

0.1 Page 1

One of the simplest example of the method in this family is random search, when you randomly sample the thing youre looking for (in case of RL its the policy $\pi(a|s)$), then you check

0.2 Page 2

More formally, the method above could be expressed as this sequence of steps.

1. Initialize learning rate α , noise standard deviation σ , initial policy parameters θ_0
2. For $t = 0, 1, 2, \dots$ do
 - (a) Sample batch of noise with a shape of the weights $\epsilon_1, \dots, \epsilon_n \sim \mathcal{N}(0, I)$
 - (b) Compute returns $F_i = F(\theta_t + \sigma \epsilon_i)$ for $i = 1, \dots, n$
 - (c) Update weights $\theta_{t+1} \leftarrow \theta_t + \alpha \frac{1}{n\sigma} \sum_{i=1}^n F_i \epsilon_i$

0.3 Page 4

The last and the central function of the method is `train_step` which takes the batch with noise and their respective rewards and calculates the update to the network parameters by applying the formula $\theta_{t+1} \leftarrow \theta_t + \alpha \frac{1}{n\sigma} \sum_{i=1}^n F_i \epsilon_i$

0.4 Page 12

1. Initialize mutation power σ , population size N , number of the selected individuals T , initial population P^0 with N randomly-initialized policies, their fitness $F^0 = \{F(P_i^0) | i = 1 \dots N\}$
2. For generation $g = 1 \dots G$
 - (a) Sort generation P^{g-1} by descending of fitness F^{g-1}
 - (b) Copy elite $P_1^g = P_1^{g-1}, F_1^g = F_1^{g-1}$
 - (c) For individual $i = 2 \dots N$
 - i. k = randomly select parent from $1 \dots T$
 - ii. Sample $\epsilon \sim \mathcal{N}(0, I)$
 - iii. Mutate parent $P_i^g = P_i^{g-1} + \sigma \epsilon$
 - iv. Get its fitness $F_i^g = F(P_i^g)$

0.5 Page 16

To implement the Novelty Search, we define so called Behaviour Characteristic $BC(\pi)$, which describes the behaviour of the policy and a distance between two BCs.