Matthew Sadler and Natasha Regan

Game Changer

AlphaZero's Groundbreaking Chess Strategies and the Promise of AI

New In Chess 2019

Contents

Preface

This book is about an exceptional chess player, a player whose published games at the time of writing total just 10, but whose name already signifies the pinnacle of chess ability. A powerful attacker, capable of defeating even the strongest handcrafted chess engines with brilliant sacrifices and original strategies; and a player that developed its creative style solely by playing games against itself.

That player is AlphaZero, a totally new kind of chess computer created by British artificial intelligence (AI) company DeepMind.

Through learning about AlphaZero we can harness the new insights that AI has uncovered in our wonderful game of chess and use them to build on and enhance our human knowledge and skills. We talk to the people who created AlphaZero, and discover the struggles that brilliant people face when aiming for goals that have never before been achieved.

The authors feel extremely privileged to have worked with the creators of AlphaZero on this project. We recognise this as a defining moment, being right at the cutting edge of fast-developing technology that will have a profound effect on all areas of human life.

Our collaboration arose following the publication of 10 AlphaZero games during the December 2017 London Chess Classic tournament. The previous year, Matthew and Natasha had won the English Chess Federation (ECF) Book of the Year award for *Chess for Life*, a compilation of interviews with icons of chess, highlighting themes and core concepts of their games. We knew we could take a similar approach to AlphaZero, offering critical insight into how the AI thinks and plays, and sharing key learnings with the wider chess-playing community.

Who should read this book?

- *keen chess players, looking to learn new strategies*

AlphaZero's chess is completely self-taught, stemming from millions of games played against itself. Much of its play matches the accepted human wisdom gathered over the past 200 years, which makes AlphaZero's play intuitive, allowing humans to learn from it. This book brings out AlphaZero's exquisite use of piece mobility and activity, with guidance from Matthew through the simple, logical, schematic ways in which AlphaZero builds up attacks against the opponent's king's position. We believe these techniques will inspire professionals and club players alike.

- *artificial intelligence enthusiasts*

As Demis Hassabis, CEO of DeepMind, explains, the application of AI to games is a means to something greater: 'We're not doing this to just solve games, although it's a fun endeavour. These are challenging and convenient benchmarks to measure our progress against. Ultimately, it's a stepping stone for us to build general-purpose algorithms that can be deployed in all sorts of ways and in all sorts of industries to achieve great things for society.'

Our interviews with the creative people who designed and built AlphaZero are full of insights that, using chess as an example, help us to better understand the opportunities and challenges afforded by AI.

- *chess enthusiasts*

As well as providing instructional material, this book is also a collection of fascinating games of astonishing quality, featuring dashing attacks, unexpected strategies, miraculous defences and crazy sacrifices. Matthew compared playing through these games to uncovering the lost notebooks of a great attacking player of the past, such as his hero Alexander Alekhine, and finding hundreds of hitherto unpublished ideas.

How to read this book

The chess content of this book is arranged in discrete chapters and designed to be read out of sequence, so it is perfectly possible to pick a theme you are interested in and start in the middle of the book. The chess content is not too heavy, with an emphasis on explanations rather than variations. We would recommend playing through the games with a chessboard. In our opinion, this promotes a measured pace of reading most conducive to learning.

Acknowledgements

We would like to thank DeepMind, and in particular Demis Hassabis, for the wonderful opportunity to study the games of AlphaZero, and for his personal involvement in making this project a success. We would like to thank Dave Silver, Lead Researcher on AlphaZero, as well as Thore Graepel, Matthew Lai, Thomas Hubert, Julian Schrittwieser and Dharshan Kumaran for their extensive technical explanations and their assistance in running test games and test positions on AlphaZero. Nenad Tomasev deserves a special mention for reviewing the chess content and giving us plenty of great feedback!

A big debt of gratitude is owed to Lorrayne Bennett, Sylvia Christie, Jon Fildes, Claire McCoy, Sarah-Jane Allen and Alice Talbert for all their amazing work in keeping this project running and helping us with all the things we needed (and the things we didn't know we needed!). We'd also like to thank everybody at DeepMind for making us feel so welcome during our visits to the London office.

Thanks are also due to Allard Hoogland and the team at New in Chess who have published this book. They have supported our unique project and have ensured that the book is beautifully presented.

We would like to thank our families for their enthusiasm and support and, in the case of Matthew Selby, also for his technical expertise in extracting whatever we wanted from our data files.

All of these amazing people contributed to what has been a madly enjoyable and memorable project.

Introduction

On 5th December 2017, London-based artificial intelligence company DeepMind published 'Mastering Chess and shogi by Self-Play with a General Reinforcement Learning Algorithm'. The paper described the company's self-learning AI AlphaZero, which, within 24 hours of starting from random play and with no domain knowledge except the game rules, achieved a superhuman level of play in the games of chess and shogi (Japanese chess) as well as Go. It convincingly defeated a world-champion program in each case. In the case of Chess, that was Stockfish[1].

This was the first time a chess computer had reached superhuman strength from being entirely self-taught. It is momentous for chess players because, for the first time, we can learn from a powerful intelligence which built its chess strategy independently of our own rich history of chess development. It is also far-reaching for AI developers, with AlphaZero achieving superhuman strength in a matter of hours without the team needing to provide any domain-specific knowledge. This opens up the possibility of using these AI techniques for applications where human domain-specific knowledge is limited.

In an interview later in this book, Demis Hassabis describes how the success of AlphaZero builds on DeepMind's earlier work creating AlphaGo, a neural network based system that applied deep learning to successfully defeat Go legend Lee Sedol in 2016, and how both are milestones in the company's mission to use AI for the benefit of mankind. DeepMind plans to positively transform the world through AI. Among other things, it seeks to:

- help address the problems of climate change and energy;
- enable medical advances in diagnostics to make excellent medical care more widely available;
- accelerate scientific research to arrive more quickly at solutions crucial to human well-being.

The importance of the AlphaZero story has impact far beyond DeepMind's own work. Seeing the results of machine learning in the fields of chess and Go, developers around the world have been motivated to invest in similar techniques in other fields. Already, others have adopted the techniques that created DeepMind's AlphaGo to produce publicly available

1 The 2016 Top Chess Engine Championship (TCEC) season 9 world champion.

professional-strength Go playing machines, in what many consider to be a tipping point for public participation in the advancement of AI. In recent months the open-source Leela Chess Zero was developed based on the AlphaZero paper, and is now a dangerous challenger to the traditional 'Big Three' engines: Stockfish, Houdini and Komodo. Of course, it's of little surprise to us chess players (who have always known that there is something uniquely important about our game) that chess should play such a central role in the development of this critical technology!

This new approach to machine self-learning in chess has given us a strong chess player with a new style and approach, and that is the crux of this book. AlphaZero has independently developed strategies that possess many similarities to human wisdom, and many that are further developed or show situations where our well-established positional 'rules' are 'broken'.

In 2018, AlphaZero cannot yet explain to us directly what it has learnt (although Demis is confident that a number of technologies and tools that DeepMind and other groups are developing will make this possible in the future). Instead, top grandmaster Matthew Sadler guides us through the main differentiating factors in AlphaZero's game, compared with the top human praxis; and through detailed explanations based on illustrative games from AlphaZero's match with Stockfish, also shows us how AlphaZero's ideas can be incorporated into our own games.

This book explores the following chess themes:

- **Outposts (Chapter 7)**: we examine the variety of ways in which AlphaZero secures valuable posts for its pieces, from the knight and bishop all the way up to the king itself.
- **Activity (Chapter 8)**: AlphaZero is skilled in maximising the mobility of its own pieces and restricting its opponent's pieces. We pay particular attention to the ways that AlphaZero restricts the opposing king.
- **The march of the rook's pawn (Chapter 9)**: AlphaZero frequently advances its rook's pawn as part of its attack and plants it close to the opponent's king.
- **Colour complexes (Chapter 10)**: Matthew explains AlphaZero's fondness for positions with opposite-coloured bishops.
- **Sacrifices for time, space and damage (Chapter 11)**: AlphaZero makes many brilliant sacrifices for long-term positional advantage.
- **Opposite-side castling (Chapter 12)**: we consider some stunning examples in which castling queenside was the prelude to a dangerous AlphaZero attack.
- **Defence (Chapter 13)**: we learn about the contrasting defensive techniques of AlphaZero and Stockfish.

In addition, we have looked at the ways in which the thinking process of AlphaZero differs from that of chess engines such as Stockfish, and the resulting effects on its play. This will be invaluable to anyone who regularly uses engine assessments in their chess studies. We explore AlphaZero's use of a probabilistic assessment to guide its choices (which we believe gives it the ability to head for generally promising positions, leading to a style of play that feels intuitive to humans). The insights we have gathered have also revealed to us some features of engine analysis that we were not fully aware of before (e.g. the prevalence of 0.00 evaluations when analysing with Stockfish and other engines), and this knowledge should better equip chess engine users to understand their assessments.

In the process of writing this book, we had access to previously unpublished games[2] and evaluations from AlphaZero. We believe that there is a large amount of new and instructive material in this book that we hope you will thoroughly enjoy reading and trying out in your games.

Matthew Sadler and
Natasha Regan,
London, November 2018

---

2 A description of the AlphaZero games we received and the technical settings used for matches is given in the Technical note (Chapter 18).

CHAPTER 4

How AlphaZero thinks

AlphaZero's self-learning design is different to handcrafted chess engines such as Stockfish. In this chapter, we take a quick tour of the mechanics of AlphaZero's thinking, as it trains and as it plays. This chapter uses information from a DeepMind scientific publication in the journal *Science* released in December 2018: 'A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play'[6]. **David Silver**, Lead Researcher on AlphaZero, explained the inner workings of AlphaZero and research scientist **Thore Graepel** and research engineer **Matthew Lai** were on hand to answer our questions.

Different communities will benefit from understanding AlphaZero's thought process, including:

Professional chess players: Professional chess players now use engines for all of their pre- and post-game analysis, frequently switching engine according to the type of position they wish to analyse. By better understanding the skill sets of the various engines, professionals can make better use of them. Understanding AlphaZero's thought processes can provide fresh insight into the strengths and weaknesses of traditional engines and help professional players to optimise their use of them.

The AlphaZero team (l-r): Tim Lillicrap, David Silver, Thomas Hubert (back), Matthew Lai (front), Thore Graepel, Demis Hassabis, Julian Schrittwieser, Ioannis Antonoglou.

Not pictured: Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran and Karen Simonyan.

6 By David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan and Demis Hassabis. *Science*, 7 Dec 2018, Vol. 362, Issue 6419, pp. 1140-1144, DOI: 10.1126/science.aar6404.

Amateur chess players:
AlphaZero's thought processes are more human-like than traditional chess engines and we can pick up tips from how it makes its decisions.

Chess program developers:
Understanding AlphaZero's thought processes gives pointers for making traditional engines stronger. In addition, developers are increasingly experimenting with using AI.

AI developers:
The architecture of AlphaZero is general and its combination of computational reasoning and intuition extends to many important problem domains beyond game play.

In this chapter we illustrate the AlphaZero thought process by taking a look under the bonnet at its analysis at a critical moment in the game – how deeply and widely it searches, what moves it considers, and how it evaluates the resulting positions.

In the next chapter, 'AlphaZero's style – meeting in the middle', we relate the design of AlphaZero to its play and advance several hypotheses about AlphaZero's playing style and evaluations. We then attempt to validate these hypotheses by observing AlphaZero's games against Stockfish.

How AlphaZero works – the theory

AlphaZero's architecture is informed by the four principles that govern DeepMind's approach to artificial intelligence as a whole:

1. Learning rather than being programmed
The algorithm learns its strategy from examples rather than drawing on pre-specified human expert knowledge.

2. General rather than specific
The algorithm applies general principles and hence can be applied to multiple domains, e.g. shogi, Go, and chess.

3. Grounded rather than logic-based
Learning is based on concrete observations rather than preconceived logical rules.

4. Active rather than passive
The machine explores the game rather than being instructed by a human.

By satisfying all four requirements, AlphaZero deviates considerably from traditional computer game-playing systems.

Thore Graepel described AlphaZero's architecture as follows:

'With approximately 10^47 different chess positions, it would be too computationally expensive to exhaustively search through every available move, and every possible sequence of moves that might follow in the game. Therefore, most chess engines – including AlphaZero – combine a search algorithm with an evaluation function that provides an estimate of how good a position is at any point in the game.

Traditional chess engines use variants of what is called alpha-beta tree search, enhanced by dozens of game-specific search heuristics, and combine this with an evaluation function designed by expert chess players. In contrast, AlphaZero instead learns entirely on its own, developing its own evaluation function and using **Monte Carlo tree search** (MCTS) – a powerful alternative to alpha-beta tree search, that has the added advantage of being able to take into account prior knowledge about which moves are promising and which ones are not. This allows the search to focus mostly on promising and relevant variations. Furthermore, MCTS is robust with respect to inaccuracies of the evaluation function, which it averages across many different positions.

Where, then, does the prior knowledge come from? This is where AlphaZero's **neural network** – a computer system loosely modelled on the connections and neurons in the brain – comes in. The neural network takes the current game position as its input, and returns **move probabilities** for each possible move to be the strongest move (this is sometimes called the 'policy network'), along with a **value estimate** for the current position (sometimes called the 'value network'). This output guides the Monte Carlo tree search towards the most promising segments of the game tree. By reducing the number of moves considered in each position, the move probabilities cut down the **breadth** of the search. Being able to estimate the value of non-terminal positions in this way reduces the **depth** of the necessary search in the tree, because the value of the outcome of a given variation can be determined even before the end of the game is reached.

Crucially, the same algorithm is able to reach superhuman ability across several games without adapting the architecture for each specific game. In other words, the system displays a degree of generality: the same process of Monte Carlo tree search guided by a neural network, trained with self-play reinforcement learning, proves effective across several domains without the need for game-specific settings or modifications.'

Now, we zoom further into AlphaZero's training for its clash with Stockfish using some simple practical examples to illustrate the process.

How AlphaZero trains

Grandmasters might train for an upcoming match by spending many hours and days researching the latest openings and chess developments, and adopt a strict diet and exercise regime. They will also prepare specifically for the opponent they are expecting to face. This specific preparation involves collecting all of the prospective opponent's games using a huge database of tournament games from all over the world, looking for weaknesses in the opponent's play, and particularly in their openings set-ups.
In the last 20 years or so, top grandmasters have found they have to remember much more than previously, as they feel the need to adopt a wide range of openings to avoid the preparation of their opponents.
By contrast, AlphaZero's training before the Stockfish match took nine hours. It began training from a clean slate with no chess knowledge other than the rules of the game, and it didn't look at Stockfish's play at all. AlphaZero also did not use any available chess openings knowledge, and instead worked out its own openings as it trained and played against itself.
At the start of these crucial nine hours, AlphaZero did play chess, but not as we know it. As anyone who has taught chess to a small child will know, random play will get you nowhere when playing chess.
To avoid endless random games, those early games were stopped after a certain number of moves and called draws. Every now and then, though, some random games would end as wins for one side, and this rare signal allowed AlphaZero to learn the evaluation function output by its value network (how good is the current position?), and its policy function (how good is each move expected to be?). Intuitively, the system adjusts the parameters of the neural network such that it makes the moves played by the winning side more likely in their move probabilities, and it evaluates the positions encountered as more favourable to the winning side.
During those nine hours, AlphaZero played a total of 44 million games against itself – more than 1,000 games per second. At the same time, it continuously adjusted the parameters of its neural network so as to capture moves and outcomes from the most recent batch of games played against itself. For each move played during self-play, the MCTS performed 800 'simulations', each of which extends the current search by one move while assessing the value of the resulting position. As an example, AlphaZero could begin to analyse the chess starting position like this:

Step in thought process	Node	Evaluation of that node
Simulation 1	1.d4	56.8% (White)
Simulation 2	1.d4 ♘f6	43.1% (Black)
Simulation 3	1.♘f3	56.8% (White)
Simulation 4	1.d4 ♘f6 2.c4	56.7% (White)
Simulation 5	1.d4 d5	43.1% (Black)
Simulation 6	1.d4 ♘f6 2.♘f3	56.6% (White)
Simulation 7	1.d4 ♘f6 2.c4 e6	42.8% (Black)
Simulation 8	1.e4 e5	44.9% (Black)

We've shown eight simulations in the table above. By the time AlphaZero has completed the 800 simulations used for each position encountered during its lightning-fast training games, it would have looked a few moves deep for the most plausible lines. However, the search during training is much shallower than during a tournament game.

Each time AlphaZero selects a variation to consider, it will be on the basis of three criteria:

1. how plausible the move is in this type of position (as determined by the policy network);
2. how promising is the outcome of the variation (as determined by the value network); and
3. how often this variation has been considered in the search.

If a given variation has not been considered many times before, if the move appears plausible and if the variation looks promising, then AlphaZero will tend to select the variation and its continuation for simulation. The evaluation of the initial position will then be the average of all position evaluations from each of the 800 simulations. It should be noted that this is quite different from how current chess engines assess positions. Rather than returning an average of all lines considered, an engine such as Stockfish bases its assessment on the so-called principal variation, i.e. the very best line for both sides according to the current search tree. We believe that this is one of the reasons why AlphaZero plays in a style that is very different to traditional chess engines, often taking a more intuitive approach. We explore this further in the next chapter.
AlphaZero also differs from traditional chess engines such as Stockfish in its evaluation function, which may account for further differences in style. Stockfish's evaluation function is a combination of positional features. An example of how Stockfish evaluates a given position can be found in its evaluation guide. These figures were produced for a sample position using online resources at https://hxim.github.io/Stockfish-Evaluation-Guide/:

Group	White middle game	White endgame	Black middle game	Black endgame	Total middle game	Total endgame
Imbalance	-11	-11	0	0	-11	-11
Initiative	0	28	0	0	0	28
King	-49	2	-128	-16	79	18
Material	8599	9187	8651	9493	-52	-306
Mobility	147	297	151	280	-4	17
Passed pawns	0	0	0	0	0	0
Pawns	20	-36	48	0	-28	-36
Pieces	-61	-90	-87	-105	26	15
Space	40	0	55	0	-15	0
Threats	114	118	117	126	-3	-8
Total	8799	9495	8807	9778	-8	-283

In the example above, Stockfish assesses the position using an evaluation function that is a linear combination of features, with two sets of weights, one for the middlegame and one for the endgame. Stockfish assesses a number of factors for both White and Black (e.g. material and mobility are showing as contributors for the above position) and the total evaluation is the weighted sum of the various components. Note that beneath these groups of factors there is a more detailed list of individual factors with Stockfish taking hundreds of positional factors into consideration.

AlphaZero also has an evaluation function. Unlike the traditional chess engines of the last 50 years, which use handcrafted functions designed by grandmasters and represented as a linear combination of positional features, AlphaZero's evaluation function is learnt and represented in terms of a neural network called the 'value network', trained to predict game outcomes based on a raw representation of chess positions.

As a result, AlphaZero is unconstrained by human design or lack of imagination and has complete flexibility in choosing the features it takes into account when evaluating a given position.

But how AlphaZero's value network works remains a bit of a mystery and cannot be explained in simple rules such as a knight being worth approximately three pawns. It is likely that AlphaZero's value network views and assesses positions in a more fluid, situation-dependent way.

So, rather than being constrained to, say, an evaluation function that adds separate assessments of material and mobility, AlphaZero can consider the interaction of different factors, for example how mobility affects the value of the material. This flexibility can be very useful in understanding the overall position, and could explain how AlphaZero implements combinations of positional motifs so effectively (see Chapter 10 on 'Colour complexes'). Whilst we cannot understand exactly how AlphaZero is thinking, we can explore the ways in which AlphaZero generates its innovative and active plans, and how it conducts its ferocious attacks through analysing its games.

We asked **David Silver** to explain a little deeper how the training process works and progresses:

How does AlphaZero's neural network give a value for a position?
AlphaZero sees the chessboard as an 8x8 grid of numerical values. These values are processed by a series of computational steps known as layers of the neural network. Each layer takes the previous 8x8 board representation and constructs a new 8x8 board that can represent richer features. This process is repeated over many layers to produce ever more powerful representations of the board. The nature of each layer is determined by millions of tunable weights, which means that the system can learn for itself what features to represent. Finally, AlphaZero combines all of these features together, using even more tunable weights, to determine the final evaluation of the position.

So Stockfish might say, you've got an open file and doubled pawns and opposite-coloured bishops, and you add those values all up to get a score. Does AlphaZero create its own function and does it have the same positional features in mind?
The key difference is that AlphaZero learns its own features by tuning the connections of its neural network. So while AlphaZero could in principle learn a feature such as 'open file with doubled pawns' it could equally see the position in a totally different way, perhaps learning complex features that are useful to the machine but hard for a human to interpret.

How does AlphaZero improve during training?
When it wins (or loses) a game, the connections in the value network are updated to evaluate each position in that game more positively (or negatively). At the same time, the connections in the policy network are strengthened so as to play more often the move recommended by AlphaZero itself, after a lot of thinking by its Monte Carlo tree search. AlphaZero plays against itself millions of times,

learning to provide better move suggestions (using the policy network) and to judge positions more reliably (using the value network) – essentially developing something akin to 'intuition' for how to play the game. This process of learning for itself, solely from its interactions, is known as 'reinforcement learning'.

If you left AlphaZero training against itself for a very long time, would it just keep getting better and better?
When we trained AlphaZero on Go, we saw its performance continue to improve over a very long training time. However, training AlphaZero on chess appears to have diminishing returns, perhaps due to the large number of draws that start occurring during self-play.

Matthew Lai explained the nature of these training games:

How does it work with those training games? Are they just very fast games?
Each training game is played very quickly, using about 40 milliseconds thinking time per move to execute a Monte Carlo tree search consisting of 800 simulations.

When AlphaZero was training against itself, did many of the games result in draws?
In the beginning, almost all games ended in draws by the 50 moves rule (no pawn moves and no pieces taken for 50 moves), because the play is almost entirely random. Towards the end of the training we observed similar draw rates to those other top chess engines find when they play against themselves – about 70-80%. This increases to >90% at tournament time controls.

Can you get a sense of how AlphaZero's play develops as it trains?
Periodically during training, we take snapshots and play through some games, using AlphaZero at each given stage in its training. We don't want to take the training games themselves because they are played at about 40 milliseconds per move, but we take a snapshot and play longer games to see how it is progressing. One interesting thing we found is that AlphaZero re-discovers opening sequences that are frequently played by human players as well. What we found most amazing is that, as training progresses, AlphaZero often discards those known variations because it finds ways to refute them!

It looks to us like AlphaZero uses piece mobility well and is a fantastic attacker. Do you think it looks at these concepts in a different way to Stockfish, perhaps more mathematically?
Those are well-known concepts in the computer chess literature, but in traditional chess engines they are usually applied with minimal or no selectivity. As a consequence, they

have to be given low weights so that they do not exert an overly strong influence when their application is not justified. In the case of AlphaZero, the highly non-linear nature of neural networks means it can potentially learn to apply them much more selectively, and with higher influence where it thinks the features are valid. In addition, since AlphaZero maximises expected score, it is not so tied to keeping the material balance.

Is the speed of training games the sort of thing you might change if you were changing the training process? You might give it longer?
Yeah. It's a trade-off between how good you want the moves to be and how many different games you want for training. And the more games the system can see, the better it can find rules to generalise across them and the less it will overfit to the particularities of any individual game.

It is important to comment on the considerable difference between the hardware used to train AlphaZero and the hardware used by AlphaZero in match play. During training, 5,000 first-generation TPUs were used to generate self-play games, and 16 second-generation TPUs were used to train the neural networks. These computing resources minimise the time taken to complete the training. By contrast, when playing Stockfish, AlphaZero used a single machine with 4 first-generation TPUs.

AlphaZero's match play

Some game-playing computers simulate outcomes as they play (for example Jellyfish, which plays Backgammon). AlphaZero simulates millions of games whilst training, but does not use the technique of simulating to the end of the game ('random rollouts') during play. Once AlphaZero's training is complete, the latest neural nets for policy (how to choose moves) and evaluation (how to assess the position) are taken for use in match play. As Matthew Lai explained to us, DeepMind's earlier versions of AlphaGo used to conduct random rollouts during play. This is not necessary for AlphaZero because its value network is already so advanced that additional rollouts during play do not add any value. As a consequence, though, there is no randomness built into AlphaZero as it plays.

We asked **Matthew Lai** about whether AlphaZero would play the same game twice:

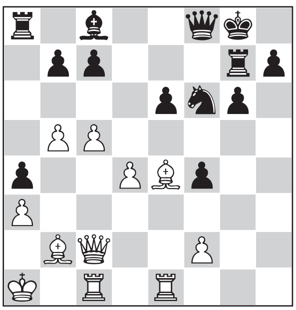
When it's playing, does AlphaZero have any randomness in its play?
When AlphaZero is playing against itself during training, it is very important that we see a wide variety of positions and moves. This is achieved by explicitly adding randomness to its move selection.

After training, when AlphaZero is playing matches, there is still some randomness due to the parallel nature of the hardware used; also we sometimes add randomness to the opening to ensure diverse evaluation.

The AlphaZero thought process – an in-depth illustration of one move

We can illustrate AlphaZero's thought process in a match situation (in other words, after it has completed its training) by looking in detail at one particular position from the match with Stockfish. Let us follow AlphaZero's steps as it thinks and decides about what move to make next. The position we have chosen comes from the game '*Exactly how to attack*': a fabulous decisive game which we will come back to several times in this book. The position we have chosen occurs after Black's 29th move:

AlphaZero
Stockfish 8
London 2018



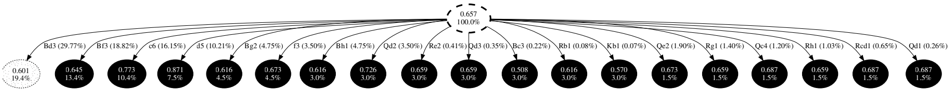
In the opening, AlphaZero had sacrificed two pawns on the kingside (the g- and h-pawns) after gaining the bishop pair. I had expected AlphaZero to line up all its pieces on one of those files, but instead AlphaZero dedicated its efforts to forcing open the centre with the goal of opening diagonals for its bishops to support its kingside offensive. We join the game at the critical moment.

In the coming pages, we will present snapshots of AlphaZero's thinking at various points in its thought process, starting at the beginning – when it has searched very few branches of its tree of variations – to the end of its thought process when it has decided on a move and refined its evaluation of the position.

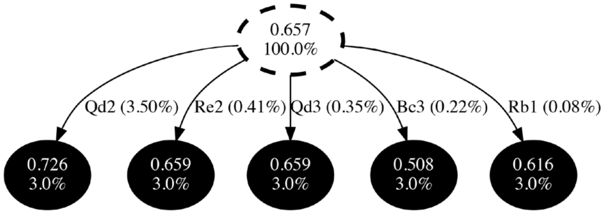
To help us, Matthew Lai has provided us with trees of the moves that AlphaZero considered, together with supplementary information such as the evaluation of the move. In the above diagram AlphaZero is considering its 30th move as White. We will now present the first tree, after just 64 nodes of search, and walk you through the moves and the information displayed:

[see next page]

First tree – 64 nodes



That looks scary doesn't it? That was exactly my thought when I saw it too, but some explanations from Matthew Lai and from my co-author Natasha helped enormously. Let's zoom into a small part of the tree:



We are showing here the root of the tree (the dotted line) and five of the 19 possibilities displayed in our tree.

The root node

1. AlphaZero's evaluation of the position
The top number (0.657) is AlphaZero's evaluation of the position from White's point of view. 0.657 means a 65.7% expected score, i.e. better for White. This expected score is made up of a combination of wins, draws and losses (though we can't tell the exact distribution). For example, 65.7% wins, no draws and 34.3% losses would give a 65.7% expected score, as would 31.4% wins, 68.6% draws and no losses.

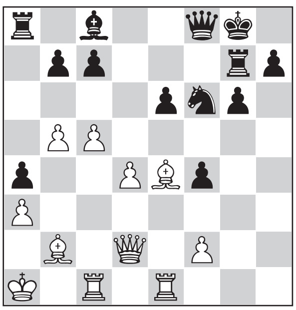
2. The total percentage of node searches spent
100% means that 100% of AlphaZero's node searches were used to produce this result. We will always see 100% for the position at the top of the tree (the root node).
As AlphaZero searches deeper down the tree into the branches, it divides up the available time and energy between moves and variations, spending the bulk of its time on the moves it considers most important.
When we move on to looking at the branches, we will see percentages less than 100% and this gives the proportion of the time that AlphaZero has allocated to this possibility.

Moves, and move probability '♕d2 (3.50%)'

The next step we see shows five moves, with a percentage value next to each of them.
The percentage represents the prior move probability, and as we will see throughout our trees, this number will never change for a given position.
The number indicates how likely AlphaZero believes it is at the beginning of its thought process that it would eventually choose this move. It is like showing a position to a grandmaster and asking them what moves are plausible at a first glance and which might be best. The grandmaster might come up with four or five moves and order them from most likely to least likely. This is just what AlphaZero is doing to prioritise its analysis.
In chess terms:

AlphaZero
Stockfish 8
London 2018

30.♕d2



AlphaZero thinks that ♕d2 is 3.50% likely to be its choice in this position. Not very likely therefore, but not impossible. Compare that to the incomprehensible and illogical 30.♗c3, which AlphaZero only gives a 0.22% chance of being selected (thankfully!).

Evaluation of the branch and resources spent on that move

AlphaZero's **evaluation** of the position after this move (30.♕d2) is 0.726 (72.6% expected score), which is pretty good, and it spent 3.0% of its total node searches on that move. In human terms, AlphaZero had a quick look at 30.♕d2, and its first impression was positive.

To recapitulate, working from top to bottom, we can see:
1. AlphaZero's overall evaluation of the position;
2. the moves it has looked at, and how likely AlphaZero thinks it is to choose each one (its first impression);
3. AlphaZero's evaluation of the position after each move;
4. how much time (as a percentage of the level above) AlphaZero spent considering the move.

The alert among you will have noticed a couple of interesting points:

1. The percentages on each level of the tree don't add up to 100%.

This is because we are not showing all the moves that AlphaZero considered (to make it easier to read).

2. The overall evaluation of the position (0.657 = 65.7% expected score) does not match the evaluation of any individual move.

An engine such as Stockfish works on the basis that the evaluation of the best move determines the evaluation of the position. So if Stockfish evaluates its best line as +0.38 pawns (remember, Stockfish evaluates in pawns, not in percentage expected score) then that is also Stockfish's evaluation of the position.

AlphaZero takes a more probabilistic view. AlphaZero essentially evaluates the position as a whole by taking into account the evaluations of all the moves it looks at, giving more weight to the moves it considers more deeply.

In the next chapter ('AlphaZero's style – meeting in the middle'), we will discuss the advantages and disadvantages of this approach and give some practical examples. Just in general however, such an approach might end up mimicking human intuition where players steer for a position because 'it feels good' and then work out a concrete line when the position arises.

The first level of the tree above is very broad, with AlphaZero's top 19 choices of first move shown. We have put the results into a table to help readability:

[see next page]

Move	Move probability	Actual % of node searches spent on the move	Evaluation (% expected score)	Rank move probability	Rank node searches	Rank eval
30.♗d3	29.77	19.4	0.601	1	1	17
30.♗f3	18.82	13.4	0.645	2	2	13
30.c6	16.15	10.4	0.773	3	3	2
30.d5	10.21	7.5	0.871	4	4	1
30.♗g2	4.75	4.5	0.616	5	5	14
30.f3	3.5	4.5	0.673	7	5	7
30.♗h1	4.75	3	0.616	5	7	14
30.♕d2	3.5	3	0.726	7	7	3
30.♖e2	0.41	3	0.659	14	7	9
30.♕d3	0.35	3	0.659	15	7	9
30.♗c3	0.22	3	0.508	17	7	19
30.♖b1	0.08	3	0.616	18	7	14
30.♔b1	0.07	3	0.57	19	7	18
30.♕e2	1.9	1.5	0.673	9	14	7
30.♖g1	1.4	1.5	0.659	10	14	9
30.♕c4	1.2	1.5	0.687	11	14	4
30.♖h1	1.03	1.5	0.659	12	14	9
30.♖cd1	0.65	1.5	0.687	13	14	4
30.♕d1	0.26	1.5	0.687	16	14	4
Overall position assessment			**0.657**			

At this very early stage of thinking, AlphaZero's search is quite broad, and AlphaZero has spent some time searching moves which it thinks are quite unlikely, even when their evaluations aren't particularly special either.

From prior expectations, AlphaZero starts off thinking 30.♗d3 is the most likely move, but its intuition is not borne out by the evaluation of the move. Its fourth most likely move – 30.d5 – storms to the top of the evaluation table!

Section B – Reducing the mobility of the opponent's forces to create opportunities

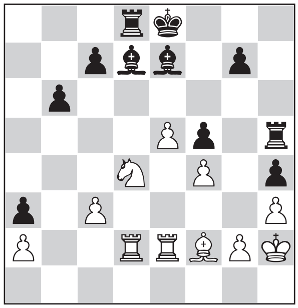
In this section AlphaZero reduces the opponent's forces to passivity using sacrificial and other techniques.

I. Squeezing with the pawns

Game: 'Endgame class'
AlphaZero unexpectedly offers up an advanced rook's pawn on the queenside – which looked to be a useful advantage for the endgame – to reduce the activity of White's pieces. The knight decentralises to win the pawn, the dark-squared bishop is restricted by pawn phalanxes on b6/c5 and h4/g3, and most importantly the king is rendered passive. AlphaZero converts its edge by keeping the white king pinned to the corner and trading off Stockfish's active pieces. The result for Stockfish is a useless knight on f6, a rook tied to the back rank and an unstoppable passed c-pawn to contend with.

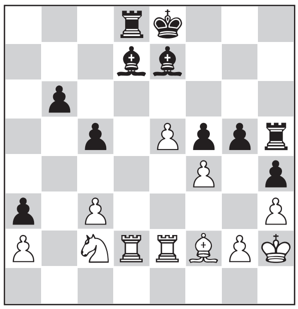
Game themes:
1. Sacrificing material to reduce the opponent's activity [31…c5]
2. Reducing the opponent's activity with pawn advances [31…c5, 32…g5, 33…g4, 34…g3]
3. Opponent's passive pieces [♔g1, ♘f6]
4. Exchanging off the opponent's active pieces to leave passive ones [51…♖xa1, 57…♗c5]
5. Lonely knight [♘f6]

Stockfish 8
AlphaZero
London 2018

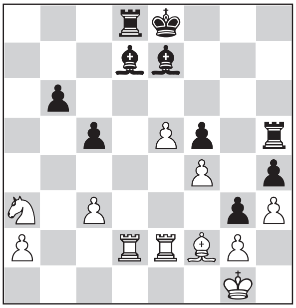


Already having some experience of AlphaZero's play, I was expecting AlphaZero to exchange off its dark-squared bishop for the knight on d4 and play the rook and opposite-coloured bishops middlegame, hoping to invade the weakened central light squares and play on the weakness of the fixed a2-pawn. That probably wasn't a bad plan, but AlphaZero's creative strategy took me completely by surprise.

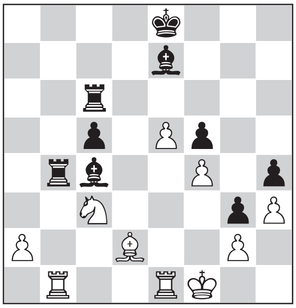
31…c5 32.♘c2 g5



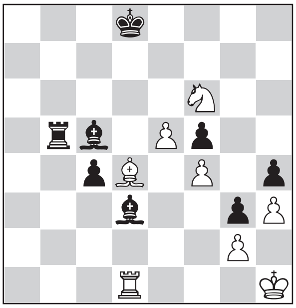
Giving away the a3-pawn, which to my eyes was one of Black's strongest assets!
However, AlphaZero is trading this asset for a series of other dynamic plusses.
33.♘xa3 g4 34.♔g1 g3



With this manoeuvre, Black has gained space on the kingside and severely restricted the freedom of the white king and dark-squared bishop.
By sacrificing the a3-pawn, Black has gained a new potential channel for entry into the White position: the a-file.
35.♗e3 ♖a8 36.♘c4 ♖h6 37.♖b2 ♖a6
White is by no means helpless as Black's position is extended and AlphaZero has a lot of squares to protect with limited forces. However, AlphaZero just seems to have it all under control and will first absorb White's current temporary activity before proceeding to push White back and reclaim all that White has gained.
38.♗c1 b5 39.♘e3 ♖a4 40.c4
40.♘d5 looked natural to me. Playing around with the engines in this position fails to deliver anything resembling clear equality: 40...♗d8 41.♘f6+ ♗xf6 42.exf6+ ♔d8 43.f7 ♖f6 44.♖e5 ♖xf7 45.♖xc5 ♖e7 46.♖e5 ♖e4 is still problematic for White due to the weakness of the back rank: 47.♔f1 ♖7xe5 48.fxe5 ♗e6 and the white king will not escape: ...♗c4+ is coming.
40...bxc4 41.♘d5 c3 42.♘xc3 ♖c4 43.♗d2 ♖c6 44.♔f1 ♗e6 45.♖b1 ♖b4 46.♖ee1 ♗c4+



AlphaZero's pieces advance inexorably, increasing the difference in activity between its pieces and the opponent's pieces.
46...♗c4+ seals the white king in its box on the kingside: the bishop stops the king escaping via f1-e2 whilst the g3-pawn covers f2 and h2.
47.♔g1 ♖c8 48.♖bc1 ♗d3 49.♘d5 ♖b2 50.♗c3 ♖xa2 51.♖a1 ♖xa1
Typical AlphaZero play, exchanging off active pieces to leave the opponent with passive pieces: we will see another example on move 57.
52.♗xa1 c4 53.♘f6+ ♔d8 54.♗c3 ♖b8 55.♗d4 ♗b4 56.♖d1 ♖b5 57.♔h1 ♗c5 0-1



White's lonely knight – established on an outpost on f6 but unable to influence the struggle to stop the c-pawn – and boxed-in king are testament to the grandeur of AlphaZero's strategy. The c-pawn will not be stopped! 57...♗c5 58.♗xc5 ♖xc5 59.♔g1 ♔e7 60.♔h1 c3 wins.

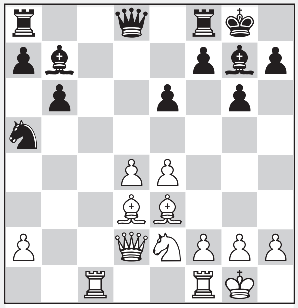
HISTORICAL PARALLEL

Magnus Carlsen's Grünfeld play

This encounter between the current World Champion Magnus Carlsen when he was just 17 and the Ukrainian genius Vasily Ivanchuk has themes of AlphaZero's games in the Grünfeld Defence, and the games in which AlphaZero built up play on the kingside while Stockfish's queenside counterplay never got going. The switchback 23.♕c1, through which the decisive invasion happens on the flank where Black should be strongest, reminds me of AlphaZero.

Magnus Carlsen 2690
Vasily Ivanchuk 2750
Morelia/Linares 2007 (11)

1.d4 ♘f6 2.c4 g6 3.♘c3 d5 4.cxd5 ♘xd5 5.e4 ♘xc3 6.bxc3 ♗g7 7.♗c4 c5 8.♘e2 ♘c6 9.♗e3 0-0 10.0-0 ♘a5 11.♗d3 b6 12.♖c1 cxd4 13.cxd4 e6 14.♕d2 ♗b7



A typical Grünfeld structure has arisen in which White's strong centre is counter-balanced by Black's queenside pawn majority. Black's most difficult problem is to find a good spot for his knight, which tends to hang around a bit in the early middlegame (as here on a5). There is another less obvious challenge to Black's position: his kingside has been weakened by the exchange of his king's knight on move five.

15.h4

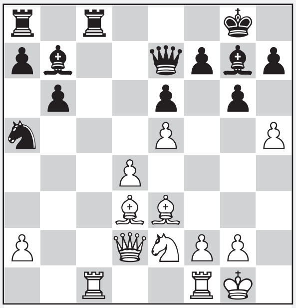
The young Carlsen plays a move that AlphaZero likes too! White exerts pressure on the black kingside with the h-pawn.

15...♕e7

The pawn is poisoned: 15...♕xh4 16.♗g5 ♕h5 17.♘g3 ♕g4 18.♗e2. 15...♕d7 is the standard move in this position. The queen assists the exchange of rooks on the c-file but loses sight of the kingside dark squares a little. A high-class game continued: 16.h5 ♖fc8 17.♖fd1 ♖xc1 18.♖xc1 ♖c8 19.♖xc8+ ♕xc8 20.♗h6

♗xh6 21.♕xh6 ♗a6 22.♕d2 ♗xd3 23.♕xd3 ♔g7 24.♕d2 f6 25.e5 ♘c6 26.exf6+ ♔xf6 27.♕h6 with a clear advantage in Kir.Georgiev-Bukavshin, Aix-les-Bains 2011.

16.h5 ♖fc8 17.e5



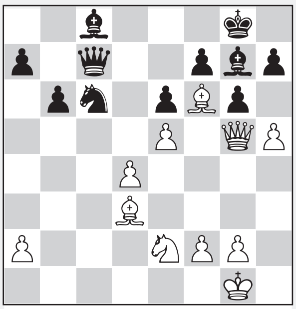
We saw this idea in AlphaZero's Grünfeld games as well: the pawn on e5 reduces the activity of Black's dark-squared bishop on g7 and fixes Black's dark-squared weaknesses on that wing too by creating an outpost on f6 for a white bishop or knight.

17...♖xc1 18.♖xc1 ♖c8 19.♖xc8+

AlphaZero chooses the same approach in this position!

Exchanging the rooks prevents the knight on a5 from activating itself (a move like 19.♖d1 would have been met by 19...♘c4) and allows White to maintain the attacking tempo.

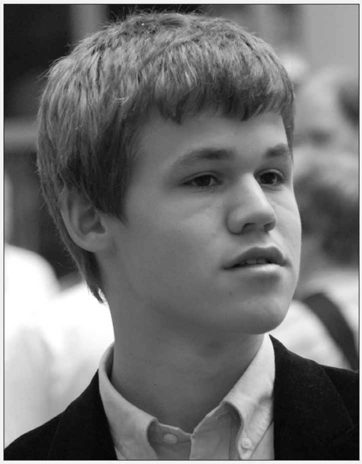
19...♗xc8 20.♗g5 ♕c7 21.♗f6 ♘c6 22.♕g5



A powerful move, threatening 23.♗xg7 and 24.h6+, making AlphaZero-style use of the advanced h-pawn to target the dark squares around the black king.

22...h6 23.♕c1

A fantastic switchback! Black must defend his kingside (which he was forced to weaken with 22...



Magnus Carlsen: an AlphaZero-like switchback at 17.

h6) but this gives White a tactical opportunity to exploit the pin on the black knight on the open c-file. The file intended for black counterplay has become White's decisive channel of entry, as so often happens when the mobility of one side is much greater than the other.

23...g5 24.♗b5 ♗d7 25.d5 exd5 26.♘d4

Winning a piece.

26...♗xf6 27.exf6 ♕d6 28.♗xc6 ♕xf6 29.♗xd7 ♕xd4 30.g3 ♕c5 31.♕xc5 bxc5 32.♗c6 d4 33.♗b5 ♔f8 34.f4 gxf4 35.gxf4 1-0

II. Restricting queen mobility

As we discuss in the chapter on 'Defence', Stockfish is skilled at using its queen to slow down the build-up of an opponent's attack by harrying the opponent's pieces and sowing confusion. The game ***'Using a queenside file to defend the kingside'*** is a fine example of this.

It was a striking theme from the games released in December 2017 that AlphaZero could render Stockfish's queen completely passive.

Let's see one of those games:

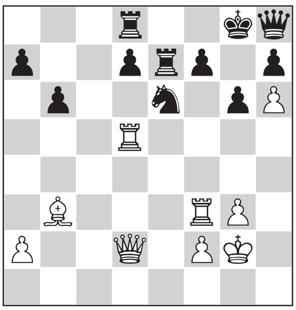
Game: 'Risky rooks'

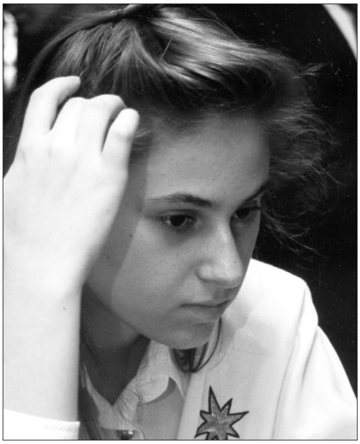
A remarkable occurrence: in the middlegame the black queen is boxed into the corner with just one legal move.

Game themes:

1. Sacrificing material to reduce the opponent's activity [47.♖xc5]
2. Opponent's passive pieces [♕h8]

AlphaZero
Stockfish 8
London 2017

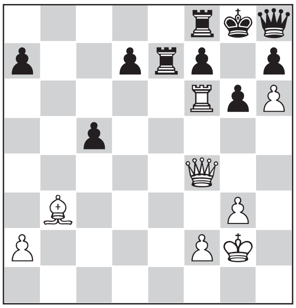




Judit Polgar: a stunning finish to box in the black king.

Black has been under pressure since White sacrificed a pawn in the opening. The strange move 45...♕h8 is the choice of my engines when analysing for six hours or more. AlphaZero finds a magical way to make the queen wish she had not retreated to the corner.

46.♕b4 ♘c5 47.♖xc5 bxc5 48.♕h4 ♖de8 49.♖f6 ♖f8 50.♕f4



The black queen is completely imprisoned. Black can only sit and await its fate.

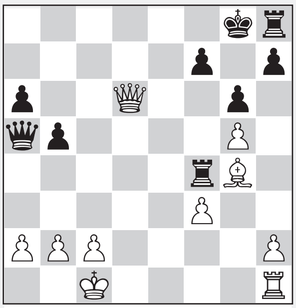
50...a5 51.g4 d5 52.♗xd5 ♖d7 53.♗c4 a4 54.g5 a3 55.♕f3 ♖c7 56.♕xa3 ♕xf6 57.gxf6 ♖fc8 58.♕d3 ♖f8 59.♕d6 ♖fc8 60.a4 1-0

HISTORICAL PARALLEL

Judit Polgar's ♕f6!

The stunning move 49.♖f6 reminded me of an idea played by Judit Polgar – one of the world's elite players until her retirement – against the French grandmaster Laurent Fressinet. The following position was reached after 24 exciting moves in a Najdorf Sicilian:

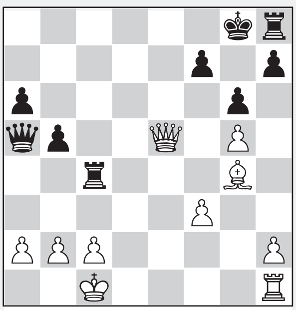
Judit Polgar 2656
Laurent Fressinet 2536
Istanbul ol 2000 (9)



24...♖c4

As Judit explains in volume two of her *Best Games* collection, she thought for 32 minutes after Black's 24th move, which shows the difficulty of the position.

25.♕b8+ ♔g7 26.♕e5+ ♔g8



27.♕f6

A fantastic move. The queen ties down the black king and the rook on h8 completely while preparing ♖h1-d1-d8+. The key difference with 27.♖d1 is seen in the game after 27...♕c7.

A) 27.♖d1, which Judit wanted to play, fails to 27...♕c7 28.♕f6 ♕f4+ 29.♔b1 ♕xf6 when the queen on f6 covers d8 and stops the intermediate check 30.♖d8+;

B) 27.♖e1 was Judit's next idea, but after much examination, she decided that 27...♖c5 would destroy the coordination of White's pieces and take control of the e5-square: the rook cannot be taken because of mate on e1. But then inspiration and tactical genius struck!

27...♕c7 28.♖e1 ♕c6

Why doesn't 28...♕f4+ work? Well in comparison to 27.♖d1, the rook is now on e1, which means that after 29.♔b1 ♕xf6 White can interpose 30.♖e8+, winning a piece after 30...♔g7 31.gxf6+ ♔xf6 32.♖xh8. Wonderful tactical ingenuity!
The defence in the game also didn't work.

29.♗e6 fxe6 30.♖d1 1-0

Black resigned as 30...♕e8 31.♖d8 ♕xd8 32.♕xd8+ ♔g7 33.♕f6+ ♔g8 34.♕xe6+ ♔g7 35.♕f6+ ♔g8 36.♕xa6 leads to a winning position for White, with queen and three enormous passed pawns on the queenside.
A stunning finish!