

# Automatic Selection and Parameter Optimization of Mathematical Models Based on Machine Learning

Shuangbo Zhang

Hefei 168 High School, Hefei 230000, China

**Abstract.** With the rapid progress of machine learning (ML) technology, more and more ML algorithms have emerged, and the complexity of models is also constantly increasing. This development trend brings two significant challenges in practice: how to choose appropriate algorithm models and how to optimize hyperparameters for these models. In this context, the concept of Automatic Machine Learning (AutoML) has emerged. Due to the applicability of different algorithm models to different data types and problem scenarios, it is crucial to automatically select the most suitable model based on the characteristics of specific tasks. AutoML integrates multiple ML algorithms and automatically filters based on the statistical characteristics of data and task requirements, aiming to provide users with the best model selection solution. Hyperparameters are parameters that ML models need to set before training, such as learning rate, number of iterations, regularization strength, etc., which have a significant impact on the performance of the model. AutoML integrates advanced hyperparameter optimization techniques to automatically find the optimal parameter combination, thereby improving the model's generalization ability and prediction accuracy. This article studies the automatic selection and parameter optimization of mathematical models based on ML.

**Keywords:** Machine learning, mathematical models, automatic selection, parameter optimization.

## 1. Introduction

With the rapid advancement of technology and its widespread penetration in various industries, the scale of data we face is experiencing explosive growth, marking the official entry of society into the era of "big data" [1]. In this ocean of information, big data not only provides us with unprecedented opportunities, but also brings many challenges [2]. To effectively process and analyze these data, ML, as the core science of artificial intelligence, is playing an increasingly important role [3]. ML is an interdisciplinary field whose core lies in optimizing the behavior and performance of computer programs through data or historical experience [4]. In the process of building an ML model, multiple key steps are usually required, including feature engineering, model selection, and parameter tuning [5]. These steps are crucial for ensuring the accuracy, stability, and efficiency of the model. However, in the practical application of ML algorithms, two core issues always accompany us: the selection of algorithm models and hyperparameter optimization. Both of these are directly related to the performance of the model on specific problems [6].

Different algorithm models may have vastly different processing abilities for the same problem, and the reasonable configuration of hyperparameters is a key factor affecting model performance [7]. Superparameters, unlike internal parameters of the model, are parameters set before model training, which determine the learning process and final performance of the model. In order to find the optimal combination of hyperparameters, we need to conduct a large number of experiments and attempts before training begins, which usually requires a lot of computational resources and time. AutoML and autonomous data analysis have become the focus of attention in the technology and industry sectors. They not only demonstrate enormous market potential, but also have a profound impact on social progress and value creation. As a sub major in the field of ML, AutoML greatly simplifies the development process of models through the automation of algorithms and technologies. It reduces manual intervention in feature engineering, model selection, parameter tuning, and other processes, allowing professionals without a strong ML background to easily build and optimize models [8].

With the explosive growth of data volume and the continuous improvement of model complexity, the emergence of AutoML undoubtedly provides an effective way to solve these problems. Autonomous data analysis goes further, not only achieving automation in data processing, but also achieving automatic interpretation and insight into data through ML and AI technologies. This technology can automatically identify patterns, trends, and correlations in data, providing valuable insights and suggestions for decision-makers. In the era of big data, independent data analysis provides more efficient and accurate data analysis and decision support for enterprises and society. The combination of AutoML and autonomous data analysis has brought tremendous changes to the fields of ML and data analysis. They not only improve the efficiency of data analysis and model construction, but also enable more people to benefit from ML and data analysis technologies through automation.

## **2. Model Selection and Parameter Optimization**

### **2.1. Model Selection**

Model selection is a core step in ML, which involves selecting the most suitable model for a specific task and dataset from multiple candidate models [9]. When selecting a model, the main analysis is based on the following aspects. Recognizability refers to whether a model can uniquely determine its parameters from the data. When selecting a model, we hope that the selected model has identifiability, that is, the parameters of the model can be uniquely determined by data, rather than having multiple parameter combinations that can fit the data. A recognizable model can provide more reliable and stable predictions. Flexibility refers to the model's ability to fit complex data. A model with high flexibility can capture complex patterns and nonlinear relationships in data, but it may also lead to overfitting, i.e. performing well on training data but having poor generalization ability on new data. Therefore, when choosing a model, it is necessary to find a balance between flexibility and overfitting. Stinginess is often referred to as the Occam's Razor Principle, which emphasizes that among multiple models, the simplest or most basic model should be chosen, that is, while maintaining sufficient performance, the complexity of the model should be minimized as much as possible. Stinginess helps to prevent overfitting and improve the model's generalization ability. A stingy model typically has fewer parameters and a simpler structure, making it easier to interpret and understand.

### **2.2. Parameter Optimization**

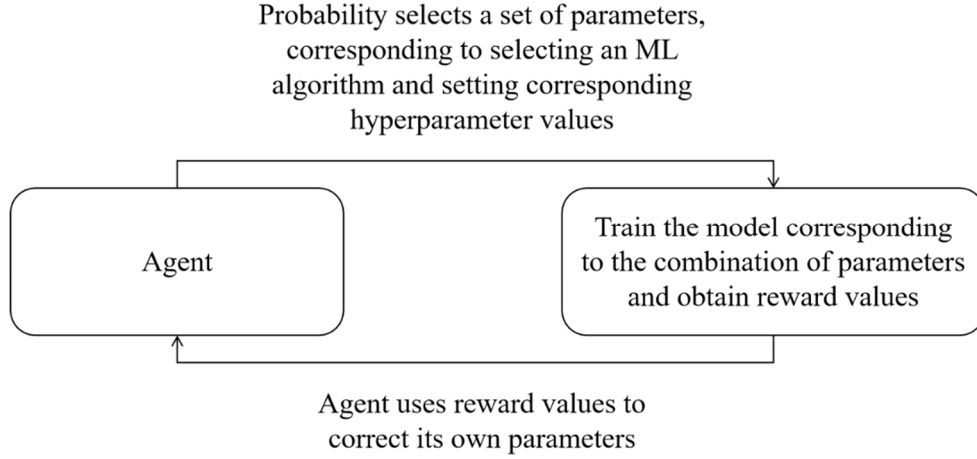
Parameter optimization is an important task in ML, usually carried out after establishing a baseline model [10]. It involves adjusting the model parameters based on optimization algorithms, aiming to improve the performance of the model. Parameter optimization can be divided into two main aspects: model selection and hyperparameter optimization. Hyperparameter optimization is the search for hyperparameters that enable ML algorithms to perform best on validation datasets. Hyperparameters are set in advance before training, unlike general model parameters. Hyperparameter optimization aims to find a set of hyperparameters that return an optimization model that can reduce predefined loss functions and improve the prediction or classification accuracy of given independent data. There are many methods for hyperparameter optimization, such as grid search, random search, Bayesian optimization, and genetic algorithm. The goal of these methods is to find the optimal combination of hyperparameters within a reasonable time range to improve the performance of the model. Overall, parameter optimization is a key step in ML, which helps to improve the predictive and generalization capabilities of the model, thereby achieving better performance in various application scenarios.

## **3. Algorithm Selection and Experimental Results**

### **3.1. Algorithm Selection**

The hyperparameter optimization problem can be seen as a multi-stage decision problem, where decisions are made on the value of a certain hyperparameter at each stage, resulting in different outputs at different times. Reinforcement learning (RL) is a machine learning method that has shown

great potential and advantages in solving decision problems. The core idea of RL is to learn how to make decisions to maximize a certain cumulative reward through the interaction between agents and the environment. In RL, the agent observes the state of the environment, takes action based on these states, and receives reward or punishment signals from the environment. This process is repeated, and the Agent learns which actions and states can lead to higher cumulative rewards by constantly trying and making mistakes. Over time, agents gradually learn how to make optimal decisions to maximize long-term benefits. The RL algorithm structure is shown in Figure 1.



**Figure 1.** RL algorithm structure

When the Agent selects the model/hyperparameter sequence  $a_{1:T}$  with probability  $P$ , the algorithm model corresponding to  $a_{1:T}$  is trained on the training dataset to converge, and the accuracy obtained by running it on the validation dataset is used as a reward signal  $R$  to optimize the Agent's parameter  $\theta$ , so that over time, the Agent will choose a model/hyperparameter combination with higher accuracy. The agent training method adopts the policy gradient in the RL algorithm, and the optimization objective of the algorithm is to maximize the expected total reward:

$$J(\theta) = \max E_{P(a_{1:T}; \theta)}[R] \quad (1)$$

In the formula,  $P(a_{1:T}; \theta)$  represents the probability of the Agent outputting the model/hyperparameter sequence  $a_{1:T}$ .

Due to the optimization goal of finding a parameter  $\theta$  to maximize the expected total reward. The gradient descent algorithm is a commonly used optimization method in RL, used to find the parameter  $\theta$  that can maximize the expected total reward:

$$\nabla_{\theta}(J(\theta)) = \sum_{t=1}^T E_{P(a_{1:T}; \theta)}[R \nabla_{\theta} \log P(a_t | a_{(t-1):1}; \theta)] \quad (2)$$

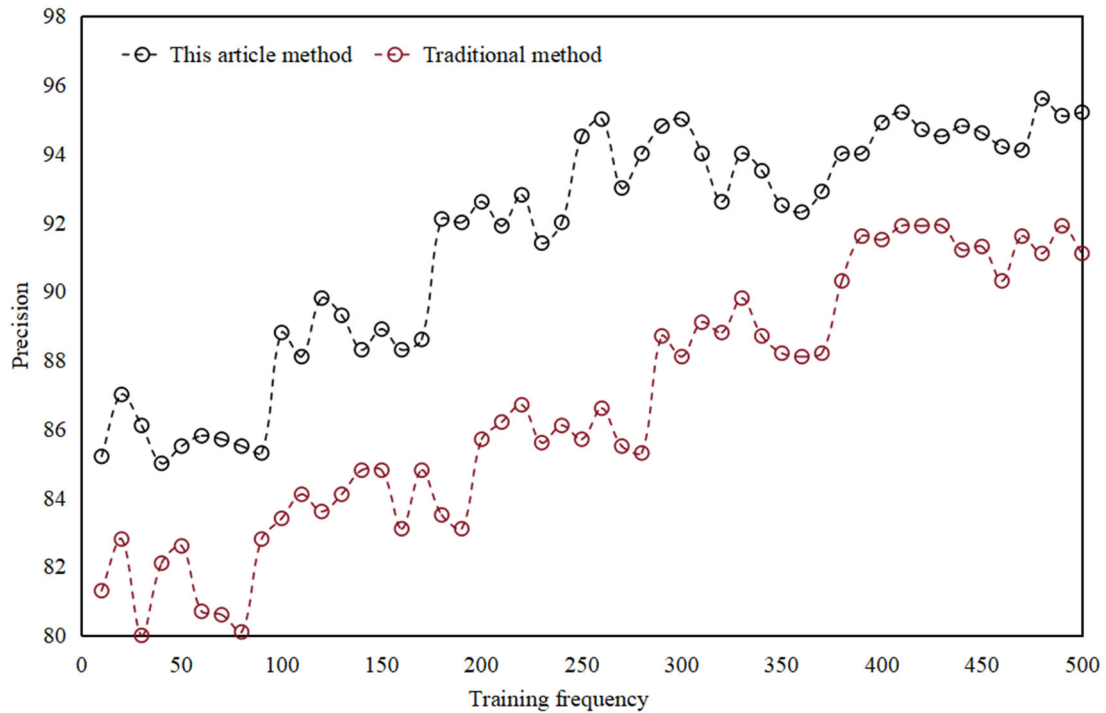
By conducting  $m$  samples, unbiased estimation is performed on the expectation, which is generally obtained by sampling an intelligent agent with an internal parameter of  $\theta$ , i.e. iteratively outputting  $m$  automated model parameter combinations  $a_{1:T}$ . By using the mean of the loss function values of  $m$  samples as an unbiased estimate of the gradient value:

$$\nabla_{\theta}(J(\theta)) \approx \frac{1}{m} \sum_{k=1}^m \sum_{t=1}^T \nabla_{\theta} \log P(a_t | a_{(t-1):1}; \theta) (R_k - b) \quad (3)$$

### 3.2. Experimental Result

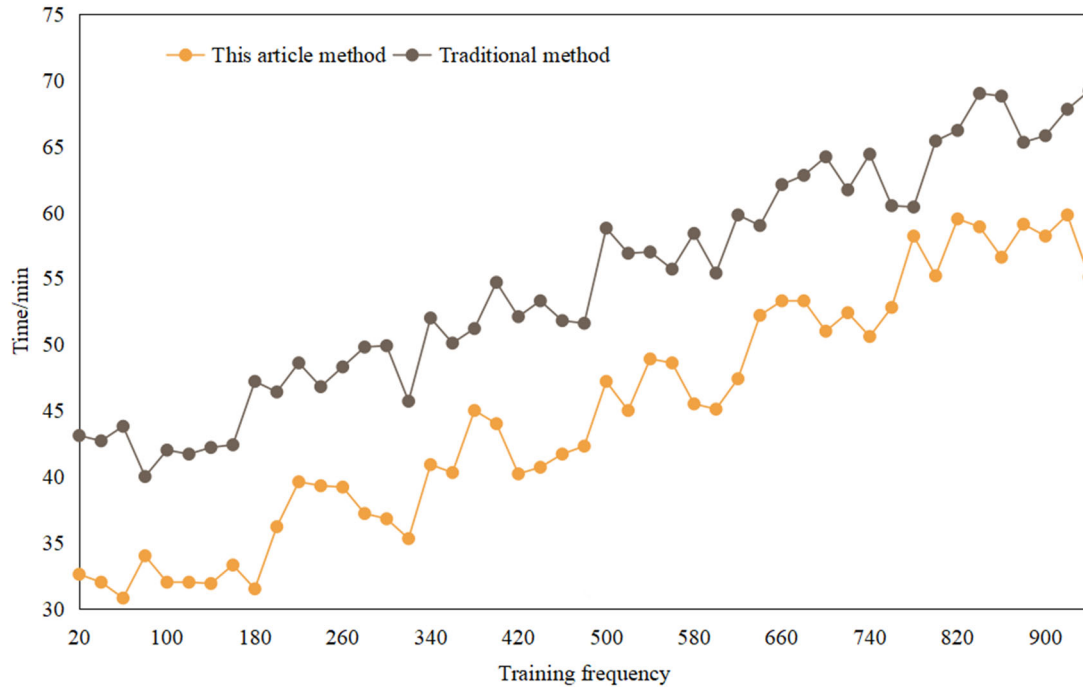
The experiment was conducted on the UCI standard dataset, and the accuracy and time efficiency of our method were compared with traditional methods. Figure 2 shows the accuracy comparison when using two different methods during the optimization process. The horizontal axis represents the

number of training iterations, i.e. the number of algorithm iterations; The vertical axis represents the accuracy of the selected model on the validation set after each training, that is, the predictive ability of the model on new data. We see that the accuracy increases with the number of training sessions, indicating that the model is gradually learning and adapting to the data. The optimization method proposed in this article has significant advantages in accuracy compared to traditional methods. It can be clearly seen from the graph that the accuracy curve of our method is always above the traditional method, which means that under the same number of training times, the model selected by our method always performs better on the validation set than the traditional method.



**Figure 2.** Comparison of accuracy of different methods

Figure 3 shows a comparison of the time consumption of different optimization methods during the training process. The horizontal axis represents the number of training or iterations, while the vertical axis represents the time spent on training. From the graph, it can be seen that the training time curve of our method always falls below that of other methods, which means that our method consumes less time under the same number of training iterations. Time performance is a very important consideration in ML and optimization problems. In practical applications, we often need to achieve the best possible model performance within a limited time. Therefore, an optimization method that can converge quickly and takes less time is usually more popular. This method has advantages in accuracy and time performance. It not only provides higher accuracy, but also requires less time during the training process. This makes the method proposed in this article more competitive in practical applications, especially in scenarios that require rapid response or processing of large amounts of data.



**Figure 3.** Comparison of training time

#### 4. Conclusions

Choosing the appropriate algorithm model and optimizing hyperparameters are the two core challenges in ML. The choice of model directly affects whether the task can be effectively solved, and hyperparameter optimization further determines the upper limit of model performance. For these two issues, traditional methods usually require a large amount of manual experimentation and adjustment, which is not only time-consuming, but also often depends on the experience and intuition of experts. The ML based automatic model selection and parameter optimization method proposed in this article transforms the model selection and hyperparameter optimization problem into a decision-making process by introducing the RL algorithm. We can use the RL algorithm to learn a strategy that can make choices between different models and hyperparameter configurations to maximize long-term rewards (i.e. model performance). The experimental results show that this method can significantly improve model performance, which fully demonstrates the effectiveness of RL based model automatic selection and parameter optimization methods. In addition, this method also has good universality and can be applied to different machine learning tasks and models. By transforming model selection and hyperparameter optimization problems into a decision-making process and utilizing RL algorithms for learning, we can achieve efficient automated selection and optimization of models and parameters. This not only greatly saves labor and time costs, but also has the potential to discover better model configurations, thereby improving model performance.

#### References

- [1] Wu Jia, Chen Senpeng, Chen Xiuyun, et al. Model selection and hyperparametric optimization based on reinforcement learning [J]. Journal of University of Electronic Science and Technology of China, 2020, 49(2):7.
- [2] Yan Dong, Yang Fubin, Zhang Hongguang, et al. Identification of key parameter sets of organic Rankine cycle machine learning model [J]. Journal of Harbin Engineering University, 2023, 44(8):1368-1374.
- [3] Zhang Shuxing, Jiang Hongwei. Theoretical analysis and research on parameter identifiability of knowledge and data-driven machine learning model [J]. Science and Technology Innovation and Application, 2020(9):2.
- [4] Liu Hui, Zhao Rongcai, Wang Qi. Research on the parameter selection method of function-level compilation optimization guided by supervised learning model [J]. Computer Engineering and Science, 2018, 040(006):957-968.
- [5] Cai Ming, Sun Jie, Yang Weifa, et al. Parameter optimization method of stochastic forest model based on adaptive genetic algorithm [J]. Intelligent Computer and Application, 2022, 12(12):175-179.

- [6] Xu Junfeng, Luo Yuanxin, Wang Chuanyao. Rubber constitutive model selection and structural parameter optimization of a new packer [J]. Manufacturing Automation, 2019, 41(1):5.
- [7] Chen Yuanfeng, Ma Xiyuan, Cheng Kai, et al. Ultra-short-term power prediction of new energy based on the selection of meteorological features and the parameter optimization of SVM model [J]. Acta Solar Energy, 2023(012):044.
- [8] Zhan Huawei, Tang Yan, Fu Jing. Parameter Optimization and Function Selection of Convolutional Neural Networks [J]. Journal of taiyuan normal University: Natural Science Edition, 2020, 19(1):4.
- [9] Liu Hui, Yao Jinyang, Zhao Rongcai. Research on the Selection Method of Compiler Optimization Parameters Based on Integrated Learning [J]. Journal of Information Engineering University, 2019, 20(3):9.
- [10] Gao Xiaoming. Optimization analysis of professional technology and skill quality evaluation model based on machine learning [J]. Bonding, 2023, 50(6):140-143.