Learning to Plan via a Multi-Step Policy Regression Method

Stefan Wagner, Michael Janschek, Tobias Uelwer, Stefan Harmeling

Heinrich Heine University Düsseldorf, Germany

Problem Definition

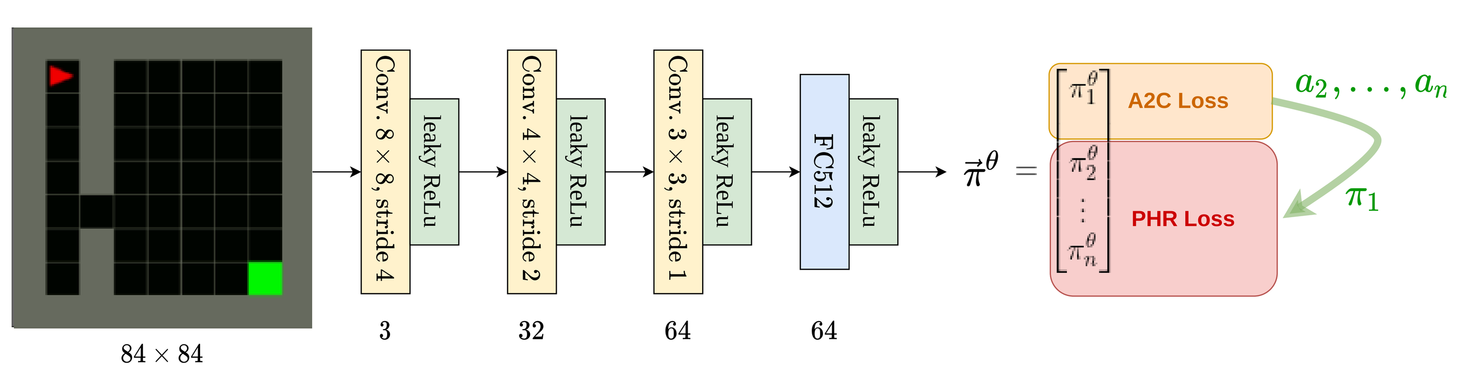
- New approach to increase inference performance in environments that require a specific sequence of actions in order to be solved.
- Policy horizon regression (PHR) learns a policy that can predict n actions in advance given an observation instead of one only action per observation.

Contributions

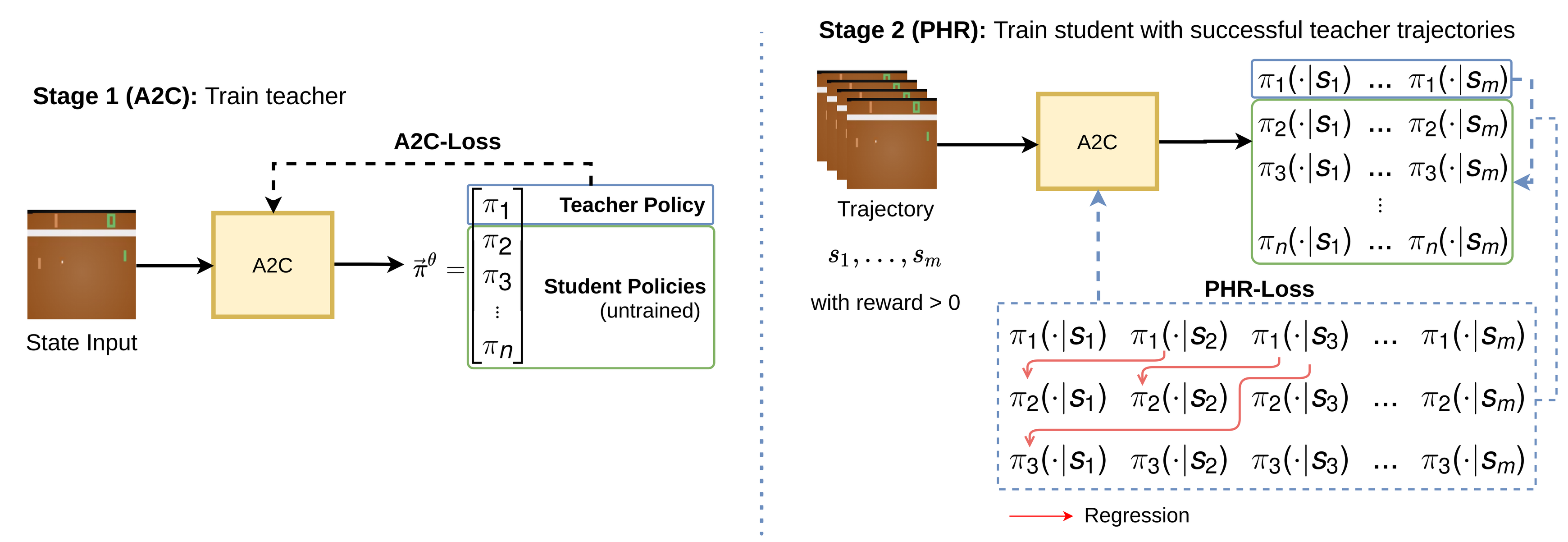
- Predicting the n dimensional action vector is much more efficient than evaluating the model n times. Thus, the agent completes the environments faster than its non-PHR counterpart.
- Useful in settings where agent has limited resources during inference time or where the agents productivity should be boosted.

Proposed Method

- Learn policy $\vec{\pi}^\theta$ that predicts n actions in advance i.e. the **policy vector** $\vec{\pi}^\theta = [\pi_1^\theta, \dots, \pi_n^\theta]^T$.
- Policy vector contains teacher policy and student policies $\pi_2^\theta, \dots, \pi_n^\theta$ that learn the optimal policy i.e actions from teacher policy π_1^θ.



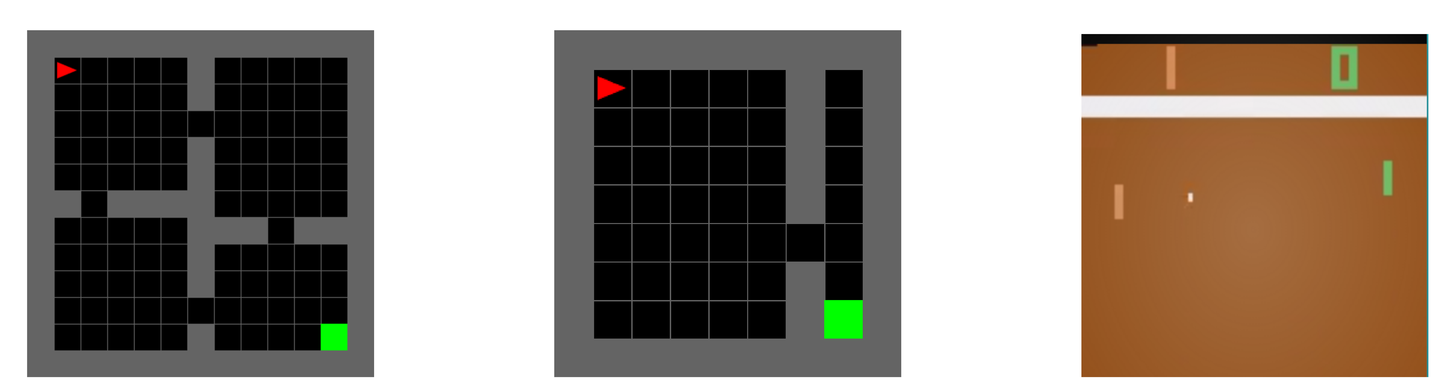
- (i) Environment is learned fully with A2C via regular policy gradient ascent $J_{\text{PG}}(\theta) = \mathbb{E}_{\pi_1}\left[\log \pi_1^\theta(s,a)\, Q^\theta(s,a)\right]$. π_1^θ will serve as *teacher policy* to learn the rest of the policy vector.
- (ii) Sample trajectories $D = \{(s_1, \pi_1(\cdot, s_1)), \dots, s_m, \pi_1(\cdot, s_m)), \dots\}$ with positive reward from π_1^θ, which will be regressed onto $\pi_2^\theta, \dots, \pi_n^\theta$, where s_m is terminal with reward $r_{m-1} > 0$.



- Take out sub-sequences $B_n = \{(s_t, \pi_1(\cdot, s_t), \dots, s_{t+n-1}, \pi_1(\cdot, s_{t+n-1})), \dots\}$ of length n from D with $1 \le t \le m - n + 1$.
- Minimize squared distance between teacher policies $\pi_1^{\theta'}(\cdot|s_i)$ and set of student policies $\pi_i^\theta(\cdot|s_t)$: $J_{\text{PHR}}(\theta, \theta') = \mathbb{E}_{\mathcal{D}}\left[\sum_{i=2}^n \left(\pi_i^\theta(\cdot|s_t) - \pi_1^{\theta'}(\cdot|s_i)\right)^2\right]$.
- Policy vector $\vec{\pi}^\theta$ learns to perform the same set of actions $a_2, \dots, a_n$ as π_1^θ just by looking at s_t.

Experiments

- Gym-minigrid [2]: Static(left) and stochastic(middle) gridworlds. Goal is to reach green square. Only here the agent receives a reward signal.
- Pong-Deterministic-v4 [1]: Test how PHR handles a more reactive agent.



- Agents evaluate environment every n steps i.e. perform n actions before evaluating the model again.
- Policy regression using MSE, KL-Divergence or Cross Entropy loss between actions.

Results

- PHR provides at least double the inference speedup in all 3 environments, only needing at least half the time to complete 100k steps.
- As policy quality is maintained, the agent is able to increase its throughput by the same factor.

Conclusions

- PHR is able to substantially speedup model inference and mantain policy quality thus increasing its throughput by the same factor.
- Opens PHR up to easy implementation in real-world applications with limited computing resources or productivity constraints.

References

[1] Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The Arcade Learning Environment: An Evaluation Platform for General Agents. *arXiv e-prints*, page arXiv:1207.4708, July 2012.

[2] Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. `https://github.com/maximecb/gym-minigrid`, 2018.

stefan.wagner@hhu.de, michael.janschek@hhu.de, tobias.uelwer@hhu.de, stefan.harmeling@hhu.de