Efficient Learning for AlphaZero via Path Consistency

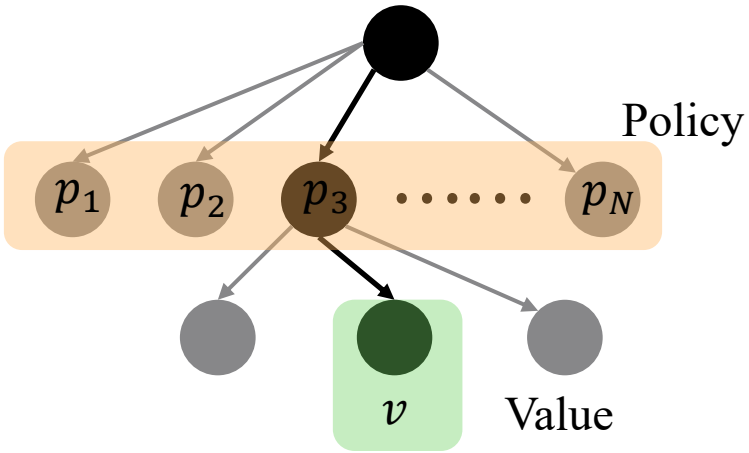
Dengwei Zhao, Shikui Tu, Lei Xu

June 2022

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

AlphaGo is successful but requires huge computational power

- Starting with AlphaGo[1-4], combining heuristic search with deep neural network has been a key to success.
- Model's performance highly depends on the number of self-play games, thus requiring huge computational power for learning.



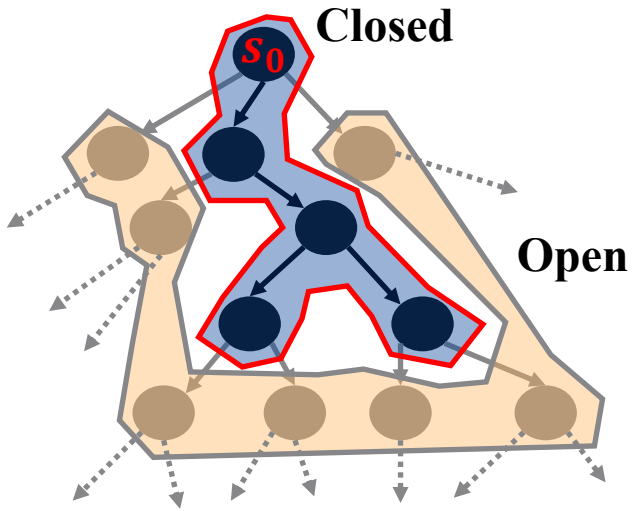
Model	Resource
AlphaZero	5000 + 64 TPUs
ELF OpenGo	2000 GPUs
MuZero	1000 + 16 TPUs

[1] Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." nature 529.7587 (2016): 484-489.
[2] Silver, David, et al. "Mastering the game of go without human knowledge." nature 550.7676 (2017): 354-359.
[3] Silver, David, et al. "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play." Science 362.6419 (2018): 1140-1144.
[4] Schrittwieser, Julian, et al. "Mastering atari, go, chess and shogi by planning with a learned model." Nature 588.7839 (2020): 604-609.

The optimal path in A* search algorithm

- A* search[5] expands node s based on evaluation value f.
- Optimality in A^* tree:
  - $f(s) = f(s_0)$ for every node s on an optimal path.

$$f(s) = \underbrace{g(s)}_{\text{accumulated cost from } s_0 \text{ to } s} + \underbrace{h(s)}_{\text{the future cost from } s \text{ to preferred termination}}$$



[5] Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. "A formal basis for the heuristic determination of minimum cost paths." IEEE transactions on Systems Science and Cybernetics 4.2 (1968): 100-107.

Path Consistency (PC) to assist A* search

- CNneim-A[6] relies on A* search's optimality to make a lookahead scouting to guide search process.
- Although PC was schematically proposed four years ago[7], it is yet unknown whether it works well.

- **Our paper** proceeds along this directions with three new developments:
  1) *Deep neural network is used to estimate $f(s)$.*
  2) *The A* search is replaced with MCTS to incorporate with AlphaZero.*
  3) *Moving average within a window of estimated optimal path is considered.*

[6] Xu, Lei, Pingfan Yan, and Tong Chang. "Algorithm cnneim-a and its mean complexity." Proc. of 2nd international conference on computers and applications. IEEE Press, Beijing. 1987.
[7] Xu, Lei. "Deep bidirectional intelligence: AlphaZero, deep IA-search, deep IA-infer, and TPC causal learning." Applied Informatics. Vol. 5. No. 1. SpringerOpen, 2018.

New ways to implement PC

- PC is turned into that "*values on one optimal search path should be identical*" in board games.
- A weighted penalty is added to the loss function:

$$L(\theta) = L_{RL}(\theta) + \lambda L_{PC}(\theta) \qquad L_{PC}(\theta) = (v - \bar{v})^2$$

- L_{PC}: deviation from the average value within a sliding window.

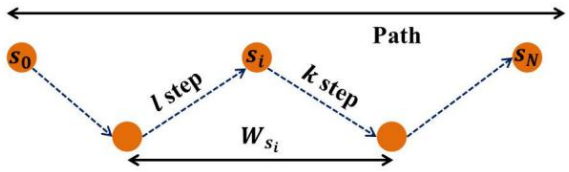


Figure 1. $\bar{v}$ calculation with historical path in a terminated game.

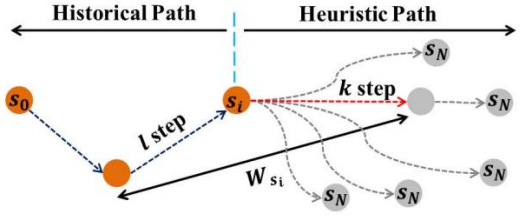


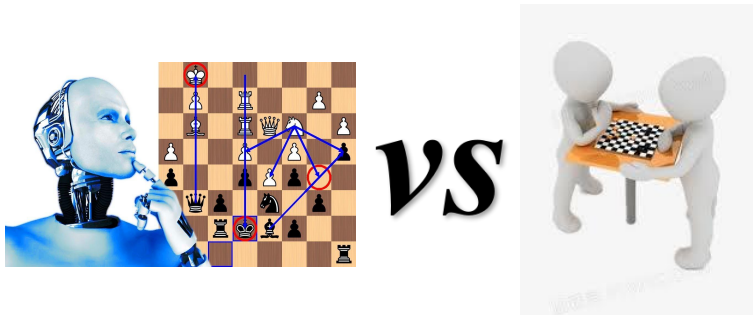
Figure 2. $\bar{v}$ calculation with both historical and heuristic path.

- PC can be imposed on the high-dimensional feature map f_v.

Our PCZero outperforms AlphaZero greatly

- PCZero obtains **94.1%** winning rate, much higher than AlphaZero's **84.3%**, when competing with MoHex 2.0, the champion of 13 × 13 Hex Computer Olympiad in 2015.
- PCZero consumes only 900K self play games during learning, which is a small-scale data that humans can make in a lifetime.

$$0.9M \approx \underbrace{2}_{\text{games per hour}} \times \underbrace{12}_{\text{hours per day}} \times \underbrace{365}_{\text{days per year}} \times \underbrace{100}_{\text{years in a lifetime}}$$



PCZero has better generalization ability

- Larger training value loss, but much lower test value loss.

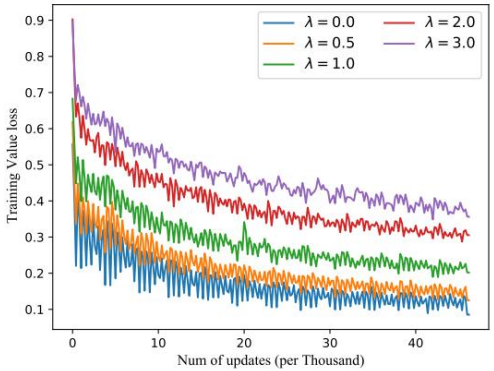


Figure 8. Training value loss on 13×13 Hex for different λ.

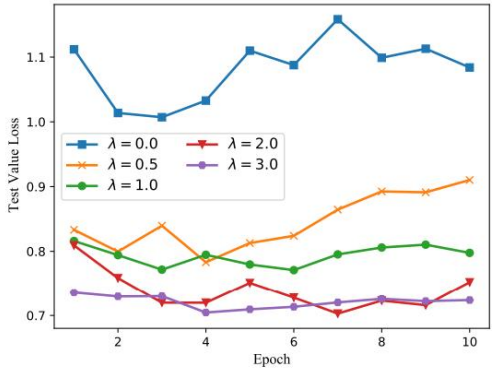


Figure 9. Test value loss on 13×13 Hex for different λ.

- Better generalization ability improve MCTS player's performance greatly.

Table 3. Winning rate of offline PCZero against offline AlphaZero at $\lambda = 2.0, \beta = 0.0$.

GAME	GREEDY PLAYER	MCTS PLAYER
HEX (8×8)	51.6%	58.6%
HEX (9×9)	53.1%	59.9%
HEX (13×13)	52.1%	61.5%
OTHELLO	50.5%	80.5%
GOMOKU	56.8%	64.0%

PC Loss is different with value loss

- Increasing the importance of value loss cannot replace the role of PC loss.
- AlphaZero's test PC loss also decreases, suggesting that PC is a nature required for strong value predictors.

Table 4. Winning rate of offline AlphaZero with different γ against offline AlphaZero with $\gamma = 1.0$.

GAME	γ	GREEDY PLAYER	MCTS PLAYER
HEX (13×13)	2.0	48.8%	45.9%
HEX (13×13)	3.0	55.9%	55.0%
OTHELLO	2.0	49.2%	34.8%
OTHELLO	3.0	42.8%	42.8%

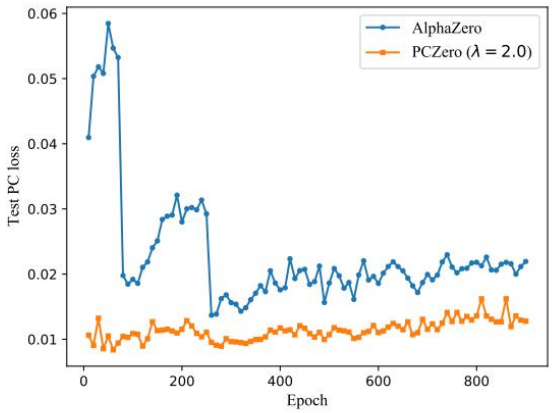


Figure 5. Test PC loss on expert dataset for online learning.

Conclusion and future work

- We proposed PCZero based on PC optimality condition. Experiment results indicate that PCZero is more efficient in learning in Hex, Othello, Gomoku for both online learning and offline learning.
- In the future:
  - Investigate the theoretical foundations under the PCZero scenario.
  - Generalize the PCZero framework to the real applications.

Thanks!