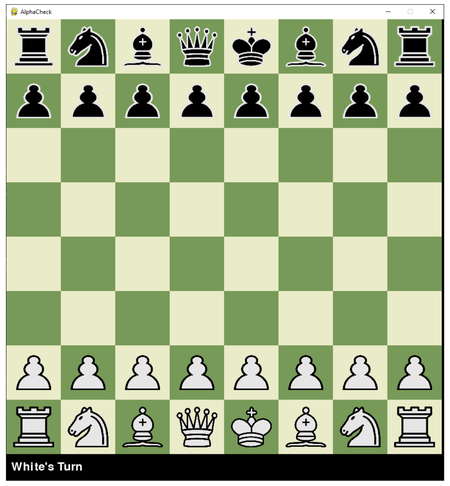
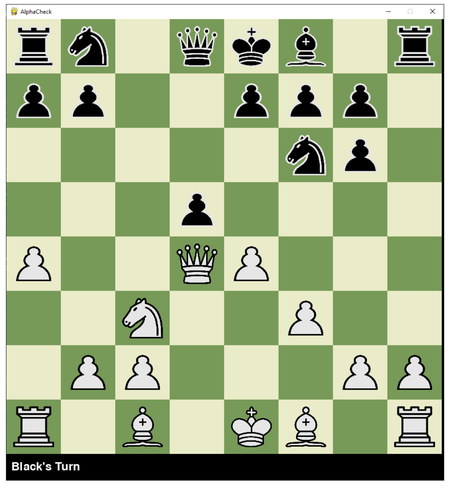
Python presents Chess AI

Scott Fisher • Dr. Michael Soltys • Comp-499

Why Machine Learning and Chess?

Chess is a game of strategy and decision-making, making it a perfect challenge for artificial intelligence. This project aims to develop a chess AI that can beat human players. The use of machine learning in this project will enable the AI to learn from human moves and strategies, becoming more proficient over time.





Methodology

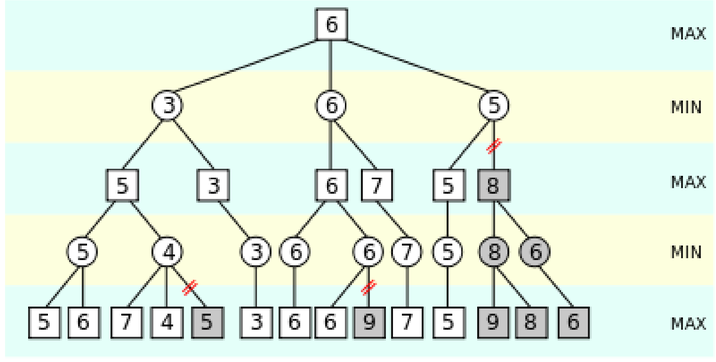
In this capstone project, the objective is to develop an intelligent chess AI capable of challenging human players, utilizing Python for the game environment and PyTorch for machine learning. The methodology encompasses establishing a functional chess game using the `python-chess` library, which handles the rules and mechanics of chess. The core of the AI will be a neural network model designed in PyTorch, trained to evaluate chessboard positions and generate strategic moves. This model will be trained on historical chess game data, requiring preprocessing of the dataset. Initially it will function as a basic heuristic-driven algorithm, which will be refined iteratively. Integration of the AI with the chess game is crucial to ensure seamless interaction, where the model's decisions are effectively translated into game moves. Rigorous testing will be conducted to evaluate the AI's performance, focusing on its decision-making abilities and adherence to chess rules. The final phase will involve assessing the AI's effectiveness and adaptability in various game scenarios, both against human opponents and other AI, to ensure it provides a challenging and engaging experience. This project not only aims to create a competent AI chess player but also serves as an exploration into the practical application of machine learning in game strategy development.

Technologies Used

Python-Chess: This is the main chess library I used for implementing all the logic behind the game of chess. This library implements: chess board construction, piece placement, move validation(including castling & en pessant), win conditions, and more!

Pygame: I used the Pygame library for the GUI of the project. With this we are able to play chess on a virtual chess board and see the pieces move. Without the graphics provided from Pygame this would have just been a game playable on the terminal with print statements.

PyTorch: PyTorch serves as the foundational machine learning framework for developing and implementing the neural network model. It facilitates the design, training, and integration of a deep learning model capable of evaluating chess positions and making strategic decisions. The main searching algorithm that will be utilized in this neural network is Alpha-beta pruning. A basic example of what this may look like can be visualized in the graph below.



Conclusions

In conclusion, this capstone project has underscored a fundamental principle in the field of machine learning: the effectiveness of an AI system is intrinsically tied to the quality and relevance of the reinforcement it receives. Through the development of a chess AI, we observed firsthand how the AI's proficiency is directly influenced by the training data and the reinforcement learning mechanisms employed. The project demonstrated that while algorithms and neural network architectures provide the necessary framework, it's the continuous and targeted reinforcement – informed by comprehensive, strategically rich datasets and iterative gameplay feedback – that truly cultivates an AI's capability to make complex decisions and adapt over time. This realization emphasizes the importance of not just technical robustness in machine learning applications but also the need for thoughtful, well-structured reinforcement paradigms to shape truly intelligent, responsive AI systems.

Literature Cited

Niklas Fiekas. (n.d.). python-chess 1.7.0 documentation. From https://python-chess.readthedocs.io/en/latest/

Chess.com. (n.d.). Games. From https://www.chess.com/games

PyTorch. (n.d.). Get Started. From https://pytorch.org/get-started/locally/

Pygame. (n.d.). News. From https://www.pygame.org/news

W3Schools. (n.d.). Python Machine Learning - Getting Started. From https://www.w3schools.com/python/python_ml_getting_started.asp

Further Development

Difficulty Levels: I would like to eventually develop different difficulty levels that you can play against the AI. This might limit the range of search that the AI is able to search to effectively "dumb it down".

Mid game and End game book: Currently the machine learning program has only worked with opening hands and uses the rest of its experience is from playing games and learning its own mid game and end game strategies.

Further AI algorithm development: New strategies are being developed and deployed for machine learning algorithms. By no means is the algorithm I have developed the most optimized search algorithm on the market. So there will always be room for development and continual optimization.