

DL project

samuelelippolis

July 2024

1 Explanation SL EA Tuning

1.1 SLEA Structure Overview

The general structure of the Samuele Lippolis Evolutionary Algorithm (SLEA) tuning is given by the following schema:

- Initialization (generate μ individuals)
- FA (how?)
- Repeat
 - Generate offsprings (build λ offsprings)
 - FA
 - Offsprings selection (select μ individuals)
 - If $FA \geq n - \text{fa-threshold}$, then return all individuals and end

1.2 Solution Encoding

An individual i belongs to $i \in G \times S \times Q$, where G is the space of the genotypes, S the space of the solutions, and Q the space of the quality (fitness) of the solution.

The spaces are defined as follows:

$$G = \mathbb{R} \times \mathbb{R}^k \times \mathbb{N} \times \mathbb{R} \times \mathbb{R}$$

$$S = \mathbb{N} \times \mathbb{N}^k \times \mathbb{N} \times \mathbb{R} \times \mathbb{R}$$

- The first component: the number of layers of the Neural Network (NN).
- The second component: the number of units in each layer.
- The third component: encoding of the activation function.
- The fourth component: learning rate.
- The fifth component: dropout rate.

1.3 Fitness Assessment

The fitness assessment of the NN works as follows:

- Train
- Test
- Measure the test loss of each epoch
- Take the minimum of the test loss of the last k epochs or the average of the test loss of the last k epochs

1.4 Generate Offsprings

- Repeat $\lambda/2$ times
 - Tournament selection (select 2 from the population)
 - Choose randomly between CX (which one?) or Mutation (which one?)
 - Perform CX or mutation

1.5 Offspring Selection

The offspring selection is a truncated selection on the offsprings (λ individuals) to generate the new population (μ individuals).

1.6 Evolutionary Operators

1.6.1 Crossover

Crossover (CX) is applied only to genotypes that are real numbers. In this case, the CX operator is a k -linear recombination, meaning that a random number r is uniformly sampled from the interval $(-k, 1 + k)$. Then, the gene of the offspring is calculated as follows:

$$\text{gene_offspring} = r \cdot \text{gene_1} + (1 - r) \cdot \text{gene_2}$$

1.6.2 Mutation Operators on Different Genotype Types

There are different mutation operators depending on the genotype type.

- If the genotype is an integer, the mutation is:
 - With probability p
 - * `af.new = random-choice(all the af)`
- If the genotype is composed of real numbers, there are two functions: `preprocess` and `gaussian-mutation`.

- The **preprocess** function works as follows:
 - * Find the maximum length across all genotypes.
 - * For each gene position in a genotype:
 - Compute the mean of all genes in that position across all genotypes: $g_mean^{(1)}, g_mean^{(2)}, \dots$
 - Compute the maximum of all genes in that position across all genotypes: $g_max^{(1)}, g_max^{(2)}, \dots$
- The **gaussian-mutation** works as follows:
 - * For each gene j
 - $gene = gene + \mathcal{N}(0, g_mean^{(j)}/10)$
 - $gene = \text{bound-constraint}(gene)$