Project Title:

Reinforcement Learning Agent for Street Fighter II using PPO in Gym-Retro

Submitted By:

Ijlal Iqbal (22K-5034)

Huzefa Saifuddin (22K-5125)

Areeb ur Rehman (22K-6003)

Course:

AI

Instructor:

Alishba Subhani

Submission Date:

March 10th, 2025

1. Project Overview

Project Topic:

This project focuses on applying reinforcement learning (RL) to the classic fighting game Street Fighter II using the Gym-Retro framework. The agent will learn how to fight against a predefined AI opponent using advanced RL algorithms. Unlike traditional board games, this real-time, fast-paced game environment introduces continuous feedback and complex decision-making in dynamic states.

Objective:

The goal is to train an AI agent that can effectively compete against the in-game AI in Street Fighter II using Proximal Policy Optimization (PPO). The project will involve customizing rewards to encourage strategic fighting behaviors and learning optimal actions through trial and error.

2. Game Description

Original Game Background:

Street Fighter II is a 2D fighting game where two characters face off in a best-of-three round match. Players can move left or right, jump, crouch, block, and execute various attacks. The goal is to reduce the opponent's health bar to zero before the round timer ends.

Innovations Introduced:

- Using RL to simulate human-like learning in a fast-paced fighting environment.
- Custom reward function emphasizing not only winning but also minimizing damage and executing advanced moves.
- Observation and action space reduction to improve learning efficiency.

These innovations increase complexity by introducing delayed rewards, real-time actions, and partial observability - common in real-world decision systems.

3. AI Approach and Methodology

AI Techniques to be Used:

- Reinforcement Learning (PPO): Proximal Policy Optimization will be used for policy updates due to its stability in high-dimensional action spaces.
- Gym-Retro: For environment emulation and integration.
- Neural Networks: For policy and value function approximation.

Heuristic Design:

- Reward for dealing damage to the opponent (+1 per hit).
- Reward for maintaining higher health than the opponent at end of round.
- Penalty for taking damage or losing a round.
- Bonus reward for winning a match.

Complexity Analysis:

- PPO has a training time complexity dependent on the number of environment steps and network size.
- The environment's large action space and high frame rate pose challenges for learning stability and efficiency.
- Real-time response and sparse rewards make convergence slow without proper tuning.

4. Game Rules and Mechanics

Modified Rules:

- Only one character (e.g., Ryu) will be used to reduce complexity.
- Fixed opponent for training (e.g., Ken AI).
- Game is limited to best-of-three rounds.
- Frame skip is introduced to reduce computational load.

Winning Conditions:

Agent wins if it defeats the opponent by reducing its health to zero in two out of three rounds.

Turn Sequence:

Both players act simultaneously in a continuous loop (frame-based turns). Each frame is treated as a decision point for the agent.

5. Implementation Plan

Programming Language:

- Python

Libraries and Tools:

- Gym-Retro
- Stable-Baselines3 (PPO)
- NumPy, OpenCV (frame processing)
- PyTorch

Milestones and Timeline:

- Week 1-2: Environment setup, preprocessing, reward function design
- Week 3-4: Basic agent training using PPO
- Week 5-6: Hyperparameter tuning and performance evaluation
- Week 7: Gameplay testing and visualization
- Week 8: Final report and demo preparation

6. References

- OpenAI Baselines and Stable-Baselines3 documentation
- Gym-Retro GitHub repository and tutorials
- Sutton & Barto - Reinforcement Learning: An Introduction
- Papers on PPO and RL in games