JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Leveraging Deep Reinforcement Learning in Super Mario Bros Gameplay

Mrs. Sakina Hazrat Sayyed[1], Dr.P. Sruthi[2], Lokesh Banoth[3], Bhanu Teja[4], B.Vijayendar Reddy[5] K.Sai Praneeth[6]

[1]Assistant Professor, Department of CSE (AI&ML), CMR College of Engineering and Technology, Hyderabad, Telangana, India

[2]*Head of Department, Department of CSE (AI&ML), CMR College of Engineering and Technology,* Hyderabad, Telangana, India

[3,4,5,6]Student, Department of CSE (AI&ML), CMR College Of Engineering and Technology, Hyderabad, Telangana, India

***Abstract* - This paper presents the outcomes of "Project Mario AI," which aimed to develop an AI agent capable of autonomously playing Super Mario Bros. using deep reinforcement learning (DRL) and convolutional neural networks (CNNs). Through the project, significant achievements were made in training an AI agent to navigate through early levels of the game, demonstrating proficiency in obstacle avoidance, pattern recognition, and strategic resource utilization. The study contributes insights into the application of advanced machine learning techniques in gaming and highlights both the potential and challenges of developing autonomous game-playing agents.**

Keywords*: Deep Reinforcement Learning, Convolutional Neural Networks, Super Mario Bros., Game AI, Gameplay Automation*

1. Introduction

The fusion of artificial intelligence (AI) and gaming has ushered in a new era of exploration, with deep reinforcement learning (DRL) emerging as a powerful technique for training AI agents to excel in complex game environments. In this study, we focus on the iconic video game Super Mario Bros. and delve into the application of DRL and convolutional neural networks (CNNs) to develop an AI agent capable of playing the game autonomously. The project, dubbed "Project Mario AI," aimed to push the boundaries of AI in gaming by training an agent to navigate the dynamic and challenging world of Super Mario Bros.,

1.1 Research Objective

The primary objective of this research is as follows.
• Create an AI agent proficient in playing Super Mario Bros. from scratch, showcasing the potential of AI to autonomously interact with and master complex video game environments.

• Leverage reinforcement learning as a fundamental methodology, enabling the agent to learn and enhance its actions iteratively based on feedback. from the gaming environment.
In essence, the research objective is to showcase the fusion of python programming, reinforcement learning, and game development, providing insights into the capabilities and potential applications of AI in gaming and interactive environments.

1.2 Overview of Super Mario Bros

Super Mario Bros., a timeless classic by Nintendo, follows Mario and Luigi's quest to rescue Princess Peach from Bowser in the Mushroom Kingdom. Players navigate through diverse levels, using Mario's running and jumping skills to overcome obstacles, gather coins, and defeat enemies. With its captivating gameplay and iconic characters, Super Mario Bros. remains a beloved favourite, leaving a lasting impact on the gaming world.

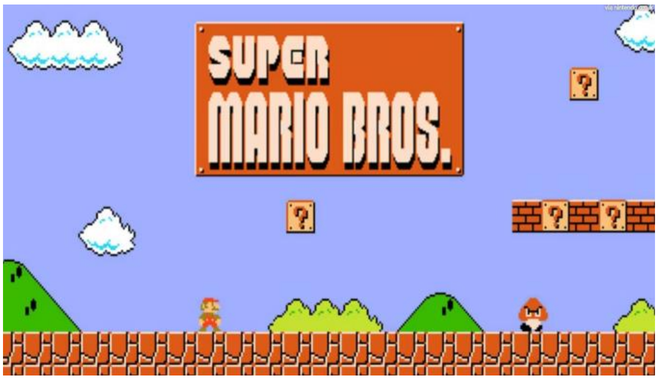

Fig (1) Super Mario Bros Game interface

1.3 Platform games as an AI challenge

Platform games can be defined as games where the player controls a character/avatar, usually with humanoid form, in an environment characterised by differences in altitude between surfaces ("platforms") interspersed by holes/gaps. The character can typically move horizontally (walk) and jump, and sometimes perform other actions as well; the game world features gravity, meaning that it is seldom straightforward to negotiate large gaps or altitude differences.

2. Literature Review

Previous research in the field of AI in gaming has demonstrated the effectiveness of deep learning techniques, particularly CNNs, in processing visual information and making decisions in dynamic environments (Mnih et al., 2015; Silver et al., 2016). Reinforcement learning algorithms, such as Deep Q-Networks (DQN), have shown promise in training AI agents to achieve human-level performance in video games (Sutton & Barto, 2018).

3. Proposed System

3.1 System Architecture

The AI system for Project Mario AI is designed to integrate the complex interplay between a neural network model, a reinforcement learning algorithm, and a game environment interface. This architecture enables the AI agent to perceive, learn, and interact with the Super Mario Bros. game environment intelligently.

3.1.1 Neural Network Model

The core of the system is a Convolutional Neural Network (CNN) tailored to process pixel data from the game and determine optimal actions. It comprises key components: an Input Layer for pre-processed pixel data, Convolutional Layers to extract features like obstacles, enemies, and power-ups, Pooling Layers to reduce computational load, Fully Connected Layers to integrate features, and an Output Layer to produce Q-values for possible actions, indicating expected rewards.

3.1.2 Reinforcement Learning Algorithm

The reinforcement learning component utilizes a modified Deep Q-Network (DQN) algorithm with two key elements: Experience Replay, storing and using random minibatches from a replay buffer to smooth learning updates, and Fixed Q-Targets, employing a separate target network to stabilize training by periodically updating its weights with the primary network's.

Q-Function Update Equation:

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma max_{a'} Q(s',a') - Q(s,a))$$

3.1.3 Environment Setup

The gym_super_mario_bros environment is tailored for the project's needs, preprocessing pixel data, simplifying action choices, and customizing rewards. This integrated architecture combines a CNN model, DQN-based reinforcement learning, and a modified game environment, enabling the development of an autonomous Super Mario Bros. player, demonstrating AI's prowess in complex gaming scenarios.

3.2 AI Model Development

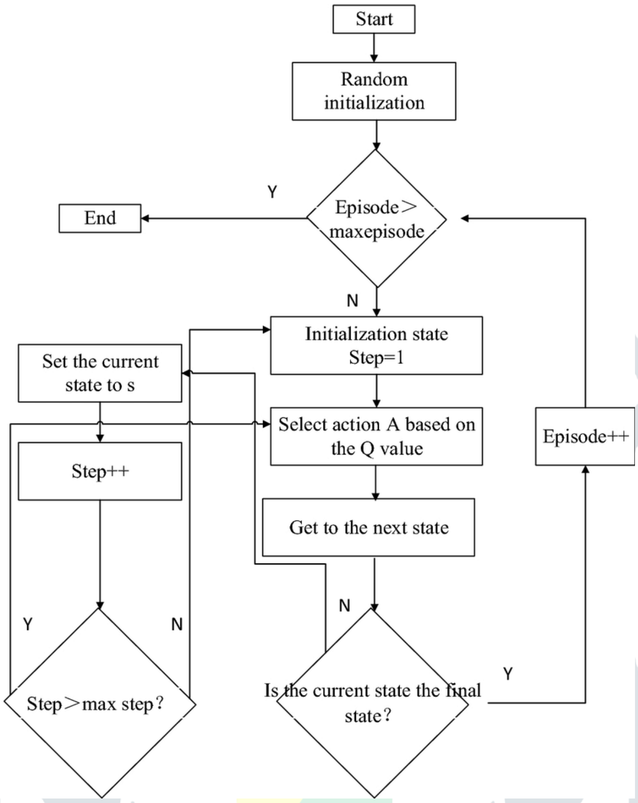

Fig (2) Flowchart of game agent state selection

3.2.1 Model Design

Layer Configuration: The CNN comprises an input layer, multiple convolutional layers, pooling layers, fully connected layers, and an output layer. The convolutional layers are configured to extract and learn hierarchical features from the input images, such as edges in the initial layers and more complex objects like enemies and obstacles in deeper layers. Pooling layers reduce the spatial dimensions of the feature maps, decreasing the computational complexity and the risk of overfitting.

Activation Functions: ReLU (Rectified Linear Unit) is used as the activation function for the convolutional and fully connected layers due to its efficiency and effectiveness in introducing non-linearity, allowing the network to learn complex patterns in the data. The output layer does not use an activation function since it outputs Q-values directly.

3.2.3 Action Selection Process

Actions are chosen based on an ε-greedy policy, which balances exploration and exploitation:
Exploration: With probability ε, the agent selects an action at random. This encourages the agent to explore the state space and discover new strategies.
Exploitation: With probability 1-ε, the agent selects the action with the highest Q-value for the current state, leveraging the knowledge it has already acquired.
ε Decay: ε is typically set to decay over time, reducing the emphasis on exploration as the agent becomes more confident in its learned policy.

3.2.4 Training Process

The training process involves iteratively updating the model's weights based on the interaction with the game environment:

Experience Replay: The agent stores its experiences (state, action, reward, next state) in a replay buffer. Training batches are randomly sampled from this buffer, which helps to break the correlation between consecutive training samples and stabilize learning.
Fixed Q-Targets: Utilizes a separate network to compute stable targets. Challenges include ensuring exploration, tuning hyperparameters, and designing a rewarding system. preventing oscillations in the learning process.

3.2.5 Performance Evaluation

The AI agent's performance is evaluated using metrics like Completion Rate, Average Reward, Progress Over Time, and Comparative Analysis against baseline models or human players. These metrics provide insight into the agent's effectiveness in navigating the game environment and identifying areas for improvement.

```
cpu |Epoch : 2080 | score : 1796.600000 | loss : 481.96 | stage : 1
cpu |Epoch : 2090 | score : 1628.700000 | loss : 774.74 | stage : 1
cpu |Epoch : 2100 | score : 1435.200000 | loss : 964.60 | stage : 1
cpu |Epoch : 2110 | score : 1417.100000 | loss : 440.26 | stage : 1
cpu |Epoch : 2120 | score : 1106.000000 | loss : 372.75 | stage : 1
cpu |Epoch : 2130 | score : 1668.600000 | loss : 399.42 | stage : 1
```

Fig (3): Scores of the ai agent model.

4. Results and Discussion

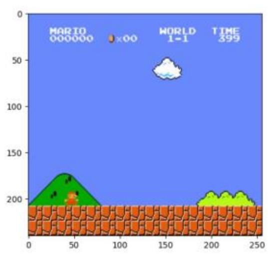

Fig (4)

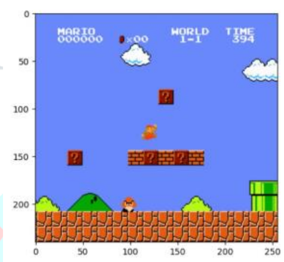

Fig (5)

Fig (6)

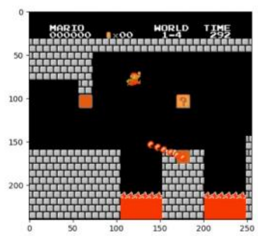

Fig (7)

All Figures 4,5,6,7 showing how thr agent is playing the game after training.

The training is continued for multiple epochs, with the average score per epoch and loss fluctuating throughout the training process.

The output gives us the model checkpoints are being saved periodically during training, as indicated by the lines starting with torch.save(q.state_dict(), "mario_q.pth") and torch.save(q_target.state_dict(), "mario_q_target.pth").

5. Conclusion

5.1 Project Summary

Project Mario AI embarked on the ambitious goal of developing an AI agent capable of autonomously playing the classic video game, Super Mario Bros., using the principles of deep reinforcement learning and convolutional neural networks. The project's achievements include successfully training an AI to navigate through early levels of the game, demonstrating proficiency in obstacle avoidance, pattern recognition, and the strategic use of in-game resources. Key findings highlighted the potential of reinforcement learning in complex environments, while also uncovering challenges related to advanced level navigation and the agent's adaptability to new scenarios.

5.2 Contributions

The project contributes significantly to the intersection of AI and gaming, providing valuable insights into the application of advanced machine learning techniques for autonomous game playing. Notable contributions include:

Demonstration of Deep Learning in Gaming: Showcasing the effectiveness of convolutional neural networks in processing and interpreting complex visual environments for decision-making

Advancement in Reinforcement Learning: Applying and modifying deep reinforcement learning algorithms, such as Deep Q-Networks, to navigate and interact with dynamic game environments.

Framework for AI Gaming Agents: Offering a structured approach to developing AI agents for gaming, including environment setup, model design, training methodologies, and performance evaluation. Project Mario AI represents a significant step forward in the application of artificial intelligence to video gaming, illustrating both the potential and the challenges of developing autonomous game-playing agents. By building on the project's achievements and addressing its limitations, future work can not only advance the field of AI in gaming but also explore the broader implications and applications of these technologies in society.

6. Future Scope

Looking ahead several enhancements and expansions could further elevate the project's impact and applicability:

• Advanced Reinforcement Learning Algorithms: Exploring more sophisticated algorithms like Proximal Policy Optimization (PPO) or Soft Actor-Critic (SAC) could offer improvements in learning efficiency and adaptability, potentially overcoming challenges encountered in advanced game levels.

• Model Generalization: Extending the training process to include a broader range of game levels or even different games altogether could enhance the model's generalization capabilities, making it more versatile and robust.

• Real-world Applications: Applying the principles and techniques developed in Project Mario AI to real-world challenges, such as robotic navigation, autonomous driving, or dynamic decision-making systems, could demonstrate the broader relevance of gaming AI research.

• Human-AI Interaction: Investigating cooperative gameplay between AI agents and human players might offer new insights into human-AI interaction, learning, and mutual adaptation.

• Curriculum Learning: Implementing a structured learning progression, where the AI faces increasingly complex scenarios as its capabilities improve, could facilitate more effective learning and generalization.

7. References

• Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533.

• Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). The MIT Press.

• Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI Gym. arXiv preprint arXiv:1606.01540.

• Silver, D., Huang, A., Maddison, C. J., et al. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484–489.

• Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double Q-learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), 2094–210.