University of Nottingham

Computer Science with Artificial Intelligence MSci

G54IRP/COMP4027 — Individual Research Project

AI for General Video Game Playing

Author
Benjamin Charlton
psybc3@nottingham.ac.uk
4262648

Supervisor
Dr. Ender Özcan
Ender.Ozcan@nottingham.ac.uk

7th December 2017

Contents

1 Introduction

1.1 Introduction

- Popularity of Video Games
- Variety of Video games being played
- Video Games as a testing ground for AI

1.2 Planning VS learning

- As the title says, just to get some definitions out of the way

1.3 Motivation

- Individual desire to get better at deep reinforcement learning
- Learning more Methods
- Applying Methods
- Learning to use more robust and real world frameworks for DRL
-
- Seeing the state of the art in DRL and AI game playing

1.4 Aims and Objectives

- Look at the project plan aims and objectives

2 Related Work

2.1 AI and Game Playing

- AI approaches have specialised heuristics or only been developed for a single game
-
- Go — previous interim report / PP
- Chess — previous interim report / PP
- ALE — Learning works for games
- OpenAIFive — Learning works on big games too, and MP
- GET MORE THINGS FROM PP
- Conclusion for this section
-
- Dartmouth Workshop

2.1.1 Early Artificial Intelligence

The history of AI game playing begins near the start of artificial intelligence as a field, in the 1950s. Strachey created a draughts player for one of the first general computing machines (Manchester Ferranti Mark I) which by 1952 could "play a complete game of draughts at a reasonable speed"[2]. Prinz wrote a simplified chess player for the Manchester Machine as well which could solve the mate-in-two problem. This meant that if there was a checkmate solution in 2 turns it could successfully find it[2]. Prinz simply used an exhaustive search technique to find the correct moves, and even though computing power was limited at the time it was clear that this wouldn't scale to full games. This lead to Turning starting to program 'Turbochamp' a chess program that would be able to play a full game of chess using heuristics[2]

These simple games were made before the term artificial intelligence was being used even in an academic setting showing how natural AI and game playing go together.

2.1.2 Deep Blue and Chess

Chess was an early and significant example in the history of AI game playing. In 1997 IBM managed to beat the reigning world champion, Garry Kasparov, using their custom developed machine, Deep Blue[1]. This was significant as, at the time, creating a winning chess AI was seen to be the next big milestone at the time in AI.

To achieve this Deep Blue used a combination of techniques with the main underlying AI technique being a search method. As Chess is a deterministic game and both players have complete information of the board state it was possible for Deep Blue to generate future board states. With this forward model its possible to generate a search tree of possible moves and their resulting game states The tree was efficiently generated by a combination of a massively parallel architecture over 30 nodes and the fact that each node has special purpose chess chips, generating around 6–8 moves ahead on average. Alpha-beta pruning was used in a MinMax algorithm to help efficiently search the tree while using the custom hardware to evaluate each node quickly.

IBM had proven that search methods could achieve strong results against human opponents but this was mostly due to brute force computing power and a heavy reliance on expert knowledge. The expert knowledge came from other grandmasters and was used in the form of an opening/closing move database and the special purpose hardware to evaluate board states. While higher computing power will always benefit AI techniques, later techniques have been developed to reduce the need of expert knowledge and to make more efficient use of hardware available.

2.1.3 AlphaGo

Although Deep Blue had proven that AI systems could beat an expert in games, the methods used wouldn't scale to more complex games such as Go, new methods would be needed to tackle more complex games and to further the field of AI. Go was set as the next milestone in AI game playing by the community as it proved vastly more complex than chess while still having few rules and a simple goal, like chess. To compare the difficulty between the 2 games, chess uses a 8×8 board while Go uses a 19×19 board. What is a better indicator of the AI performance is the branching factor of the games for each move, as this more accurately shows how large the search tree grows for each possible move. On average, Chess has a branching factor of 35 and Go has a branching factor of 250[3]. In 2016 DeepMind reached this milestone with AlphaGo claiming its victory against a 9th dan professional Go player, Lee Sedol, in March 2016[5].

This achievement was accomplished with a combination of Monte Carlo Tree Search (MCTS) and 2 Deep Neural Networks trained with reinforcement learning, one was a value network to assess the board predicting the winner and the other a policy network to predict which move would be played next by an expert. By augmenting MCTS with the policy network, the problem of the large branching factor was greatly reduced, the value network made the addition that it was quicker and more accurate at determining who would win than conventional methods and didn't require an ending game state to be reached. Initially these networks were trained on a large number of human played games as a labeled dataset, after this 'boot strapping' phase they were trained through instance of self play.

Between October 2015 and AlphaGo's retirement in May 2017 there were 3 significant versions of the AI system. AlphaGo Fan was named after its opponent Fan Hui a European 2nd dan professional Go player, where it won 5–0 being the first ever computer to beat a human professional at Go without a handicap. AlphaGo Lee was named after its opponent Lee Sedol a South Korean 9th dan professional Go player, in this match AlphaGo won 4–1 and was seen as an AI system successfully beating a Go champion at their own game. AlphaGo Master was the final version of the system and was named after the online account it played on for 60 Games against human opponents, it managed to win all 60 games and was finally revealed to have been a computer system to the Go community. It finally retired at the Future of Go summit where it played a further few games beating another champion Ke Jie and a human team of 5 Go professionals. After the summit it was clear that AlphaGo was a champion of the game and as a gift to the Go community they presented a series of 50 games for people to analyse.

2.1.4 AlphaGo Zero

After retiring AlphaGo Deep Mind was already working on a successor that would seek to fix some of the downsides of the AlphaGo system. They sought to create a system that could play 'tabula rasa' (latin for from a blank slate) requiring human expert decisions to learn from as they say that 'Expert data sets are often expensive, unreliable or simply unavailable'[6]. The problem came from the initial supervised learning stage of AlphaGo's development, as it required millions of human expert games with the correct outcomes to initially learn from before it began using reinforcement learning and self play to hone its skills.

There are many issues often associated with using supervised learning, 3 of the main issues were brought up when describing the expense, reliability and the availability of these datasets. While AlphaGo had proven that there was suitable enough data for learning the game of Go, a motivating factor for the team was showing that this wasn't needed and hopefully find a method that could then later be adapted to other problems with these issues. Another issue was a notion of a skill ceiling, as such to say that the system can only learn to be as good as the data provided to it an idea that is also seen in human learning where a student can only be as intelligent as their teachers. This was to say that if a system learnt from human experts then it would only slightly build upon the expert knowledge where as their could potentially be ideas unconsidered by humans that could lead to a stronger system. AlphaGo had shown evidence of this when it played a so-called 'divine move' against Lee Sedol indicating there was more potential for innovation in strategy even in an ancient classic game like Go. Furthermore by also not having a reliance on human expert knowledge an AI system could be developed to complete task where there is limited or no expert knowledge to help guide it.

AlphaGo Zero differs in 4 key ways to its predecessor AlphaGo:

- It trained by selfplay reinforcement learning exclusively, starting with random play.
- It only used the 19x19 board with the stone colours as input.
- It used a single neural network rather than 2.
- A simpler tree search was used not requiring a Monte Carlo rollouts

By using this form of selfplay it means that AlphaGo Zero gradually improved after every game, but it also allowed for it to have an equally strong opponent at every step of the learning process. This results in games which much more can be learnt from as it is hard to learn from a system if you are always winning or losing.

The single neural network was designed to take in the raw current board state and output a prediction value of the chance to win and a vector of move probabilities. This was compared with the true winner of the game from the self play and the search tree probabilities. The comparison formed the loss function of the neural network and was updated via gradient descent.

The network is used by the search algorithm by selecting the likely moves to make and searching a few moves forward, using the networks winner prediction to select a strong move to make where it has the greatest chance of winning.

Ultimately when tested against different versions of AlphaGo, AlphaGo Zero won with conclusive results. Winning 100–0 against AlphaGo Lee and 89–11 against AlphaGo Master. They also gave a rating to a version of AlphaGo Zero that just used the neural network, selecting the move it thought would have the highest probability of winning without using a tree to look ahead. This neural network only system didn't out perform any version of AlphaGo but it came about 3% lower than AlphaGo Fan and was 50% stronger than any other Go AI tested.

Although AlphaGo Zero was intended to work without any human domain knowledge, some was used to help the training process. In particular the game of Go is invariant under rotation, reflection and colour transposition allowing for the same board to be used for training multiple times, vastly increasing the amount of training data generated for the neural network. The MCTS that was used to help train the network (not play) was given perfect knowledge of the rules of Go as well.

2.1.5 OpenAI Five

In recent years research into video game playing AIs have begun, due to many simpler games being played to a high level via AI techniques and the field moving towards tackling more complex systems. One of the most notable examples is OpenAI who have created a team of 5 bots that can play a limited version of DotA 2[4]. This was done using an LSTM neural network and has managed to beat several amateur and semi-pro teams. While it cannot currently beat a professional team it does hold up against them quite well.

2.1.6 notes

- OpenAI Five
- Atari

2.1.7 Conclusions

- Deep Blue proves that games can effectively use forward planning to beat human experts, early techniques require a lot of power so intelligent methods are needed to speed up results/improve search depth
- AlphaGo Fan, Lee and Master show planning is still effective but by using more advanced techniques we can generate search trees better (ANNs). Also shows how learning can benefit AI techniques and boost the powerful planning techniques.
- AlphaGo Zero shows how learning doesn't need to require expert knowledge and can just learn from lots of experience in the environment
- AlphaZero shows how more generic systems can still perform well allowing for techniques that could be transplanted into many environments and still perform well. Also showing that although dedicated hardware was used no purpose built hardware or software was used to exploit features of the games unlike previous versions or Deep Blue.

- AlphaGO Zero using only the network doesn't achieve state of the art performance but comes close and doesn't rely on a look ahead style search tree.

2.2 GVGAI and VGDL

- TODO Make some notes up here

2.3 OPEN AI GYM and GVGAI GYM

- What is OPEN AI GYM
- Why is OPEN AI GYM helpful for RL
- GVGAI GYM
- Initial results from GVG AI GYM paper
- Maybe some more here

[7]

2.4 Subsection to end all subsections

- Maybe something about the open AI baselines
- Depends what technique I would be using but I should look into that
- Where does reinforcement learning come into this bad buoy

3 Appendix

3.1 Meeting Minutes

3.2 Work Plan

3.3 References

References

[1] Murray Campbell, A Joseph Hoane, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

[2] Jake Copeland. A brief history of computing. http://www.alanturing.net/turing_archive/pages/Reference%20Articles/BriefHistofComp.html#MUC, June 2000.

[3] Alan Levinovitz. The mystery of go, the ancient game that computers still can't win. https://www.wired.com/2014/05/the-world-of-computer-go/, May 2014.

[4] OpenAI. Openai five. https://blog.openai.com/openai-five/, 2018.

[5] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[6] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[7] Ruben Rodriguez Torrado, Philip Bontrager, Julian Togelius, Jialin Liu, and Diego Perez-Liebana. Deep reinforcement learning for general video game ai. *arXiv preprint arXiv:1806.02448*, 2018.