Enhancement Two: Algorithms and Data Structure

By Sam Blanton

CS-499

July 29, 2024

Narrative for ePortfolio Assignment

Artifact Description

The artifact I have selected for inclusion in my ePortfolio is a reinforcement learning project focused on training an agent using the Q-learning algorithm. This project was originally created during a prior course in the Computer Science program. The artifact includes a Python implementation of a Treasure Maze game where an agent learns to navigate the maze and find the treasure using Q-learning. As part of my enhancements, I upgraded the algorithm to use Double Q-learning to improve the training performance and stability of the agent.

Justification for Inclusion

I selected this artifact for my ePortfolio because it effectively demonstrates my skills and abilities in algorithms and data structures, specifically in the domain of reinforcement learning. The project showcases my ability to implement complex algorithms, understand their theoretical foundations, and apply them to practical problems. The transition from Q-learning to Double Q-learning required a deep understanding of the underlying concepts and the ability to modify and optimize the existing codebase. These skills are crucial for any computer scientist and are highly valued in both academic and professional settings.

Enhancement Process and Challenges

GPU Utilization Issue:

One of the primary challenges I encountered was ensuring that the model training utilized my GPU. Initially, I noticed that the Apporto computer provided during the class had significantly faster processing capabilities compared to my hardware. This discrepancy highlighted the importance of leveraging GPU acceleration for intensive computations. After configuring TensorFlow to use my GPU, I observed some improvements, but optimizing the code further remains a priority.

Learning Double Q-learning:

To effectively implement Double Q-learning, I immersed myself in numerous videos and examples. This extensive research phase was critical for building a solid understanding of both Q-learning and Double Q-learning. Double Q-learning addresses the overestimation bias found in Q-learning by maintaining two separate value estimates. This enhancement required dissecting and modifying the existing GameExperience.py to accommodate the new algorithm.

Code Modification:

Altering the GameExperience.py file to implement Double Q-learning was a significant challenge. I had to ensure that the two sets of Q-values were updated correctly and that the action selection process

leveraged both sets appropriately. This modification required careful attention to detail to maintain the integrity of the training process.

Implementation in Main Function:

After modifying GameExperience.py, I integrated the changes into the main function. This step involved ensuring that the training loop correctly utilized the new Double Q-learning method. Testing and debugging this implementation was a meticulous process, requiring several iterations to achieve the desired performance.

Optimization for Hardware:

Despite successfully implementing Double Q-learning, training the model on my hardware remains slow. This experience has underscored the need for further research into optimization techniques, both in terms of code efficiency and hardware utilization. I plan to continue exploring ways to enhance the training speed and performance on my system.

Learning and Reflection

The process of enhancing and modifying this artifact was highly educational. It reinforced the importance of understanding both the theoretical and practical aspects of algorithms. I learned the intricacies of reinforcement learning and the specific advantages of Double Q-learning over traditional Q-learning. Additionally, I gained valuable experience in troubleshooting and optimizing machine learning code for different hardware configurations.

Course Outcomes

Algorithmic Solutions: The primary focus of this enhancement was to design and evaluate computing solutions that solve a given problem using algorithmic principles. By transitioning from Q-learning to Double Q-learning, I demonstrated my ability to apply computer science practices and standards to manage the trade-offs involved in algorithm design choices.

Innovative Techniques and Tools: Implementing Double Q-learning and optimizing the code to run efficiently on my hardware required the use of well-founded and innovative techniques, skills, and tools in computing practices. This project showcases my ability to implement computer solutions that deliver value and accomplish industry-specific goals in the field of machine learning.

The project not only demonstrates my proficiency in reinforcement learning and algorithm optimization but also reflects my shows progression to meeting the course outcomes and continuously improving my skills in computer science. Moving forward, I aim to further refine my abilities, particularly in optimizing code for better hardware performance and exploring more advanced reinforcement learning techniques.

Conclusion

This artifact has contributed to my growth and capabilities in computer science. It gave me the chance to learn and apply advanced algorithms, troubleshoot and optimize code, and adapt to different hardware environments. Including this project in my ePortfolio not only showcases my technical skills but also reflects my dedication to continuous learning and improvement in the field of computer science.