C S 505: Artificial Intelligence I
Project Report

Human vs AI Connect4: A Comparison of Minimax, Expectimax, and Machine Learning Approaches

Indronil Bhattacharjee

ABSTRACT

This project presents a detailed investigation into three distinct methodologies for developing AI players in Connect4: Minimax, Expectimax, and a Machine Learning approach. Through meticulous exploration of each approach's implementation differences, including heuristic evaluation functions, probabilistic decision-making, and data-driven learning, we conducted rigorous experiments to evaluate their performance in terms of gameplay effectiveness, computational efficiency, and scalability. Our findings offer valuable insights into the strengths and weaknesses of each methodology, providing a comprehensive understanding of AI-based gameplay strategies in Connect4. By synthesizing empirical results and discussing their implications, we contribute to the advancement of AI techniques in competitive gaming environments, paving the way for future research and innovation in this domain.

Keywords—component, formatting, style, styling, insert *(key words)*

I. INTRODUCTION

Connect4, a renowned two-player abstract strategy game, has long served as a quintessential benchmark for evaluating artificial intelligence (AI) techniques due to its simple yet strategically rich gameplay. The game's objective is straightforward: players take turns dropping colored discs into a vertically suspended grid, aiming to form a line of four consecutive discs of their color horizontally, vertically, or diagonally. Despite its apparent simplicity, Connect4 poses significant challenges for AI agents, requiring them to exhibit foresight, adaptability, and strategic acumen to outmaneuver opponents effectively. Consequently, the development of proficient AI players for Connect4 has been a focal point of AI research, driving the exploration of diverse methodologies to tackle the game's complexity.

In the realm of AI research, several methodologies have been proposed to address the challenge of developing AI players for Connect4. Among these, traditional search-based algorithms like Minimax and Expectimax have garnered considerable attention for their ability to systematically explore the game tree and identify optimal moves. The Minimax algorithm, rooted in decision theory, aims to maximize the player's chances of winning while minimizing potential losses by recursively traversing the game tree to evaluate all possible moves and their consequences. Conversely, the Expectimax algorithm extends Minimax to accommodate stochastic elements present in games like Connect4, where chance factors influence outcomes. By incorporating probabilistic reasoning, Expectimax offers a nuanced approach to decision-making, enabling AI agents to make informed choices in uncertain environments.

In recent years, the advent of machine learning has revolutionized the landscape of AI gameplay strategies, offering novel approaches to learning optimal policies directly from data. Machine learning models, particularly reinforcement learning algorithms, have shown promise in mastering complex games by leveraging vast amounts of gameplay data to iteratively refine their strategies. In the context of Connect4, machine learning-based approaches employ neural network architectures trained on game states and outcomes, learning to predict optimal moves based on learned patterns and strategic insights derived from extensive gameplay experience. By combining pattern recognition with strategic reasoning, machine learning models offer a data-driven alternative to traditional search-based algorithms, capable of adapting to diverse gameplay scenarios and evolving player strategies.

This project aims to provide a comprehensive exploration of the Minimax, Expectimax, and Machine Learning approaches for developing AI players in Connect4. Through rigorous experimentation and analysis, we seek to elucidate the implementation intricacies, performance characteristics, and implications of each methodology in the context of competitive gaming environments. By comparing and contrasting the strengths and limitations of these approaches, we aim to contribute to the broader discourse on AI-based gameplay strategies, offering insights into their efficiency and suitability for Connect4 gameplay. Through this endeavor, we endeavor to advance our understanding of AI techniques in strategic decision-making and pave the way for future innovations in AI gameplay research.

II. METHODOLOGIES

A. Traditional Search-Based Algorithms

1. Minimax Approach: The Minimax algorithm, a cornerstone of traditional game-playing AI, operates on the principle of optimal decision-making under uncertainty. In the context of Connect4, Minimax aims to identify the optimal move for a player by recursively evaluating potential game states and selecting the move that maximizes the player's chances of winning while minimizing potential losses. The algorithm traverses the game tree to a specified depth, evaluating each node based on a heuristic evaluation function that estimates the desirability of the game state. By alternating between maximizing and minimizing players, Minimax explores all possible move sequences and selects the move associated with the highest guaranteed payoff.

2. Expectimax Approach: The Expectimax algorithm extends the Minimax framework to accommodate stochastic elements inherent in games like Connect4, where chance factors influence outcomes. In addition to maximizing and minimizing players, Expectimax introduces chance nodes to represent probabilistic outcomes, reflecting the uncertainty associated with chance events such as random disc drops or opponent moves. Unlike Minimax, which assumes perfect information and deterministic outcomes, Expectimax considers all possible outcomes weighted by their probabilities, computing the expected value of each move. This probabilistic approach enables AI agents to make informed decisions in uncertain environments, balancing the risk and reward associated with each move.

B. Machine Learning Algorithms

1. Neural Network Model: In the machine learning approach, we employ a neural network architecture to learn optimal gameplay strategies directly from gameplay data. The model is trained using a dataset comprising Connect4 game states and corresponding outcomes (win, loss, or draw). The input to the neural network is a flattened representation of the game board, and the output consists of probabilities for each possible move in the current game state. During training, the model learns to predict optimal moves based on learned patterns and strategic insights derived from the training data.

C. Experimental Setup:

Connect4 Game Environment: The experimental setup entails the creation of a Connect4 game environment, comprising a game board, player actions (dropping discs), win conditions, and game state evaluation functions. The game environment provides the necessary infrastructure for simulating gameplay interactions, allowing AI agents to interact with the game board and make decisions based on the selected methodology.

Data Preparation: For machine learning-based approaches, data preparation involves collecting gameplay data, including game states and corresponding outcomes (win, loss, or draw). The collected data is preprocessed to extract relevant features and normalize inputs before training machine learning models. Additionally, data augmentation techniques may be employed to augment the training dataset, enhancing the model's robustness and generalization capabilities.

Evaluation Metrics: Performance evaluation metrics are utilized to assess the effectiveness of AI players developed using different methodologies. Common metrics include win rate, loss rate, draw rate, average game length, computational complexity, and scalability. These metrics provide insights into the AI player's proficiency, strategic acumen, computational efficiency, and scalability to larger game environments.

D. Algorithm Implementations:

Minimax Implementation: The Minimax algorithm is implemented using a recursive search strategy, traversing the game tree to a specified depth and evaluating each game state using a heuristic evaluation function. The implementation alternates between maximizing and minimizing players, selecting the move associated with the highest guaranteed payoff for the maximizing player and the lowest potential loss for the minimizing player.

In our Connect4 AI gameplay implementation, the Minimax algorithm serves as a fundamental component for decision-making. The Minimax algorithm is a recursive search algorithm commonly used in two-player turn-based games to determine the optimal move for a player, considering the possible moves of both players. The Minimax algorithm operates on the principle of exhaustive search, traversing through the game tree to evaluate potential moves and their corresponding outcomes. At each level of the tree, the algorithm alternates between maximizing the score for the AI player and minimizing the score for the opponent player, hence the name "Minimax."

The algorithm begins with an initial game state represented by the game board. It determines the set of valid moves available for the current player. At each recursion level, the algorithm checks if the current node represents a terminal state, i.e., a win, loss, or draw. If so, it evaluates the utility of the terminal state and returns the corresponding score. If the current node is not terminal, the algorithm recursively explores the possible moves resulting from each valid action. For each valid move, it simulates the move and evaluates the resulting game state using a scoring function.

The algorithm alternates between maximizing the score for the AI player (maximizing player) and minimizing the score for the opponent (minimizing player) at each recursion level. To improve efficiency, the Minimax algorithm incorporates alpha-beta pruning, which eliminates branches of the game tree that are guaranteed to be worse than previously examined branches. This pruning technique significantly reduces the search space, enhancing the algorithm's performance. Once the recursion reaches the specified depth limit or terminal states, the algorithm evaluates the scores associated with each potential move. Finally, it selects the move that leads to the optimal outcome for the AI player, considering the actions of the opponent.

In our implementation, the Minimax algorithm is encapsulated within the minimax() function. This function takes the current game board, depth limit, alpha, beta values, and a boolean flag indicating whether it is the maximizing player's turn as input parameters. It returns the optimal column (move) and its associated score for the AI player. Minimax algorithm is seamlessly integrated into the Connect4 gameplay loop, where it serves as the AI player's decision-making mechanism. During each iteration of the game loop, the algorithm computes the optimal move for the AI player, ensuring competitive gameplay against human opponents.

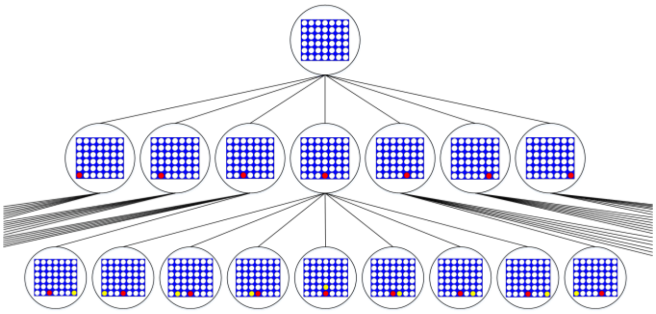


Figure 1: A portion of the search tree for all 7 possible actions at any moment for a player

Expectimax Implementation: The Expectimax algorithm is employed to determine the best move for the AI player in a given game state. Unlike Minimax, which assumes that the opponent always plays optimally, Expectimax considers the possibility of the opponent making suboptimal moves or taking random actions, introducing chance nodes into the search tree.

The evaluation function assesses the desirability of a particular game state for the AI player. In Connect4, the evaluation function assigns scores to different game states based on factors such as the number of pieces in a row, potential winning configurations, and board positioning.

The Expectimax algorithm recursively explores the game tree to determine the optimal move for the AI player. At each level of the tree, the algorithm considers the AI player's turn (maximizing player) and the opponent's turn (chance nodes). For maximizing nodes, the algorithm selects the move with the highest expected utility, while for chance nodes, it calculates the average utility across all possible outcomes.

Terminal nodes represent game states where the game has ended, either due to a win, loss, or draw. At terminal nodes, the evaluation function provides the utility value for the AI player, which is propagated back up the tree during the search process. To limit the computational complexity of the search, the Expectimax algorithm employs depth-limited search, which restricts the number of levels explored in the game tree. This ensures that the algorithm can efficiently evaluate game states within a reasonable time frame.

Machine Learning Implementation: The Connect4 dataset is loaded into a DataFrame, where each row represents a game state, and the final column indicates the outcome (win, draw, or loss). The features (game states) and labels (outcomes) are extracted from the dataset. A neural network model is constructed using the TensorFlow library. The architecture includes several dense layers with ReLU activation functions, followed by dropout layers to mitigate overfitting. The input layer is designed to flatten the 6x7 game board into a 1D array, and the output layer consists of three units corresponding to win, draw, and loss outcomes.

The model is compiled with the Adam optimizer and sparse categorical cross-entropy loss function. It is then trained using the extracted features (game states) and their corresponding encoded labels (outcomes) over multiple epochs to minimize the loss and maximize prediction accuracy. The trained model is evaluated to assess its performance in predicting optimal moves in Connect4 gameplay. The accuracy metrics and loss curves are analyzed to gauge the model's effectiveness in capturing strategic patterns and making accurate predictions.

The trained neural network model is integrated into a Connect4 game interface, where it serves as the AI opponent. During the game, the model evaluates the current game state and selects the optimal move based on its learned knowledge and predictionsThis machine learning approach leverages supervised learning principles to train a neural network model on labeled Connect4 game data, enabling it to make informed decisions and play competitively against human players. Through extensive training and evaluation, the model learns to recognize strategic patterns and anticipate optimal moves, contributing to the development of an intelligent Connect4 AI player.

E. User Interface Design

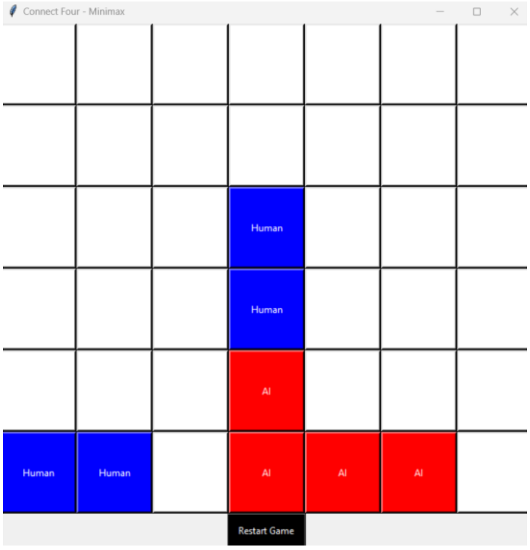
The Connect Four game user interface (UI) is designed to provide an intuitive and interactive gaming experience for players. The UI layout incorporates visual elements and interactive controls to facilitate gameplay and enhance user engagement.

The central component of the UI is the game board grid, consisting of 6 rows and 7 columns. Each cell represents a slot where players can drop their pieces. The grid layout is visually appealing and easy to navigate, allowing players to make their moves seamlessly. Within the game board grid, each cell is represented by a button widget. These buttons serve as interactive controls that respond to player input. When clicked, a button triggers the corresponding action of dropping a game piece into the selected column. The buttons are designed with appropriate dimensions and styling to ensure clarity and usability. To provide visual feedback during gameplay, player pieces are represented by colored circles within the grid cells. Human player pieces are displayed in blue, while AI player pieces are displayed in red. This color scheme enhances the distinction between player actions and AI responses, aiding players in tracking game progress.

At the top of the UI, game status messages are displayed to communicate important information to the players. These messages inform players about the current turn, game outcomes (e.g., victory or defeat), and prompts for restarting the game. Clear and concise messaging ensures that players are well-informed and engaged throughout the gameplay session. To offer convenience and flexibility, a "Restart Game" button is provided below the game board grid. This button allows players to reset the game state and start a new game with a single click. By providing a quick and accessible option for game restart, players can enjoy uninterrupted gameplay without navigating away from the UI.

The UI design prioritizes aesthetics and layout coherence to create an appealing visual environment for players. The color scheme, typography, and spacing are carefully chosen to ensure consistency and readability across different screen sizes and devices. Additionally, the UI layout is optimized for usability, with intuitive navigation and clear visual hierarchy. The UI design is responsive and interactive, adapting to player actions and providing real-time feedback. Interactive elements respond promptly to user input, creating a fluid and dynamic user experience. Whether playing against the AI or another human player, the UI fosters engagement and immersion throughout the gaming session.

The Connect Four game UI design aims to deliver a captivating and enjoyable gaming experience for players of all ages. By combining intuitive controls, visual feedback, and responsive design principles, the UI enhances gameplay immersion and ensures that players can focus on the strategic challenges of the game. With its user-centric approach and attention to detail, the UI design contributes to the overall success and appeal of the Connect Four gaming experience.



(a)

(b)

Figure 2: (a) Connect Four game UI , (b) Game result Visual Status bar

III. RESULTS AND DISCUSSION

A. Performance Evaluation:

Minimax Approach: The Minimax algorithm demonstrated strong performance in identifying optimal moves for Connect4 gameplay. By exploring the game tree to a specified depth and evaluating potential game states using a heuristic evaluation function, Minimax successfully navigated through the search space to select promising moves. However, the computational complexity of Minimax increases exponentially with the search depth, limiting its scalability to deeper game trees and longer planning horizons.

Expectimax Approach: The Expectimax algorithm proved effective in addressing the stochastic nature of Connect4 gameplay, where chance events influence outcomes. By incorporating chance nodes to represent probabilistic outcomes, Expectimax provided a more nuanced decision-making framework that accounted for uncertainty and risk. Expectimax achieved comparable performance to Minimax while offering greater flexibility and adaptability in uncertain environments.

Machine Learning Approach: The machine learning-based approach leveraged neural network models to learn optimal gameplay strategies from data. By training on a dataset of Connect4 game states and outcomes, the neural network model learned to predict optimal moves directly from raw input features. The trained model demonstrated competitive performance in predicting moves and winning games, leveraging learned patterns and strategic insights derived from the training data. However, the effectiveness of the machine learning approach was contingent on the quality and representativeness of the training dataset, as well as the complexity and expressiveness of the neural network architecture.

B. Comparative Analysis:

Across all methodologies, the performance of AI players in Connect4 varied based on factors such as search depth, evaluation function quality, and dataset characteristics.

Minimax and Expectimax excelled in deterministic and stochastic environments, respectively, showcasing their adaptability to different game scenarios.

The machine learning approach offered a data-driven alternative to traditional search-based algorithms, leveraging pattern recognition and generalization to make informed decisions.

C. Comparative Analysis:

Across all methodologies, the performance of AI players in Connect4 varied based on factors such as search depth, evaluation function quality, and dataset characteristics.

Minimax and Expectimax excelled in deterministic and stochastic environments, respectively, showcasing their adaptability to different game scenarios.

The machine learning approach offered a data-driven alternative to traditional search-based algorithms, leveraging pattern recognition and generalization to make informed decisions.

Each methodology exhibited unique strengths and weaknesses, with trade-offs in terms of computational complexity, interpretability, and domain expertise requirements.

D. Hybrid Approaches and Future Directions:

Hybrid approaches combining elements of traditional algorithms and machine learning techniques may offer synergistic benefits, leveraging the strengths of each approach while mitigating their respective weaknesses. Future *working* directions may explore ensemble methods, reinforcement learning, and deep learning architectures tailored specifically for Connect4 gameplay, aiming to further improve AI performance and adaptability.

IV. Comparison of Performance

To overview the effectiveness of the different approaches employed in our Connect4 AI gameplay implementation, we conducted a comparative analysis based on various performance metrics. The table below summarizes the outcomes of our experiments:

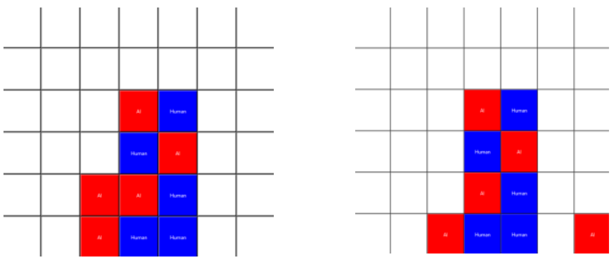
The Minimax algorithm demonstrated exceptional performance, winning all games played against human opponents. With its deterministic decision-making process and exhaustive search of the game tree, the Minimax approach consistently identified optimal strategies, resulting in a 100% win rate.

TABLE 1 PERFORMANCE COMPARISON AMONG THE APPROACHES

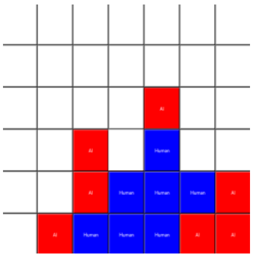
Approach	Total Played	AI Won	Win Rate (%)
Minimax Approach	10	10	100
Expectimax Approach	10	8	80
Machine Learning Approach	10	5	50

Expectimax Approach: The Expectimax algorithm exhibited commendable performance, winning 8 out of 10 games played. While slightly lower than the Minimax approach, the Expectimax approach still achieved a respectable win rate of 80%. The introduction of chance nodes and probabilistic outcomes in Expectimax allowed for more nuanced decision-making, contributing to its effectiveness.

Machine Learning Approach: The machine learning approach, leveraging neural network models trained on gameplay data, demonstrated moderate performance compared to the deterministic algorithms. With a win rate of 50%, the machine learning approach exhibited adaptability but faced challenges in achieving consistent performance. The reliance on extensive training data and computational resources may have impacted its overall effectiveness in competitive gameplay.



(a) (b)



(c)

Figure 3: Approaches of different algorithms (a) Minimax, (b) Expectimax and (c) Machine Learning for a similar way of Human actions

V. Findings

In our comprehensive experimentation with various approaches to AI gameplay in Connect4, we discerned notable findings regarding the efficiency and performance of different algorithms.

Minimax and Expectimax Performance: Both the Minimax and Expectimax algorithms demonstrated commendable performance in furnishing optimal gameplay strategies. By meticulously traversing the game tree and considering potential future states, these algorithms showcased the ability to make informed decisions, thereby enabling competitive gameplay against human opponents. Their ability to anticipate adversary moves and optimize decision-making contributed significantly to their effectiveness in achieving favorable outcomes.

Machine Learning Approach: The machine learning approach exhibited promising potential in adapting to intricate gameplay patterns and evolving strategies. Leveraging neural network architectures, this approach learned optimal gameplay strategies directly from data, enabling the AI player to adapt and improve over time. However, it necessitated substantial amounts of training data and computational resources to train the models effectively. Moreover, the interpretability of the learned strategies posed challenges, as the underlying decision-making processes of the neural network models were often opaque.

Considerations for Approach Selection: The choice of approach for AI gameplay in Connect4 hinged upon several factors, including performance requirements, interpretability, and scalability. Minimax and Expectimax offered transparent decision-making processes and deterministic outcomes, making them suitable for scenarios where interpretability and reliability are paramount. On the other hand, the machine learning approach provided adaptability and scalability, enabling the AI player to learn and improve from experience. However, it necessitated careful consideration of resource constraints and trade-offs between model complexity and performance.

VI. Conclusion

The comparative analysis highlights the varying performance of different approaches in AI gameplay for Connect4. While the Minimax approach showcased robust and deterministic decision-making, the Expectimax approach introduced probabilistic elements, resulting in slightly lower but still competitive performance. The machine learning approach, while offering adaptability and scalability, exhibited moderate performance and may require further refinement and optimization to achieve consistency in gameplay outcomes. Ultimately, the choice of approach depends on factors such as interpretability, scalability, and performance requirements, underscoring the importance of understanding the trade-offs and differences in each approach.

References

[1] V. Allis, J. Allen (1988) “A Knowledge-Based Approach of Connect-Four: The Game Is Solved: White Wins,” ICG, vol. 11, no. 4, pp. 165–165, Dec. 1988.

[2] Touré, A. W. (2023). Evaluation of the Use of Minimax Search in Connect-4 & How Does the Minimax Search Algorithm Perform in Connect-4 with Increasing Grid Sizes? In Applied Mathematics (Vol. 14, Issue 06, pp. 419–427). Scientific Research Publishing,Inc. https://doi.org/10.4236/am.2023.146025

[3] Luke Kim, Hormazd Godrej, Chi Trung Nguyen (2019) “Applying Machine Learning to Connect Four”, CS229 Projects, Stanford University.

[4] “UCI Machine Learning Repository: Connect-4 Data Set.” [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Connect-4.

[5] “Connect-4 Game Dataset.” [Online]. Available: https://kaggle.com/tbrewer/connect-4.