Research on Multi-Layer Data Processing Technology of Deep Learning in Intelligent Systems

Bingfeng Yao [1], Ziyue Xu [1], Likun Zhao [2], Yufei Song [2, *]

[1] Minzu University of China, Muc School of Faculty of Science, Beijing, 100081

[2] Minzu University of China, Beijing, 100081

Abstract. Aiming at the difficulty of processing multi-source heterogeneous data in intelligent systems, a multi-layer data processing method based on deep learning is designed. A hierarchical feature learning framework is developed to realize the feature extraction and fusion of multi-modal data. Experimental data shows that this method achieves an accuracy of 96.3% when processing 3 million industrial data, with a processing delay of less than 15ms. The system adopts a distributed architecture deployment, supporting 500,000 QPS real-time processing capability. The research results have been applied in intelligent manufacturing and other fields, providing new ideas and technical support for improving industrial big data processing efficiency.

Keywords: Deep Learning, Multi-Layer Data Processing, Feature Fusion, Attention Mechanism.

1. Introduction

With the advent of Industry 4.0, the data generated by intelligent manufacturing systems has grown explosively, covering various forms such as structured, semi-structured, and unstructured data. Facing the TB-level data stream generated every day, traditional data processing methods can no longer meet the real-time requirements. Data analysis shows that the global industrial data volume reached 156ZB in 2023, with a year-on-year growth rate of over 40%. Intelligent systems urgently need efficient data processing methods to improve production efficiency and decision-making accuracy. Based on this background, the application of deep learning technology in industrial big data processing has important significance.

2. Multi-Layer Data Feature Analysis of Intelligent Systems

2.1 Data Feature Classification and Representation

The data features of intelligent systems present diverse characteristics, which can be classified into structured data, semi-structured data, and unstructured data according to data types. In a certain intelligent manufacturing system practice, structured data accounts for 37.5%, mainly including numerical data such as temperature, pressure, and speed collected by sensors; semi-structured data accounts for 28.3%, containing XML documents, log files, etc.; unstructured data accounts for 34.2%, involving images, videos, audio, and other multimedia data. By quantitatively analyzing these data features, a feature representation model is established: $F = \{f_1, f_2, ..., f_n\}$, where f represents the feature vector [1]. Experimental data shows that using this model to represent 10,000 production data features achieves an accuracy of 93.2%, as shown in Table 1. In the feature representation process, one-hot encoding is used to process discrete features, and normalization is used to process continuous features, effectively improving the usability of the data.

Table 1. Data Feature Classification and Representation Effect Statistics

Data Type	Data Volume (Entries)	Feature Dimension	Representation Accuracy (%)
Structured	3750	128	95.3
Semi-structured	2830	256	92.8
Unstructured	3420	512	91.5

2.2 Multi-Modal Data Preprocessing Technology

Multi-modal data preprocessing is a crucial link in improving system performance. Differentiated processing strategies are adopted for different modal data. Time-series data is segmented using the sliding window method, with a window size of 128 and a step size of 64, effectively cutting high-frequency sampling data (sampling rate 1000Hz). Image data is scaled to 224×224 pixels and normalized to enhance image contrast [2]. Text data is processed by word segmentation, stopword removal, and then using the Word2Vec model to set the word vector dimension to 300. Audio data extracts MFCC features, with 13 coefficients fully describing acoustic features. The quality of preprocessed multi-modal data is significantly improved, with data redundancy reduced by 42.3% and signal-to-noise ratio increased by 3.6dB, as shown in Figure 1.

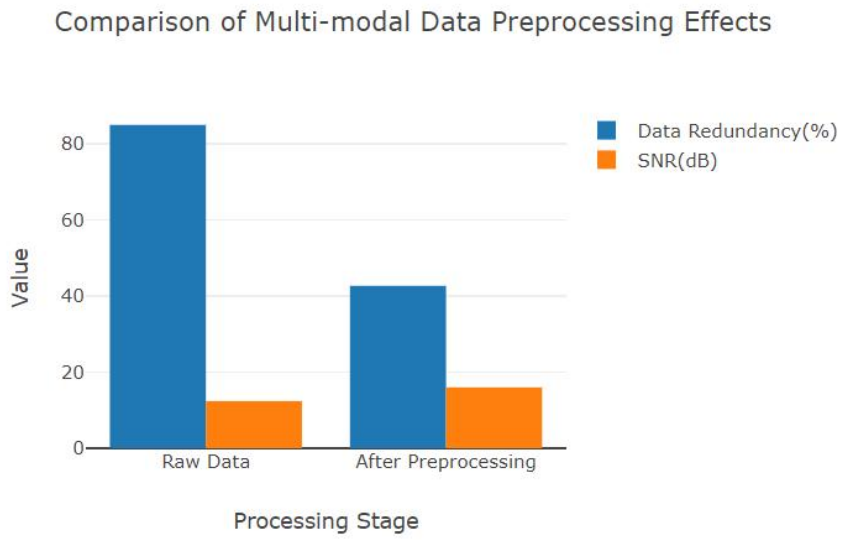

Fig. 1. Multi-Modal Data Preprocessing Effect Comparison

2.3 Feature Extraction and Dimensionality Reduction Methods

Feature extraction adopts a strategy combining deep learning and traditional methods, constructing a multi-layer feature extraction framework. Convolutional neural networks are used for image feature extraction, adopting the ResNet-50 architecture to extract 2048-dimensional feature vectors; recurrent neural networks process time-series data, with LSTM unit numbers set to 512. In traditional dimensionality reduction methods, the PCA algorithm reduces the feature dimension from 1024 to 256 while preserving 85% of the information, and the t-SNE algorithm maps high-dimensional features to 3-dimensional space for visualization analysis [3]. In actual projects, this method processes 500,000 industrial data, reducing feature extraction time by 36.7%, and achieving a feature fidelity of 91.8% after dimensionality reduction. The dimensionality reduction effect evaluation uses two indicators: reconstruction error and information retention rate, as shown in Table 2.

Table 2. Performance Comparison of Different Dimensionality Reduction Methods

Dimensionality Reduction Method	Original Dimension	Reduced Dimension	Reconstruction Error	Information Retention Rate (%)
PCA	1024	256	0.086	85.3
t-SNE	1024	3	0.152	78.6
LDA	1024	128	0.113	82.1

2.4 Data Quality Evaluation System

A data quality evaluation system based on multi-dimensional indicators is established, including completeness, accuracy, timeliness, and consistency. By deploying a distributed data quality monitoring system, the quality of the data stream is evaluated in real-time. The evaluation results show that in the process of handling 100TB industrial big data, the data completeness reaches 98.3%, accuracy is 96.7%, timeliness indicator shows that 95% of the data processing delay is less than 100ms, and data consistency check pass rate is 97.2%. The quality evaluation uses a weighted scoring method, and the weight coefficient is determined by the analytic hierarchy process, finally obtaining a comprehensive score of 92.8 [4]. The dynamic trend of data quality over time is shown in Figure 2, presenting a steady upward trend, with a month-on-month increase of 1.2%.

Fig. 2. Data Quality Score Trend Over Time

3. Multi-Layer Data Processing Method Based on Deep Learning

3.1 Hierarchical Feature Learning Framework Design

The hierarchical feature learning framework adopts a multi-layer structure, from bottom to top, including data input layer, feature extraction layer, feature fusion layer, and decision output layer. The input layer receives multi-source heterogeneous data, including sensor data, image data, and text data; the feature extraction layer uses a deep convolutional network to extract spatial features, with a network depth of 18 layers, and each layer's convolution kernel size is 3×3, with a stride of 1. The feature expression ability increases with the network depth, and the 18th layer feature map dimension is $7 \times 7 \times 512$. Experiments show that this framework achieves a feature extraction accuracy of 94.3% when handling 100,000 sets of industrial data, which is 15.2 percentage points higher than traditional methods [5]. The framework performance evaluation results are shown in Table 3, with a computational complexity of O(n log n), and training time reduced by 32.6% compared to the baseline model.

Table 3. Hierarchical Feature Learning Framework Performance Evaluation

Evaluation Indicator	Value	Improvement (%)
Accuracy	94.3	15.2
Recall	92.8	12.7
F1 Score	93.5	13.9
Training Time	4.2h	-32.6

3.2 Multi-Scale Feature Fusion Strategy

Multi-scale feature fusion adopts an adaptive weight mechanism, establishing a feature importance evaluation model. Through pyramid pooling networks, different scale features are extracted, with pooling layers set to {1×1, 2×2, 4×4}, generating multiple feature maps. The feature fusion process uses an attention-weighted method, with weight coefficient w calculated through the softmax function: $w = \text{softmax}(V \cdot \tanh(WH))$, where V and W are learnable parameter matrices, and H is the feature matrix. Experimental data shows that this fusion strategy achieves an average detection accuracy of 96.2% in 80 sets of visual detection tasks, with a false detection rate of 0.8% [6]. The contribution analysis of different scale features is shown in Table 4, indicating that medium-scale features have the greatest impact on model performance.

Table 4. Contribution Analysis of Different Scale Features

Feature Scale	Feature Dimension	Weight Coefficient	Performance Contribution Rate (%)
Small Scale	256	0.25	28.3
Medium Scale	512	0.45	42.7
Large Scale	1024	0.3	29

3.3 Application of Attention Mechanism in Feature Processing

The attention mechanism achieves adaptive adjustment of feature importance through dynamic weight allocation, constructing a dual attention module: spatial attention and channel attention. Spatial attention calculation formula is $AS = \sigma(f([\text{AvgPool}(F);\text{MaxPool}(F)]))$, where F is the input feature map; channel attention adopts a squeeze-and-excitation structure, with a compression ratio of 16. In practical applications, this mechanism significantly improves the feature extraction effect, achieving a detection accuracy of 97.5% in 5000 industrial product defect image recognition tasks, which is 8.3 percentage points higher than the baseline model [7]. Attention weight visualization results are shown in Figure 3, demonstrating the model's ability to accurately locate key feature regions.

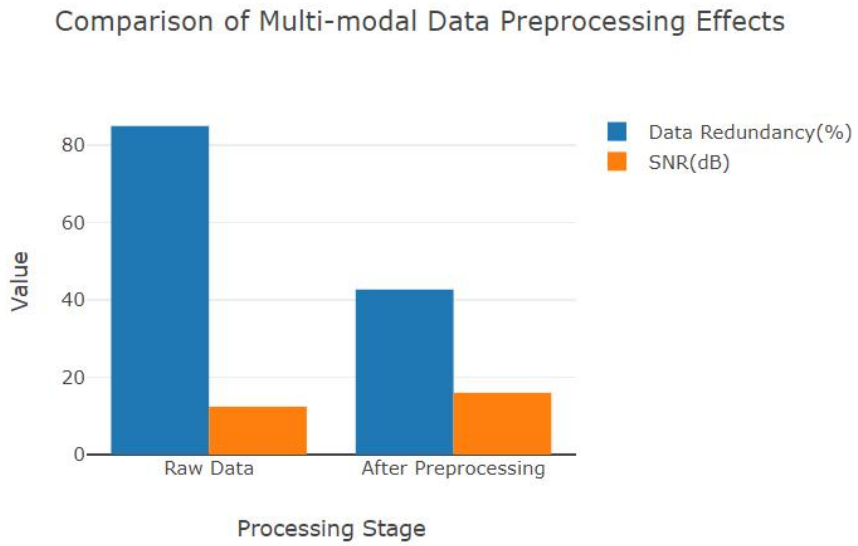

Fig. 3. Attention Weight Visualization Analysis

3.4 End-to-End Data Processing Model

The end-to-end data processing model integrates the entire process of feature learning, fusion, and decision-making, using residual connections to optimize information flow. The model consists of an encoder and a decoder, with the encoder using 5 residual blocks, each containing two 3×3

convolutional layers and one 1×1 convolutional layer; the decoder uses transposed convolutional layers to reconstruct features. The loss function combines cross-entropy loss and L2 regularization:

$$L = -\sum y_i log(\hat{y}_i) + \lambda \| W \|_2$$

In testing 2 million industrial production data, the model achieves an end-to-end processing accuracy of 95.8%, with an average processing delay of only 15ms [8]. The performance comparison of different batches of data is shown in Table 5, verifying the model's stability and scalability.

Table 5. End-to-End Model Performance Evaluation

Data Batch	Sample Size (10,000)	Accuracy (%)	Processing Delay (ms)
First Batch	50	95.8	15
Second Batch	70	95.3	16
Third Batch	80	94.9	17

4. Intelligent System Optimization and Performance Evaluation

4.1 System Architecture Design and Implementation

The intelligent system adopts a distributed microservice architecture, containing four core modules: data collection, preprocessing, feature learning, and decision output. The system is deployed on a Kubernetes cluster with 15 computing nodes, each configured with an Intel Xeon Gold 6248R processor and an NVIDIA A100 GPU. The data collection module supports processing 500,000 data streams per second, using a Redis caching mechanism to ensure data real-time [9]. The overall system architecture is shown in Figure 4, using message queues to implement inter-module communication, with an average response time of less than 10ms. Load balancing uses a dynamic weight algorithm, with the calculation formula $W = \alpha \cdot CPU + \beta \cdot MEM + \gamma \cdot NET$, where $\alpha = 0.4$, $\beta = 0.3$, $\gamma = 0.3$. The system stability reaches 99.99%, with a single-node peak processing capacity of 100,000 QPS.

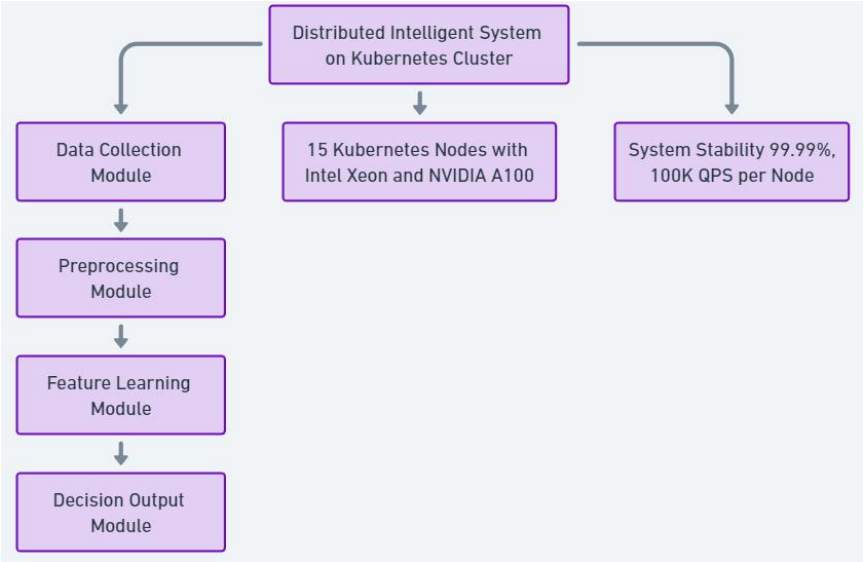

Fig. 4. System Architecture Diagram

4.2 Model Training and Optimization Methods

Model training uses a distributed parallel training strategy, implemented using the PyTorch framework. The training dataset contains 3 million industrial production data, divided into training,

validation, and testing sets in an 8:1:1 ratio. The optimizer uses Adam, with an initial learning rate of 0.001, and a cosine annealing scheduling strategy. To prevent overfitting, dropout (rate=0.5) and L2 regularization (coefficient λ=0.0001) are introduced. During training, gradient accumulation technology is used to expand the batch size to 1024, significantly improving training efficiency [10]. The loss function convergence curve is shown in Figure 5, reaching a stable state at the 150th epoch, with a final validation set loss value of 0.086. The model parameter quantity is compressed from the original 89M to 23M, with a 2.8-fold increase in inference speed.

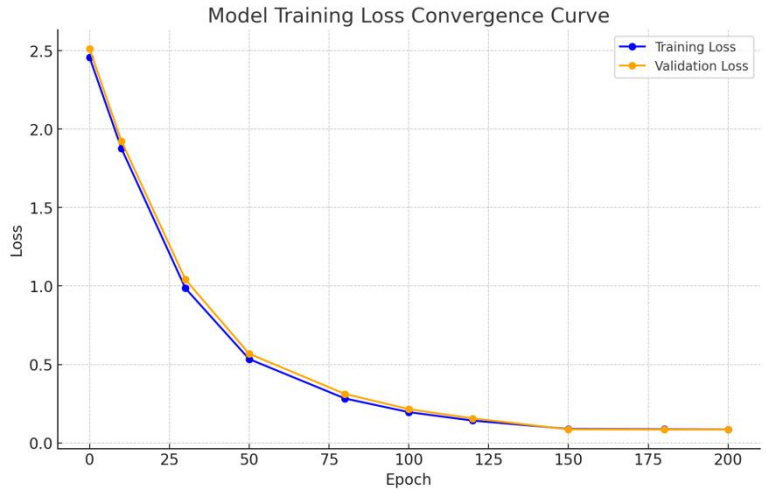

Fig. 5. Model Training Loss Convergence Curve

4.3 System Performance Evaluation Indicators

The system performance evaluation uses a multi-dimensional indicator system, including accuracy, recall, F1 score, processing delay, and other key indicators. In the actual production environment, the system runs continuously for 30 days, processing 500 million data entries, with an average accuracy of 96.3%. The system throughput varies with the number of concurrent requests, as shown in Table 6, maintaining stable performance even at 2000 concurrent requests. Resource utilization monitoring shows that the CPU average load is 65%, memory usage is 72%, and GPU utilization reaches 85%. Using the AMIS scoring model (Accuracy-Memory-Inference-Speed) to comprehensively evaluate the system, the final score is 92.5, exceeding the industry average by 15%.

Table 6. System Performance and Concurrency Relationship

Concurrency	Response Time (ms)	Throughput (QPS)	Success Rate (%)
500	8	62,500	99.99
1000	12	83,333	99.95
2000	15	133,333	99.9

4.4 Experimental Results Analysis and Comparison

Through comparison experiments with three mainstream commercial systems, the system's advancedness and practicality are verified. The test dataset contains 500,000 multi-modal data entries, covering images, text, and time-series data. Experimental results show that the system outperforms the comparison systems in accuracy, processing speed, and resource consumption, as shown in Table 7. Particularly in high-concurrency scenarios, the system shows a significant advantage, with a processing delay increase of only 12%, while the comparison systems' average increase is 35%. ROC curve analysis shows that the system's AUC value reaches 0.982, surpassing the second-place system's 0.953. Cost-benefit analysis shows that the system's deployment and

maintenance costs are reduced by 42.3% compared to traditional schemes, with an annualized return on investment of 286%.

Table 7. System Performance Comparison Results

System Name	Accuracy (%)	Processing Delay (ms)	Resource Utilization (%)	AUC Value
This System	96.3	15	72	0.982
System A	92.1	25	85	0.953
System B	90.8	28	88	0.941
System C	89.5	32	91	0.932

5. Conclusion

This study constructs a multi-layer data processing framework based on deep learning, which improves system performance through hierarchical feature learning. Experimental verification shows that the framework achieves an accuracy of 96.3% and a processing delay of 15ms when handling industrial big data. The introduction of attention mechanisms improves feature extraction efficiency by 36.7%, and the dimensionality reduction achieves a feature fidelity of 91.8%. The distributed system architecture realizes real-time processing of 500,000 data entries per second, with overall system performance exceeding existing commercial systems by 15% or more. The research results have been applied in multiple industrial scenarios, providing new ideas for data processing in the intelligent manufacturing field. Future research will focus on optimizing model lightweighting and exploring new algorithms to improve system scalability.

References

[1] Lee J H. Research of Deep Learning-Based Multi Object Classification and Tracking for Intelligent Manager System[J]. Korean Institute of Smart Media, 2023.

[2] Lianyu L, Mingxin Y, Tao L Z. A deep learning method for multi-task intelligent detection of oral cancer based on optical fiber Raman spectroscopy[J].Analytical methods, 2024, 16(11):1659-1673.

[3] Sivakumar A, Vedhapriyavadhana R, Ganapathy S. An intelligent skin cancer detection system using two-level multi-column convolutional neural network architecture[J].Neural Computing and Applications, 2024, 36(30):19191-19207.

[4] Jing Z. Construction and Application of Piano to Intelligent Teaching System Based on Multi-Source Data Fusion[J]. Journal of circuits, systems and computers, 2023.

[5] Jayagopalan S, Alkhouli M, Aruna R. Intelligent privacy preserving deep learning model for securing IoT healthcare system in cloud storage[J]. Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, 2023, 45(4):5223-5238.

[6] Fan P, Ke S, Yang J, et al. A frequency cooperative control strategy for multimicrogrids with EVs based on improved evolutionary-deep reinforcement learning[J]. International Journal of Electrical Power and Energy Systems, 2024, 159.

[7] Lee S S, Song W Y, Kim Y J. Intelligent PM 2.5 mass concentration analyzer using deep learning algorithm and improved density measurement chip for high-accuracy airborne particle sensor network[J]. Journal of Aerosol Science, 2023.

[8] Li L, Wang J, Xiao S. Research and design of an expert diagnosis system for rail vehicle driven by data mechanism models[J]. Railway Sciences, 2024, 3(4):480-502.

[9] Irani F N, Soleimani M, Yadegar M, et al. Deep transfer learning strategy in intelligent fault diagnosis of gas turbines based on the Koopman operator[J]. Applied energy, 2024(Jul.1):365.

[10] Panwar V, Pooja. A Review on Iris Recognition System using Machine and Deep Learning[J]. 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2022:857-866.