

Creating Action Library

Database collection

We used the games of grandmasters from 1984 to 2007 stored as PGN (Portable Game Notation) format

Database parsing

We created an algorithm to parse PGN games to separate white moves and black reactions, then store the moves in action library along with the FEN (Forsyth–Edwards Notation) representation for each move

Building simple chess AI

We used a step-by-step guide to building a simple chess AI from:

<https://medium.freecodecamp.org/simple-chess-ai-step-by-step-1d55a9266977>

It depends on 4 steps to build the AI:

- Move generation
- Board evaluation
- Minimax
- Alpha beta pruning

But we did not use minimax and alpha beta pruning steps.

Move generation

Chess.js was used for move generation. And chessboard.js was used to visualize the board. These libraries provide all rules for playing chess along with the legal moves for each position except that AI that is not provided.

We started with getting all legal moves for each position using chess.js library

Board evaluation

Evaluate the board for each move using the evaluation function from:

https://www.chessprogramming.org/Simplified_Evaluation_Function

Searching for the best move to play

My algorithm till now does not use minimax and alpha beta to get the best move. It just play the move with highest evaluation value using these steps:

- 1- at each position, get all legal moves using chess.js library
- 2- evaluate the board for all these moves.
- 3- get the move with the highest evaluation value, store it as BestGameMove.

- 4- retrieve moves from action library according to opponent move similarity. For example if the played move is d3, then we retrieve all reactions for the move d3.
- 5- evaluate all the retrieved reactions.
- 6- get the reaction with the highest evaluation value and store it as LibBestMove.
- 7- compare BestGameMove with LibBestMove and play the move that is highest.
- 8- back to step 1.

Papers

Similarity

- 1- On the automatic generation of case libraries by chunking chess games 1995

يتركز على بناء ال case library عن طريق تقطيع ال board لمجموعه من ال chunkes بناء على العلاقات المباشرة بين القطع زي القطع اللى بتهاجم بعض بيحطهم فى chunk و القطع اللى بتدافع عن بعض بيحطهم فى chunk و بيسترجع على اساس ال chunks دى بحيث انه بيقسم ال board الحالى ل chunks برده و يسترجع ال case اللى فيها اكبر تشابه لل chunks

- 2- Using example-based reasoning for selective move generation in two player adversarial Games 1998

مش فاهمه منها اى حاجه خالص

- 3- Retrieval of similar chess positions 2014

بيعمل encoding string لكل position و بيرجع position واحد من كل game و بيعتمد فى التشابه على تشابه الخطط كمان مش بس مكان كل قطعه

انا كودت الجزء دا ماعدا ال x-ray attack هبقى اعمله و ال retrieve عندى مختلف علشان ماعرفتش اعمل بتاعه

Suggested addition add features for retrieving similar cases from action library such as board similarity

Evaluation

- 1- Case-based evaluation for computer games 1995
- 2- An Adaptive Evolutionary Algorithm Based on Typical Chess Problems for Tuning a Chess Evaluation Function 2001
- 3- Using an Evolutionary Algorithm for the Tuning of a Chess Evaluation Function Based on a Dynamic Boundary Strategy 2006
- 4- An evolutionary algorithm for tuning a chess evaluation function 2011
- 5- An evolutionary algorithm coupled with the hooke-jeeves algorithm for tuning a chess

Evaluation function 2012

6- An adaptive evolutionary algorithm based on tactical and positional chess problems to adjust

The weights of a chess engine 2013

7- An evolutionary algorithm with a history mechanism for tuning a chess evaluation function

2013

Suggested addition improve weights of the evaluation function or add features to it