

Hello to all, depict the fact that I am not even a computer science student, this project is accomplished during the first summer at McMaster University on my own initiatives and efforts. It is a great exploration in searching the values and potential power of ARS algorithm.

This project involves the application of latest and the most superior algorithm that is used to create and implement AI-ARS-the argument random search. Simply put, comparing to the existing algorithm, the ARS has following advantages:

Firstly, its exploration takes place in the Policy Space while the other AI algorithm has exploration in the Action Space.

Secondly, it embedded the method of finite differences(\*) while the gradient-descent-algorithm is usually found out in other AI algorithm.

\*what is method if finite difference? Well, basically let's say that we have a small point, a positive shift ,a negative shift or a positive delta, negative delta we have the rewards and based on that we can calculate how to adjust our weights.

Lastly, ARS-shallow learning(\*) vs Other AI-deep learning(a typical deep learning modules has many hidden layers, my other project is accomplished with deep learning modules & algorithm)

(\*) the uniqueness of art perceptron: where you have lots of inputs and lots of outputs, each input is connected to multiple outputs, yet, there is no hidden layer.

```
91
92 for step in range(hp.nb_steps):
93
94     # Initializing the perturbations
95     deltas = policy.sample_deltas()
96     positive_rewards = [0] * hp.nb_dir
97     negative_rewards = [0] * hp.nb_dir
98
99     # Getting the positive rewards in
100     for k in range(hp.nb_directions):
101         positive_rewards[k] = explore(
102
103     # Getting the negative rewards in
104     for k in range(hp.nb_directions):
105         negative_rewards[k] = explore(
106
107     # Gathering all the positive/negat
108     all_rewards = np.array(positive_re
109     sigma_r = all_rewards.std()
110
111     # Sorting the rollouts by the max(
112     scores = {k:max(r_pos, r_neg) for
113     order = sorted(scores.keys(), key
114     rollouts = [(positive_rewards[k],
115
116     # Updating our policy
117     policy.update(rollouts, sigma_r)
118
119     # Printing the final reward of the
120     reward_evaluation = explore(env, r
121     print('Step:', step, 'Reward:', re
122
123 # Running the main code
124
125 mkdir(base, name):
126 path = os.path.join(base, name)
127 if not os.path.exists(path):
128     os.makedirs(path)
129 return path
130 k_dir = mkdir('exp', 'brs')
131 itor_dir = mkdir(work_dir, 'monitor')
132
133 = Hp()
134 random.seed(hp.seed)
135 = gym.make(hp.env_name)
136 = wrappers.Monitor(env, monitor_dir, f
137 inputs = env.observation_space.shape[0]
138 outputs = env.action_space.shape[0]
139 icy = Policy(nb_inputs, nb_outputs)
140 malizer = Normalizer(nb_inputs)
141 in(env, policy, normalizer, hp)
142
```

am	Type	Size	Value
Step: 770	Reward:	916.0792432075705	
Step: 771	Reward:	927.0764313073696	
Step: 772	Reward:	927.7149275599177	
Step: 773	Reward:	923.6049740607339	
Step: 774	Reward:	935.791171768914	
Step: 775	Reward:	926.6311610697232	
Step: 776	Reward:	934.3808473417455	
Step: 777	Reward:	916.6793017697578	
Step: 778	Reward:	917.1143727151068	
Step: 779	Reward:	907.179454931309	
Step: 780	Reward:	902.5065827550952	
Step: 781	Reward:	913.6924032509667	
Step: 782	Reward:	889.2544859320824	
Step: 783	Reward:	921.623459484888	
Step: 784	Reward:	927.3397061990382	
Step: 785	Reward:	924.1048935509657	
Step: 786	Reward:	917.5864643439356	
Step: 787	Reward:	929.3301672754438	
Step: 788	Reward:	914.1227301345021	
Step: 789	Reward:	905.6977925958045	
Step: 790	Reward:	933.0644013186192	
Step: 791	Reward:	915.6791305471695	
Step: 792	Reward:	926.5743160465	
Step: 793	Reward:	927.8513616467826	
Step: 794	Reward:	910.9370900973374	
Step: 795	Reward:	915.7750302571591	
Step: 796	Reward:	924.9638868025368	
Step: 797	Reward:	873.7701286315888	

As image shows, it took about 35 min to reach the maximum reward which was 927.85...., up to this stage, the dog is fully capable to run straightly without causing failure or falling off like it did in early stages(before reaching reward of 300, the video that records the movement of dog is also available, please check out the .mp4 file if you are interested.