

# Evaluation of Strategy by AlphaZero and StockFish on Chess Game

Sreerag SR

M.TECH, NIT Kurukshetra  
E-mail: sreerag.sr@gmail.com

**Abstract**—The chess game is the most researched domain in the history of artificial intelligence. The winning of this game needs calculating the whole probability of the possibility of the player and opponents. The AlphaZero program recently achieved superhuman performance in the game of chess learning from games of self-play. StockFish is dominant long time for chess game performance. StockFish failed to win even one game against AlphaZero, but 28 games won by AlphaZero out of 100. In this Paper we discussed the strategy used by AlphaZero by evaluating selective games between StockFish and AlphaZero.

## Introduction

Artificial intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans and other animals. AI is going to be using on all researches and in whole technology and its performance is measured with complex games like chess. StockFish and AlphaZero are two most successful and developed programmed based on artificial intelligence.

StockFish is a free and open-source chess engine, available for various desktop and mobile platforms. It is developed by Marco Costalba, Joona Kiiski, Gary Linscott and Tord Romstad, with many contributions from a community of open-source developers. Up to the end of 2017 it was most winning chess programme but atn December 2017, StockFish failed to AlphaZero in a tournament of 100 games and AlphaZero won 28 games against Stock Fish.

AlphaZero was developed by Alphabet Inc.'s Google Deep Mind in London. AlphaZero's underlying algorithms are potentially more general-purpose, and may be evidence that the scientific community is making progress towards artificial general intelligence. It replaces the handcrafted knowledge and domain specific augmentations used in traditional game-playing programs with deep neural networks and a tabula rasa reinforcement learning algorithm.

## Evaluation of game

On December 4th, 2017, Google Headquarters in London applied their AlphaZero to the game of chess. The event was

more of an experiment than a chess exhibition, and the results are ground-breaking in both the fields of computing and chess.

Rather than relying on the classic "alpha-beta algorithm" common to conventional chess software, AlphaZero uses a deep neural network and is trained solely by reinforcement learning from games of self-play. It scans only 80, 000 positions per second compared to Stock Fish's 70 million. AlphaZero played StockFish 100 games, winning 28 and drawing the rest. Table 1 is listing the AlphaZero performance against Stock Fish.

**Table 1: AlphaZero Performance against StockFish in chess**

White	Black	Win	Draw	Loss
AlphaZero	Stock Fish	25	25	0
Stock Fish	AlphaZero	3	47	0

Out of 100 games we are evaluating 2 games which AlphaZero won the game. And In this paper two important games is evaluated deeply and about 3 other games general strategies has been discussed.

## Game1: AlphaZero-Black, Stock Fish-White

1. e4 e5 2. Nf3 Nc6 3.Bb5 Nf6 4.d3 Bc5 5. Bxc6 dxc6 6.0-0 Nd7 7.c3 0-0 8. d4 Bd6 9. Bg5 Qe8 10.Re1 f6 11.Bh4 Qf7 12.Nbd2 a5 13.Bg3 Re8 14.Qc2 Nf8 15.c4 c5 16.d5 b6 17.Nh4 g6 18.Nhf3Bd7 19.Rad1 Re7 20.h3 Qg7 21. Qc3 Rae8 22.a3 h6 23.Bh4 Rf7 24.Bg3 Rfe7 25.Bh4 Rf7 26.Bg3a4 27.Kh1 Rfe7 28.Bh4 Rf7 29.Bg3 Rfe7 30.Bh4 g5 31.Bg3 Ng6 32.Nf1 Rf7 33.Ne3 Ne7 34.Qd3h5 35.h4 Nc8 36. Re2 g4 37.Nd2 Qh7 38.Kg1 Bf8 39.Nb1 Nd6 40.Nc3 Bh6 41.Rf1 Ra8 42.Kh2Kf8 43.Kg1 Qg6 44.f4 gxf3 45.Rxf3 Bxe3+ 46.Rfxe3 Ke7 47.Be1 Qh7 48.Rg3 Rg7 49.Rxg7+Qxg7 50.Re3 Rg8 51.Rg3 Qh8 52.Nb1 Rxg3 53.Bxg3 Qh6 54.Nd2 Bg4 55.Kh2 Kd7 56.b3 axb357.Nxb3 Qg6 58.Nd2 Bd1 59.Nf3 Ba4 60.Nd2 Ke7 61.Bf2 Qg4 62.Qf3 Bd1 63.Qxg4 Bxg4 64.a4Nb7 65. Nb1 Na5 66.Be3 Nxc4 67.Bc1 Bd7 68.Nc3 c6 69.Kg1 cxd5 70.exd5 Bf5 71. Kf2 Nd6 72.Be3 Ne4+ 73.Nxe4 Bxe4 74.a5 bxa5 75.Bxc5+ Kd7 76.d6 Bf5 77. Ba3 Kc6 78.Ke1 Kd5 79.Kd2Ke4 80.Bb2 Kf4 81.Bc1 Kg3 82.Ke2 a4 83.Kf1 Kxh4 84.Kf2 Kg4 85. Ba3 Bd7 86. Bc1 Kf5 87.Ke3 Ke6 0-1

### Strategy of AlphaZero

- While AlphaZero is playing by black, it likes to develop closed pawn system. It eases the prediction of the opponent's movement; also it gives time for AlphaZero for preparing from defense to offence position.
- Pawn Structure is important before start of exchanging the important pieces, the pieces to be sacrificed are decided based on the pawn structure.
- It has great importance of the pieces which is not in board to decide the pawn structure.
- AlphaZero choose castling on the same side while playing defence and prefer an early castling.
- King is having important role on the attack. The early movement of king in the game shows calculation of the AlphaZero.
- AlphaZero ease the game by reducing the opponent's pieces by exchanging and reducing possibility of activating the opponent's pieces.

#### StockFish weakness:

- StockFish fails to use the pawns for breaking the closed pawn structure.
- King was too late for attacking position.
- StockFish fails to predicts importance of the its and opponents pieces before exchanging pieces.
- While StockFish was playing with white, it can form good pawn structure earlier but it failed.

#### Game2: AlphaZero-White, Stock Fish-Black

1.Nf3 Nf6 2. d4 e6 3.c4 b6 4.g3 Bb7 5. Bg2 Be7 6.0-0 0-0 7.d5 exd5 8.Nh4 c6 9.cxd5 Nxd5 10.Nf5 Nc7 11.e4 d5 12.exd5 Nxd5 13. Nc3 Nxc3 14.Qg4 g6 15.Nh6+ Kg7 16.bxc3 Bc8 17. Qf4 Qd6 18.Qa4 g5 19.Re1 Kxh6 20.h4 f6 21.Be3 Bf5 22.Rad1 Qa3 23.Qc4 b5 24.hxg5+ fxg5 25. Qh4+Kg6 26.Qh1 Kg7 27.Be4 Bg6 28.Bxg6 hxg6 29.Qh3 Bf6 30.Kg2 Qxa2 31.Rh1 Qg8 32.c4 Re8 33.Bd4 Bxd4 34.Rxd4 Rd8 35.Rxd8 Qxd8 36.Qe6 Nd7 37.Rd1 Nc5 38.Rxd8 Nxe6 39.Rxa8 Kf6 40.cxb5 cxb5 41.Kf3 Nd4+ 42.Ke4 Nc6 43.Rc8 Ne7 44.Rb8 Nf5 45.g4 Nh6 46. f3 Nf7 47. Ra8 Nd6+48.Kd5 Nc4 49.Rxa7 Ne3+ 50.Ke4 Nc4 51.Ra6+ Kg7 52.Rc6 Kf7 53.Rc5 Ke6 54.Rxg5 Kf6 55.Rc5 g5 56. Kd4 1-0

### Strategy of AlphaZero

- Freeing the central of the board from pieces is important because and it fastens the attack.
- Bishop on G2 is the most attacking and defensive piece while castling has done on king side.
- Rook getting D and E files and vacating central board and bishop on G2 fast the attack.

- Breaking the pawn structure near opposite king eases attack.
- Deactivating of opposite pieces by reducing its movement possibility strengthens the attack.

#### StockFish Weakness:

- Open files are fully exposed to rook attack of AlphaZero and StockFish failed to connect the rook for defending the same.
- StockFish fails to control the centre of the board.
- StockFish fail to built-up proper pawn structure of the centre of the board.

### Conclusion

- Attacking on E & F file is important and during attack AlphaZero clears this file without any pawns for attacking fast.
- Placing bishop on G2 and castling on king side is the best attacking defensive position
- Compared to human game, before fully development of pieces, AlphaZero starts attack and taking advantage of it.
- AlphaZero gives important for positioning the piece than developing the pieces.
- AlphaZero simplifies the game by exchanging the pieces.
- Castling early is a defense strategy of AlphaZero.

### References

- [1] Online chess games database, 365chess, 2017. URL: <https://www.365chess.com/>.
- [2] Robert Levinson, T Anthony Mersl and, and David E.Wilkins. The role of chess in Artificial Intelligence Research
- [3] Emanuel Lasker. Common Sense in Chess. Dover Publications, 1965.
- [4] Wilhelm Steinitz. The Modern Chess Instructor. Edition Olms AG, 1990.
- [5] Hsu et al., 1990] F- Hsu, T. Anantharaman, M. S. Camp-bell, and A. Nowatzyk. A grandmaster chess machine.
- [6] [de Groot, 1965] A- D. de Groot. Thought and Choice in Chess, The Hague, 1965.
- [7] Niblett and Shapiro, 1981] T. Niblett and A. Shapiro. Automatic induction of classification rules for chess endgames. Technical Report MIP-R-129, Machine Intelligence Research Unit, University of Edinburgh, 1981.
- [8] [Levinson, 1989] R. Levinson. A self-learning, pattern oriented chess program. Intern. Computer Chess Assoc. Journal, 12(4):207-215, December 1989.