Xây dựng game cờ caro sử dụng Reinforcement Learning

Nguyễn Thị Kim Anh - Phạm Trung Hiếu - Phạm Thu Hoài
20020628 - 20020664 - 20020665
Trường đại học Công Nghệ , Đại học Quốc gia Hà Nội
Hà Nội, 24/12/2023

Tóm tắt: *Bài luận trình bày phương pháp học tăng cường trong game cờ caro. Trí tuệ nhân tạo AlphaGo Zero, sẽ được tiến hành huấn luyện để học được cách chơi chờ caro. Bên cạnh đó, báo cáo cũng đánh giá việc sử dụng Reinforcement Learning trong bài toán về chơi game đối kháng so với các phương pháp truyền thống khác ...*

Keywords: *Monte carlo tree search(MCTS), AGZ(AlphaGo Zero), Deep neural network ...*

I. Tổng quan

AlphaGo Zero là một chương trình trí tuệ nhân tạo được phát triển bởi DeepMind, công ty con của Alphabet (công ty mẹ của Google). Được giới thiệu vào tháng 10 năm 2017, AlphaGo Zero là phiên bản cải tiến của AlphaGo. AlphaGo Zero (AGZ) do Deepmind phát triển đã đánh bại cao thủ cờ vây Lee Sedol (người đã từng 18 lần vô địch cờ vây thế giới và được coi là người chơi cờ vây giỏi nhất trong vòng 1 thập kỳ qua) với tỷ số 4-1. Sự kiện này đã gây chấn động rất lớn tới những người quan tâm đến trí tuệ nhân tạo vì từ trước tới nay con người vẫn luôn tự tin độ phức tạp của cờ vây khiến các thuật toán tìm kiếm của máy tính không bao giờ có thể đánh bại được. AlphaGo Zero không cần học bất cứ một thế cờ nào từ trước, thậm chí ở giai đoạn đầu "nó" còn không biết chơi cờ vây. AlphaGo Zero đã tự chơi với chính mình hàng triệu ván cờ trước khi trở thành AI cờ vây mạnh nhất thế giới.
Điều khiến AlphaGo Zero mạnh mẽ đến như vậy nằm ở 3 yếu tối chính:

- *Mạng học sâu:* nhận input là state hiện tại, đầu ra tách thành 2 nhánh trả về value (phần trăm thắng) và policy.
- *Monte Carlo Tree Search (MTCS):* là một thuật toán tìm kiếm các bước di chuyển tiềm năng nhất dựa trên các thông tin từ mạng học sâu. Một mạng học sâu nếu được train tốt sẽ giúp MTCS loại bỏ phần lớn các nước đi kém, khiến nó không phải brute-force như các engine cờ truyền thống.
- *Self-play:* AlphaGo Zero chơi với chính nó qua một số ván cờ và cập nhật phiên bản có tỉ lệ thắng cao hơn.

Trong bài báo cáo này, nhóm sẽ ứng dụng AlphaGo Zero vào trong bài toán chơi cờ caro. Cờ caro là một trò chơi đối kháng hai người chơi, thường được chơi trên một bảng với ô vuông và sử dụng các quân cờ đen và trắng. Mục tiêu của trò chơi là tạo thành một dãy liên tiếp gồm năm quân cờ của mình theo chiều ngang, dọc, hoặc chéo trên bảng, trước khi đối phương làm được điều đó. Mục tiêu đặt ra sẽ là dạy máy tính chơi được trên bàn cờ 9x9.

II. Nền tảng lý thuyết

A. Quá trình huấn luyện

Quá trình huấn luyện được thực hiện với 3 bước thực hiện song song:

1) Self-play

Trong bước này AlphaGo Zero tự chơi với chính nó để thu thập dữ liệu trận đấu, phục vụ training mạng học sâu. Tại những ván cờ đầu tiên, AlphaGo Zero chơi rất "ngây thơ" vì output mạng học sâu lúc này (cả value và policy) chỉ là những giá trị random. Nhưng chơi càng lâu, mô hình càng trở nên xuất sắc. Thuật toán sử dụng MCTS để chọn nước đi. Với mỗi nước đi, các giá trị sau được lưu lại:

Action probabilities: Phân bố xác suất các hành động kế tiếp. Phân bố này được tính toán từ MCTS.

The winner: +1 nếu lượt này thắng, -1 nếu lượt này thua. Giá trị này chỉ xuất hiện khi game kết thúc.

2) Retrain network

Dựa trên bộ dữ liệu đã thu thập (và được update liên lục) từ quá trình self-play, thuật toán thực hiện train lại mạng neuron và tối ưu trọng số mạng. Vòng lặp training diễn ra như sau:

Bước 1: Đánh giá mạng sau mỗi 1000 vòng lặp

Bước 2: Mạng học sâu này sẽ gồm có hai nhánh do vậy nên nó cần 2 hàm loss (Policy loss và Value loss) để back propagation. Policy loss là cross-entropy giữa phân bố xác suất policy do mạng dự đoán và phân bố xác suất policy thực tế do MCTS tính được. Value loss là mean-squared error giữa predicted value và value của MCTS tính toán.

Bước 3: Optimizer sử dụng là gradient descent với momentum thay vì adam.

Bước 4: Sample mini-batch 2048 nước đi (là bộ 3 thông tin ở phần trên) từ games mới nhất, lấy game state làm input cho mạng.

Bước 5: Train mạng với bộ dữ liệu này.

Hình 1: Quá trình train

3) Evaluate network

Bước này ta thực hiện đánh giá mô hình học sâu mới train để kiểm tra liệu nó có tốt hơn phiên bản tốt nhất không. Hai mạng sẽ được "đấu" với nhau 400 games, cả 2 đều sử dụng MCTS để chọn nước đi. Mô hình mới train phải thắng ít nhất 55% số ván đấu để trở thành "người chiến thắng", khi đó trọng số của mạng được cập nhật theo trọng số mới nhất.

B. Monte Carlo Tree Search

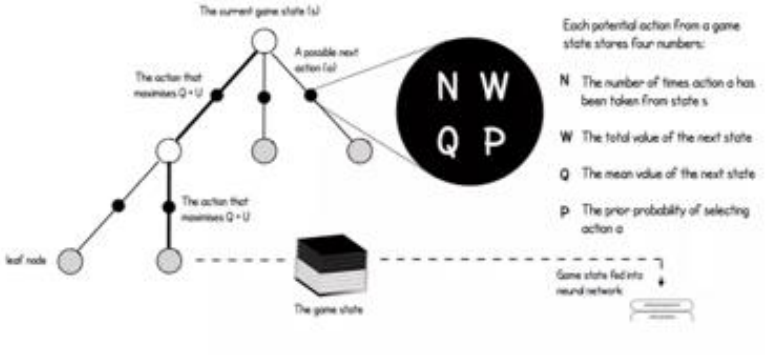
1) Tree statistics

Thuật toán MCTS bao gồm 4 bước:

Selection: Từ node gốc, chọn đường đi tiềm năng nhất (dựa trên một vài statistics) cho đến khi gặp node lá.

Expansion: Tạo một node lá từ nốt hiện tại
Simulation: Mở rộng đến khi game kết thúc
Backup (back propagation): Lưu lại đường đi này và update statistics của các cạnh trên path đã chọn theo hướng từ dưới lên trên (từ node lá đến node gốc).

Lặp lại các bước này sau nhiều lần ta sẽ có 1 cây với node gốc là state hiện tại và từ đó chọn ra nước đi tiếp theo tốt nhất dựa trên statistics của cây. MCTS trong AGZ, mỗi cạnh sẽ là một action và có 4 statistics:



Hình 2: Tree statistics

Trong đó:
N: Số lần action a được chọn tại state s.
W: Tổng value của state tiếp theo. Mỗi khi ta đến node lá MCTS query mạng neuron để lấy value của state đó; và trong quá trình backup, giá trị này cộng dồn (từ dưới lên trên) vào W của các node trên đường đi.
Q: value trung bình của state. Đơn giản là lấy W/N. Ý tưởng của Q cũng tương tự như Q-value trong Q-learning.
P: Prior probabilities lựa chọn action a tại state s. Giá trị này lấy từ nhánh policy của mạng học sâu.

2) Cơ chế hoạt động

Thuật toán bắt đầu với node gốc đại diện cho state hiện tại, MCTS chọn action có (Q+U) lớn nhất với Q là value trung bình của state tiếp theo,U là một hàm của P và N trong đó U tăng khi action đó chưa được chọn nhiều (exploration) hoặc prior probability của action đó cao (exploitation).

$$U(s,a) = c_{puct}.P(s,a).\frac{\sqrt{\sum_b N(s,b)}}{1+N(s,a)}$$

Trong đó: N(s,a) là số lần chọn action a từ state s, cpuct là hyperparameter điều chỉnh trade-off giữa exploration và exploitation

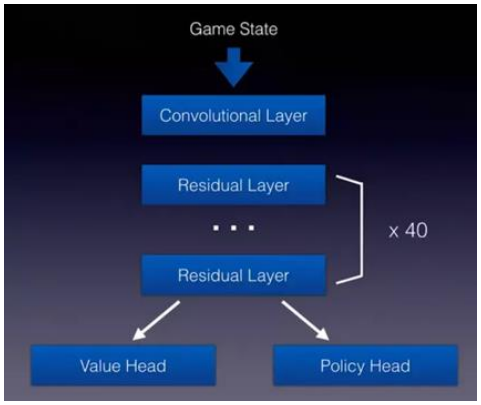
Sau đó, ta tiếp tục mở rộng đến khi gặp node lá (terminal state). Game state của node này được đưa vào mạng neuron để predict ra hai giá trị p (phân bố xác suất các nước đi kế tiếp - lấy từ nhánh policy) và v (value của state này - lấy từ nhánh value). Cuối cùng tiến hành backup các cạnh trước đó cho đến tận node gốc.

C. Deep Neural Network

Để MCTS hoạt động hiệu quả nhất, cần có sự hỗ trợ rất lớn từ mạng học sâu để làm giảm không gian tìm kiếm. AGZ sử dụng kiến trúc ResNet, đầu vào là game state, qua một convolution layer vào 40 residual layer trước khi tách 2 nhánh value và policy. Đây chính là kiến trúc Actor-Critic trong Reinforcement Learning.

Value head: Input đưa thẳng qua 1 lớp convolution 1x1, tiếp theo là batch norm và ReLU. Sau đó là 1 lớp fully connected với 256 units, 1 lớp ReLU sau đó đi qua 1 lớp convolution nữa. Activation lúc này là *tanh* để co giá trị về [-1, 1].

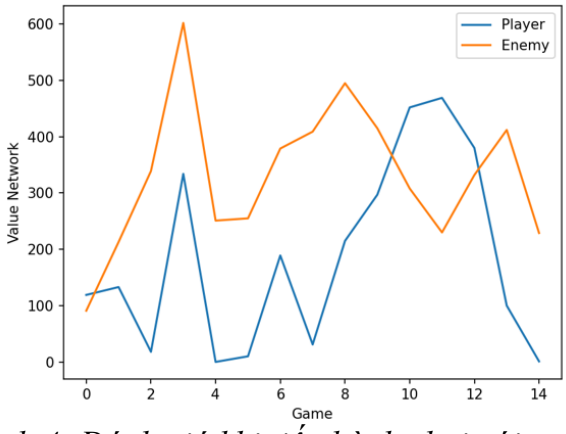
Policy head: Input được đưa qua một lớp convolution 2 filters 1x1, sau đó đi qua batch norm, ReLU và mạng fully connected. Mạng sẽ dự đoán xác suất cho từng ô vuông trong bàn cờ 9x9.



Hình 3: Mạng học sâu

III. Đánh giá và kết luận

Kết quả đào tạo mô hình tốt, mô hình có khả năng học và hiểu được các quy tắc và chiến thuật trong trò chơi cờ Caro, hiệu suất chơi cao. Mô hình có thể đánh bại hoặc chơi hòa với người chơi trong nhiều trường hợp.



Hình 4: Đánh giá khi tiến hành chơi với người

Trong các trò chơi tương tác giữa hai người như cờ caro, cờ vua, cờ tướng, ... đã xuất hiện một số phương pháp giải quyết, trong đó có Min-Max. Min-Max đặt đối thủ vào vai trò một người chơi hoàn hảo, và mục tiêu của người chơi là tối ưu hóa điểm của mình và đồng thời giảm thiểu điểm của đối phương. Tuy nhiên, hạn chế của phương pháp này phát sinh từ giả định rằng cả hai người chơi đều thực hiện những nước đi tối ưu mà không xem xét các hành động ngẫu nhiên hoặc sai lầm từ

đối thủ. Hơn nữa, thuật toán chỉ phát huy hiệu quả trong trường hợp không gian trạng thái nhỏ và không đối mặt với đối thủ hoàn hảo. Nhược điểm trên của Min-Max đã được Reinforcement Learning một cách hoàn hảo. Thay vào đó, mô hình sẽ tự học thông qua những kinh nhiệm thu được từ quá trình tương tác với môi trường. Trí tuệ nhân tạo AlphaGo Zero được trình bày trong báo cáo này có thể rút ra được các điểm chính sau đây:

Ưu điểm

- Tự học: AlphaGo Zero và MCTS sử dụng học tăng cường và tự học thông tin , điều này cho phép các thuật toán tự động cải thiện thời gian và học các chiến thuật tốt hơn.
- Tính linh hoạt: MCTS cung cấp khả năng khám phá không gian nước đi và đánh giá chúng. Điều này giúp AlphaGo Zero tìm kiếm và lựa chọn những nước đi tiềm năng và tạo ra các dạng chiến thuật đa dạng. Đồng thời hoạt động khá tốt trong không gian trạng thái lớn.

Nhược điểm

- MCTS mất thời gian tính toán đáng kể để tìm kiếm và đánh giá các nước đi. Điều này làm giảm hiệu suất chơi và thời gian đáp ứng chậm đi.

IV. Tài liệu tham khảo

1. (No date) *Mastering the game of go without human knowledge*. Available at: https://discovery.ucl.ac.uk/id/eprint/10045895/1/agz_unformatted_nature.pdf (Accessed: 25 December 2023).
2. Junxiaosong (no date) *Junxiaosong/alphazero_gomoku: An implementation of the AlphaZero algorithm for Gomoku (also called Gobang or five in a row), GitHub*. Available at: https://github.com/junxiaosong/AlphaZero_Gomoku (Accessed: 25 December 2023).