1 SOLVING THE REACHER ENVIRONMENT

1.1 LEARNING ALGORITHM

The algorithm is a Proximal Policy Optimization using a Critic Network as a baseline. So, the network is composed with:

- A Gaussian Actor Network
- A Fully Connected Critic Network

The pseudo code is as follow:

- Repeat until the environment is solved
  - Gather N-Trajectories of max_t timesteps where N is the number of parallel agents using the Actor Network
    - Read the state of the environment and request for the actions to be done from the Actor
    - Clip the Actions to [-1,1]
    - Store the state, actions, log_prob of the actions, rewards, and next_state
  - Calculate the cumulated discounted rewards of each timestep of each agent
    - The return of the last timestep is based on the evaluation of the Critic Network
      - $R_t = r_t + \text{gamma} * V(s_{t+1})$
    - The cumulated return of each timestep is
      - $R_t = r_t + \text{gamma} * R_{t+1}$
  - Calculate the advantages of each step using the following formulas
    - Taken from the paper [High-Dimensional Continuous Control Using Generalized Advantage Estimation](#) by John Schulman et al.
    - $\delta_t^V = r_t + \gamma V(s_{t+1}) - V(s_t)$
    - $\sum_{l=0}^{\infty} (\gamma\lambda)^l \tilde{r}(s_{t+l}, a_t, s_{t+l+1}) = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V = \hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)}$
  - Shuffle the trajectories to break the time correlation and retrieve the trajectories, cumulated rewards, and advantages by batch
    - Optimize the Actor Network using this formula according to the paper [Proximal Policy Optimization Algorithms](#) by John Schulman et al.
      - $r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)}$
      - $L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$
      - Maximize L-clip
    - Optimize the Critic Network by minimizing the mean square error between
      - $R_t = r_t + \text{gamma} * R_{t+1}$
      - And $V(s_t)$

1.2 ARCHITECTURE AND HYPERPARAMETERS

Actor Network	Critic Network
Fully Connected (33, 128)	Fully Connected (33, 128)
ReLU	ReLU
Fully Connected (128, 128)	Fully Connected (128, 128)
ReLU	ReLU
Fully Connected (128, 4)	Fully Connected (128, 1)
Normal Distribution (4, std) where std is a trainable Tensor (4) and the mean is the output of the previous layer	

During the experiment, the following factor will cause the network to oscillate:

- one ReLU between the last fully connected and the normal distribution generator
- not initializing the weight of the last fully connected layer to a small number e.g. 1e-3

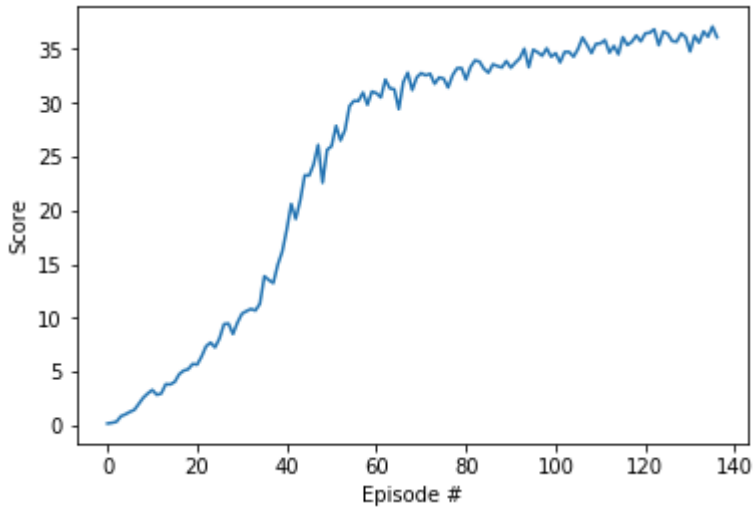
1.3 RESULTS

The environment was solved at the 137th Episodes. The following table show the progress:

Eps No	Avg Score of last <100 Eps	Last Avg Score	Best Agent Score
1	0.18	0.18	0.95
2	0.21	0.24	0.67
3	0.25	0.34	0.91
4	0.41	0.86	1.73
5	0.53	1.03	1.94
6	0.65	1.26	3.13
7	0.77	1.45	2.6
8	0.93	2.03	3.46
9	1.11	2.59	4.57
10	1.30	2.98	5.06
11	1.48	3.30	5.11
12	1.59	2.85	5.23
13	1.70	2.97	4.73
14	1.85	3.84	6.46
15	1.98	3.81	7.43
16	2.11	4.05	5.67
17	2.27	4.77	7.58
18	2.42	5.10	8.16
19	2.57	5.22	7.34
20	2.73	5.72	8.66
21	2.87	5.66	10.2
22	3.03	6.41	10.7
23	3.22	7.36	10.2
24	3.41	7.70	11.2
25	3.56	7.26	12.9
26	3.73	8.10	12.5
27	3.95	9.43	13
28	4.14	9.47	13.6
29	4.29	8.51	13.1
30	4.47	9.57	15.4
31	4.66	10.39	13.7
32	4.85	10.63	16.4
33	5.03	10.83	16.9
34	5.19	10.69	14.7
36	5.61	13.88	17.8
37	5.82	13.52	21.6
38	6.02	13.25	19.4
39	6.24	14.96	20.2
40	6.49	16.18	20.3
41	6.78	18.22	22.6
42	7.11	20.60	33.1
43	7.39	19.21	24.9
44	7.70	20.92	29.1
45	8.04	23.26	32.7
46	8.37	23.24	28.3
47	8.71	24.29	30.2
48	9.07	26.09	36.8
49	9.35	22.58	33.5
50	9.68	25.63	30.6
51	9.99	25.97	31.5
52	10.34	27.86	35.9
53	10.64	26.51	33
54	10.96	27.48	37.5
55	11.30	29.70	39.1
56	11.63	30.19	39.5
57	11.96	30.17	34.3
58	12.29	30.96	34.9
59	12.58	29.83	33.6
60	12.89	31.06	38.9
61	13.19	30.91	38.3
62	13.47	30.51	35.3
63	13.76	32.17	36.9
64	14.04	31.36	39.1
65	14.30	31.27	35.7
66	14.53	29.42	37.5
67	14.79	31.93	36
68	15.06	32.81	38.9
69	15.29	31.20	39.1
71	15.78	32.76	39.1
72	16.01	32.56	38.7
73	16.24	32.71	38.8
74	16.45	31.77	36
75	16.66	32.37	39.3
76	16.87	32.26	37.1
77	17.06	31.44	38.9
78	17.25	32.56	39.3
79	17.46	33.22	38.3
80	17.65	33.24	37.3
81	17.83	32.18	36.6
82	18.02	33.34	38
83	18.21	33.95	39.4
84	18.40	33.85	37.1
85	18.57	33.18	38.3
86	18.74	32.81	35.6
87	18.91	33.59	38.5
88	19.08	33.43	38.9
89	19.24	33.30	37.9
90	19.40	33.89	38.4
91	19.55	33.27	39.1
92	19.71	33.74	38
93	19.86	34.15	38.8
94	20.02	35.03	39.4
95	20.16	33.33	37.5
96	20.32	34.95	39.5
97	20.46	34.71	37.5
98	20.61	34.38	37.6
99	20.75	35.08	39
100	20.89	34.30	37.5
101	21.23	34.62	39.1
102	21.57	33.77	38.3
103	21.91	34.74	38.6
104	22.25	34.75	38.4
106	22.92	35.02	39.5
107	23.27	36.08	39.5
108	23.60	35.39	38.6
109	23.92	34.64	39.3
110	24.25	35.49	38.8
111	24.57	35.53	39.1
112	24.90	35.84	39.5
113	25.22	34.68	37.8
114	25.53	35.30	39.1
115	25.84	34.53	39.2
116	26.16	36.11	38.8
117	26.46	35.38	39.4
118	26.77	35.68	39.1
119	27.08	36.27	39.5
120	27.38	35.72	38.8
121	27.69	36.46	39
122	27.99	36.53	39.1
123	28.28	36.84	39.5
124	28.56	35.37	38.1
125	28.85	36.62	39.5
126	29.14	36.44	39.2
127	29.40	35.76	39.3
128	29.66	35.69	38.3
129	29.94	36.46	39.5
130	30.21	36.13	38.9
131	30.45	34.78	37.7
132	30.71	36.24	39.5
133	30.96	35.60	38.2
134	31.22	36.65	39
135	31.46	36.16	38.9
136	31.70	37.10	39.3
137	31.92	36.11	38.5

35	5.37	11.31	15.5	70	15.53	32.38	39	105	22.58	34.30	37.7

The graph is as follow:



1.4 IDEAS FOR FUTURE WORK

- The critic network being used here was optimized based on the cumulated rewards another approach could be the use of a fixed target for its optimization