IRJMETS

e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:04/Issue:05/May-2022 Impact Factor- 6.752 www.irjmets.com

IMPLEMENTATION OF A CHESS GAME PLAYING SYSTEM USING MACHINE LEARNING

Mr. Suhas Raut*1, Mr. Suksham Fulzele*2, Mr. Parthav Fulzele*3, Mr. Pralay Wahane*4, Prof. Kaveri Deosarkar*5

*1,2,3,4B.E. Student, Wainganga College Of Engineering And Management, Nagpur, Maharashtra, India.

*5Professor, Wainganga College Of Engineering And Management, Nagpur, Maharashtra, India.

ABSTRACT

Chess game playing is an important field of machine learning research. With the emergence of powerful chess engines like Stockfish, Alpha Zero, Leela chess zero, and others. Building a powerful chess engine capable of playing at a superhuman level is no longer the most challenging task. The majority of these engines still rely on highly optimised look-ahead algorithms. CNN (convolutional neural networks), which is usually used for photos and matrix-like data, has proven to be successful at games like chess and go. Chess is being treated as a regression problem in this project. We suggested a supervised learning strategy employing a convolutional neural network with a limited look ahead in this research.

Keywords: Chess Engine, CNN, Neural Network.

I. INTRODUCTION

The concept of creating a chess-playing machine dates back to the eighteenth century. The Turk, a chess-playing automaton, rose to fame around 1769 before being discovered as a fraud. Since then, chess fans and computer engineers have constructed chess-playing devices and computer programmes with varying degrees of seriousness and success. Former World Chess Champion Mikhail Botvinnik was one of the few chess grandmasters to commit himself substantially to computer chess, and he produced several works on the subject. In 1973, the team from Northwestern University, which was responsible for the Chess series of programmes and won the first three ACM Computer Chess Championships, abandoned type B searching. Deep Blue, a brute-force machine capable of inspecting 500 million nodes per second, defeated World Champion Garry Kasparov in 1997, marking the first time in conventional time control when a computer defeated a reigning world chess champion. On December 5, 2017, Alpha Zero, developed by the DeepMind team, achieved a superhuman level of play in these three games within 24 hours of training, defeating worldchampion programmes stockfish and the three-day version of AlphaGo Zero. Self-play reinforcement learning is used to train AlphaGo Zero. To accomplish consistent learning, it integrates a neural network and Monte Carlo Tree Search in an elegant policy iteration architecture.

Chess is a thoroughly observed two-player zero-sum game. The chessboard is defined as a simple 8 by 8 matrix with pieces and blank areas. There are two players, one of whom is Black and the other is White. Pawns, rooks, knights, bishops, kings, and queens are the six pieces of chess.. Each piece moves independently on the board. The goal is to checkmate your opponent's king. Short Castling, Long Castling, and En passant are additional moves that are achievable under certain conditions.

II. METHODOLOGY

A. DATASET

[Dataset Source, Dataset Form , Data Set Analysis]

We have referred to the same FICS (https://www.freechess.org/) chess database with an Elo rating of 2000 and above for the year 2020, as we did in Oshri, Barak [2015]. The database contains all chess games in PGN (Portable Game Notation) format. The chess database comprises 44029 chess games in total.

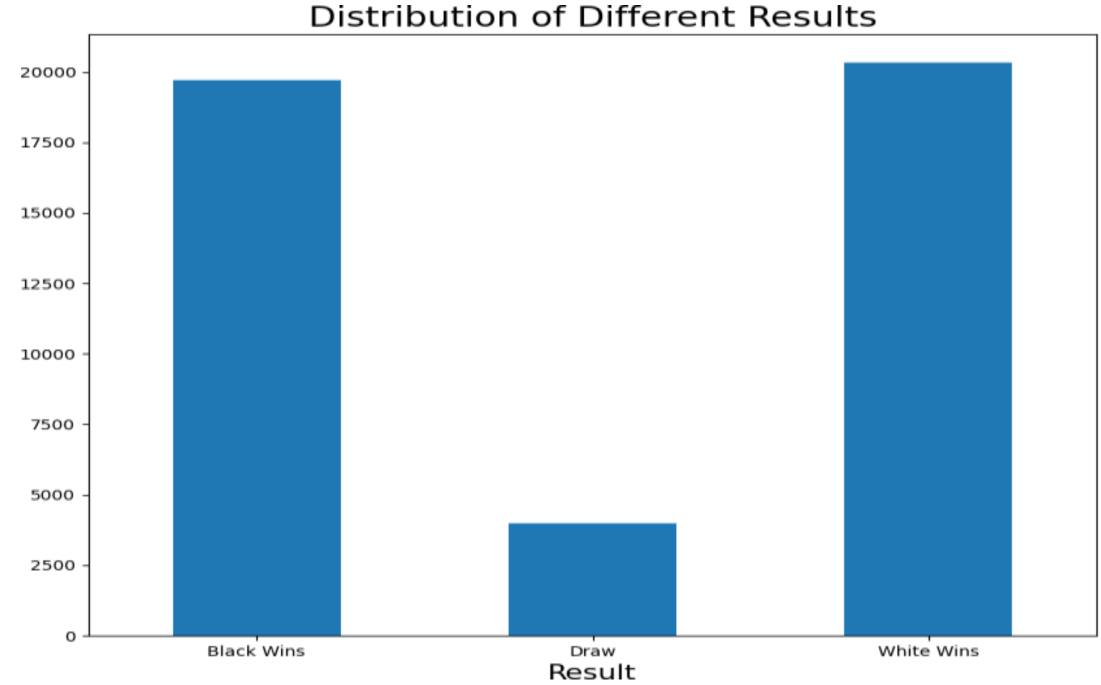

Fig 1: Histogram displaying total black/white and draw wins

B. PREPROCESSING OF DATA

In comparison to normal ACN(Algebraic chess notations) format, bitboard representation delivers the best performance during data training. The chess board is represented as an 8x8x12 bit vector of 0s and 1s in the bitboard representation, with each of the 12 pieces represented by an array of 1s and blank space by 0s.

Chess is depicted in a categorical manner. Turn bit (single bit to indicate turn), castling bit (four bits to denote castling right of both pieces), and check bit are among the other attributes (two bits which represent whose king is in check).

A vector with 775 binary characteristics represents the chessboard. The bitboard representation is used to show where each piece in the vector is located. The first 64 features indicate white pawn locations, the next 64 features represent white knight placements, and so on. The following is the bitboard representation of white pawns:

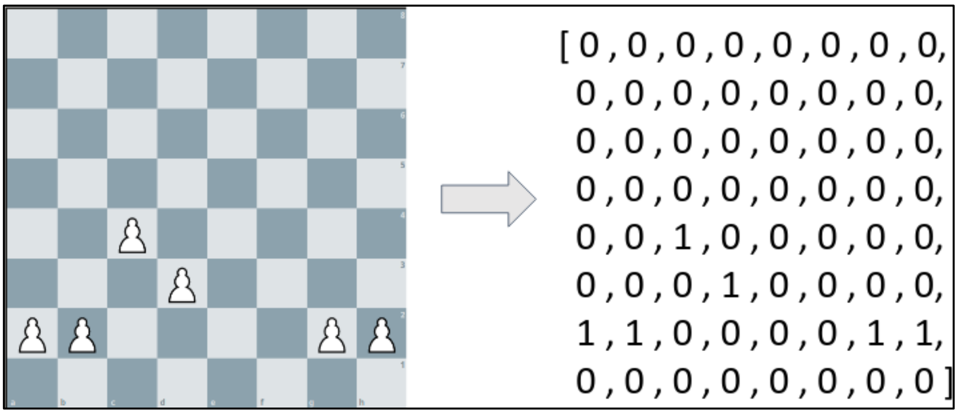

Fig 2: Figure displaying white bitboard representation

Pawns on the board (The figure on the right depicts a One Dimensional Matrix)

The turn bit, casting bit, and check bit are then concatenated with all of the piece vectors.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:05/May-2022 Impact Factor- 6.752 www.irjmets.com

C. MODEL

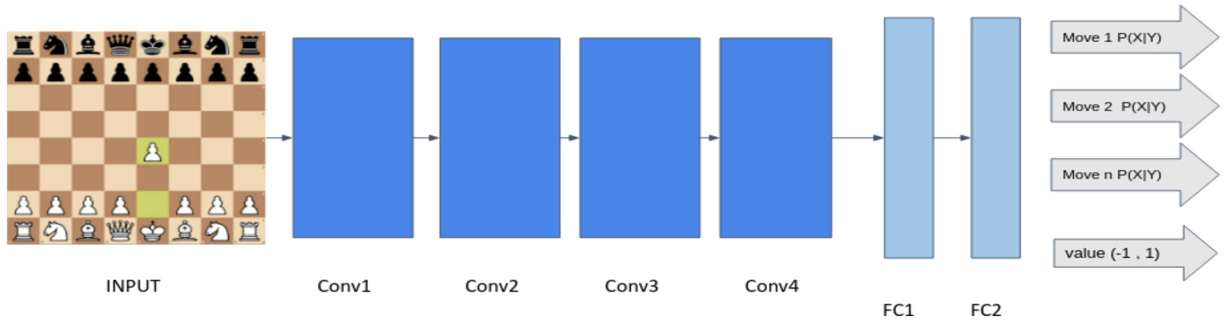

Fig 3: Visualization of neural network architecture and move prediction network

PyTorch is used to implement the neural network. The chessboard is fed into the convolutional neural network in the form of FEN (Forsyth Edward notation), which is then translated into a bitboard representation and followed by four 2D convolutional layers and two fully connected layers. The first layer's kernel size is 3x3, the second layer's size is 2x2, and the remaining two layers' sizes are 1x1. The filters utilised were in the numbers 8, 16, 32, and 64. ReLUactivation is used in every layer. The tanh activation function was utilised for value activation. The loss function was MSE (Mean squared error), and Adam's optimizer with a batch size of 256 was employed. The training was spread out across 100 epochs.

III. RESULT

When tested against a Stockfish engine with a 1300 Elo rating, it was able to win about 10% of the games and draw 30% of the games with a sample size of One Hundred games. It lost all of the games using an Elo 2000 stockfish engine.

It was capable of detecting numerous tactics, as evidenced by the boards below.

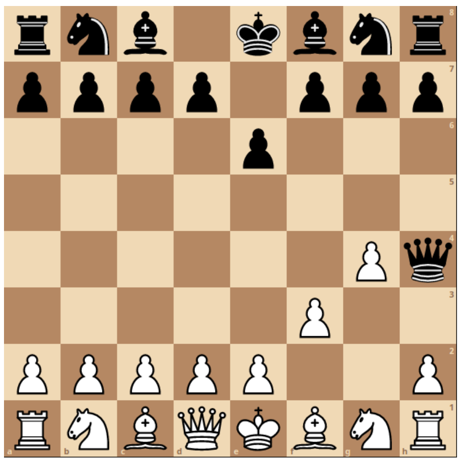

Figure 4: In one move, checkmate Predicted

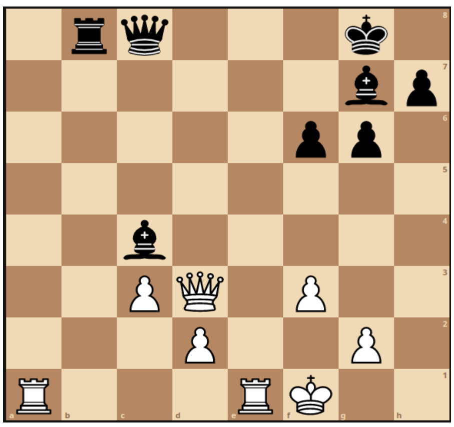

Figure 5: Pinning the Queen

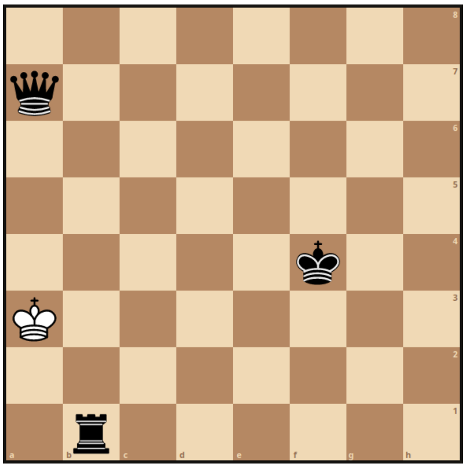

Figure 6: The checkmate was delivered in three moves in the endgame..

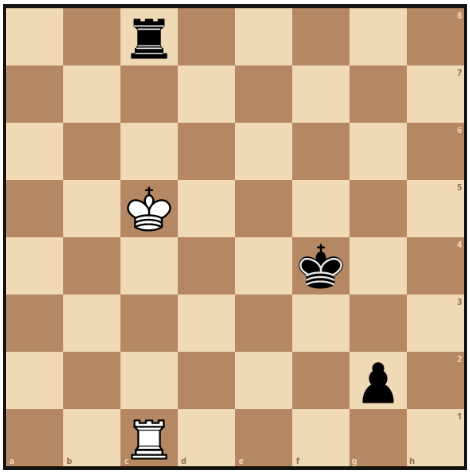

Figure 7: To win the rook, Checked the king

IV. CONCLUSION

Using publicly available data from FICS, we trained the convolutional neural network to play chess. The trained model we received was capable of detecting patterns on the chessboard and delivering numerous methods in order to improve and win the game. We conclude that a more refined and larger dataset with more high-level games can yield a better model. The model can be enhanced by providing additional training time and benefiting from the use of better hardware. The neural network can be improved by utilising a Residual Convolutional Neural Network.

V. REFERENCES

[1] T.Sabatelli, Matthia, Francesco Bidoia, Valeriu Codreanu, and Marco Wiering. "Learning to Evaluate Chess Positions with Deep Neural Networks and Limited Lookahead." InICPRAM, pp. 276-283. 2018.

[2] Vikström, Joel. "Training a Convolutional Neural Network to Evaluate ChessPositions." 2019.

[3] Oshri, Barak. "Predicting Moves in Chess using Convolutional Neural Networks." 2015.

[4] Schrittwieser, Julian, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, LaurentSifre, Simon Schmitt, Arthur Guez et al."Mastering atari, go, chess and shogi by planning with a learned model." arXiv preprint arXiv:1911.08265. 2019

[5] Maesumi, Arman. "Playing Chess with Limited Look Ahead." arXiv preprintarXiv:2007.02130. 2020.

[6] David, Omid E., Nathan S. Netanyahu, and Lior Wolf. "Deepchess: End-to-end deep neural network for automatic learning in chess." In International Conference on Artificial Neural Networks, pp. 88-96. Springer, Cham, 2016.

[7] Maddison, Chris J., Aja Huang, Ilya Sutskever, and David Silver. "Move evaluation in Gousing deep convolutional neural networks." arXiv preprint arXiv:1412.6564. 2014.

[8] Mcllroy-Young, Reid, Siddhartha Sen, Jon Kleinberg, and Ashton Anderson. "Aligning Superhuman Al with Human Behavior: Chess as a Model System." In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1677-1687.2020.

[9] Thrun, Sebastian. "Learning to play the game of chess." In Advances in neural information processing systems, pp. 1069-1076. 1995.

[10] Lai, Matthew. "Giraffe: Using deep reinforcement learning to play chess." arXivpreprint arXiv: 1509. 01549 2015

[11] Chess Computer History https://en.wikipedia.org/wiki/Computer_chess

[12] Chess Programming wiki https://www.chessprogramming.org/Main_Page

[13] Alpha zero tutorial https://web.stanford.edu/~surag/posts/alphazero.html

[14] Erik Bernhardsson, "Deep Learning for... Chess", 2014, https://erikbern.com/2014/11/29/deep-learning-for chess.html

[15] Chess Programming Wiki "Neural Networks for chess" 2020, https://www.chessprogramming.org/Neural_Networks

[16] Al Chess Algorithms https://www.cs.cornell.edu/boom/2004sp/ProjectArch/Chess/algorithms.html

[17] Twitch chess https://github.com/geohot/twitchchess

[18] Predicting Professional Players Moves with Deep Learning https://towardsdatascience.com/predicting professional-players-chess-moves-with-deep-learning 9de6e305109e

[19] David Foster "Alpha go zero explained in one diagram"2017, https://medium.com/applied-datascience/alphago-zero-explained-in-one-diagram365f5abf67e0