Scholar Publishing Group
Machine Learning Theory and Practice
https://doi.org/10.38007/ML.2020.010205
ISSN 2790-0983 Vol. 1, Issue 2: 37-44

Theory and Practice of Super Parameter Optimization for Machine Learning Algorithm

Lijuan Shan*
Philippine Christian University, Philippine
shanlijuan2014@163.com
**corresponding author*

Keywords: Machine Learning, Super Parameter Optimization, Automl Structure, Mars Algorithm

Abstract: Before the rise of large machine learning algorithms, most people manually adjusted the super parameters of the model by relying on experience. However, with the increasing complexity of the model, this method obviously cannot meet the needs. This paper mainly studies the theory and practice of super parameter optimization of machine learning algorithm. This thesis proposes a regression-based hyperparameter optimization algorithm that has the same data-based optimization algorithm as the optimization algorithm Bayesian. The optimization algorithm is based on the Gaussian regression process. In addition to being affected by the super parameters of the kernel function in the process of GP regression fitting, the calculation amount of the algorithm will also increase significantly. The experimental results show that, compared to the optimization algorithm, the parameter optimization results of this algorithm are similar to those of the optimization algorithm.

1. Introduction

Machine learning models often need a lot of manual adjustment before use, in order to better apply to practical problems. However, manual intervention is inefficient, and not all scenarios are guided by experienced human experts. Automated Machine Learning (AutoML for short) came into being [1-2]. AutoML is the combination of automation technology and machine learning. By designing a series of advanced systems that are easy to use, configuring and adjusting machine learning models, it enables automatic learning without manual or less manual intervention. The main problems of AutoML include algorithm selection, super parameter optimization and model selection. The key technologies are divided into optimization technology and evaluation strategy. Hyper parameters Optimization (HPO) is one of the key links of AutoML [3]. In engineering, the optimization of neural network parameters is of great significance, but also a huge challenge. There

are generally two types of parameters in machine learning, one of which is the parameter represented by the weight in neural networks (NN), which can be obtained during training; The other is empirical value, which is obtained by empirical estimation, namely, super parameter. The super parameters need to be configured before training, and generally will not change during the whole training process [4-5]. The ultimate goal of super parameter optimization is to configure a set of the most appropriate parameters for the model, so that the model can obtain better calculation results or have faster convergence capability. However, because it is difficult to obtain the gradient of super parameters in machine learning, the traditional gradient descent method and Newton method are difficult to be applied to HPO problems [6]. Therefore, some researchers proposed to apply black box optimization methods such as random search method and evolutionary algorithm without gradient information to HPO problem, avoiding the difficulty of calculating super parameter gradient [7].

Hyper parameter optimization, also known as Hyper parameter tuning, is essentially 'optimized optimization'. Hyperparametric optimization is also a step in AutoML [8]. The ability and efficiency of neural networks are largely determined by the configuration of super parameters. Nowadays, researchers generally believe that the adjusted super parameters are better than the default parameter settings provided in ordinary machine learning libraries [9]. A good set of super parameters can train a more efficient machine learning model, so super parameter tuning is very important in machine learning, and it is also one of the hotspots of scholars' research in recent years [10]. Before the advent of automatic machine learning, the main method of super parameter tuning was manual tuning, which was adjusted according to the user's experience. This adjustment method has low efficiency and high trial and error rate, which affects the training efficiency and results of machine learning [11]. Hyperparametric optimization methods can be divided into two categories, one is black box optimization methods, in which grid method, random search method, Bayesian optimization method and evolutionary algorithm are all black box optimization methods, and the other is multi fidelity optimization. With the increase of data sets, the machine learning model will become more complex, thus increasing the computational complexity, making the evaluation of black box optimization very expensive [12]. Therefore, some scholars in this field also apply the multi fidelity optimization idea to the super parameter optimization, and use the low fidelity algorithm as the approximate evaluation of the true value.

The traditional manual adjustment of parameters is not enough to meet the needs of the model, especially the needs of the deep learning model. The selection of super parameters is facing difficulties. For a given model, we often do not know the internal performance of the model (such as gradient information). We only know the input information when making super parameter selection, and there is no way to establish an objective function for the super parameters of the model. Therefore, the super parameter selection problem is a "black box problem" that requires expert experience. It is crucial to develop an efficient automatic search method for super parameters.

2. Machine Learning Hyperparameter Optimization Based on MARS

2.1. AutoML Structure and Hyperparameter Strategy

(1) AutoML Structure

Automatic machine learning (AutoML) refers to the automatic configuration of the links related to important steps in the machine learning process, such as feature selection, model selection, super parameter optimization and model evaluation, that is, it is the automation of the whole process. But in fact, the automatic configuration and optimization of one or more links in machine learning alone also belongs to AutoML, so the structure and technology discussed in this section belong to

AutoML in a broad sense [13-14].

From the perspective of the optimization scope of the machine learning process in AutoML, it can be divided into five categories: feature selection oriented, model selection oriented, super parameter optimization oriented, and partial or full scope oriented. In addition, according to the different tasks of machine learning applications, from the perspective of whether the data contains labels, AutoML can be divided into three types: supervised, semi supervised and unsupervised. The current research work is almost for supervised, and a small amount is for semi supervised. Because unsupervised machine learning is difficult, there is no AutoML work for unsupervised machine learning [15].

The general structure of AutoML is shown in Figure 1. Regardless of the type of AutoML, the optimizer and evaluator are its core components, which constitute an iterative search process including configuration generation and evaluation [16]. Among them, the optimizer is responsible for generating potential candidate configurations for the evaluator, and its search space is determined by the optimization scope; the evaluator is responsible for building an algorithm model on the training set using the configuration provided by the optimizer, and evaluating the performance of the algorithm model. Whether the evaluator feeds back the results to the optimizer depends on the type of optimizer [17]. It can be seen from the structure of AutoML that the performance of AutoML tools depends on the search strategy of the optimizer and the evaluation strategy of the evaluator.

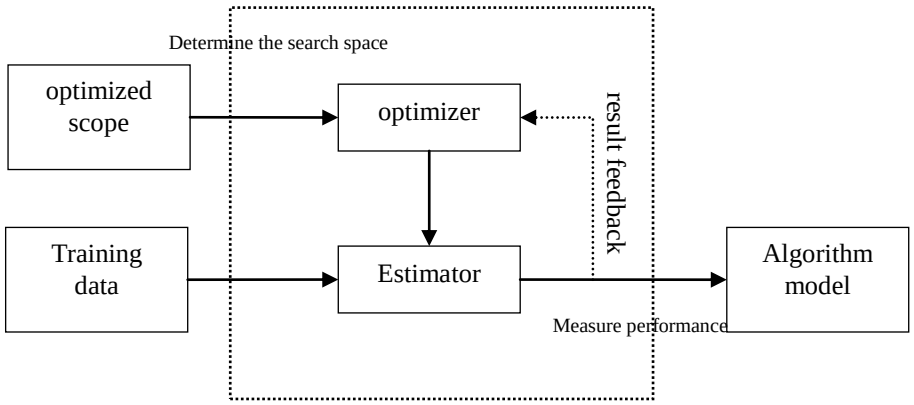

Figure 1. Generic structure of AutoML

(2) Select Policy

The selection of key super parameters is similar to feature selection, and its purpose is to find the optimal subset by adding or removing elements (features or super parameters) [18]. The existing work uses the AutoML tool to collect a large number of performance data sets offline, build an empirical performance model and analyze the contribution of the super parameters to the model performance to identify the key super parameters, so as to obtain insights on how the super parameters of different algorithms affect the model performance.

Positive selection is a common method of selecting key variables in model construction. For the performance data set of a particular algorithm, the method first divides the data set into a training set and a verification set, and then repeatedly adds the superparameters to the subset from the empty superparameter subset, so that the random forest regression model determined in the training set by the hyperparameter subset produces the minimum average square root error (root error) in the verification set.

Variation function analysis is a method that uses a random forest prediction model and variance function analysis to analyze the importance of hyperparameters or subsets of hyperparameters. fANOVA first constructs a prediction model based on random forests to predict the average performance of each configuration over the entire problem area. Then, the performance variance of the entire configuration is broken down into additional components through functional variance analysis, and each additional element corresponds to a subset of superparameters.

The subtraction analysis determines the important parameters on the path by controlling the performance change of the default configuration along the subtraction path, takes the optimal configuration and calculates the contribution rate of each parameter to the improvement of performance and the total yield achieved. This process can also be reversed to select the super parameter with the least performance loss. This method must run an algorithm each time superparameters are modified during the search process, so it takes a long time.

2.2. Hyperparameter Tuning Based on MARS Regression

MARS is an efficient regression method for processing large scale data. Multivariate adaptive regression spline has the advantages of processing large scale data and fast, efficient and accurate modeling. At present, multiple adaptive regression spline algorithm is more accurate than other methods in business management, geological exploration, chemical analysis, ecotourism, industrial design and other fields.

First of all, it is necessary to know that generating prediction models through training sets is the target problem to be solved by regression problem. Each section of region can be expressed as the following formula (1) :

$$y = a_0 + \sum_{m=1}^{M} a_m S_m(x) = a_0 + \sum_{m=1}^{M} a_m \prod_{k=1}^{k_m} [S_{km}(x_{v(k,m)} - t_{km})] \tag{1}$$

In the above formula, is the prediction result of the model, a0,a1,a2... ,am are coefficients, Sm(x),M are basis functions and the number of basis functions respectively.

Finally, the model fitting result is a linear combination of the following formula:

$$f(x) = a_0 + \sum_{m=1}^{M} a_m B_m(x) \tag{2}$$

In Equation (2) above, am is the coefficient am estimated by the sum of squares of minimum residuals, which can also be understood as Bm(x) formed by multiplication of multiple spline functions, which is determined by standard linear regression.

The front process is to divide the sample space. However, after performing this procedure, many splines will be generated, which may lead to the risk of overfitting.

After generating many splines in the previous step, a backward process is required, in which pruning operations are performed at each step, in which the terms deleted minimize the increase in the sum of squared residuals to obtain the estimate fλ(x) of the optimal model for each λ. Cross validation can be used to estimate the optimal λ, but in order to save calculation, the results of the generalized cross validation of MARS process are used as the benchmark, and the generalized cross validation criterion is defined as:

$$GCV(\lambda) = \frac{\sum_{i=1}^{N} (y_i - f_\lambda(x_i))^2}{(1 - \frac{M(\lambda)}{N})^2} \tag{3}$$

In equation (3) above, M(λ) is the number of terms in the final model.

The basic framework of hyperparameter optimization algorithm based on MARS is shown in the following algorithm.

Input: Functions that need to be optimized

1. Generate n initial points and evaluate them, and build the database D={xiyi, i=1,2,... n}.
2. while the termination condition is not reached
{
3. Multiple adaptive regression spline model MARS is constructed using the data in D.
4. A large number of random candidate solutions are generated by adaptive sampling method.
5. Generate the next iteration point xn+1 from these candidate solutions based on MARS model.
6. Evaluate xn+1, yn+1=f(xn+1)
7. Add data {xn+1,yn+1} to D and update the MARS regression model.
}

Output: D optimal solution.

3. Hyperparameter Optimization Experiment

3.1. Experimental Environment

In this paper, the hyperparameter optimization method based on MARS regression spline is compared with the classical Bayesian optimization algorithm, and the experimental results show that the hyperparameter optimization method based on MARS greatly improves the time efficiency when the quality of the solution is not lower than that of the Bayesian optimization. This experiment is completed in the hardware environment of Intel Core (TM) i5-6500 with 3.2GHZ main frequency, 16GB memory and Linux16.04 python 2.7x software environment.

3.2. Experimental Description

The comparison algorithm of the algorithm proposed in this paper is a series of Bayesian optimization methods, such as Bayesian optimization based on PI acquisition function (GP-PI) and Bayesian optimization based on EI acquisition function (GP-EI).

The hyperparameter optimization algorithm based on MARS is implemented using the open source tool pySOT, and the Bayesian optimization algorithm is implemented based on the open source framework GPyOpt. Machine learning models, such as support vector machines, random forests, and neural networks, are implemented based on the sklearn open source library.

The first problem in this paper is to adjust the hyperparameters of support vector machine. RBF kernel is very suitable for recognizing handwritten digits. Therefore, in this problem, only two hyperparameters are optimized in this paper, whose dB is the nuclear bandwidth δ coefficient C. These two hyperparameters have a great impact on the generalization performance of SVM, and the detailed information is given in Table 1.

Table 1. Details of the SVM to be optimized

Super parameter	Type	Select interval
δ	Continuous	[1.0,1000.0]
C	Continuous	[0.0001,1.0]

The second problem is to adjust the hyperparameters of the neural network on MNIST script. For this purpose, we use a 3-layer neural network to do the experiment, and its learning style is adaptive. In addition, the neural network uses the "earlystopping" policy. Table 2 shows the details.

Table 2. Details of the neural network to be optimized

Super parameter	Type	Select interval
Hidden layer neuron	Discrete	Integer range[10 630]
Solver	Discrete	SGD,Adam
Activation function	Discrete	tanh,sigmoid,relu

4. Analysis of Experimental Results

Figures 2 and 3 summarize the statistical results of the hyperparameter optimization algorithm based on MARS and the Bayesian optimization algorithm on each experimental object. Min, Max, Mean and Std respectively represent the best result, the worst result, the average result and the variance obtained by various data-driven optimization algorithms when they perform multiple minimization on the same optimization problem.

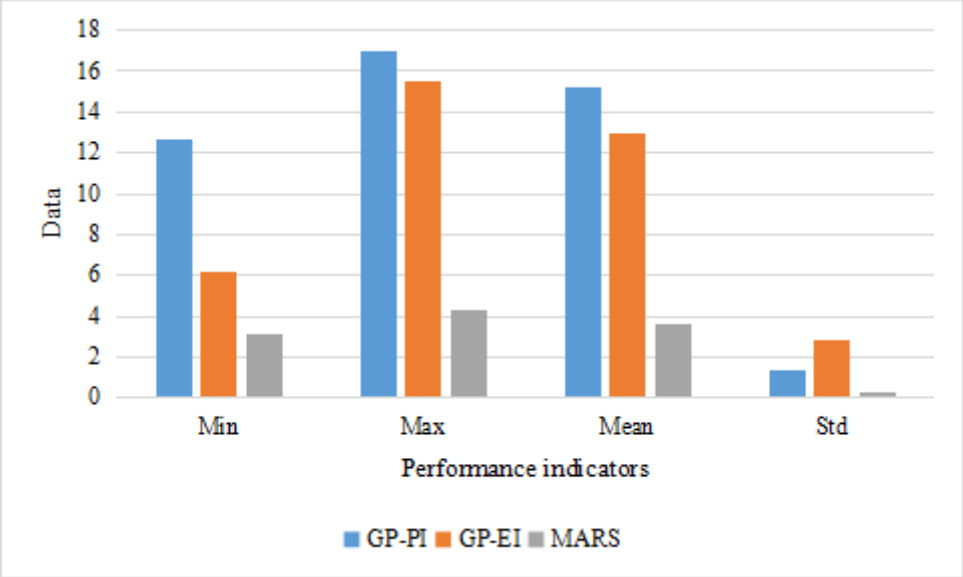

Figure 2. Comparison of optimized SVM data

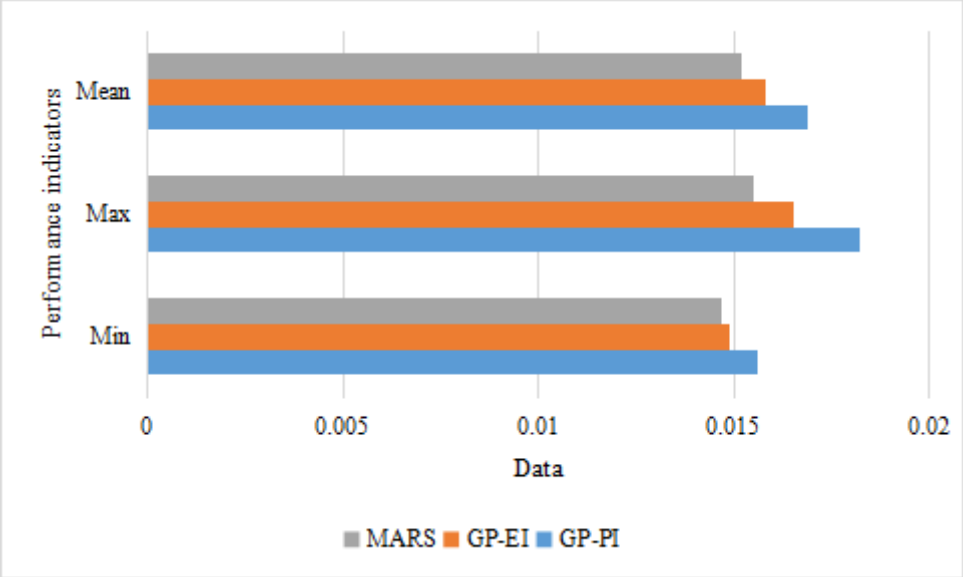

Figure 3. Optimized neural network data comparison

As shown in Figure 2 and Figure 3, it can be seen that the hyperparameter optimization based on MARS is much more efficient than the Bayesian optimization algorithm, and the final result is no less than the result of Bayesian optimization.

5. Conclusion

This paper first shows that the parameter tuning problem of machine learning model is a black-box expensive function optimization problem. Then we introduce a very classic data-driven optimization algorithm, Bayesian optimization, and one of the innovative points of this paper. Finally, the effectiveness of multivariate adaptive regression spline is explained and demonstrated through experiments and theoretical analysis. Due to the large size of machine learning models, it usually takes a long time to train and evaluate them once on the CPU. The realization of machine learning model parallel computing, distributed computing, is the focus of our future work. Only when this problem is solved can the data-driven optimization framework be applied to the hyperparameter optimization of various deep learning models in this paper.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Yousefi A , Pishvaee M S . A hybrid machine learning-optimization approach to pricing and train formation problem under demand uncertainty. RAIRO - Operations Research, 2020, 56(3):1429-1451.

[2] Ramaiah N S , Ahmed S T . An IoT Based Treatment Optimization and Priority Assignment Using Machine Learning. ECS transactions, 2020(1):107. https://doi.org/10.1149/10701.1487ecst

[3] Durand A, Wiesner T, Gardner M A, et al. A machine learning approach for online automated optimization of super-resolution optical microscopy. Nature communications, 2018, 9(1): 1-16. https://doi.org/10.1038/s41467-018-07668-y

[4] Genty G, Salmela L, Dudley J M, et al. Machine learning and applications in ultrafast photonics. Nature Photonics, 2020, 15(2): 91-101. https://doi.org/10.1038/s41566-020-00716-4

[5] Yu M, Yang S, Wu C, et al. Machine learning the Hubbard U parameter in DFT+ U using Bayesian optimization. npj Computational Materials, 2020, 6(1): 1-6. https://doi.org/10.1038/s41524-020-00446-9

[6] Fukami K, Fukagata K, Taira K. Assessment of supervised machine learning methods for fluid flows. Theoretical and Computational Fluid Dynamics, 2020, 34(4): 497-519. https://doi.org/10.1007/s00162-020-00518-y

[7] Owoyele O, Pal P, Vidal Torreira A, et al. Application of an automated machine learning-genetic algorithm (AutoML-GA) coupled with computational fluid dynamics simulations for rapid engine design optimization. International Journal of Engine Research, 2020, 23(9): 1586-1601. https://doi.org/10.1177/14680874211023466

[8] Karthikeyan R, Alli P. Feature selection and parameters optimization of support vector machines based on hybrid glowworm swarm optimization for classification of diabetic retinopathy. Journal of medical systems, 2018, 42(10): 1-11. https://doi.org/10.1007/s10916-018-1055-x

[9] Nain S S, Garg D, Kumar S. Performance evaluation of the WEDM process of aeronautics super alloy. Materials and Manufacturing Processes, 2018, 33(16): 1793-1808. https://doi.org/10.1080/10426914.2018.1476761

[10] Chugh S, Ghosh S, Gulistan A, et al. Machine learning regression approach to the nanophotonic waveguide analyses. Journal of Lightwave Technology, 2019, 37(24): 6080-6089. https://doi.org/10.1109/JLT.2019.2946572

[11] Rittenhouse K J, Vwalika B, Keil A, et al. Improving preterm newborn identification in low-resource settings with machine learning. PLoS One, 2019, 14(2): e0198919. https://doi.org/10.1371/journal.pone.0198919

[12] Berrar D, Lopes P, Dubitzky W. Incorporating domain knowledge in machine learning for soccer outcome prediction. Machine learning, 2019, 108(1): 97-126. https://doi.org/10.1007/s10994-018-5747-8

[13] Kim T, Moon S, Xu K. Information-rich localization microscopy through machine learning. Nature communications, 2019, 10(1): 1-8. https://doi.org/10.1038/s41467-019-10036-z

[14] Klinkowski M, Ksieniewicz P, Jaworski M, et al. Machine learning assisted optimization of dynamic crosstalk-aware spectrally-spatially flexible optical networks. Journal of Lightwave Technology, 2020, 38(7): 1625-1635. https://doi.org/10.1109/JLT.2020.2967087

[15] Culos A, Tsai A S, Stanley N, et al. Integration of mechanistic immunological knowledge into a machine learning pipeline improves predictions. Nature machine intelligence, 2020, 2(10): 619-628. https://doi.org/10.1038/s42256-020-00232-8

[16] Currie G. Intelligent imaging: anatomy of machine learning and deep learning. Journal of nuclear medicine technology, 2019, 47(4): 273-281. https://doi.org/10.2967/jnmt.119.232470

[17] Tyralis H, Papacharalampous G, Langousis A. Super ensemble learning for daily streamflow forecasting: Large-scale demonstration and comparison with multiple machine learning algorithms. Neural Computing and Applications, 2020, 33(8): 3053-3068. https://doi.org/10.1007/s00521-020-05172-3

[18] Banadkooki F B, Ehteram M, Ahmed A N, et al. Enhancement of groundwater-level prediction using an integrated machine learning model optimized by whale algorithm. Natural resources research, 2020, 29(5): 3233-3252. https://doi.org/10.1007/s11053-020-09634-2