AlphaZero – What's Missing?

Ivan Bratko
University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, Ljubljana
E-mail: bratko@fri.uni-lj.si

Keywords: computer game playing, computer chess, machine learning, explainable AI

Received: March 8, 2018

In December 2017, the game playing program AlphaZero was reported to have learned in less than 24 hours to play each of the games of chess, Go and shogi better than any human, and better than any other existing specialised computer program for these games. This was achieved just by self-play, without access to any knowledge of these games other than the rules of the game. In this paper we consider some limitations to this spectacular success. The program was trained in well-defined and relatively small domains (admittedly with enormous combinatorial complexity) compared to many real world problems, and it was possible to generate large amounts of learning data through simulated games which is typically not possible in real life domains. When it comes to understanding the games played by AlphaZero, the program's inability to explain its games and the knowledge acquired in human-understandable terms is a serious limitation.

Povzetek: Decembra 2017 so poročali, da se je program AlphaZero v manj kot 24 urah naučil igrati šah, go in shogi bolje, kot katerikoli človek in katerikoli drug računalniški program specializiran za to igro. To je dosegel kar z igranjem s samim seboj, brez dostopa do kakršnegakoli znanja o teh igrah, razen samih pravil igre. Vsiljuje se vprašanje, ali obstajajo kakšne omejitve tega neverjetnega podviga. Program se je učil v dobro definiranih in razmeroma enostavnih domenah (čeprav je res, da imajo te igre ogromno kombinatorično zahtevnost) v primerjavi z mnogimi problemi realnega sveta. Za te igre je bilo mogoče s simulacijo generirati ogromne količine učnih podatkov, kar navadno ni možno v domenah iz realnega življenja. Osnovna pomanjkljivost programa AlphaZero je tudi njegova nezmožnost, da bi svoje odigrane partije razložil na človeku razumljiv način.

1 Introduction

In December 2017, an amazing achievement has been reported (Silver, Hubert et al. 2017). DeepMind's program AlphaZero was able to learn in less than 24 hours to play each of the games of chess, Go and shogi better than any human, and better than any other existing specialised computer program for these games.

This was a third event in the success story at DeepMind with game playing programs with the word Alpha in their names. It started with the famous program AlphaGo (Silver et al. 2016) which convincingly defeated one of the best human go players in a match of five games. That was the first time ever that a computer program was able to defeat a leading human player at Go. AlphaGo was specialised at Go, and learned from exemplary high quality games of Go previously played by strong human players. AlphaGo Zero (Silver, Schrittwieser et al. 2017) was able to learn to play Go even better. The impressive difference between AlphaGo and AlphaGo Zero was that the latter can learn from games just played by itself, thus without having access to examples of well-played games or any other source of game-specific knowledge of the game, except the bare rules of the game.

Finally, AlphaZero is a general game playing program not specialised to Go, so it can learn to play any game of this kind just by self-play. For example, to get to the strength level of the best human chess players, AlphaZero needed no more than one and a half hours of learning by self-play.

The basic architecture of AlphaZero is as follows. AlphaZero learns by reinforcement learning from simulated games against itself. It uses a deep neural network that learns to estimate the values of positions and the probabilities of playing possible moves in a position. To select a move to play in the current board position, AlphaZero performs Monte Carlo Tree Search (MCTS). This search consists of simulating random games from the current positions, in which the probabilities of random moves increase with the move probabilities returned by the neural network, and decrease with the moves' visit counts. The use of MCTS in chess is in contrast to search in other strong chess programs. They perform Alpha-Beta search which had been considered before AlphaZero much more appropriate for chess.

2 An interesting observation about AlphaZero training in chess

To appreciate this achievement, let us consider some illustrative quantitative facts about AlphaZero at chess. As reported by Silver, Hubert et al. (2017), in chess training AlphaZero played about 44 million games against itself in nine hours of self-play. This took 700

thousand "steps" of training. According to the plots of chess rating improvement in time of training (Silver, Hubert et al., 2017), AlphaZero attained the chess strength of best human players after about 110 thousand training steps. By that time, AlphaZero had played about 6.9 million games with itself.

Now let us consider some quantitative facts from the human history of chess. ChessBases's Mega Database is a comprehensive collection of chess games played in all history of human chess. Mega Database is very representative of about all important chess games ever played by humans, so it is well representative of all chess concepts and ideas ever found by human players. The 2018 version of Mega Database contains 7.1 million games which quite amazingly matches AlphaZero's estimated 6.9 million games needed to reach the best humans' chess strength. Of course it may be argued that this is a mere coincidence. And it can be rightfully observed that this comparison is rather crude: it is not true that the best human players derive their skill from *all* 7.1 million games. It is certainly not true that all the games in Mega Database are needed to subsume the present chess knowledge by mankind. Therefore Mega Database, viewed as a kind of codification of total human chess knowledge, contains a lot of redundancy. Nevertheless, the numbers do seem to offer a first "feasibility check" of AlphaZero's achievement.

3 Are there any limitations to AlphaZero approach?

Given that the games of chess, Go and shogi are so difficult for humans, and that AlphaZero made the same progress at chess, say, in 1.5 hours of self-play as the mankind did in over a hundred years, this looks impressive indeed. If the problem of such difficulty for humans can be mastered in one and a half hours by a machine using AI techniques, then an impression is that AI can now do everything.

But let us consider whether this impression is really so true in general. What are the limitations? Let us look at the problems dealt with by AlphaZero from a little broader perspective.

(1) These games are limited to the board worlds, which amounts to 64 squares for chess, and 381 squares for Go. True, these small worlds give rise to combinatorial complexities of astronomic proportions. For chess, an old estimate by Claude Shannon (1950) is: there are over 10^{40} possible chess positions, and over 10^{120} possible games. The magnitude of these numbers is popularly illustrated by their comparison to the number of all atoms in the observable universe, which is of the order 10^{80}. The number of possible games of chess is thus incomparably larger than the size of the universe. And, also true, both Go and shogi are in these terms much more complex than chess. On the other hand, the combinatorial complexity of these games is rather deceiving. Compared to many real world domains studied by biology, chemistry and physics, these games are small.

(2) The rules of these games are simple and known. Therefore almost unlimited experimentation with these games through simulated games is possible. This gives rise to the automatic generation of very large numbers of training instances from which AlphaZero could learn.

This is very different from many complex real-world domains in which learning data is collected through time consuming and expensive experiments, and therefore the amount of training data is much more limited. In contrast to machine learning from big data, in such domains the *scarcity* of data is often the problem. For example, Wiley et al. (2016) describe reinforcement learning by a tracked robot for which no sufficiently accurate simulation model was available. Therefore, experimentation had to be carried out with the actual physical robot, so the number of trials was severely limited due to time constraints and wear and tear of the robot. More elaborate methods of machine learning were needed to enable more effective use of available data. The situation with available data may be even more constrained, like in medicine where examples of patients with a disease under study can only be "generated by nature". For machine learning to be successful with "small data", different machine learning methods and algorithms are needed. In particular, it is desired that the learning method can make use of domain background knowledge. In this way lack of data can be compensated by prior knowledge. For example, the learning program may use the laws of physics that are already known prior to learning.

4 Does AlphaZero play chess "more like humans"?

There have been some speculations that AlphaZero is not only by far the strongest chess playing program, but that it also plays chess in a way that is more similar to the way strong human players play chess.

This conjecture is based on a particular comparison between AlphaZero and Stockfish, one of the strongest chess programs before AlphaZero. AlphaZero convincingly defeated Stockfish in a match of 100 games in which AlphaZero won 28 games and drew the rest. The particular point of comparison is the number of positions searched per second by the two programs. Stockfish searched 70 million of positions per second compared with 80,000 by AlphaZero. This was interpreted as indicative of a more human-like style by AlphaZero simply because in general computers base their strength on the brute force computational power which allows them to search deeper. By contrast, humans can only search typically of the order of a few tens of positions per move, or something like a few positions per second.

Therefore the humans, to compensate for this inferior search ability, have to rely on deeper chess knowledge and intuition. The argument then is that AlphaZero with about thousand times lower search speed than Stockfish must have better chess knowledge to still be able to win. This argument is not completely convincing. In terms of search speed, AlphaZero is still incomparably faster than humans. Another big difference between AlphaZero and

human style of play is in the search method used. AlphaZero uses Monte Carlo tree search technique which is based on random simulations of possible games from the current position, and counting favourable outcomes resulting from moves tried. This is certainly not the way that humans perform their search. On the other hand, moves played as the result of MCTS indeed seem to resemble human players' decisions more than moves played by typical chess engines. In particular, it appears that moves played by AlphaZero better reflect long-term positional judgement in chess that is attributed to strong human players' deep understanding of the game. We will return to this question in the next section when analysing a surprising positional sacrifice by AlphaZero in one of the games against Stockfish.

5 Examples of super play by AlphaZero

The world of chess was stunned by examples of play by AlphaZero from some of the published games between AlphaZero and Stockfish. Probably the most spectacular example comes from the following game in which AphaZero had White pieces. This example was discussed many times in numerous chess media, for example in (Guid 2018). After 18 moves, the position in Fig. 1 occurred in the game, with White to move. Here Black is threatening to capture White knight on h6 with the queen or the king. So it seems that White knight has to retreat to g4, which a reasonable human player would actually do. After that, if both sides played their best moves, White knight would eventually escape to safety, but Black would come out with a somewhat better position. However, in position in Fig. 1, AlphaZero played incredibly **19 Rf1-e1**, leaving his knight on h6 to be captured by Black. In the game, Stockfish indeed took the knight and appeared to be winning. AlphaZero did have some positional compensation for the knight, but that did not appear to be anything nearly enough for the material disadvantage. But AlphaZero's judgement turned out to be better in the long run. White managed to create threats virtually out of nothing, and 20 moves later managed to achieve a clear advantage. To appreciate the details of all this requires some chess knowledge, so further chess comments are given in the Appendix.

It is very hard to clearly explain that **19 Rf1-e1** was really a good move, and how AlphaZero was able to find this decision. It seems that the combination of AlphaZero's Monte Carlo Tree Search and AlphaZero's move evaluation stored in its neural network somehow resulted in such a deep positional judgement.

One possible explanation for this might be as follows. Positional evaluation in chess takes into account static features in the current position. Such features tend not to change quickly during play, so they have long-lasting effects. An example of such a positional feature is weak pawns that cannot move and are hard to defend, and can thus become targets for enemy pieces in the course of the game. Another example are chain formations of blocked pawns that create more space for one of the sides. More space gives to one side better

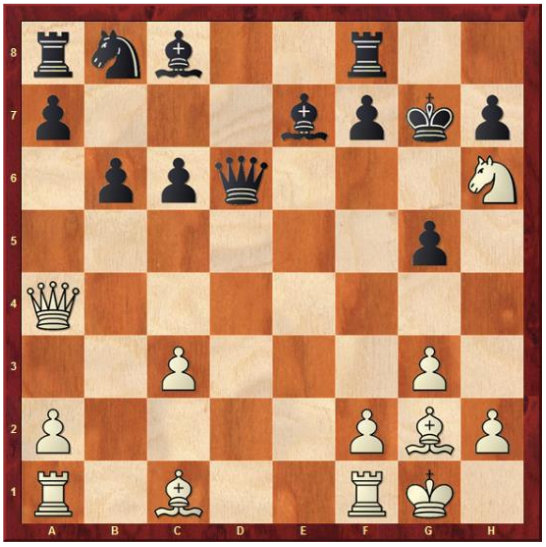

Figure 1: Position after Black's move **18 ... g6-g5**. AlphaZero here played the surprising **19 Rf1-e1**, leaving White knight on h6 undefended.

chances to manoeuver their pieces and thus create chances for attack in the long run on the part of the board with space advantage. However, it usually takes many moves before such positional advantages can be exploited and turned into something more tangible like material advantage. It may also happen that positional advantage cannot be exploited at all. In such cases, the positional advantage simply evaporates in the long run. It is very difficult for humans to estimate whether positional advantage can eventually be converted or not because it is hard to see so far into the future of the game. It is often far enough that this may also be a problem for a typical chess engine that uses Alpha-Beta search. Here it is that Monte Carlo Tree Search might be much more appropriate because it is more selective and can therefore go much deeper than Alpha-Beta. Of course, for Monte Carlo search to be successful, it has to be well guided by the move probability estimates, which seems to be a major strength in AlphaZero. In position of Figure 1, the positional advantage of AlphaZero's knight sacrifice was only converted into material gains after twenty moves. This is too deep for Alpha-Beta search, but possible to see by MCTS. Now although it seems that random trials of MCTS are quite absurd to be carried out by a human player, it can be imagined that something roughly similar is actually done by strong human players. When a good player tries to estimate how concrete the consequences of a positional advantage may become, he or she tries to calculate very deeply and selectively sample variations. Favourable results from these variations will increase the player's confidence into the correctness of a positional sacrifice.

6 Can humans understand and learn from Alpha Zero?

The chess moves played by AlphaZero in the example above call for an explanation. Ideally, AlphaZero would

be able of comment on its games and explain its decisions in human-understandable terms. So humans would be able to learn from AlphaZero new chess concepts and ideas, enhance their own chess knowledge and be able to use it in their own play.

In this respect, the lack of explanation facility is a serious limitation of AlphaZero paradigm, and many forms of machine learning in general. Many of the present successful ML methods that can outperform humans have this same limitation. It is hard for humans, and human experts, to understand what has actually been learned by the program. Ideally, learning programs should be able to explain what they discovered through learning, so that this new knowledge could also be used by humans.

This idea has been around in the area of machine learning almost from its beginning, at that time also known under the term "machine synthesis of expert knowledge". This phrase was coined by Donald Michie in early 1980's, some time before the idea became generally accepted. Donald also set up an international association called ISSEK (International School for the Synthesis of Expert Knowledge). The main activity of ISSEK was a series of workshops in 1980's and 1990's to enable a collaboration among research laboratories interested in developing machine learning techniques for the synthesis of new knowledge from data. As an attempt at precisely defining these aspects of machine learning, Michie (1988) defined three criteria for machine learning, and it will be useful to repeat them here. Essentially, these criteria were:

(1) Weak criterion: the learning system improves its performance through learning from experience

(2) Strong criterion: as (1), plus the system can output what it has learned in explicit symbolic form

(3) Ultra-strong criterion: as (2), plus the explicit symbolic description produced can be used by a human operationally, that is to improve the human's own performance at solving the task

By far the strongest attention in machine learning has been devoted to criterion (1), and the imbalance of attention between the three criteria has probably been increasing over time. Importance of criteria (2) and (3) with relevant examples was discussed for example in (Bratko 1997).

There has been recent renewed interest in relation to the latter two criteria within Explainable AI (XAI 2017; Miller 2017). A related issue is the question of comprehensibility of a description by humans. For a human to be able to use operationally what was learned, the human at least has to understand the result of learning. Therefore, for the ultra-strong criterion to be applicable in practice as a measure of success, a measure of comprehensibility by a human of a (machine-generated) description is required. Although such a need has been often observed in machine learning, little progress seems to have been made in this respect. (Muggleton et al. 2018) is a rare recent attempt at defining an operational measure of comprehensibility.

In terms of Donald Michie's criteria, AlphaZero has been a tremendous success in terms of the weak criterion for machine learning, but no attention seems to have been paid to the other two criteria in the development of AlphaZero. As a result, AlphaZero has miraculously acquired a lot of new game-specific knowledge, but at the moment it is hidden from humans in a black box. As described by Voosen (2017), a human interested in that knowledge can play a time consuming game of an AI detective to uncover small bits of that knowledge in the box. Fundamental progress in terms of Michie's ultra-strong criterion with AlphaZero, and other similarly influential systems that will appear in the future, will be needed to increase their impact in the important direction of improving human knowledge.

7 Conclusion

In this paper, we considered some limitations of AI techniques on which AlphaZero is based. These limitations are indicative of some directions for future research in AI. Many games played by AlphaZero are very interesting, and it seems that, at least in chess, AlphaZero has discovered new concepts that human players are not aware of. At the moment, humans can only make guesses about what these new concepts might be. Therefore, the development of explanation techniques, aiming at human-friendly conceptualisation of the automatically acquired game-playing knowledge would be very well motivated. Also, improving machine learning methods towards more data-efficient learning would be important for applicability in many real-world domains.

8 Acknowledgements

I would like to thank Matej Guid, Martin Moina and Marjan Šemrl, a former correspondence chess world champion, for discussion.

9 References

[1] I. Bratko, Machine learning: between accuracy and interpretability. In: *Machine Learning, Networks and Statistics*. Eds. G. Della Riccia, H.-J. Lenz, R. Kruse. Springer, 1997. pp. 163-178.

[2] M. Guid, AlphaZero. *Šahovska misel* (*Chess Thought Magazine*), Februar 2018 (in Slovene).

[3] D. Michie, Machine learning in the next five years. *Proc. Third European Working Session on Learning*, pages 107–122. Pitman, 1988.

[4] T. Miller, Explanation in AI: Insights from the social sciences. 2017. arXiv.org > cs > arXiv:1706.07269

[5] S.H. Muggleton, U. Schmid, C. Zeller, A. Tamaddoni-Nezhad, and T. Besold. Ultra-strong machine learning - comprehensibility of programs learned with ILP. *Machine Learning*, 2018. In Press.

[6] Claude Shannon (1950). Programming a computer for playing chess. *Philosophical Magazine*. **41** (314)

[7] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I.

Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[8] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550:354–359, 2017.

[9] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, D. Hassabis, Mastering chess and chogi by celf-clay with a ceneral Reinforcement Learning algorithm. 2017. arXiv.org > cs > arXiv:1712.01815

[10] T. Wiley, C. Sammut, B. Hengst, I. Bratko, A multi-strategy architecture for on-line learning of robotic behaviours using qualitative reasoning. *Advances in Cognitive Systems Journal*, **4** (2016), pp. 93-111.

[11] P. Voosen, How AI detectives are cracking open the black box of deep learning. Science, July 2017.

[12] XAI 2017 (Proc. IJCAI-17 Workshop on Explainable AI), 2017. http://www.intelligentrobots.org/files/IJCAI2017/IJCAI-17_XAI_WS_Proceedings.pdf

10 Appendix: Detailed analysis of the game AlphaZero vs. Stockfish from position of Fig. 1

In position of Fig. 1, White knight is in trouble and it seems that he has to retreat from h6 to g4. This is the only safe square for the knight. The knight is now under attack of Black bishop on c8, but the knight is defended by White queen. However, White's problem is not completely over because Black can try to chase White queen away from defending the knight on g4. Thus the following continuation is logical: **19 Nh6-g4 b6-b5** (attacking White queen), **20 Qa4-e4** (the only square from which the queen can still defend the knight, but now Black has double attack on White queen and knight with the next pawn move) **f7-f5**. Fortunately for White, White can check Black king and the following variation is more or less forced: **21 Qe4-e5+ Kg7-f7 22 Qe5xd6 Be7xd6 23 Rf1-d1 Bd6-c7 24 Ng4-e3**. Now White knight has survived the trouble, but Black is a pawn up and the position is somewhat better for Black. This variation is also given as the best possibility for White by typical chess programs, and it is what every reasonable human player would do, accepting a worse position as the least possible damage. AlphaZero however very surprisingly played **19 Rf1-e1**, leaving the unfortunate knight on h6 under threat. The knight can now be immediately captured by Black king: **19 ... Kg7xh6** which Stockfish actually did in the game. A typical chess program now evaluates the position as considerably better for Black. Black is a whole piece up. True, White can play **20 h2-h4** and Black king will be feeling a little uncomfortable, so White does have some compensation for the sacrificed piece. But is this compensation sufficient? The answer appears to be a clear "no" to practically any human player, as well as any chess program other than AlphaZero. Black has big material advantage, and White seems to have no tangible compensation in return. It is too complex to calculate all the possible continuations to sufficient depth in this position because there are no forced variations clearly favourable to White or Black. So in practice this position can only be evaluated through a kind of "intuitive positional judgement" (in quotes when it refers to a computer). In this case, AlphaZero was in fact capable of such deep positional judgement, something that is extremely difficult for humans, and so far has been considered even harder for machines. In the game, after **19 Kg7xh6**, the following moves were played: **20 h2-h4 f7-f6 21 Bc1-e3 Bc8-f5 22 Ra1-d1 Qd6-a3 23 Qa4-c4 b6-b5 24 h4xg5+ f7xg5 25 Qc4-h4+ Kh6-g6 26 Qh4-h1**. The position at this point is shown in Fig. 2.

White queen now looks very passive in the corner, and thus White, still with a piece down, seems considerably worse. But the prospects of White queen on h1 are actually excellent. The idea is that the queen at h1 supports the move by White bishop **g2–e4**, and after the exchange of the light coloured bishops, White queen will

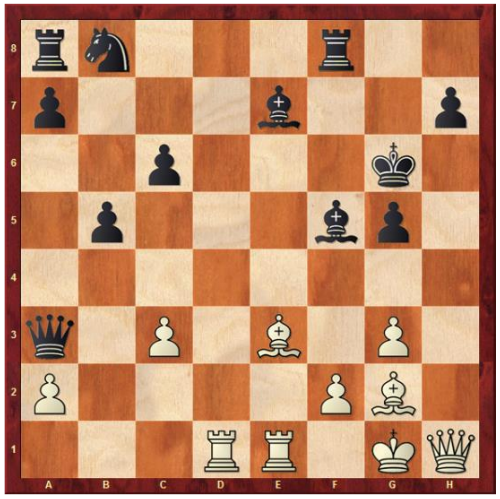

Figure 2: Position after 26 Qh4-h1.

threaten to enter the center via light squares with great force. This actually happened in the game and 15 moves later White achieved a clear advantage. So the controversial move **19 Rf1-e1** by AlphaZero in position of Fig. 1 turned out to be a brilliant positional sacrifice much admired by the chess world.