A Machine Learning Approach to Building a Chess Engine

Daniel Cloutier Sagar Patel

December 15, 2020

Contents

Abstract

In the current era, high volumes of data are being collected at an incredible velocity. Much of this data is embedded with valuable knowledge. [2] One example of the types of data that is collected is chess data. In fact, lichess.org, one of many websites where chess can be played online against opponents from around the world, contains a data dump of over one and a half billion games, seventy eight million of them being played in the month of November 2020 alone [5]. We intend to present an automatic learning method for chess, utilizing deep neural networks. We created our deep neural network using a combination of both unsupervised and supervised training. Employing unsupervised training, our engine pre-trains to identify and extract high-level features that exist in a dataset of chess board positions. The supervised training compares two chess positions and evaluates the board which is more favourable, from the white player's perspective. Therefore, we create a deep neural network that can understand chess without incorporating the rules of the game and using no prior manually extracted features. Instead, the system is trained from end to end on a large dataset of chess positions. The resulting deep neural network should allow for simulations that extend the possibilities of chess strategies, highly valuable to chess players at all levels.

1 Introduction

The game of chess is a very popular board game and has become a well liked study for computer enthusiasts, especially in the domain of Artificial Intelligence, from the creation of Deep Blue in 1996 which defeated the world chess champion at the time, Garry Kasparov, up to modern data analysis techniques being tested on chess games as a show for it is possible uses [6][7][8]. Chess has become a good starting point for both more advanced software developers attempting to test something innovative or for newcomers trying to increase their knowledge.

One of the many reasons for the game's popularity is that chess data is collected at an incredibly high volume and velocity. Not only this, but the data collected is often readily available to the public. Databases such as chessbase and lichess contain thousands, if

not millions of games in a standardized notation format for chess called Portable Game Notation. This makes the game of chess a perfect starting point for new tools in Data Science to be tested.

Another of the many domains of study that has been recently using chess as a means of improving itself is that of Machine Learning. Namely, with the advent of algorithms such as AlphaZero [8], some new ideas have risen in an attempt to create algorithms that not only play chess better than humans, but even some that try and mimic human play [6].

Keeping all of this in mind, we wanted to take advantage of the very large amount of data stored by lichess in order to implement and perhaps try and reinvent some already existing ideas using this data that, as far as we could tell, has not been used much in recently published material.

2 Review of Research and Ideas

2.1 Chess Game Categorization

A possibility for using this data is to attempt to categorize chess data into different groups based on some sort of requirements that we could find. For example, perhaps we could group games based on similar openings, similar player ratings or perhaps try to find some different parameters to use to categorize them. One example comes by defining a distance between two games.

One such way is defined as follows. Defining a nine dimensional space, consisting of the x-y location of a piece before it has been moved as well as after, a weight for the piece before it was moved as well as after (because of piece promotion) as well as the piece that was captured, as well as x-y locations for the piece that was captured. From here you can compare moves between games and calculate the distance between them to figure out what games are similar and what games are outliers [2]. This information could be easily stored although incredibly voluminous. Every move of the game would be stored as nine seperate values. Figures 1 and 2 show how storing this information could look.

Move Number	Piece Before Move	Piece After Move	Piece Captured	Moved From	Moved To	Captured At
1	P	P	NULL	e2	e4	e4
2	P	P	NULL	c7	c5	C5
odd#	NULL	NULL	NULL	e1	e1	e1
even#	NULL	NULL	NULL	e8	e8	e8
11	K	K	NULL	e1	g1	g1
11	R	R	NULL	h1	f1	f1
13	P	Q	NULL	e7	e8	e8
21	P	P	P	d5	c6	c5
22	P	P	P	b7	c6	c6

Figure 1: How storing the above information would look like

	K	Q	R	B	N	P	NULL
w()	12	9	5	3	2	1	0

Figure 2: Sample weight values for pieces

From here on we can define the distance between two moves as a euclidean distance between two moves defined by this nine dimensional space. To extend this further, we can define the distance between two whole games as a sum of the distance between corresponding moves. This would be defined as

$$dist(F, G) = \sum_{i=1}^{M} dist(F[i], G[i]) \tag{1}$$

where F[i] is the i-th move of game F, G[i] being the i-th move of game G and M is the greater of the maximum number of moves from either game [2]. Appending a NULL move as necessary.

This kind of categorization could be very useful for new and old players alike. To be able to choose a game and immediately be given results for similar games is a very powerful learning tool. There are often patterns that can emerge from chess positions which can likely be spotted by thorough study of similar positions or games. Also, for an advanced player, studying games that lie outside of the norm can be a good way to surprise unexpecting opponents with new ideas.

The above is in fact a type of unsupervised artificial intelligence. But what we can also do by using data mining techniques is create a supervised pattern matching algorithm to classify similar positions [2]. Once again, this idea can help categorize games into buckets

which can then be looked at if there are certain themes that you know of which appear in a certain group. Seeing other similar games could grant some insight into those types of positions, whether you want to learn the basics of them or want some fresh ideas to improve your play.

2.2 Predicting Game Results

Following from the previous idea, we could try to use the data in an attempt to predict the result of a game. The lichess database contains a lot of information which others might not. Based on our research, most uses of chess data was limited almost exclusively to Grandmaster games, in a specific time control. Lichess contains players of all levels, games are played in all sorts of time controls including fast one minute "bullet" games, all the way up to thirty minute slow games [5]. Using all of this information, along with what was defined previously and the fact that we know the result of every game in the database, we could try and define some way to predict whether a current game of chess, given the player ratings, other similar games, time control and so on, is going to be a win for white, black or a draw.

For example, say we could plot all games according to their similarity distance as defined above [2]. However, if we were to include things like player rating, opening used, time control and so on, we could plot games on multiple dimensions. From here, by looking at games similar to an input game and looking at their results, we could make an attempt at predicting what the result of that game will be in its current state.

There's even more than we can do still. The lichess database starting April 2017 also includes the exact clock times on every single move played. Keeping this in mind means we can improve the above idea further. Some of these could be including the current amount of time each player has, trying to improve the way we define how similar two games are or even by defining a function for determining whether a board position is winning or losing from the white player's perspective assuming best play, not unlike what modern chess computers do [9].

Using the above information we could try and create an algorithm that would predict

whether a game is winning or losing by looking at not only perfect play, but also factoring in things like time trouble and player strength. For a spectator or someone doing analysis, this could be much more insightful as to the dynamics of a position when looking at a game between two specific players instead of looking at a number which assumes best play from both sides.

2.3 Creating a Chess Engine

Moving on with this idea raises the main point for this project. Could we use all of the previous information to create an algorithm that could actually play the game of chess at a human or superhuman level. This appears to be a common theme in the field of Data Science; how to use data to improve the way we look at a certain field. We could attempt to create our own heuristics for what is important and what is not for trying to consider a certain move, however we decided on a different idea.

With a lot of superhuman chess engines like Komodo, Houdini and Stockfish, they have evaluation functions that are based on hard coded values for things like pieces and piece placement, as well as the stage of the game. With this in mind, we considered using something similar along with the rest of the data that we had, but try and make the piece values more accurate via some other resources like regression analysis [7]. This paired with the rest of our data would hopefully allow us to more easily change the difficulty of our chess engine.

Keeping this in mind with the rest of our information, we considered some sort of artificial intelligence that could learn on its own instead of us giving the parameters. We are not chess professionals, but there are indeed ways of getting a computer to learn, especially with large volumes of data like we have at hand [1][3]. This would mean we could generate something useful despite lack of knowledge in the field. Based on our research, this theme comes up often in Data Science, especially when it comes to knowledge discovery. If we give it the parameters that determines whether a position is winning or not, for the computer to make its move, it will only reflect our own knowledge of the game. Of course, seeing as we are not expert chess players, this is not what we

want. If we want to learn something new by creating this chess engine, we believed the best approach would be to let the computer use this data to learn on its own.

3 Problem Statement

Now that we know what we want to do, how exactly do we approach this problem. Surely we are not the only ones that have thought of creating a chess engine that learns on its own. This of course is indeed the case. There have been some attempts at making a self-learning chess engine, however these attempts mainly focus on learning from self-play; the computer knows the rules of chess exclusively and by playing against itself millions of times can learn on its own what wins and what does not [4][8]. This is not necessarily what we want. We already have the data we need, it's now just a matter of getting the computer to learn from it.

From this we can form a formal problem definition: How do we use modern Data Science techniques in order to get a computer to learn how to play chess at a human or superhuman level?

4 Our Approach/Model

5 Algorithm and Discussion

6 Implementation

7 Conclusions

References

[1] Taiwo Oladipupo Ayodele. "Types of machine learning algorithms". In: *New advances in machine learning* 3 (2010), pp. 19–48.

[2] J. A. Brown et al. "A Machine Learning Tool for Supporting Advanced Knowledge Discovery from Chess Game Data". In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2017, pp. 649–654. DOI: 10.1109/ICMLA.2017.00-87.

[3] Peter Harrington. *Machine Learning in Action*. USA: Manning Publications Co., 2012. ISBN: 1617290181.

[4] *Leela Chess Zero codebase*. URL: https://github.com/LeelaChessZero/lc0.

[5] *lichess.org game database*. URL: https://database.lichess.org/. (accessed: 22.10.2020).

[6] Reid McIlroy-Young et al. "Aligning Superhuman AI with Human Behavior: Chess as a Model System". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '20. Virtual Event, CA, USA: Association for Computing Machinery, 2020, pp. 1677–1687. ISBN: 9781450379984. DOI: 10.1145/3394486.3403219. URL: https://doi.org/10.1145/3394486.3403219.

[7] Vladimir Medvedev. *Point Value by Regression Analysis*. URL: https://www.chessprogramming.org/Point_Value_by_Regression_Analysis.

[8] David Silver et al. *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. 2017. arXiv: 1712.01815 [cs.AI].

[9] *Stockfish Code Base*. URL: https://github.com/official-stockfish/Stockfish/blob/master/src/types.h.