Improving the Performance of Evolutionary Algorithms in Deep Reinforcement Learning via Gradient-Based Initialization

Chris Waites, Andrew Shoats, Matthew Prelich

Introduction

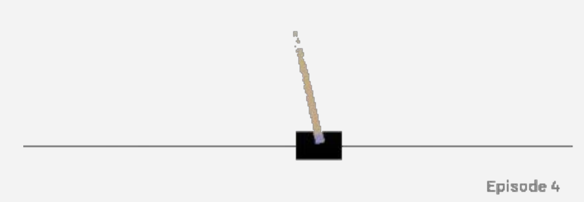
Gradient-based methods in RL are able to move quickly from regions of initialization to broad regions of lower loss

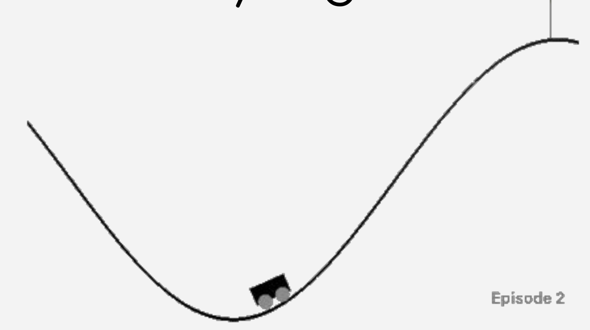
- Struggle to escape locally optimal solutions

Genetic algorithms place a heightened emphasis on global search and can reach superior solutions

- Take a long time to obtain reasonable candidate solutions

Idea: propose algorithm which takes hybrid approach, combining gradient-based methods in RL with gradient-free evolutionary algorithms





Methodology

N DQN models are trained using the following update rule to initialize GA population

$$\theta_{t+1} = \theta_t - \nabla_{\theta_t}\left(Q_{\theta_t}(s_t, a_t) - (r_t + \gamma \max_{a_t'} Q_{\theta_t}(s_t', a_t'))\right)^2$$

GA evolves a population of N individuals through what are called generations

The best performing network in the generation is preserved for the next generation. A parameter selected uniformly at random from the top T performing networks is mutated by adding Gaussian noise

In effect, the best performing networks are passed down by generation and thus the networks keep improving as a function of generations

Experiments

Quantitative Evaluation

Table 1: Expected cumulative reward for given episode limit, averaged over 20 algorithm initializations

CartPole-v1										
Algorithm	50	100	150	200	250	300	350	400	450	500
DQN	51.94	138.55	162.73	177.85	181.92	229.35	320.41	393.59	435.27	483.21
EA	21.91	26.29	27.73	31.545	35.205	36.835	41.635	47.65	52.27	57.725
Hybrid	70.16	142.21	183.19	223.53	275.69	380.71	480.12	500.00	500.00	500.00

Table 2: Expected number of episodes to achieve given reward threshold, averaged over 20 algorithm initializations

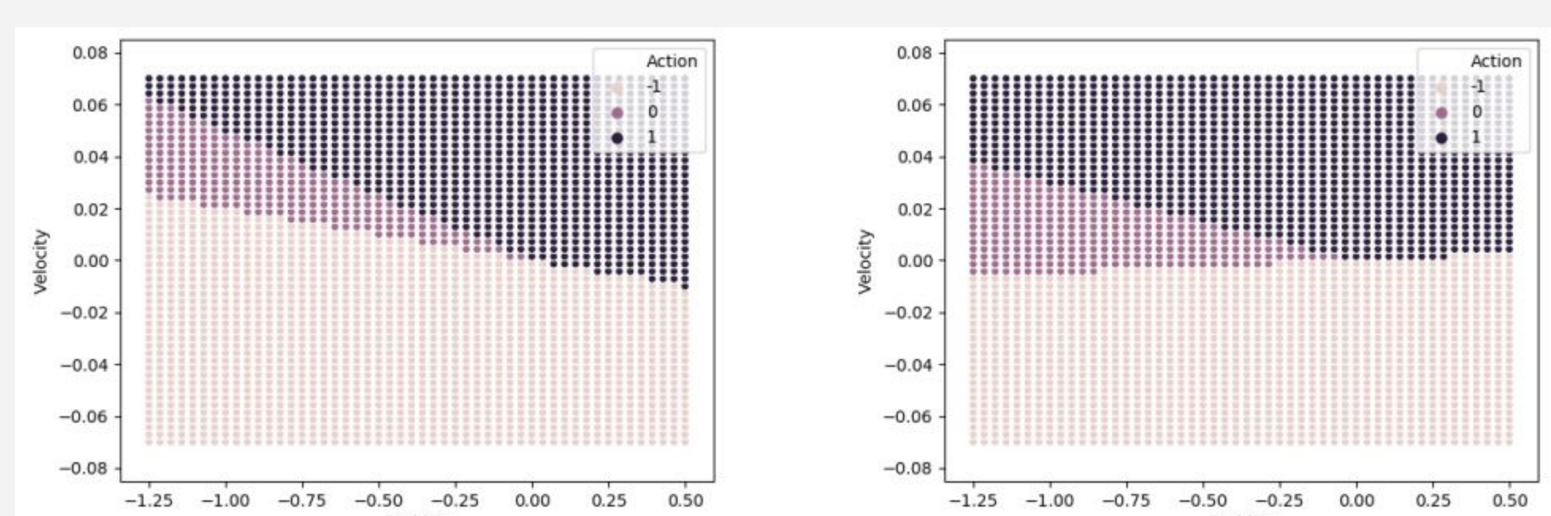
CartPole-v1										
Algorithm	50	100	150	200	250	300	350	400	450	500
DQN	48	84	121	284	327	340	376	405	478	531
EA	970	1875	2490	3000	3614	4012	4211	4397	4591	1623
Hybrid	21	23	27	29	30	32	34	37	40	41

Table 3: Expected CPU runtime to achieve given reward threshold, averaged over 20 algorithm initializations

CartPole-v1										
Algorithm	50	100	150	200	250	300	350	400	450	500
DQN	2	5	9	17	22	45	60	93	123	162
EA	3	6	13	28	53	94	130	153	172	210
Hybrid	3	5	11	24	37	51	87	125	153	188

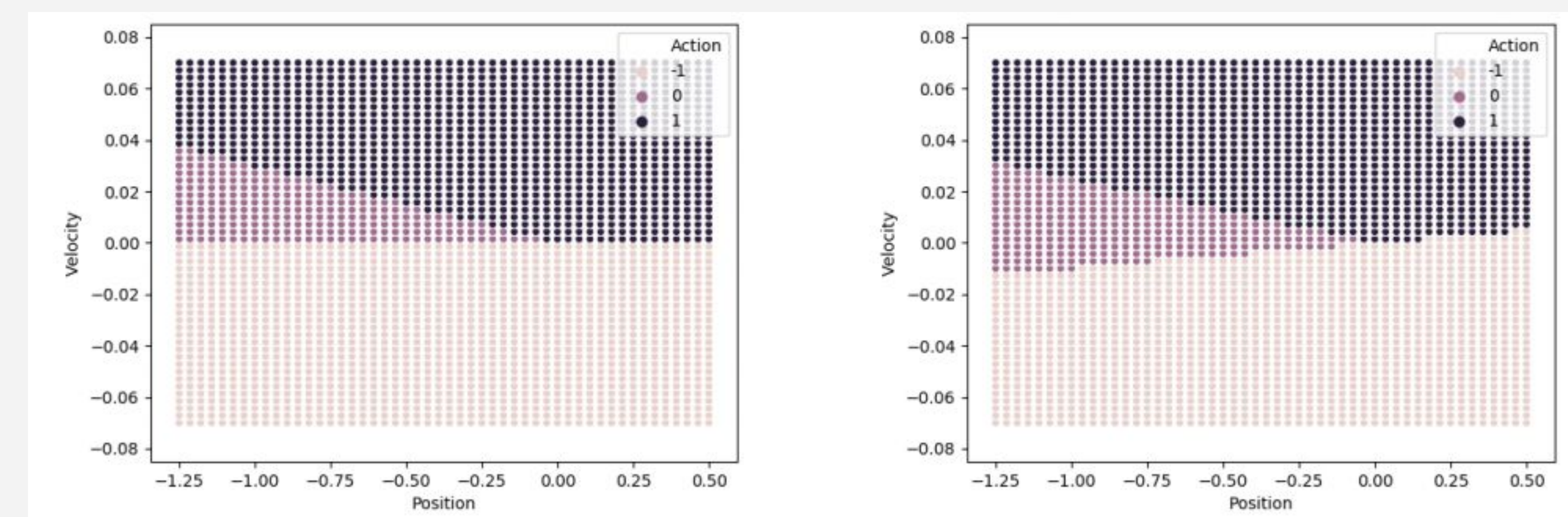
Qualitative Evaluation

- Policy Visualization



Findings

Overall, our work introduces a hybrid algorithm which effectively surpasses DQN (with Adam optimizer) and GAs alone in the popular RL bechmark of CartPole terms of both expected reward and time complexity.



These results can be even further improved due to the highly parallelizable nature of the algorithm

A 100-fold decrease in the number of forward propagations per generation was seen (this should be more methodically analyzed in future work) in comparison to GA without DQN initializations

Future Work

1. Apply methods to more challenging environments (Atari, MuJoCo, etc.)
2. Investigate generalization of proposed method which incorporate mix of gradient-based and non gradient-based approaches
3. Strongly supervised learning: Would the same positive results hold? Or would strictly gradient-based methods still reign?