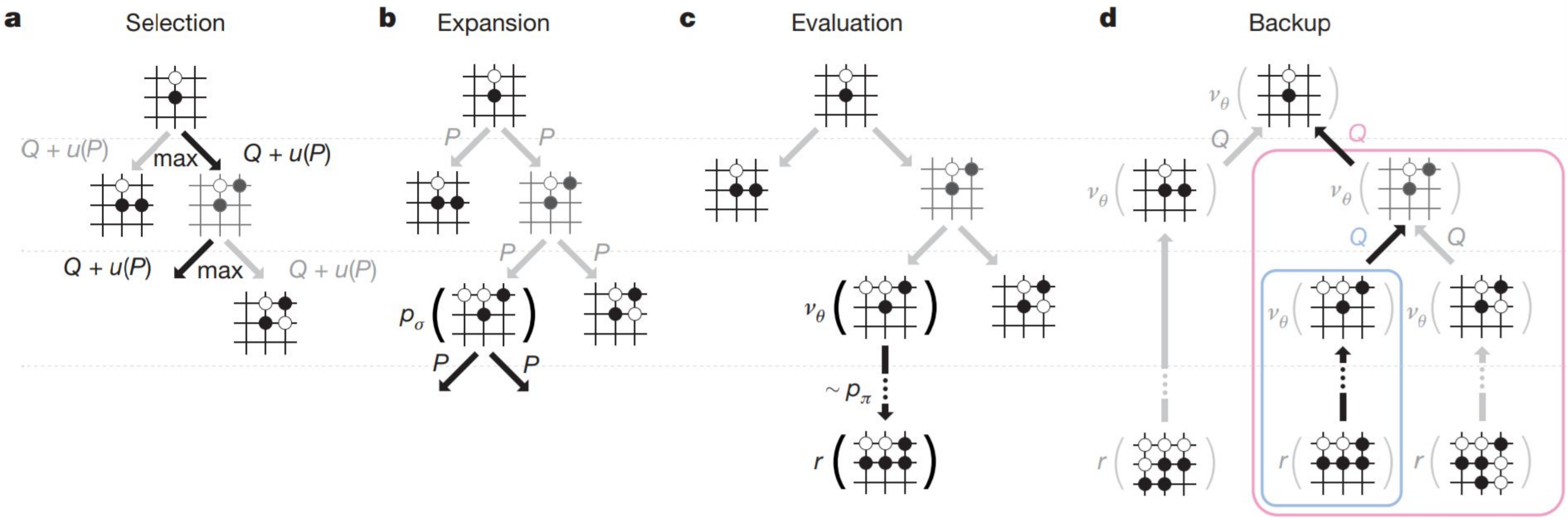
Selection

$$\{P(s,a),\ \ N_v(s,a),\ \ N_r(s,a),\ \ W_v(s,a),\ \ W_r(s,a),\ \ Q(s,a)\}$$

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

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

Search algorithm



Backup

$$N_r(s_t, a_t) \leftarrow N_r(s_t, a_t) + n_{vl}; W_r(s_t, a_t) \leftarrow W_r(s_t, a_t) - n_{vl}$$

$$N_r(s_t, a_t) \leftarrow N_r(s_t, a_t) - n_{\mathrm{vl}} + 1; \; W_r(s_t, a_t) \leftarrow W_r(s_t, a_t) + n_{\mathrm{vl}} + z_t$$

$$N_v(s_t, a_t) \leftarrow N_v(s_t, a_t) + 1, \; W_v(s_t, a_t) \leftarrow W_v(s_t, a_t) + v_\theta(s_L)$$

$$Q(s, a) = (1 - \lambda)\frac{W_v(s, a)}{N_v(s, a)} + \lambda\frac{W_r(s, a)}{N_r(s, a)}$$

Literature used

- https://habr.com/ru/post/343590/
- https://habr.com/ru/post/279071/
- https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf
- https://habr.com/ru/post/282522/
- https://habr.com/ru/post/330092/
- https://www.slideshare.net/KarelHa1/alphago-mastering-the-game-of-go-with-deep-neural-networks-and-tree-search
- https://becominghuman.ai/summary-of-the-alphago-paper-b55ce24d8a7c
- https://medium.com/@karpathy/alphago-in-context-c47718cb95a5
- https://deepmind.com/blog/alphago-zero-learning-scratch/
- https://www.nature.com/articles/nature24270.epdf?author_access_token=VJXbVjaSHxFoctQQ4p2k4tRgN0jAjWel9jnR3ZoTv0PVW4gB86EEpGqTRDtplz-2rmo8-KG06gqVobU5NSCFeHILHcVFUeMsbvwS-lxjqQGg98faovwjxeTUgZAUMnRQ
- https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf

THANK YOU
FOR LISTENING