

Title: Optuna

URL: <https://en.wikipedia.org/wiki/Optuna>

PageID: 80384604

Categories: Category:Deep learning, Category:Free and open-source software, Category:Machine learning

Source: Wikipedia (CC BY-SA 4.0).

Optuna is an open-source Python library for automatic hyperparameter tuning of machine learning models. [1] It was first introduced in 2018 by Preferred Networks , a Japanese startup that works on practical applications of deep learning in various fields. [2] [3] [4] [5] The beta version of Optuna was released at the end of the year, with the subsequent first major stable release announced in January 2020. [1]

Hyperparameter optimization

Hyperparameter optimization involves finding the optimal value of non-trainable parameters, defined by the user. Examples of hyperparameters are learning rate , number of layers or neurons, regularization strength and tree depth. However, they strongly depend on the specific algorithm (e.g., classification , regression , clustering , etc.). [6]

Hyperparameter optimization can be especially relevant when dealing with large-scale problems or limited resources, as it improves model accuracy, reduces overfitting, and decreases training time. [1] However, when the hyperparameter space increases, it may become computationally expensive. Hence, there are methods (e.g., grid search , random search , or bayesian optimization) that considerably simplify this process. [1]

Features

Optuna is designed to optimize the model hyperparameters, by searching large spaces and discarding combinations that show no significant improvements in the model. Moreover, it can parallelize the hyperparameter search over multiple threads or processes. Optuna works with a high degree of modularity, allowing the definition of different configurations for searching hyperparameters. It is possible to choose which hyperparameters to optimize and how to optimize them. Additionally, it permits parameter customization at runtime, meaning that the values to explore for the hyperparameters (i.e., the search space) can be defined while writing the code, rather than being defined in advance. [1]

Sampling

Optuna exploits well-established algorithms to perform hyperparameter optimization, progressively reducing the search space, in light of objective values. Examples are gaussian-process-based algorithms (i.e., a gaussian process to model the objective function [7]), tree-structured parzen estimator (TPE) (i.e., a model-based optimization method that estimates the objective function and selects the best hyperparameters [8]), and random search (i.e., a basic optimization approach used for benchmarking [9]).

Early stopping

Optuna includes a pruning feature to stop trials early, when the results show no significant improvement in the model. This allows for the prevention of unnecessary computation, and it is used for models with long training times, in order to save time and computational resources. Specifically, Optuna exploits techniques such as median and threshold pruning . [10] [11]

Scalability

Optuna is designed to scale with distributed computing resources, supporting parallel execution . This feature allows users to run optimization trials across multiple processors or machines. [12]

Integration with third-part libraries

Optuna integrates with various ML libraries and frameworks: [1]

Catboost

Dask

Fast.ai

Keras

LightGBM

MLflow

PyTorch

PyTorch Ignite

PyTorch Lightning

TensorBoard

TensorFlow

tf.keras

Weights & Biases

XGBoost

Moreover, Optuna offers a real-time dashboard that allows to monitor, through graphs and tables, the optimization history and the hyperparameter importance. [13]

Applications

Optuna was designed to be framework agnostic, so that it can be used with any machine-learning (ML) or deep-learning (DL) framework. [1]

Machine learning

Optuna can be used to optimize hyperparameters of ML models. Examples are:

Random forest : number of trees, maximum depth, and minimum samples per leaf. [14]

Gradient boosting machines (GBM): learning rate, number of estimators, and maximum depth. [15]

Support vector machines (SVM): regularization parameter (C), kernel type (e.g., linear, radial basis function), and gamma (gamma). [16]

K-nearest neighbors (KNN): number of neighbors (k), distance metrics (e.g., Euclidean or Manhattan), and weight function. [17]

Linear and logistic regression: alpha in Ridge Regression or C in Logistic Regression. [18]

Naive Bayes : smoothing coefficients. [19]

Deep learning

In the context of deep learning, Optuna can be exploited during the training of neural networks (NN), to optimize learning rate, batch size, and the number of hidden layers. For example, it can be used for:

Convolutional neural networks (CNNs), for image classification, object detection, and semantic-segmentation tasks. [20]

Recurrent neural networks (RNNs), for sequence-based tasks such as time-series forecasting and natural language processing. [21]

Transformers , for NLP tasks such as text classification, sentiment analysis, and machine translation. [22]

Domains

Optuna has found applications in various research studies and industry implementations across different applicative domains. [2] [3] [4] [5]

Healthcare

In healthcare , Optuna is currently exploited for medical image analysis to optimize DL models for tumor detection, [23] [24] disease identification, [25] multi-organ semantic segmentation, [26] and radiological image analysis.

It can be also exploited for disease prediction , in particular for the improvement in the accuracy of predictive models for disease diagnosis and treatment outcomes. [27] Its application can be also found in genomics , to predict genetic diseases and to identify genetic variations. [28]

Finance

In finance, Optuna is used to optimize models for algorithmic trading . It allows for predicting market movements, given the ability in handling wide parameter ranges and complex model structures. [29] [30] In particular, it is exploited for financial risk analysis and forecasting. Topics addressed are credit, market, and operational risks. [30]

Autonomous systems

Optuna is used for real-time applications in autonomous systems for robotics , [5] supporting decision making in dynamic environments.

It is also exploited in the context of self-driving cars , to optimize the model to navigate safely in complex environments. For example, Optuna can be used in scenarios where there is the need to evaluate the rate and the severity of accidents [31] or to address the issue of network intrusion attacks due to possible vulnerabilities that might occur. [32] [33]

Natural language processing (NLP)

Optuna has also been applied in NLP, for example, for text classification to classify a piece of written text into categories (e.g., spam vs. not spam, topic modeling). [34] Another task is sentiment analysis , namely the detection of feelings expressed in text, particularly exploited for the analysis of content from social media and customer reviews. [35]

Reinforcement learning (RL)

For what concerns RL, Optuna is exploited in the gaming field, to improve the model performance in games and virtual environments, in robotics to optimize decision-making processes in robotic systems for tasks like manipulation and navigation, and in the field of autonomous vehicles , to optimize RL models to obtain enhanced safety, and a more efficient navigation strategy. [36]

See also

Hyperparameter tuning

Bayesian optimization

Machine learning

Deep learning

References

Further reading

Wang, Xilu; Jin, Yaochu; Schmitt, Sebastian; Olhofer, Markus (2023-07-13). "Recent Advances in Bayesian Optimization" . ACM Comput. Surv . 55 (13s): 287:1–287:36. doi : 10.1145/3582078 . ISSN 0360-0300 .

Shinde, Pramila P.; Shah, Seema (2018-08-16). "A Review of Machine Learning and Deep Learning Applications". 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) . pp. 1– 6. doi : 10.1109/ICCUBEA.2018.8697857 . ISBN 978-1-5386-5257-2 .

External links

[Official website](#)

[Optuna on GitHub](#)

[Optuna Documentation](#)

[Tutorial](#)

[Optuna Dashboard](#)

[Optuna Dashboard Documentation](#)