Safer Reinforcement Learning through Evolved Instincts

Djordje Grbic
IT University Copenhagen
djgr@itu.dk

Sebastian Risi
IT University Copenhagen
sebr@itu.dk

ABSTRACT

An important goal in reinforcement learning is to create agents that can quickly adapt to new goals but at the same time avoid situations that might cause damage to themselves or their environments. One way agents learn is through exploration mechanisms, which are needed to discover new policies. However, in deep reinforcement learning, exploration is normally done by injecting noise in the action space. While performing well in many domains, this setup has the inherent risk that the noisy actions lead agents to unsafe environment states. In this paper, we introduce a novel approach called *Meta-Learned Instinctual Networks* (MLIN) that allows agents to perform lifetime learning while avoiding hazardous states. At the core of the approach is a plastic network trained through reinforcement learning and an evolved "instinctual" network, which does not change during the agent's lifetime but can modulate the noisy output of the plastic network. We test our idea on a simple 2D navigation task with hazard zones, in which the agent has to learn to approach new targets during deployment. While a standard meta-trained network performs poorly in these tasks, MLIN allows agents to learn to navigate to new targets while minimizing collisions with hazard zones. These results suggest that meta-learning augmented with an instinctual network is a promising approach for safe AI.

KEYWORDS

Reinforcement learning, Meta-learning, Lifetime Learning, AI Safety

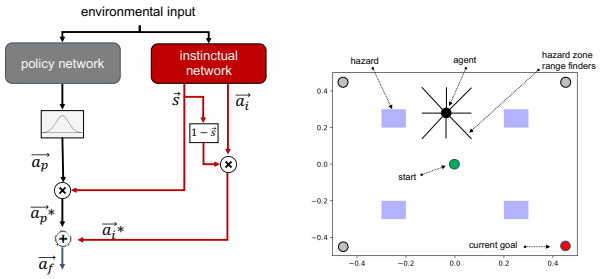
ACM Reference Format:
Djordje Grbic and Sebastian Risi. 2020. Safer Reinforcement Learning through Evolved Instincts. In *Genetic and Evolutionary Computation Conference Companion (GECCO '20 Companion), July 8–12, 2020, Cancún, Mexico.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377929.3389946

1 INTRODUCTION & RELATED WORK

Creating agents that can adapt quickly is one of the long-term goals in AI research. While current deep learning systems are good at learning a particular task, they still struggle to learn new tasks quickly; meta-learning tries to address this challenge. A recent trend in meta-learning is to find good initial weights through gradient-based optimization methods from which adaptation can be performed in a few iterations [2, 3]. While these meta-learning approaches allow agents to adapt faster, they do not take into account any safety constraints [1], which are states and behaviors the system should avoid. In contrast to previous work on safer exploration in RL [5], the approach presented here formulates safe learning in the context of meta-learning. The results in a simple 2D navigation domain (Fig. 1b) demonstrate that MLIN allows agents to learn to navigate to different target areas *during deployment* while avoiding hazardous areas in the environment. In the future, the idea of combining meta-learning with an instinctual network could enable safer forms of AI across a range of different tasks.

GECCO '20 Companion, July 8–12, 2020, Cancún, Mexico

ACM ISBN 978-1-4503-7127-8/20/07…$15.00
https://doi.org/10.1145/3377929.3389946



(a) (b)

Figure 1: (a) Model architecture, and **(b)** 2D-navigation environment with hazards.

2 MODEL ARCHITECTURE

The model architecture introduced in this paper consists of two neural network modules: a policy network and an instinctual network (Fig. 1a). The policy network is a neural network module that is trained to solve a specific task through reinforcement learning, while the instinctual network is kept fixed during task adaptation. The goal of the instinctual network is to override noisy actions of the policy network if the agent finds itself in potentially dangerous situations. The specific architecture described here is suitable for reinforcement learning problems with continuous action spaces. The modulation of the policy network follows the steps below:

(1) instinct network outputs two vectors, $\vec{s}$ and instinctual action $\vec{a_i}$, where $\vec{s} \in [0, 1]$
(2) policy networks outputs action $\vec{a_p}$;
(3) $\vec{a_p}$ gets modulated with the suppression vector, $\vec{a_p^*} = \vec{s} \odot \vec{a_p}$, where $\odot$ is the element-wise multiplication of vectors;
(4) $\vec{a_i^*} = \vec{a_i} \odot (\vec{1} - \vec{s})$;
(5) final action vector $\vec{a_f} = \vec{a_p^*} + \vec{a_i^*}$.

3 META-TRAINING

The question here is how to train an instinctual network that keeps the agent out of harm's way together with a policy network that should be able to adapt quickly to new goals. One of the main insights in the work presented here is that we can use an evolutionary meta-learning approach [2] to train a policy that can adapt quickly and *safely* to different tasks. The whole training procedure runs two

training loops: an evolutionary outer loop, and a task-adaptation inner loop.

In the outer evolutionary loop, a simple genetic algorithm (GA) is optimizing the initial weights of the policy network, the weights of the instinctual network, and a learning rate used by the RL algorithm in the inner loop. Importantly, the weights of the instinctual network are only updated through mutations during the outer loop and are not modified in the inner loop. In other words, instincts are fixed during an agent's lifetime.

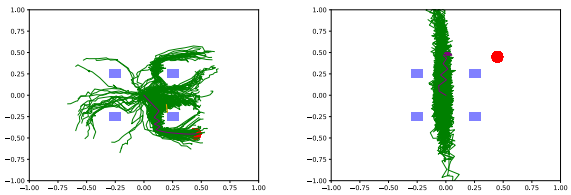
Our specific implementation uses the proximal policy optimization (PPO) algorithm [7] for the policy gradient calculation $\nabla\mathcal{L}(f_{\theta^p})$, and the Adam optimizer [4] for the gradient update of the policy network f with parameters θ^p. The PPO algorithm takes the action log-probabilities ($\log f_{\theta^p}(s, \vec{a_p})$) sampled from the policy network, not the final instinct-modulated actions $\vec{a_f}$.

After the gradient-based update performed in the inner loop, the algorithm samples the final trajectory where the policy network generates actions by taking the mean a_μ action of the $f_{\theta^p}(\cdot)$ distribution. The cumulative episode reward is added to the training hazard violation punishments to get the task evaluation. The policy weights optimized through the gradient update are discarded after each task (i.e. non-Lamarckian evolution). The final evaluation of the evolved parameters is the sum of task evaluations F_g for each task visited in the inner loop. The parameters θ_g^p (policy network weights), θ_g^i (instinct network weights) and α_g (gradient update learning rate) are optimized in the outer loop based on the evaluation values F_g.

4 TASK ENVIRONMENT

The test domain in this paper is a 2D navigation task with four hazardous areas (Fig. 1b), inspired by the simpler 2D navigation (without hazardous areas) used to evaluate the MAML algorithm [3]. The environment consists of an agent starting at the coordinate (0,0). The goal of the agent is to learn how to reach one of four goals $T_i \in [(\pm 0.45, \pm 0.45)]$ only through the reward it receives at each time step. The inner loop cycles through all four goals and rewards the agent for how close it can approach them. The agent does not know the location of the current goal and has to reach it only by adapting the policy through rewards. If the agent could see the goal through sensors, a static policy would be able to reach each goal without having to re-adapt, defeating the purpose of meta-learning.

The reward is based on the negative distance of the current position to the goal state $r_{d,t}$. A penalty of $r_{h,t} = -10$ is given for each timestep in one of the hazard zones. The total state reward is $R(s_t) = r_{d,t} + r_{h,t}$. An episode terminates if the agent gets within 0.01 units to the goal state or the episode exceeds a maximum of 20 timesteps (Horizon $H = 20$). The hazardous areas in the environment test the agent's ability to adapt to new goal positions in a safe way. The policy network and the instinctual network get the position the agent currently occupies (x, y) and the eight range-finders, which detect the proximity of the hazardous areas, as input. The range-finders see in directions: $(0, \pm 0.1)$, $(\pm 0.1, 0)$, $(\pm 0.1, \pm 0.1)$ around the agent. One range-finder returns the fraction of the distance that an edge of a hazardous zone occupies, in $[0, 1]$ range. The agent outputs a movement vector $(\Delta x, \Delta y) \in [-0.1, 0.1]^2$.



(a) Learning with Instincts **(b) Learning *without* instincts**

Figure 2: Exploration trajectories in the 2D navigation environment. The green lines show the exploration trajectories of the best meta-trained policies for 4000 steps. The purple line is the deterministic trajectory of the model after the first gradient update.

5 RESULTS & DISCUSSION

The instinctual network is able to avoid colliding with the hazards in the environment (Reward: -3.9 ± 1.5; collisions: 0.05 ± 0.2), while a network meta-trained without an instinct is not able to avoid collisions with these areas during the noisy rollouts (Reward: -13.3 ± 5.0; collisions: 0.2 ± 0.7). We also compared MLIN to a pure PPO-based reinforcement learning setup (with the same architecture as the MLIN network minus the instinct module) that has to learn to approach different targets starting from randomly initialized weights. We rolled out 4000 steps state-action samples which were used to perform one gradient update with PPO with the evolved learning rate (Reward: -13.3 ± 2.4; collisions: 75.9 ± 48.3).

Fig. 2 shows the exploration trajectories for the models meta-trained with and without instincts. While MLIN can quickly adapt to new goals, a policy meta-trained without the instinct network did not learn to approach any of the targets. Analysis of the instinct networks revealed that they have innate knowledge about the hazard positions in the environment. Future work will include applying the approach to more complicated domains, such as the recently suggested new safety benchmark by OpenAI [6].

ACKNOWLEDGMENTS

This work was supported by the Lifelong Learning Machines program from DARPA/MTO under Contract No. FA8750-18-C-0103. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA.

REFERENCES

[1] Eitan Altman. 1999. *Constrained Markov decision processes*. Vol. 7. CRC Press.
[2] Chrisantha Thomas Fernando, Jakub Sygnowski, Simon Osindero, Jane Wang, Tom Schaul, Denis Teplyashin, Pablo Sprechmann, Alexander Pritzel, and Andrei A Rusu. 2018. Meta Learning by the Baldwin Effect. *arXiv preprint arXiv:1806.07917* (2018).
[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400* (2017).
[4] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[5] Zachary C Lipton, Abhishek Kumar, Jianfeng Gao, Lihong Li, and Li Deng. 2016. Combating deep reinforcement learning's sisyphean curse with reinforcement learning. *arXiv preprint arXiv:1611.01211* (2016).
[6] Alex Ray, Joshua Achiam, and Dario Amodei. [n. d.]. Benchmarking Safe Exploration in Deep Reinforcement Learning. ([n. d.]).
[7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. (2017). arXiv:cs.LG/1707.06347