AlphaGoStop

Applying Deep Reinforcement Learning for Playing Go-Stop

Team 36 (Keonwoo Kim, 20170058)

Problem Setting

Go-Stop + AlphaZero

Main Goal

+

AlphaZero

How to Play Go-Stop

Cards

Jan.

Feb.

Mar.

Apr.

⋮

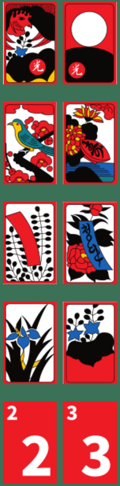
Bright …

Animal …

Ribbon …

Junk …

Bonus



How to Play Go-Stop

Turns

How to Play Go-Stop

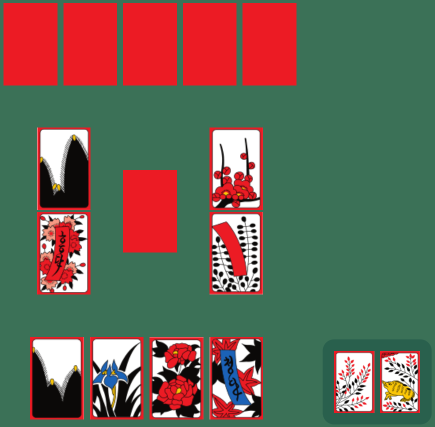
Turns

How to Play Go-Stop

Turns

How to Play Go-Stop

Turns



Motivation of Game Choice

- Popularity
- Non-triviality (game is not too small)
- Randomness and hidden (incomplete) information

AlphaZero

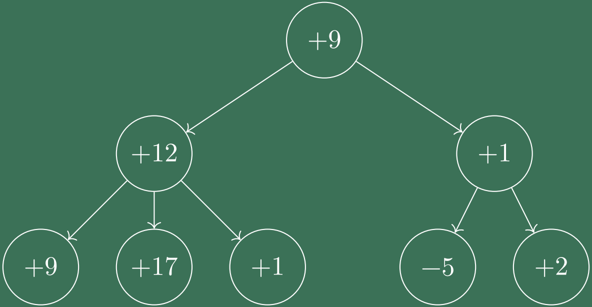
AlphaZero

Base Architecture

$$\text{AlphaZero} = \overset{\text{a variant of}}{\text{PUCT}} + \overset{\text{convolutional}}{\text{ResNet}}$$

Monte–Carlo Tree Search

An Efficient Way to Achieve Higher Reward



$$\pi(a|s) \propto N(s,a)^{1/\tau}$$

π: resulting probability distribution after the search,

$N(s,a)$: the # of visits to a at s, τ: temperature constant

PUCT

Polynomial Upper Confidence bounds applied to Trees

Exploitation of nodes with high rewards vs. Exploration of new nodes

PUCT

Polynomial Upper Confidence bounds applied to Trees

$$a_t = \underset{a\text{: action}}{\arg\max} \left[Q(s_t, a) + c_{\text{puct}} P(a|s_t) \frac{\sqrt{\sum_b N(s_t, b)}}{1 + N(s_t, a)} \right]$$

s_t: t^{th} state,

a_t: action to choose at t^{th} step,

$P(a|s_t)$: prior probability.

Use of Neural Network

AlphaZero gets the prior probability distribution and the expected value of unvisited nodes from the neural network.

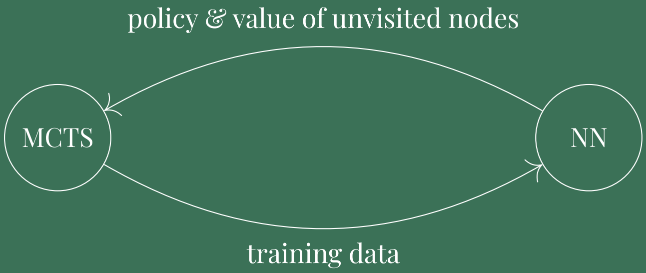
$$\text{NN} : \text{nonterminal state} \mapsto (\text{policy}, \text{value}).$$

In fact, $P(a|s)$ is the weighted average of the policy from the NN and a Dirichlet random noise.

$$P(\cdot|s) = (1-\epsilon)\,\text{NN}_{\text{policy}}(s) + \epsilon E, \quad E \sim \underset{\boldsymbol{\alpha}=(\alpha,\ldots,\alpha)}{\text{Dir}(\boldsymbol{\alpha})}.$$

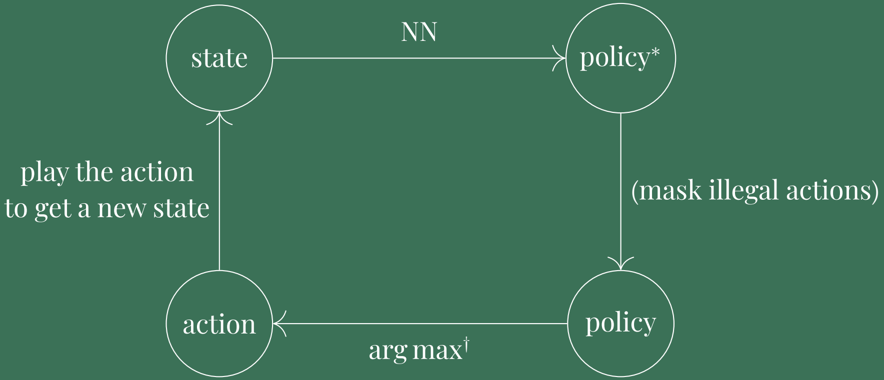
Train the Neural Network

The training data is provided by the MCTS simulation.



Self-Play

Evolve by Competing with Itself



†: In fact, AlphaZero randomly chose an action from $\mathrm{NN}_{\text{policy}}(\cdot|s)^{1/\tau}$ for $\tau \approx 0$, but it is not that different from the arg max.

Key Differences from AlphaZero

PIMC: Perfect Information Monte–Carlo

Run MCTS on Games with Hidden Information

As Go-Stop is a game with hidden information, the neural net must not obtain the full state of the game.

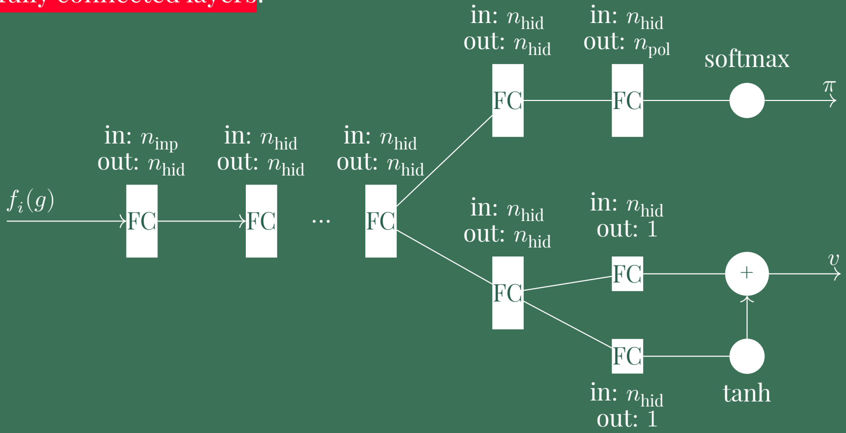
$$\underset{\text{game state}}{g} \xrightarrow[\text{observation}]{f_i} \underset{\substack{\text{state observable}\\ \text{to player } i}}{f_i(g)} \xrightarrow{f_i^{-1}} \underset{\substack{\text{the set of states}\\ \text{that seem identical to } i}}{f_i^{-1}(f_i(g))} \longrightarrow \underset{\text{a sample}}{\mathcal{D}(g)}$$

We sampled some games from $f_i^{-1}(f_i(g))$ randomly and ran MCTS with those games at each step of MCTS.

Network Layers

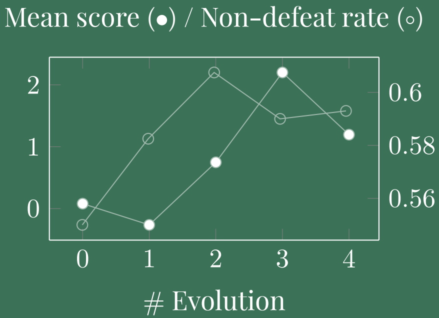
(for now)

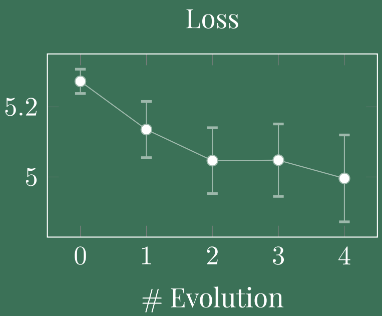
As Go-Stop boards have no geometric properties, it is weird to use convolutional layers in the network. Instead, we replaced them with fully connected layers.



Results

Quantitative Result: AlphaGoStop vs. Random





Live Demo

https://gostop.kanu.kim

Further Suggestions & Plans

Further Suggestions & Plans

- Train more and more, and more!
- Change hyperparameters and the reward function.
- When sampling from $f_i^{-1}(f_i(g))$, we may use trainable sampler to make it learn the distribution of possible games from the given observation. This may make the sampling process more stable.
- Use ISMCTS or $\alpha\mu$ search to handle incomplete information. PIMC has been criticized due to its critical problems: strategy fusion and non-locality. By using other algorithms to handle hidden information, we may get a better result.
- Or use different architecture like ReBeL to handle imperfect information better.

Thank you for listening.

Addendum

Implementation Details

Hyperparameters (for now)

- Batch size $B = 32$,
- Input dimension $n_{\text{inp}} = 307$,
- Hidden dimension $n_{\text{hid}} = 256$,
- Policy output dimension $n_{\text{pol}} = 167$,
- Learning rate: $0.1 \cdot 0.95^{\max(0, j-9)}$,

Implementation Details

Hyperparameters (for now)

- $\epsilon = 0.25$, $\alpha = 1$ (Dirichlet noise),
- # of search in one MCTS simulation: 96,
- # of simulations in one network update: 30,
  - This gives about 600 data triples per update.
- PUCT constant $c_{\text{puct}} = 1$,
- Temperature parameter $\tau^{(j)} = 1$ if $j < 10$ else 0.1,
- The size of a sample $\mathcal{D}(g)$ from $f_i^{-1}(f_i(g))$:
  $n_{\text{samp}}^{(j)} = \max(\lfloor 2^{7.5-j/8} \rfloor, 4)$,

How to Play

Special cards

- The following cards are called bonus cards, and they are treated as junk cards whose valued are 2- or 3 times of normal junk cards, resp.

- The values of the following junk cards are double of normal ones'.


- The animal card of September  can be used as a double junk card, and it can be moved only once in a game, usually when the player declares Go or Stop.

How to Play

Throwing a Bomb

- ★ When there are 3 (or 2) cards of the same month and there are 1 (or 2) card(s) of the same month on the center field, then the player can throw a bomb, meaning that throwing all 3 (or 2) cards simultaneously. The player obtains 2 (or 1) chances to flip the top card without throwing a card from their hand at their turn. The player takes those 4 cards of the same month and an additional junk card from the opponent, if possible.

How to Play

Flipping

- When the flipped card was a bonus card, the player can flip the next top card once more and they takes an additional junk card which the opponent captured.
- When a player threw a card without matches on the center field and then the flipped card was matched with the card thrown, it is called a discard-and-match. The player takes those cards and an additional junk card from the opponent.
- When a player threw a card with one matched card on the center field and then the flipped card was again matched with the card thrown, it is called a stacking.

How to Play

Flipping

- When a player threw a card with two matched cards on the center field and then the flipped card was again matched with the card thrown, it is called a *ttadak*. The player takes those four cards of the same month and an additional junk card from the opponent.
- When the center field was cleared after one's turn, it is called clearing and the player taken the last turn takes an additional junk card from the opponent.

How to Play

Go and Stop

- ★ When either the temporary score of a player (refer to the next slide) is ≥ 7 and they did not declare 'Go' yet, or the temporary score of a player has incrased since the last declaration of 'Go' in the other cases, the player can declare 'Go' (again) or 'Stop.'
- ★ When the player declared 'Stop,' the game is over and the winner calculates the final score (refer to the next slide.)
- ★ When the player declared 'Go,' the game continues.

How to Play

Calculation of Scores

- ★ There are two kinds of score factors: additive ones and multiplicative ones.
- ★ We calculate the temporary score by adding all the additive scores.
- ★ The temporary score is used to decide whether a player can declare 'Go' or 'Stop.'
- ★ We calculate the final score by multiplying the other factors to the temporary score.
- ★ Go-Stop is a zero-sum game, meaning that the loser's score is the same with the winner's score multiplied by -1.

How to Play

Additive Score Factors

- ★ When a player captured n junk cards, its score is $\max(0, n-9)$. Double junks and triple junks are counted as 2 and 3 cards, following the multiplicity of them.
- ★ When a player captured n ribbon cards, its score is $\max(0, n-4)$.
- ★ When a player captured n animal cards, its score is $\max(0, n-4)$.

How to Play

Additive Score Factors

- ★ When a player captured 3 bright cards with the bright of December , it worths 2 points.
- ★ When a player captured 3 bright cards without the bright of December , it worths 3 points.
- ★ When a player captured 4 bright cards, it worth 4 points.
- ★ When a player captured 5 bright cards, it worth 15 points.

How to Play

Additive Score Factors

- ★ When a player captured all blue ribbons , it worths 3 points.
- ★ When a player captured all red ribbons , it worths 3 points.
- ★ When a player captured all plant ribbons , it worths 3 points.
- ★ When a player captured all *five birds* cards , it worths 5 points.

How to Play

Additive Score Factors

- When a player declared ‘Go’ n times, it worths n points.
- When there was a four of a month in one’s hand (all 4 cards of the same month were in one’s hand,) then the game ends and the player having the four of a month gets 10 points. If both players had those, then nullify the game.
- When a player made three stackings, then the player wins with 10 points (without counting any other additive score factors).

How to Play

Multiplicative Score Factors

- When a player declared 'Go' n times, the score is multiplied by $2^{\max(0,n-2)}$. (Together with the additional scores.)
- When the winner *shaked* or *threw a bomb* n times, the score is multiplied by 2^n.

How to Play

Multiplicative Score Factors

- ★ When the winner has captured 3 or more bright cards and the opponent has no bright cards, it is called bright penalty and the score is multiplied by 2.
- ★ When the winner has captured 7 or more animal cards, it is called animal penalty and the score is multiplied by 2.
- ★ When the winner has captured 10 or more junk cards and the opponent has ≤ 7 junk cards, it is called junk penalty and the score is multiplied by 2.
- ★ When the loser had declared 'Go' at least once before, it is called go penalty and the score is multiplied by 2.

References

References

- ★ Mastering the game of Go without human knowledge
  https://www.nature.com/articles/nature24270.epdf
- ★ A Simple Alpha(Go) Zero Tutorial
  https://web.stanford.edu/~surag/posts/alphazero.html
- ★ Lessons From Implementing AlphaZero https://medium.com/oracledevs/lessons-from-implementing-alphazero-7e36e9054191
- ★ From-scratch implementation of AlphaZero for Connect4
  https://towardsdatascience.com/from-scratch-implementation-of-alphazero-for-connect4-f73d4554002a