Evolution Strategies as a Scalable Alternative to Reinforcement Learning

Yoonho Lee

Department of Computer Science and Engineering
Pohang University of Science and Technology

July 12, 2017

Deep Learning for RL

Different rules apply:

- tanh works better than relu on certain problems[5]
- Some state-of-the-art architectures use ≤ 3 layers
- Parallelization[2] gives more performance boost than sophisticated methods[5] or architectures[6]
- Memorizing everything is a viable strategy[1]

This paper suggests not using backprop(!)

ES Pseudocode

Algorithm 1 Evolution Strategies

Input: Learning rate α, noise standard deviation σ, initial policy parameters θ_0
for $t = 0, 1, 2, \ldots$ **do**
    Sample $\epsilon_1, \ldots \epsilon_n \sim \mathcal{N}(0, I)$
    Compute returns $F_i = F(\theta_t + \sigma\epsilon_i)$ for $i = 1, \ldots, n$
    Set $\theta_{t+1} \leftarrow \theta_t + \alpha \frac{1}{n\sigma} \sum_{i=1}^{n} F_i \epsilon_i$
end for

- Score function estimator on trajectory returns in parameter space

Parallel ES Pseudocode

Algorithm 2 Parallelized Evolution Strategies

Input: Learning rate α, noise standard deviation σ, initial policy parameters θ_0

Initialize: n workers with known random seeds, and initial parameters θ_0

for $t = 0, 1, 2, \ldots$ **do**

  for each worker $i = 1, \ldots, n$ **do**

    Sample $\epsilon_i \sim \mathcal{N}(0, I)$

    Compute returns $F_i = F(\theta_t + \sigma\epsilon_i)$

  end for

  Send all scalar returns F_i from each worker to every other worker

  for each worker $i = 1, \ldots, n$ **do**

    Reconstruct all perturbations ϵ_j for $j = 1, \ldots, n$

    Set $\theta_{t+1} \leftarrow \theta_t + \alpha \frac{1}{n\sigma} \sum_{j=1}^{n} F_j \epsilon_j$

  end for

end for

Results

scores

Table 1. MuJoCo tasks: Ratio of ES timesteps to TRPO timesteps needed to reach various percentages of TRPO's learning progress at 5 million timesteps.

Environment	25%	50%	75%	100%
HalfCheetah	0.15	0.49	0.42	0.58
Hopper	0.53	3.64	6.05	6.94
InvertedDoublePendulum	0.46	0.48	0.49	1.23
InvertedPendulum	0.28	0.52	0.78	0.88
Swimmer	0.56	0.47	0.53	0.30
Walker2d	0.41	5.69	8.02	7.88

Results

sensitivity to frameskip

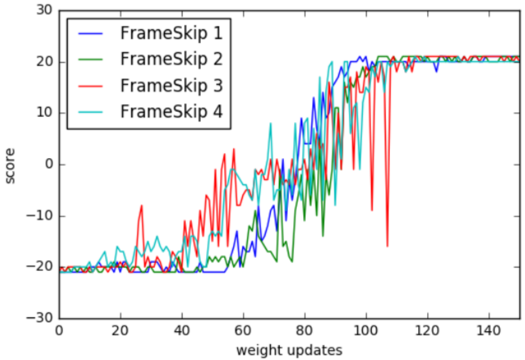


Figure 2. Learning curves for Pong using varying frame-skip parameters. Although performance is stochastic, each setting leads to about equally fast learning, with each run converging in around 100 weight updates.

Results

speedup

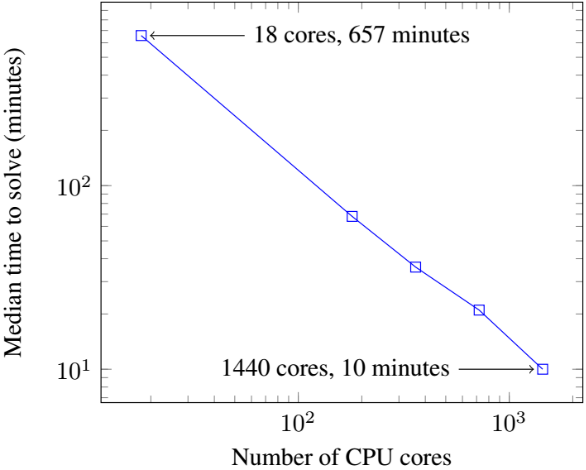
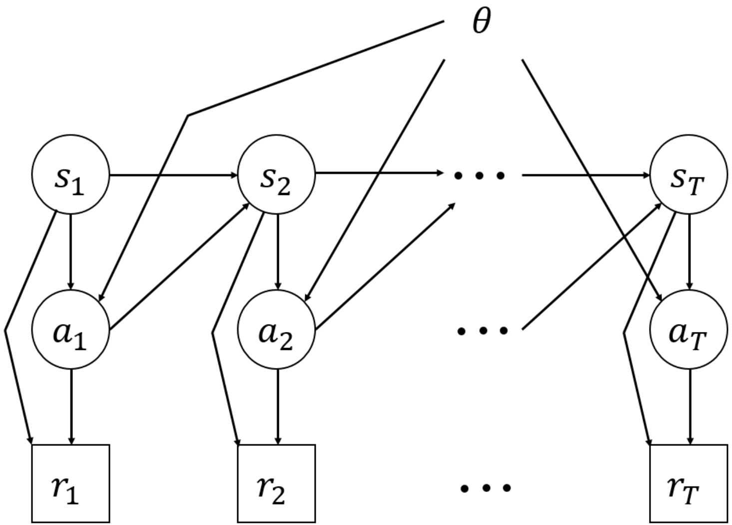
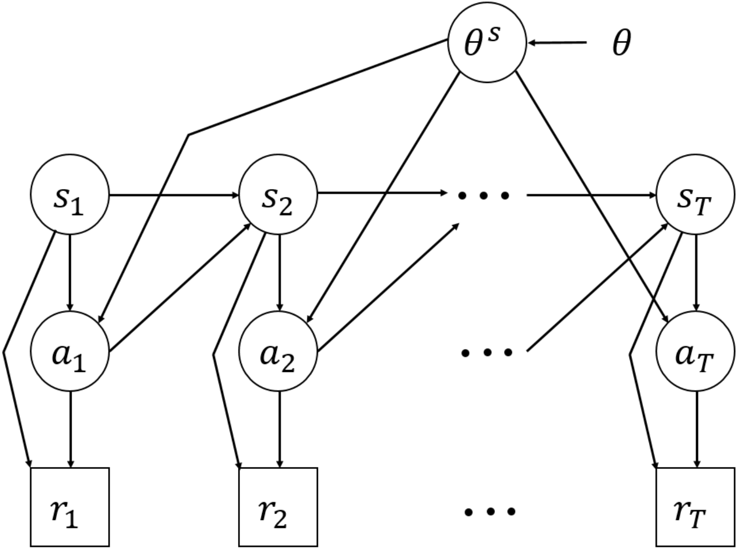


Figure 1. Time to reach a score of 6000 on 3D Humanoid with different number of CPU cores. Experiments are repeated 7 times and median time is reported.

Policy Gradient vs ES



Policy Gradient vs ES



Strengths of ES

- Only affected by intrinsic dimensionality of problem itself
- Only needs to communicate one scalar signal between devices: massively parallelizable
- Does not require backprop: we can use non-differentiable elements such as step function activations or hard attention[7]
- All tricks that work with backprop SGD still work here (Adam, batchnorm etc.)

Discussion

- Should we be surprised that this works, or have we been overestimating our benchmark tasks?
- If backprop isn't required for competitive RL performance, what made deep RL work better than traditional RL in the first place?
- Can we find a middle ground between ES and policy gradients? Define 'episodes' to be shorter subsequences?
- Where else can the 'parallelize via communication of random seeds' idea be used?

References I

[1] Charles Blundell et al. "Model-Free Episodic Control". In: (2016). URL: https://arxiv.org/pdf/1606.04460.pdf.

[2] Volodymyr Mnih et al. "Asynchronous Methods for Deep Reinforcement Learning". In: *ICML* (2016).

[3] Tim Salimans et al. "Evolution Strategies as a Scalable Alternative to Reinforcement Learning". In: (2017). URL: https://arxiv.org/pdf/1703.03864.pdf.

[4] John Schulman et al. "Gradient Estimation Using Stochastic Computation Graphs". In: *NIPS* (2015). URL: https://arxiv.org/pdf/1506.05254.pdf.

[5] John Schulman et al. "Trust Region Policy Optimization". In: *ICML* (2015).

[6] Ziyu Wang et al. "Dueling Network Architectures for Deep Reinforcement Learning". In: *ICML* (2016).

References II

[7] Kelvin Xu et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *ICML* (2015).

POSTECH

Thank You