CartPole RL Enhancement

A Deep Reinforcement Learning Approach with Curriculum Learning and Energy Consumption Optimization

Louis Gauthier

January 13, 2025

Table of Contents

Project Overview

Environment Description

Steps Towards the Solution

Possible Extensions

Possible Shortcomings

Results and Analysis

Conclusion

Project Overview

- **Objective:** Enhance the classic CartPole environment by introducing a third action ("Do Nothing") and develop a deep RL solution to manage the modified environment.
- **Key Features:**
  - Addition of a neutral action to the action space.
  - Implementation of curriculum learning to adjust reward weights dynamically.
  - Incorporation of temperature-scaled softmax for exploration-exploitation balance.
  - Logging and monitoring using TensorBoard.
  - Energy consumption as a reward component.
- **Outcome:** Achieved a high success rate with efficient energy usage after 100k training steps.

Environment Description

- **Classic CartPole:**
  - Two actions: Apply a force to the cart to push it left or right.
  - Observation Space: Cart position, cart velocity, pole angle, pole angular velocity.
  - Objective: Balance the pole by moving the cart. Episode ends if pole angle exceeds a threshold or cart moves out of bounds.

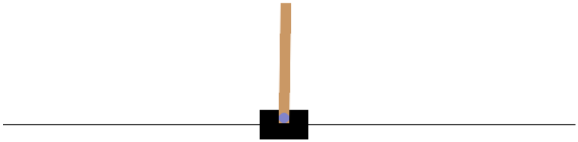


Figure: The CartPole Environments

Environment Description

- **Modified CartPole:**
  - **Third Action:** *Do Nothing* - No force applied to the cart.
  - **Reward Structure:** Split into four equal components: Alive, Distance to Center, Pole Angle, and Energy Usage.
  - **Curriculum Learning:** Adjusts reward weights based on training phases.

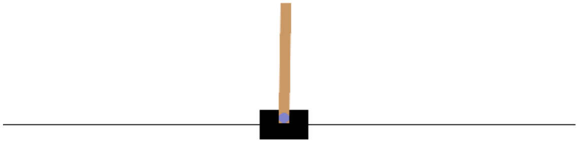


Figure: The CartPole Environments

Steps Towards the Solution

1. **Environment Modification:**
   - Added a third "Do Nothing" action to the action space.
   - Adjusted the reward function to incorporate energy consumption.
2. **Algorithm Development:**
   - Implemented a combination of Double DQN and Dueling DQN for stable learning.
   - Used a target network to improve training stability.
   - Utilized Prioritized Experience Replay to sample important transitions.
   - Applied Gradient Clipping to prevent exploding gradients.
3. **Enhancements:**
   - Introduced Curriculum Learning to adjust reward weights dynamically.
   - Integrated Temperature-Scaled Softmax for exploration-exploitation balance.
   - Tested Normalization for observations and rewards.

Steps Towards the Solution

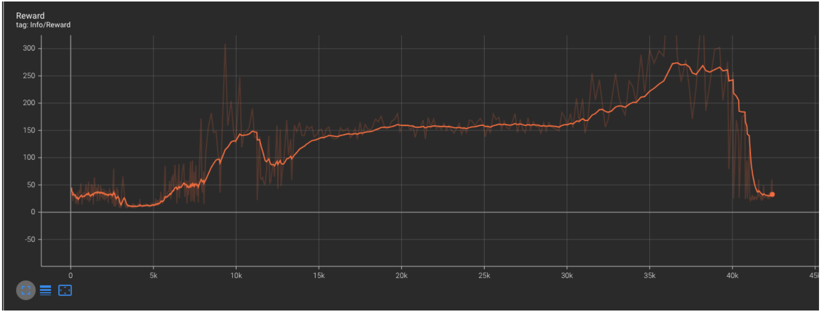


Figure: Catastrophic Forgetting occuring in the training process

Steps Towards the Solution

4. **Problems encountered:**
   - Catastrophic forgetting: The agent forgets previously learned behaviors.
     - Solution: Use non-deterministic actions and temperature scheduling.
   - Reward function is not a good metric of performance during training.
     - Solution: Run evaluations every 2k steps to track performance.
5. **Logging and Monitoring:**
   - Set up TensorBoard for real-time monitoring of training metrics.
   - Logged individual sub-rewards and penalties for detailed analysis.
   - Evaluation every 2k steps to track performance.
6. **Checkpointing:**
   - Enabled resuming training from saved checkpoints.
   - Handling crashes or interruptions during training.

TensorBoard Metrics

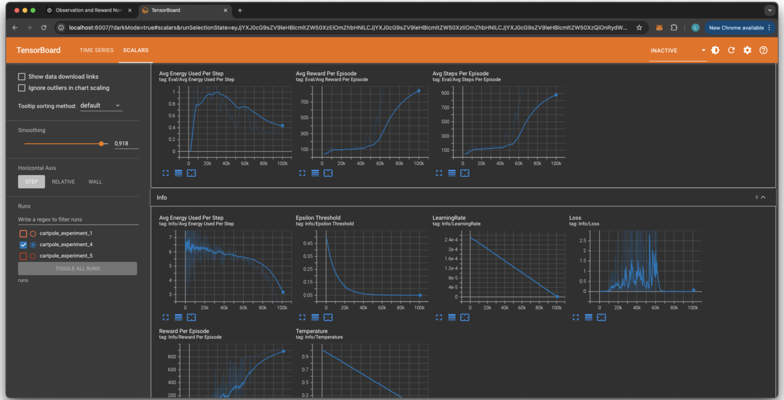


Figure: TensorBoard Dashboard showing training metrics

Algorithm Enhancements

- **Double DQN:** Mitigates overestimation bias by decoupling action selection and evaluation.
- **Prioritized Experience Replay:** Samples important transitions more frequently to accelerate learning.
- **Normalization:**
  - *Observation Normalization:* Stabilizes training by normalizing input features.
  - *Reward Normalization:* Ensures consistent reward scaling.
- **Curriculum Learning:**
  - Phases:
    1. **0-20k steps:** 50% Alive Reward, 50% Pole Angle Reward.
    2. **20k-50k steps:** 33% Alive, 33% Pole Angle, 33% Distance to Center.
    3. **Above 50k steps:** 25% each for Alive, Pole Angle, Distance to Center, and Energy Usage.

Algorithm Enhancements

- **Temperature-Scaled Softmax:**
  - Initial Temperature: 1.0
  - Final Temperature: 0.1
  - Linearly decreases over training to reduce exploration over time.

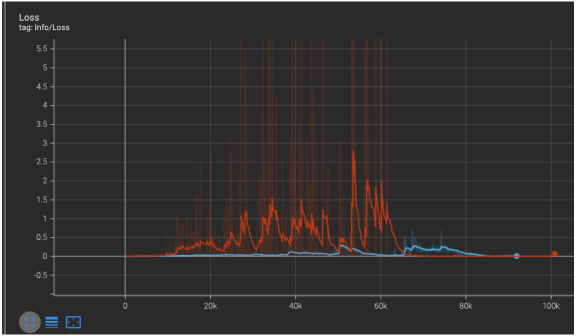


Figure: Loss Stabilization: Prioritized (blue) vs. Uniform Replay (red)

Possible Extensions

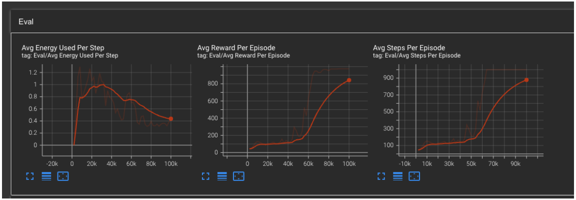
- **Stratification:**
  - Implement stratified sampling to ensure diverse experiences in replay buffer.
- **Distribution-Aware Buffer:**
  - Explore Cosine Similarity or Variational Autoencoders (VAEs) to model state distributions and enhance replay buffer sampling.
- **Parallel Environments:**
  - Increase to 16 parallel environments for faster experience collection and improved training efficiency.
- **Reward Function Refinement:**
  - Schedule Pole Angle and Distance to Center rewards to decrease over time, focusing more on energy optimization.
- **Model Checkpointing:**
  - Maintain multiple checkpoints to preserve best-performing models across different training phases.
- **Advanced Exploration Strategies:**
  - Incorporate other exploration methods like entropy regularization or intrinsic motivation (e.g., curiosity).

Possible Shortcomings

- **Algorithm Tuning:**
  - Deep RL algorithms require extensive hyperparameter tuning for optimal performance.
- **Overfitting:**
  - The model may overfit to the specific environment configuration, limiting generalizability.
- **Fixed Length Episodes:**
  - Fixed-length episodes combined with energy consumption reward may lead to suboptimal behavior where the agent overfits to short-term gains.
- **Training For Too Long Might Decrease Performance:**
  - The agent might overfit to near-perfect behavior and forget more complex situations.
- **Learning Stability:**
  - Other on-policy algorithm like PPO might be more stable and efficient.

Results and Analysis

- **Training Outcome after 100k Steps:**
  - **Average Reward:** 975/1000
  - **Energy Consumption:** Under 0.4 per step (Full Power = 10 per step)
  - **Success Rate:** 100% during evaluation (No failures)
- **Evaluation Strategy:**
  - Conducted evaluations every 2k steps.
  - Essential for monitoring performance due to off-policy learning, non-deterministic actions, and exploration.
- **Prioritization Analysis:**
  - Prioritized Experience Replay helped stabilize loss.
  - Did not significantly enhance overall performance compared to uniform sampling.



Conclusion

- Successfully enhanced the CartPole environment with a third "Do Nothing" action and optimized reward structure for energy consumption.
- Developed a robust deep RL solution incorporating curriculum learning and temperature-scaled softmax for effective exploration-exploitation balance.
- Achieved high performance with minimal energy usage and a perfect success rate in evaluations.
- Identified areas for improvement, including further algorithm tuning and exploring advanced buffer management techniques.