Enterprise Marketing Decision: Advertising Click Through Rate Prediction Based on Deep Neural Networks

Luyao Zhan
Faculty of Business and Accountancy, Henan Open University, Zhengzhou, 450008, China

***Abstract*—With the high-speed growth of modern information technology, online advertising, as a new form of advertising on the Internet, has begun to emerge, demonstrating enormous development potential. To improve the accurate estimation of advertising placement and improve the operational efficiency of the advertising placement system, an improved deep neural network model for forecasting advertising click through rate was studied and designed. Meanwhile, the values of the activation function and the parameter dropout are determined, and the prediction accuracy of the deep neural network model and the improved model is compared and analyzed. The experimental results show that the training time of the improved prediction model has been shortened by about 73.25%, resulting in a significant improvement in computational efficiency. When the number of iterations is 110, the logarithmic loss function value is 0.208, and the logarithmic loss function value of the improved model is 0.207, with an average loss reduction of 0.4%. In the area comparison under the receiver operating characteristic curve, the pre improved model was 0.7092, and the improved model was 0.7207. Meanwhile, compared to before the improvement, the prediction accuracy of the improved model increased by 1.6%. The data validates that the optimized model has high prediction precision and efficiency, and has certain application potential and commercial value in marketing.**

***Keywords*—Click through rate prediction; deep learning; deep neural network; online advertising; marketing**

I. INTRODUCTION

Nowadays, the growth of internet companies has been inseparable from advertising marketing, and Click Through Rate (CTR) prediction remains a key issue in the advertising field. With the continuous improvement of internet commerce and search engines, online advertising has become one of the main ways for businesses to promote and market [1]. With the development of information technology such as the Internet and intelligent terminals, the scale of the domestic advertising industry market has been continuously expanding in the past few years. The consecutive expansion of the advertising industry has driven the sustained growth of internet advertising. In internet marketing, CTR is the ratio of the click numbers on a certain content on a website page to the quantity of times. It shows the level of attention paid to a certain content on a webpage and is taken to measure the attractiveness of advertisements [2]. Based on information such as user behavior attributes and advertising characteristics, a prediction model can be constructed using deep learning methods for advertising CTR prediction [3]. Deep learning methods have achieved good results in speech and image recognition, and can also reduce manual repetition in the field of advertising CTR prediction, controlling the accuracy and efficiency of advertising production [4]. The advantage of Deep Learning Model (DLM) in automatically extracting higher-order features improves accuracy, but it lacks a certain degree of interpretability compared to manually extracted features [5]. Therefore, this study designed the advertising CTR prediction model of the Deep Neural Network based on Sampling (SDNN), and determined the activation function and the parameter dropout. This article also compares and analyzes the prediction accuracy of four models: Deep Neural Network (DNN), SDNN, DNN trained with dropout, and SDNN trained with dropout. The research aims to improve the operational efficiency of the advertising placement system, so that enterprise marketing can determine more accurate decisions. The research content mainly includes six sections. The second section is a review of the current research status of advertising CTR prediction and deep learning both domestically and internationally. The third section constructs an improved DNN advertising CTR prediction model. Under this, the first sub-section designs an advertising CTR model, and the second sub-section proposes an SDNN advertising CTR prediction model. Section four analyzes the results of the improved DNN advertising CTR prediction model. The fifth section is a discussion on improving the advertising CTR prediction model of DNN. Section six is the conclusion.

II. RELATED WORKS

The high accuracy of CTR prediction can help advertisers and advertising platforms increase revenue and gain greater benefits. Many researchers have proposed various ideas and methods here. Cai et al. constructed a neural network and global attention mechanism model to precisely forecast the possibility of users clicking on advertisements, achieving interaction between low and high order nonlinear features, while promoting optimization of deep structures. This model has good predictive performance [6]. Xue and other professionals have constructed adaptive hash algorithms and DNN models to reduce redundant features in deep CTR prediction problems. It can automatically select practical features for high-order interaction. This model has good performance, low complexity, and requires less training time [7]. Liu and other scholars proposed an A-CTR-P that combines big data analysis to achieve mobile computing of advertising CTR logs, and used power-law distribution for log preprocessing and category feature extraction. This model has

good prediction accuracy [8]. Zhou's team proposed a model that combines advertising topic distribution network and recurrent neural network to control data transmission for e-commerce product advertising recommendation. This model has high accuracy while reducing computational complexity [9]. To improve advertising profits and promote user experience, Ghorbel et al. built the upper confidence limit and A-CTR-P of GA built on the LSTM network to improve the feature selection of micro targeting technology and optimize hyper-parameter. The accuracy of this method reaches 87%, the precision reaches 89%, and the recall rate reaches 92% [10]. Liu et al. designed a user preference network model for a recommendation system that combines attention for CTR prediction under video recommendation. This model has good predictive performance and solves the time series problem of user feedback information [11].

Deep learning has made good progress in the image and natural language processing, and is widely used in daily life. Hung's team proposed an early warning model that combines machine learning and deep learning algorithms to identify learners at risk in order to analyze student behavior performance. This model captured 59% of high-risk students, with an overall accuracy of 86.8% [12]. Ma et al. built transfer learning and deep learning models for the prediction of residual life transfer of batteries with different formulations to avoid the loss of battery information. It saved testing costs and ensures high temperature robustness, with high prediction accuracy, while optimizing batteries with different formulations [13]. To avoid employee turnover, professionals such as Ozmen EP have constructed a hybrid extended convolutional decision tree model with good classification accuracy based on convolutional neural networks and grid search optimization [14]. Deng and other scholars designed a topology optimization method combining geometric depth learning to elaborate the density distribution function for compliance and pressure constraints. It ensured the boundary smoothness and effectively reduces design variables and controls structural complexity, making it very practical [15]. Hu et al. raised a deep reinforcement learning method driven by curiosity to optimize intelligent and interconnected automotive power control systems to accelerate training speed and achieve a good balance of universality. Under this algorithm, the control behavior rate had been optimized by 50.43%, and the learning productivity had been improved by 74.29% [16]. To optimize the energy efficiency of cooperative spectrum sensing, He H's team has built a high-performance reinforcement learning and deep learning framework for graphical neural networks to promote the improvement of system energy efficiency and spectrum efficiency [17].

In summary, many professionals have conducted research on CTR prediction and deep learning, and applied them to various fields. However, there are still few research results on using deep learning to predict CTR, and this direction has strong potential application value for lifting the accuracy of advertising prediction.

III. A-CTR-P BASED ON DNN

To rise the advertising prediction accuracy and the operational efficiency of the advertising delivery system, A-CTR-P is designed to improve DNN, and the activation function and dropout values are determined.

A. Construction of A-CTR-P

In today's era, traffic monetization and product promotion depend on advertising. The precise recommendation of advertisements helps to improve user experience and the ability to monetize platform traffic. The key factor in achieving precise recommendations is CTR estimation. The accuracy of CTR estimation affects the decisions of advertisers and advertising platforms regarding advertising placement, thereby determining bids based on user click behavior. When advertising platforms calculate the bidding price of each advertiser, CTR estimation is the core link. The relationship between internet platform users and advertisers is Fig. 1.

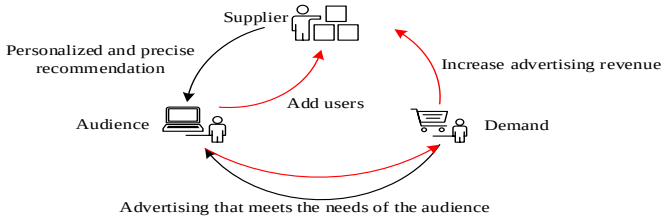


Fig. 1. Calculate the tripartite relationship of advertising.

In Fig. 1, internet platforms enhance user experience and increase user stickiness through personalized and precise recommendations. The user's click behavior on the pushed product is used as feedback to promote the optimization of the push product mechanism on various internet platforms. This can increase platform revenue while improving user experience. The data related to CTR prediction comes from the Criteo dataset. 0 means the user without clicking, and 1 indicates that the user clicked. There are 13 numerical features and 26 string features in the data [18]. Before building a prediction model, it is necessary to complete data processing and feature engineering. The data preprocessing process involves descriptive statistics of the overall data, as Fig. 2.

In Fig. 2, first of all, the data is processed separately according to the type in the database, and the string data is characteristic processed, that is, One-hot coding. The data that has been uniquely hot coded will become a high-dimensional sparse matrix [19]. Next, data normalization and descriptive statistics were performed on the processed string and numerical data, and the data was segmented into a test and a training set in a 1:4 ratio. Feature engineering includes feature processing and feature selection. Before establishing a prediction model, the first step is to standardize the features, scale the original data proportionally, and map it to the inter cell range. The expression of the Max Min standardized transformation is Eq. (1).

$$x = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (1)$$

In Eq. (1), after feature processing, the numerical data is uniformly normalized to the range of [0,1] intervals. The technical process of feature selection is Fig. 3.

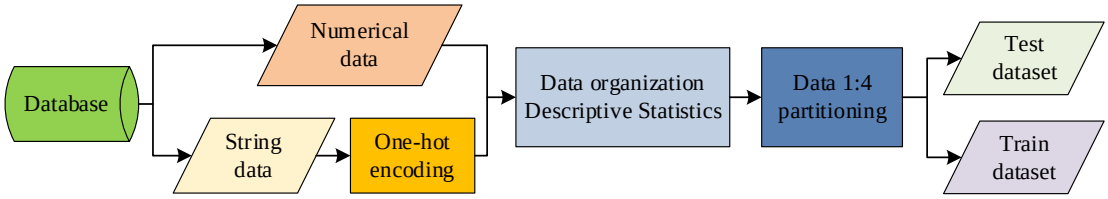


Fig. 2. Data preprocessing flowchart.

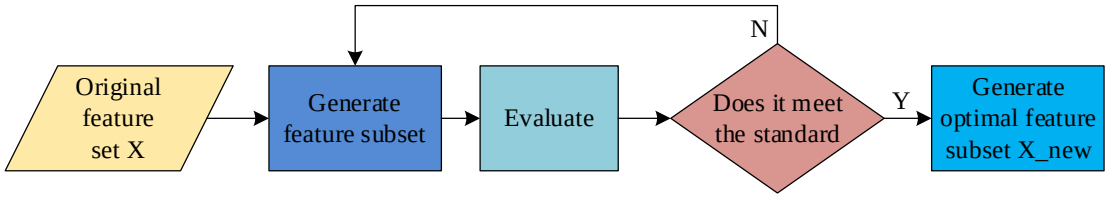


Fig. 3. Feature selection technical process.

In Fig. 3, a part of the feature subset is generated from the original feature set, and the optimal feature subset is generated by evaluating the selected subset that meets the criteria. If the criteria are not met, continue selecting in the feature subset. In the process of calculating advertising click logs, there are extremely few cases where users click on the logs, with the vast majority being cases where users have not clicked. There is a problem of category imbalance in CTR prediction. Category imbalance refers to the significant difference in the number of training samples for sample data in classification tasks. To avoid imbalanced class distribution, data sampling uses an under sampling method that removes the number of multiple class samples. The CTR prediction results only have two possibilities: clicking or not clicking, which belongs to the binary classification problem: user clicks are represented by y=1, and user clicks are represented by y=0. Table I shows the binary confusion matrix on the obtained test set.

TABLE I. DICHOTOMOUS CONFUSION MATRIX

Predictive Value / **True Value**	**Regular Class** $(y=1)$	**Negative Class** $(y=0)$
Regular class $(\hat{y}=1)$	TP	FN
Negative class $(\hat{y}=0)$	FP	TN

TP represents the amount of correctly predicted positive samples. FP is the prediction error numbers for negative examples. FN represents the quantity of prediction errors for active samples. TN refers to the correct negative example amounts predicted. From the confusion matrix, the total evaluation index of the method is Eq. (2).

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (2)$$

In Eq. (2), Accuracy, Precision and Recall rate are all used as measurement indicators. Precision is the possibility of true prediction in the positive sample predicted by the model, expressed as follows.

$$\Pr ecision = \frac{TP}{TP + FP} \quad (3)$$

The recall rate represents the proportion of all active examples, as expressed in Eq. (4).

$$\mathrm{Re}\,call = \frac{TP}{TP + FN} \quad (4)$$

The ROC indicator is connected with True Positive Rate (TPR) and False Positive Ratio (FPR), where the TPR calculation formula is Eq. (5)

$$TPR = TruePositiveRate = \frac{TP}{TP + FN} \quad (5)$$

The FPR calculation formula is Eq. (6).

$$FPR = FalsePositiveRate = \frac{FP}{FP + TN} \quad (6)$$

TPR and FPR are interdependent, and the larger the TPR and the smaller the FPR, the more superior the classification performance. Logloss index of the logarithmic loss function measures the prediction accuracy of the classifier, and can also be used as the standard of the classification effect, as shown in Eq. (7).

$$\log loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{i=1}^{M} y_{ij} \log\left(p_{ij}\right) \quad (7)$$

In Eq. (7), N is the gross samples, and p_{ij} represents the predicted CTR of users to advertisements. The logarithmic loss of all samples represents the average logarithmic loss of each sample, and the smaller the value, the better the model performance. The evaluation indicator Area Under Curve (AUC) of the method is expressed as the area enclosed by the coordinate axis under the ROC curve. In terms of CTR prediction, the larger the AUC, the better the training effect. The smaller the Logloss value, the higher the accuracy of the prediction model [20].

B. DNN-based Prediction of Advertising CTR

On the basis of constructing A-CTR-P, further research is conducted on the use of DNN for CTR prediction. DNN is a multi-layer unsupervised neural network that uses the output features of the previous layer as inputs to the next layer for feature learning. After layer by layer feature mapping, the features of existing spatial samples are mapped to another feature space to learn better feature expression for existing inputs. DNN has multiple nonlinear mapping feature transformations that can fit highly complex functions [21]. The structure is Fig. 4.

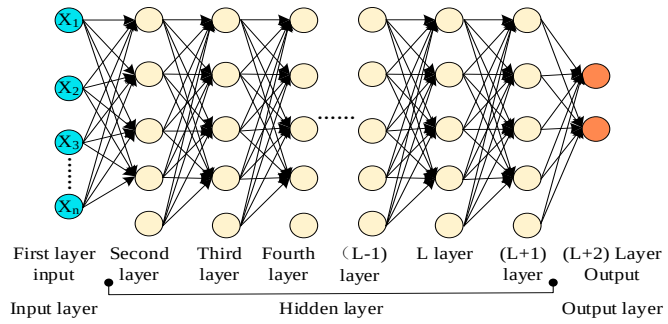


Fig. 4. DNN model structure.

In Fig. 4, DNN includes input, hidden and output layers. In CTR prediction, i.e. the binary classification {0,1} problem, the DNN input layer with an L-layer hidden layer is Eq. (8).

$$X = \{X_1, X_2, X_3, ..., X_n\} \quad (8)$$

n in Eq. (8) means the input quantities. The hidden layer is Eq. (9).

$$h^{(l)} = f\left(W^{(l)}h^{(l-1)} + b^{(l)}\right)\left(\forall l \in 1, 2, 3, ..., L-1\right) \quad (9)$$

In Eq. (9), $h^{(l)}$ represents the input vector of L+1, $f(\cdot)$ is the activation function. $W^{(l)}$ represents the weight matrix of L-1. $b^{(l)}$ is the offset vector of L. The output layer is Eq. (10).

$$y_{pre} = \arg\max_{C} P\left(y = C, X; w, b\right) \quad (10)$$

In Eq. (10), X represents the input vector, and $P\left(y = C, X; w, b\right)$ represents the probability that the output is equal to C. y_{pre} is the output of the final model and the corresponding category C when $P\left(y = C, X; w, b\right)$ is at its maximum, resulting in category 0 or 1. The key solution $\{w, b\}$ of DNN model calculation makes the loss function the smallest, and the DNN weight could be trained through Stochastic Gradient Descent (SGD). SGD is an optimization algorithm used to update DNN parameters on a gradient basis. Each iteration will randomly select a small batch of samples to calculate the gradient of the loss function, and use the gradient to update the parameters. This random characteristic makes the algorithm more robust, avoiding getting stuck in local minima, and also accelerates training speed [22]. Random gradient descent is the average extraction of a small batch of samples $B = \left\{x^{(1)}, ...x^{(m)}\right\}$ from the training set. m is usually a relatively small number. When the amount of data is large, it is necessary to iterate with abundant samples to gain the optimal solution, as expressed in Eq. (11).

$$g = \frac{1}{m}\nabla_\theta \sum_{i=1}^{m} L\left(x^{(i)}, y^{(i)}, \theta\right) \quad (11)$$

Eq. (11) uses samples from a small batch of B. To obtain the minimum cost function and the optimal parameters, the weight update rules used in the training of DNN are Eq. (12).

$$\varpi^{(r+1)} = \varpi^{(r)} - \nabla E\left(\varpi^{r}\right) \quad (12)$$

In Eq. (12), ϖ is the weight, ∇ is the gradient, and E is the error function. The selection of activation function includes Sigmaid and Rectified Linear Unit (Relu). In the sigmoid function, when any input value is x, the output value is between the intervals (0, 1). When the input value is 0, the output value is 0.5. The sigmoid function can be represented as Eq. (13).

$$f\left(x\right) = \frac{1}{1 + \exp\left(-x\right)} \quad (13)$$

The Relu function is Eq. (14).

$$g\left(x\right) = \max\left(0, x\right) \quad (14)$$

The derivative of $g\left(x\right)$ can be obtained from Eq. (14), which can be gained as Eq. (15).

$$g\left(x\right) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \quad (15)$$

In Eq. (15), if the input value > 0, the output is equal to it; If it < 0, the output = 0. DNN can handle high-dimensional sparse category features, but its ability to learn samples with multiple parameters, longer training time, and fewer categories is limited. SDNN is a research on random undersampling of data based on DNN, in order to improve the noise and imbalance of the data. Random undersampling can improve runtime and solve storage problems by reducing the number of samples. The SDNN structure is listed in Fig. 5.

In Fig. 5, the input numeric data and string data are normalized and heat coded respectively to obtain a normalized numeric matrix and a sparse matrix. The two are connected by a matrix to form training data. The training data will be normalized using resampling techniques. Input the resampled data for deep feature learning of DNN, and then output the predicted value probability and corresponding predicted labels. SDNN is a new method improved in DNN. After normalizing and encoding the input data, a random under sampling process is added to the data, thereby improving the impact of data imbalance on DNN and mining complex association relationships in features. The process framework of the SDNN algorithm is Fig. 6.

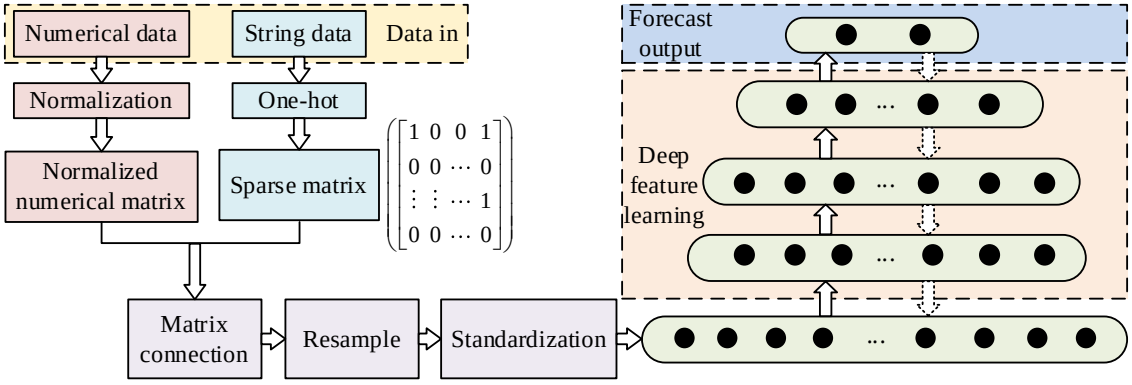


Fig. 5. SDNN structure.

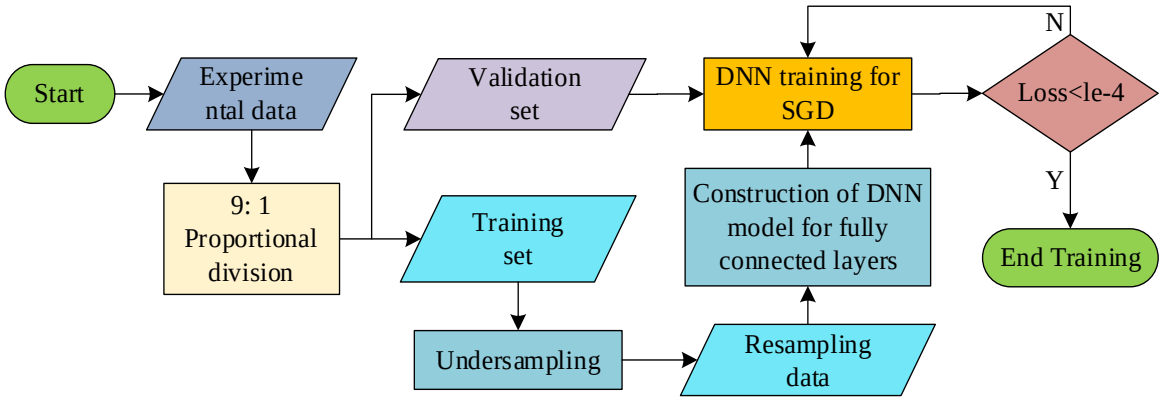


Fig. 6. SDNN algorithm process.

In Fig. 6, the input training data is separated into a training and a validation set in 9:1. Before constructing the DNN, a random under sampling technique is taken to the training set, and then a fully connected DNN is established on the resampled dataset. In training, the loss function on the verification set needs to be calculated. When the number of iterations increases and the loss function no longer changes or changes very little, that is, loss<1e-4, the training ends. For parameter issues, the study adopts the dropout technique proposed by Hintion. Dropout can ensure that weight updates no longer rely on the joint action between hidden layer nodes. Overall, the network structure undergoes changes during each DNN training session. The general effect of the final model depends on the synthesis of different model predictions each time.

IV. Analysis of A-CTR-P Results Based on DNN

Starting from DNN, to construct a prediction model and first conduct model exploration experiments based on DNN. To comprehensively verify the effectiveness of the DNN, the DNN model and the improved model SDNN were studied and designed. The activation function Relu and Sigmaid were compared and analyzed, and the key parameter dropout was effectively explored. First, a comparative analysis of the activation function Relu and Sigmaid is carried out, as listed in Fig. 7.

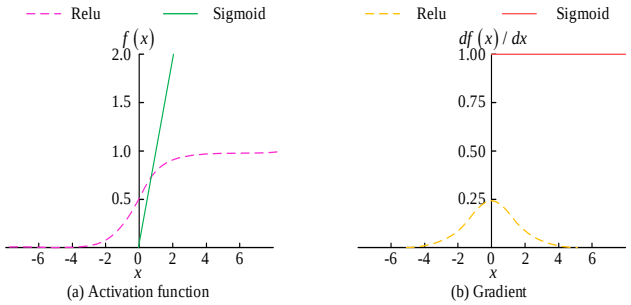
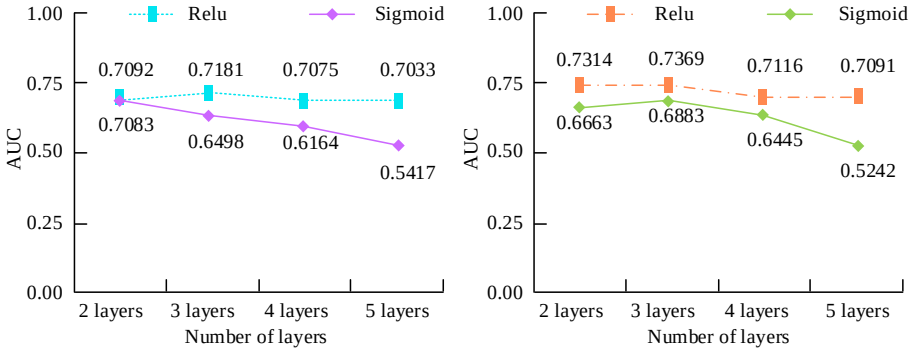


Fig. 7. Relu and sigmoid activation function and gradient comparison diagram.

In Fig. 7(a), the input value of the sigmoid activation function is less than 0 and approaches 0, while the output value is greater than 0 and approaches 1. In the process of backpropagation, only when the input is around 0 has good activation. In Fig. 7(b), the sigmoid function makes the neural network better at feature recognition, but generally causes the gradient to disappear within five layers. The Relu activation function is constant at gradients greater than 0, effectively avoiding the matter of gradient disappearance. The gradient of the Relu function is relatively stable compared to the sigmoid function, indicating that the AUC value of the Relu is relatively stable. To study the feature learning ability of hidden layers, the effects of the model structures of Relu DNN/SDNN and Sigmaid DNN/SDNN on AUC were compared and analyzed, as displayed in Fig. 8.



(a) Comparison effect of adjusting the number of hidden layers in Sigmoid/Relu DNN

(b) Comparison effect of Sigmoid/Relu SDNN hidden layer parameter adjustment

Fig. 8. Comparison rendering Relu/sigmoid activation function and gradient comparison diagram.

In Fig. 8(a), the AUC values of Sigmaid and Relu are similar when the hidden layers are 2. The AUC curve of Sigmaid DNN gradually deducts as the hidden layers increases. When the hidden layer amounts are 5, the AUC value is the lowest, at 0.5417. The AUC values of Relu DNN are all around 0.70, and the curve is relatively stable. The AUC reaches its peak at 0.7181 when the layers are 3. The AUC of Relu DNN has significantly lifted, and Relu represents the relative stability of the prediction effect. In Fig. 8(b), until the layers are 5, the AUC value of Sigmaid SDNN is the lowest, at 0.5242. The AUC value of Relu SDNN is relatively stable, reaching a peak of 0.7369 when the number of hidden layers is 3. Under the same model structure, the AUC value of SDNN is higher than that of DNN, indicating that DNN random under sampling helps to improve the AUC value. The Relu SDNN model was selected for the study, and the parameter settings are Table II.

TABLE II. DNN PARAMETER SETTINGS

Items	Settings
Model structure	2022-1024-1024-800-2
Objective function	Mean_squared_error
Max-iterations for training	200
Activation function	Relu
Regularizer	L2

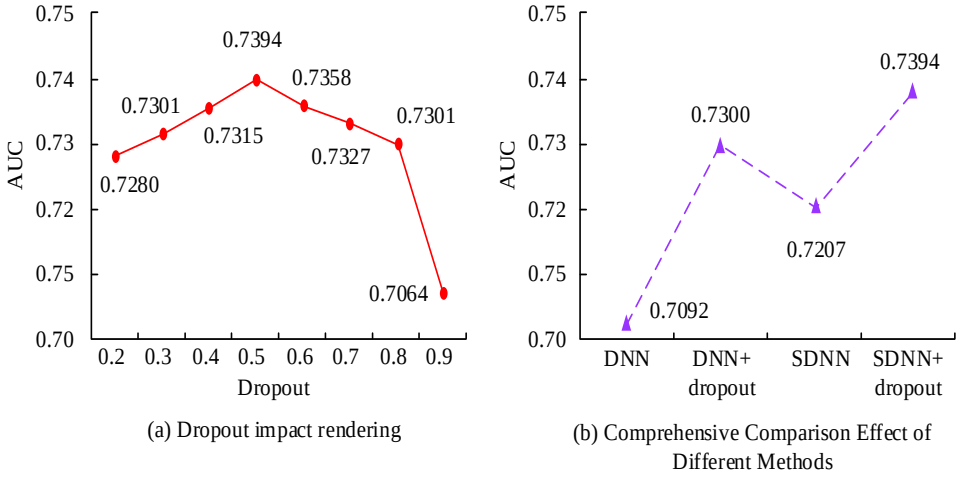
To ensure the effectiveness of the algorithm, research was conducted under GPU parallel acceleration. The training time of the DNN model is 277.1801s, and the SDNN is 74.1490s, which is approximately four times that of the SDNN. The training time of SDNN has been reduced by about 73.25%, greatly improving the computational efficiency of DNN. DNN trained on the original data and reduced the data samples by 60% after under sampling. The training scale of SDNN has decreased. However, the reduction of training data did not make the prediction performance of SDNN worse; on the contrary, it also improved the computational efficiency of the algorithm. Considering that the number of network layers and the number of iterations in the training phase of SDNN also have an impact on the prediction results of the model, in order to obtain more reasonable parameter values, the experiment was trained on a training set with a data size of 400000 samples. In the experiment, different network layers and iterations were selected to obtain the AUC values of the model, as shown in Table III.

TABLE III. AUC VALUES OF THE MODEL UNDER DIFFERENT NETWORK LAYERS AND ITERATIONS

Number of layers	Iterations					
	50	70	90	110	130	150
2	0.8103	0.8213	0.8237	0.8251	0.8248	0.8242
3	0.7821	0.8103	0.8194	0.8215	0.8202	0.8214
4	0.8215	0.8327	0.8392	0.8482	0.8501	0.8503
5	0.7979	0.8267	0.8372	0.8380	0.8376	0.8371
6	0.8064	0.8132	0.8158	0.8263	0.8257	0.8278

In Table III, regardless of the number of iterations, the AUC of the model is the highest when the number of network layers is 4. Regardless of the number of network layers, when the number of iterations is less than 110, the AUC value of the model remains increased. When the number of iterations is 110 or above, the AUC value of the model is relatively stable and does not change much. To determine the values of the key parameter dropout in the DNN model, the study was conducted from 0.2 to 0.9 in steps of 0.1. The impact of adjusting the dropout parameter on the prediction performance of SDNN, as well as the comprehensive comparison of the four models with a dropout of 0.5 is exhibited in Fig. 9.

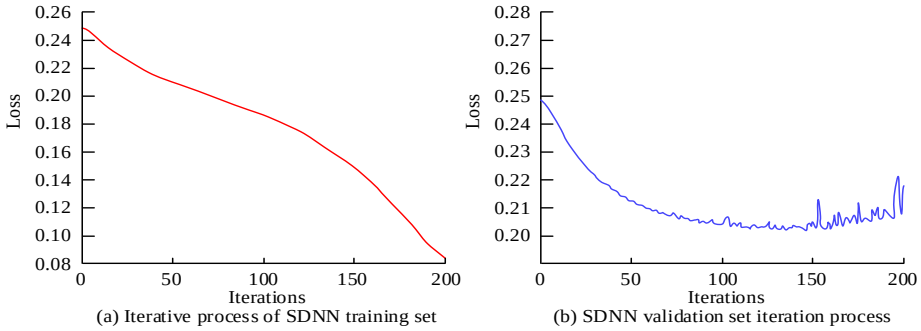
In Fig. 9(a), as the dropout increases, the AUC curve gradually rises and gradually decreases after reaching its peak. When the dropout is greater than 0.8, the AUC curve sharply decreases. When the dropout is 0.5, AUC reaches its peak at 0.7394. When the dropout is less than 0.8, the AUC curve is relatively stable. Therefore, the parameter dropout value is set to 0.5. In Fig. 9(b), the AUC value of DNN is 0.7092, the AUC value of DNN model trained with dropout is 0.7300, and the AUC value of SDNN is 0.7207. The SDNN model trained with dropout has the highest AUC, which is 0.7394. The AUC value of SDNN has been improved, indicating that resampling can achieve the goal of eliminating data imbalance. Balanced data categories have a certain effect on improving prediction performance. As a result of the large size of the data itself, the SDNN was selected for comparative analysis between the training and the test set. When the dropout parameter is 0.5, the training iteration process of SDNN is Fig. 10.



(a) Dropout impact rendering

(b) Comprehensive Comparison Effect of Different Methods

Fig. 9. Comprehensive comparison chart.



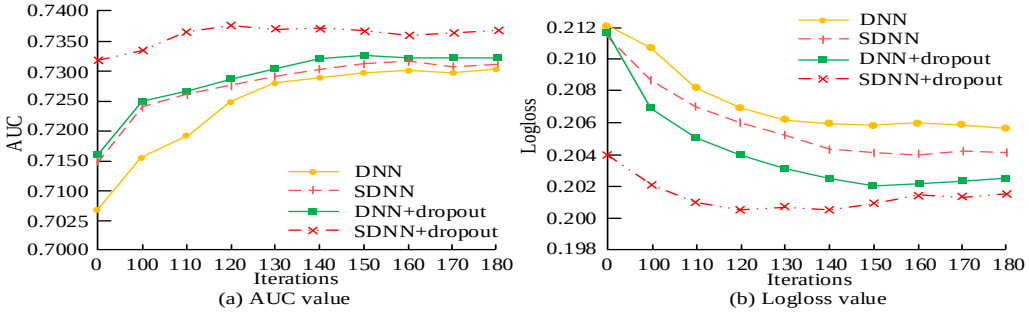
(a) Iterative process of SDNN training set

(b) SDNN validation set iteration process

Fig. 10. The training iteration process of SDNN.

In Fig. 10(a), when the number of iterations of the SDNN model training set trained with dropout is 200, the loss function curve still shows a downward trend, and the loss function is 0.085. In Fig. 10(b), when the number of iterations of the verification set is the 110th, the loss function converges, and the value of the loss function is 0.201. The number of iterations decreases rapidly before 110, and after 110, the curve oscillates and no longer decreases. SDNN is effectively trained. To evaluate the predictive precision, a comparative analysis was performed on four models: DNN, SDNN, DNN trained with dropout, and SDNN trained with dropout. The AUC and Logloss values of the four methods are listed in Fig. 11.

In Fig. 11(a), the SDNN trained with dropout has the highest average AUC, reaching convergence at 110 iterations with an AUC of 0.7375. The DNN trained with dropout converges at 130 iterations, with an AUC of 0.7300. SDNN converges at 120 iterations, with an AUC of 0.7260. The average AUC of DNN is the lowest, and it gradually converges when the number of iterations is 140, at which point the AUC is 0.7255. In Fig. 11(b), the Logloss curve of the SDNN trained with dropout converges at 110 iterations, with a Logloss value of 0.201. When the number of iterations is 110, the Logloss value of DNN is 0.208, the Logloss value of SDNN is 0.207, and the Logloss value of DNN trained with dropout is 0.203. At this point, the Logloss value of SDNN decreased by 0.4% compared to DNN. The application effects of the four models in the test training set are represented by ROC curves. At the same time, to compare the relationship between the ROC curves of various models more clearly, the area of curve FPR0.2 to 0.3 is enlarged. The comparison of ROC curves and ROC partial curves of the four models are demonstrated in Fig. 12.



(a) AUC value

(b) Logloss value

Fig. 11. Comparison of convergence rates under different models.

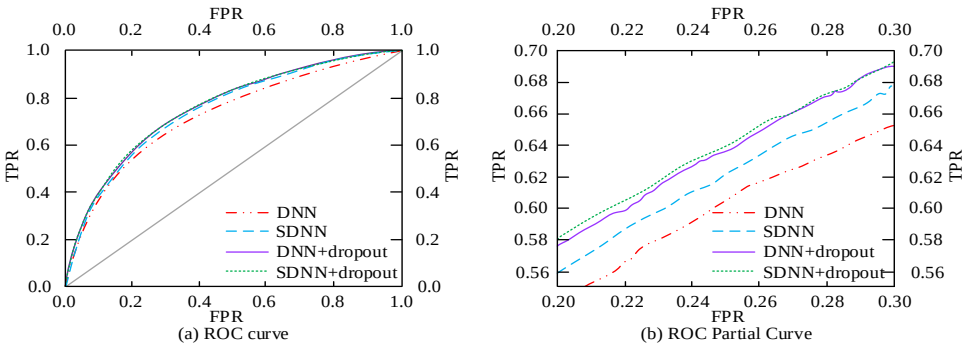


Fig. 12. Comparison between ROC curve and ROC local curve.

In Fig. 12(a), the area under the ROC of DNN is the smallest, at 0.7092, indicating that its AUC is the smallest among the four models. The area under the ROC curve of DNN and SDNN trained with dropout is similar to that of SDNN trained with dropout. In Fig. 12(b), the AUC value of DNN trained with dropout is 0.7300, SDNN trained with dropout is 0.7394, and SDNN is 0.7207. Compared to DNN, SDNN increased AUC by 1.6%. Although the improvement is not significant solely in terms of numerical values, the CTR prediction problem of internet advertising companies involves massive amounts of data every day. Due to the difficulty in estimating costs, any slight increase in AUC value will result in significant advertising effectiveness.

V. Discussion

The complexity of the model is related to its performance. The higher the complexity of the model, the more accurate the error of the training set at the end of the month, and the resulting model accuracy. However, higher complexity may lead to an increase in the computational workload and training time of the model, leading to an increase in research costs. Therefore, it is necessary to study and select the appropriate number of network structure layers and iterations. The study selects a network layer of 4, iteration number of 110, and parameter dropout value of 0.5. Under these conditions, the model studied can improve the predictive accuracy of advertising placement in a targeted manner, which plays a certain role in making marketing decisions for enterprises.

VI. Conclusion

To improve the prediction accuracy of advertising placement, an improved DNN based A-CTR-P: SDNN was studied and designed, while the activation function and parameter dropout were determined. The prediction accuracy of DNN, DNN trained with dropout, SDNN, and SDNN trained with dropout were compared and analyzed. The results indicate that when the hidden layers have 5, the AUC of Sigmaid DNN is the lowest, at 0.5417. When there are 3 hidden layers, the AUC of Relu DNN reaches its peak at 0.7181, and the AUC is relatively stable at around 0.70. The AUC value of Relu DNN has greatly lifted, and Relu represents the relative stability of the prediction effect. Under the same model structure, the AUC value of SDNN is higher than that of DNN, and the improvement effect of SDNN on DNN is conducive to the improvement of AUC. When the dropout is around 0.5 and the SDNN prediction model training set has 200 iterations, the loss function curve still maintains a downward trend, at which point the loss function is 0.085. The loss function curve does not decrease until the 110th time of the validation set, at which point the function value is 0.201. When the number of iterations is 110, the Logloss value of DNN is 0.208, and the Logloss value of SDNN is 0.207. At this time, the Logloss value of SDNN decreases by 0.4% compared to DNN. Comparing the ROC curves of DNN and SDNN, the AUC of DNN is 0.7092 and that of SDNN is 0.7207. The prediction accuracy of SDNN is higher than that of DNN, with an increase of 1.6% in AUC. This research result can accurately target users and save the operating costs of the advertising placement system, providing a new direction for advertising companies' marketing and effectively ensuring corporate profits. One of the future research directions is to integrate user social information into advertising CTR prediction models, enrich user information, and improve user expression information. However, due to equipment limitations and limited training data, the advantages of deep learning cannot be fully utilized. So in the future, the amount of training data will be increased so that the model can perform better.

References

[1] Wang Q, Liu F, Huang P, Xing, S., & Zhao, X. A Hierarchical Attention Model for CTR Prediction Based on User Interest. IEEE Systems Journal, 2020, 14(3):4015-4024.

[2] Erico C, Salim R, Suwarto S, & Isa, S. M. Mobile Advertisement Click through Rate Prediction. MATTER: International Journal of Science and Technology, 2020, 29(4s):1534-1540.

[3] Fan B, Fan W, Smith C, & Garner, H. S. Adverse drug event detection and extraction from open data: A deep learning approach. Information Processing & Management, 2020, 57(1):102131.1-102131.14.

[4] Lyu B, Hu Y, Zhang W, Du, Y., Luo, B., & Sun, X. Fusion Method Combining Ground-Level Observations with Chemical Transport Model Predictions Using an Ensemble Deep Learning Framework: Application in China to Estimate Spatiotemporally-Resolved PM_(2.5) Exposure Fields in 2014-2017. Environmental Science & Technology, 2019, 53(13):7306-7315.

[5] Feng C, Zhang H, Wang S, Li, Y., Wang, H., & Yan, F. Structural Damage Detection using Deep Convolutional Neural Network and Transfer Learning. KSCE journal of civil engineering, 2019, 23(10):4493-4502.

[6] Cai W, Wang Y, Ma J, & Jin, Q. CAN: Effective cross features by global attention mechanism and neural network for ad click prediction. Tsinghua Science and Technology, 2021, 27(1): 186-195.

[7] Xue N, Liu B, Guo H, Tang, R., Zhou, F., Zafeiriou, S., ... & Li, Z. Autohash: Learning higher-order feature interactions for deep ctr prediction. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(6): 2653-2666.

[8] Liu Y, Pang L, Lu X. Click-through Rate Prediction Based on Mobile Computing and Big Data Analysis. Ingénierie des Systèmes D Information, 2019, 24(3):313-319.

[9] Zhou L. Product advertising recommendation in e-commerce based on deep learning and distributed expression. Electronic Commerce Research, 2020, 20(2): 321-342.

[10] Ghorbel A, Souissi B. Upper confidence bound integrated genetic algorithm-optimized long short-term memory network for click-through rate prediction. Applied Stochastic Models in Business and Industry, 2022, 38(3):475-496.

[11] Liu Y, Yang T, Qi T. An Attention-Based User Preference Matching Network for Recommender System. IEEE Access, 2020, 8(99):41100-41107.

[12] Hung J L, Rice K, Kepka J, & Yang, J. Improving predictive power through deep learning analysis of K-12 online student behaviors and discussion board content. Information Discovery and Delivery, 2020, 48(4):199-212.

[13] Ma J, Shang P, Zou X, Ma, N., Ding, Y., & Su, Y. Remaining Useful Life Transfer Prediction and Cycle Life Test Optimization for Different Formula Li-ion Power Batteries Using a Robust Deep Learning Method. IFAC-PapersOnLine, 2020, 53(3):54-59.

[14] Ozmen E P, Ozcan T. A novel deep learning model based on convolutional neural networks for employee churn prediction. Journal of Forecasting, 2022, 41(3):539-550.

[15] Deng H, To A C. Topology optimization based on deep representation learning (DRL) for compliance and stress-constrained design. Computational Mechanics, 2020, 66(2):449-469.

[16] Hu B, Li J. An Edge Computing Framework for Powertrain Control System Optimization of Intelligent and Connected Vehicles based on Curiosity-driven Deep Reinforcement Learning. IEEE Transactions on Industrial Electronics, 2021, 68(8):7652-7661.

[17] He H, Jiang H. Deep Learning Based Energy Efficiency Optimization for Distributed Cooperative Spectrum Sensing. IEEE Wireless Communications, 2019, 26(3):32-39.

[18] Polvara R, Sharma S, Wan J, Manning, A., & Sutton, R. Autonomous Vehicular Landings on the Deck of an Unmanned Surface Vehicle using Deep Reinforcement Learning. Robotica, 2019, 37(11):1867-1882.

[19] Jafarzadehpour F, Molahosseini A S, Zarandi A, & Sousa, L. Efficient Modular Adder Designs Based on Thermometer and One-Hot Coding. IEEE transactions on very large scale integration (VLSI) systems, 2019, 27(9):2142-2155.

[20] Heo S K, Nam K J, Loy-Benitez J, Li, Q., Lee, S. C., & Yoo, C. K. A deep reinforcement learning-based autonomous ventilation control system for smart indoor air quality management in a subway station. Energy and Buildings, 2019, 202(Nov.):109440.1-109440.16.

[21] Wang X, Cheng M, Eaton J, et al. Fake node attacks on graph convolutional networks. Journal of Computational and Cognitive Engineering, 2022, 1(4): 165-173.

[22] Nimrah S, Saifullah S. Context-Free Word Importance Scores for Attacking Neural Networks. Journal of Computational and Cognitive Engineering, 2022, 1(4): 187-192.