ES-MAML: Learning to Adapt with Evolution Strategies

Xingyou Song*[1], Wenbo Gao*[2], Yuxiang Yang[1], Krzysztof Choromanski[1], Aldo Pacchiano[3], Yunhao Tang[2]

*equal contribution [1]Google Brain [2]Columbia University [3]UC Berkeley

Meta-Learning with Evolution Strategies

Meta-learning is a paradigm for models and training algorithms that can adapt to new tasks. *Model agnostic meta learning* (MAML) [1] is a framework for meta-learning that allows a meta-policy to *adapt* to a given task T through an adaptation operator $U(\theta, T)$. The meta-policy is learned by maximizing the *expected reward after adaptation*.

$$\max_{\theta} J(\theta) := \mathbb{E}_T R^T(U(\theta, T))$$

The standard adaptation operator is to take a gradient step for the task T: $U(\theta, T) = \theta + \alpha \nabla R^T(\theta)$. This makes any first-order algorithm for MAML a *second-order method* with respect to the task rewards R^T. Expressed in terms of expectations over trajectories τ generated by policy θ, we have

$$\nabla_\theta U(\theta, T) = I + \alpha \int \mathcal{P}_T(\tau|\theta) \nabla_\theta^2 \log \pi_\theta(\tau) R^T(\tau) d\tau + \alpha \int \mathcal{P}_T(\tau|\theta) \nabla_\theta \log \pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau)^T R^T(\tau) d\tau$$

This is notoriously hard to estimate accurately with automatic differentiation, leading to the development of enhanced methods such as ProMP [2] and T-MAML [3]. But these are also complicated!

Instead, we *avoid* the difficulty of estimating $\nabla_\theta U(\theta, T)$ by applying *evolution strategies* (ES) to MAML.

$$J_\sigma(\theta) := \mathbb{E}_g J(\theta + \sigma g) \qquad \text{The } \textit{Gaussian smoothing } (g \sim N(0, I)) \text{ of } J(\theta)$$
$$\nabla J_\sigma(\theta) = \mathbb{E}_g[J(\theta + \sigma g) g] \qquad \text{The gradient of } J_\sigma$$

Now the gradient $\nabla J_\theta(\sigma)$ can be estimated using only zero-order evaluations of $J(\theta + \sigma g)$.

Furthermore: we can use new adaptation operators which may even be nonsmooth, such as hill climbing (local argmax). We can also use any enhancements of the ES algorithm.

Algorithm

Data: initial policy θ_0, meta step size β
for $t = 0, 1, \ldots$ **do**
  Sample n tasks $T_1, \ldots, T_n$ and iid vectors $\mathbf{g}_1, \ldots, \mathbf{g}_n \sim \mathcal{N}(0, \mathbf{I})$;
  foreach $(T_i, \mathbf{g}_i)$ **do**
    $v_i \leftarrow f^{T_i}(U(\theta_t + \sigma \mathbf{g}_i, T_i))$
  end
  $\theta_{t+1} \leftarrow \theta_t + \frac{\beta}{\sigma n} \sum_{i=1}^n v_i \mathbf{g}_i$
end

Algorithm 1: Zero-Order ES-MAML (general adaptation operator $U(\cdot, T)$)

Data: initial policy θ_0, adaptation step size α, meta step size β, number of queries K
for $t = 0, 1, \ldots$ **do**
  Sample n tasks $T_1, \ldots, T_n$ and iid vectors $\mathbf{g}_1, \ldots, \mathbf{g}_n \sim \mathcal{N}(0, \mathbf{I})$;
  foreach $(T_i, \mathbf{g}_i)$ **do**
    $\mathbf{d}^{(i)} \leftarrow \text{ESGrad}(f^{T_i}, \theta_t + \sigma \mathbf{g}_i, K, \sigma)$;
    $\theta_t^{(i)} \leftarrow \theta_t + \sigma \mathbf{g}_i + \alpha \mathbf{d}^{(i)}$;
    $v_i \leftarrow f^{T_i}(\theta_t^{(i)})$;
  end
  $\theta_{t+1} \leftarrow \theta_t + \frac{\beta}{\sigma n} \sum_{i=1}^n v_i \mathbf{g}_i$;
end

Algorithm 2: Zero-Order ES-MAML with ES-Gradient Adaptation

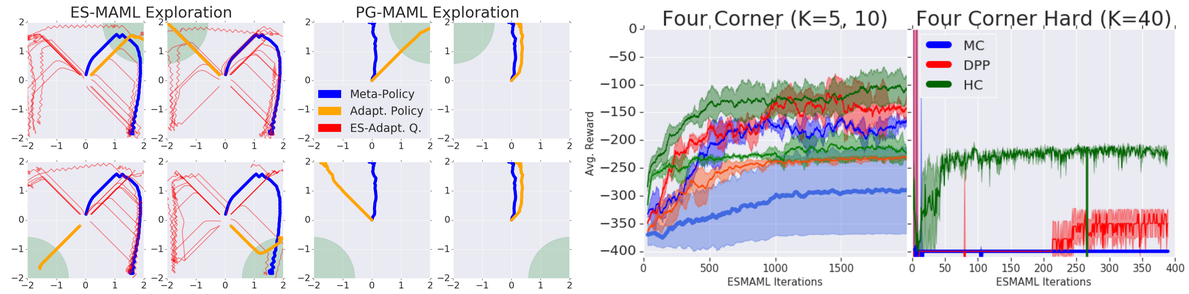
References

[1] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1126–1135, 2017.

[2] Jonas Rothfuss, Dennis Lee, Ignasi Clavera, Tamim Asfour, and Pieter Abbeel. Promp: Proximal meta-policy search. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[3] Hao Liu, Richard Socher, and Caiming Xiong. Taming MAML: efficient unbiased meta-reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 4061–4071, 2019.

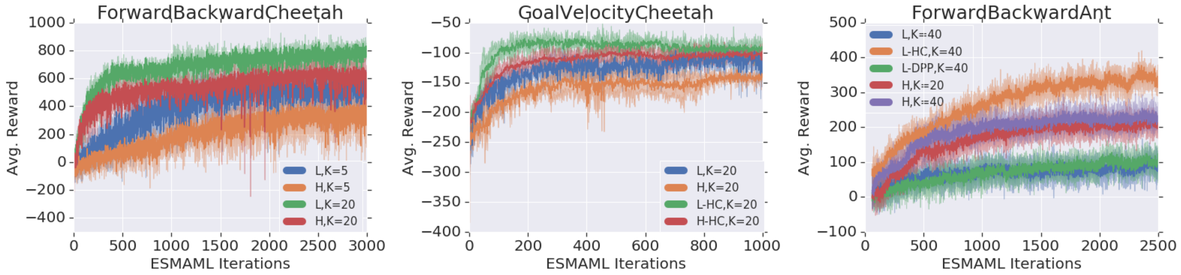
Exploration in Sparse Environments



The *four corners* task requires the agent to locate a target corner. The meta-policy learned by ES moves in different directions when perturbed, allowing it to explore and find the correct corner. In comparison, policy gradient MAML without modification is unable to explore sufficiently to locate the target.

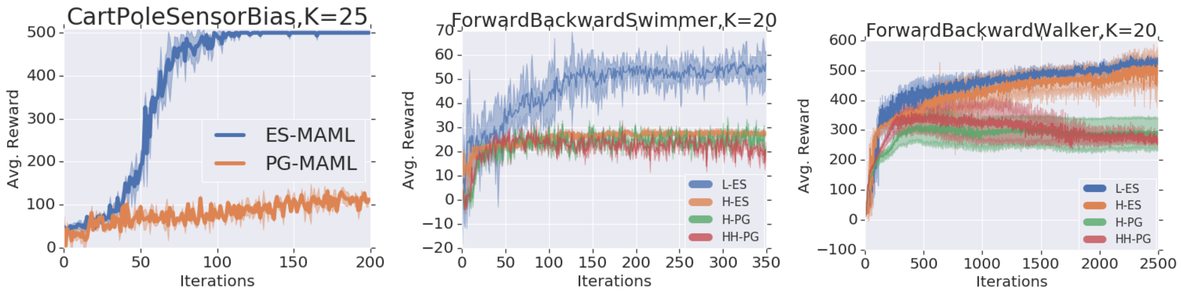
Exploratory behavior in policy gradient MAML arises from the stochasticity of the policy; it is difficult for a single meta-policy to generate efficient exploration (in this case, circular trajectories). In contrast, ES-MAML generates exploration from perturbations, which is more effective.

Compact Architectures



ES-MAML works well when using *linear* or *compact* NN policies. Simpler architectures are faster to train and faster to execute on possibly limited controller hardware.

Deterministic Policies



Deterministic policies produce predictable and more stable behavior than stochastic policies, and randomized actions can lead to catastrophic outcomes in real life. ES-MAML explores in parameter space, which helps to mitigate this issue and makes it safer for robotics applications. In contrast, PG-MAML relies on action randomization to generate exploration, and thus can only use stochastic policies.