Artificial Intelligence and the Applications of Game Theory in Zero Sum Games

On my word of honor, I have neither given nor received any unauthorized assistance on this assignment.

Logan Carlisle

Culminating 2023-2024

Abstract

Game theory is applicable in many fields of study, Ranging from mathematical economics, evolutionary game theory,computer science, among other interactions with at their core lies strategic interaction. The purpose of this field initially is to find mathematically proven strategies for a 2 player zero-sum game that can be defined as. Using the methods from the field of game theory alongside artificial intelligence which is often commonly used in tandem, I wanted to be able to create an algorithm that could play chess. An endeavor that has been done by many coders since the dawn of computers and even before then, starting with **El Ajedrecista.** Using tried and true methods standing on the shoulders of giants I took inspiration from many notable chess engines such as StockFish, DeepBlue, AlphaZero,ect. Through this project I was able to create an algorithm that was able to play simple chess to a degree.Many issues arose due to the efficiency of the algorithms largely in due part being written in python, a coding language that is notorious for being slow. However through this research and development of a program I was able to build a base of knowledge for computer science, a field I hold many interest in.

Methodology `

The methodology for this chess engine product involved a varied search and analysis of relevant parts of various Chess Engines.Considering prior experience in coding in python the project was developed in that language. The search utilized a combination Inclusion criteria to make sure my product was well constructed. The initial screening process involved evaluating differences between top chess engines and what skills the researcher currently has. Thematic analysis was then conducted to identify common themes and patterns in the inspiration for the engine, which were subsequently summarized to provide a coherent and correct overview of the main points and arguments in the code. The findings were discussed in relation to the research objectives and research questions, and limitations of the software were acknowledged.

Question

How are algorithms and Artificial Intelligence used in the field of game theory?\

Rationale

Interested in both chess and artificial intelligence, the researcher looked into what is needed to develop a chess engine that can play the game utilizing game theory and neural network applications. It seemed like a great opportunity to merge my interests and learn about game theory at the same time. With my interest in computer science and mathematics, this research and study will be useful for my college objectives and coursework. Intending to use my culminating study to create a well-considered article and presentation that will effectively communicate my independent research.

Research Classification

Conducting both scientific and investigative research.This essay will culminate the learning of the research on the topics of Game Theory, AI, Chess, as well the various histories and intracities of all.

Introduction

The domain of Game theory is a mathematical endeavor with many applications and tools.Widely used for Zero-Sum-games which are defined as a game with two sides where one's player advantage is another loss. The analysis and playthrough of these Zero-Sum games can be done in a variety of ways, searching will be done through a heuristic algorithm which learns through trial and error. These algorithms can be used in Chess as a general environment, which fits the requirements for a Zero-Sum game, and there is the added benefit of evaluating a given position in chess is relatively simple. Evaluation can be done through a Hard coded style or have the given chess position evaluated by anEfficiently Updateable Neural Network or NNUE for short. An application of artificial intelligence.

History of Chess Engines

Chess engines, which are programs that run on computers to play chess. With the dawn of computers brought forth by amazing minds such as Alan Turing and Cluade Shannon, interest in using computers to test/benchmark and¨solve chess¨ became an idea. With Cluade Shannon in particular publishing a program that could possibly work, and later Turing implementing it.(citation needed)

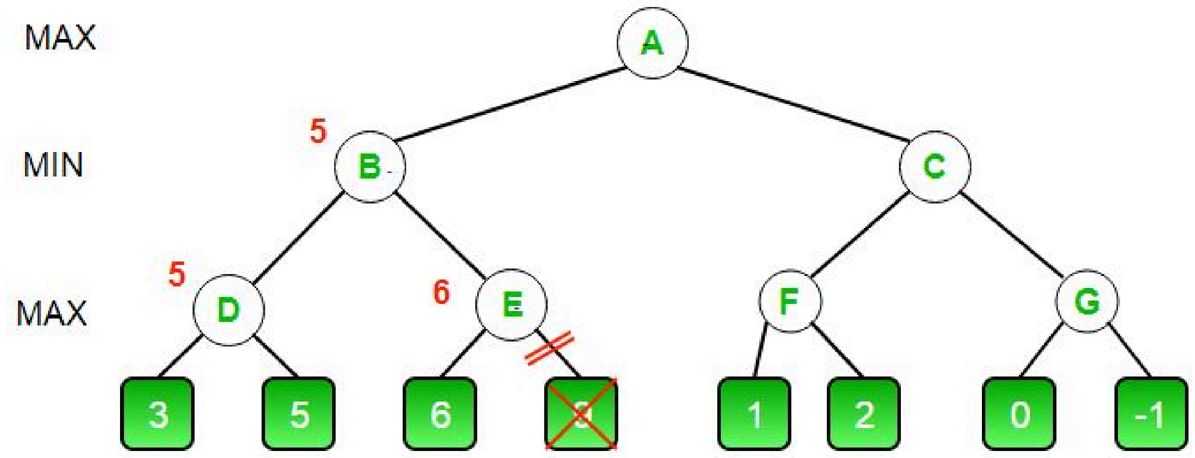
As the years passed and computing power continued to advance, pushing the envelope, IBM used their supercomputer to play the world number 1 at the time, Gary Kasparov. IBM, wanting to showcase their powerful technology, built a supercomputer to play a grandmaster in the game of chess which at the time was thought to be almost impossible due to the sheer amount of chess positions considered to be around 10 to the 40th power. However DEEP BLUE with all of its processing power was able to parallelize the alpha-beta pruning method and in doing so was able to evaluate upwards of 200 million chess positions a second,defeating Kaspraov in .

Evaluation function

In chess evaluation functions are used to judge a given board state,a state is the current chess position. To evaluate these positions there are multiple things to look at,the simple beginner evaluation function usually starts with a simple count material function. Which simply sums up who has more pieces. However there are some flaws to this example such as positions where sacrifice greatly improve performance.So on top of that simple material evaluation you can also analyze positional and strategic play such as a bishop on a strong diagonal of a open file

Minimax

The Minimax algorithm is a search algorithm used in a variety of zero sum two player games, it's also used in the field of statistics and philosophy in niche cases. The algorithm in chess explores a move tree, by recursively simulating moves to a certain depth, then with an evaluation function scoring that move and returning the move with the highest winning probability. In chess the branching factor of this tree is roughly 10 to the 30 power, at a simple depth of four with an unoptimized Minimax algorithm,roughly 3.5 millions positions are looked at.This can be a time and computation problem as with each increase in the depth(how many moves the chess engine looks ahead), the number of moves grows exponentially. So to improve this search algorithm Alpha-beta pruning was invented, the essence behind Alpha-beta pruning is that, why look at a move if the opponent is most likely going to play it.



However it is to note that Alpha-beta pruning is only good when you can prune moves, if by chance the moves are ordered worst to best by coincidence, then no move would be able to be pruned. How can we exploit this? We can prune branches to the highest extent when moves are ordered best to worse, while we don't know what the best moves are that's the whole point of this algorithm, we can guess if a move is good or not.For example commonly good moves are capturing pieces of higher material with those of lower material such as a pawn capturing a queen. Another good move would be promoting a pawn, likewise we can also guess if a move is bad, such as putting your strongest piece in a square attacked by pawns. Beyond optimizing software for the search algorithm we can also optimize our hardware.When we run code we usually run it on a single Central Processing Unit or CPU for short, and while using a CPU we usually run code on a single thread in most cases. The CPU can have upwards to 128 threads,my PC's CPU only has 12 threads. The importance off these threads is that they can each run their own code in parallel or at the same time.Fully utilizing the full amount of threads your CPU offer can greatly speed of search algorithms.Think about through this analogy, There are 5 mountains with five different orchards and you want to see which orchard has the best apple. So do you

A) Climb each mountain alone and find the best apple.

B) Get five friends to climb each mountain separately and bring the apples back to try them.

B if you're able to find 5 people is the correct answer. This is the same thing as assigning multiple CPU threads to look at different parts of the move tree to find the best move. And is the method behind looking at movies in parallel.

Monte Carlo Alternative

One way is not the only way,the Minimax is used extensively in the top search algorithms, however the another search algorithm that dominates online search algorithms is the Monte Carlo search tree.A algorithm built in complete randomness and brute force you would think would not work well in chess, however it does. Used in top level chess engines such as Leela Chess Zero and AlphaZero, former inspired by the latter produced by google deepmind.The monte carlo search tree is used in conjunction with a deep ai neural network.The monte carlo search tree is divided into 4 parts . selection where it starts at a certain board position The predecessor of the monte carlo method served as inspiration for Rémi Coulomb's 2006 description of applying the Monte Carlo approach to game-tree search, which led to the creation of the term "Monte Carlo tree search." L. Kocsis and Cs. Szepesvári created the UCT (Upper Confidence bounds applied to Trees) algorithm. The uct formula is used to pick the next move

$$UCT_j = X_j + C * \sqrt{\frac{ln(n)}{n_j}}$$

. It takes into account the success of moves done in the past and switching it up to find something new.

Conclusion

Dating back to at least Eastern Zhou Period game theory was used by people Sun Tzu in the art of war. Heavily steeped in statistics and odds, game theory is utilized in many varying fields such as evolutionary biology under maynard smith. With the dawn of computer and automation game theory bloomed, speeding up the process of calculation originally done by human hand. A very interesting field that could be used in a variety of ways. Multiple different tools of game theory can be mixed with an equal number of AI related tools to form an engine to play chess. Hardware and optimization is at the crux of creating a strong chess engine.

Works Cited

Alpha-Beta Pruning Algorithm: The Intelligence behind Strategy Games, www.researchgate.net/publication/360872512_Alpha-Beta_Pruning_Algorithm_The_Intelligence_Behind_Strategy_Games. Accessed 5 Jan. 2024.

Buber, Ebubekir & Diri, Banu. (2018). Performance Analysis and CPU vs GPU Comparison for Deep Learning. 1-6. 10.1109/CEIT.2018.8751930.

Jouppi, Norman P., et al. "In-Datacenter Performance Analysis of a Tensor Processing Unit." *arXiv.Org*, 16 Apr. 2017, arxiv.org/abs/1704.04760.

Karpathy, Andrej. "Let's Build GPT: From Scratch, in Code, Spelled Out." *YouTube*, YouTube, 17 Jan. 2023, www.youtube.com/watch?v=kCc8FmEb1nY&t=31s.

Lague, Sebastian. "Coding Adventure: Making A Better Chess Bot." *YouTube*, YouTube, 30 June 2023, www.youtube.com/watch?v=_vqlIPDR2TU&t=16s.

Lague."Coding Adventure: Making A Better Chess Bot." *YouTube*, YouTube, 30 June 2023, www.youtube.com/watch?v=_vqlIPDR2TU.

Maharaj, Shiva, et al. "Chess AI: Competing Paradigms for Machine Intelligence." *arXiv.Org*, 23 Sept. 2021, arxiv.org/abs/2109.11602.

Nasu, Yu. " Efficiently Updatable Neural-Network-Based Evaluation Functions for Computer Shogi." *GitHub*, github.com/ynasu87/nnue. Accessed 5 Jan. 2024.

Silver, David, et al. "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm." *arXiv.Org*, 5 Dec. 2017, arxiv.org/abs/1712.01815.

Vaswani, Ashish, et al. "Attention Is All You Need." *arXiv.Org*, 2 Aug. 2023, arxiv.org/abs/1706.03762.

Zhou, Xiao, et al. "Efficient Neural Network Training via Forward and Backward Propagation Sparsification." *arXiv.Org*, 10 Nov. 2021, arxiv.org/abs/2111.05685.