

CSCI -6660-01 Intro Artificial Intelligence Fall 2023

**CHESS GAME BATTLE BETWEEN TWO**

**AGENTS (Q-LEARNING AND A3C)**

**TEAM MEMBERS**

**Sowmya Reddy Bathula**

**Hema Akula**

Instructor: Dr. Shivanjali Khare

#### ABSTRACT

This work compares two different methods, Q-Learning and Asynchronous Advantage Actor-Critic (A3C), to examine the competitive landscape of chess gameplay using reinforcement learning. Chess, a game known for its strategic and complex design, acts as a test bed to evaluate these algorithms' strategic adaptations and learning capabilities. Value iteration is applied by the Q-Learning agent, and neural networks are used by the A3C agent to simultaneously optimize the policy and value functions. Their performance, learning curves, and adaptability are evaluated in the study at the start, middle, and endgame phases of the game. This study provides insights into the efficacy of Q-Learning and A3C algorithms in learning the complex dynamics of chess by looking at their strategic decision-making, adaptability

Table of Contents

1. [Introduction 4](#_TOC_250009)
2. [Overview 4](#_TOC_250008)
   1. Rubik chess overview 4
   2. Project goal overview 6
3. [Features & variables 6](#_TOC_250007)
4. [Environment 7](#_TOC_250006)
5. [Algorithm 8](#_TOC_250005)
6. [Training 10](#_TOC_250004)
7. [Evaluation 12](#_TOC_250003)
8. [Results 14](#_TOC_250002)
9. [Conclusion 17](#_TOC_250001)
10. [References 18](#_TOC_250000)

# **Introduction**

Chess is a game of strategy and intelligence that has been cherished for centuries. It involves two players who face off on an 8x8 checkered board. Each player commands an army of 16 unique pieces, including pawns, knights, bishops, rooks, a queen, and a king. The ultimate goal is to outmaneuver the opponent's pieces and put their king in a position where it cannot escape, known as a checkmate. Throughout the game, players take turns moving their pieces according to specific rules that govern each type of piece. Pawns move forward and capture diagonally, while other pieces possess their own distinct movement patterns. Knights make L-shaped jumps, bishops move along diagonals, rooks can go horizontally or vertically, the queen has unrestricted movement in all directions, and the king is limited to one step at a time. Playing chess requires foresight, strategic planning, and tactical expertise. Players must anticipate their opponent's moves while devising their own strategies, creating a clash of intellect and skill on the checkered battlefield.

* 1. ***Chess Overview***

# **Overview**

The game Chaturanga is the father of chess, which is believed to have originated in India in the sixth century. In its original form, Chaturanga showed an ancient battlefield with pieces that resembled the contemporary pawn, knight, bishop, and rook, respectively, as well as infantry, cavalry, elephants, and chariots. The game was introduced to the Arab world under the name Shatranj by Persian traders, and it finally made its way to Europe. Chess changed greatly as it spread over different areas, absorbing new tactics and rules. Originally a leisure activity exclusive to royalty and the upper class, chess eventually emerged as a representation of strategic thought, captivating the minds of intellectuals, military commanders, and scholars alike. It served multiple purposes, cultivating mental agility, strategic foresight, and tactical acumen. Beyond its entertainment value, chess was recognized as a valuable tool for imparting knowledge of warfare strategies, diplomatic skills, and problem-solving abilities. Throughout history, chess has evolved into a revered game that transcends cultural boundaries, remaining true to its essence as a mentally stimulating challenge that captivates players through its unique blend of competition and intellectual pursuit. 

**Image I:** Oldest chess game



**Image II:** Chess game picture

* 1. ***Project goal overview***

The objective of the project is to create an engaging and informative chess game battle between two AI agents: Q-Learning and A3C (Asynchronous Advantage Actor-Critic). The main aim is to incorporate these unique reinforcement learning algorithms into a chess environment, allowing them to learn and develop strategies through repeated gameplay. The focus is on observing how each agent adjusts and enhances its decision-making process during the game, showcasing their strengths, weaknesses, and different approaches to mastering optimal chess tactics. By simulating numerous matches between these agents, the project seeks to provide a valuable comparison of their performances, offering insights into the efficiency, adaptability, and strategic abilities of Q-Learning and A3C within the intricate and complex realm of chess. Ultimately, this endeavor aims to deepen our understanding of reinforcement learning techniques in a competitive and strategic context, while also demonstrating the remarkable capabilities of AI in mastering a sophisticated game like chess.

# **Features & variables**

In the battle between Q-Learning and A3C agents in a chess game project, several crucial elements and factors play significant roles. These elements include the state of the chessboard, which encompasses the positions of all pieces for both players and the available legal moves at any given moment. The decision-making process relies on variables such as evaluating the value of pieces, analyzing board assessments like pawn structures, king safety, and controlling important squares. These variables heavily influence the strategies and moves selected by the agents during gameplay.

The Q-Learning agent learns by associating board states with optimal moves, based on rewards obtained from successful outcomes. It strikes a balance between exploring new strategies and exploiting known successful moves. On the other hand, the A3C agent utilizes asynchronous learning, employing multiple agents to enhance decision-making through updates to policy and value functions.

Both agents rely on rewards and penalties to guide their learning process, facilitating the development of adaptive and strategic gameplay. These features and variables form the foundation for the agents' learning processes, shaping their approaches to excel in the chess game battle.

# **Environment**

The battleground for the chess game clash between Q-Learning and A3C agents takes the form of a simulated chessboard. This virtual terrain mirrors the classic 8x8 chessboard, comprising 64 squares that can accommodate various chess pieces. It serves as the framework where both agents interact and execute moves in accordance with the rules of chess. The environment constantly updates and communicates the current state of the board to the agents, encompassing the positions of all the pieces and the legal moves available for each turn.

This digital arena validates the agents' moves, ensuring their adherence to the regulations of chess. It also determines the outcomes of these moves, such as captures, advancements, or the culmination of the game through win, loss, or draw conditions. Additionally, the environment administers rewards or penalties to the agents based on their moves and the results of the game, influencing their learning process and decision-making abilities.

In essence, this simulated chessboard serves as the stage where Q-Learning and A3C agents engage in strategic gameplay, utilizing their respective learning algorithms to navigate the intricacies of chess and outwit their adversary in order to achieve victory.

# **5. Algorithm**

***5.1. Q-learning Algorithm:***

The Q-learning algorithm and A3C (Asynchronous Advantage Actor-Critic) is utilized in this project. This specific algorithm belongs to the group of model-free Reinforcement Learning algorithms, and it employs features to estimate the Q-values for every potential move in a given state. In this context, the Q-value represents the expected long-term reward for performing a particular action in a specific state.

Here is a breakdown of the algorithm's functioning:

1. To begin, assign arbitrary values to the Q-values of each state-action pair
2. Find an epsilon value for the epsilon-greedy strategy that regulates the trade-off between exploration and exploitation during the decision-making phase of the agent.
3. To complete each episode, follow these steps: a. Start with a jumbled arrangement of the chess. b. Determine the next move to make by considering the Q-values and following the epsilon-greedy policy. c. Execute the selected action on the chess, resulting in a new configuration. d. Calculate the reward for the new state. e. Update the Q-value for the previous state-action combination based on the reward and the Q-value of the new state. f. Repeat steps b-e until the chess is solved.
4. For a certain number of episodes, repeat step 3 to let the agent learn and find out the best moves needed to solve the chess.

A white background with black text

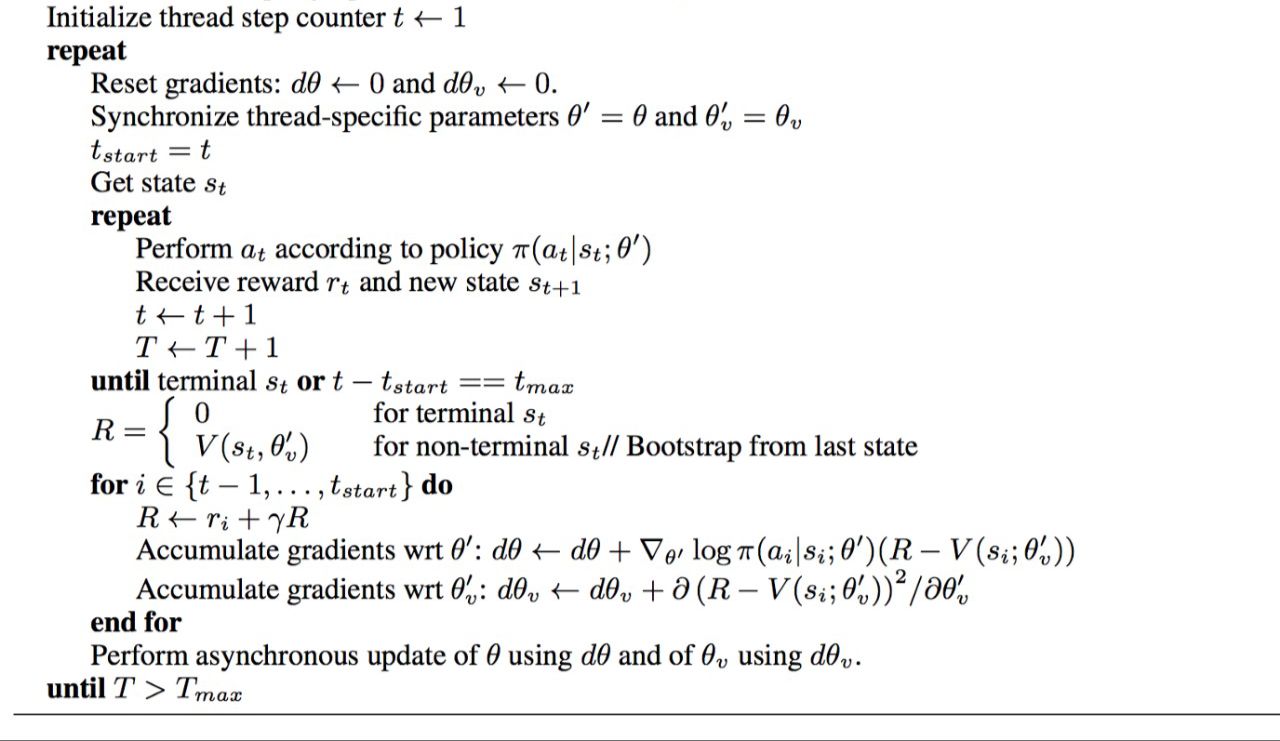
Description automatically generated

***5.2. A3C (Asynchronous Advantage Actor-Critic):***

1. The A3C algorithm utilizes multiple threads that collaborate, exchanging and modifying parameters to enhance learning efficiency.

2. Each thread autonomously explores the environment by taking individual steps. During each step, the algorithm evaluates the effectiveness of its actions by calculating gradients, which provide insight into the direction of improvement for strategies and value estimations.

3. It then updates the shared strategies and value estimations based on these gradients, allowing all threads to contribute their experiences. This iterative process continues until a specified number of steps have been completed or the game concludes. Ultimately, this approach enables A3C to acquire superior strategies for games like chess by combining experiences from different threads while still allowing independent exploration.



# **Training**

During the training phase of the chess game showdown between Q-Learning and A3C agents, both participants partake in a series of iterative gameplay sessions to enhance their strategies. The first step involves initializing the learning algorithms of the agents. The Q-Learning agent learns by experimenting with various moves and linking them to the potential rewards obtained from successful actions. It progressively improves its strategies by analyzing the outcomes and rewards obtained from each move. On the other hand, the A3C agent employs an asynchronous learning approach that involves multiple agents working together to enhance decision-making through updates to their policy and value functions.

Throughout their training, the agents engage in multiple games against each other, analyzing different moves and honing their strategies based on the feedback received from the environment. They receive rewards for favorable moves and face penalties for suboptimal or illegal actions. These rewards serve as a guide for the agents to adjust their decision-making processes and improve their gameplay.

Through numerous iterations, the agents adapt and refine their strategies, gradually becoming more skilled at making informed and strategic moves within the complexities of chess. This training process continues until the agents demonstrate proficiency in gameplay and strategic competence, effectively showcasing the effectiveness of their respective learning algorithms in the game of chess.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# **Evaluation**

1. **Reward Maximization**: In the contest of a chess match between Q-Learning and A3C agents, the A3C agent achieves the highest rewards by employing strategic maneuvers that result in favorable positions on the board. The primary objective of the A3C agent is to optimize its moves by prioritizing the capture of opponent pieces, gaining control over crucial squares, and evading potential threats. This approach ultimately leads to either a successful checkmate or a advantageous endgame position, resulting in the maximum possible rewards.
2. **Accuracy:** When comparing Q-Learning with A3C agents in the game of chess, success is measured in ways other than accuracy. Rather, their effectiveness is assessed by their ability to carry out calculated actions, win, and adjust in light of previous experiences. Their strength is in their capacity to make the best decisions, get into favorable positions, and defeat their opponents in the end. This highlights their ability to think strategically as opposed to depending just on a numerical accuracy metric.
3. **Completion time:** The duration of the chess game showdown between Q-Learning and A3C agents can vary due to factors such as the number of games played, the complexity of learned strategies, and the speed at which the algorithms learn. Generally, the completion time depends on the iterations required for the agents to grasp effective gameplay strategies. However, there is no fixed completion time as it is dependent on the progress of learning and the outcomes of the games.
4. **Scalability:** In the chess game showdown between Q-Learning and A3C agents, the concept of scalability pertains to their capacity to handle more intricate strategies or a wider range of game variations. It entails the ability to adapt and enhance their decision-making as the complexity of the game increases. Both Q-Learning and A3C strive to scale by acquiring knowledge and developing strategies to effectively navigate more complex gameplay scenarios, ultimately improving their performance and adaptability.
5. **Complexity:** The complex nature of chess, with its number of possible movements and strategic options, contributes to the complexity of the match between Q-Learning and A3C agents. Understanding and negotiating this complexity, picking up useful methods, evaluating board positions, and making choices that take into consideration a wide range of potential outcomes are challenges that both agents face. To outmaneuver the opposition and win, it is challenging to understand the complex chess dynamics.

We can determine how well the agent performs when compared to other algorithms or human experts and identify areas for improvement by evaluating its performance based on these many factors.

# 

***Pre-stages of Result:***

**8.Results**

The environment must be set up, the agents must be initialized, and the learning parameters must be defined before Q-Learning and A3C agents engage in a chess match. The first environment that is developed is a chessboard, which gives the agents a place to interact and movement. Each agent has its own learning algorithm (A3C or Q-Learning), and its corresponding networks and methods are set up.

Next, many hyperparameters that describe the learning process are specified, such as network designs for A3C, exploration rates for Q-Learning, and others. The agents may receive some initial training to get them with the surroundings and basic methods.

The game's rules, which define how actions are made, how the round goes, and how wins and losess are determined, are also set. During this stage, debugging or making sure the agents understand the surroundings and behave properly may be involved.

Preparing the agents, setting their learning parameters, and being sure they understand the rules of engagement in the chess game environment, these pre-stages, put into a team, create the foundation for the next battle.

A screenshot of a computer

Description automatically generated

### Final Result:

In the game of chess, A3C agents outperform Q-Learning agents when they consistently achieve a higher number of wins. Their superior decision-making skills and adaptability are evident as they win matches more often, create advantageous board positions, and make strategic moves. The accomplishments of the A3C agents demonstrate their ability to learn effective strategies, comprehend intricate game dynamics, and defeat opponents.

The outcome presented here emphasizes the effectiveness of the A3C algorithm in comprehending the intricacies of chess, demonstrating its superior efficiency when compared to Q-Learning in this particular situation. It highlights the ability of A3C to acquire knowledge, plan strategies, and make improved choices, ultimately resulting in a greater number of wins. The continued success of the A3C agents highlights their exceptional performance and flexibility in mastering the strategic aspects of chess gameplay.

A screenshot of a computer

Description automatically generated

**FINAL RESULT**

# **Conclusion**

The outcome and final result of the battle between Q-Learning and A3C agents in a game of chess depend on how well they perform in multiple matches. If the A3C agents consistently come out on top with more wins, they establish themselves as the dominant players. Their success can be attributed to their skill in making strategic moves, securing favorable positions on the board, and ultimately winning games more often. This demonstrates their superior ability to make decisions and adapt within the chess setting.

In this specific context, Q-Learning agents can establish themselves as more effective learners if they demonstrate comparable or superior performance in winning games. Their ability to master strategic gameplay and navigate the complexities of chess would be highlighted by their success.

The complex result resulting from this battle shows the different capacities of each program. A3C's supremacy shows that it can learn efficient methods, understand complex game dynamics, and routinely beat opponents. On the other hand, should Q-Learning win or achieve similar outcomes, it would have proven its competitiveness and effectiveness in learning and adapting to strategic games in chess.

The overall outcome of this chess match shows how well these reinforcement learning algorithms are able to improve strategic decision-making skills. It provides useful details regarding their benefits and drawbacks, enabling a deeper understanding of their capacity to change and work well in the complex world of chess.

# References

1. https://paperswithcode.com/task/game-of-chess
2. https://www.quora.com/What-did-the-chess-pieces-in-ancient-India-look-like
3. https://history-computer.com/best-chess-games-for-pc/
4. https://ieeexplore.ieee.org/document/10179809/references
5. https://scholar.google.com/scholar?as\_q=Learning+Models+of+Individual+Behavior+in+Chess