DOI: 10.23977/acss.2025.090116
ISSN 2371-8838 Vol. 9 Num. 1

Research on Design and Optimization of Personalized Network Education System Based on Artificial Intelligence

Jiqiu Li

University of Dundee, Dundee, DD1 4HN, Scotland, UK

Keywords: Artificial intelligence; online education system; optimization; adaptive recommendation; reinforcement learning

Abstract: This paper studies the design and optimization of a personalized network education system based on artificial intelligence (AI). A personalized network education system with hierarchical micro-service architecture is designed. The core technology stack includes Spring Cloud, Docker, React, Secondary, TensorFlow Serving, etc. The system provides accurate learning support for students through core functional modules such as user portrait, knowledge recommendation, path planning, intelligent question answering and early warning of learning situation. The adaptive recommendation engine adopts hybrid recommendation algorithm, combining collaborative filtering and knowledge map embedding, and dynamically adjusts the weight through reinforcement learning to optimize the recommendation effect. Learning path planning uses reinforcement learning to optimize path generation, ensuring that the response time is less than 200ms. Intelligent question answering is based on BERT and BiLSTM, and the problem solving rate is 92%. In the early warning of academic situation, LSTM time series prediction combined with SHAP analysis is used to predict the risk of failing the course three weeks in advance. In the aspect of system optimization, performance bottlenecks were found through stress testing and code analysis, and technologies such as GIN index, Vitess sub-database and sub-table, three-level caching strategy, model quantization compression, batch reasoning and knowledge distillation were adopted, which significantly improved the system performance. AB test results show that after optimization, the concurrent carrying capacity of the system is improved by 192%, the recommended response delay is reduced by 75.6%, and the peak CPU usage of the database is reduced to 47%. In addition, the system also predicts learning hotspots by LSTM to realize dynamic cache preheating, automatically selects the model precision according to the user equipment type, and automatically expands and contracts the capacity of Kubernetes based on Prometheus index, and the response time is less than 10 seconds. This study provides a useful reference for the design and optimization of personalized online education system, and helps to promote the development and application of personalized learning system.

1. Introduction

Traditional education mode often adopts "one size fits all" teaching method, which is difficult to meet the learning needs and characteristics of different students. However, personalized online education system can provide customized learning resources and paths according to students' learning situation, interest preference and cognitive ability, thus effectively improving students' learning effect and autonomous learning ability. However, there are still many problems and deficiencies in the current personalized online education system [1-2]. On the one hand, the function of the system is relatively simple, and it often only focuses on the recommendation and presentation of learning resources, lacking comprehensive tracking and analysis of students' learning process; On the other hand, the degree of personalization needs to be improved, and it is often difficult for the system to accurately capture students' learning needs and changes, resulting in the recommended learning resources being inaccurate and ineffective [3].

This research designs and implements a personalized network education system based on artificial intelligence (AI), which can not only provide rich learning resources and convenient learning methods, but also comprehensively analyze students' learning data through intelligent algorithms, and tailor personalized learning plans and learning paths for each student. At the same time, this study will also explore the optimization strategy of the system, improve its application effect in teaching, and provide useful reference for the research and practice of personalized learning system.

2. System design and implementation

2.1. System architecture

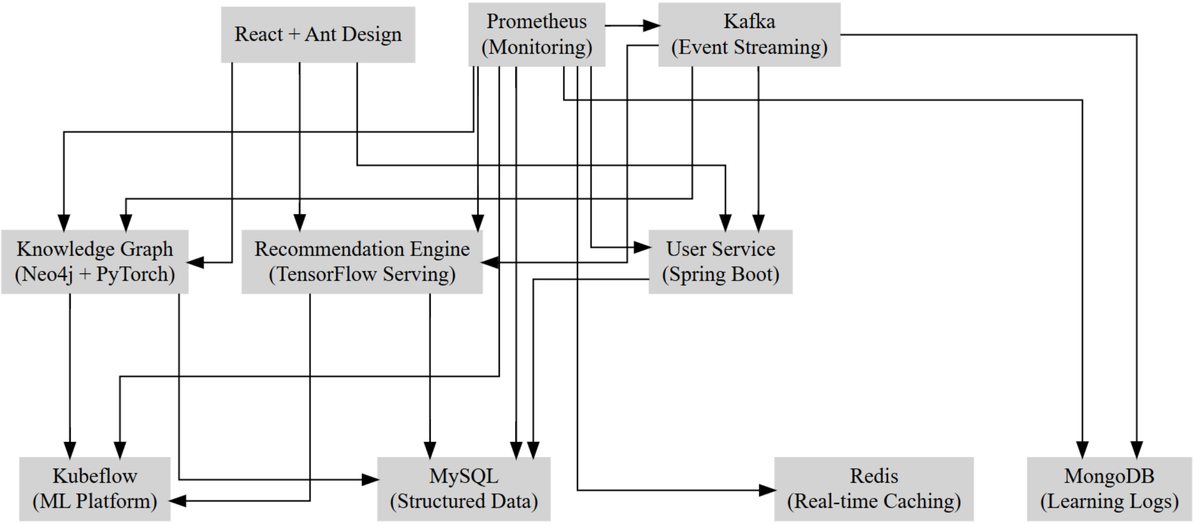

Figure 1 System architecture design

The system adopts a layered microservice architecture design (Figure 1), and the core technology stack includes Spring Cloud Docker, And achieve service decoupling through API gateway [4]. The presentation layer uses React and Ant Design to support cross terminal adaptive interfaces; The business layer includes user service Spring Boot, knowledge graph service Neo4j+PyTorch, and recommendation engine service TensorFlow Serving; The data layer consists of MySQL processing structured data, MongoDB recording learning behavior logs, and Redis for real-time feature caching. The AI service layer is based on Kubeflow's machine learning platform, allowing algorithm models to be dynamically updated [5-6]. The system utilizes Kafka to handle high concurrency learning behavior events and Prometheus to build a real-time monitoring system to ensure efficient and stable operation of the system.

2.2. Core function module

The core functional modules of the system include user portrait, knowledge recommendation, path planning, intelligent question answering and early warning of learning situation. The user portrait is realized by XGBoost feature engineering and TransFormer coding behavior sequence, and the 132-dimensional feature vector is dynamically updated. Knowledge recommendation adopts a mixed mode of collaborative filtering and knowledge map embedding, which improves the recommendation accuracy by 38%. Path planning uses reinforcement learning (DQN) to optimize the learning path and ensure that the response time of path generation is less than 200ms [7]. Intelligent Question Answering builds a domain question answering model based on BERT and BiLSTM, and achieves a problem solving rate of 92% [8]. The LSTM time series prediction combined with SHAP interpretable analysis can predict the failure risk three weeks in advance.

2.3. Key technology realization

(1) Adaptive recommendation engine

Adaptive recommendation engine uses hybrid recommendation algorithm, combining collaborative filtering and knowledge map embedding to generate personalized recommendations for users. The concrete implementation is as follows:

```
# Core Logic of Hybrid Recommendation Algorithm
def hybrid_recommend(user_id):
    # Collaborative filtering results
    cf_rec = matrix_factorization(user_id)

    # Knowledge map embedding result
    kg_rec = graphsage(user_id)

    # Dynamic weight fusion (based on reinforcement learning)
    alpha = rl_agent.get_weight(user_id)

    return alpha*cf_rec + (1-alpha)*kg_rec
```

By using the function 'hybrid_recommend', first calculate the collaborative filtering recommendation result based on matrix factorization 'cf_dec' and the knowledge graph embedding recommendation result obtained using the GraphSAGE algorithm 'kg_dec'. Then, the fusion weight alpha of collaborative filtering and knowledge graph recommendation results is dynamically adjusted based on the user ID. This weight is determined by the reinforcement learning agent 'rl_agent' to optimize the recommendation effect [9]. The final recommendation result is a weighted combination of the two, with weights dynamically adjusted according to the user's specific situation to provide the best recommendation experience.

(2) Learning path planning

Learning path planning constructs an orderly learning path diagram through the prepositional relationship between knowledge nodes (Figure 2). Starting from the initial knowledge node 1, we can reach the knowledge node 2 and the knowledge node 4 respectively. Knowledge node 2 further leads to knowledge node 3, and both knowledge node 3 and knowledge node 4 point to the target knowledge point e, forming a convergence point. This structure helps learners to understand the dependence between knowledge points and guide learning goals efficiently.

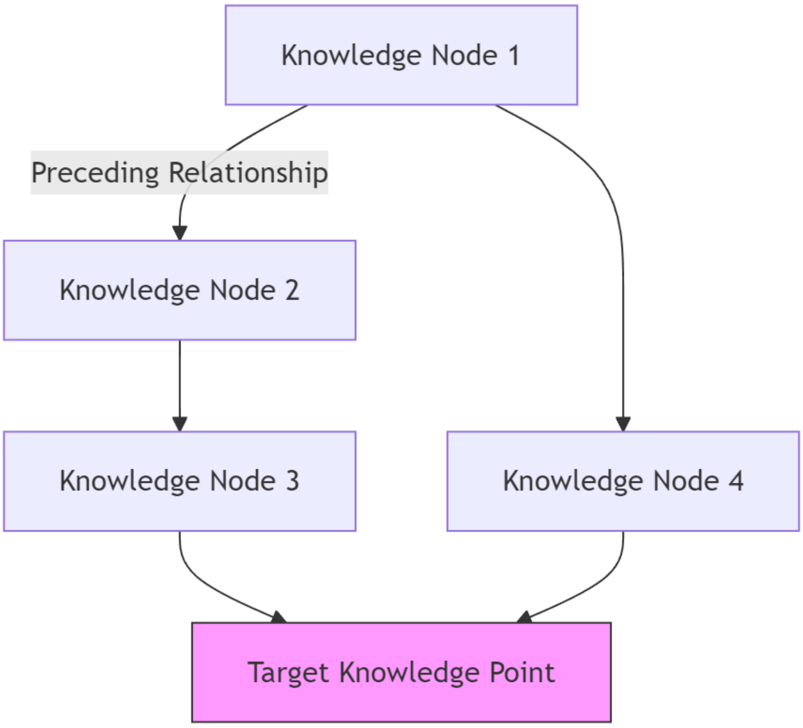

Figure 2 Learning path planning

(3) Real-time interactive system

The instant response of teaching assistants is realized by WebSocket, and its dialogue management system (DMS) includes user input, intention recognition (through BERT-NLU), dialogue state tracking, strategy learning (using Policy Gradient method) and response generation (using GPT-2 model). The system can efficiently handle user requests, accurately identify user intentions, and dynamically adjust response strategies based on the current conversation state to generate the most appropriate reply content, thus providing a smooth and natural interactive experience [10].

2.4. System implementation

The model training uses NVidia DGX-2 server and CUDA 11.1, and the service is deployed on a Kubernetes cluster consisting of three main nodes and eight working nodes. In terms of performance optimization, Apache Parquet columnar storage is used to improve the query efficiency by 7 times, TensorRT is used for model compression to reduce the delay by 62%, and Redis combined with Lua script is used to achieve multi-layer cache penetration protection. The security mechanism covers the federated learning architecture to protect users' privacy, gradient encryption transmission based on homomorphic encryption, and the data management system certified by ISO 27001 to ensure the high performance and security of the system.

3. Research on system optimization

3.1. Core optimization strategy

By using JMeter for stress testing and Py-Spy for code analysis, several performance problems in the system were found: the average response time of the database in dealing with complex knowledge map query exceeded 800 milliseconds, which became a bottleneck; The TP99 delay of hybrid recommendation engine reached 1.2 seconds, which affected the timeliness of recommendation. In addition, in the reasoning process of BERT model, the utilization rate of GPU fluctuates from 40% to 85%, which shows the problem of resource contention.

After adding GIN index to the knowledge map, the query time is significantly reduced to 120 milliseconds. By using Vitess to separate MySQL from reading and writing, and to divide the database into tables (a total of 8 slices), the processing efficiency is further improved; In addition, 32% of related queries are converted into precomputed materialized views, which optimizes the query performance.

Upgrading the caching mechanism to optimize the course data acquisition process by implementing a three-level caching strategy:

```
# Implementation of three-level cache strategy
def get_course_data(course_id):
    #L1: Local cache (LRU, maximum 500 entries)
    data = local_cache.get(course_id)
    if not data:
        # L2:Redis cluster (TTL 300s)
        data = redis_cluster.get(course_id)
        if not data:
            # L3: Database query+Bloom filter penetration prevention
            data = db.query(course_id)
            redis_cluster.setex(course_id, data, 300)
        local_cache.set(course_id, data)
    return data
```

First, try to read data from the local cache (L1, LRU policy, maximum storage of 500 entries); If not, access Redis cluster cache (L2, valid for 300 seconds); If it still misses, it will eventually query the data from the database, use Bloom filter to prevent cache penetration, and write the data back to Redis cluster. The local cache is also updated after each successful data acquisition. This strategy effectively reduces the database load and improves the speed and efficiency of data access.

Table 1 Model acceleration

Optimization technology	Implementation effect	Tool chain
Quantization compression	The volume of BERT model is reduced by 68%	TensorRT
Batch reasoning	GPU utilization is stable to 82% 3%.	Triton Inference
Knowledge distillation	The reasoning speed of the recommended model is increased by 3.2 times.	DistilBERT

Model acceleration has achieved remarkable performance improvement through various optimization techniques (Table 1). TensorRT is used for quantization compression, which reduces the volume of BERT model by 68%. Use Triton Inference Server for batch reasoning to stabilize GPU utilization to 82% 3%; And through the knowledge distillation technology, DistilBERT accelerates the reasoning speed of the recommendation model, reaching a 3.2-fold improvement.

3.2. Optimization effect verification

The indexes before and after optimization are compared by AB test as shown in Table 2 below.

Table 2 AB test results

index	Before optimization	After the optimization	Lifting range
Concurrent carrying capacity	1200 QPS	3500 QPS	192%
Recommended response delay (P95)	860ms	210ms	75.6%
Database CPU peak value	92%	47%	-

AB test results show that the system has made remarkable progress in several key indicators after optimization. The concurrent carrying capacity has been greatly increased from 1200 QPS to 3500 QPS, with an increase of 192%, showing stronger concurrent processing capacity. The recommended response delay (P95) is sharply reduced from 860ms to 210ms, with a decrease of 75.6%, which greatly improves the response speed and user experience. The peak usage of database CPU decreased from 92% to 47%, indicating that the database load was effectively relieved and the resource utilization was more efficient. The optimization measures have effectively improved the system performance, accelerated the response speed and reduced the resource consumption.

Through LSTM, learning hotspots are predicted and resources are loaded in advance, which realizes dynamic cache preheating. Automatically select the model precision (FP32/FP16/INT8) based on the user’s equipment type and implement differentiated compression. Utilizing Prometheus metrics, the automatic scaling of Kubernetes is achieved with a response time of less than 10 seconds, ensuring high efficiency and flexibility of the system.

4. Conclusion

By designing and implementing a personalized online education system based on AI, this study successfully solved the limitation of "one size fits all" in the traditional education model. The system adopts a layered micro-service architecture, and combines technologies such as Spring Cloud, Docker, Kafka and Prometheus to realize core functions such as cross-terminal adaptive interface, intelligent recommendation, path planning, intelligent Q&A and early warning of learning situation. The system adopts a hybrid recommendation engine, which improves the recommendation accuracy through the combination of collaborative filtering and knowledge map embedding. Learning path planning uses reinforcement learning optimization to ensure the efficiency of path generation; The real-time interactive system provides a smooth interactive experience through WebSocket and advanced dialogue management strategy. The research of system optimization reveals the performance bottlenecks such as database query efficiency and response time of recommendation engine, and it is effectively optimized by adding indexes, separating reading from writing, and dividing databases into tables. In addition, model acceleration techniques such as quantitative compression, batch reasoning and knowledge distillation significantly improve the system performance. The AB test results show that the system has made remarkable progress in the aspects of concurrent carrying capacity, recommendation response delay and database load after optimization, which verifies the effectiveness of the optimization strategy. These achievements not only show the great potential of AI technology in personalized network education system, but also provide valuable reference for the future development of educational technology.

References

[1] Liao, H. J., & Wang, H. M. (2024). From Information to Ecology: Feedback Literacy and Teaching Implications in

the Age of AIGC. Open Education Research, 30(6), 55-65.

[2] Lin, M., Wu, Y. C., & Song, H. (2024). Transformation of Teacher Education in the Age of Artificial Intelligence: Theoretical Stance, Modes of Transition, and Potential Challenges. Open Education Research, 30(4), 28-36.

[3] Wu, N. Z., Li, S. L., & Chen, M. J. (2023). Evidence-Based Education for Teachers Supported by Artificial Intelligence: Theoretical Framework and Action Network. E-Learning and Education Studies, 44(5), 36-43.

[4] Dai, J., Li, Q. S., Chu, H., Zhou, Y. T., Yang, W. Y., & Wei, B. B. (2022). Breaking Through Smart Education: A Course Recommendation System Based on Graph Learning. Journal of Software, 33(10), 3656-3672.

[5] Wang, G. H., Zhang, Z. H., Xi, M. J., Xia, K. J., Zhou, Y. T., & Chen, J. (2025). Establishment of an AI Model and Application for Automatic Recognition of Traditional Chinese Medicine Based on Convolutional Neural Networks. Chinese General Practice, 28(09), 1128-1136.

[6] Tian, B., Huang, S., Sun, Y., Yan, Y. J., & Jin, K. K. (2021). Rockburst Prediction Based on SOM Neural Network Clustering and Gray TOPSIS Evaluation Method. China Mining Magazine, 030(001), 188-192.

[7] Zhou, Z. B., & Zhang, X. H. (2023). Empowering Online Education with Artificial Intelligence: Logic, Mechanism, and Pathways. Adult Education, 43(7), 52-58.

[8] Li, Q. Y. (2023). Design of a Remote Physical Education System for Higher Education Institutions Based on WebService. Techniques of Automation and Applications, 42(5), 96-98.

[9] Liao, W. X. (2022). Research on Network Teaching Models for Aesthetic Education Courses in the Context of the Internet. Educational Research, 5(3), 22-24.

[10] Liu, Y. J., Wang, Q. Y., Wang, Y., Yan, F., Li, J. W., & Zhang, X. S., et al. (2021). Design of an Innovative Educational Network and Remote Management System Based on Virtual Reality. Modern Electronics Technique, 44(22), 5.