Reinforcement Learning for Leaf and Fruit Collection in a Grid-Based Tree Environment

In-Depth Analysis of PPO and Dueling DQN with LSTM Classifier

Nitin (CS22B2047)

Indian Institute of Information Technology, Design and Manufacturing Kancheepuram
Department of Computer Science and Engineering
Under the Guidance of Dr. Rahul Raman

May 10, 2025

Objective:

- Train Reinforcement Learning (RL) agents to collect falling leaves and fruits in a **20×20 grid-based TreeEnvironment**.

Inspiration:

- Derived from dynamic scheduling problems
  (Chang et al., 2019)

RL Frameworks Used:

- Proximal Policy Optimization (PPO)
- Dueling Deep Q-Network (Dueling DQN) with LSTM Classifier

Focus Areas:

- Detailed working of both algorithms
- Comprehensive performance comparison between PPO and Dueling DQN + LSTM

TreeEnvironment Overview

Setup:

- 20×20 grid; each cell represents a tree branch
- Each branch can hold **0–5 leaves** and **0–5 fruits**
- Leaves and fruits fall based on **random timers (1–4 time units)**

Action Space:

- Move **left**, **right**, **up**, **down**, or **stay**

Reward Structure:

- **+1** for collecting a leaf
- **+5** for collecting one fruit
- **+6** for collecting a fruit + leaf together
- **Penalty:** $-(\text{leaves} + \text{fruits}) \times 5$ if fruit has **2 or more leaves**

Episode Termination:

- Episode ends after **100 time steps**

PPO – Algorithm Overview

Algorithm:

- Proximal Policy Optimization (PPO)

Key Features:

- Policy gradient method
- Clipped surrogate objective for training stability

Architecture:

- **Actor Network:** 128 $\rightarrow$ 64 units, softmax output (action probabilities)
- **Critic Network:** 128 $\rightarrow$ 64 units, linear output (state values)

Hyperparameters:

- Actor LR: 0.0001, Critic LR: 0.0005
- Discount Factor (γ): 0.99
- GAE Lambda (λ): 0.95
- Clipping Epsilon: 0.3

PPO – Working and Training

Workflow:

- Actor selects action using policy $\pi_\theta(a|s)$
- Environment returns reward r_t and next state s_{t+1}
- Critic computes advantages:

$$A_t = \sum (\gamma\lambda)^k \delta_{t+k}, \quad \text{where } \delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$$

Training Setup:

- 600 episodes, 100 steps each

Actor Update:

- Clipped PPO loss:

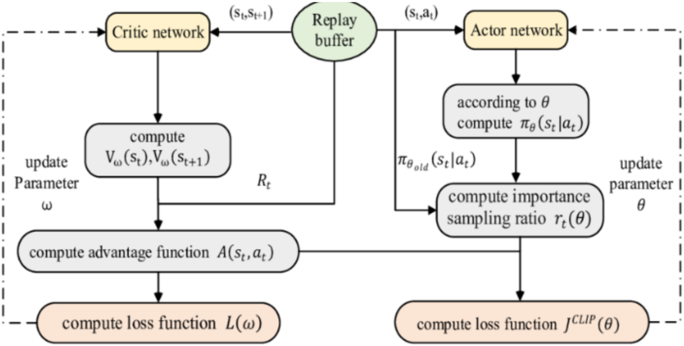
$$L_{\text{actor}}(\theta) = \mathbb{E}_t \left[\min \left(\frac{\pi_\theta}{\pi_{\theta_{\text{old}}}} A_t, \text{clip} \left(\frac{\pi_\theta}{\pi_{\theta_{\text{old}}}}, 1-\epsilon, 1+\epsilon \right) A_t \right) \right]$$

Critic Update:

- MSE Loss:

PPO - Training Architecture Diagram

- Illustrates the interaction between Actor-Critic networks in PPO.
- **Workflow Visualized:**
  - Actor network computes action probabilities using policy π_θ.
  - Critic evaluates the state using value function $V_\omega(s)$.
  - Advantage and loss functions are computed for updates.
  - Stabilization technique: Importance sampling ratio and clipped surrogate objective.



Dueling DQN with LSTM Classifier - Algorithm Overview

Algorithm: Dueling DQN + LSTM Classifier

LSTM Classifier:

- Predicts item availability (leaves/fruits).
- Architecture:
  - Two LSTM layers: 64 units and 32 units.
  - Followed by Dense layers: 16 units and 2 units with softmax activation.

Dueling DQN:

- Separates value and advantage streams.
- Architecture:
  - LSTM layer with 128 units.
  - Followed by Dense layers: 64 and 32 units.

Hyperparameters:

- $\gamma = 0.95$
- ϵ: decays from 1.0 to 0.01 (decay $= 0.999$)
- Learning Rate (LR): 0.001

Dueling DQN with LSTM Classifier - Working (Part 1)

LSTM Classifier Workflow:

- Trained on 1000 sequences (length 10) to predict item presence.
- **Inputs:** Historical timer patterns.
- **Outputs:** Probability of leaves/fruits availability.

Dueling DQN Workflow:

- Uses LSTM predictions to inform action selection.
- Computes Q-values using the formula:

$$Q(s,a;\theta) = V(s;\eta) + \left(A(s,a;\psi) - \frac{1}{N_{\text{action}}} \sum_{a'} A(s,a';\psi) \right), \quad N_{\text{action}} =$$

- Action selection: Epsilon-greedy policy (ϵ decays from 1.0 to 0.01).

Dueling DQN with LSTM Classifier - Working (Part 2)

Training Process:

- 600 episodes, 100 steps each.
- Stores transitions in replay buffer (size 5000).
- Samples batch (size 32) for training.

Updates:

- **Target Q-value:**

$$Q_{\text{target}}(s_t, a_t) = r_t + \gamma \max_{a'} Q_{\text{target}}(s_{t+1}, a'; \theta^-), \quad \gamma = 0.95$$

- **DQN update: Bellman error:**

$$L(\theta) = \mathbb{E}[(Q(s, a; \theta) - Q_{\text{target}}(s, a))^2], \quad \text{Learning Rate} = 0.001$$

- Target network updated every 10 steps.

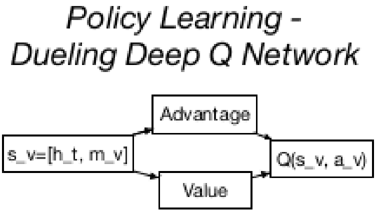
Dueling DQN and LSTM Architecture

Top: Dueling DQN Module

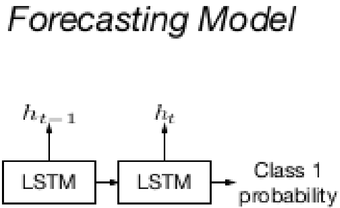
- Decomposes Q-values:
  - **Advantage function:** Measures importance of actions.
  - **Value function:** Measures importance of states.
  - Combines both: $Q(s, a) = V(s) + A(s, a)$

Bottom: LSTM Classifier

- Used for sequence prediction and state forecasting.
- LSTM cell passes hidden state $h_{t-1} \rightarrow h_t$, outputting class probabilities.



Dueling DQN



Results Overview

PPO:

- Training reward: **-433.49**, Time: **6043.28s**
- Evaluation reward: **-385.76**, Success rate: **0.00%**
- Penalty avoidance: **51.87%**

Dueling DQN with LSTM:

- Training time: **1162.36s** (~5x faster)
- Evaluation reward: **8.32**, Success rate: **60.50%**
- Penalty avoidance: **96.50%**

Comparison of Algorithms - Metrics

Metric	PPO	Dueling DQN with LSTM
Average Reward	-385.76	8.32
Success Rate	0.00%	60.50%
Penalty Avoidance	51.87%	96.50%
Good Outcomes Accuracy	51.87%	60.50%
Training Time (seconds)	6043.28	1162.36

Key Insight: *Dueling DQN outperforms PPO across all metrics.*

Comparison of Algorithms - Reward Distribution

PPO Reward Distribution:

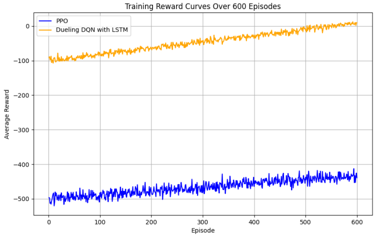
- Penalty: 48.13%
- Other: 29.60%
- Leaves: 6.43%
- One Fruit: 9.98%
- Fruit + Leaf: 5.85%

Dueling DQN Reward Distribution:

- Penalty: 3.50%
- None: 31.75%
- Leaves: 24.75%
- One Fruit: 19.75%
- Fruit + Leaf: 16.00%

Insight: *Dueling DQN achieves balanced collection, significantly reducing penalties.*

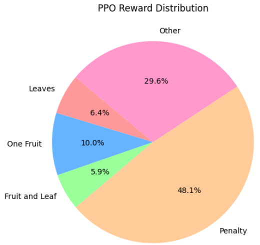
Training Reward Curve vs Metrics Comparison



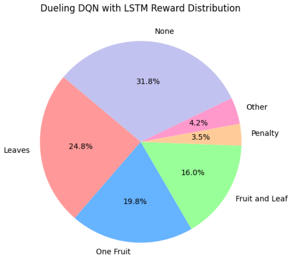
figureReward Curves

figureMetrics Comparison

Reward Distribution: PPO vs DQN



figurePPO Reward Distribution



figureDQN Reward Distribution

Comparison of Algorithms: Why Dueling DQN Excels

- **Temporal Prediction:**
  - LSTM predicts item falls, enabling strategic navigation.
- **Dueling Architecture:**
  - Separates value and advantage for better Q-value estimation.
- **Sequential Actions:**
  - Processes actions incrementally, reducing action space.
- **Exploration:**
  - Epsilon-greedy + replay ensures robust state-action coverage.
- **Penalty Avoidance:**
  - Dueling DQN: **3.50%** penalties vs. PPO: **48.13%**.

Applications

- **Reinforcement Learning Research:**
  - Benchmark for dynamic RL tasks.
- **Robotics:**
  - Autonomous navigation for agricultural robots.
- **Real-Time Example:**
  - Robot in an orchard: LSTM predicts falls, Dueling DQN navigates.
- **Benefits:**
  - Efficiency, cost reduction, scalability, sustainability.

Conclusion

- **Summary:**
  - Dueling DQN with LSTM Classifier outperforms PPO.
  - Superior reward (8.32 vs. -385.76), success rate (60.50%), and penalty avoidance (96.50%).
  - Driven by temporal prediction, dueling architecture, and efficient exploration.
- **Impact:**
  - Advances RL research and robotics applications.
  - Aligns with Chang et al. (2019) insights.

Future Work and References

- **Future Work:**
  - Scale to larger grids (e.g., 50x50).
  - Explore advanced RL methods (Soft Actor-Critic, Rainbow DQN).
  - Integrate with physical robots.
- **References:**

  - Chang et al. (2019). *Dynamic Measurement Scheduling*. PMLR 97.
  - Schulman et al. (2017). *PPO Algorithms*. arXiv:1707.06347.
  - Wang et al. (2016). *Dueling DQN*. ICML.

End !

Thank You!