Literature Review: Reinforcement Learning and Policy Optimization

Introduction

Reinforcement Learning (RL) is a machine learning paradigm where an agent interacts with an environment, receiving feedback in the form of rewards to optimize its behavior.

This review covers:

- Markov Decision Processes (MDPs)
- Q-learning & Deep Q-learning
- Policy optimization methods: PPO & GRPO
- Application to our poker project

Foundations of Reinforcement Learning

Markov Decision Processes (MDPs)

An MDP is defined by:

- **States (s):** Environment's possible configurations.
- **Actions (a):** Choices available to the agent.
- **Reward (r):** Feedback signal evaluating action quality.
- **Transition Function (T):** Defines state transitions.
- **Policy (π):** Function dictating action selection.

Goal: Learn a policy that maximizes cumulative reward.

Reinforcement Learning in Large Language Models (LLMs)

- **State:** Current text sequence.
- **Action:** Next token prediction.
- **Reward:** Evaluates token quality.
- **Use Case:** RL fine-tuning for better LLM responses.

Q-Learning

Q-learning is an off-policy algorithm that estimates the expected cumulative reward:

$$Q(s, a) = E[R|s, a]$$

Bellman Equation Update:

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

Deep Q-Learning (DQN)

To handle large state spaces, DQN employs:

1. **Experience Replay**: Stores past transitions to reduce correlation.
2. **Target Networks**: Stabilizes training by using a separate Q-network.
3. **Loss Function**: MSE between predicted and target Q-values.

Policy Optimization Methods

Policy Gradient Methods

Optimize a policy π_θ directly by maximizing expected return:

$$J(\theta) = E_{\tau \sim \pi_\theta}[R(\tau)]$$

Updated using stochastic gradient ascent.

Proximal Policy Optimization (PPO)

PPO stabilizes policy updates using a **clipped surrogate objective**:

$$L(\theta) = E\left[\min(r_t(\theta)A_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)\right]$$

where $r_t(\theta)$ is the probability ratio of new and old policies.

Guaranteed Reward Policy Optimization (GRPO)

GRPO improves policy optimization with:

- **Theoretical Guarantees:** Ensures monotonic policy improvement.
- **Reward-Centric Updates:** Adjusts updates based on long-term rewards.
- **Empirical Performance:** Outperforms PPO and TRPO in benchmarks.

Poker Model Progress

Current Work

1. Attempted **DeepSeek (1.5B Parameters)** fine-tuning, but faced **GPU vRAM limitations** in Colab.
2. Exploring **Unsloth + GRPO on Llama 3B**:
    - Designing custom reward functions for optimal poker actions.
    - Assigning higher rewards for correctly predicting the best move.

Conclusion

Reinforcement Learning has evolved through:

- **Q-learning → DQN → PPO → GRPO**
- Applied to complex decision-making tasks like **poker AI**

Our goal: Train an RL-powered reasoning model to optimize poker decision-making.

Thank You!

Questions?