COMPSCI 4AL3 Final Project Milestone 1

Team Members
Youcef Boumar - boumary@mcmaster.ca. Roles: Data preprocessing, feature engineering, tuning
Anthony Hunt - hunta12@mcmaster.ca. Roles: Model creation, training, tuning, evaluation
Muzamil Janjua - janjua@mcmaster.ca. Roles: Input and output visualization, model tuning

Context

Computation in the game of chess has historically served as a mainstream benchmark and demonstration of evolving computational power. From Deep Blue to Stockfish, deterministic optimization techniques for turn-based game engines have been continuously refined and improved for both efficiency and performance. In 2017, Google DeepMind introduced AlphaZero, a neural network-based approach to the game that bested the even highest rated engines of its time, advancing chess bots far beyond the abilities of the best human players[1]. As such, playing against bots of all levels has become deeply intertwined with the act of learning the game.

Modern chess websites like chess.com and lichess.org have provided several tools in evaluating positions and predicting moves to aid players in the process of climbing the rating later. However, these predictions assume that players will always play optimally for the entire game, judging the score of a move by objective strategic or material gains, far beyond the proficiency level of average players. Therefore, when attempting to predict the probability of a player winning, deterministic algorithms fail to account for the hidden relationships, human biases, and differences in player abilities.

On the other hand, a data-oriented machine learning algorithm focused on player behaviour rather than in-the-moment position analysis could help players determine the effects of specific moves on the outcome of a game. In turn, we may be able to use this knowledge to incorporate player mindsets and abilities in evaluating match performance, providing intuitive, personalized, and helpful feedback. We specifically aim to address questions that will help players fill in gaps left by deterministic analysis of games, such as, “Will a large material loss in the first 10 moves result in an immediate loss, even when facing a lower rated opponent?”, “How many moves must each side make before we can determine a winner with reasonable accuracy?”, or even “Is it possible to predict the winner of a chess match with consideration to player ability and historic performance?”.

Dataset

The Lichess database contains over 1.92 TB of data, equivalent to ~6 billion recorded chess games, starting from 2013[2]. In October 2024 alone, over 94 million games were recorded

[1] Silver, D., Kasparov, G., & Habu, Y. (2018, December 6). Alphazero: Shedding new light on chess, Shogi, and go. Google DeepMind. https://deepmind.google/discover/blog/alphazero-shedding-new-light-on-chess-shogi-and-go/

[2] Lichess.org. (2024, November). Lichess.org open database. lichess.org open database. https://database.lichess.org/#standard_games

through rated matches played on lichess.org, with a compressed size of 30 GB. Games are stored in a standardized Portable Game Notation (PGN) format[3], containing information about the date, player names, final result, player ratings, time limits, and an ordered list of moves in algebraic chess notation[4]. 17 features can be built from game metadata, while moves contain information about the board state. Additionally, approximately 6% of all games include deterministic engine evaluations and timestamps for each move. Given the complexity and nuance of written game notation, we will need to conduct extensive feature engineering tests to extract relevant information.

Solution Proposal

For this project, we will attempt to build a predictive neural network model to determine the probability of a player winning based on their comparative performance of the first *X* moves. We will compare the accuracy of different active sampling techniques in addition to limiting the amount of data required before making an accurate prediction. Given the time-based nature of some features and the overall flexibility of the dataset, recurrent neural networks may be a good choice to perform this analysis.

Although this idea independently originated from our personal experience learning from deterministic engines like Stockfish[5], similar projects and papers have utilized decision trees, SVMs, naive bayes, random forest, and ensemble learning techniques. In particular, a similar open-source project[6] trained a decision tree classifier on 20,000 games to predict the outcome of a match with 95% accuracy within the first 8 moves. Two theses explore the idea further, comparing various models and methods of interpreting the data[7][8]. It should be noted that none of these methods attempted to use neural network models.

Data processing will involve a primary conversion of PGN format to CSV for easier tensor translations. Moves will then need to be interpreted into a more usable score, likely using the per-move engine evaluations and timestamps built into some of the lichess database. After doing so, features will consist mostly of single numerical entries alongside high dimensional tensors to represent individual moves. The labels will consist of the match results (win, loss, or draw), as recorded by lichess. Python library pgn2data will be useful in handling the PGN dataset, and pytorch will be essential for the creation of the neural network model. Sklearn may additionally prove useful in processing the large quantity of data, in particular categorical variables and scaled numerical data.

[3] Chess PGN (portable game notation). Chess.com. (n.d.-b). https://www.chess.com/terms/chess-pgn

[4] Chess notation & algebraic notation. Chess.com. (n.d.-a). https://www.chess.com/terms/chess-notation

[5] Stockfish. (n.d.). https://stockfishchess.org/

[6] Predicting chess match results based on opening moves. 4641Project. (n.d.). https://samiamkhan.github.io/4641Project/

[7] Rosales Pulido, H. A. (2016). Predicting the outcome of a chess game by statistical and machine learning techniques (Master's thesis, Universitat Politècnica de Catalunya). https://upcommons.upc.edu/bitstream/handle/2117/106389/119749.pdf?sequence=1&isAllowed=y

[8] DeCredico, S. (2024). Using Machine Learning Algorithms to Predict Outcomes of Chess Games Using Player Data (Master's thesis, Rochester Institute of Technology). https://repository.rit.edu/cgi/viewcontent.cgi?article=13036&context=theses