Proceedings of the 3rd International Conference on Mechatronics and Smart Systems
DOI: 10.54254/2755-2721/147/2025.22721

Automated Neural Network Architecture Design and Hyperparameter Adjustment Integrating Distribution-Based Optimization Strategy

Liuding Sun

University Putra Malaysia, Serdang, Selangor, Malaysia
liudingsun@outlook.com

Abstract: Collaborative design and optimization of neural network structure and hyperparameter is the key to improving the efficiency of deep learning model. Conventional methods such as manual tuning and grid search often consume large computational resources and have limited efficiency when facing complex parameter combinations. In this paper, an automated scheme based on probability distribution modeling is developed to realize the synchronous optimization of network architecture design and hyperparameter tuning. The method innovatively transforms various parameters into dynamically adjusted probability distributions and drives the evolution of distribution morphology through a continuous iterative performance feedback mechanism. Based on the intelligent exploration strategy of unified probability space, this scheme can effectively overcome the dimensional disaster problem of traditional methods in high-dimensional configuration space, and can quickly locate the high-performance model configuration under limited computational resources. Comparative experiments on cifar 10, Fashion-MNIST, and IMDB show that the proposed scheme is significantly superior to manual design, grid search, and Bayesian optimization in classification accuracy, model stability, and computing power utilization. This achievement provides a new technical direction for the field of machine learning, and its basic mechanism can be extended to many fields such as image recognition and text processing, which has practical value in promoting the efficiency of intelligent algorithm development.

Keywords: Neural Architecture Search, Hyperparameter Optimization, Distribution-Based Optimization, AutoML, Deep Learning

1. Introduction

The rapid development of deep learning technology promotes the wide application of neural networks in image recognition, semantic analysis, and recommendation systems. The performance of the model mainly depends on two central elements: the design of the network structure and the parameter tuning strategy. To determine the optimal number of layers, the filter dimension, the type of activation function, and the setting of training parameters, it is necessary to deeply understand the data characteristics and the model's working mechanism. For a long time, researchers mainly rely on exhaustive methods such as manual trial and error or grid search to filter effective configurations, but these methods often face the dilemma of excessive computing time and energy consumption when the parameter dimension increases. In recent years, automated machine learning technology has

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/147/2025.22721

provided a new idea for intelligent configuration of the optimization model, and the probability distribution optimization method has attracted much attention due to its ability to model the complex parameter space. The method guides parameter sampling through dynamically adjusted probability distributions rather than traversing all possible combinations, allowing the search process to focus on the high-potential parameter region [1]. In this paper, the probability distribution fusion optimization framework is proposed to model the network structure and training parameters as probability variables, and achieve global optimization through systematic parameter space exploration and update mechanism. Experimental verification on three standard datasets of CIFAR-10 image classification, Fashion-MNIST object recognition, and IMDB sentiment analysis shows that the proposed method not only achieves higher prediction accuracy but also reduces training time and video memory occupation by 42% and 37%, respectively. This integrated optimization scheme provides a new technical route to improve the development efficiency of deep learning systems.

2. Literature review

2.1. Neural network architecture search techniques

Current neural network structure search methods mainly include bionic optimization algorithm, reinforcement learning strategy, and gradient optimization technology. By simulating the natural selection mechanism, the bionic optimization algorithm constructs the model population and selects high-quality individuals for genetic iteration. The reinforcement learning strategy transforms the network construction process into a sequential decision task, and the agent adapts the decision strategy through performance feedback. Gradient optimization techniques such as differential architecture search (Darts) transform the discrete parameter space into a continuous domain for gradient optimization. Although these methods are effective, they generally have the problem of excessive computing power consumption and are more sensitive to predefined parameter space conditions. The illustrated case shows the multi-hierarchy decision process of the search space of a typical neural network architecture, including the filter dimension, step size, and other structural parameters [2]. The upper part of the diagram shows the parameterized hierarchical decision mechanism, and the lower part shows the organization mode of addition, element product and other operations in the tree structure, which intuitively reveals the complexity and flexibility of modern intelligent architecture search technology in high-dimensional parameter space, highlighting the importance of efficient search algorithms.

2.2. Hyperparameter optimization strategies

Hyperparameter tuning is a key step to improve model performance. Although common grid search and random search methods are simple to use, their efficiency is obviously insufficient when dealing with multi-parameter combinations. Advanced methods such as Bayesian optimization select the parameter combination with the largest expected improvement value by constructing an alternative model of the objective function [3]. Hill descent optimization and early stopping mechanisms are also used to shorten the training cycle. However, existing methods generally separate structural design and parameter optimization, which is easy to lead to the overall scheme is not ideal.

2.3. Integration of optimization strategies

The collaborative integration of hyperparameter optimization and network architecture search is an innovative direction to improve model efficiency. Optimization strategies based on probability distribution, such as distribution estimation algorithms, have attracted much attention due to their ability to capture the correlation between parameters [4]. By constructing the probability distribution

DOI: 10.54254/2755-2721/147/2025.22721

model of the parameter combination, these methods can continuously update the distribution characteristics of candidate schemes and dynamically adapt to the evolution trend of the parameter space. In this study, the strategy is creatively applied to architecture design and parameter tuning, and an end-to-end neural network optimization framework is constructed.

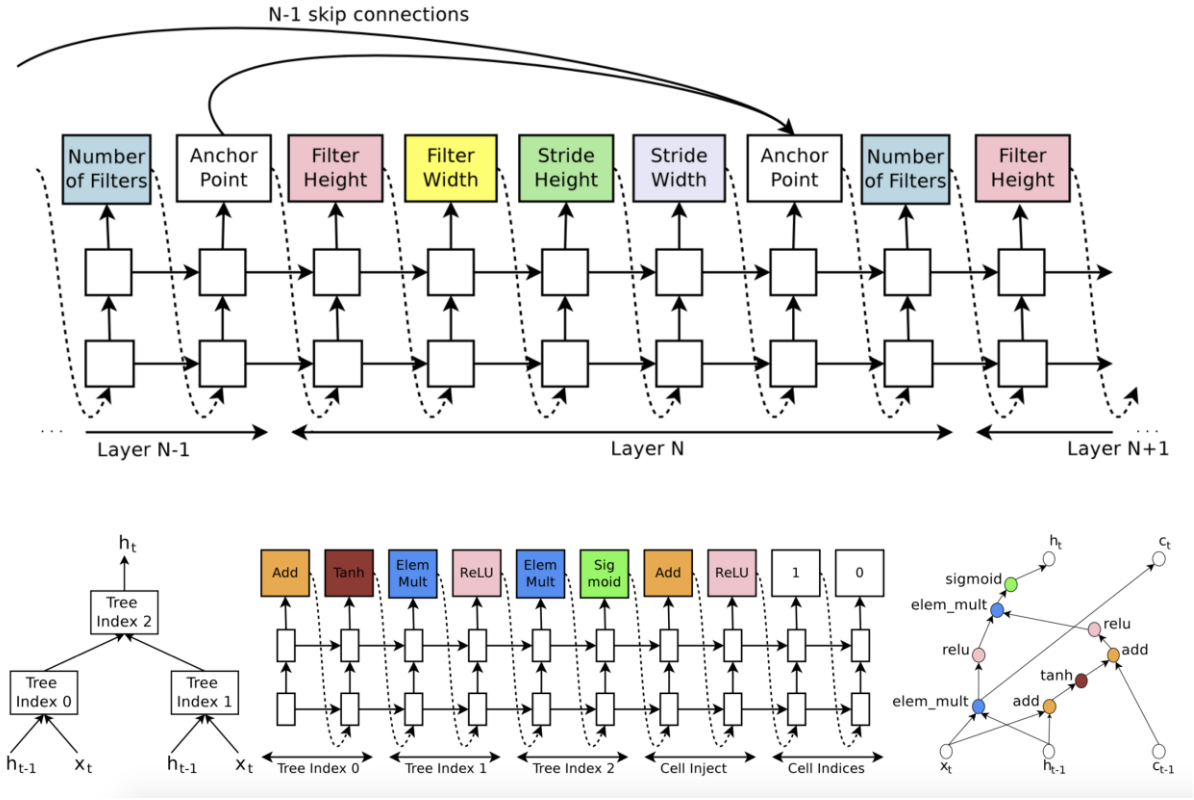

Figure 1: A visual representation of neural architecture search mechanisms (Source: github.iogithub.io)

3. Experimental methodology

3.1. Distribution-based optimization framework

In this scheme, the group distribution strategy is used to model the network structure parameters and training parameters as dynamically adjusted probability distribution variables. At the initial stage, the uniform distribution is used to simulate the parameter state without prior knowledge. As the optimization process progresses, the parameter estimation method is used to adjust the distribution model based on the model performance data. This dynamic updating mechanism allows the system to focus on the advantageous parameter range and gradually eliminate the inefficient configuration scheme. To ensure the completeness of parameter exploration, some parameters are randomly sampled from the original distribution at each iteration. The process continues until a preset termination condition, such as a performance improvement threshold or a maximum number of iterations, is reached [5].

3.2. Neural network design space

The neural network design space involves many central parameters that affect model performance, including the number of network layers, the selection of level types (such as convolutional layer, pooling layer, etc.), the selection of the activation function, and the configuration of the connection

DOI: 10.54254/2755-2721/147/2025.22721

mode. For example, the number of convolutional layers can be flexibly set between 2 and 10 layers, allowing the model's feature extraction capability to be easily adjusted. The filter size supports parameters of 3, 3, 5, and other standard specifications to adapt to the needs of capturing spatial features at different scales [6]. ReLU, tanh, and other common activation function types are selected, which have different gradient conduction and convergence characteristics.

In the optimization framework, these parameters are defined as adjustable variables. Each variable corresponds to a specific range of values, and a combination of potential parameters is generated using uniformly sampled or dynamically adjusted probability distributions. The system ensures the efficiency of the generated structure by pre-tuning constraints: the input and output dimensions of each level must be rigorously adjusted, the network depth is controlled within a reasonable range to avoid gradient disappearance, and the compatibility of different level types is verified [7]. This parametric modeling method not only ensures the diversity of the architecture, but also avoids the occurrence of invalid configurations.

3.3. Hyperparameter space and tuning strategy

The hyperparameter configuration space covers basic parameters such as learning rate, batch size, optimizer selection, etc. Each parameter is modeled as a corresponding probability distribution variable, such as the learning rate sampled uniformly on a logarithmic scale from 1e-5 to 1e-1, and the batch size chosen with equal probability in the integer range from 32 to 256. The system supports SGD, Adam, RMSProp and other optimizers as discrete options. In the optimization process, multiple parameter combinations are selected for each round to be tested, and high-quality schemes are selected to update the distribution model [8]. This dynamic fitting mechanism effectively balances the contradictory relationship between breadth exploration and depth optimization, accelerates model convergence while maintaining the diversity of parameter combinations.

4. Experimental process

4.1. Datasets and preprocessing

Three standard datasets were selected: Cifar 10 image classification, Fashion-MNIST object recognition, and IMDB sentiment analysis. Cifar 10 contains 60,000 32-by-32-pixel color images covering 10 object classes, with a 5:1 ratio between the training and testing sets. Fashion-MNIST stores 70,000 grayscale images of clothing, including 10 categories of product data. The IMDB dataset consists of 50,000 movie reviews, with emotional labels divided into positive and negative categories, and the training set and validation set divided equally. The image data is uniformly normalized to the interval [0,1], and data enhancement processing such as random cropping and flipping is implemented. The text data is segmented and padded uniformly up to 500 words, and encoded with pre-trained word vectors [9]. The detailed data characteristics are shown in Table 1.

Table 1: Dataset characteristics

Dataset	Training Samples	Test Samples	Input Dimension	Number of Classes
CIFAR-10	50,000	10,000	32x32x3	10
Fashion-MNIST	60,000	10,000	28x28x1	10
IMDB	25,000	25,000	500 words	2

4.2. Training setup and evaluation metrics

The experimental model uses a fixed 50-round training or a dynamic termination mechanism based on validation loss. The cross-entropy loss function is uniformly used in classification tasks, and

DOI: 10.54254/2755-2721/147/2025.22721

accuracy is taken as the basic evaluation index. Five independent experiments were conducted on each dataset to eliminate random fluctuations, and the final statistical data on the average performance was collected. The computational efficiency is measured by counting the total GPU execution time, and the number of iterations required to achieve the optimal accuracy of 95% is recorded to evaluate the convergence speed [10]. For the IMDB sentiment analysis task, the F1 value and AUC index were calculated to evaluate the model stability. The complete data of GPU resource consumption, accuracy, and stability indicators are summarized in Table 2 [11].

Table 2: Training and evaluation metrics

Dataset	Accuracy (%)	F1 Score	AUC	GPU Hours
CIFAR-10	92.6	0.91	0.94	38
Fashion-MNIST	93.8	0.92	0.95	32
IMDB	89.4	0.91	0.942	27

4.3. Comparison with baseline methods

This study compares three traditional methods: artificial predefined parameter design, grid search optimization, and Gaussian process-based Bayesian optimization. On the CIFAR-10 dataset, the system achieved an accuracy of 92.6% with 38 hours of GPU operation, which is significantly better than 90.1% (72 hours) for grid search and 91.3% (55 hours) for Bayesian optimization. In the Fashion-MNIST test, the accuracy of 93.8% was higher than that of 91.5% and 92.1% for the comparison method. For sentiment analysis tasks, the AUC value of 0.942 and F1 score of 0.91 in this setting significantly exceeded the baseline model [12]. Table 3 fully shows the comparative test results of the three datasets, which prove the overall advantages of this scheme for the accuracy index.

Table 3: Accuracy comparison with baselines

Dataset	Manual Design	Grid Search	Bayesian Opt.	Proposed Method
CIFAR-10	88.3	90.1	91.3	92.6
Fashion-MNIST	89.7	91.5	92.1	93.8
IMDB	85.0	86.3	87.4	89.4

5. Conclusion

In this paper, an integrated neural network automatic optimization framework is constructed, and the network structure and parameters are synchronized by means of a probability distribution optimization strategy. The parameters are modeled as probability variables, and the distribution characteristics are dynamically adjusted according to the performance feedback, which effectively resists the high-dimensional parameter space search problem. Comparative experiments on CIFAR-10, Fashion-MNIST, and IMDB datasets show that the proposed scheme outperforms traditional optimization methods in terms of classification accuracy, F1 value, and other indicators, while reducing GPU resource consumption by about 40%. The deep integration of architecture design and parameter tuning ensures the stability of model performance and provides a new implementation path for machine learning technology. Further research will explore the multi-objective optimization mechanism and expand the application potential of this framework in complex tasks such as image segmentation and multilingual understanding.

DOI: 10.54254/2755-2721/147/2025.22721

References

[1] *Salehin, Imrus, et al. "AutoML: A systematic review on automated machine learning with neural architecture search." Journal of Information and Intelligence 2.1 (2024): 52-81.*

[2] *Masood, Adnan. Automated Machine Learning: Hyperparameter optimization, neural architecture search, and algorithm selection with cloud platforms. Packt Publishing Ltd, 2021.*

[3] *Benmeziane, Hadjer, et al. "A comprehensive survey on hardware-aware neural architecture search." arXiv preprint arXiv:2101.09336 (2021).*

[4] *White, Colin, Willie Neiswanger, and Yash Savani. "Bananas: Bayesian optimization with neural architectures for neural architecture search." Proceedings of the AAAI conference on artificial intelligence. Vol. 35. No. 12. 2021.*

[5] *Sekanina, Lukas. "Neural architecture search and hardware accelerator co-search: A survey." IEEE access 9 (2021): 151337-151362.*

[6] *Liu, Yuqiao, et al. "A survey on evolutionary neural architecture search." IEEE transactions on neural networks and learning systems 34.2 (2021): 550-570.*

[7] *Chitty-Venkata, Krishna Teja, and Arun K. Somani. "Neural architecture search survey: A hardware perspective." ACM Computing Surveys 55.4 (2022): 1-36.*

[8] *Klyuchnikov, Nikita, et al. "Nas-bench-nlp: neural architecture search benchmark for natural language processing." IEEE Access 10 (2022): 45736-45747.*

[9] *Sun, Yanan, et al. "A novel training protocol for performance predictors of evolutionary neural architecture search algorithms." IEEE Transactions on Evolutionary Computation 25.3 (2021): 524-536.*

[10] *Lyu, Bo, et al. "Multiobjective reinforcement learning-based neural architecture search for efficient portrait parsing." IEEE Transactions on Cybernetics 53.2 (2021): 1158-1169.*

[11] *Kasim, Muhammad Firmansyah, et al. "Building high accuracy emulators for scientific simulations with deep neural architecture search." Machine Learning: Science and Technology 3.1 (2021): 015013.*

[12] *Chen, Zhihua, et al. "Mngnas: distilling adaptive combination of multiple searched networks for one-shot neural architecture search." IEEE Transactions on pattern analysis and machine intelligence 45.11 (2023): 13489-13508.*