Predicting Chess Player Elo Using Machine Learning: An Analysis of Move Sets

Douglas Zeller
Colgate University
dzeller@colgate.edu

Abstract

In this study, we attempt to predict chess players' skill level (ELO) through the moves they use in individual games. We use a Long Short-Term Memory (LSTM) neural network to analyze games in portable game notation (PGN). Our model can accurately predict the ELO rating of a player within a margin of +/- 200 roughly 16.5% of the time. This surpasses our baseline model (guessing the median ELO) by around 2%. While our results are less than optimal, our analysis revealed an interesting trend: as the number of moves tracked by the LSTM increases, the prediction accuracy decreases. Beyond the apparent indicating factor of a limited dataset, this finding highlights the difficulty of neural networks generalizing with such a large possibility-of-moves set.

1 Introduction

Chess is one of the oldest and most popular games in human history. Billions, if not trillions, of games have been played since its advent. As such, chess is a subject of study that ranges from game theory to artificial intelligence. Part of what drives its study is the extraordinarily large number of "sensible[1]" games that exist ($\sim 10^{40}$). The trillions of games that have been played represent only $1/(1 * 10^{28})$ of all sensible games.

One of the most significant breakthroughs of the 20th century was the defeat of Kasparov vs. Deep Blue[2] in a chess match. The victory indicated the looming information technology age, artificial intelligence, and the rapid advancement of computer systems. As we migrate into an age of deep learning intertwining with society to an unbelievable degree, we imagine no better prospect for our research of neural networks' current capabilities than chess.

1.1 Chess Fundamentals

To better understand our research, it is critical to have a high-level understanding of chess and the Elo rating system. The objective of chess is to capture the opponent's king. Each piece has different movement abilities, valued by how versatile that movement is.
Briefly:

1) **The Pawn** moves forward one space at a time and captures forward diagonally.
2) **The Bishop** moves cardinally on the diagonals.
3) **The Rook** moves cardinally up, down, left, and right.

[1] A sensible game refers to games where the player makes moves that are not strictly game-losing. For example, at the start of the game, you could sacrifice your queen with no strategic gain. These moves, and all resultant moves after the fact, are ignored in this number. Including non-sensible games, the possible outcomes skyrocket to a lower bound of 10^{123}. However, given that these games are, in essence, "non-sensical," we find it more apt to focus on the games that computationally matter. This gives us our number of roughly 10^{40}.

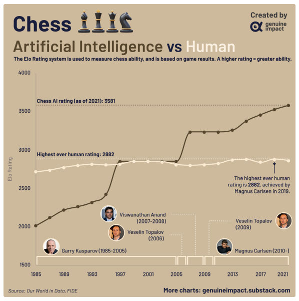
[2] Kasparov was the world chess champion at the time, and Deep Blue was the first supercomputer capable of surpassing him in a chess match, in 1997.

4) **The Knight** moves in an L-shape (2 spaces forward and one to the left or right **or** 1 space forward and two spaces left or right), with the unique property to "jump" over pieces blocking its path.
5) **The Queen** has the movement of the Bishop and Rook combined.
6) **The King** can move to any one space directly around itself.

As a general rule of thumb, the player only wants one of their pieces captured if they can capture a piece equal to or greater in value than their sacrificed piece. When a king has no possible places to move without being captured on the next turn, that is considered "checkmate," and the game is over.

1.2 The Elo Rating System

A chess player's best gauge of their capabilities is their Elo rating. This value ranges from 0-~3600[3]. The Elo rating has no technical cap, and the highest ratings from machines are constantly improving. The system is not a perfect measure of ability, as this can fluctuate between games. Instead, it is a way of roughly gauging skill between players. A player rated 200 points higher than their opponent is expected to win roughly 3 out of 4 games played.



Source: reddit: https://www.reddit.com/r/dataisbeautiful/comments/113mll8/oc_ai_vs_human_chess_elo_ratings_over_time/

1.3 Our Question

Chess.com, the largest online chess platform in the world, currently employs a system that predicts the rating of the white player and black player in each game. The system, while not public, likely considers the calculated accuracy of moves and historical data from millions of games. Despite this abundance of data, the predicted outcomes are extremely volatile, and the correct Elo of either player is rarely predicted. We aim to create a simpler model to predict player Elo more accurately.

We trained this model with the portable game notation(PGN) of ~30,000 games. PGN stores the moves of games in a standard notation that a model can be trained on while maintaining the most relevant data. Finding a strong correlation between move sets and the predicted Elo would be extraordinarily beneficial in placing players against equally skilled opponents. This would encourage new players to continue playing chess and benefit the chess community as a whole. Such a discovery could open new avenues for scientists to investigate chess and other similar strategy games, such as Go. These discoveries could also impact behavioral economics, where new analysis could be on time-pressure decision-making and biases. However, what these specific discoveries may be and their implications exist beyond the scope of this paper and our research.

2 Background

Many studies have been conducted focusing on machine learning in chess games. While none specifically tackle predicting the Elo of players with an ML

[3] The highest rating ever achieved by a human is 2882, by Magnus Carlson in 2019. Since Kasporov's defeat in 1997, computers have far surpassed humans in chess.

model, their work was necessary in creating the foundation for our analysis.

To ensure we had a proper analysis of Elo ratings, we leveraged two of Chess.com's articles, one titled "Elo Rating System" and the other "Chess Ratings - How They Work." By understanding how deviation in ratings occur and the intended meaning of these numbers, we could conduct our research in a manner that helped us align our data and set our expectations well. The former article helped us to align Elo as a probability-based system, where Elo is used as a gauge to estimate the win rate of one player against another. The latter broke down the mathematics behind updating Elo, which, while not used in our analysis, was essential in understanding the volatility of Elo over time.

Another study we used, "Predicting the Outcome of a Chess Game by Statistical and Machine Learning Techniques" by Hector Pulido, focused primarily on creating a robust dataset over tuning parameters. Despite a different end goal, having an established path of preprocessing was beneficial.

"Analyzing Positional Play in Chess using Machine Learning," by Bagadia et al., attempts to identify positional elements to improve the capabilities of chess engines. Instead of purely focusing on depth, where moves are computed based on search trees of moves 20-30 moves deep, the researchers work on teaching the machine about beneficial board positions, something that comes a lot more intuitively to humans than neural networks. For instance, having double-stacked pawns often leads to a disadvantage, further out than a depth of 30 can necessarily find. Their research provided us with a handful of important pieces of information. First, their use of FEN, which keeps track of board position in model training, would have been a powerful tool in prediction. However, time constraints and a limited dataset prevented us from using this implementation. Second, their model accuracy results helped us narrow our model selection. For one, in their research, SVMs performed poorly, so we elected not to test with a support vector machine in our research. Additionally, their metrics on model performance helped us to establish a baseline understanding of what results we would realistically expect with our LSTM model and a more complex task.

Other papers, such as "Learning to Identify Top Elo Ratings: A Dueling Bandits Approach" by Yan et al., do not contribute as profoundly to our work but nonetheless provided helpful insights into the current sphere of academic work in chess. Yan's paper focuses on creating a more efficient way of assigning an accurate Elo so that less processing power/games must be played to gauge a player's ability. Their goal is similar to ours: improved estimation of players' capabilities so that better matchmaking happens faster.

These studies provided a framework to capture, preprocess, and analyze our data. Without it, much of our time would have been spent recreating their findings. While none delve into predicting Elo with machine learning, our findings act as an extension and application of their work.

3 Methods

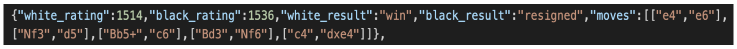
We needed a deep learning model capable of long-term memory and processing sequential data to predict chess players' Elo ratings. As such, to build our model, we used a Long Short-Term Memory (LSTM) model, a variation of a recurrent neural network, for our predictions. This was the optimal choice to ensure the model could understand long-term dependencies and the importance of early-stage moves.

3.1 Datasets

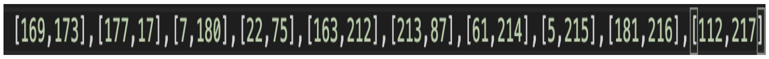
We leveraged the “60,000+ Chess Game Dataset” from Kaggle for our dataset. We narrowed our dataset to exclude chess games that would skew our data. For instance, there exists “game modes” of chess such as “bughouse” and “chess960,” where the rules vary greatly, which would add unnecessary noise to our model. After refining our dataset, we were left with approximately 30,000 games to train and test our model.

3.2 Preprocessing

To preprocess our data, we first stored the move sets in an array of two-element arrays, where the first value represents the white player’s move and the second represents the second player's move.



```
{"white_rating":1514,"black_rating":1536,"white_result":"win","black_result":"resigned","moves":[["e4","e6"],
["Nf3","d5"],["Bb5+","c6"],["Bd3","Nf6"],["c4","dxe4"]]},
```

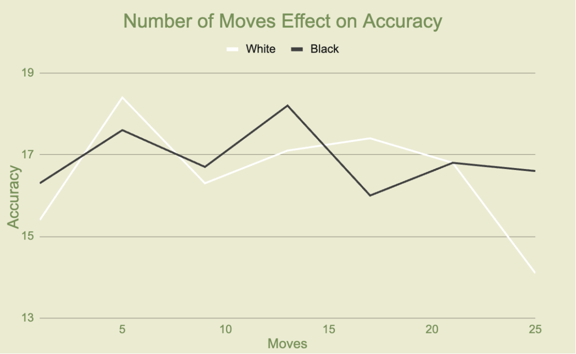


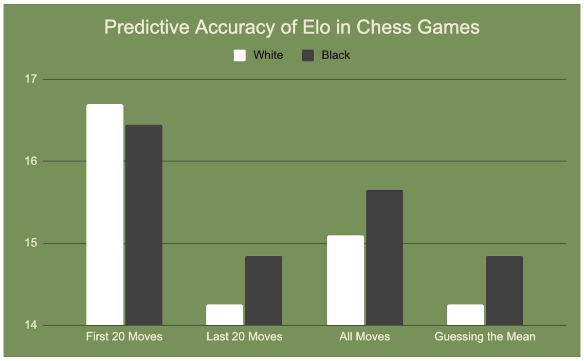
```
[169,173],[177,17],[7,180],[22,75],[163,212],[213,87],[61,214],[5,215],[181,216],[112,217]
```

Then, we would encode each move into a numerical value so that the model could learn from the data. By building the information sequentially, the LSTM can perform at its best.

3.3 Evaluation

We used a combination of two methods to measure our model's effectiveness. The first was cross-entropy loss, which we used to optimize our model while training to make the difference between actual Elo and Elo rating predictions smaller. The second method, from which we derived our results, is accuracy. Instead of predicting an exact accuracy, we used a +/- 200 Elo range. As mentioned in the analysis above of Elo, predicting an exact number for Elo does not encapsulate the purpose of the score. Elo is used as a general gauge of score, not an exact skill level. Using a +/- 200 range, we get the range of a player having approximately a three-in-four chance of beating the predicted score on the high end and a one-in-four chance of beating the predicted score on the lower end. For our results, this was a strict enough metric. However, we did vary this +/- number in our testing. This was extraordinarily beneficial in debugging and in general model evaluation. For the physical steps of our process, we first defined the Elo range for our tolerance to consider passing or not while the model trains. From there, we processed each batch of data, adjusting the hidden to keep our model consistent. We could then test the accuracy using our chosen Elo range. Again, we used an Elo range of +/- 200 for our final results.





4 Result

In our testing, we noticed that lowering the number of considered moves significantly increased our accuracy. As a result, we directed our research towards experimentation with the number of moves and their respective positions in the LSTM model consideration set.

4.1 Data Analyzed

We tested various move consideration sets to test our hypothesis that lowering moves increases accuracy. We tested with sequences of 1 move, 5 moves, 10 moves, 15 moves, 20 moves, and 25 moves. For games included in the set with fewer than the considered moves, we padded right, filling in the remaining moves in the sequence with zeroes to align sizes. Additionally, we tested the last 20 moves, with right padding, so the order was not compromised. We compared this to a simple median guesser baseline model. This model merely guesses the middle of the test data set for every player. While simplistic in nature, it achieved our goal of comparing our model to a pure guessing strategy.

4.2 Insights

1) Impact of less padding: Many of the games in our set were less than 25 moves. In chess, players have the option to resign, and often, after a fatal mistake, they will believe they have lost the game, which may occur within ten moves. This padding of zeroes can make it harder for the model to generalize. By focusing on the first moves, there is a greater guarantee of less/no padding.

2) Point of diminishing returns: In our work, a set of considered moves that is too small tends to overgeneralize the model. Simultaneously, with too many moves, padding would lead to difficulty generalizing at all. Through our work analyzing different move sequences, we identified the point where the move count starts to degrade the quality of the model. This happens around the 10-15 move range.

3) Relative significance of opening moves: Beyond the point of diminishing returns, we had another question. Are the first moves more important than the last moves? This is the reason we tested the last 20 moves. Through our results, we saw that, yes, in fact, the last 20 moves are less significant. This makes sense because

the last moves are more varied than the first ones. Two things make the last moves varied. The first is that the model does not keep track of board position. Because the model does not actively know where any pieces are, it does not comprehend whether a late-game knight move creates a major play or a massive blunder. Second, because of how many games there are in chess, it is likely that in the dataset of 30,000, no endgames are very similar. As a result, the model struggles to find any plausible correlations between moves played and predicted Elo. The first few moves in each game are more standardized than the later moves, and a deviation from this move set can be a strong indicator of a professional or a beginner. As a result, by focusing on these first 10-15 moves, the model can generalize to a much greater degree.

5 Discussion

Our original question was whether an LSTM model can predict player Elo ratings based solely on moves from a singular game. This question formulated a hypothesis about the number of moves and which specific moves work best in creating this prediction. With the first 10-15 moves tracked, we see a notable increase from our baseline model (~2%). From our research, we find viable ways to organize data, what parts of that data to train on, and how the LSTM model should be applied.

Our hope is that future researchers and developers can use our findings with a larger dataset and create a model with a much higher accuracy rate. This model could accurately match new players to games, resulting in a higher retention rate for beginners. This would benefit the chess community as new players grow and inspire the existing sphere.

5.1 Limitations

We identified three main limitations that prevented our model from being more accurate.

1) Limitations from sample size. Our sample, despite its seemingly large size of 30,000, represents a small sample of the possible games of chess that exist. Given that over 10^{40} chess games exist, the model likely often saw games in its test that had no similarity to anything in its train. So, even if the model were generalizing perfectly, it would need help finding connections to patterns it had not seen before. A partial solution to solving this is to focus on the first few moves. However, even in the first 15 moves of a chess game, there are far more than a quadrillion possible outcomes. With a much larger dataset, the model could make these predictions with better accuracy.

2) The fluidity of player capabilities. While Elo can be a strong predictor of the general capabilities of a player, the quality of play in a single game can vary greatly. A single blunder can cause a player to lose the game, and the model can mispredict the player's ability as such. Conversely, an average player can make great moves in a game, either because of pure luck or having played a similar variant before. As

a result, gauging player skill in a singular game can be extremely difficult. Future researchers could look to predict Elo from multiple games off of one player instead of focusing on the outcome and moves of a single game.

3) Lack of board position knowledge. Our model did not consider the board's position in given moves. This means that the model did not understand what players were doing when they made certain moves and could not use the board as a whole to see if a player had an advantage. Using an encoding method such as FEN would help to train the model and generalize better based on board position. For this, a larger dataset would be needed, as with a small dataset, the information on board positions would be too broad for the model to recognize repeated patterns.

5.2 Future Work

Beyond the above-listed improvements to our model, there are two ways are work should be continued. Briefly, these above improvements were: using a larger dataset, predicting player Elo of multiple games, and using FEN to keep track of board positions.

1) The first way future work could be accomplished is to train the model on chess theory openings: hardcoding the opening to a high Elo and providing more weight to the moves following to adjust the Elo downward could be a powerful way of understanding players’ true Elo purely based off their move set. Augmenting the model this way could reveal what deviating moves from theory differentiates a professional player from a beginner.

2) Creating a website where players could input their game data to see their predicted Elo would benefit the community. While this second extension is not research-based, it provides a practical use case of our work that creates a net positive in the chess community. Such a site would encourage new and veteran players to see how well they played in a given game and can help them track their progress and identify their stronger and weaker games.

5.3 Concluding Notes

While more work is required to make our findings a viable product, our work in producing such a model is substantial. Our model’s accuracy results have shown that using an LSTM model focusing on the first 10-15 moves of a move set leads to an increase in accuracy prediction over the baseline. Our ~16.5% accuracy rate shows that while this problem is extraordinarily challenging, predicting a player's Elo with just the moves they play in a game is possible. Future work should be directed towards improving this model and creating a product that chess players can use so that these findings are used for more than just pure research purposes.

Acknowledgements:

Anderson, Mark Robert. “Twenty Years on from Deep Blue vs Kasparov: How a Chess Match Started the Big Data Revolution.” *The*

Conversation, 6 Oct. 2022, theconversation.com/twenty-years-on-from-deep-blue-vs-kasparov-how-a-chess-match-started-the-big-data-revolution-76882.

Bagadia, Sameep, et al. “Analyzing Positional Play in Chess Using Machine Learning.” *Stanford.Edu*, Stanford, Dec. 2014, cs229.stanford.edu/proj2014/Sameep Bagadia, Pranav Jindal, Rohit Mundra, Analyzing Positional Play in Chess Using Machine Learning.pdf.

“Elo Rating System - Chess Terms.” *Chess.Com*, www.chess.com/terms/elo-rating-chess#:~:text=The%20Elo%20rating%20system%20measures,measured%20their%20players’%20skill%20levels. Accessed 1 May 2024.

Glickman, Mark. *The GLICKO System*, 1999, www.glicko.net/research/gdescrip.pdf.

Pulido, H´ector Apolo Rosales. “Predicting the Outcome of a Chess Game by Statistical and Machine Learning Techniques.” *Pàgina Inicial de UPCommons*, Oct. 2016, upcommons.upc.edu/.

Reddy, K. Balachandra, et al. *Predicting Outcomes of Chess Endgames Using Machine Learning*, Apr. 2019, www.ijraset.com/fileserve.php?FID=21847.

Yan12, Xue, and Corresponding to Yali Du ⟨⟨\langleyali.du@kcl.ac.uk⟩⟩\rangle. “Learning to Identify Top Elo Ratings: A Dueling Bandits Approach.” *Ar5iv*, Mar. 2024, ar5iv.labs.arxiv.org/html/2201.04480.