Warm-Starting Deep Reinforcement Learning with Genetic Algorithms

Caleb Dame
James Williams

705.643: Deep Learning Developments with PyTorch

September 21, 2025

Project Proposal

Motivation

Deep reinforcement learning (DRL) policy methods (such as PPO and DQN) are powerful, but in practice they often suffer from fragility: training can fail without careful reward shaping and hyperparameter tuning, and training can be highly sensitive to random seeds. In contrast, evolution-based methods like genetic algorithms (GAs) explore spaces broadly and can discover solutions reliably with only a fitness function without setting policy-specific parameters, though at a higher compute cost.

This project would explore whether combining these two paradigms—using a GA to produce a *warm-start policy* before fine-tuning with PPO/DQN—can improve training stability and success rate over a standard random weight initialization.

Research Questions

- Does GA warm-starting increase the probability that DRL converges to a "successful" policy? If so, how successful?
- Does the upfront search time reduce DRL training time or statistically help prevent failed runs?
- At what additional wall-clock/runtime cost?
- How large can the network grow before using a GA where phenotypes representing an individual layer's weights and biases are no longer viably created or useful population evolution becomes too time-prohibitive?
- How robust are GA-initialized policies to added noise in the environment compared to those discovered with Deep Reinforcement Learning alone?

Methodology

We will implement a two-stage pipeline:

1. **GA pretraining:** Run a lightweight GA or evolution strategy (population 32–64, 20–40 generations) to evolve policy network weights on a given environment. Stop once a modest performance threshold is reached or a fixed budget is used.
2. **DRL fine-tuning:** Initialize PPO or DQN with the evolved weights (or seed replay buffers with GA rollouts), then continue training under a fixed environment step budget.

We will benchmark across several environments:

- **Classic control:** CartPole-v1, LunarLander-v2
- **MinAtar:** Breakout, Asterix
- **clubs_gym:** Leduc Poker, Kuhn Poker
- (Stretch Goal) Image-based Pong or Procgen Environments

Evaluation Metrics

- **Success rate:** Fraction of runs/seeds reaching a target reward threshold.
- **Sample efficiency:** Environment steps to reach target reward.
- **Final performance:** Average reward after full budget.
- **Fragility:** Variance across seeds; sensitivity to small hyperparameter changes.
- **Cost:** Wall-clock time, GPU/CPU time for GA+DRL vs DRL alone.

Planned Analyses

We will compare:

1. DRL from scratch (baseline).
2. GA warm-start with top network $\rightarrow$ DRL.
3. GA warm-start with all top k networks $\rightarrow$ DRL .

Plots will include reward vs steps, success rates, robustness under noise, and cost comparisons.

Risks and Mitigation

Potential risks include excessive compute costs or GA pretraining that fails to generalize. To mitigate, we will:

- Use small environments and networks (MLPs, MinAtar, clubs_gym) for feasibility.
- Develop downstream DRL network baselines for evaluating GA quality.
- Limit GA to a fixed fraction of total steps.
- Run multiple seeds (≤ 5).