AlphaZero: The New Boss

Akshat Kumar Jain[1], Gulista Khan[2]
Faculty of Engineering and Computing Sciences
iakshatkumarjain@gmail.com, Gulista.khan@gmail.com

***Abstract*— The field of chess raises very interesting challenges to computer science and in particular to Artificial Intelligence. Indeed, as we will see, computational models of chess need to take into account important elements of advanced human problem solving capabilities such as knowledge representation, reasoning, and learning. In this paper I describe and compare two main engines on the basis of a chess tournament. A computer program is capable of carrying and describing some creative moves of chess. The results will definitely surprising.**

Keywords **-- AlphaZero, Reinforcement Learning , Stockfish, IBM Deep Blue ,MONTE Carlo tree search, Alpha-beta tree search.**

INTRODUCTION

There has been research going on since the last few decades on how to beat a human expert or master level player with the algorithm. Many scientists like Charles babbage (The turk[1]) , Alan turing , CLAUDE E. SHANNON[2] , John von Neumann[3], Planning algorithms for artificial intelligence is always challenging. In designing the Model for the Chess game , different scientists give different approaches like when we talk about the John von Neumann he proposed two algorithms . The first one is based on the brute force search and the second one works on the concept of intelligence strategy that can simulate human intelligence (i.e. think like a human.Systems that have been successful in chess**:**There are some highly specified systems that have been successful in chess and also have had impact in real world domains like robotics , industrial control , or intelligent assistants. Mostly these systems use Model-based Reinforcement Learning (RL) that first learn the model of the environment's dynamics , and then planning with respect to learning model . Or we have Model-free RL -i.e. they took the rules from the or we say optimal policies /value functions and start from the scratch Some highly sophisticated systems are:

IBM's Deep Blue

The IBM Deep Blue supercomputer that defeated world chess champion Garry kasparov in 1997 employed 480 custom chess chips. This technique describes the design philosophy, general architecture, and performance of the chess chips, which provided most of deep blue's computational power. Article given by [4] in 1999 given a engine for this technique that was under IEEE micro.

A. STOCKFISH-

Stockfish is an open source chess engine developed by Tord Romstud , Marco Costalba and Joona Kiiski. This project started in November 2008, since then it has evolved only. It has state-of-art-engines based on Alpha-Beta Search and till now it has reached an indisputably superhuman in terms of chess. It can compare 70,000,000 moves per second as a player. This engine has been evaluated by human grandmasters since 2008. Also search extensions that are highly optimized using game specific heuristic.

B. AlphaZero-

Now, the latest technology is using two main methodologies, first one is deep neural networks and the second one is general reinforcement learning algorithm. It starts learning the game from scratch (Like babies learn games from scratch) but obviously with more computational power .

II. LITERATURE REVIEW

In 2017 ,Google's AlphaZero takes on chess on 4th december in Google headquarters, London. **AlphaZero surpasses the Stockfish 8 in 4 hours .**

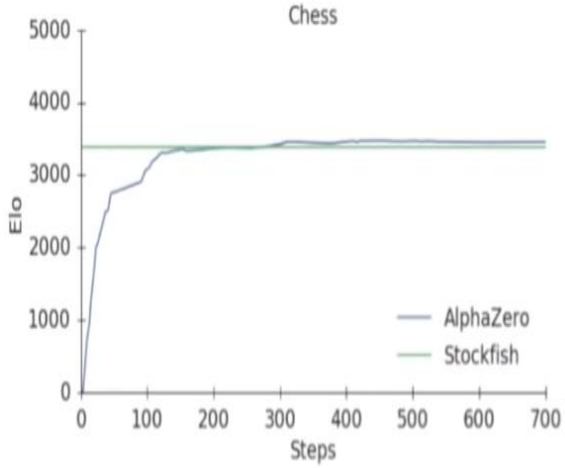

Figure 1: Elo rating of Stockfish v/s AlphaZeroa

In figure 1 x axis represents the measure of time which is concluded as training steps in thousands and y axis represents the elo rating in the game of chess.

Table 1:Summary of AlphaZero Results

Summary of AlphaZero Results

	Win	Draw	Lose
Chess vs. 2016 TCEC world champion *Stockfish*	28	72	0

Condition for this game is, both the engines will be allowed to have 1 minute per move. Stockfish is the strongest skill using 64 threads, same algorithm setting and architecture (except input/output planes) for all games.

Anatomy Of A World Champion Chess Engine : STOCKFISH

Domain knowledge, extensions, heuristics in 2016 **Top Chess Engine Championship(**TCEC) world champion Stockfish.


Anatomy of a World Champion Chess Engine

Domain knowledge, extensions, heuristics in 2016 TCEC world champion *Stockfish*:

Board Representation: Bitboards with Little-Endian Rank-File Mapping (LERF), Magic Bitboards, BMI2 - PEXT Bitboards, Piece-Lists, **Search:** Iterative Deepening, Aspiration Windows, Parallel Search using Threads, YBWC, Lazy SMP, Principal Variation Search. **Transposition Table:** Shared Hash Table, Depth-preferred Replacement Strategy, No PV-Node probing, Prefetch **Move Ordering:** Countermove Heuristic, Counter Moves History, History Heuristic, Internal Iterative Deepening, Killer Heuristic, MVV/LVA, SEE, **Selectivity:** Check Extensions if SEE >= 0, Restricted Singular Extensions, Futility Pruning, Move Count Based Pruning, Null Move Pruning, Dynamic Depth Reduction based on depth and value, Static Null Move Pruning, Verification search at high depths, ProbCut, SEE Pruning, Late Move Reductions, Razoring, Quiescence Search, **Evaluation:** Tapered Eval, Score Grain, Point Values

Figure 2: Depicting the Anatomy of a world champion chess Engine

B. Anatomy Of AlphaZero

Self-Play Reinforcement Learning with Monte Carlo Tree Search.


Anatomy of AlphaZero

Self-play reinforcement learning + self-play Monte-Carlo search

Board Representation: Bitboards with Little-Endian Rank-File Mapping (LERF), Magic Bitboards, BMI2 - PEXT Bitboards, Piece-Lists, **Search:** Iterative Deepening, Aspiration Windows, Parallel Search using Threads, YBWC, Lazy SMP, Principal Variation Search. **Transposition Table:** Shared Hash Table, Depth-preferred Replacement Strategy, No PV-Node probing, Prefetch **Move Ordering:** Countermove Heuristic, Counter Moves History, History Heuristic, Internal Iterative Deepening, Killer Heuristic, MVV/LVA, SEE, **Selectivity:** Check Extensions if SEE >= 0, Restricted Singular Extensions, Futility Pruning, Move Count Based Pruning, Null Move Pruning, Dynamic Depth Reduction based on depth and value, Static Null Move Pruning, Verification search at high depths, ProbCut, SEE Pruning, Late Move Reductions, Razoring, Quiescence Search, **Evaluation:** Tapered Eval, Score Grain, Point Values Midgame: 198, 817, 836, 1270, 2521, Endgame: 258, 846, 857, 1278, 2558, Bishop Pair, Imbalance Tables, Material Hash Table, Piece-Square Tables, Trapped Pieces, Rooks on (Semi) Open Files, Outposts, Pawn Hash Table, Backward Pawn, Doubled Pawn, Isolated Pawn, Phalanx, Passed Pawn,

Figure 3: Depicting the Anatomy of AlphaZero

III. ANALYSIS OF ALGORITHMS

Now, we see how AlphaZero dominates Stockfish 8 in the 100 game match and didn't lose a single match against it. And take into consideration that Stockfish was already 10 years old when the match happened between these two engines. So stockfish was more greater in terms of experience and as we go further we will see that stockfish 8 check 70,000,000 moves per second and AlphaZero is around 80,000 moves per second still the matches goes in the favour of AlphaZero. In AlphaZero we use Monte Carlo Tree Search (MCTS) instead of Alpha-Beta Tree Search which was previously used in earlier systems.

The factors we take into consideration :

Scalability With Search Time

As we have seen alpha-beta search is dominated in these domains for 50 years now and there have been many studies that suggested MCTS or any other algorithm could not be competitive to alpha-beta search but here we that MCTS not only outperform the other programs but actually scaled up more efficiently than alpha-beta search .

Table 2: Depicting Moves per second by Stockfish and AlphaZero

Positions / second	*Stockfish*	*Elmo*	*AlphaZero*
Chess	70,000,000		80,000

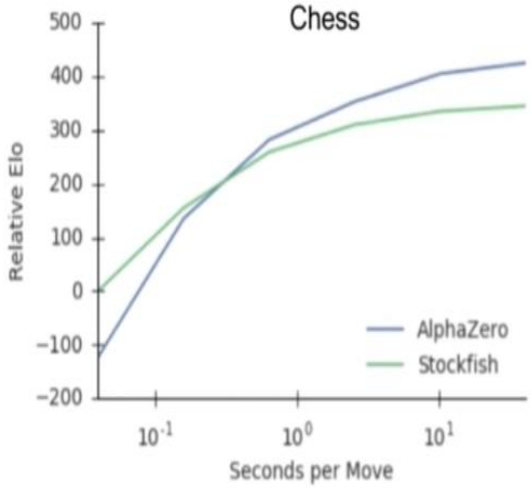

Figure 4: Depicting Moves per second by Stockfish and AlphaZero

It is similar to what John von neumann proposed in Type B[3].

B. AlphaZero Self Play

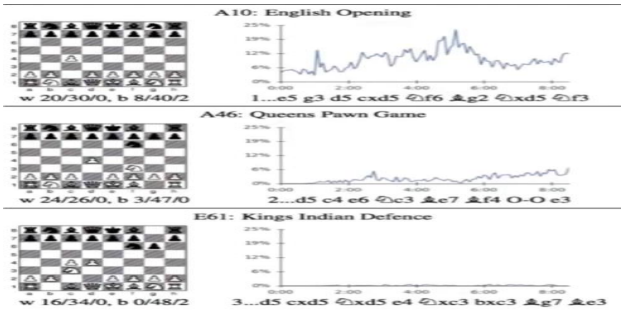

Figure 5(a): Opening moves Alphazero self play

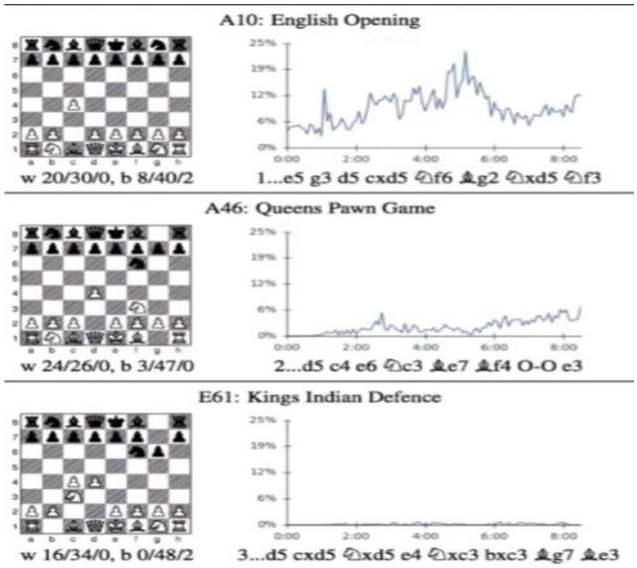

Figure 5(b): Opening moves Alphazero self play

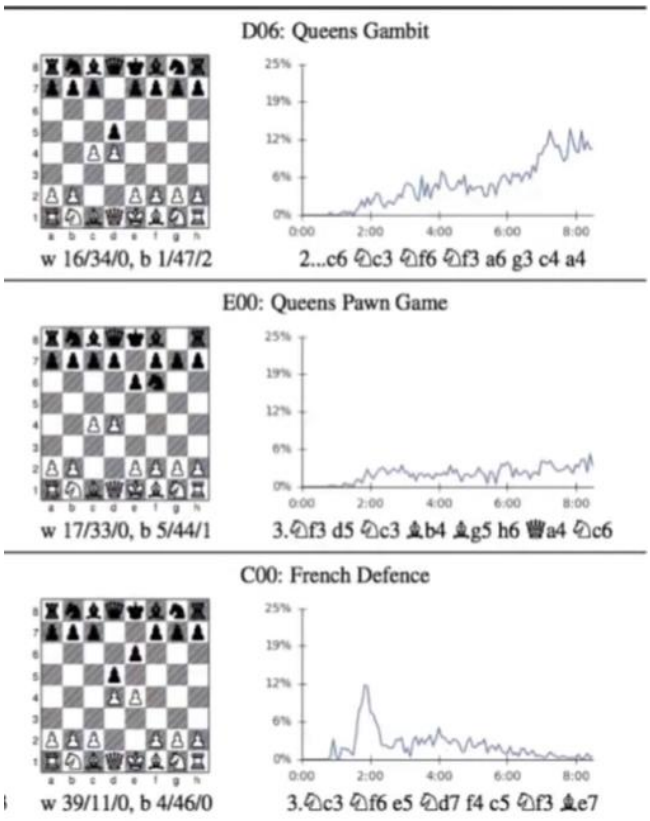

Figure 5(c): Opening moves Alphazero self play

AlphaZero trained for 9 hours by playing 44 million games against itself. Alpha Zero's openings are amazing. AlphaZero plays in a universal and balanced way, having both, the best of the humans and the computers. AZ's tactical strength is overwhelming, but it comes together with a deep strategic knowledge, only seen until now exclusively with humans, as you can see in the game using the Berliner Defence, which wins in 87 movements. AZ plays very well in blocked positions, as shown during the two games played with the French Defence too. AZ's openings are amazing: it has discovered and even overcome five centuries of human effort in hardly two hours of training. When AZ plays with black pieces, it shows a very solid position and rapidly occupies the center in a symmetrical way, like Karpov's style. When playing with white pieces, AZ likes to start with the Queen, but in an aggressive way, like Kasparov's style. AZ loves to give away pieces in the long term, with tactical sacrifices such as Tal's and other positional ones who could be signed by Petrosian. The way AZ plays against the Queens Indian Defence is sublime, from another galaxy, with outstanding ideas such as the sacrifice of the pawn in c4 during the match that wind in 68 movements. It is also impressive AZ's relentless execution of its positional advantage in some of the matches where AZ plays in inferiority, as it is its faultless technique in the final positions. We could easily say, without a risk of making a mistake that, from the analysis of the 10 games, AZ plays close to perfection.

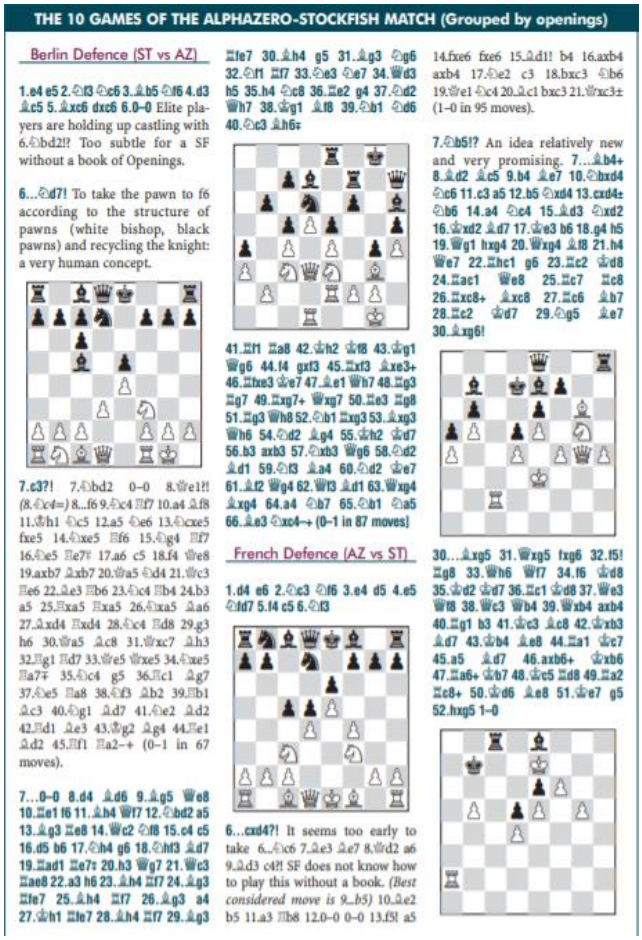

Figure 6. Games play by Stockfish and AlphaZero

CONCLUSIONS

Computer chess is most studied domain in history of Artificial Intelligence. AI algorithm development has been started since few decades. Some highly specified systems have been successful in chess like Deep Blue defeated kasparov in 1997, state of art now in disputable superhuman. Chess have more win over stockfish. State of art engines are based on alpha beta search.

REFERENCES

[1] Sources available at link https://spectrum.ieee.org/tech-talk /tech-history/ dawn-of-electronics/untold-history-of-ai-charles-babbage-and-the-turk

[2] .Sources available at link https://www.pi.infn.it/~carosi/chess/ shannon.txt

[3] Sources available at linhttps://timmccloud.net/blog-6%e2%80%8a-%e2%80%8aearly-ai-geniuses-claude-shannon-and-chess/

[4] .Sources available at link http://www.csis.pace.edu/~ctappert/dps/pdf/ai-chess-deep.pdf

[5] Sources available at link https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go

[6] Sources available at link https://deepmind.com/research/publications/Mastering-Atari-Go-Chess-and-Shogi-by-Planning-with-a-Learned-Model

[7] Sources available at link https://deepmind.com/research/publications/general-reinforcement-learning-algorithm-masters-chess-shogi-and-go-through-self-play

[8] Sources Available at linkhttps://blog.stockfishchess.org/

.