Prediction of Reactor Core Neutronics Parameters Using Neural Network Hyperparameter Optimization Methods

Authors: Zhang Fan, Zhang Junda, Sun Qizheng, Chauvet, Liu Xiaojing, Zhang Tengfei, Zhang Tengfei

Date: 2024-06-04T00:00:00+00:00

Abstract

Neural networks can learn the relationships between input and output variables based on large datasets, possess powerful fitting capabilities, and are commonly employed as surrogate models for programs in fields including nuclear engineering calculations. Neutron transport calculations, as one of the core components of neutronics simulation, suffer from time-consuming issues that can be addressed through the utilization of neural network models. However, neural network models entail a series of hyperparameters requiring configuration, and manual tuning of these hyperparameters is labor-intensive, repetitive, tedious, relies exclusively on empirical experience, and such hyperparameters are not transferable across different problems. To resolve these issues, this paper proposes a methodology employing the Bayesian Optimization algorithm to tune neural network hyperparameters, incorporating learning rate decay and loss function optimization techniques. This approach can automatically search for optimal hyperparameter combinations tailored to datasets from different problems to achieve peak performance, demonstrating high flexibility, efficiency, and strong generalization capability. This study fits key core parameters derived from the TAKEDA benchmark problem, with results indicating that the average error of the effective multiplication factor keff remains within 150 pcm, while the average error rate of regional integrated flux Φ on the TAKEDA1 dataset is 1.72% and the maximum error rate is 7.56%. This research can provide valuable reference for the application of artificial intelligence in core physics calculation theory.

Full Text

Preamble

Vol. XX, No. X, Month 20XX
NUCLEAR TECHNIQUES ChinaXiv Vol.XX, No.X XXX 20XX

Research on Core Neutronic Parameter Prediction Based on Neural Network Hyperparameter Optimization Method

ZHANG Fan[1], ZHANG Junda[2], SUN Qizheng[2], XIAO Wei[2], LIU Xiaojing[2], ZHANG Tengfei[2]
[1](College of Smart Energy, Shanghai Jiao Tong University, Shanghai 200240, China)
[2](School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract

Neural networks possess powerful fitting capabilities and can learn complex relationships between input and output variables from large datasets, making them effective surrogate models for physics-based codes in nuclear engineering calculations. Neutron transport calculations represent a core component of neutronics simulations but suffer from prohibitive computational costs. While neural networks offer a promising solution to this challenge, they require careful tuning of numerous hyperparameters—a process that is labor-intensive, experience-dependent, and problem-specific, with limited transferability across different applications.

To address these limitations, this paper proposes an automated hyperparameter optimization framework that integrates Bayesian optimization with adaptive learning rate decay and loss function optimization. This approach automatically searches for optimal hyperparameter configurations tailored to specific datasets, delivering superior performance with high flexibility, efficiency, and generalization capability. Using datasets generated from the TAKEDA benchmark problems via the VITAS code, we demonstrate that the optimized neural network achieves an average error in the effective multiplication factor (k_{eff}) within 150×10^{-5}. For the TAKEDA1 dataset, the regional integral flux (f) exhibits an average error rate of 1.72% and a maximum error rate of 7.56%. These results validate the feasibility and advantages of neural network-based parameter prediction in reactor physics calculations and provide valuable insights for advancing artificial intelligence applications in nuclear engineering.

Keywords: Bayesian hyperparameter optimization, fully connected neural network, neutron transport calculation, learning rate decay, loss function optimization methods

Introduction

In nuclear reactor physics calculations, the Monte Carlo method and deterministic approaches constitute the two primary classes of techniques for solving neutron transport problems. Commonly employed deterministic methods include the Finite Element Method (FEM), Method of Characteristics (MOC), and Variational Nodal Method (VNM). However, these traditional neutron transport methods share a common drawback of low computational efficiency and long execution times, rendering them impractical for frequent core design optimization and fuel reloading studies.

Machine learning and artificial intelligence offer distinct advantages in handling complex, large-scale data processing tasks. In recent years, interdisciplinary applications at the intersection of nuclear physics and machine learning have emerged with promising prospects. For instance, Song et al. developed a core parameter prediction program based on back-propagation (BP) artificial neural networks achieving relative errors within 10%, which sparked subsequent research interest. Akkoyun employed artificial neural networks to predict total fusion and fusion-evaporation reaction cross-sections, while Guo et al. utilized a hybrid artificial neural network to simulate the thermodynamic behavior of the Sequoia nuclear power plant. Among these approaches, the fully connected neural network (FCNN) represents a relatively simple yet versatile artificial neural network architecture belonging to the family of BP neural networks. Comprising only an input layer, hidden layers, and an output layer—with multiple neurons possible in each hidden layer—FCNNs serve as a universal learning method applicable to nearly all tasks including classification, regression, and unsupervised learning. The ability of machine learning to approximate complex analytical computations while reducing manual and computational costs has fueled its rapid development.

VITAS is a general-purpose computational code developed based on VNM for accurately solving neutron transport problems. Integrating multiple computational methods and leveraging matrix operations and numerical integration techniques, VITAS can handle multi-dimensional, multi-group, steady-state, and transient neutron transport problems across various mesh types. To overcome the computational inefficiencies inherent in traditional transport methods, this study develops a surrogate model for core transport calculations based on neural networks to replace the computational functionality of the VITAS program.

However, neural networks involve numerous hyperparameters that critically determine FCNN performance. Manual hyperparameter tuning is not only burdensome and experience-dependent but also becomes inadequate as deep learning models involve far more hyperparameters than traditional machine learning approaches. In recent years, Bayesian optimization has gained widespread application in solving black-box function problems and has become the mainstream method for hyperparameter optimization. Also known as Sequential Kriging Optimization (SKO), Sequential Model-Based Optimization (SMBO),

or Efficient Global Optimization (EGO) across different domains, Bayesian optimization is a global optimization approach that employs a surrogate model to approximate the expensive-to-evaluate objective function. By actively selecting the most "promising" evaluation points based on the surrogate model and leveraging complete historical information to improve search efficiency, it can find optimal solutions for complex functions with minimal evaluations—hence its designation as "active optimization." The Bayesian optimization framework effectively utilizes historical information to enhance search efficiency while maintaining computational economy.

This study employs Bayesian optimization combined with learning rate decay and loss function optimization methods to systematically optimize FCNN hyperparameters. Using experimental data computed by VITAS for the TAKEDA benchmark problems, we evaluate the prediction accuracy of the hyperparameter-optimized FCNN and compare it against manually tuned networks. Our objective is to provide a more efficient approach for solving core reloading optimization problems and to explore further applications of artificial intelligence in the nuclear industry.

1.1 Bayesian Optimization Method

The Bayesian optimization framework comprises two key components: (1) a probabilistic surrogate model that approximates the expensive-to-evaluate objective function, and (2) an acquisition function constructed from the surrogate model's posterior information to actively select evaluation points. In practice, appropriate models must be selected for specific problems. This section defines the hyperparameter optimization problem and outlines the Bayesian optimization procedure, concluding with the selected FCNN hyperparameters and their search ranges.

The hyperparameter optimization problem is defined as:

$$\arg\min_{x \in X} f(x)$$

where x represents a specific hyperparameter configuration and X denotes the hyperparameter search space. The function $f(x)$ represents the objective to be optimized. In this study, we employ the loss function as $f(x)$; the selection and comparison of loss functions are discussed in Section 1.2(2) on loss function optimization. The goal of hyperparameter optimization is to find the global optimum as efficiently as possible.

The Bayesian optimization procedure proceeds as follows:

1. Randomly initialize n_0 hyperparameter configurations x_{init} in the search space X;
2. Obtain their corresponding function values $f(x)$ to form the initial dataset $D_0 = \{(x_i, f(x_i))\}$;
3. Construct a surrogate model $g(x)$ based on the current dataset distribution;

4. Maximize the acquisition function based on $g(x)$ to obtain the next evaluation point: $x_t = \arg\max_{x \in X} \alpha(x; D_{t-1})$;
5. Evaluate $f(x_t)$ and add it to the dataset: $D_t = D_{t-1} \cup \{(x_t, f(x_t))\}$. If t is less than the maximum iteration count N, return to step 3;
6. Upon reaching the maximum iteration count, output the optimal evaluation point $\{x^*, f(x^*)\}$.

This study implements the Bayesian optimization process using TensorFlow's Bayesian Optimization function. By specifying hyperparameters and their value ranges, the optimization is initiated. The selected FCNN hyperparameters and their ranges are presented in Table 1 .

Table 1 Selected hyperparameters and their ranges

Hyperparameters	Ranges
Batch size (batch_{size})	[1, 1000]
Number of hidden layers (num_{{{hidden}}_{{layers}}})	[1, 5]
Threshold (min_{delta})	[1e-6, 1e-4]
Decay factor (factor)	[0.1, 0.9]
Number of neurons in hidden layers (num_{neurons})	[1, 2000]
Loss function parameter (loss_{delta})	[0.1, 10]

All hyperparameters listed in the table are critical parameters in FCNN, as discussed below.

1.2 Fully Connected Neural Network

For the surrogate model in neutron transport solving, this work selects the fully connected neural network (FCNN). FCNN is a relatively simple artificial neural network architecture belonging to the family of feedforward neural networks, consisting only of an input layer, hidden layers, and an output layer, with multiple neurons possible in each hidden layer, as illustrated in Figure 1 [Figure 1: see original paper].

The entire network is built using TensorFlow, an open-source deep learning framework developed by Google in 2015. Key parameters in FCNN training include:

Epoch: The process of feeding the entire training dataset through the neural network once constitutes one epoch.

Batch Size: Due to computational constraints or other considerations, when the entire dataset cannot be processed simultaneously, the dataset is divided into multiple batches. A small subset of training samples is used for each backward pass to update model weights.

Learning Rate (lr): This parameter determines the step size for adjusting model parameters during neural network training, directly influencing model performance and training effectiveness.

(1) Adaptive Learning Rate Adjustment

An excessively large learning rate increases step size, causing oscillations that prevent convergence to high-precision solutions. Figure 2 [Figure 2: see original paper] (left) demonstrates loss function oscillation caused by an overly large learning rate, resulting in unstable convergence. Conversely, an excessively small learning rate slows convergence and risks getting trapped in local optima. To address this, we propose an adaptive learning rate adjustment strategy: starting with a relatively large learning rate for rapid approximation, then gradually decreasing it to accommodate the model parameter update dynamics, eliminating late-stage loss function oscillations and stabilizing the loss curve. Figure 2 (right) shows the loss function decay curve after adaptive adjustment, which effectively eliminates oscillations.

This study employs the ReduceLROnPlateau function from TensorFlow, a learning rate scheduler that automatically adjusts the learning rate based on monitored metric changes. When performance on the validation set stops improving, ReduceLROnPlateau gradually reduces the learning rate to facilitate better model convergence. Its key parameters include:

- **lr**: Initial learning rate value;
- **factor**: The factor by which the learning rate is reduced each time, applied as $lr \leftarrow lr \times factor$;
- **patience**: The number of epochs to wait without improvement before triggering a learning rate reduction;
- **min_{delta}**: Threshold value; when the model performance improvement is less than min_{delta}, it is considered no improvement.

Both factor and min_{delta} are selected as hyperparameters for Bayesian optimization in this study.

(2) Loss Function Optimization

Mean Square Error (MSE) measures the average squared difference between predicted values $f(x)$ and true values y:

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2$$

Mean Absolute Error (MAE) measures the average absolute difference between predictions and true values:

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}|y_i - f(x_i)|$$

where y_i and $f(x_i)$ represent the true and predicted values for the i-th sample, respectively, and m is the number of samples.

A major issue with using MAE as a loss function is its consistently large gradient, which may cause gradient descent to overshoot the minimum. In contrast, MSE's gradient diminishes as the loss approaches its minimum, enabling more precise convergence. This limitation can be mitigated using Huber Loss, which combines the advantages of both MSE and MAE. Huber Loss reduces gradients near the minimum (addressing MAE's overshooting problem) while being more robust to outliers than MSE. This study adopts Huber Loss as the loss function:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{if } |y - f(x)| \leq \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}$$

Huber Loss integrates MSE and MAE through the hyperparameter δ, which determines the emphasis on each component. As $\delta \to 0$, Huber Loss approaches MAE; as $\delta \to \infty$, it approaches MSE. In Table 1, δ corresponds to loss_{delta}.

Since these parameter settings critically determine FCNN performance, they are designated as hyperparameters for Bayesian optimization, with their search spaces defined in Table 1.

1.3 Dataset Acquisition

The datasets used in this study are generated through VITAS calculations of the TAKEDA1 and TAKEDA2 benchmark problems, comprising 10,000 and 20,000 data groups, respectively. Each data sample consists of a core loading pattern and two core parameters: the effective multiplication factor (k_{eff}) and the regional integral flux (f). Input dimensions are determined by the reactor core loading configuration, while output dimensions correspond to core parameters. TAKEDA1 models a 1/8 symmetric light water reactor, whereas TAKEDA2 models a 1/4 fast breeder reactor.

TAKEDA1 includes three component types: Control Rod (CR), Reflector, and Fuel (Core). Its three-dimensional schematic and x-y cross-sectional core layout are shown in Figure 3 [Figure 3: see original paper]. To ensure physical meaningfulness, the outermost reflector layer in the 1/4 x-y cross-section is fixed while the remaining components are shuffled and then mirrored to form a complete core, as illustrated in Figure 4 [Figure 4: see original paper]. For FCNN processing, the three components are mapped to -1, 0, and 1, respectively. This generates 10,000 distinct random configurations, where each input vector's sequence of -1, 0, 1 corresponds to the shuffled component arrangement. The input dimension is 16, and two independent FCNNs are trained to predict the two physical quantities: k_{eff} (output dimension: 1) and f (output dimension: 6).

TAKEDA2 includes three component types: Control Rod (CR), Axial Blanket, and Radial Blanket, as shown in Figure 5 [Figure 5: see original paper]. To

maintain physical validity, the outermost three layers of Radial Blanket in the 1/4 x-y cross-section are fixed while other components are shuffled and mirrored, as depicted in Figure 6 [Figure 6: see original paper]. The components are mapped to -1, 0, and 1 for FCNN processing, generating 20,000 distinct random configurations. Each input vector dimension is 121, with two independent FCNNs predicting k_{eff} (output dimension: 1) and f (output dimension: 20).

For FCNN training, raw data must be converted into a standardized dataset. Assuming N training samples, each sample should be:

$$T_k = (X_k, Y_k), \quad k = 1, 2, \ldots, N$$

where X_k and Y_k represent the input and output variables for the k-th sample, with dimensions i and j, respectively. For TAKEDA1: $k = 10,000$, input vectors are sequences of -1, 0, 1 with dimension $i = 16$, and two FCNNs predict k_{eff} ($j = 1$) and f ($j = 6$). For TAKEDA2: $k = 20,000$, input dimension $i = 121$, with two FCNNs predicting k_{eff} ($j = 1$) and f ($j = 20$).

1.4 Model Validation

This study utilizes 10,000 TAKEDA1 samples and 20,000 TAKEDA2 samples, with inputs representing reactor core loading patterns and outputs comprising k_{eff} and f. The datasets are partitioned into training and validation sets at a 6:4 ratio. The algorithm flowchart is presented in Figure 7 [Figure 7: see original paper].

Using TAKEDA1 as an example, Table 2 compares manually specified hyperparameters against those obtained through Bayesian optimization:

Table 2 Hyperparameter selection for TAKEDA1

Hyperparameters	Manual Setting	Bayesian Optimization
Batch_{size}	64	85
Hidden_{layers}	4	5
Min_{delta}	5.5e-5	4.5e-5
Factor	0.5	0.3
Num_{neurons}	800, 900, 1600, 800	400, 800, 1800, 600, 50
Loss_{delta}	1.248e-5	8.927e-05

The training process curves for the two prediction models on TAKEDA1 are presented below.

Figure 8 [Figure 8: see original paper] compares loss function curves between Bayesian optimization (left) and manual tuning (right) for k_{eff}. With decreasing learning rate, the training loss gradually declines. The test loss for the Bayesian-optimized model stabilizes after 400 epochs, whereas the manually tuned model

exhibits severe oscillation and overfitting, with test loss initially decreasing then increasing.

Figure 9 [Figure 9: see original paper] presents similar comparisons for f. The Bayesian-optimized model shows stable test loss convergence after 250 epochs, while the manually tuned model suffers from overfitting and dramatic oscillations, with test loss increasing despite decreasing training loss.

Results and Discussion

Evaluation Metrics: Definitions for MAE and MSE are provided in equations (2) and (3). Mean Absolute Percentage Error (MAPE) and coefficient of determination (R^2) are defined as:

$$\text{MAPE} = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%$$

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2}$$

where y_i denotes true values, $\hat{y}_i$ predicted values, and $\bar{y}$ the sample mean. R^2 measures the proportion of variance in the dependent variable explained by the independent variables, ranging from 0 to 1, with values closer to 1 indicating better fit.

Tables 3 and 4 compare prediction errors for k_{eff} and f between Bayesian-optimized and manually tuned FCNN models on TAKEDA1 and TAKEDA2 datasets.

Table 3 Comparison of errors between Bayesian optimization and manual hyperparameter tuning for k_{eff}

Metric	TAKEDA1	TAKEDA2
	Manual	Bayesian
MAE ($\times 10^{-5}$)	1525	118
MSE ($\times 10^{-5}$)	2800	180
R^2	0.95	0.999

Bayesian optimization demonstrates significant advantages over manual tuning in both error magnitude and goodness-of-fit. The Bayesian-optimized models achieve MAE of 118×10^{-5} for TAKEDA1 and 132×10^{-5} for TAKEDA2—approximately 1/13 and 1/12 of the manual tuning errors, respectively. The R^2 values exceed 0.999, substantially outperforming manual tuning results.

Table 4 Comparison of errors between Bayesian optimization and manual hyperparameter tuning for f

Metric	TAKEDA1	TAKEDA2
	Manual	Bayesian
MAPE (%)	15.5	1.72
Max Percentage Error (%)	380	7.56
R^2	0.85	0.99

For f, Bayesian optimization achieves MAPE of 1.72% (maximum 7.56%) on TAKEDA1 and 0.82% (maximum 11.13%) on TAKEDA2—representing improvements of approximately 9× and 5× over manual tuning, respectively. Maximum errors are reduced by factors of 50× and 8×. The R^2 values around 0.99 confirm superior fitting performance.

Figure 10 [Figure 10: see original paper] shows error distribution histograms for k_{eff} predictions. Errors follow normal distributions with high concentration: 90% of errors fall within 500×10^{-5}, and 50% within 150×10^{-5} for both benchmarks.

Figure 11 [Figure 11: see original paper] presents percentage error distributions for f. Both benchmarks exhibit normal distributions, with TAKEDA2 showing a tighter, "taller" distribution. For TAKEDA1, 90% of percentage errors are within 3.5%; for TAKEDA2, 90% are within 1%.

Conclusion

To enable more efficient core reloading optimization, this study proposes a Bayesian optimization framework for neural network hyperparameters as a surrogate model for core transport calculations. Using datasets from TAKEDA1 and TAKEDA2 benchmarks, we trained and validated the approach against manual tuning, yielding the following key findings:

1. The Bayesian-optimized FCNN effectively approximates VITAS code results, achieving average k_{eff} errors within 150×10^{-5}, TAKEDA1 f MAPE of 1.72%, and TAKEDA2 f MAPE of 0.82%—all within acceptable accuracy limits.

2. FCNN models built with Bayesian-optimized hyperparameters significantly outperform manually tuned models in both prediction error and goodness-of-fit metrics.

3. While the specific hyperparameters and FCNN architecture are tailored to the TAKEDA1 and TAKEDA2 datasets from VITAS, the framework is generalizable. For predicting other core parameters, one can substitute the corresponding dataset and follow the workflow in Figure 7 to obtain new optimal hyperparameter configurations, eliminating manual tuning efforts.

4. This work validates the feasibility and advantages of neural network-based parameter prediction in reactor physics calculations. Further exploration of different neural network architectures for nuclear engineering applications represents a promising research direction.

Author Contributions: ZHANG Fan performed simulations and drafted the manuscript; ZHANG Junda, SUN Qizheng, and XIAO Wei provided technical support; LIU Xiaojing secured funding; ZHANG Tengfei secured funding and revised the manuscript.

References

1. Kazuteru S, Toshikazu T. An improvement of the transverse leakage treatment for the nodal SN transport calculation method in hexagonal-z geometry[J]. Journal of Nuclear Science and Technology, 1996(8). DOI:10.3327/jnst.33.620.

2. Palmiotti G, Lewis E E, Carrico C B. VARIANT: VARIational Anisotropic Nodal Transport for multidimensional Cartesian and hexagonal geometry calculation: ANL-95/40[R]. Argonne, IL, USA: Argonne National Laboratory, 1995.

3. Zhang T F, Wu H C, Cao L Z, et al. An improved variational nodal method for the solution of the three dimensional steady-state multi-group neutron transport equation[J]. Nuclear Engineering Design, 2018, 419-427. DOI:10.1016/j.nucengdes.2018.07.009.

4. Zhang Tengfei, Zhu Lei, Wu Hongchun, et al. Development and validation of high flux engineering test reactor fuel management platform HEFT[J]. Atomic Energy Science Technology, 2013, 47(S1): 467-471. DOI: 10.7538/yzk.2013.47.S1.0467.

5. Ji Nan, Yi Jinhao, Zhao Pengcheng, et al. Research on adaptive RBF neural network prediction method for core thermal-hydraulic parameters reactor[J]. Nuclear Techniques, 2022, 45(09): 65-74. DOI: 10.11889/j.0253-3219.2022.hjs.45.090601.

6. Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. Nature, 1986, 323: 533-536. DOI:10.1038/323533a0.

7. Huang Yu, Liu Junqiang, Liu Le. Application of BP neural network in DNBR prediction[J]. Nuclear Techniques, 2015, 38(7): 070606. DOI: 10.11889/j.0253-3219.2015.hjs.38.070606.

8. Lyu Y Q, Zhou Q W, Li Y F, et al. A predictive maintenance system for multi-granularity faults based on AdaBelief-BP neural network and fuzzy decision making[J]. Advanced Engineering Informatics, 2021, 49: 101318. DOI: 10.1016/j.aei.2021.101318.

9. He Can, Qiu Xiaoping, Sun Zheng, et al. Radiation shielding optimization of space reactor based on intelligent decision support system[J]. Nuclear Techniques, 2022, 45(05): 65-75. DOI: 10.11889/j.0253-3219.2022.hjs.45.050601.

10. Akkoyun S. Estimation of fusion reaction cross-sections by artificial neural networks[J]. Nuclear Instruments and Methods in Physics Research B, 2020, 462: 51-54. DOI:10.1016/j.nimb.2019.11.014.

11. Guo Z, Uhrig R E. Use of artificial neural networks to analyze nuclear power plant performance[J]. Nuclear Technology, 1992, 99(1): 36-42. DOI:10.13182/NT92-A34701.

12. Hou S H, Fang S L, Zeng Q Y, et al. Element grouping faceted fully connected network based on RIS[J]. Journal of System Simulation, 2024, 36(4): 1017-1027.

13. Zhang T F, Xiao W, Yin H, et al. VITAS: A multi-purpose simulation code for the solution of neutron transport problems based on variational nodal methods[J]. Annals of Nuclear Energy, 2022. DOI:10.1016/j.anucene.2022.109335.

14. Zhang Tengfei, Yin Han, Sun Qizheng, et al. Application research on VITAS — a general-purpose neutron transport code[J]. Nuclear Power Engineering, 2023, 44(02): 15-23. DOI:10.13832/j.jnpe.2023.02.0015.

15. Tang Lin, Li Yong, Tang Yufeng, et al. Application of an LSTM model based on deep learning through X-ray fluorescence spectroscopy[J]. Nuclear Techniques, 2023, 46(07): 070502. DOI: 10.11889/j.0253-3219.2023.hjs.46.070502.

16. Hosain M T, Jim J R, Mridha M F, et al. Explainable AI approaches in deep learning: Advancements, applications and challenges[J]. Computers and Electrical Engineering, 2024, 117. DOI:10.1016/j.compeleceng.2024.109246.

17. Cui Jiaxu, Yang Bo. Survey on Bayesian optimization methodology applications[J]. Journal of Software, 2018, 29(10): 3068-3090. DOI:10.13328/j.cnki.jos.005607.

18. Ghahramani. Probabilistic machine learning artificial intelligence[J]. Nature, 2015, 521: 452-459. DOI:10.1038/nature14541.

19. Martin A, Paul B, Jianmin C, et al. TensorFlow: A system for large-scale machine learning[J]. USENIX Association, 2016. DOI:10.48550/arXiv.1605.08695.

Note: Figure translations are in progress. See original paper for figures.