Project Proposal

Background

Since Deep Blue first beat Kasparov in 1997 there have been many chess engines which can far outperform the best humans in both speed and chess ability, such as Stockfish which is currently regarded as the most advanced chess engine. However there are not many chess engines which also provide explanations, or a way of interpreting why the suggested move and evaluation is correct. There is a commercially sold piece of software which does claim to achieve this, however they have not published details of how they determine the explanations.

Terms

- **ELO** - A method used to calculate the relative strength of players in chess and other zero-sum games
- **Tactic** - A move or sequence of moves which gives a short term advantage to a player, often by forcing the opposing player to choose between two negative outcomes (such as by attacking two pieces at the same time)
- **Strategy** - Long term ideas in the chess game to try to gain an advantage over the opponent, such as the structure of the pawns
- **Evaluation** - In the context of chess, the evaluation function attempts to assign a value to the advantage that either player has, in terms of the number of pawns that the advantage is equivalent to (with an advantage for the white player being positive, and negative for white)

Introduction

Chess engines such as Stockfish are becoming increasingly more powerful, and are now able to easily outperform even the best humans with much less computation/thinking time per move. They achieve this by searching through the many possible positions that could be reached from the starting position, and evaluating them based on a number of features of the position. The best move can then be determined using the minimax algorithm, by selecting the move which results in the most favourable position (in terms of the evaluation) for the player. Many players use the evaluations and continuations of moves suggested by these engines to better understand positions and improve their ability. They often achieve this by looking through the various continuations of moves that the engine suggests in order to identify the tactical or strategical advantage that one move has over another.

However, less experienced players will have more difficulty identifying these differences, since they will have encountered the tactical and strategic ideas less frequently in the games they have played. Furthermore, many beginners will not have been explicitly taught about some tactical ideas, and may not be able to understand why a position is favourable for one player even after going deep down a path of suggested moves. Understanding why a certain move was better than another is one of the most important aspects needed to learn from a mistake made in one game, and to make a better move in a similar position in future games. My project aims to provide both move suggestions and reasons for why the move is the best available. The reasoning will be in the form of which tactical and strategical features of the position had the biggest impact on determining which move is the best available in the position.

Starting Point

The project will start from scratch, so I will devise and build the interpretable evaluation model first. This work is likely to use a library such as NumPy for the implementation of a linear regression model and any other models I use, since it will be better tested and optimized than my implementation of the same mathematical model would be.

I have an understanding of the algorithms that I will be using for the engine, from the IB Artificial Intelligence course, the IB Data Science course, the IA Algorithms course and from my own interests, however I have no experience in building a chess engine. The Part II Natural Language Processing unit of assessment that I will take in Michaelmas 2023 will also add to my existing knowledge. I know how to play chess, and have a reasonable ability and understanding of the game.

Success Criteria

- The project should be considered a success if the chess engine is able to produce evaluations of all legal chess positions, as well as the tactical and strategical aspects that contributed most to that evaluation.
- The project should be considered a success if the evaluation matches Stockfish in 80% of test positions in assigning a position to approximately equal (-1.0 to 1.0 pawns), winning for white or winning for black.
- The project should be considered very successful if I produce a chess engine that has an estimated ELO of above 1500 (estimated by testing the engine against other engines of a known ELO).
- The project should be considered very successful if it is able to produce explanations for its evaluations and moves in natural language.
- The project should be considered very successful if the explanations match the expected explanation for 80% of the positions in a test suite (these positions will come from puzzles where the best move and explanation for this move is known, and then manually reviewed to check that the explanations match closely).

Work to be Undertaken

1. **Implementing the chess engine:**
   I will implement a chess engine using the minimax algorithm. The engine will be programmed in such a way that any function that takes in a position and outputs an evaluation can be used. During this stage I will test that the move generation is correct, and that all legal positions are being explored.
2. **Extracting features from a position:**
   During this stage I will write functions to encode a chess position into a set of interpretable tactical and positional features, such as the material count of each side and whether a piece is attacking two other pieces at once. I will then create the initial evaluation function by manually deciding on weights for these features. In order to decide on which features to use, I will use educational chess material such as textbooks.

3. **Extending the chess engine to provide interpretations:**
   I will extend the chess engine to use a weighted sum of the extracted features in the children positions of each node in the search tree. These feature weights will be passed up the search tree to the root node, to give weights to the most important features in determining the evaluation of the position.

4. **Fitting the evaluation function model:**
   I will fit a linear regression model using a dataset of chess positions and the evaluation assigned to each position by Stockfish. The linear regression model will assign weights to each of the features extracted from the position, which will ensure the evaluations are still interpretable. I will also explore the possibility of other models that could be used here.

5. **Evaluating the chess engine:**
   There are two aspects of the chess engine which will need to be evaluated, the performance of the chess engine measured in ELO, and the explanations of its evaluations. The accuracy of the evaluations can be evaluated by testing its ability to beat chess engines of a variety of known ELO levels. The interpretations will be more difficult to evaluate, as they are often more subjective. One approach could be to find a dataset of puzzles which have a defined explanation for which is the best move.

Extensions

1. **LLM to provide natural language explanations:**
   In order to make the explanations for the evaluation easier to understand for beginners, I could use a Large Language Model to combine the most important factors with explanations of how they create an advantage for one player to provide a more natural explanation.

2. **Using techniques to increase search depth:**
   In order to improve the search depth that can be reached I could use search techniques such as alpha-beta pruning and iterative deepening in my chess engine to reduce the number of positions that need to be searched and evaluated at each depth. This will allow a greater search depth to be reached in the same amount of time.

Timetable

1. Michaelmas weeks 2-3 (12th October - 25th October)

   - Familiarize myself with the representation of chess positions and evaluations in the dataset
   - Research existing chess engines and their approaches to generating evaluations and moves
   - Set up the project and the repository, researching any libraries that I will need

   Deliverables

   - Final project proposal (Deadline 16th October)

   Other work

- Unit of assessment assignment (Deadline 26th October)

2. Michaelmas weeks 4-5 (26th October - 8th November)

   - Write a framework to support representing chess positions, and applying moves to a position
   - Write a function to generate all of the legal moves in a position, and a function to check if a position is a valid chess board
   - Write unit tests to ensure that the framework is able to determine legal moves and positions

   Other work

   - Unit of assessment assignment (Deadline 9th November)

3. Michaelmas weeks 6-7 (9th November - 22nd November)

   - Write a trivial evaluation function (material count and checkmate)
   - Implement the minimax search function

   Deliverables

   - A functional chess engine

4. Michaelmas weeks 8 & Christmas Vacation Week 1 (23rd November - 6th December)

   - Decide on which features to extract from the position
   - Write the functions which extract the features from a position
   - Write an evaluation function which uses manually decided weightings of these features

   Other work

   - Unit of assessment assignment (Deadline 1st December)

5. Christmas Vacation weeks 2-3 (7th December - 20th December)

   - Extend the minimax algorithm to pass weightings of how much each feature contributed to the evaluation up the search tree
   - Fit a linear regression model to get new weights for the features

   Deliverables

   - Chess engine that provides weightings of features as well as evaluations

6. Christmas Vacation weeks 4-5 (21st December - 3rd January)

   - Evaluate the chess engine against other chess engines in order to estimate its ELO
   - Evaluate the explanations by testing the engine on a puzzle with known tactical or strategic themes, to see if the engine is able to identify them correctly

7. Christmas Vacation weeks 6-7 (4th January - 17th January)

   - Apply alpha-beta pruning to the engine

- Write a first draft of the progress report using the evaluations to be reviewed by my supervisor

Deliverables

- First draft of progress report

8. Lent weeks 1-2 (18th January - 31st January)
    - Apply iterative deepening and other techniques I come across during research to improve the depth that the engine can search to
    - Finish the progress report with feedback from supervisor

    Deliverables

    - Progress report (Deadline 2nd February)
9. Lent weeks 3-4 (1st February - 14th February)
    - Set up framework for using an LLM to convert the feature weights into an explanation in natural language
    - Prepare and present progress report

    Deliverables

    - Progress report presentation (Deadline 7th February)

    Other work

    - Unit of assessment assignment (Deadline 16th February)
10. Lent weeks 5-6 (15th February - 28th February)
    - Fine tune the prompt for the LLM
    - Run the evaluations for the updated versions of the chess engine
11. Lent weeks 7-8 (29th February - 13th March)
    - Begin writing preparation section of the dissertation

    Deliverables

    - Completed project

    Other work

    - Unit of assessment assignment (Deadline 15th March)
12. Easter Vacation weeks 1-2 (14th March - 27th March)
    - Finish writing preparation section of dissertation
    - Write implementation section of dissertation
13. Easter Vacation weeks 3-4 (28th March - 10th April)
    - Write evaluation and conclusion sections of dissertation

Deliverables

- First draft of dissertation submitted to supervisor and director of studies

14. Easter Vacation weeks 5-6 (11th April - 24th April)

    - Review feedback and make changes to the dissertation

15. Easter weeks 1-2 (25th April - 8th May)

    - Make final changes to dissertation based on continued feedback

    Deliverables

    - Project source code and dissertation (deadline 10th May)

Resources

1. For the development and writing of the project and dissertation, I will primarily use my personal laptop (M1 MacBook, 256GB SSD, 8GB memory). In case of a laptop failure, I have access to college computers in the library, spare laptops from my family, or purchasing a replacement laptop. I will use GitHub for version control, as well as Google Drive for backing up my code and any written work. The Google Drive backups will be made automatically whenever changes are made to the files. I accept full responsibility for this machine and I have made contingency plans to protect myself against hardware and/or software failure.

2. I will use a dataset of chess positions which I will acquire from the internet to be used for training and testing data. The positions will be labelled with evaluations from the Stockfish chess engine, and will be used as a source of "ground-truth" evaluations for training and testing

3. A set of puzzles with solutions which I will acquire from the internet to be used in the evaluation of the interpretations that my engine produces.

4. Access to 4 RTX 8000 GPUs through my supervisor's group, which can be used to run an LLM to generate natural language explanations.