Project Proposal

Introduction

Chess engines such as Stockfish are becoming increasingly more powerful, and are now able to easily outperform even the best humans in much less time. Many players use the evaluations and continuations 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 combine approaches to interpretable machine learning models with techniques from more traditional chess engines in order to provide accurate evaluations and move suggestions, as well as giving reasons for them. This would be achieved by training an interpretable model on a dataset of chess positions, with the features being tactical and strategical aspects of the position and the output being the evaluation assigned to the position by a leading chess engine. A chess engine will then be built using the trained model as the evaluation function, and combining the interpretable output of the positions in the search tree to provide reasoning for the evaluation in terms of tactics and strategy.

Starting Point

The project will start from scratch, so I will devise and build the interpreatable evaluation model first. This work is likely to use a library such as NumPy for the implementations of the models, since it will be better tested and optimized than my implementation of the same mathematical model would be. I will use an existing chess engine, however this will only be used in the training of the interpratable model that I use as the evaluation function.

I have an understanding of the algorithms that I will be using for the engine, from the IB Artificial Intelligence course and from my own interests, however I have no experience in building a chess engine.

Work to be Undertaken

1. **Preparing the data:**
   I will need to familiarize myself with different representations of chess positions, and decide on the best one to use based on how quickly I will be able to extract the tactical and strategical features, as well as which is supported by the engine I will use for my training data evaluations. I will also need to find and clean a dataset of chess positions in this representation to use as training and testing data for my interpretable model, by

calculating the evaluation of each position using an existing chess engine.

2. **Devising the model:**
   I will need to investigate and decide on the model which is most appropriate for the project. This will involve researching and experimenting with multiple interpretable models to find one that provides both accurate evaluations as well as having clear interpretations. During this stage I will also need to decide on which features I use. The speed at which the model can generate evaluations for a new position will also be an important factor, since it will determine the search depth that I will be able to use in the engine.

3. **Training the model:**
   Once I have chosen which model to use, I will need to train the model using the dataset that I have prepared. This stage may also involve the tuning of some hyperparameters depending on the model which I choose to use.

4. **Implementing the chess engine:**
   Using the trained model as an evaluation function, I will implement a chess engine using the minimax algorithm. During this stage I will also need to determine how to combine the interpretations of the children into the interpretation for a position.

5. **Evaluating the chess engine:**
   There are two aspects of the chess engine which will need to be evaluated, the accuracy of the evaluations it assigns to positions, and the explanations for these evaluations. The accuracy of the evaluations can be evaluated by comparing them to the evaluations assigned by the original chess engine that was used to train the model. 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. **GUI to highlight reasoning:**
   In order to make it easier for users to visualize the interpretations, I could design a graphical user interface which highlights and explains the tactical and strategic aspects of the position that contribute most to the evaluation.

2. **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.

3. **Using techniques to increase search depth:**
   In order to improve the search depth that can be reached I could use techniques such as alpha-beta pruning and iterative deepening in my chess engine to improve the accuracy of evaluations in the same amount of time.

Success Criteria

The project should be considered a success if the chess engine is able to produce sensible evaluations of chess positions, as well as the tactical and strategical aspects that contributed

most to that evaluation. The project should be considered very successful if the evaluations align closely with those from the best chess engines, and if it produces explanations for the evaluations in natural language and/or through a visualisation.

Timetable

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

   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)
   -

   Deliverables
   -

   Other work
   - Unit of assessment assignment (Deadline 9th November)
3. Michaelmas weeks 6-7 (9th November - 22nd November)
   -

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

   Deliverables
   -

   Other work
   - Unit of assessment assignment (Deadline 1st December)
5. Christmas Vacation weeks 2-3 (7th December - 20th December)
   -

   Deliverables
   -

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

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

    Deliverables
    -
8. Lent weeks 1-2 (18th January - 31st January)
    -

    Deliverables
    - Progress report (Deadline 2nd February)
9. Lent weeks 3-4 (1st February - 14th February)
    -

    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)
    -

    Deliverables
    -
11. Lent weeks 7-8 (29th February - 13th March)
    -

    Deliverables
    -

    Other work
    - Unit of assessment assignment (Deadline 15th March)
12. Easter Vacation weeks 1-2 (14th March - 27th March)
    -

Deliverables

-

13. Easter Vacation weeks 3-4 (28th March - 10th April)

    -

    Deliverables

    -

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

    -

    Deliverables

    -

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

    -

    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.