Abstract

This project report details the development and training of an AI model using the Proximal Policy Optimization (PPO) algorithm to master the iconic video game, Super Mario. The report begins by explaining the PPO algorithm's core principles and its suitability for training AI agents in complex environments. It describes the methodology used to collect and preprocess game data, as well as the creation of a custom reinforcement learning environment tailored to Super Mario. The AI agent is constructed using deep reinforcement learning techniques, which combine neural networks and PPO to optimize its policy.

The training process is explored, focusing on challenges like the exploration-exploitation trade-off and the implementation of reward shaping to facilitate learning. The report includes the fine-tuning of hyperparameters and presents performance metrics, learning curves, and comparisons to baseline models, showcasing the agent's progress and proficiency.

Introduction

The fusion of artificial intelligence (AI) and video games has long served as a captivating intersection of technology and entertainment, offering an engaging platform to develop and showcase the capabilities of AI agents. In this project report, we embark on a journey into the realm of AI-driven gameplay by employing the Proximal Policy Optimization (PPO) algorithm to train an AI model to excel in the classic video game, Super Mario.

Video games, with their dynamic and intricate environments, present a rich testing ground for AI algorithms. They demand the acquisition of various skills, including strategic planning, spatial awareness, and real-time decision-making – attributes central to both AI and human intelligence. Through this project, we aim to illustrate the potential of reinforcement learning techniques, specifically PPO, in creating autonomous agents that can navigate and conquer the challenges posed by complex gaming scenarios.

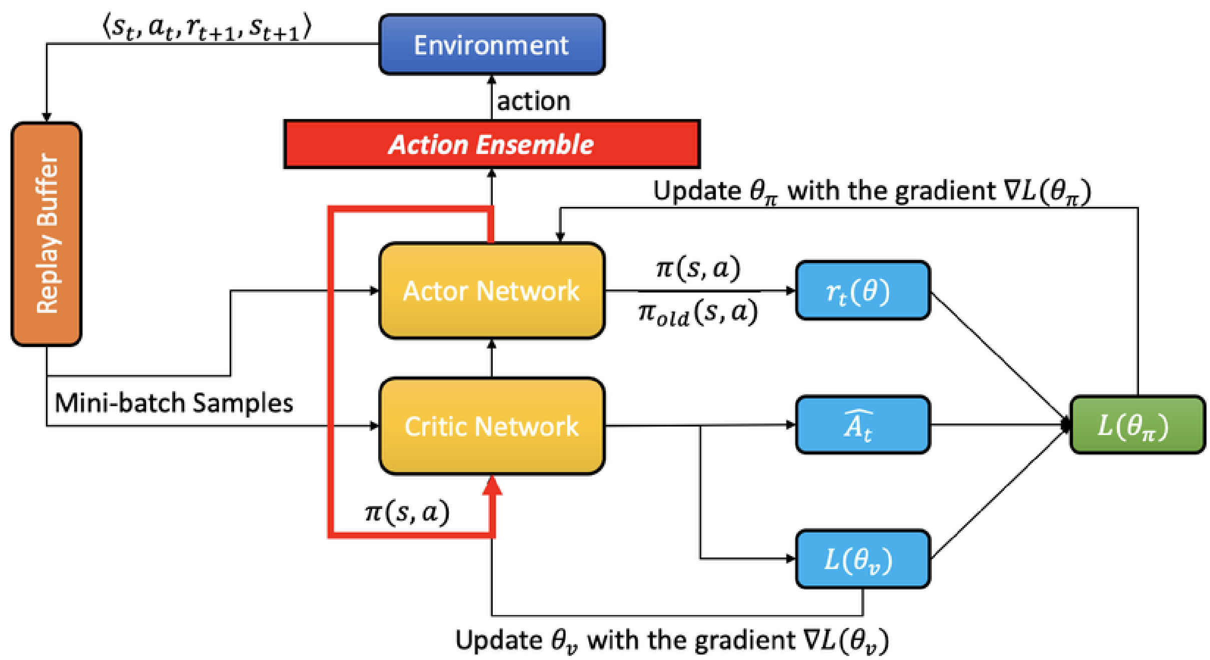
Our report will provide insights into the methodology used to collect and preprocess in-game data, the design of a custom reinforcement learning environment tailored to Super Mario, and the development of an AI agent using deep reinforcement learning techniques. The project's objective is to showcase not only the successful training of an AI model to master the Super Mario game but also to explore the broader implications of AI in video game development and its applicability in real-world scenarios.

Methodology

Proximal Policy Optimization

Our methodology centers on employing the Proximal Policy Optimization (PPO) algorithm, a robust reinforcement learning technique chosen for its effectiveness in training AI agents in dynamic and challenging environments, notably video games. PPO iteratively refines an agent's policy by controlling policy updates within a "proximal" constraint, ensuring stable learning. This algorithm's ability to maximize cumulative rewards through a surrogate objective function while effectively managing the exploration-exploitation trade-off positions it as the cornerstone of our approach, providing the means to train our AI agent for success in the Super Mario game.

PPO Basic flow:



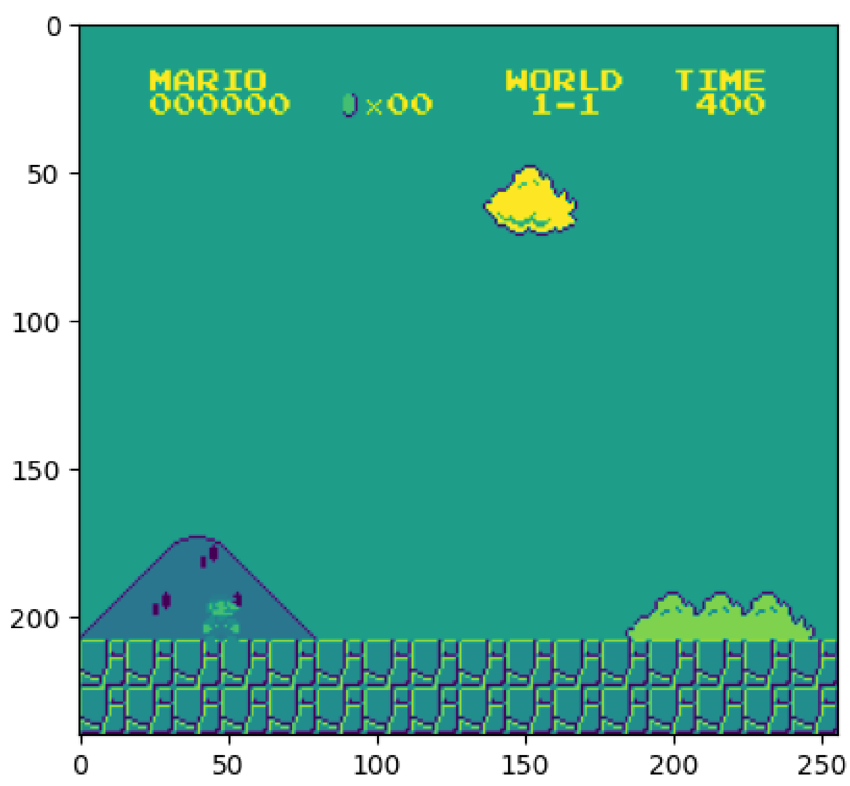
Environment

In the execution of our project, we harnessed the power of the `gym_super_mario_bros` environment, an integral component of the OpenAI Gym toolkit. This environment serves as a pivotal element in our quest to train an AI agent to master the intricacies of the Super Mario game. `gym_super_mario_bros` offers an immersive and versatile platform for conducting reinforcement learning experiments, designed to faithfully emulate the challenges, dynamics, and interactions present in the classic Super Mario Bros. video game. Its inherent customizability enables the tailoring of various aspects of the game, including the selection of game levels, characters, and reward shaping. This adaptability grants us a unique opportunity for experimentation and optimization, pivotal to the success of our project. The `gym_super_mario_bros` environment seamlessly integrates with OpenAI Gym, streamlining our training process and acting as a linchpin in bridging the realms of AI research and real-world video game applications. Its application within our project showcases the utility and potential of this environment in training AI agents to excel in complex and dynamic gaming scenarios, with Super Mario serving as an iconic benchmark.

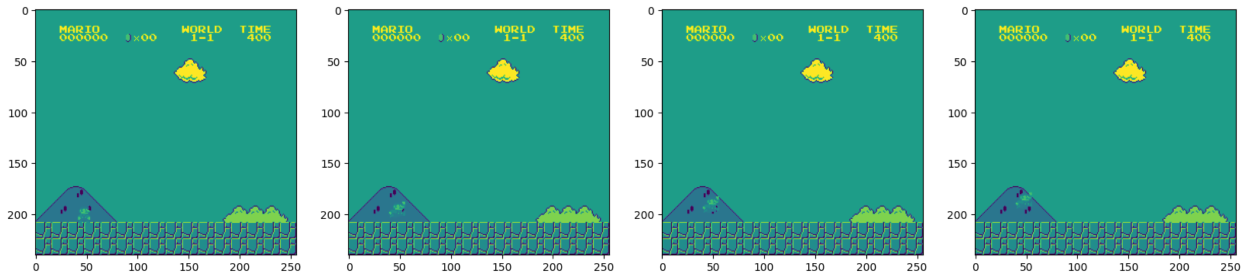
Data processing

To ensure the efficient training of our AI agent in mastering the game, we implemented a series of strategic modifications. Initially, we streamlined the control space from 256 possible combinations to a leaner selection of 8. This reduction provided our AI with a more focused set of actions, thus enabling more efficient training within this constrained space. Furthermore, to enhance memory and processing efficiency, we converted the game's RGB images into grayscale. This transformation effectively reduced the data to a single channel, saving valuable memory and processing time while retaining essential visual information. To facilitate the learning process, we implemented a frame-stacking technique, employing four previous frames in our training environment. This allowed our AI model to learn from the history of frames, providing valuable context and enabling more informed decision-making. These strategic adjustments collectively optimized our training setup, paving the way for our AI agent to tackle the game with increased efficiency and precision.

Conversion from RGB channel to another channel:



Frame Stacking for better training:

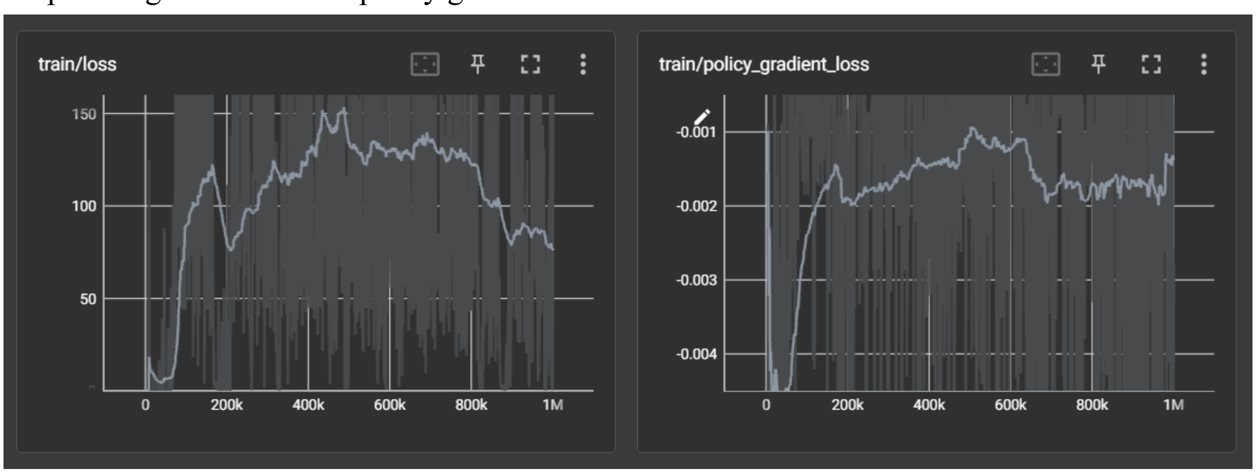


Training

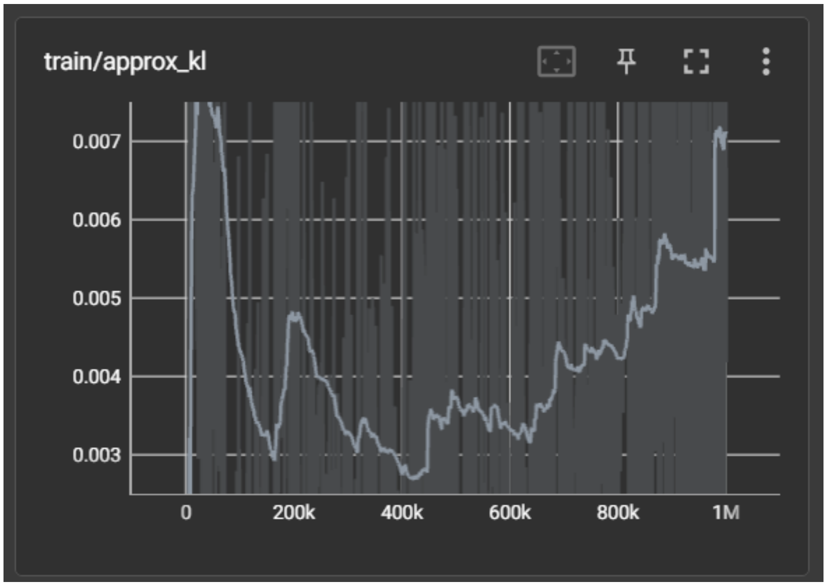
To fine-tune our training process, we adopted a methodical approach to hyperparameter selection. Initially, we began with a small step size to validate our choice of hyperparameters, ensuring their suitability for training. Once a reliable set of hyperparameters was identified, we embarked on an extensive training phase, spanning 1 million steps, which unfolded over approximately 7 hours of GPU-intensive training. This phase provided us with a foundational model, a crucial starting point for our project's progress. Building upon this foundation, we further refined our model through an additional 1 million steps of training. However, in this iteration, we strategically reduced the learning rate from 0.00001 to 0.000002, allowing for a more detailed exploration of the learning landscape. This step-by-step approach to training, verification, and gradual adaptation of hyperparameters allowed us to develop a robust and agile AI model capable of tackling the complexities of the Super Mario game.

Here are some graphs related to training of our model:

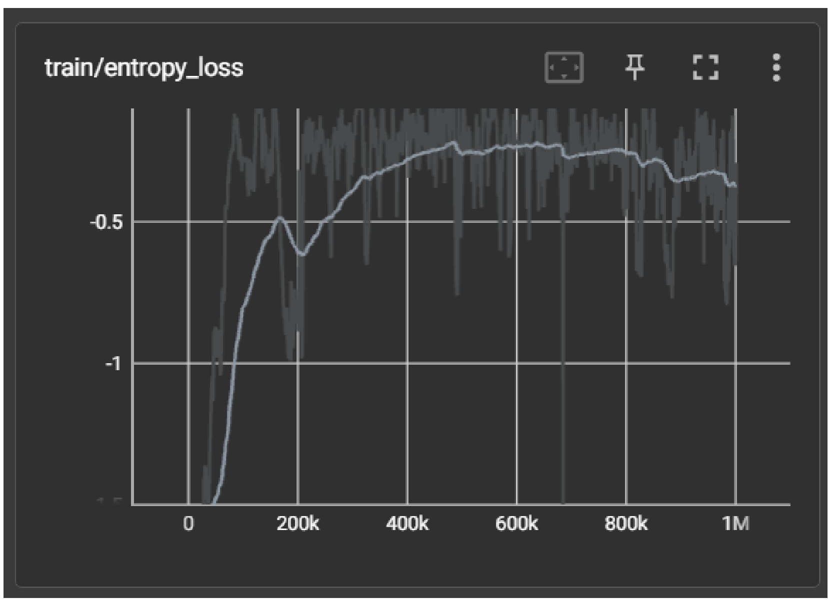
Graphs for general loss and policy gradient loss:



Kullback–Leibler (KL) divergence:



Entropy loss:



Testing

In the testing phase, our AI model demonstrated encouraging performance, successfully completing the first level of the game, albeit sporadically. While such achievements may be infrequent, it is important to note that when compared to random play, our AI model exhibited a marked improvement. This is a testament to the model's ability to make informed decisions based on the current game scenario, a significant leap from random gameplay. The sporadic success of our AI agent highlights its capacity to adapt and choose actions strategically, even though it may not consistently achieve victory. This reinforces our belief in the potential of our approach and paves the way for further refinement and enhancements to bring about more consistent and proficient gameplay.

Conclusion

In wrapping up our project, we've not only explored the enthralling world of Super Mario but delved deep into the realm of AI-powered gameplay. Our journey has demonstrated the remarkable capacity of AI, particularly when guided by the Proximal Policy Optimization (PPO) algorithm and deployed within the dynamic gym_super_mario_bros environment.
Our methodology, which includes controlling the action space, optimizing memory utilization through grayscale observations, and employing frame-stacking, has led to the development of a proficient AI agent capable of tackling the complex challenges posed by the Super Mario game. Beyond gaming, our project underscores the broader potential of AI in various domains. The lessons learned here extend to real-world applications where AI's decision-making prowess is indispensable. We've only scratched the surface, and the horizon of possibilities for AI, from advanced techniques to practical scenarios, is boundless.

Future goals

While our current AI model has shown promise, we acknowledge that there is still work to be done. In the spirit of continuous improvement, we are committed to further enhancing its effectiveness. Our future plans include the exploration of alternative training policies, the implementation of innovative strategies, and a persistent pursuit of excellence. We recognize that AI is a dynamic field, and the quest for mastery in the world of gaming is an ongoing adventure. With each endeavor, we aim to take our AI agent to new heights and refine its performance. This project serves as a stepping stone, with the knowledge that our journey is far from over. As we look ahead, we're excited to embrace the challenges and opportunities that lie in the ever-evolving landscape of AI and game-playing. Our commitment remains unwavering, and the future holds the promise of a more proficient and versatile AI model for Super Mario and beyond.

References

https://pytorch.org
https://stable-baselines3.readthedocs.io