POET

Tuesday, 13 October 2020 12:35

- Paired Open-Ended Trailblazer (POET)
  Endlessly generating increasingly complex and diverse learning environments and their solutions.
- POET simultaneously explores many different paths through the space of possible problems and solutions, and critically, allows those stepping-stone solutions to transfer between problems.
- The ability to transfer solutions from one environment to another proves essential to unlocking the potential of the system.
- POET owes its inspiration to:
  1. Minimal Criterion Coevolution [Jonathan '17].
     MCC shows that the environments and solutions can effectively co-evolve.
  2. MAP-Elite - Illuminating search spaces by mapping elites [Mouret '15]; Understanding innovation engines [Nguyen '16]:
     exploit the opportunity to transfer high-quality solutions from one objective among many to another.
  3. CMOEA (Combinatorial Multi-Objective Evolutionary Algorithm)
     Evolving multimodal robot behavior via many stepping stones [Huizinga '18]
     extends the innovation engine to combinatorial tasks.
- Background:
  1. Behavioral diversity and stepping stones:
     Problem to solve: becoming trapped on local optima in stochastic optimization and search algorithms.
     Scenarios for the problem:
     1. domains or problems evolving over the course of search could stop increasing their complexity
        → stop becoming increasingly interesting.
     2. simultaneously-optimizing solutions could be stuck at sub-optimal levels and fail to solve challenges that are solvable.

     Possible Solutions:
     1. Population-based algorithms; Novelty search(NS) [Lehman'11]
        - encourage behavioral diversity (less susceptible to local optima).
        open-endness encourages divergence instead of convergence.
        - divergent algorithms reward and perserve diverse behaviors;
        to facilitate the preservation of potential stepping stones
        → pave the way ① to a solution to a particular problem and ② to genuinly open-ended search.
     2. Quality Diversity algorithms (QD) [Pugh '16]
        - based on goal switching; tries to discover stepping stones by periodically testing the performance of offspring from one niche in other niches.
        e.g. Innovative Engine DNN on ImageNet.
        CMOEA multimodal robotic behaviors.

     → POET harness goal switching within divergent search.
  2. Open-Ended search via Minimal Criterion Coevolution (MCC):
     Problem to solve: Environments remains static in NS & QD;
     limiting the scope of what can be found in the long run.

     Possible solutions:
     1. Coevolution: Coevolutionary principles [Paparizi '12]
        - different individuals in a population interact with each other while they are evolving.
        e.g. GAN, self-play
        - but coevolutionary systems the abiotic (non-opanent-based) aspect of the environment remains fixed.
     2. Minimal Criterion Coevolution (MCC):
        - pairs a population of evolving problem (i.e. environmental changes) with a coevolving population of solutions.
        - with a new kind of evolution; evolves two interlocking populations; whose members earn the right to reproduce by satisfying a minimal criterion w.r.t the other population; as both populations are gradually shifting.
  3. Evolution Strategies (ES) [Rechenberg '78]
     Problem to solve: in MCC there is no force for optimization within each environment.

     Possible solution:
     - ES to play the role of the optimizer.
       population-based optimization algorithm
       popularized by: ES as a scalable alternative to RL [Salimans '17]
     - In the context of RL:
       $E(\cdot)$ - the environment
       ω - the parameters of the policy
       $E(\omega)$ - the reward
       the objective: $\max_{\omega} E(\omega)$
     - In ES:
       $E(\omega)$ - the stochastic reward experienced over a full episode
       the objective: maximize the expected fitness over a population of ω.
       ω is sampled from a probability distribution $p_\theta(\omega)$ parametrized by θ. ($\omega \to \theta$;)
       the parameters of each agent θ_i is sampled $\theta_i \sim \mathcal{N}(\theta, \sigma^2 I)$
       or by using additive Gaussian noise $\theta_i = \theta + \sigma\epsilon_i$: $\epsilon_i \sim \mathcal{N}(0, I)$
     - the objective is to maximize $J(\theta)$:

$$J(\theta) = \mathbb{E}_{\omega \sim p_\theta(\omega)}[E(\omega)] = \int p_\theta(\omega)\, E(\omega)\, d\omega$$

     the derivative w.r.t θ

$$\nabla_\theta J(\theta) = \nabla_\theta \int p_\theta(\omega) E(\omega) d\omega = \int E(\omega)\, \nabla_\theta p_\theta(\omega)\, d\omega$$

     by using log-likelihood trick ($\nabla_\theta \log p_\theta(\omega) = \frac{\nabla_\theta p_\theta(\omega)}{p_\theta(\omega)}$)

$$\nabla_\theta J(\theta) = \int E(\omega) \frac{p_\theta(\omega)}{p_\theta(\omega)} \nabla_\theta p_\theta(\omega)\, d\omega = \int E(\omega)\, p_\theta(\omega)\, \nabla_\theta \log p_\theta(\omega)\, d\omega = \mathbb{E}_{\omega \sim p_\theta(\omega)}[E(\omega)\, \nabla_\theta \log p_\theta(\omega)]$$

     estimating the expectation over n samples (Monte Carlo):

$$\nabla_\theta J(\theta) = \frac{1}{n} \sum_{i=1}^{n} E(\theta_i)\, \nabla_\theta \log p_\theta(\theta_i)$$

     $\nabla_\theta \log p_\theta(\theta_i)$ - the likelihood of sampling the i-th agent

     when using additive noise:

$$\mathbb{E}_{\omega \sim p_\theta(\omega)}[E(\omega)] = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)}[E(\theta + \sigma\epsilon)]$$

$$\Rightarrow \nabla_\theta \mathbb{E}_{\omega \sim p_\theta(\omega)}[E(\omega)] = \nabla_\theta \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)}[E(\theta + \sigma\epsilon)] = \nabla_\theta \int p(\epsilon)\, E(\theta + \sigma\epsilon)\, d\epsilon$$

$$p(\epsilon) = (2\pi)^{-\frac{n}{2}} \exp\left(-\tfrac{1}{2}\epsilon^T\epsilon\right)$$

$$= \int p(\epsilon)\, \nabla_\epsilon \log p(\epsilon)\, \nabla_\theta \tfrac{\epsilon}{\sigma}\, E(\theta + \sigma\epsilon)\, d\epsilon \quad \text{(log-likelihood trick)}$$

$$= \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)}\left[\nabla_\epsilon \left(-\tfrac{1}{2}\epsilon^T\epsilon\right) \nabla_\theta \left(\tfrac{\omega - \theta}{\sigma}\right) E(\theta + \sigma\epsilon)\right]$$

$$= \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)}\left[(-\epsilon)\left(\tfrac{1}{\sigma}\right) E(\theta + \sigma\epsilon)\right] = \frac{1}{\sigma}\, \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)}[\epsilon F(\theta + \sigma\epsilon)]$$

     for n samples:

$$\nabla_\theta J(\theta) = \frac{1}{n\sigma} \sum_{i=1}^{n} E(\theta + \sigma\epsilon_i)\, \epsilon_i$$

     → we can sample many ϵ_i and evaluate the fitness in parallel.

- POET:
  ① Maintains: 1. a population of environments.
  2. a population of agents.
  → each environment is paired with an agent
  ② In the spirit of MCC; POET implements an ongoing divergent coevolutionary interaction among all its agents and environments.
  ③ In the spirit of CMOEA; aims to optimize the behavior of each agent within its paired environment
  ④ POET elaborates the minimal criterion of MCC; aiming to maintain only newly-generated environments that are not too hard and not too easy for the current population of agents.

- POET cannot guarantee reaching a particular preconceived target.
- In the experiment;
  3 runs of POET ≈ 25,200 POET iterations
  (population size 20 active environment
  & number of samples for ES 512)
  that takes 10 days on 256 CPU cores.