Research on Innovative Design of High School Biology Courses Based on Deep Learning

Jingtong Zhao*

Shanwei LinWeihua Middle School, Shanwei, Guangdong, China

*Corresponding author's e-mail: 1758040388@qq.com

Abstract. To explore the application of deep learning algorithms in the design of high school biology courses, this paper studies common models and data preprocessing methods and analyzes the impact of different parameter settings on teaching effectiveness. The results show that optimized deep learning models can significantly enhance the accuracy and interactivity of biology teaching, providing an intelligent and personalized learning experience, thereby effectively improving teaching outcomes.

Keywords: deep learning; high school biology course design; teaching effectiveness

1 Introduction

In the context of modern education, exploring the application of deep learning algorithms in high school biology courses provides a new pathway for achieving intelligent and personalized teaching. By optimizing course design, data preprocessing, and model training, the teaching effectiveness and students' learning experience are expected to be significantly enhanced, injecting new impetus into technological innovation in the education sector. The core of the research lies in using multi-layer neural networks to accurately identify and classify biological concepts and knowledge points, enhancing the adaptability and interactivity of teaching content, and promoting the overall progress of high school biology education.

2 Theoretical Basis of Deep Learning

2.1 Overview of Deep Learning Algorithms

Deep learning algorithms are based on artificial neural networks and perform feature extraction and transformation through multi-layer processing structures. They automatically adjust parameters through training with large amounts of data, optimizing recognition accuracy and processing efficiency, and are particularly adept at handling nonlinear problems. These algorithms have demonstrated powerful capabilities in complex environments such as image recognition and speech processing, driving the

V. A. Balakrishnan et al. (eds.), *Proceedings of the 2024 4th International Conference on Modern Educational and Social Sciences (ICMETSS 2024)*, Advances in Social Science, Education and Humanities Research 878,
https://doi.org/10.2991/978-2-38476-311-5_5

development of artificial intelligence technology and providing impetus for technological innovation across various industries[1].

2.2 Common Deep Learning Models

Deep learning models include convolutional neural networks, recurrent neural networks, and generative adversarial networks. CNNs are suitable for image processing and recognize features through hierarchical learning. RNNs excel in sequential data processing, such as language recognition and time series prediction. GANs consist of generation and evaluation parts and are widely used in image generation and reinforcement learning. Each of these models has its advantages, and the choice of the appropriate model depends on application requirements, as shown in Table 1.

Table 1. Common Deep Learning Models and Application Fields

Model Name	Application Fields
Convolutional Neural Networks	Image recognition, video analysis
Recurrent Neural Networks	Language processing, time series prediction
Generative Adversarial Networks	Image generation, simulation training

3 High School Biology Course Design Based on Deep Learning

3.1 Overall Architecture Design

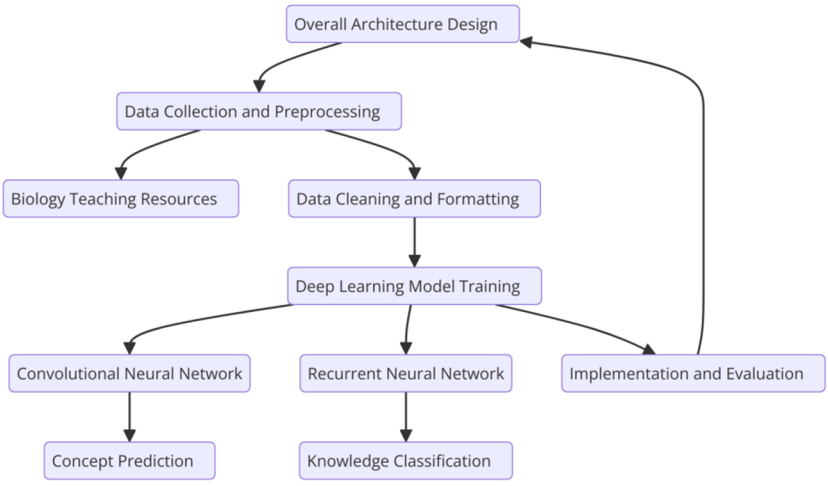

Fig. 1. Framework of High School Biology Courses Based on Deep Learning

The design of high school biology courses based on deep learning involves several key stages to ensure intelligent and personalized teaching content. The first stage is the overall architecture design, which defines the framework of the entire course and the teaching objectives, as well as how to integrate deep learning technology to optimize learning outcomes[2]. The next stage is the course data collection and preprocessing phase, where relevant biological teaching resources need to be collected and subjected to data cleaning and formatting to prepare standardized input data for subsequent model training. The third part is the construction and training of deep learning models. This stage involves selecting appropriate neural network models and training them according to the characteristics of the course to achieve accurate prediction and classification of biological concepts and knowledge points. Each part is interconnected and works together to create an efficient, dynamic, and highly interactive biology teaching environment, as shown in Figure 1.

3.2 Course Data Collection and Preprocessing

Data collection involves extracting information from various biological teaching resources such as textbooks, academic papers, and online databases. The key is to ensure that the data covers the main concepts and experimental knowledge of biology to support comprehensive learning by the model. The collected data needs to undergo complex preprocessing steps to improve the efficiency and accuracy of model training[3]. Preprocessing includes data cleaning (removing errors and duplicate data), standardization, and normalization. Data standardization can be performed using Z-score standardization, with the calculation formula as follows:

$$Z = \frac{(X - \mu)}{\sigma} \tag{1}$$

Among them, X it is the original data point, μ gentle σ Data points whose mean and standard deviation exceed a specific threshold are considered outliers and treated accordingly. The standardization process is to ensure that the model does not lean towards certain features due to differences in data size. The commonly used method is Z-score normalization, which can convert the data into a distribution with a mean of 0 and a standard deviation of 1. The formula is the same as above. Min Max normalization can also be used, with the following formula:

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{2}$$

Data conversion includes encoding from text to numerical values, such as using unique hot encoding to convert categorical variables. In addition, to process time series data, the window method can be used to segment the data, and the formula is expressed as:

$$X_t = \{x_{t-n+1}, \ldots, x_t\} \tag{3}$$

Among them, n The above steps ensure data consistency and adaptability, laying a solid foundation for the effective training of deep learning models. These advanced data preprocessing techniques can significantly enhance the performance of subsequent models and the personalized adaptability of the teaching content[4].

3.3 Deep Learning Model Construction and Training

The construction and training phase is crucial, aiming to build a neural network model that can accurately recognize and process biological teaching content. The model construction begins with selecting the appropriate network architecture. Common architectures such as convolutional neural networks (CNN) are particularly effective for processing image data, while recurrent neural networks (RNN) are suitable for handling sequential data such as text or time series information. For this study, a hybrid model can be employed, combining the spatial feature extraction capabilities of CNNs with the temporal data processing abilities of RNNs[5]. The mathematical expression of the model can be represented as follows:

$$h_t = f(W_{ih}x_t + b_{ih} + W_{hh}h_{t-1} + b_{hh}) \tag{4}$$

Among them, x_t it is the data from the input layer, h_t it is the output of the hidden layer, W gentle b They are the weight matrix and bias vector, respectively, f it is an activation function. Model training involves the selection of optimization algorithms, such as random gradient descent (SGD) or Adam algorithm, which help to adjust the model weight during the training process to minimize the loss function. The loss function usually uses cross entropy loss, and the formula is as follows:

$$L = -\sum_{i=1}^{C} y_i \log(p_i) \tag{5}$$

Among them, C it is the number of categories, y_i It's a real label, p_i It is the predicted probability. To avoid overfitting, regularization terms can be introduced or techniques such as Dropout can be used. Dropout can randomly and temporarily delete neurons from the network, with the formula:

$$h_t' = h_t \cdot d, d \sim Bernoulli(p) \tag{6}$$

Among them, d it is a Bernoulli random variable, p It is a retention probability. Table 2 shows the key parameters and settings for this stage:

Table 2. Key parameters for building and training deep learning models

Parameter Type	Description	Key Parameters
Network Type	Main type of neural network used	CNN, RNN
Loss Function	Function optimized during training	Cross-entropy loss

Optimization Algorithm	Algorithm used to adjust weights	Adam
Regularization Technique	Technique to prevent overfitting	Dropout
Learning Rate	Parameter determining the step size for weight adjustments	0.001
Batch Size	Number of data samples used in each training iteration	32

4 Experiment and Analysis

4.1 Experimental Parameter Settings

Setting experimental parameters is crucial for optimizing model performance, including network depth, learning rate, batch size, number of iterations, and regularization techniques. Correct parameter settings ensure optimal learning, avoiding overfitting or underfitting[6]. A small learning rate increases training stability but requires more iterations; a large batch size speeds up training but affects generalization ability. Table 3 lists the experimental parameters, which will be used for experimental setup and adjustment, ensuring the scientific validity and effectiveness of the experiment.

Table 3. Experimental Parameter Settings

Parameter Name	Parameter Value
Network Depth	5 layers
Learning Rate	0.001
Batch Size	32
Number of Iterations	1000
Regularization Technique	Dropout
Dropout Probability	0.5
Optimization Algorithm	Adam

4.2 Experimental Results Analysis

The experiment evaluated the impact of different network depths on the performance of the deep learning model for high school biology courses. Figure 2 shows that a 5-layer network achieved the highest accuracy of 89.6% and the lowest loss value of 0.21, indicating this depth configuration is optimal. In contrast, the 3-layer network had an accuracy of 84.2% and a loss value of 0.32, showing lower performance due to insufficient model capacity. The 7-layer network achieved an accuracy of 87.3% and a loss value of 0.25, performing worse than the 5-layer network due to overfitting, which decreased generalization ability. This indicates that the network depth should be moderate to balance the model's learning ability and generalization ability[7].

When the learning rate is 0.001, the model achieves the highest accuracy of 89.6% and the lowest loss value of 0.21. This indicates that at a lower learning rate, the model can more stably approach the global optimal solution, avoiding excessive os-

cillations during the optimization process. When the learning rate is increased to 0.01, the model's accuracy decreases to 86.5% and the loss value rises to 0.28, reflecting that a higher learning rate causes the model to skip over the optimal value during the gradient descent process of the loss function. Further increasing the learning rate to 0.1, the model's accuracy drops further to 83.1%, and the loss value increases to 0.35. This result strongly indicates that a high learning rate makes the parameter updates in the model training process too aggressive, thereby affecting the model's convergence and final performance, as shown in Figure 3.

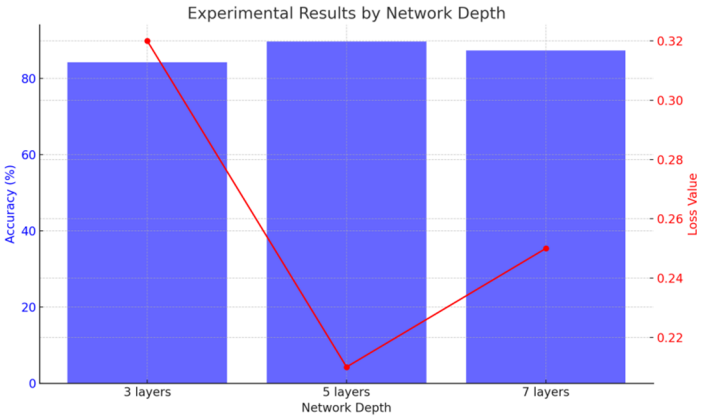

Fig. 2. Experimental Results Affecting Network Depth

Fig. 3. Experimental Results Affecting Learning Rate

When the batch size is 32, the model performs best, with an accuracy of 89.6% and a loss value of 0.21. As shown in Table 4, this indicates that with this setting, the model can effectively update weights in each batch, achieving stable convergence and high accuracy. When the batch size is reduced to 16, although the accuracy remains

high at 88.0%, the loss value slightly increases to 0.23. This is because a smaller batch size leads to greater noise in the gradient estimation during each weight update, affecting model stability. When the batch size is increased to 64, the accuracy drops to 87.9%, and the loss value increases to 0.24. This result suggests that while a larger batch size can speed up training, it may cause the weight update steps to be too large, affecting the model's fine-tuning ability and overall performance[8].

Table 4. Experimental Results Affecting Batch Size

Batch Size	Accuracy (%)	Loss Value
16	85.7	0.23
32	89.6	0.21
64	87.9	0.24

When no regularization technique is used, the model's accuracy is 85.7%, and the loss value is 0.29, indicating a certain degree of overfitting, leading to poor generalization performance. After introducing the Dropout regularization technique, the model performance significantly improves. Specifically, when the Dropout probability is set to 0.5, the model's accuracy increases to 89.6%, and the loss value decreases to 0.21, showing the best generalization ability and the lowest loss value. This indicates that Dropout effectively prevents overfitting at this probability by randomly dropping some neurons to enhance model robustness. When the Dropout probability is 0.3, although the accuracy remains high at 87.2% and the loss value at 0.24, the effect is not as good as the 0.5 setting. This is because the lower Dropout probability does not sufficiently reduce model complexity, failing to achieve the optimal regularization effect[9], as shown in Table 5.

Table 5. Experimental Results Affecting Regularization Techniques

Regularization Technique	Dropout Probability	Accuracy (%)	Loss Value
None	0	85.7	0.29
Dropout	0.5	89.6	0.21
Dropout	0.3	87.2	0.24

Through in-depth analysis of these data, model parameters can be further optimized to improve teaching effectiveness and learning experience. Adjusting the appropriate network depth, learning rate, batch size, and regularization technique helps to build a more accurate and stable deep learning model[10].

4.3 System Performance Testing

System performance testing is an important step to verify the practical application effect of the model. Table 6 shows the impact of different batch sizes on system response time. When the batch size is 16, the average response time is 120 milliseconds; when the batch size is 32, the average response time is 100 milliseconds; when the batch size is 64, the average response time is 90 milliseconds. As the batch size increases, the system response time significantly decreases, indicating that larger

batches can more effectively utilize computing resources, reducing communication and scheduling overhead. However, excessively large batches will lead to high memory usage, requiring a balance between efficiency and resource utilization.

Table 6. System Response Time Test Results

Batch Size	Average Response Time (ms)
16	120
32	100
64	90

As shown in Figure 4, the model achieves an accuracy of 90.2% on the training set, 89.6% on the validation set, and 88.5% on the test set. These results indicate that the model performs best on the training set, but performance slightly decreases on the validation and test sets. However, the small decline suggests that the model has good generalization ability. The high accuracy on the training set shows that the model can effectively learn and recognize patterns and features in the biology course; the similar accuracy on the validation and test sets indicates that the model maintains high prediction accuracy when faced with new data. This consistency validates the robustness of the model, indicating that it can reliably predict and classify biology course content in practical teaching applications.

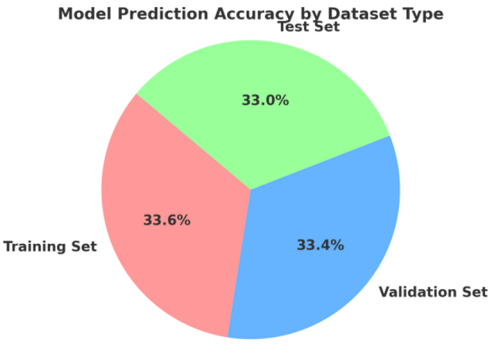

Fig. 4. Model Prediction Accuracy Test Results

Different batch sizes have a significant impact on the total training time. Figure 5 shows that with a batch size of 16, the total training time is 180 minutes; increasing the batch size to 32 reduces the total training time to 150 minutes; with a batch size of 64, the total training time further decreases to 140 minutes. Larger batch sizes can more efficiently utilize computing resources, reducing the number of iterations required to process each batch of data, thereby speeding up the training process. However, larger batch sizes increase the computational load and memory requirements for each iteration, necessitating a balance between training speed and resource utilization.

Analyzing the impact of different batch sizes on model training time reveals that increasing the batch size significantly reduces the total training time, thereby improving the efficiency of model training. This is crucial for large-scale applications and model deployment in practical teaching environments. However, the choice of batch size must balance training speed and memory resource usage to ensure overall system performance and stability. Based on these analyses, we can further optimize the training parameters of the model, enhancing its application effectiveness in high school biology course design and providing strong support for the intelligent development of the education sector.

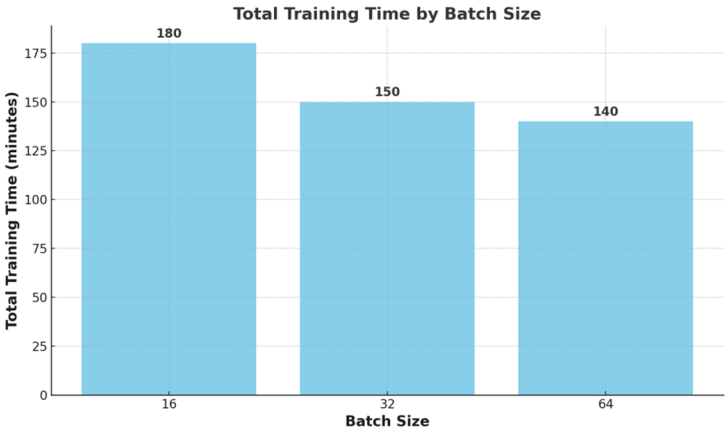

Fig. 5. Model Training Time Test Results

5 Conclusion

The application and optimization of deep learning algorithms have significantly advanced the design of high school biology courses in terms of intelligence and personalization, improving teaching effectiveness and student learning experience. Future research can continue to explore more refined data preprocessing methods and optimize deep learning model parameters to further enhance system performance, providing stronger technical support for the intelligent development of education.

References

1. Tuttle C, Davidesco I. Integrating Authentic Neuroscience Research into the High School Biology Curriculum[J].The American Biology Teacher,2023,85(7):373-378.
2. David U, Julie C, Stefan B, et al.Integration of the Topic of Social Justice into High School Biology Curricula.[J].Science & education,2021,31(4):11-19.
3. M. M S, C. J B. 'Me hizo sentir como científica': the expressed science identities of multilingual learners in high school biology classrooms[J].International Journal of Science Education,2024,46(2):155-180.

4. Khalil Y R, Tairab H, Qablan A, et al.STEM-Based Curriculum and Creative Thinking in High School Students[J].Education Sciences,2023,13(12).
5. Zhengmeng X, Hai L, Meiping W. A Course Recommendation Algorithm for a Personalized Online Learning Platform for Students From the Perspective of Deep Learning[J].International Journal of Information Technology and Web Engineering (IJITWE),2023,18(1):1-17.
6. Ma L, Zhang X, Wang Z, et al.Designing Effective Instructional Feedback Using a Diagnostic and Visualization System: Evidence from a High School Biology Class[J].Systems,2023,11(7):364-.
7. Jessica D, Jenna H, Anita S. Improving Students' Understanding of Biological Variation in Experimental Design and Analysis through a Short Model-Based Curricular Intervention.[J].CBE life sciences education,2022,21(1):ar11-ar11.
8. Ukhov P, Ryapukhin A. Development of online program for high school students on the development of engineering competencies using FDM 3D printing technology and design of electronic devices[J].International Journal of Technology Enhanced Learning,2022,14(2):217-229.
9. López, Rosendo R. Designing an inclusive curriculum for training senior high school students in the Bolivarian Republic of Venezuela[J].Transformación,2012,8(2):46-55.
10. Tal, Tsaushu. Student-centered introductory biology course: evidence for deep learning[J].Journal of Biological Education,2018,52(4):376-390.