Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

Technical Documentation Discovery Lab Global (DLG) Team Quantum Blue

By

Matthew Goldsberry

Jacob Parento

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

Executive Summary:

In the realm of artificial intelligence, our team embarked on the Smart Pong project armed with little prior AI knowledge but a solid foundation in Python. Under the guidance of Dr. Williams and his program at Discovery Lab Global, and with the invaluable support of ChatGPT, we navigated through the intricacies of AI development, culminating in the creation of an intelligent Pong-playing agent. This executive summary provides a nuanced overview of our transformative journey, detailing our approach, challenges, and remarkable outcomes.

Building a solid foundation for our project, the introductory sections, comprising the “Introduction”, “Key Concepts”, and “Environment Setup”, served as crucial pillars. The introduction not only set the stage for our project but also painted a comprehensive picture, providing the context for the entire endeavor. Simultaneously, the key concepts section delved into the foundational principles, offering an understanding of machine learning, reinforcement learning, neural networks, q-learning, and policy gradients. Furthermore, the environment setup section played a pivotal role in highlighting the prerequisites essential for project implementation, such as the integrated development environment (IDE), Python, and requisite libraries. Together, these sections established the groundwork necessary for the subsequent phases of our project.

In the "Pong Environment Setup" section, our focus shifted to the practical implementation of our project's foundation. Here, we undertook the meticulous task of constructing a custom Pong environment, leveraging frameworks such as Gym. With precision and attention to detail, we navigated through the intricacies of creating a bespoke Pong class, dissecting the code piece by piece. Utilizing the Gym framework, we ensured that our custom environment provided the optimal conditions for training and evaluating our intelligent Pong-playing agent. This section not only encapsulated the technical nuances of our project but also laid the groundwork for subsequent phases, setting the stage for the practical implementation of our AI model.

The subsequent sections, "AI Development", and "Training," represent the heart of our project, where theoretical concepts metamorphosed into a functional and intelligent Pong-playing agent. In the "AI Development" phase, we delved into the intricacies of the code constructed for the agent, dissecting its components, understanding the underlying mechanisms, and crafting a dynamic game loop. This game loop became the heartbeat of our project, facilitating seamless interaction between the AI and the custom Pong environment. It not only showcased our technical prowess but also highlighted the meticulous crafting of a custom Pong environment class, which served as the framework for our intelligent agent.

The "Training" section, a critical aspect of our project, chronicled our experiences and insights gained during the training process. It emphasized the nuanced approach we adopted, challenging the conventional belief that more extensive training necessarily yields superior results. Our findings underscored the significance of the quality of training over the quantity of episodes, a revelation that contributed to the efficiency of our model. This section not only provided a glimpse into our practical experiences but also conveyed the importance of thoughtful and strategic training in developing a high-performing AI agent. Together, these sections represent the culmination of our efforts, transforming theoretical knowledge into a tangible and effective solution for our Smart Pong project.

The "Challenges and Solutions" section encapsulates the hurdles we encountered during the creation of our intelligent Pong-playing agent and the adept strategies we employed to overcome them. This segment serves as a testament to our team's resilience and problem-solving acumen. From algorithmic complexities to technical hitches, each challenge became an opportunity for innovation and growth. This section not only highlights our ability to navigate obstacles but also underscores the adaptability and collaborative spirit that defined our approach throughout the project.

In the "Results and Performance" section, we confronted the challenge of definitively determining the superior agent without the luxury of real game simulations. To address this, we implemented a sophisticated solution by enhancing our code to allow agents to engage in simulated games, each lasting until 21 points in a game of Pong. This simulation not only provided a comprehensive performance metric but also allowed us to track and compare the records and scores of competing agents.

In a surprising revelation, our minimally trained agent emerged as the superior performer. It not only outperformed the other agent in average opponent score but also demonstrated remarkable consistency, winning an astonishing 2,500 games consecutively without a single loss before we concluded the simulation. This outcome underscored the significance of our strategic approach to training, emphasizing the quality of training episodes over sheer quantity. The "Results and Performance" section thus stands as a testament to the effectiveness of our methodology, showcasing the prowess of our minimally trained agent in achieving unparalleled success in the simulated Pong games.

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

Following the comprehensive discussion on the Smart Pong project, the document transitions into a detailed Technical Analysis of the blog post "Pong from Pixels." This blog post provides a profound exploration of training an AI agent to play Pong using pixel inputs, emphasizing the significance of convolutional neural networks (CNNs) and reinforcement learning in achieving superior performance. The author highlights the key role of deep learning in processing pixel information and extracting meaningful features, leading to the development of an adept Pong-playing agent.

Continuing our exploration, the document delves into a Technical Analysis of "Attention is All You Need." This influential paper introduces the Transformer architecture, a paradigm shift in natural language processing and sequence-to-sequence learning. The key innovation lies in the self-attention mechanism, allowing the model to weigh different parts of the input sequence differently, leading to enhanced performance and efficiency in various tasks.

The collaboration with ChatGPT, outlined in "How we used ChatGPT," played a pivotal role in enriching our project experience. Leveraging ChatGPT, an advanced language model, enhanced our project's depth and sophistication. The model proved instrumental in generating insightful responses, offering valuable suggestions, and contributing to the overall refinement of our project.

In the "Prompt Engineering" section, we take a deep dive into the crucial art of effectively formulating prompts. This segment is a testament to the strategic thinking that underpinned our interactions with ChatGPT, highlighting the importance of crafting precise and context-aware prompts to elicit meaningful and desired responses. The section provides valuable insights into optimizing the engagement with ChatGPT, showcasing the meticulous approach we adopted to extract maximum utility from this powerful language model.

Then emerges with the "Introduction to AI with Smart Pong (High School AI Program)" section within the technical document. This segment introduces a meticulously structured high school AI program, seamlessly weaving theory and practical application around the Smart Pong project. Each component is dissected, offering a comprehensive understanding for students. Serving as a bridge from theory to practice, this educational initiative not only enriches our project but also nurtures the growth of the next generation of AI enthusiasts.

In conclusion, our Smart Pong project exemplifies a journey from limited knowledge to an accomplished development team. The combination of theory, practical implementation, collaboration with advanced models, and insightful analyses reflects a milestone in our collective learning and growth in the field of artificial intelligence.

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

Acknowledgements:

We extend our sincere gratitude to the individuals and organizations whose support and contributions were instrumental in the successful completion of the Smart Pong project.

First and foremost, we appreciate Dr. Williams, the Program Director, for providing invaluable guidance and fostering an environment of innovation throughout the development process.

Our thanks go to Discovery Lab Global for their continuous support and resources, which played a crucial role in the realization of the Smart Pong project. The collaborative and dynamic atmosphere at Discovery Lab Global provided fertile ground for creativity and problem-solving.

A special acknowledgment goes to our teammates at Quantum Blue, with Matthew Goldsberry serving as the Team Lead, and Jacob Parento. Their expertise, dedication, and collaborative spirit were pivotal in overcoming challenges and achieving the project's goals.

Lastly, we want to express our appreciation to ChatGPT, a valuable resource throughout the project. ChatGPT's ability to provide insights, suggestions, and assistance enhanced the overall development process and added a unique dimension to our problem-solving approach.

This project was truly a collaborative effort, and we are grateful to have had the opportunity to work together. Each person mentioned played a vital role in the success of the Smart Pong project, and we are thankful for the enriching experience they provided.

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

Table of Contents

Handling Complex Mappings:

One of the strengths of neural networks lies in their capacity to handle complex mappings. RL tasks often involve intricate relationships between the current state, chosen actions, and received rewards. Neural networks excel at discerning these complex patterns, providing the RL agent with the ability to navigate the nuanced dynamics of Smart Pong gameplay.

Integration with RL:

In the Smart Pong project, neural networks seamlessly integrate with reinforcement learning. The RL agent, responsible for controlling the paddle, relies on the computational prowess of neural networks to understand and learn from the environment. As the agent interacts with the game, the neural network refines its internal representation, mapping states to actions and optimizing its decision-making strategy based on the received rewards.

In essence, neural networks form the backbone of the RL framework in Smart Pong, contributing to the adaptability and learning capabilities of the AI agent. Through their ability to capture and generalize complex mappings, neural networks enhance the RL agent's capacity to make informed decisions and continually improve its gameplay over time.

2.04 Q-learning

Introduction to Q-learning:

Definition:

Q-learning represents a fundamental model-free reinforcement learning algorithm with significant applications in optimizing decision-making processes. Q-learning enables the agent to autonomously learn and adapt its strategy over time by estimating the optimal action-value functions.

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

Q-value:

Central to Q-learning is the concept of Q-values, denoting the expected cumulative reward associated with a specific action in a given state. These values encapsulate the agent's learned knowledge, guiding it to make informed decisions in the dynamic environment of Smart Pong. The Q-value for a state-action pair (s, a) is denoted as Q(s, a).

Q-learning Process:

Exploration and Exploitation:

Exploration:

Q-learning initiates with exploration, where the AI deliberately selects actions to observe their consequences. This phase is critical for the agent to build an understanding of the environment and uncover the underlying dynamics. During exploration, the AI may employ strategies like epsilon-greedy, favoring exploration with a certain probability.

Exploitation:

Exploitation follows exploration and involves leveraging the learned Q-values to make decisions aimed at maximizing cumulative rewards. The agent, equipped with accumulated knowledge, strategically chooses actions based on the current estimates of Q-values.

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

Updating Q-values:

Bellman Equation:

The crux of Q-learning lies in the iterative update of Q-values based on the Bellman equation. This equation captures the recursive relationship between the Q-value of a current state-action pair and the expected future rewards. The update rule is represented as:

$$Q(s, a) \leftarrow (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (R + \gamma \cdot \max_{a'} Q(s', a'))$$

- Q(s,a): Current Q-value for state-action pair (s, a).
- α: Learning rate, determining the impact of new information on the current estimate.
- R: Immediate reward for taking action in a state s.
- γ: Discount factor, indicating the importance of future rewards.
- $\max_{a'} Q(s', a')$: Maximum Q-value for the next state s′ and all possible actions a′.

Temporal Difference:

Q-learning employs a temporal difference approach, comparing the difference between the current estimate and the updated estimate, then incorporating this difference with the learning rate to refine the Q-value.

Q-learning Example:

Initialization:

At the onset, Q-values are initialized, often with small random values. The AI's understanding of the environment is limited, and Q-values reflect this uncertainty.

Exploration and Learning:

As the Smart Pong AI explores the game, it observes the consequences of actions and updates Q-values accordingly. The agent learns to associate actions with rewards and refines its understanding of optimal strategies.

Exploitation:

As Q-values converge over multiple iterations, the AI transitions towards exploitation, leveraging the accumulated knowledge to make strategic decisions during gameplay.

Convergence:

With continued exploration and exploitation, Q-values converge to accurate representations of the optimal action-value functions. The AI, armed with learned Q-values, navigates the game environment with refined and strategic decision-making.

Integration with Smart Pong:

In the Smart Pong project, Q-learning seamlessly integrates with the AI's decision-making process. As the agent controls the paddle, Q-values guide actions in response to changing game states, ensuring the AI adapts its strategy based on learned knowledge.

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

4.03 Environment Structure

Embarking on the construction of our custom Pong environment requires a thorough understanding of its internal architecture. In this section, we'll dissect the components that form the backbone of our environment, from the initialization of critical parameters to the continuous updating of the game's visual representation. Let's delve into the inner workings of our Custom Pong Environment Class and explore the intricacies that make it a dynamic and responsive arena for reinforcement learning.

Overview of the Custom Pong Environment Class:

The Custom Pong Environment Class serves as the backbone of our virtual Pong world, encapsulating the game's dynamics and interactions. This class achieves a seamless integration between the reinforcement learning agent and the game environment by orchestrating critical functionalities. Through its initialization method, it establishes the initial conditions of the game, defining the action and observation spaces, initializing paddle and ball positions, and configuring other essential parameters. The Pygame initialization method ensures a visually immersive experience by setting up the graphical interface, while the display update method dynamically renders the evolving state of the game. The step function manages the core logic of the environment, handling actions, updating the state, and providing feedback to the RL agent. Additionally, the reset function ensures a consistent starting point for each episode, facilitating iterative learning. In essence, the Custom Pong Environment Class acts as the conduit through which the RL agent perceives, interacts, and learns within the dynamic environment, laying the groundwork for an effective and engaging learning experience.

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

5.04 Game State Representation

In Smart Pong's journey towards intelligent gameplay, the representation of the game state is a pivotal aspect. This section delves into the intricacies of the get_state() function, unraveling the methodology behind calculating the game state. As we explore the eight distinct states defined within this function, each with its unique significance, readers will gain insights into how Smart Pong captures and categorizes the nuances of the game environment. This in-depth understanding forms the foundation for the AI's decision-making process, allowing it to navigate the Pong arena with precision and adaptability.

get_state() Function:

The get_state() function is a pivotal component in Smart Pong's Q-learning algorithm, translating the intricate ball and paddle dynamics into a meaningful representation for the AI:

```
def get_state():

    # Calculate the vertical distance between the ball and the paddle
    difference = env.get_ball_position('y') - env.get_your_paddle_position()

    # Determine the game state based on the calculated difference
    if difference <= -env.get_paddle_height():
        return 0
    elif difference <= -env.get_paddle_height() // 2:
        return 1
    elif difference < 0:
        return 2
    elif difference == 0:
        return 3
    elif difference < env.get_paddle_height() // 2:
        return 4
    elif difference < env.get_paddle_height():
        return 5
    elif difference < env.get_paddle_height() * 3 / 2:
        return 6
    else:
        return 7
```

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

Understanding Python's '//' Operator

The '//' operator in Python denotes floor division, where the result of the division is rounded down to the nearest whole number. In the context of difference <= -env.get_paddle_height() // 2, it ensures that the calculated difference is less than or equal to half of the player's paddle height. This operator is particularly useful when you need to ensure integer division, disregarding any fractional part.

Calculation of Game State:

The vertical difference (difference) is computed by subtracting the player's paddle position retrieved using env.get_your_paddle_position() from the ball's position along the y-axis obtained through env.get_ball_position('y'). This dynamic calculation ensures a real-time understanding of the spatial relationship between the ball and paddle.

Defined States and Explanations:

- 0: Ball is well above the paddle:

  Indicates that the ball is positioned well above the middle of the paddle, more than one paddle height. This is determined by the condition difference <= -env.get_paddle_height().

- 1: Ball is somewhat above the paddle:

  Implies that the ball is positioned above the middle of the paddle by less than one paddle height but at least half the paddle height. This is determined by the condition difference <= -env.get_paddle_height() // 2.

- 2: Ball is in the upper half of the paddle:

  The ball is positioned above the middle of the player's paddle, as indicated by the condition difference < 0. Since it is after state 0 and 1 for it to be state 2 it must be above the middle of paddle by less than half the paddle height.

- 3: Ball is at the same height as the middle of the paddle:

  This state denotes that the ball and paddle share the same vertical level, with the ball directly aligned with the middle of the paddle (difference == 0).

- 4: Ball is in the lower half of the paddle:

  Signifies that the ball is positioned below the middle of the paddle by less than half a paddle height. This is determined by the condition difference < env.get_paddle_height() // 2.

- 5: Ball is somewhat below the paddle:

  Indicates that the ball is positioned below the middle of the player's paddle by at least half a paddle height but less than one paddle length (difference < env.get_paddle_height()).

- 6: Ball is moderately below the paddle:

  This state denotes that the ball is positioned more than a paddle height below the middle of the player's paddle but less than one and a half paddle heights (difference < env.get_paddle_height() * 3 / 2).

- 7: Ball is significantly below the paddle:

Serves as a state for any positions not explicitly covered in the defined states, indicating that the ball is significantly below the paddle.

Q-Learning Implications:

The precise categorization of states provides the Q-learning algorithm with a clear understanding of the ball's relative position to the player's paddle. This nuanced representation enables the AI to make informed decisions, optimizing its gameplay strategy based on the current spatial dynamics of the game. The values for the ball and paddle positions are retrieved using getter functions (env.get_ball_position('y') and env.get_your_paddle_position()), ensuring a dynamic and accurate calculation of the game state.

5.05 Action Selection

In the realm of Smart Pong's artificial intelligence, the choose_action() function plays a pivotal role in determining the agent's actions. This section delves into the intricacies of action selection, shedding light on the epsilon-greedy policy employed by the AI. Understanding how the AI balances exploration and exploitation through this policy is crucial for grasping the decision-making process within the game environment.

choose_action() Function:

The choose_action() function serves as the maestro of the AI's decision-making process, deftly applying the epsilon-greedy policy:

```
def choose_action(state):
    if np.random.rand() < EPSILON:
        # Exploration: With probability epsilon, choose a random action
        return np.random.choice(NUM_ACTIONS)
    else:
        # Exploitation: choose the action with the highest Q-value
        return np.argmax(q_table[state])
```

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

Epsilon-Greedy Policy Demystified:

- At its core, the epsilon-greedy policy governs the AI's strategic decision-making. The choose_action() function encapsulates this policy, offering a nuanced approach to navigating the trade-off between exploration and exploitation.

Exploration (Random Action) with np.random.rand() Implementation:

- The line if np.random.rand() < EPSILON: embodies the exploration phase. With a probability of EPSILON, the AI ventures into uncharted territory by selecting a random action using np.random.choice(NUM_ACTIONS) which returns a value of 0, 1, or 2.

Exploitation (Action with Highest Q-value) - np.argmax:

- The exploitation phase, activated when the condition is not met (probability 1 - EPSILON), relies on the np.argmax(q_table[state]) expression. Here, np.argmax returns the index of the maximum Q-value from the Q-table for the given state which happens to be 0, 1, or 2 (the different action values). In the context of Q-learning, the Q-value represents the expected future cumulative reward for taking a specific action in a particular state. By selecting the action with the highest Q-value, the AI strategically leverages its learned knowledge to make decisions that maximize long-term rewards. Notably, the given state determines the row in the Q-table that is examined, and the action with the highest Q-value in that row is selected.

Understanding the nuanced dynamics of np.argmax and its role in choosing actions based on Q-values provides a key insight into the AI's decision-making process. This section delves into the intricacies of exploration and exploitation, inviting readers to explore the deeper dimensions of Smart Pong's Q-learning algorithm.

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

6.02 Training Strategies

In the dynamic landscape of developing intelligent agents, the training phase serves is where raw capabilities transform into refined expertise. In this section, we navigate through the intricacies of our training strategies, illuminating the deliberate decisions and methodologies that shape the learning trajectory of our agent. Two distinct approaches unfold—each a unique avenue toward enhancing the agent's proficiency. Approach 1 entails an extensive training regimen exceeding 5,000 episodes, driven by specific goals and a commitment to exhaustive learning. Conversely, Approach 2 adopts a strategic early stopping mechanism, ceaselessly refining the agent until a remarkable 96% success threshold is reached. As we unravel the purpose, challenges, and outcomes of each approach, a comprehensive understanding emerges, spotlighting the essence of training in sculpting our agent's competence. Join us on this journey as we delve into the intentional choices that mold our agent's learning process and define its prowess in navigating complex environments.

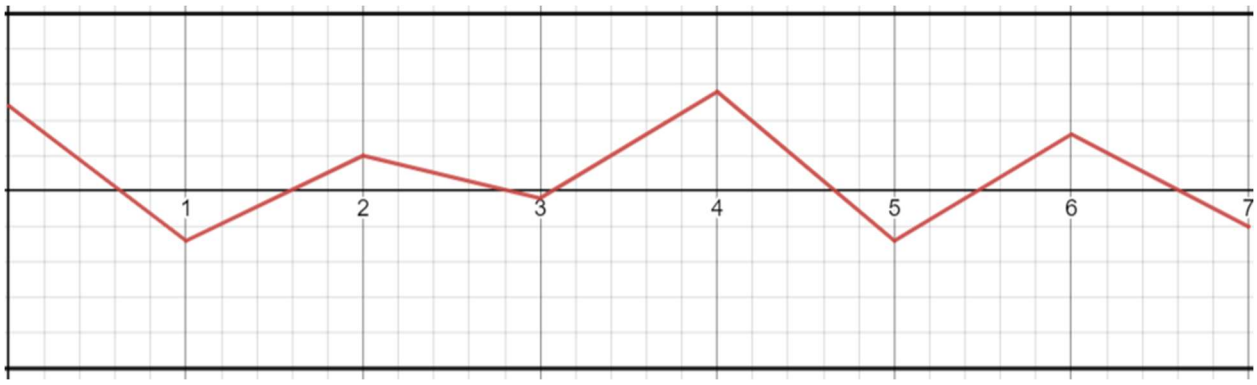
Approach 1: Extended Training (5k+ episodes)

Purpose and Goals:

The decision to embark on an extended training regimen exceeding 5,000 episodes was driven by the intention to expose the agent to a myriad of scenarios, fostering a comprehensive understanding of the environment. The overarching goal was to allow the agent to navigate diverse situations, encouraging adaptability and a nuanced response to varying stimuli. This extended training aimed to cultivate a robust decision-making framework by immersing the agent in a wide spectrum of challenges, thereby enhancing its capacity to handle unforeseen scenarios.

Results and Observations:

Following the extensive training sessions, a wealth of outcomes and observations were gleaned. The agent exhibited a noticeable trajectory of improvement, showcasing an evolving understanding of the environment. However, the results also revealed a notable oscillation in the agent's success rate. The agent would oscillate between periods of relative success and instances of less optimal performance. This pattern of inconsistency indicated a need for further analysis and refinement in the training approach.



This is a visual representation of the average reward of batches (25 episodes) from episodes 1601-1800. The range of rewards spans from 1 (win) to -1 (loss). As you can see there is a lot of inconsistency and no sign of convergence, and this is how it would look for any section of episodes you would choose in the training set.

Challenges:

The prolonged training period presented several challenges, chief among them being the agent's struggle to achieve a consistent level of success. Despite exposure to a diverse range of scenarios, the agent demonstrated difficulty in maintaining a sustained high-performance rate. This challenge prompted a closer examination of the training dynamics, leading to the realization that the agent's learning process was characterized by fluctuations between states of effectiveness and periods of suboptimal decision-making.

Approach 2: Early Stopping at 96% Success

Decision to Stop Early:

In contrast to the extended training approach, Approach 2 incorporated a strategic early stopping mechanism. The decision to halt training was grounded in the agent's remarkable success, achieving a 96% win rate after only 125 episodes. This decision was driven by the recognition that further training might lead to diminishing returns and potentially perpetuate the oscillatory behavior observed in the extended training approach. The code modification introduced for early stopping aimed to refine the agent's learning trajectory, steering it away from prolonged and unnecessary training phases.

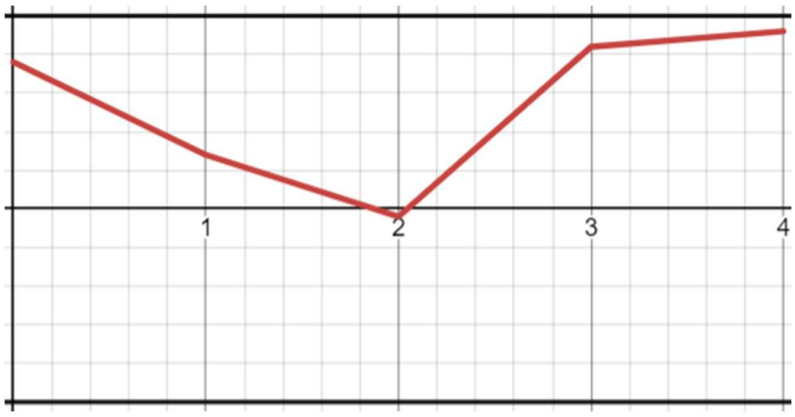
Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

Significance of 96% Success:

The choice of a 96% success threshold held specific significance in determining when to conclude the training process. This threshold was deemed sufficient to signify a highly competent agent, capable of consistently outperforming the environment. By setting this benchmark, the goal was to strike a balance between achieving a high success rate and avoiding overfitting or persistent oscillations in performance.

Results and Observations:

The rapid convergence of the agent to the 96% success threshold following early stopping showcased the effectiveness of this approach. In a mere 125 episodes, the agent demonstrated a proficiency that surpassed the extended training approach. The success trajectory was notably steady, avoiding the oscillatory patterns observed in the prolonged training scenario. This outcome provided valuable insights into the potential advantages of strategically halting training based on performance criteria.



This is a visual representation of the average reward of batches (25 episodes) from episodes 1-125. The range of rewards spans from 1 (win) to -1 (loss). As you can see after a short dip the average reward jumps up and converges to the 96 percent success rate.

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

8.02 Results

This section provides a comprehensive analysis of the intelligent agent's performance in the game of Pong. We delve into key performance metrics, including the win-loss record, individual game scores, and success rate, offering quantitative insights into the agent's strategic effectiveness. Furthermore, a detailed comparison between two distinct training versions sheds light on the adaptability and proficiency of the agent under varied training conditions. The subsequent analysis of results aims to uncover factors influencing performance and general trends observed during gameplay.

Performance Metrics:

In evaluating the performance of our intelligent agent in the game of Pong, we rely on key metrics that provide a comprehensive understanding of its strategic capabilities and overall effectiveness.

1. Win-Loss Record

The win-loss record serves as a fundamental indicator of the agent's success in competitive gameplay. It represents the number of games won versus the number of games lost, providing a clear measure of the agent's proficiency in achieving the primary objective of outplaying the opponent. A comprehensive examination of the win-loss record goes beyond measuring the agent's success; it serves as a nuanced lens through which we gain insights into the agent's adaptability and strategic decision-making throughout gameplay. This analysis allows us to explore the agent's ability to navigate dynamic scenarios, adapt to varying opponent strategies, and make informed decisions, providing a deeper understanding of its overall performance in the competitive Pong environment.

2. Individual Game Scores

Beyond the overall win-loss record, understanding the individual game scores contributes to a finer analysis of the agent's performance. This metric provides insights into the agent's consistency and effectiveness in each game, shedding light on its ability to maintain high scores, respond to opponent moves, and exhibit strategic prowess on a per-game basis.

3. Success Rate

The success rate, in the context of our training procedures, is a crucial performance metric focused on the agent's learning capabilities. Unlike the win-loss record, which gauges the agent's success in actual gameplay, the success rate reflects the agent's ability to learn and improve during the training phase. This metric is particularly relevant to the training section, where we explore the agent's capacity to adapt and refine its strategies over time. A higher success rate indicates effective learning and adaptation, showcasing the agent's capacity to evolve in response to the training environment.

Comparison between Training Versions:

In this section, we conduct a detailed comparison between two distinct training versions of our intelligent agent, each revealing unique aspects of its learning and adaptation capabilities.

1. Extensive Training Version (5,000+ episodes)

a. Win-Loss Record: 864-1

The Extensive Training Version, involving an extensive training regimen of over 5,000 episodes (single-point matches), exhibited a remarkable win-loss record of 864 wins to only 1 loss. This outstanding performance underscores the agent's capacity to consistently outplay opponents, demonstrating a high level of proficiency in the Pong environment. The exceptional win-Loss record suggests robust strategic decision-making and adaptability, as the agent successfully navigated various gameplay scenarios over the course of an extended training period.

b. Individual Game Scores

Examining individual game scores in the Extensive Training Version revealed consistent opponent scores around 7, with slight variations of around ± 3. There were instances of lower opponent scores, and occasional spikes close to 21, with one notable game reaching the maximum score, 21, when the agent experienced its sole loss.

c. Observations during Training

Throughout the training process, we observed notable oscillations in the success rate. This dynamic behavior suggests that the agent continuously refined its strategies and adapted to changing conditions during training. The oscillations may indicate periods of exploration and exploitation, showcasing the agent's ability to strike a balance between trying new approaches and exploiting known successful strategies.

2. Shorter Training Version (Stopped after certain success)

a. Win-Loss Record: 865-0

The Shorter Training Version, strategically halted after achieving a predefined success rate goal of 96%, demonstrated an exceptional win-loss record of 865 wins without a single loss. It was also allowed to run over 2,500 games total and did not lose one, giving it a total win-loss record of 2,500-0. This decisive performance underscores the agent's rapid learning and adaptability within a shorter training timeframe. The agent quickly reached its success rate goal, showcasing efficiency in strategy acquisition.

b. Individual Game Scores

Examining individual game scores in the Shorter Training Version revealed opponent scores hovering around 4, with occasional variations of ±2. There were sporadic spikes in the low teens, showcasing the agent's adaptability to varying opponent strategies. This pattern indicates a more controlled and consistent performance in individual games.

c. Observations during Training

Remarkably, the Shorter Training Version achieved its success rate goal of 96% in just 125 episodes, highlighting its accelerated learning curve. After a brief dip, the agent swiftly recovered and consistently improved, surpassing the initial success rate target. This efficient training period suggests a focused learning trajectory, minimizing the risk of overfitting and oscillation while achieving high-performance outcomes.

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

Analysis of Results:

In this section, we conduct a comprehensive analysis of the results obtained from the Extensive Training Version and the Shorter Training Version, aiming to identify factors influencing performance and elucidate general trends observed during gameplay.

1. Factors Affecting Performance

a. Extensive Training Version

The performance of the Extensive Training Version, encompassing over 5,000 episodes, can be attributed to its prolonged exposure to diverse gameplay scenarios. The extended training period allowed the agent to explore a wide range of strategies, refine its decision-making processes, and adapt to varying opponent behaviors. However, the observed oscillations in the success rate suggest a potential challenge in maintaining consistent performance, with periods of exploration and exploitation influencing the agent's learning trajectory.

b. Shorter Training Version

Contrastingly, the Shorter Training Version demonstrated a highly efficient learning curve, achieving its success rate goal of 96% in just 125 episodes. The decision to halt training after reaching a specific success rate mitigated the risk of overfitting and prolonged oscillations, ensuring a focused learning trajectory. This strategic approach resulted in a rapid and sustained high-performance outcome, showcasing the effectiveness of a more concise training period.

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

2. General Trends Observed

a. Adaptability and Consistency

Both training versions exhibited notable adaptability, as evidenced by their impressive win-loss records. However, the Shorter Training Version stood out for its ability to consistently outperform opponents, maintaining a flawless win-loss record of 865-0 and continuing the flawless record for over an impressive total of 2,500 games before ending the simulation. The controlled individual game scores in the Shorter Training Version, with occasional spikes into the low teens, suggest a higher level of adaptability and strategic consistency compared to the Extensive Training Version.

b. Efficiency in Learning

The Shorter Training Version's accelerated learning curve, achieving the success rate goal efficiently, indicates a focused and effective learning trajectory. The agent rapidly improved its performance, demonstrating the capacity to quickly adapt and refine strategies in response to the training environment. This efficiency contrasts with the Extensive Training Version's prolonged training period, emphasizing the benefits of a targeted approach to training duration.

Conclusion:

The analysis of results strongly suggests that the Shorter Training Version outperformed the Extensive Training Version in terms of efficiency, adaptability, and consistency. The decision to conclude training after achieving a predefined success rate proved effective in producing a highly capable and consistently successful intelligent agent in the dynamic game of Pong.

Dr. Rob Williams, DLG Executive Director
Email: rob.williams@discoverylabglobal.com
Website: https://www.discoverylabglobal.com/

9.01.05 Final Skills Assessment

1. Professional Skills

After the completion of the project, my professional skills experienced substantial growth. I honed my written communication abilities through daily and weekly email summaries, learning to articulate ideas clearly in various documents, including the one you're currently reading. Crafting detailed meeting summaries became second nature, providing valuable resources for future reference. On the oral communication front, my skills saw significant improvement. Weekly progress presentations and diverse sessions, ranging from content presentations to tutorials, allowed me to refine my presentation skills. Additionally, serving as the team lead enhanced my communication prowess, acting as a link between Dr. Williams and the team, effectively relaying information in both directions.

2. AI Knowledge

Upon concluding the project, a better sense of enhanced AI understanding has settled in. As I delved into various articles and documents, my knowledge expanded across different facets of AI, encompassing generative AI, policy gradients, and Q-learning. The confidence gained in understanding Q-learning was particularly notable, especially as I successfully implemented it in the Pong environment. This journey has transformed my initial uncertainties into a newfound assurance and mastery in different AI concepts. Embracing the learning process, exploring diverse resources, and actively applying newfound knowledge are essential principles that have contributed to this enriching experience in the realm of AI.

3. Python Skills

While my Python skills didn't experience as dramatic a surge as my AI knowledge, there was still a noticeable improvement. Examining code snippets from articles, my team, and engaging with ChatGPT allowed me to glean several new approaches and methods in Python programming. Additionally, I acquired knowledge about a couple of new libraries—Pygame and pickle—and expanded my understanding of NumPy. This incremental growth in Python proficiency, coupled with the broader AI insights gained, reflects the holistic learning experience fostered by exploring diverse resources and embracing the collaborative aspect of the project.

4. ChatGPT

My ChatGPT skills has undergone significant improvement, evident in the evolution of my prompts. Initially, my queries were brief and lacked specificity, but over time, they have grown longer and more detailed. The learning process has equipped me with strategies to effectively leverage ChatGPT for distinct tasks, including dissecting articles, debugging, and composing documents. Consequently, I now extract information from ChatGPT more efficiently and with markedly improved quality. This transformation in my interaction with ChatGPT showcases not only the platform's capabilities but also the strides made in mastering its utility for diverse purposes.

5. Document Writing

My proficiency in document writing has seen substantial growth throughout the project. The process of creating this technical document served as a valuable learning experience in effectively harnessing ChatGPT for document generation. As I navigated through the document, my approach to interacting with ChatGPT evolved, driven by a quest for increased effectiveness and efficiency in extracting desired information. This iterative refinement in my utilization of ChatGPT has not only honed my document writing skills but also underscored the dynamic nature of learning and adapting to make the most out of the tools at my disposal.

9.01.06 Conclusion

In embarking on the Smart Pong project, my initial aim was to delve into the realm of artificial intelligence, hoping to expand my knowledge in a field I only knew at the surface level. This testimony reflects on my journey, encompassing the evolution of my skills, the challenges faced, and the transformative impact this experience has had on my understanding of AI.

The Smart Pong project has been nothing short of transformative. As a novice with minimal AI knowledge, I navigated the complexities of Q-learning, generative AI, and policy gradients. The challenges faced in Python coding, particularly in creating a Pong environment and integrating Pygame, were formidable but ultimately instrumental in honing my coding skills. ChatGPT emerged as an invaluable ally, aiding not just in problem-solving but also in enhancing my understanding of AI concepts and writing technical documentation. Through this project, my skills evolved, and I stand today with a better understanding of AI, fortified Python capabilities, and an adeptness in utilizing tools like ChatGPT.