Word difficulty prediction—based on GA-BP neural network and TOPSIS-GRA method

Keyang Wang*, Jialun Zheng, Yilei Xu

School of Data Sciences, Zhejiang University of Finance and Economics, Hangzhou, China, 310018

* Corresponding Author Email: 1447899019@qq.com

Abstract. The game WORDLE is very popular around the world, so our team tried to build a suitable model to analyze the difficulty of the words in the game. Firstly, this paper preprocesses the data to ensure the accuracy and integrity of the data. After extracting five representative characteristics, such as "letter frequency score", "word distance" and "syllable harmony", GA-BP neural network was selected to predict the percentage distribution of word attempts. The results show that compared with the traditional BP neural network, the GA-BP model with learning rate of 0.3, crossover rate of 0.7 and variation rate of 0.01 maintains a higher prediction accuracy, and the MSE value is significantly reduced. In addition, this paper combines GRA model and TOPSIS method, and uses entropy weight method to calculate the objective weight of indicators to comprehensively evaluate the difficulty of words. The example analysis shows that the improved model is more distinguishable and overcomes the defect of TOPSIS method that Euclidean distance can't distinguish ranking. At the same time, the ranking results of the two are close, which indicates the reliability and rationality of the difficulty prediction by using the model.

Keywords: GA-BP neural network, Word difficulty Prediction, TOPSIS-GRA method.

1. Introduction

WORDLE is a popular word guessing game that challenges players' logical reasoning skills. The player has to guess a word containing five letters. After each guess, the game provides feedback based on the correctness of the guessed letters and their positions in the word. Black indicates a correct letter in the right position, while gray represents a correct letter in the wrong position. By using this feedback to analyze and reason, players narrow down the possible word options until they guess the correct word.

As a universal puzzle game, the relationship between its intrinsic properties and solving difficulty has been the subject of much research. Experiments by Hans Stuyck's team confirmed that there is a strong correlation between the ease of solving crossword puzzles and certain factors. [1] In terms of model building, Erum Naz team for Word tie puzzle has tried to solve the word tiling problem with the help of Bee Colony algorithm in order to find the maximum number of words by moving the tiles up and down, left and right.[2] Goldblum, N. & Frost, R. (1988studied the internal structure of mental lexicon by means of crossword puzzles, found that syllabic units can be well used as retrieval tools, and proposed an interactive model.

However, none of these findings involved quantifying the difficulty of crossword puzzles. Therefore, this paper uses the NLP model to extract the features of words in the crossword puzzle, and uses the improved BP-GA model and TOPSIS-GRA method to quantitatively analyze the difficulty of the crossword puzzle WORDLE from different angles.

2. Materials and Methods

2.1. Data Source and Data Preprocessing

The data used in this paper is sourced from the New York Times and consists of 359 records representing words. Each record contains 7 features indicating the percentage distribution of word attempt counts. To ensure a normal distribution of attempt counts, three missing values in the dataset

are excluded. NLP is employed to extract features from the preprocessed data in order to quantitatively study word difficulty.

(1) Letter frequency scoring: This study aims to provide accurate estimates of English letter frequencies. In light of the strong correlation between the present results and estimates made by Morse over 160 years ago, the stability of English letter frequencies seems apparent. For each letter, the frequencies and percents estimated in the study is different. The frequency of each letter in the dataset is calculated as its score, reflecting its impact on word difficulty. The total score for letter frequency is obtained by summing the scores of individual letters in a word.

(2) Word Distance: The order of the alphabet may influence word guessing, as native English speakers are exposed to it during early language learning. The alphabetic distance between adjacent letters in a word is computed (e.g., A and C have a distance of 2), and the distances are summed to define "word distance".

(3) Part of speech: Words are categorized into nouns, verbs, adjectives, adverbs, and others, which have varying familiarity levels. Points are assigned based on daily-life familiarity to each category.

(4) Syllabic harmony degree: The "syllabic harmony degree" is introduced to measure the writing-pronunciation harmony in words. It is calculated as the number of syllables divided by the total number of letters. For example, the word "brisk" has one syllable and five letters, resulting in a "syllabic harmony degree" of 0.2.

(5) Number of non-repeating letters: The count of unique letters in a word is considered as an important feature affecting complexity.

2.2. The basic fundamental of GA-BP model

BP neural network is a mathematical model established by simulating the human brain nervous system, with strong learning ability and adaptability, but it requires a large number of parameters such as initial values of weights and thresholds. Therefore, we introduce genetic algorithms into the construction process of BP neural networks to determine the best performing parameters. This model can be abbreviated as the GA-BP model.

BP neural network model mainly consists of two steps: first, the input layer data is transmitted forward through the established neural network to obtain the prediction result; The second is to compare the predicted result with the real output, transfer the error to the input layer in reverse, use the activation function to constantly adjust the weight between neurons, and when the error is reduced to a reasonable range, the learning algorithm ends.

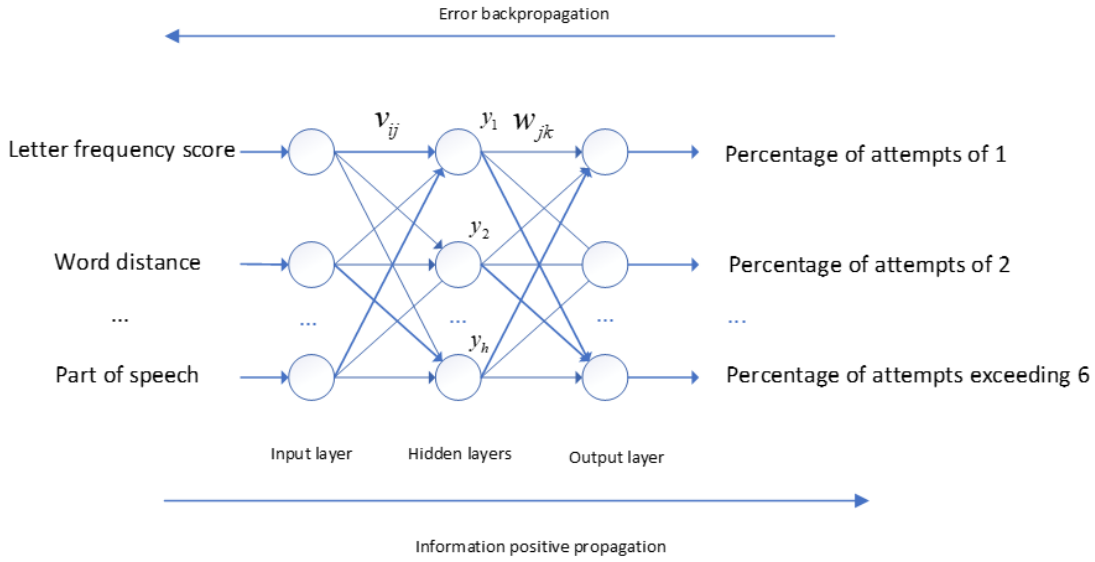

Figure 1. BP neural network topology

Figure 1 illustrates the topology of a BP neural network used for predicting the number of attempts. This neural network consists of three layers: the input layer, the hidden layer, and the output layer.

For general prediction tasks, a three-layer network can effectively address the problem. In this neural network, the input layer consists of various features extracted from the words, such as"Letter frequency score", "Word distance". The hidden layer, denoted as the output layer, represents the percentages of different attempt counts. The weight between the i-th neuron in the input layer and the h-th neuron in the hidden layer is represented as v_{ij}, and the threshold of the h-th neuron in the hidden layer is denoted as γ_h. The weight between the j-th neuron in the hidden layer and the k-th neuron in the output layer is denoted as w_{jk}, and the threshold of the k-th neuron in the output layer is denoted as θ_k. Let $\boldsymbol{x}$ be the input matrix and $y(k)$ be the final output value after k iterations, the forward propagation can be expressed as Equation (1).

$$y = f(w\boldsymbol{x} + \delta) \tag{1}$$

$$f = \frac{1}{1+e^{-x}} \tag{2}$$

In the equation, w represents the weight matrix for each neuron, δ represents the threshold matrix for each layer of neurons, and f denotes the activation function used to introduce nonlinearity, allowing the model to better approximate nonlinear functions for improved fitting performance. In this paper, the chosen activation function is the sigmoid function represented by Equation (2).

The BP neural network is an iterative process that involves calculating the residual between the output values y and the actual results at the output layer. It utilizes backpropagation to continuously update the weights and thresholds in each layer of the network. Let $E(i)$ denote the training error for the i-th sample, $d_k(i)$ represent the actual value of the k-th output in the i-th sample, and $y_k(i)$ represent the output value of the k-th output in the i-th sample. The training error for a single sample can be expressed as Equation (3), and the global error can be represented by Equation (4).

$$E(i) = \frac{1}{2}\sum_{k=1}^{n}\left(d_k(i) - y_k(i)\right)^2 \tag{3}$$

$$E = \frac{1}{p}\sum_{i=1}^{p} E(i) \tag{4}$$

If we denote the parameters that need to be updated in each iteration as v, the update formula can be represented by Equation (5)

$$v \leftarrow v + \Delta v \tag{5}$$

The above process represents the steps of the traditional backpropagation (BP) neural network, in which the global error E calculated using Equation (4) in each iteration is prone to getting stuck in local optima. However, the BP neural network improved by genetic algorithms utilizes E as the fitness function. After a series of operations including selection, crossover, and mutation, it can significantly enhance the issue of local optima and obtain high-quality solutions. The algorithm flowchart for the BP neural network improved by genetic algorithms is shown in Figure 2.

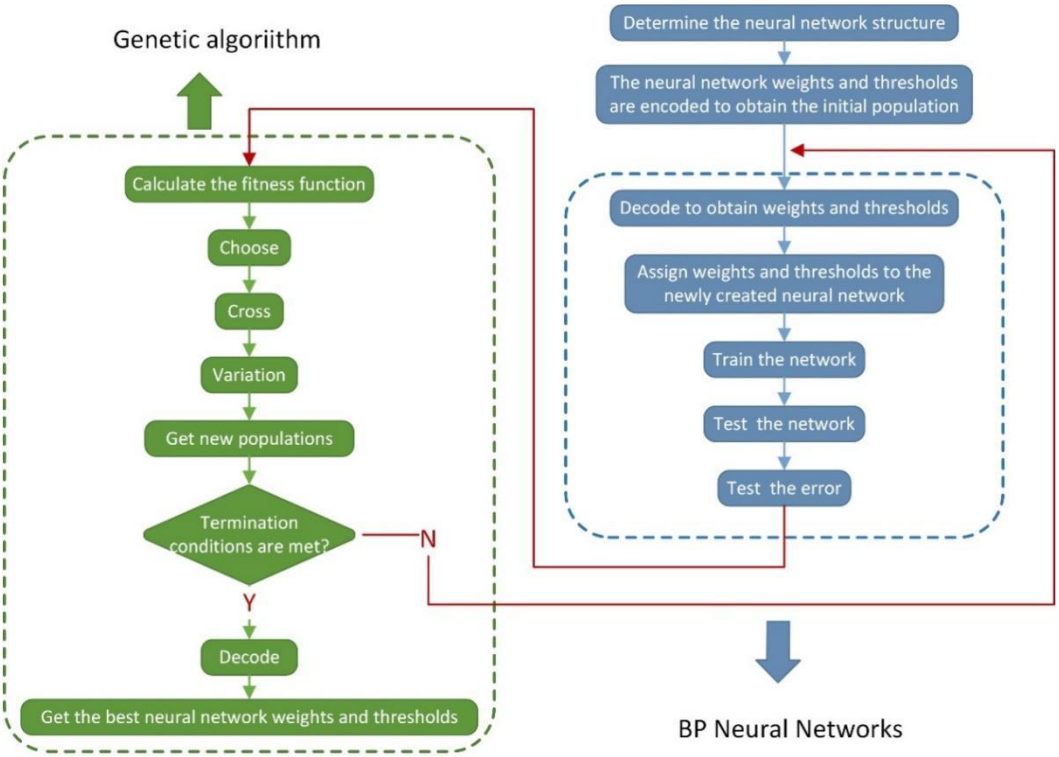

Figure 2. GA-BP neural network flowchart

2.3. The basic fundamental of TOPSIS-GRA method

2.3.1 TOPSIS method

TOPSIS method is a multi-attribute decision analysis method. The core idea is to build ideal solutions in evaluation objects, that is, the best or worst value of evaluation index values, calculate the Euclidean distance between each evaluation object and the ideal solution, obtain the relative closeness to the ideal solution, and rank the advantages and disadvantages of evaluation objects according to the relative closeness [3]. The specific calculation steps are as follows.

(1) Evaluation matrix standardization. The initial evaluation matrix $X = (x_{ij})_{m\times n}$ is formed by m evaluation objects and n evaluation indicators. Since all of them are extremely large indexes, the standard evaluation matrix $Y = (y_{ij})_{m\times n}$ is obtained from equation (6).

$$y_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} {x_{ij}}^2}}, i = 1,2,\dots,m; j = 1,2,\dots,n \tag{6}$$

(2) Construct weighted evaluation matrix. The weighted evaluation matrix $Z = (z_{ij})_{m\times n}$ is obtained by multiplying the evaluation matrix Y and the weight ω by equation (7).

$$z_{ij} = y_{ij} \times \omega_j, i = 1,2,\dots,m; j = 1,2,\dots,n \tag{7}$$

The calculation formula of ω is as follows.

$$p_{ij} = \frac{y_{ij}}{\sum_{i=1}^{m} y_{ij}}, j = 1,2,\dots,n \tag{8}$$

$$e_j = -\frac{1}{ln\, m}\sum_{i=1}^{m} ln\, p_{ij}, j = 1,2,\dots,n \tag{9}$$

$$\omega_j = \frac{1-e_j}{n-\sum_{j=1}^{n} e_j}, j = 1,2,\dots,n \tag{10}$$

(3) Calculate the distance between each scheme and the optimal solution and the worst solution. Let's call the optimal solution

$$z^+ = [z_1^+, \dots, z_m^+] = [max\{z_{11}, z_{21}, \dots, z_{m1}\}, \dots, max\{z_{1n}, z_{2n}, \dots, z_{mn}\}] \quad (11)$$

Then the distance between the scheme and the optimal solution can be calculated by equation (12). Similarly, the distance between each scheme and the worst solution can be obtained.

$$D_i^+ = \sqrt{\sum_{j=1}^{m}\left(z_j^+ - z_{ij}\right)^2} \quad (12)$$

(4) Calculate the final score of each scheme. The comprehensive difficulty score of each word is calculated by the formula (13). The higher the score, the more difficult the corresponding word is.

$$C_i = \frac{D_i^-}{D_i^- + D_i^+}, i = 1,2, \dots, m \quad (13)$$

2.3.2 TOPSIS-GRA method

However, when TOPSIS method is used, Euclidean distance will fail if there is a linear relationship between all indicators: in practical application, Euclidean distance is close to the positive and negative ideal solutions, and this method cannot comprehensively evaluate and sort the objects. In this paper, TOPSIS-GRA method is used to replace the Euclidean distance measure with the comprehensive relative proximity degree. The specific calculation steps are as follows.

(1) Calculate the grey correlation coefficient. Taking the positive and negative ideal solutions obtained as the reference sequence of grey correlation analysis, the grey correlation coefficients R_{ij}^+ and R_{ij}^- of each scheme and the positive and negative ideal solutions with respect to the j^{th} evaluation index are calculated by equation (14) and equation (15). Where ξ is the resolution coefficient, usually 0.5.

$$R_{ij}^+ = \frac{\min\limits_i \min\limits_j \left|z_j^+ - z_{ij}\right| + \xi \max\limits_i \max\limits_j \left|z_j^+ - z_{ij}\right|}{\left|z_j^+ - z_{ij}\right| + \xi \max\limits_i \max\limits_j \left|z_j^+ - z_{ij}\right|} \quad (14)$$

$$R_{ij}^- = \frac{\min\limits_i \min\limits_j \left|z_j^- - z_{ij}\right| + \xi \max\limits_i \max\limits_j \left|z_j^- - z_{ij}\right|}{\left|z_j^- - z_{ij}\right| + \xi \max\limits_i \max\limits_j \left|z_j^- - z_{ij}\right|} \quad (15)$$

(2) Calculate the grey correlation degree R_j^+ and R_j^- between each evaluation object and the ideal solution.

$$R_i^+ = \frac{1}{n}\sum_{j=1}^{n} R_{ij}^+, R_i^- = \frac{1}{n}\sum_{j=1}^{n} R_{ij}^-, i = 1,2, \dots, m \quad (16)$$

(3) Non-dimensional processing.

$$d_i^+ = \frac{D_i^+}{max\, D_i^+}, d_i^- = \frac{D_i^-}{max\, D_i^-}, r_i^+ = \frac{R_i^+}{max\, R_i^+}, r_i^- = \frac{R_i^-}{max\, R_i^-} \quad (17)$$

(4) Calculate the progress of comprehensive paste. In the formula, both α and β values are 0.5[4]. d_i^- and r_i^+ in formula (17) are fused with formula (18) according to a certain proportion to get the comprehensive paste progress S_i^+.

$$S_i^+ = \alpha d_i^- + \beta r_i^+, S_i^- = \alpha d_i^+ + \beta r_i^-, i = 1,2, \dots, m \quad (18)$$

(5) Calculate the comprehensive relative proximity degree. Equation (19) is used to calculate n_i. It reflects the similarity and difference of the relative position and shape of each evaluation object and ideal solution. The greater the relative progress of the synthesis, the higher the difficulty of the word synthesis.

$$n_i = \frac{S_i^+}{S_i^+ + S_i^-} \quad (19)$$

3. Model Construction and Solution

3.1. Construction and Solution of GA-BP neural network

After data preprocessing and preliminary determination of the topology structure of the BP neural network, it is necessary to further determine the number of neurons in the hidden layer of the model, as well as various parameters in the BP neural network and genetic algorithm.

Having too few neurons in the hidden layer can lead to insufficient fitting capacity of the neural network, thereby being unable to effectively extract features from the data. On the other hand, having too many neurons can reduce the efficiency of the program and potentially increase computational errors.

In a three-layer neural network, there is an approximate relationship between the number of neurons in the hidden layer n_2 and the number of neurons in the input layer n_1, which can be represented by Equation (20).

$$n_2 = 2 * n_1 + 1 \tag{20}$$

In this paper, since 5 features extracted from words are used as inputs, the model contains 5 inputs and 7 outputs. The calculated number of neurons in the hidden layer is 11. Therefore, the final structure of the BP neural network is set as 5-11-7, which means the input layer contains 5 neurons, the hidden layer contains 11 neurons, and the output layer contains 7 neurons.

In the BP neural network, the most important parameter is the learning rateη, which affects the update step size in each iteration. If η is set too small, it will slow down the convergence speed. If η is set too large, it may cause oscillations and hinder finding the optimal solution. After repeated testing of the model, the value of η is finally set to 0.3.

In the genetic algorithm, the parameters to be set include the crossover rate and mutation rate. Both the crossover rate and mutation rate affect the iteration speed of the model. If their values are set too low, it may lead to getting trapped in local optima. If their values are set too high, it may make the algorithm difficult to converge. Moreover, the selection of these two rates is not independent. The mutation rate should be appropriately reduced as the crossover rate increases, and the recommended range for the mutation rate should be further narrowed down to 0.001 to 0.1. After multiple attempts, the crossover rate is finally set to 0.7, and the mutation rate is set to 0.01.

3.2. Construction and Solution of TOPSIS-GRA method

This paper selects the data corresponding to the last 8 words in the preprocessed data set to verify the validity of the model. Using the type (1) ~ (4) to calculate the word try different number of the corresponding objective weight for ω = (0.2731, 0.0556, 0.1270, 0.2699, 0.0027, 0.1427, 0.1313). ω was substituted into relative equation to calculate the weighted evaluation matrix, and then TOPSIS-GRA method was used to calculate the comprehensive paste progress of each word. The calculation process is shown in the table 1.

Table 1. calculation process

word	D_i^+	D_i^-	R_i^+	R_i^-	S_i^+	S_i^-	n_i	order
angry	0.1357	0.1897	0.8336	0.6157	1.0000	0.7345	0.5736	1
abbey	0.1743	0.1198	0.7115	0.7682	0.7046	0.9605	0.4186	8
favor	0.1622	0.1511	0.6683	0.7236	0.7452	0.8545	0.4674	5
drink	0.1534	0.1633	0.7347	0.6814	0.8423	0.7856	0.5203	2
query	0.1742	0.1832	0.6959	0.7342	0.8345	0.8956	0.5143	3
gorge	0.1745	0.1754	0.6982	0.7023	0.8124	0.8977	0.5034	4
crank	0.1748	0.1721	0.6845	0.7486	0.8023	0.9317	0.4597	6
slump	0.2053	0.1299	0.6701	0.7984	0.7898	1.0000	0.4296	7

4. Results and analysis

4.1. Results and analysis of GA-BP neural network

To evaluate the performance of the GA-BP model in predicting the number of attempts, this paper selects two indicators, namely Mean Squared Error MSE and Prediction Accuracy J, to make judgments. Their calculation formulas are given by equations (21) to (23).

$$MSE = \frac{1}{n}(\hat{y}_i - y_i)^2 \tag{21}$$

$$J = \frac{1}{n}\sum_{i=1}^{n} H_i \times 100\% \tag{22}$$

$$H_i = \begin{cases} 1, |\hat{y}_i - y_i| < 0.05 \\ 0, \hat{y}_i - y_i \geq 0.05 \end{cases} \tag{23}$$

y_i represents the true value of the i-th sample, and $\hat{y}_i$ represents the predicted value of the i-th sample. MSE represents the deviation between the true value and the predicted value, and a smaller value indicates that the model's predictions are closer to the true values. J represents the proportion of correctly predicted samples within the allowed error range, and a value closer to 1 indicates better predictive performance of the model.

The MSE versus the number of words, as shown in Figure 3, demonstrates that the MSE values of the GA-BP model decrease continuously with the increase in the number of words, eventually converging to a low level. On the other hand, although the initial value of the prediction accuracy rate J is relatively low, it quickly stabilizes at a high level of 98%. These results indicate that the constructed GA-BP model exhibits good predictive performance. Furthermore, when comparing the MSE values between the traditional BP neural network and the GA-BP model, it is evident that the latter consistently achieves significantly lower MSE values. This indicates that the optimized GA-BP model has greatly improved performance compared to the traditional BP model.

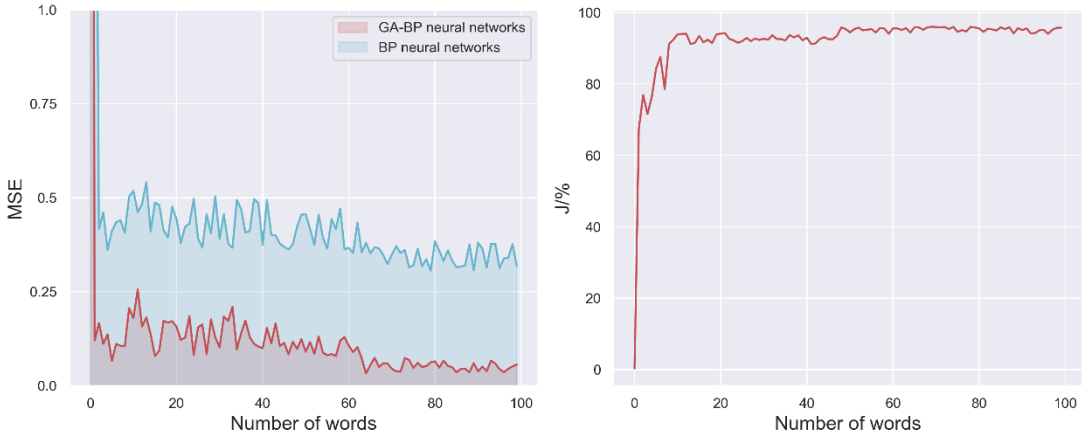

Figure 3. MSE and Accuracy Variation with Word Count for the GA-BP Model

4.2. Results and analysis of TOPSIS-GRA method

In order to further verify the validity and rationality of the model, TOPSIS-GRA was used for evaluation and the traditional TOPSIS method was used for verification. The objective weights of both were calculated by EWM, and the results are shown in Table2.

Table 2. the ranking results of different models

word	tangy	abbey	favor	drink	query	gorge	crank	slump
TOPSIS-GRA	1	8	5	2	3	4	6	7
TOPSIS	1	8	4	3	2	5	6	7

As can be seen from the results in Table 2, the prediction results of the improved TOPSIS model after GRA are basically consistent with those of the traditional TOPSIS model, which means that the TOPSIS-GRA model has good stability. In addition, it can be seen from the results in Table 1 that the traditional TOPSIS method has the same situation in the positive and negative ideal solution of the European distance, for example, the D_i^+ and D_i^- of the word gorge are both 0.1745, which will lead to distortion of the sorting result, but the improved model effectively avoids this situation, and the sorting result is more distinguished, making the result more convincing, and the application of the model is improved.

5. Conclusions

5.1. Strengths

(1) By incorporating genetic algorithm into the BP neural network, the GA-BP model achieves higher accuracy in predicting word difficulty compared to the traditional BP neural network.

(2) In the selection of genetic algorithm parameters, the correlation between crossover rate and mutation rate is considered, and parameters are chosen from a more precise range. This significantly enhances the optimization effect of the genetic algorithm on the BP neural network.

(3) The comprehensive relative proximity degree of grey correlation TOPSIS method is used to replace the Euclidean distance measure, which solves the problem that the evaluation object can't be sorted when the evaluation object is close to the ideal solution in the traditional TOPSIS comprehensive evaluation method, and makes the evaluation result more scientific and reasonable.

5.2. Weaknesses and further improvements

(1) The performance of the GA-BP model is somewhat limited by the small size of the dataset used for model construction. Therefore, it is recommended to increase the collection of relevant data and expand the existing dataset appropriately to enhance the predictive capability of the model.

(2) The iteration count of the GA-BP neural network is still relatively low. Therefore, it is advisable to increase the number of iterations to improve the predictive performance of the model.

References

[1] Stuyck, H., Aben, B., Cleeremans, A., & Van den Bussche, E. (2021). The Aha! moment: Is insight a different form of problem solving? Consciousness and Cognition, 90, 1030-552.

[2] Naz, E., Al-Dabbas, K., Abrishami, M., Mehnen, L., & Cvetkovic, M. (Publication year not mentioned). Solving Word Tile Puzzle using Bee Colony Algorithm. International Journal of Advanced Computer Science and ApplicationsMa Kunlong. Short term distributed load forecasting method based on big data [D]. Changsha: Hunan University, 2014.

[3] XIAO Jichuan, XING Ying. Vulnerability Evaluation Method for Metro Station Operation Based on Entropy Weight TOPSIS Model [J]. Journal of Transportation Engineering and Information, 2020, 18(2): 163-169.

[4] ZHANG Haitao, LI Zezhong, LIU Yan, et al. Comprehensive Evaluation of Value Flow in Business Network Information Ecochain Based on Combination Weighted Grey Correlation Grade TOPSIS Model [J]. Information Science, 2019, 37(12): 150-158.

[5] Heng, J., Wang, C., Zhao, X., & Wang, J. (2016). A Hybrid Forecasting Model Based on Empirical Mode Decomposition and the Cuckoo Search Algorithm: A Case Study for Power Load. Mathematical Problems in Engineering, 2016, 1-28.

[6] Yuen, J. K. K., & Lee, E. W. M. (2012). The effect of overtaking behavior on unidirectional pedestrian flow. Safety Science, 50(8), 1704-1714.

[7] WANG Lifang. Classification Evaluation of Urgency of Disaster Point Emergency Material Demand Based on Combination Weighting and Improved TOPSIS Method [J]. Safety and Environmental Engineering, 2017, 24(6): 94-100.

[8] JIANG Fucai, ZHOU Congcong, MA Quandang. Risk Assessment for the Inland River Pilotage Based on the Set Pair Analysis Model Via Integrated Bestow [J]. Journal of Safety and Environment, 2021, 21(3): 990-996.

[9] Underwood, G., Deihim, C., & Batt, V. (1994). Expert performance in solving word puzzles: From retrieval cues to crossword clues. Applied Cognitive Psychology, 8(6), 531-548.

[10] Min Z. A study of the number of Wordle users and experience predictions [J]. Academic Journal of Mathematical Sciences, 2023, 4(2).