Adaptive RL Agent for Atari Breakout

Rushikesh Waghmare

https://github.com/iamrealrushi/Adaptive-RL-Agent-for-Atari-Breakout

Implementation Approach

- Developed a Deep Q-Network (DQN) from scratch using TensorFlow.
- Preprocessed environment frames to grayscale and resized to 84×84.
- Implemented experience replay and a target network.
- Curriculum learning applied:
  - Early episodes used standard Breakout.
  - Gradually introduced dynamic difficulty: paddle speed, ball speed, brick regeneration, paddle size.
- Simulated environment variations by modifying frame processing and step calls manually.

Results and Performance

- Achieved a reward of 7 in dynamic environments.
- Model handled gradual difficulty shifts reasonably well.
- Performance dropped slightly when sudden ball speed or paddle shrinkage occurred.
- Trained over 400 episodes with a decaying epsilon-greedy exploration strategy.

Challenges and Solutions

- **Uncompatibility of tensorflow with python 3.13:** Reinstalled python 3.11 and installed all required packages on PyCharm.

Future Improvements

- Integrate RNNs or LSTMs to remember state history and better detect pattern changes.
- Implement dynamic difficulty prediction from recent frame sequences.

- Visualize and analyze learned feature maps to understand adaptation mechanisms.
- Test with longer training (e.g., 1000+ episodes) for improved policy stability.