Online-Journals.org

iJET International Journal of Emerging Technologies in Learning

iJET | eISSN: 1863-0383 | Vol. 18 No. 22 (2023) | OPEN ACCESS

https://doi.org/10.3991/ijet.v18i22.44371

PAPER

New Teaching Model of Professional Big Data Courses in Universities Based on an Outcome-Oriented Educational Concept

Haiyang Lv(✉), Jingnan Liu

College of Engineering, Yantai Nanshan University, Yantai, China

Haiyang_Lv2023@outlook.com

ABSTRACT

Most traditional Big Data courses focus on the cultivation of students' professional theoretical knowledge but neglect the training of students' programming skills. To test the effectiveness of students' programming learning, the study constructed a classification algorithm based on the TCNN model from the perspective of program identification for identifying programs written by students on their own. To improve the convergence speed and to retain more data features of the programs, the study used the Softplus-Relu combined activation function. To avoid overfitting the model, a dropout strategy was also introduced to optimize the existing TCNN model. The experimental results show that the TCNN model with the Softplus-Relu activation function converges faster, and the classification accuracy obtained is higher than 95%. The loss values and classification accuracies obtained with the TCNN-Dropout model are better than those of the GIST-KNN algorithm. The former has a loss value close to 0 and an accuracy rate of 98.9%. Thus, this indicates that the improved TCNN model proposed in the study has advantages in the identification procedure and can be used as a teaching aid for the training of big data professionals.

KEYWORDS

outcome-based education (OBE), big data professional, TCNN, Softplus-ReLU, dropout, program recognition

1 INTRODUCTION

The teaching and training program for Big Data students includes both theoretical learning of Big Data fundamentals and technologies as well as practical training such as program writing [1]. The purpose of the Big Data major is to cultivate data analysts who meet the needs of the industry through theoretical learning and comprehensive practical training [2]. However, there are certain problems with the training methods of big data majors in most universities today. For example,

Lv, H., Liu, J. (2023). New Teaching Model of Professional Big Data Courses in Universities Based on an Outcome-Oriented Educational Concept. *International Journal of Emerging Technologies in Learning (iJET)*, 18(22), pp. 214–227. https://doi.org/10.3991/ijet.v18i22.44371

Article submitted 2023-08-25. Revision uploaded 2023-09-25. Final acceptance 2023-09-25.

many universities only use a single examination paper to evaluate the students' good or bad learning of the course. This simplistic method of assessment cannot evaluate the real professional level of students [3]. Outcome-based education (OBE) is an outcome-oriented educational philosophy derived from behaviorist learning theory and proposed by an American academic in 1981 [4]. This concept places a higher weight on the students' ability to output knowledge and results after learning the course. Therefore, unlike traditional indoctrination education, the OBE teaching model no longer formulates students' curriculum training programs from top to bottom but designs teaching programs from bottom to top [5]. Based on the OBE concept, new changes will also be ushered in for the professional curriculum training of big data. This will allow traditional single teaching resources to be used to achieve more diverse teaching methods, such as video teaching and online teaching, with the help of the Internet and emerging algorithmic technologies. In addition, the establishment of a diversified assessment system will also help students gain two-way competence in theory and practice [6]. Therefore, the study proposes a TCNN model based on the dual optimization of the Softplus-ReLU activation function and dropout strategy to examine the strengths and weaknesses of the programming level of Big Data students, aiming to provide a more scientific means of program identification and evaluation for the training of talents in this profession.

2 RELATED WORK

Many scholars have conducted diverse research based on the OBE concept. Jiang M and other researchers developed a new, diverse blended online and offline learning model based on the OBE concept to address the inability of ELT to meet the diverse needs of students. Evidence from classroom practice suggests that diverse blended learning models can motivate students and increase their engagement and participation [7]. Yang et al. designed a learning model for a digital self-study course for university students based on the OBE concept. The results showed that students' attitudes towards self-directed learning were rated at 4.11 in terms of the content surveyed. Thus, digital learning tools incorporating the OBE concept can help students carry out self-directed learning effectively [8]. Yu et al. applied the OBE educational concept to a filmmaking course and developed a hybrid teaching model. In practice, this model was found to be effective in improving the quality of teaching and enhancing students' independent learning and overall application skills [9]. Liu et al. investigated a 'student-centered' teaching model that combined the OBE concept with the PAD model. The results showed that the model helped to develop students' innovation and improve their practical skills [10]. Based on the OBE concept, Li described and summarized a blended learning model based on the OBE concept and a public computing course for engineering students [11]. Pan et al. used the construction of an electronics course in a university as an example of curriculum reform as an important tool to promote capacity building. Based on the OBE concept, they studied the curriculum reform agenda to adapt it to the competency needs of higher education talents in the new era and provided useful guidance to improve the training of talents in higher education [12]. Researcher Zhu F. introduced the concept of OBE learning into the teaching of Java programming courses and used the OBE concept to set teaching objectives. It was found that the OBE-based teaching approach could improve students' learning and technical skills [13]. Chen W.P. and other researchers proposed a new model of OBE-based education for talents related to engineering education. In the study, a multi-level and multi-disciplinary engineering education system was successfully constructed, which provided insights for active research on engineering education reform in universities [14].

Many scholars have also conducted rich research on different course cultivation models. Liang Z. et al. designed a new generation of information-based learning models based on MOOC online teaching and reviewed its impact on teaching and learning as well as its feasibility and effectiveness through practical teaching applications, promoting the improvement of the efficiency and quality of online teaching in Chinese universities [15]. Researcher Huafeng L. summarized, on the one hand, that Researcher Liang and other researchers used big data technology to conduct an in-depth analysis of the existing data of the smart university and pointed out the direction for building a learning environment [16]. Researcher Cai proposed a pedagogical reform of the C language course based on the OBE concept. These recommendations will help optimize the teaching and learning of C language courses and improve the quality of teaching and learning [17]. Apiola et al. proposed design science research (DSR) as an appropriate framework for integrating computational thinking and software engineering in higher education. The results show that DSR is a powerful framework for learning to solve urgent and complex problems and for linking engineering projects to research methods and knowledge from other disciplines [18].

In summary, scholars focus primarily on the application of OBE concepts in blended learning. At the same time, scholars mainly focus on the application of digital technology in the overall curriculum training reform of computing. Fewer studies have focused on the improvement measures of the curriculum training model for big data majors. Therefore, the study will provide a student program identification model based on the TCNN model for big data majors to assist in the efficient development of practical courses.

3 TCNN PROGRAM RECOGNITION MODEL CONSTRUCTS BASED ON SOFTPLUS-RELU AND DROPOUT IMPROVEMENTS

3.1 TCNN model construction based on improved Softplus-Relu activation function

A schematic diagram of the TCNN model construction is shown in Figure 1. From left to right, the first part of the model is the pre-training unit. This unit increases the classification accuracy of the model by using a combination of existing activation functions. The second part of the model has several subtree feature detectors. This unit detects structural feature information about the program in the abstract syntax tree with the help of a tree convolution kernel. This feature information is then pooled through the pooling layer and ultimately into the fully connected and output layers.

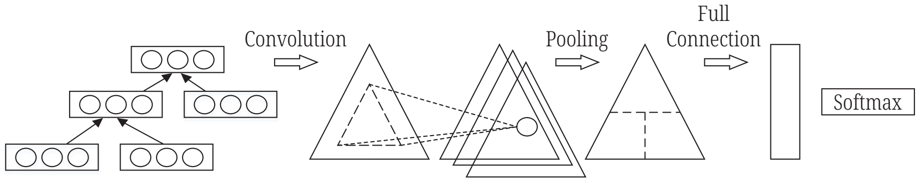


Fig. 1. Schematic diagram of TCNN model construction

Gradient descent is a common pre-training method for deep neural networks. However, as the number of training layers increases, the gradient value becomes smaller and smaller. The decreasing gradient will cause the shallow neurons of the network to delay updating their parameters, so that eventually the whole network will lose the ability to acquire the effective features of the input information. To avoid this gradient dispersion, the activation function chosen for the network needs to be a non-saturating function, such as the ReLU function, which has the advantage of being able to converge faster without the gradient dropping to a smaller value. However, the disadvantage of the ReLU function is that the sparsity of the activation function is very large. The large sparsity will reduce the effectiveness of information feature extraction. In this regard, the study uses a combination of the ReLU function and the Softplus function to retain all the non-linear mappings, i.e., to retain the effective features of the data, by using the smoothness of the Softplus function curve. Equation (1) is the mathematical expression of the combined activation function.

$$f(x) = \max(\log_e(1+e^x) - \log_e(2), x) \tag{1}$$

In equation (1), x represents the input data. When $x < 0$ is used, equation (1) is equivalent to the Softplus function. In this case, the Softplus function preserves the data information by correcting the trend of the data distribution. When $x > 0$ is used, equation (1) is equivalent to the ReLU function, which increases the convergence speed of the network. Once the input data has undergone processing by the pre-training unit, the symbols within the abstract syntax tree are converted into a sequence of real vectors. These real vectors will be used as input vectors for subtree feature detection. The subtree feature detection process can be understood as a tree convolution process. This process can be expressed with the help of equation (2).

$$y \approx \tanh\left(\sum_{i=1}^{n} W_{conv,i} \cdot x_i + b_{conv}\right) \tag{2}$$

In equation (2), n denotes the number of nodes, x_i denotes the vector of real numbers, y denotes the convolution output value, and b_{conv} denotes the bias parameter. $W_{conv,i}$ denotes the weight matrix. For a network with a sliding window depth value of 2, the convolution process is as follows: first, the sliding window slides through the abstract syntax tree vector of the program to be detected in left-to-right, top-to-bottom order. When passing a leaf node x_1 that has no children, the network automatically completes zero for it. Afterwards, the sliding window is convolved with each node. The result of the calculation requires the application of a bias and the use of a tanh activation function to obtain the final convolutional output. It's worth noting that the size of the new tree obtained through convolution remains unchanged, but the abstract syntax tree varies from one program to another. This difference will lead to uncertainty in the number of weight matrices. To improve this problem, the study establishes the triangular coordinates of the nodes. This method sets the number of weight matrices to a fixed value and determines the distribution of positions between the nodes with the help of weight coefficients. Equation (3) is a combined expression for the weight matrix.

$$W_{conv,i} = t_i \cdot W_{conv}^t + l_i \cdot W_{conv}^l + r_i \cdot W_{conv}^r \tag{3}$$

In equation (3), W_{conv}^{t}, W_{conv}^{l} and W_{conv}^{r} denote the neuron weights. $t\,l$ indicates that the node is located at the left position and r indicates that the node is located at the right position. $t_i\,l_i$ and r_i denote the weight coefficients. Where t_i denotes the depth of the node, r_i denotes the magnitude of the node near the right end, and l_i denotes the magnitude of the node near the left end. Equation (4) is the expression for the calculation of l_i.

$$t_i = \frac{d_i - 1}{d - 1} \tag{4}$$

In equation (4), d_i indicates the size of the depth at which the node is located. d indicates the total depth of the sliding window. The reason for calculating t_i is to clarify the relationship between the top and bottom distribution of the nodes in the subtree. Equation (5) is the expression for the calculation of r_i and l_i

$$\begin{cases} r_i = \left(1 - t_i\right)\dfrac{p_i - 1}{n - 1} \\ l_i = \left(1 - t_i\right)\dfrac{n - p_i}{n - 1} \end{cases} \tag{5}$$

In equation (5), p_i indicates the location of the node. n denotes the total number of nodes located at the same level in the sliding window. The introduction of node triangular coordinates serves a dual purpose: it aids in determining the node's location and streamlines the extraction of structural features by the convolution kernel. The output of the convolution is the feature information of the abstract syntax tree vector. Since the size and structure of the convolutional kernel vary from program to program, this feature information cannot be fed directly into the fully connected layer but needs to be compressed by the pooling layer. Equation (6) is the mathematical expression for the pooling operation.

$$p_j = \max_{i \in R_i} y_i \tag{6}$$

In equation (6), p_j represents the pooled output value corresponding to the region j. y_i represents the activation value obtained from the pooled region after the convolution calculation. i denotes the activation value number. After the pooling process is completed, all feature vectors will be sent to the hidden layer via the fully connected layer, and finally, supervised training will be carried out in the output layer.

3.2 TCNN model construction based on dropout strategy optimization

The TCNN model has a high network complexity to learn the mapping relationship between the network input and the network output. In cases where the number of samples available for model training is limited, there is a higher probability that the mapping relationships learned by the model may be influenced by sample noise, potentially resulting in overfitting. Overfitting is a phenomenon where the error in the training sample set decreases as the number of training session increases, but the error in the validation sample set increases. To reduce overfitting, the dropout strategy is used to improve the generalization of the model and hence the network's ability to resist overfitting. Figure 2 shows the structure of a dropout neural network compared to a conventional neural network.

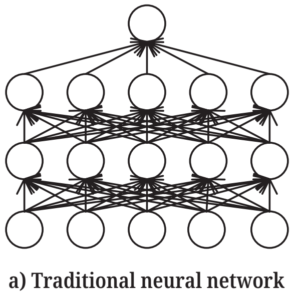


Fig. 2. Structure diagram of dropout neural network and traditional neural network

Figure 2a shows a conventional neural network. Figure 2b shows a dropout neural network, where the dropout strategy means that the model actively and randomly mutes some of the nodes (still retaining the weight values of these nodes) to achieve different combinations of nodes during network training. Enhanced node independence is achieved through multiple combinations, resulting in the generation of effective features. As the latter uses the dropout strategy, the dropout neural network can randomly activate node neurons and assign the output of inactive neurons to 0 when the network starts training and learning. The model can also obtain more data features. Equations (7) and (8) represent the mathematical expressions for the feedforward operation of a conventional neural network.

$$z_i^{(l+1)} = w_i^{(l+1)} y^{(l)} + b_i^{(l+1)} \tag{7}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \tag{8}$$

In equations (7) and (8), l denote the hidden layer number, $z_i^{(l+1)}$ denotes the input vector, $y^{(l)}$ denotes the output vector, $w_i^{(l+1)}$ denotes the weight value, $b_i^{(l+1)}$ denotes bias, and f denotes the activation function. Equation (9) is the mathematical expression corresponding to the network after the introduction of the dropout strategy.

$$\begin{cases} r_j^{(l)} \sim Bernoulli(p) \\ z_i^{(l+1)} = w_i^{(l+1)} \tilde{y} + b_i^{(l+1)} \\ \tilde{y}^{(l)} = r^{(l)} * y^{(l)} \\ y_i^{(l+1)} = f(z_i^{(l+1)}) \end{cases} \tag{9}$$

In equation (9), * indicates that the vectors are multiplied according to their elemental counterparts. $r_j^{(l)}$ denotes a random variable that follows a Bernoulli distribution. $\tilde{y}^{(l)}$ is the product of this random variable and the output of the layer $y^{(l)}$ and is used as the input value of the next layer. Figure 3 shows a schematic diagram of the node weight adjustment process after the introduction of the dropout strategy.

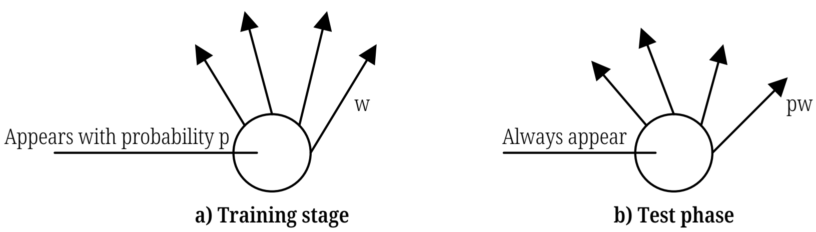


Fig. 3. Schematic diagram of node weight adjustment process after the introduction of the dropout strategy

Figure 3 shows that if a node appears with probability p during network training, then the weights of that node should be multiplied by the same probability p during the testing phase. Equation (10) is the weight adjustment formula.

$$W_{test}^{(l)} = pW^{(l)} \tag{10}$$

In equation (10), $W_{test}^{(l)}$ denotes the test phase node weights. $W^{(l)}$ The introduction of the dropout layer can improve the generalization ability of the TCNN model and, hence, the overfitting phenomenon of the network. However, it is a question of whether the dropout layer should be added to the convolutional layer or the fully connected layer of the network. From a purely theoretical point of view, the dropout layer can be connected behind the convolutional layer or the fully connected layer, and both connections can increase the accuracy of the model classification. However, in practice, when the dropout layer is connected to the convolutional layer, the accuracy of the training phase decreases significantly. It is only when the dropout layer is added to the fully connected layer that it becomes effective. The reason for this is that the number of parameters in the fully connected layer is much larger than the number of parameters in the convolutional layer, thus accommodating the randomness of node activation caused by the dropout layer. Figure 4 shows the structure of the network with the dropout layer plugged in.

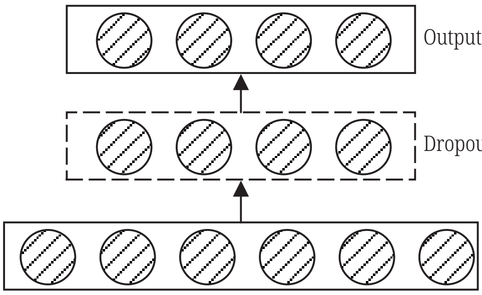


Fig. 4. Network structure diagram of access to dropout layer

Given that the dropout layer is linked after the fully connected layer, during updates to the network weights of the program AST vector, the nodes within the fully connected layer will be randomly activated with a probability of p. Equation (11) is the mathematical expression corresponding to the dropout layer connected to the fully connected layer.

$$r = mask * f(Wv + b) \tag{11}$$

In equation (11), mask denotes the random vector that is used to determine whether the inactive nodes in the network are being used. It is important to note that the mask vector obeys the Bernoulli distribution.

4 UTILITY ANALYSIS OF SOFTPLUS-RELU-DROPOUT-TCNN-BASED PROGRAM RECOGNITION MODELS

4.1 Softplus-Relu-TCNN model utility analysis

For the study, 600 C programs from an open-source website were chosen as the experimental data set. These programs contain recursive algorithms, lookup algorithms, and sorting algorithms. The data structures used are strings, queues, etc. To verify the advantages of the proposed combined activation functions in reducing the gradient dispersion and preserving the effective features of the data, four common activation functions, Sigmoid, Tanh, Softplus, and ReLU, were selected as the control functions for the Softplus-ReLU activation functions, and the loss value of the functions was chosen as the evaluation index. Figure 5 shows the loss value curves corresponding to the five activation functions.

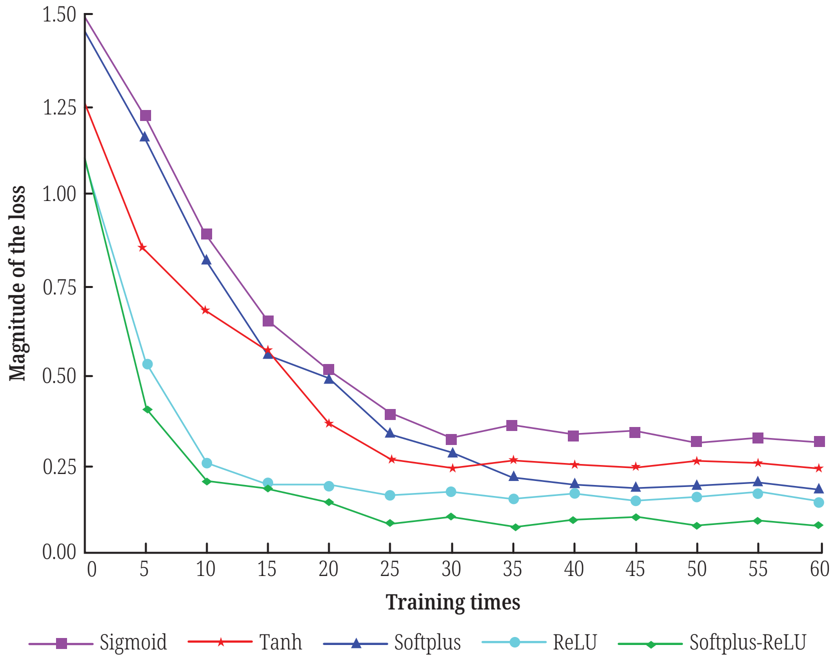
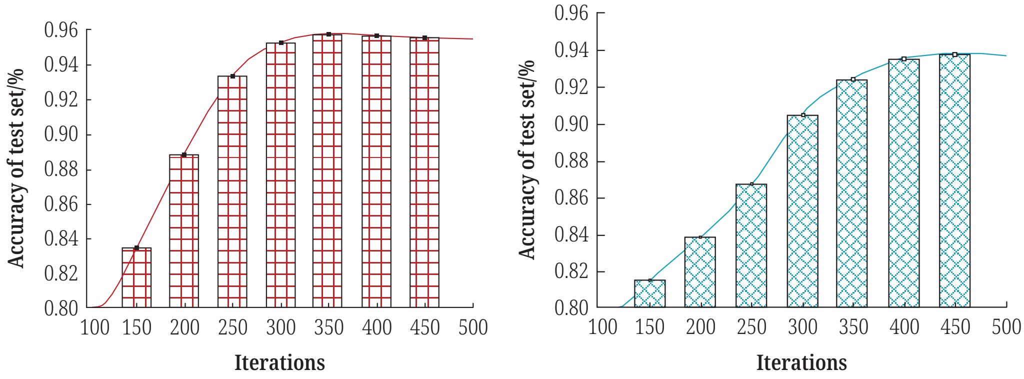


Fig. 5. Curve of loss value corresponding to different activation functions

Looking at the trend of the curves in Figure 5, we can see that the sigmoid activation function starts to converge at 30 iterations, and the loss value of the function at the end of the iteration is 0.32. The Tanh activation function tends to converge at 35 iterations, and the final loss value of the function is 0.24. The Softplus activation function tends to converge at 45 iterations, and the final loss value of the function at 60 iterations is 0.19. The ReLU activation function converges at 25 iterations, with a final loss of 0.15. The combined activation function Softplus-ReLU converges at 25 iterations, with a final loss of 0.07. From the above data, we can see that sigmoid has the largest loss, followed by Softplus. In addition, Softplus-ReLU has the same convergence speed as ReLU, taking 20 iterations less than Softplus. Thus, the combined activation function Softplus-ReLU has the smallest loss value and the fastest convergence speed, which will help to reduce the training error and improve the computational efficiency of the network.

In order to assess the impact of the Softplus-ReLU activation function on the classification effectiveness of the TCNN model, the ReLU function was chosen, which exhibited a lower loss value, for comparison experiments. The dataset chosen for the experiments was taken from open-source websites. The number of algorithm categories is 100, and the number of programs included in each category is 300. In addition, the learning rate of the TCNN model was set to 0.05, and the size of the convolutional kernel was set to 600*30. Figure 6 shows the classification results of the TCNN model.

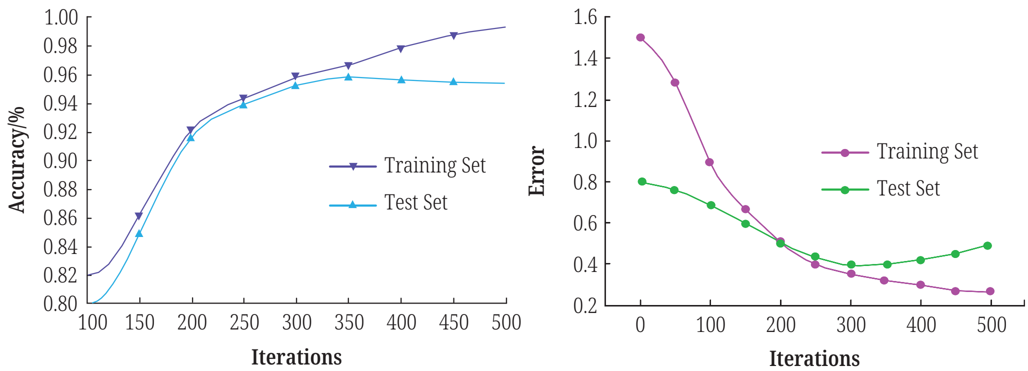


a) Classification results of TCNN model based on Softplus-ReLU

b) Classification results of TCNN model based on ReLU

Fig. 6. TCNN model classification results

Figure 6 shows that the TCNN model with the Softplus-ReLU activation function achieves higher test set accuracy than the TCNN model with the ReLU activation function when the number of iterations varies in the interval [150, 450]. The TCNN model with the Softplus-ReLU activation function started to converge at 300 iterations when the classification accuracy was 0.952. After that, the classification accuracy was 0.958 at 350 iterations, 0.957 at 400 iterations, and 0.956 at 450 iterations. Therefore, the classification accuracy of the TCNN model with the Softplus-ReLU activation function was higher than 95% at the convergence stage. The TCNN model with the ReLU activation function began to converge at 400 iterations when the classification accuracy was 0.938, and then at 450 iterations when the classification accuracy was 0.939. Therefore, the classification accuracy of the TCNN model with the ReLU activation function was less than 94%, and the convergence rate was lower than the former model. Thus, it can be seen that the Softplus-ReLU activation function has superiority in improving the classification effect of the model.



a) Learning curve

b) Error curve

Fig. 7. Learning curve and error curve of TCNN model on the training set and test set

Figure 7a shows the learning curves obtained for the TCNN model on the training and test sets. From Figure 7a, it can be seen that the number of iterations is 350, which is the cut-off point for the change and direction of the two curves. When the number of iterations varies between [100, 350], although the accuracy obtained in the training set is slightly higher than that of the test saw curve, both curves show a trend of rapid increase and then decreasing increase. When the number of iterations varied between [350, 500], the classification accuracy obtained from the learning curve for the training set continued to increase, while the classification accuracy corresponding to the learning curve for the test set started to decrease. This is an initial indication that the TCNN model is overfitting. Figure 7b shows the error curves obtained for the TCNN model for both the training and test sets. In particular, the error curve for the training set has maintained a continuous decreasing trend and obtained a minimum error of 0.26. The test set error curve shows a decreasing and then increasing trend with the number of iterations as the 300-curve inflexion point and obtains a minimum error of 0.39. This elevation of the error value at one point in the test set curve is evidence that the single TCNN model is overfitting. This, therefore, requires intervention using the dropout strategy proposed by the study.

4.2 Improved TCNN-dropout model utility analysis

In validating the classification effectiveness of the TCNN-dropout model, the dataset chosen for the experiments was kept constant. Specific information on the sample dataset is listed here. The average number of lines of code is 34.8, the average number of AST nodes is 168.2, the average number of leaf nodes is 6.8, and the average AST depth is 13.1. In addition, the model learning rate is 0.05, the AST vector dimension is 30, the convolutional kernel size is 600*30, and the fully connected layer dimension is 600. Since the magnitude of the probability p-value directly affects the classification effectiveness of the model, the study first examined the role of p-value variation on the classification accuracy of the model. Figure 8 shows the test set accuracy curves and the TCNN-dropout model learning curve under the variation of the p-value.

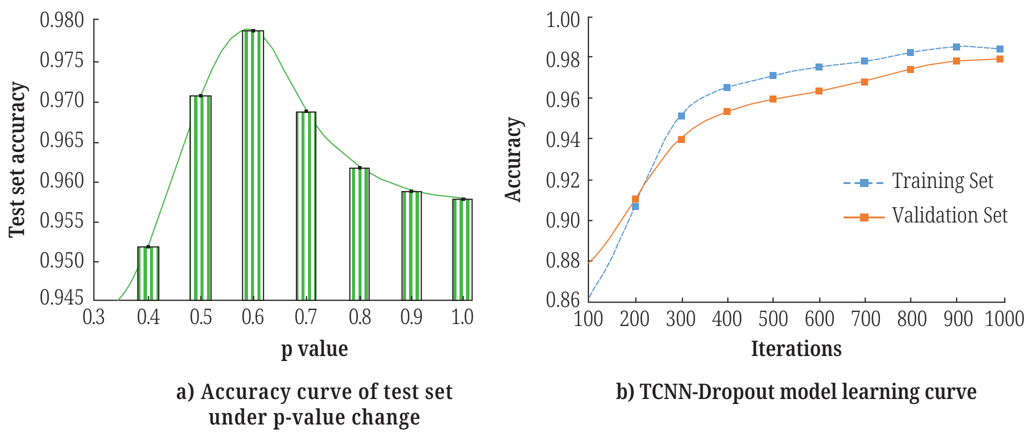


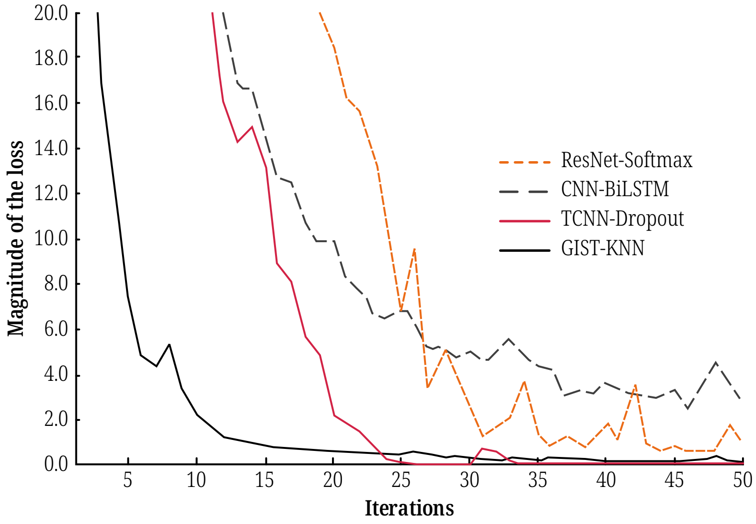
Fig. 8. Accuracy curve of the test set and TCNN-dropout model learning curve under p-value change

From Figure 8a, it can be seen that when the p-value varies between [0.3 and 1.0], the accuracy of the test set shows a trend of increasing and then decreasing. This indicates that when the p-value is too small, the number of silent neurons far exceeds that of neurons in the activated state, which will make the fully connected layer operation less stable and eventually lead to a decrease in the accuracy rate. And when the p-value is too large, the model will gradually lose the diversity of neuron combinations, which is also not conducive to the classification of the model. Therefore, only when the p-value is 0.6 does the model achieve the highest accuracy, which is 0.978. As can be seen from Figure 8b, due to the introduction of the dropout strategy, the test set curve has the same trend as the training set curve, which both shows a rapid increase and then a slow increase. At this point, the test set curve does not show any decrease in accuracy. Comparing Figure 8b with Figure 7a, we can see that the classification accuracy of the model for the test set increased by 2% after the introduction of the dropout strategy. This all indicates that the dropout strategy effectively improves the overfitting phenomenon of the TCNN model. Table 1 shows the metric results obtained after testing the TCNN-dropout model against the classical program function.

Table 1. Test results of TCNN-dropout model for classic program functions

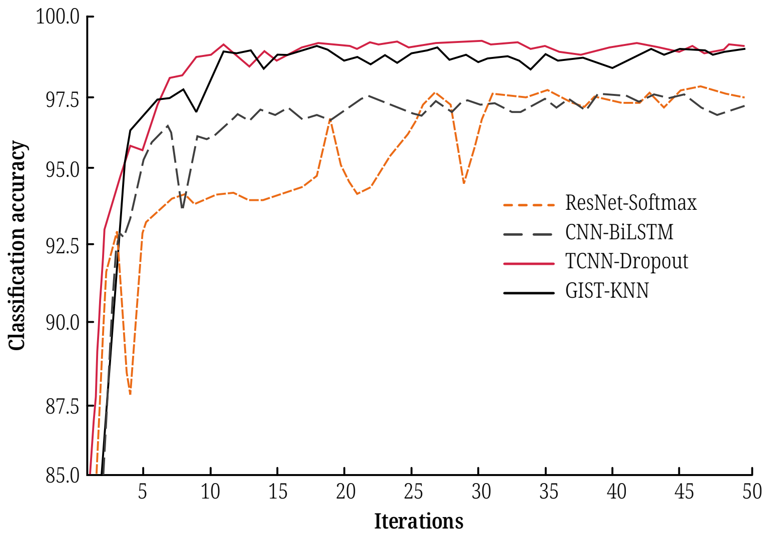
Category	Accuracy Rate (%)	Recall Rate (%)	F1 Value (%)
Bubble sort	93.1	96.2	94.1
Print linked list	90.5	91.8	90.4
Fibonacci sequence	97.9	99.8	99.1
Rebuild binary tree	97.8	99.3	98.4
Replace spaces	97.5	95.1	95.8
Array lookup	82.5	84.9	83.4
Reverse linked list	98.1	99.6	98.8
Image of binary tree	96.8	98.9	99.5
Linked list merge	99.1	99.3	97.6
Depth of binary tree	98.9	96.5	97.5
Average value	95.2	96.1	95.5

As can be seen from Table 1, the TCNN-dropout model obtained high accuracy, recall, and F1 values for all of the above 10 typical procedures tested. For example, the model obtained over 96% accuracy for seven programs such as the Fibonacci series and reconstructed binary tree; the model obtained over 95% recall for nine programs such as bubble sort and inverted linked table and only fell below 95% in the test for the printed linked table program; and the model obtained over 95% F1 value for eight programs such as replace spaces and inverted linked table. This, therefore, indicates the effectiveness of the model in learning the effective features of the programs as well as the depth features. Figure 9 shows the loss value curves and accuracy curves for the different algorithms in the test set.



a) Loss value curve of different algorithms on test set

Fig. 9. *(Continued)*



b) Accuracy curves of different algorithms on the test set

Fig. 9. Loss value curve and accuracy curve of different algorithms in the test set

As can be seen from Figure 9a, the loss value curve corresponding to the algorithm ResNet-Softmax has a large oscillation, especially after the number of iterations reaches 25. The loss value of the algorithm has difficulty in achieving convergence. The oscillations of the CNN-BiLSTM are weaker, and the amplitude is reduced compared to ResNet-Softmax. Although the algorithm tends to converge after 35 iterations, the curve still fails to converge in the end. The GIST-KNN algorithm converges quickly, starting to converge at 15 iterations with a final loss value of 0.32. The TCNN-dropout algorithm proposed in this study starts to converge at 25 iterations, with a final loss value close to 0. It can be seen that the algorithm proposed in this study achieves the smallest loss value and converges faster. As can be seen from Figure 9b, the classification accuracy curves obtained by the algorithms ResNet-Softmax, CNN-BiLSTM, and GIST-KNN continue to oscillate to varying degrees and eventually fail to obtain convergence values. The TCNN-dropout algorithm, on the other hand, started to converge after 20 iterations and eventually obtained an accuracy of 98.9%. Therefore, this indicates that the algorithm used in the study has an advantage in terms of its effectiveness in program recognition classification.

5 CONCLUSION

The ability to write programs is a key part of the training of Big Data professionals. Due to the high complexity of programs and the large amount of code, it is inefficient to test them manually with the help of teachers. Therefore, to assist teachers in understanding students' programming level more efficiently, a TCNN model based on the Softplus-Relu activation function and dropout layer optimization was designed to identify the effective features of students' written programs. The results show that the Softplus-Relu activation function has the same convergence speed as ReLU, and the final loss value obtained is 0.12 less than Softplus and 0.08 less than ReLU. Softplus-Relu converges faster than the Sigmoid and Tanh activation functions. Softplus-Relu obtains the highest classification accuracy obtained for the TCNN model with the Softplus- ReLU activation function, is 95.8%, which is a 1.2% improvement over the highest classification accuracy obtained for the TCNN model with the ReLU activation function. The TCNN models with the Softplus-ReLU activation function all achieved classification accuracies above 95% at the convergence stage, but the TCNN models with the ReLU activation function all achieved classification accuracies below 94%. In addition, the TCNN-dropout model obtained the optimal property at a p-value of 0.6. Moreover, the introduction of the dropout strategy improved the classification accuracy of the model for the test set by 2%. Compared with other classification algorithms, the TCNN-dropout model reduced by approximately 0.32 compared to GIST-KNN in terms of loss values and obtained the highest classification accuracy of 98.9%. Therefore, this indicates that the study model has advantages in program identification and can be applied to program identification for big data students.

6 REFERENCES

[1] B. K. Daniel, "Big Data and data science: A critical review of issues for educational research," *British Journal of Educational Technology*, vol. 50, no. 1, pp. 101–113, 2019. https://doi.org/10.1111/bjet.12595

[2] K. Rybinski and E. Kopciuszewska, "Will artificial intelligence revolutionise the student evaluation of teaching? A big data study of 1.6 million student reviews," *Assessment & Evaluation in Higher Education*, vol. 46, no. 7, pp. 1127–1139, 2021. https://doi.org/10.1080/02602938.2020.1844866

[3] J. Woldt, S. Prasad, and C. Ozgur, "Big data and supply chain analytics: Implications for teaching," *Decision Sciences Journal of Innovative Education*, vol. 14, pp. 155–176, 2020.

[4] C. Jiang and G. Cheng, "Problem-based learning teaching method applied to pharmaceutical engineering experiment teaching based on the outcome-based," *Indian J. Pharm. Educ. Res.*, vol. 55, no. 1, pp. 56–62, 2021. https://doi.org/10.5530/ijper.55.1.8

[5] I. Solikhah, "Revisiting the EFL curriculum in the outcome-based education framework and freedom to learn program," *Journal of Social Studies Education Research*, vol. 13, no. 2, pp. 243–264, 2022.

[6] D. Wu, P. Guo, and C. Zhang, "Research and practice of data structure curriculum reform based on outcome-based education and Chaoxing platform," *International Journal of Information and Education Technology*, vol. 11, no. 8, pp. 375–380, 2021. https://doi.org/10.18178/ijiet.2021.11.8.1537

[7] M. Jiang and X. Wu, "Exploration and practice of college english multi-blended teaching model based on OBE concept," *Open Journal of Modern Linguistics*, vol. 12, no. 4, pp. 367–379, 2022. https://doi.org/10.4236/ojml.2022.124026

[8] F. Yang and J. Fan, "Construction of OBE concept autonomous learning mode in university teaching based on the internet," *Journal of Cases on Information Technology (JCIT)*, vol. 24, no. 5, pp. 1–20, 2022. https://doi.org/10.4018/JCIT.295250

[9] Y. Yu, T. Hao, and H. Zhang, "Research and practice of hybrid teaching mode of 'film and television production technology and art,'" *Creative Education*, vol. 12, no. 9, pp. 2066–2073, 2021. https://doi.org/10.4236/ce.2021.129158

[10] L. Liu and Y. Li, "Teaching paradigm reform of economic and management courses based on the integration of OBE concept and PAD mode," *Advances in Applied Sociology*, vol. 12, no. 9, pp. 470–483, 2022. https://doi.org/10.4236/aasoci.2022.129036

[11] J. Li, "Research on the hybrid teaching model in computer courses based on the OBE concept," *Journal of Contemporary Educational Research*, vol. 5, no. 12, pp. 52–57, 2021. https://doi.org/10.26689/jcer.v5i12.2845

[12] J. Pan, L. Wen, L. Qi, and Y. T. Zhao, "Integrated reform of electronic technology courses in higher vocational colleges based on the OBE concept," *Journal of Contemporary Educational Research*, vol. 5, no. 9, pp. 20–25, 2021. https://doi.org/10.26689/jcer.v5i9.2529

[13] F. Zhu, "Reform and exploration on teaching of Java programming course based on the concept of OBE," *International Journal of Education and Humanities*, vol. 5, no. 2, pp. 72–74, 2022. https://doi.org/10.54097/ijeh.v5i2.2108

[14] W. P. Chen, Y. X. Lin, Z. Y. Ren, and D. Shen, "Exploration and practical research on teaching reforms of engineering practice center based on 3I-CDIO," *Computer Applications in Engineering Education*, vol. 29, no. 1, pp. 114–129, 2021. https://doi.org/10.1002/cae.22248

[15] Z. Liang, G. Zhang, and Y. Yu, *et al*., "Research and practice of 'M+ LIF' information-based teaching application mode based on teacher-student dual subject," *Creative Education*, vol. 13, no. 7, pp. 2201–2209, 2022. https://doi.org/10.4236/ce.2022.137139

[16] Z. Liang, G. Zhang, and S. Qiao, "Research and practice of informatization teaching reform based on the ubiquitous learning environment and education big data," *Open Journal of Social Sciences*, vol. 9, no. 2, pp. 334–341, 2021.

[17] M. Cai, "Research on teaching reform of C language course under the engineering education certification," *Open Journal of Social Sciences*, vol. 10, no. 6, pp. 120–126, 2022. https://doi.org/10.4236/jss.2022.106011

[18] M. Apiola and E. Sutinen, "Design science research for learning software engineering and computational thinking: Four cases," *Computer Applications in Engineering Educatio*n, vol. 29, no. 1, pp. 83–101, 2021. https://doi.org/10.1002/cae.22291

7 AUTHORS

Haiyang Lv, February 1982, male, Tai'an County Liaoning Province, Manchu, 2004 bachelor's degree in computer science and technology from Anshan Normal University, 2007 master's degree in Harbin University of Commerce Computer Application Technology, research Direction: Big Data Analysis. Work Experience: 2007-so far, Yantai Nanshan University, Dean of the College of Technology and Data. He has published 7 academic papers, 6 academic works and textbooks, 4 research projects, 2 patents, and 2 academic awards (E-mail: Haiyang_Lv2023@outlook.com).

Jingnan Liu, February 1981, female, Dongfeng County, Jilin province, Han nationality, 2004 Bachelor's degree in Jilin Normal University Electronic and Information Engineering, research Direction: Practice teaching. Work Experience: 2004-so far, Yantai Nanshan University, College of Intelligent Science and Engineering. She has published 8 academic papers, 4 academic works and textbooks, 4 research projects, 1 patent, and 1 academic award.