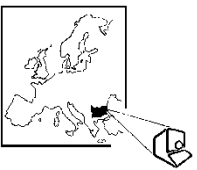
2022 International Conference on Information Technologies (InfoTech-2022 – 36th issue),
IEEE Conference Rec.# 55606
15-16 September 2022

Optimization of hyperparameters with constraints on time and memory for the classification model of the hard drives states



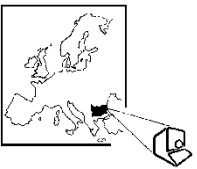
Authors: Liliya A. Demidova
Anton V. Filatov
Russian State University - MIREA

Hyperparameters optimization

In machine learning, a hyperparameter is a parameter whose value is used to control the learning process.

One of the important tasks when training a model is to adjust the hyperparameters of the model to obtain the most accurate result in the shortest possible time.

In real work, developers often do not have an excessive amount of power and time resources to check the maximum efficiency of the developed product.



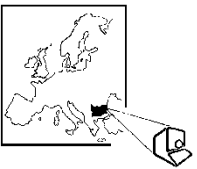
APPROACHES AND TECHNOLOGIES

In this study, an approach to optimizing hyperparameters of machine learning models is considered using the example of the problem of classifying the states of hard drives.

Classification models are trained based on a publicly available dataset from BackBlaze.

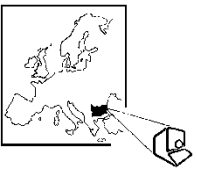
The models used in the work were based on:

- Recurrent neural network of the Long Short-Term Memory (LSTM) type;
- Random Forest (RF) algorithm.



Hyperparameter optimization methods

- RandomSearch, which implements hyperparameter tuning using a random search scheme;
- Grid Search, which implements a complete search over a manually set subset of hyperparameter values;
- TreeStructured Parzen Estimators (TPE), which implements the application of the Parsen Window method and the Baisen optimization algorithm;
- Covariance Matrix Adaptation which is the evolutionary algorithm for complex non-linear non-convex optimization problems.



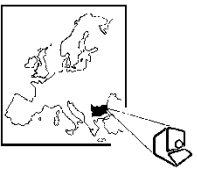
Hyperparameters of models

RF model hyperparameters:

- n_estimators – the number of trees in the forest;
- max_depth – the maximum depth of the tree;
- min_samples_split – the minimum number of samples required to split an internal node in the tree;
- min_samples_leaf – the minimum number of samples that should be located in the final node;
- max_features – the number of functions to consider when searching for the best separation;
- bootstrap parameter.

LSTM model hyperparameters:

- batch size;
- percentage of disabled neurons in layers named as the set_dropout;
- number of neurons in the first LSTM layer as the lstm_units;
- number of neurons in the second LSTM layer as the lstm_units2;



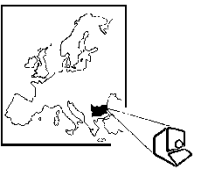
Assessment of classification quality

- Precision - the proportion of objects called positive by the classifier and actually being positive (1);
- Recall - proportion of objects of a positive class out of all objects of a positive class the algorithm found (2);
- F-score - proportion of objects of a positive class out of all objects of a positive class the algorithm found (3).

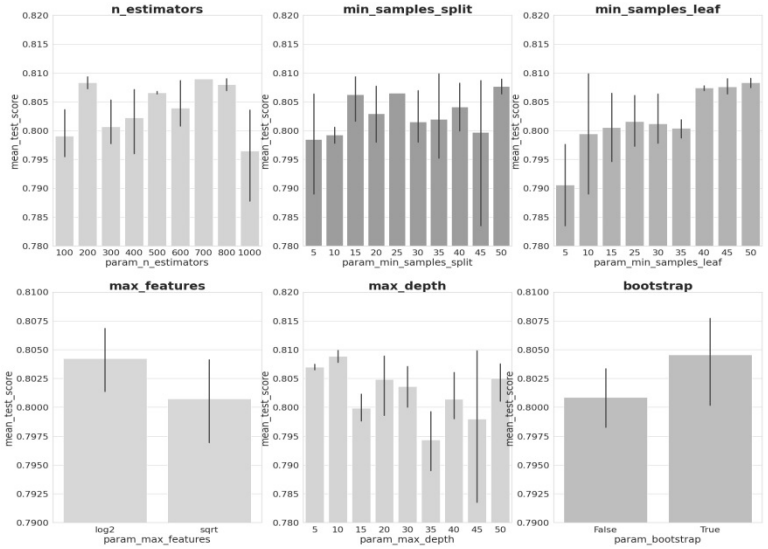
$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

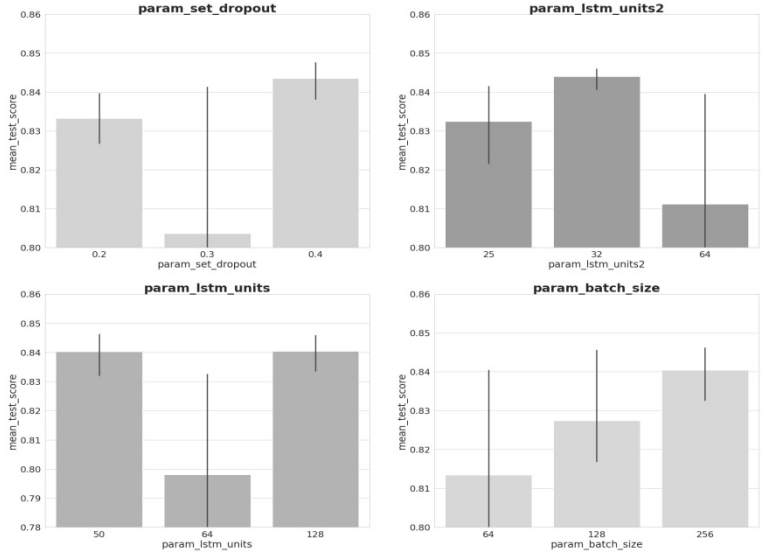
$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$



RandomSearch and GridSearch



Mean score of RF hyperparameters*



Mean score of LSTM hyperparameters*

```
n_estimators = [200, 700, 800]
max_features = ['log2']
max_depth = [10, 20, 45]
min_samples_split = [40, 50]
min_samples_leaf = [40, 45, 50]
bootstrap = [True]
```

GridSearch RF hyperparameters

```
'set_dropout': [0.4],
'lstm_units': [50, 128],
'lstm_units2': [25, 32],
'batch_size':[128, 256],
```

GridSearch LSTM hyperparameters

"*" The columns in the histograms are based on the average results of the models in which the hyperparameter value indicated on the axis was used. The black lines indicate the results of the minimum and maximum effective models in which this hyperparameter value participated.

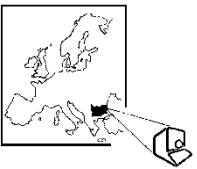
TPE and CMA

The TPE method takes a hierarchical search space with a priori probabilities as input, and at each step, using the Parsen Window method, it clarifies the distributions of "good" and "bad" points based on the values of the target metric.

Existing implementations of the TPE method included in the following libraries were used:

- HyperOpt;
- Optuna.

The CMA-ES method implements an evolutionary strategy with adaptation of the covariance matrix. It uses a covariance matrix to track dependencies between decision variables. This method has a high-quality implementation in the Optuna library.



Comparison of results

RandomForest models scores

Model type	Precision	Recall	F-score	Time, s
Default	0.6746	0.9311	0.7823	104*
RS	0.6168	0.9362	0.7437	8431
GS	0.5680	0.9323	0.7059	9662
TPE Hyperopt	0.6695	0.9503	0.7855	8868
TPE Optuna	0.6748	0.9747	0.7974	7241
CMA-ES	0.6592	0.9672	0.7840	9724

LSTM models scores

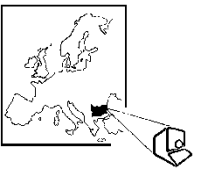
Model type	Precision	Recall	F-score	Time, s
Default	0.8292	0.9411	0.8817	176*
RS	0.8236	0.9524	0.8833	7382
GS	0.8197	0.9629	0.8908	4242
TPE Hyperopt	0.8617	0.9611	0.9087	6964
TPE Optuna	0.8457	0.9643	0.9011	5942
CMA-ES	0.8774	0.9178	0.8971	6281

"*" indicates that the results were obtained by running the learning algorithm once with the default hyperparameters values.

Red - the optimization method did not improve the result of the model.

Yellow - the optimization method improved the results of the model, but performed worse in comparison with other methods.

Green - the optimization method showed the highest improvement in results.



Conclusion

The conducted research confirms the effectiveness of using hyperparameters optimization methods under time and memory constraints.

A comparative analysis of the results of the optimization methods revealed a clear advantage of using the TPE method to solve the problem of classifying the states of hard drives.