Exploration of DyNA PPO with Dynamic Ensemble

Leihao (Eric) Lin*
*Department of Computer Science, Western University
Email: llin286@uwo.ca

***Abstract*—This is the abstract of the paper.**

I. Introduction

Introduction of the project ...

II. Backgrounds and Related Works

Using PPO, a stable policy-gradient RL method to solve the black box optimization of biological sequence design. It proposes DyNA PPO, a variant of Proximal Policy Optimization:

- Learns a surrogate reward model through supervised regression on data collected.
- Using cross validated R2, selects from a pool of candidate regressors whose predictions are above a threshold, and uses their ensemble average as a simulator for policy updates.
- Falls back to model-free PPO when no accurate surrogate is available, avoiding model bias.
- Adds exploration bonus penalizing proposals too similar to past sequences to encourage diversity.

III. Goal and Objectives

1) Reproduce the standard DyNA PPO from the original paper.
2) Formulate the surrogate ensemble reward $r'(x)$ with weights w_i chosen to minimize a combination of surrogate bias and variance under cross-validation estimates.
3) Combine several surrogate models into one reward function:
$$r'(x) = \sum_{i=1}^{K} w_i f'_i(x).$$
4) Prove a bound on the regret of the model-based policy update step that decomposes into:
   a) model bias terms, and
   b) policy-optimization error,

   showing conditions under which weighted ensembling strictly improves sample efficiency over uniform averaging.
5) Define regret as the loss in reward by following the approximate surrogate-based policy update, compared to using the true fitness function at every step.
6) Decompose regret into:
   - How wrong the surrogate is, and
   - How imperfect our policy update on that surrogate is.

   This allows us to optimally choose model weights to shrink model bias, rather than equally averaging all models. As a result, the overall regret is smaller and fewer real samples are needed to learn an effective policy.
7) Implement the weighted DyNA PPO algorithm, integrating it into the existing PPO plus surrogate loop. Re-estimate weights each round automatically via a small convex optimization step.
8) Add an extra optimization step each round to resolve for the best ensemble weights.
9) Empirically compare the weighted DyNA PPO against the standard DyNA PPO on benchmark tasks.

IV. Methodology

V. Experiments and Results

A. Base Comparison

We compare the following approaches:

- Standard with average ensemble
- R^2-based weighted ensemble
- Dynamic optimal ensemble
- Pure PPO

Metrics to track:

- Final best reward achieved
- Convergence speed (rounds to reach 90% of best)
- Reward Variance

B. Ablation Study

We test the importance of each component by removing them individually. Configurations:

- no_warmup
- no_diversity_penalty
- fixed_threshold
- uniform_weights_only
- no_context_encoding

C. Model Contribution Analysis

We analyze the contribution of each model over time. For each round, we log:

- Individual model R^2 scores
- Assigned weights
- Prediction accuracy

Visualization: stacked area charts will show the weight distribution over time.

VI. Conclusion

Your conclusion goes here.