Weighted Random Search for Hyperparameter Optimization

Galina Boeva

MIPT, 2023

November 27, 2023

Motivation

Main idea

The motivation is to modernize the generally accepted approaches with an improvement in the quality of work. A weighted search method is proposed, which suggests that a value that has already led to a good result is a good candidate for a new test and should be tested in new combinations of hyperparameter values.

The WRS Method

Algorithm 1 A WRS Step - Objective Function Maximization

```
Input: F; (X^k, F(X^k)); p_i, k_i, P_i(x), i = 1, ..., d
Output: (X^{k+1}, F(X^{k+1}))
Randomly generate p, uniform in (0,1)
for i = 1 to d do
    if (p_i ≥ p or k ≤ k_i) then
        // either the probability condition is met or more samples are needed
        Generate x_i^{k+1} according to P_i(x)
    else
        x_i^{k+1} = x_i^k
    end if
end for
// usually this is the most time consuming step
Compute F(X^{k+1})
if F(X^{k+1}) ≥ F(X^k) then
    return (X^{k+1}, F(X^{k+1}))
else
    return (X^k, F(X^k))
end if
```

The WRS Method

Algorithm 2 WRS - Objective Function Maximization

Input: F; N; $P_i(x), i = 1, \dots, d$
Output: $(X^N, F(X^N))$

// Phase 1 - Run RS
for $k = 1$ to $N_0 < N$ **do**
    Perform RS step, compute $(X^k, F(X^k))$
end for
// Intermediate phase, determine input for WRS
Determine the probability of change $p_i, i = 1, \dots, d$
Determine the minimum number of required values $k_i, i = 1, \dots, d$
// Phase 2 - Run WRS
for $k = N_0 + 1$ to N **do**
    Perform WRS Step described in Algorithm 1, compute $(X^k, F(X^k))$
end for
return $(X^N, F(X^N))$

Theoretical Aspects and Convergence

Multi-dimensional case

For the general case of optimizing a function $F : S_1 \times S_2 \cdots \times S_d \to R$, with $Si, i = 1, \ldots, d$ countable sets and under the same assumption that the variables are not statistically correlated, P_{RS} and P_{WRS} are defined as:

$$p_{RS} = \prod_{i=1}^{d} \frac{1}{|S_i|}, p_{WRS} = \frac{1}{|S_1|} \prod_{i=2}^{d} \left(p_i \frac{1}{|S_i|} + (1 p_i) \frac{1}{|S_i| - m_i + 1} \right)$$

where m_i is the number of distinct values already generated for x_i.

Theorem

For any function $F : S_1 \times S_2 \cdots \times S_d \to R$ there exist $k_i, i = 1, \ldots, d$, so that $p_{WRS:n} \geq p_{RS:n}$.

An Example: Griewank Function Optimization

Grievank function

$$G_d = 1 + \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos\frac{x_i}{\sqrt{i}}$$

We use a slightly modified version of G_6, given by:

$$G_6^* = 1 + \frac{i-1}{4000}\sum_{i=1}^{6} x_i^2 - \prod_{i=1}^{6} \cos\frac{x_i}{\sqrt{i}}$$

An Example: Griewank Function Optimization

Parameter	x_1	x_2	x_3	x_4	x_5	x_6
Weight	0.07	0.18	1.24	7.77	23.52	43.96
Probability	0.002	0.004	0.028	0.177	0.535	1.00

Figure: Parameter weights and probabilities for G_6^*.

Optimizer	Best Found Value	Average Value	SD
RS	-1.50	-33.10	14.06
WRS	**-1.28**	**-14.58**	**10.63**

Figure: WRS vs. RS results for G_6^* - values for 1000 runs.

An Example: Griewank Function Optimization

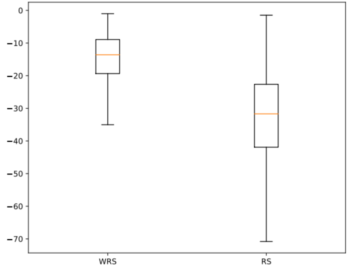


Figure 1: Performance of WRS vs. RS for the G_6^* optimization

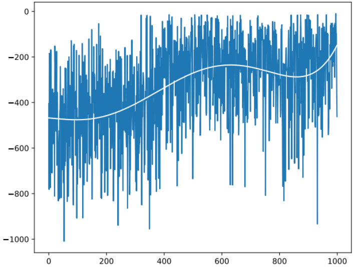


Figure 2: Convergence of WRS for the G_6^* function

CNN Hyperparameter Optimization

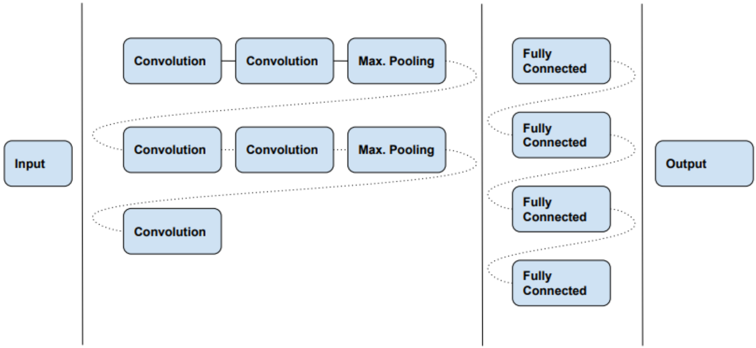
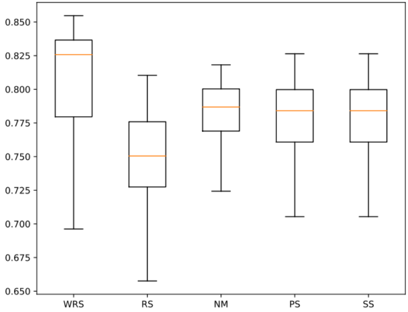


Figure: CNN architecture.

CNN Hyperparameter Optimization



Optimizer	Best Result	Average	SD
WRS	**0.85**	**0.79**	0.09
RS	0.81	0.75	0.04
NM	0.81	0.77	0.03
PS	0.83	0.78	0.03
SS	0.82	0.75	0.05

Figure: 1. Performance of WRS, RS, NM, PS and SS for CNN optimization. 2. Algorithms' results for CNN accuracy on CIFAR-10.

Literature

1. **Main article** Weighted Random Search for Hyperparameter Optimization.