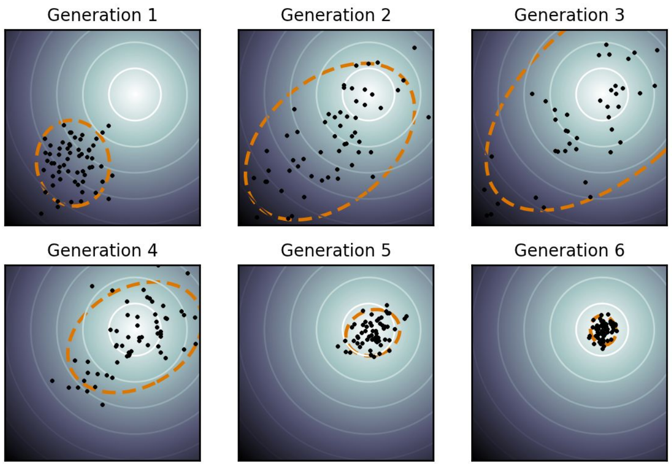
Evolution Strategies as a Scalable Alternative to RL

Cons

Evolution? Just random search with hill climbing

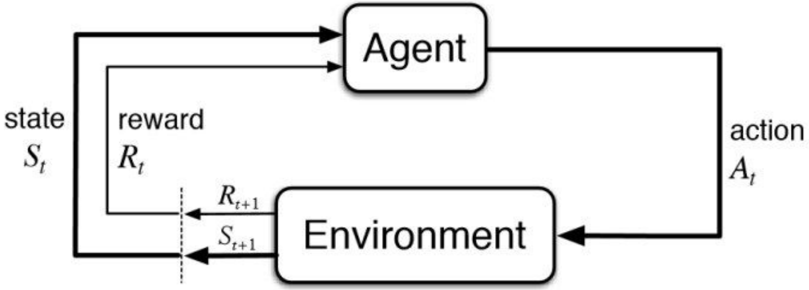
- No persistent population or elites
    - No history or memory like CMA-ES
- No mutation
- No crossover



Alternative to Reinforcement Learning?

Wikipedia:

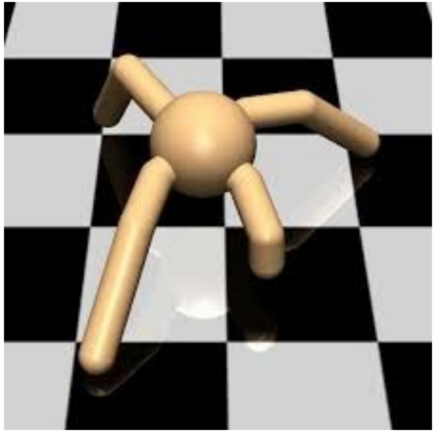
“**Reinforcement learning** (**RL**) is an area of machine learning concerned with how software agents ought to take *actions* in an *environment* so as to maximize some notion of cumulative *reward*. “



Better:

Alternative to policy gradients algorithms and value function approximation?

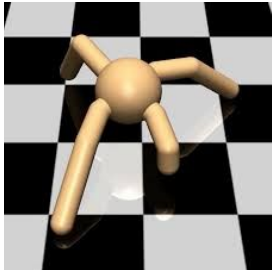
“...hardest environments studied by the deep RL community today...”



Benchmarks are easily solved with random search

- Uber AI -- pure random search over conv nets can beat RL
  - Such, Felipe Petroski, et al. "Deep neuroevolution: genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning." *arXiv preprint arXiv:1712.06567* (2017).

- Ben Recht -- random search with linear controllers beat RL
  - Mania, Horia, Aurelia Guy, and Benjamin Recht. "Simple random search provides a competitive approach to reinforcement learning." *arXiv preprint arXiv:1803.07055* (2018).



“We found the evolution strategies method to be robust”

- Used fixed hyperparameters
  - Sensitivity analysis?
- Results are pretty hit and miss

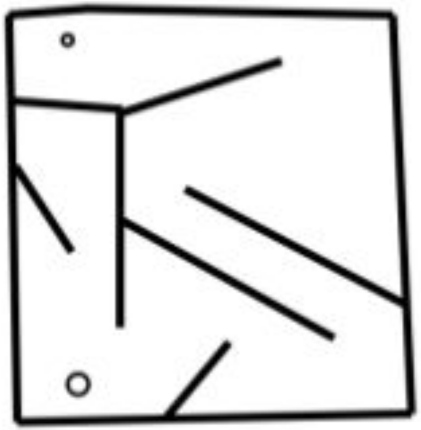
Game	DQN	A3C FF, 1 day	HyperNEAT	ES FF, 1 hour	A2C FF
Amidar	133.4	283.9	184.4	112.0	**548.2**
Assault	3332.3	**3746.1**	912.6	1673.9	2026.6
Asterix	124.5	**6723.0**	2340.0	1440.0	3779.7
Asteroids	697.1	**3009.4**	1694.0	1562.0	1733.4
Atlantis	76108.0	772392.0	61260.0	1267410.0	**2872644.8**
Bank Heist	176.3	**946.0**	214.0	225.0	724.1
Battle Zone	17560.0	11340.0	**36200.0**	16600.0	8406.2
Beam Rider	8672.4	**13235.9**	1412.8	744.0	4438.9
Berzerk		**1433.4**	1394.0	686.0	720.6
Bowling	41.2	36.2	**135.8**	30.0	28.9
Boxing	25.8	33.7	16.4	49.8	**95.8**
Breakout	303.9	**551.6**	2.8	9.5	368.5
Centipede	3773.1	3306.5	**25275.2**	7783.9	2773.3
Chopper Command	3046.0	**4669.0**	3960.0	3710.0	1700.0
Crazy Climber	50992.0	**101624.0**	0.0	26430.0	100034.4
Demon Attack	12835.2	**84997.5**	14620.0	1166.5	23657.7
Double Dunk	**21.6**	0.1	2.0	0.2	3.2
Enduro	**475.6**	82.2	93.6	95.0	0.0
Fishing Derby	2.3	13.6	**49.8**	49.0	33.9
Freeway	25.8	0.1	29.0	**31.0**	0.0
Frostbite	157.4	180.1	**2260.0**	370.0	266.6
Gopher	2731.8	**8442.8**	364.0	582.0	6266.2
Gravitar	216.5	269.5	370.0	**805.0**	256.2
Ice Hockey	3.8	4.7	**10.6**	4.1	4.9
Kangaroo	2696.0	106.0	800.0	**11200.0**	1357.6
Krull	3864.0	8066.6	**12601.4**	8647.2	6411.5
Montezuma’s Revenge	50.0	**53.0**	0.0	0.0	0.0
Name This Game	5439.9	5614.0	**6742.0**	4503.0	5532.8
Phoenix		**28181.8**	1762.0	4041.0	14104.7
Pit Fall		**123.0**	0.0	0.0	8.2

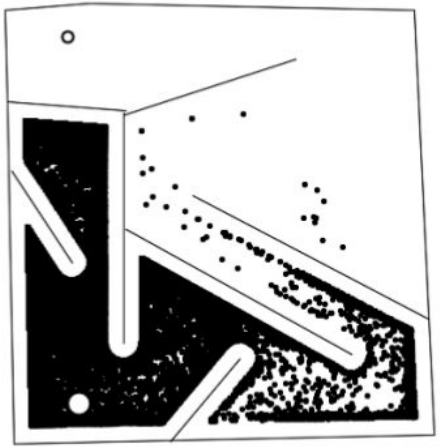
Not as simple as it seems...

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

- Mirrored sampling
  - -epsilon, +epsilon
- Uses fitness ranks rather than returns
- Requires virtual batch normalization (??) to work

Deceptive optimization problems?





- Lehman, Joel, and Kenneth O. Stanley. "Exploiting open-endedness to solve problems through the search for novelty." *ALIFE*. 2008.

Brute force search

- Requires massive parallelization
- Can require up to 10x as much data
- Wasteful, it just throws away rollouts after computing returns

Is it applicable to real problems?