Robust Design of Machine Translation System Based on Convolutional Neural Network

Pei Pei[1(✉)] and Jun Ren[2]

[1] Department of Foreign Languages, Changchun University of Finance and Economics, Changchun 130200, China
peip414@163.com
[2] North Automatic Control Technology Institute, Taiyuan 030000, China

Abstract. Aiming at the problem of low load robustness coefficient and recovery robustness coefficient of machine translation system in different scenarios and working conditions, which leads to poor robustness, convolutional neural network algorithm is used to optimize the robustness of machine translation system. The operation process of the machine translation system is simulated through the steps of corpus preprocessing according to the composition and working principle of the machine translation system, word alignment processing, and phrase extraction. Obtain the load data of the machine translation system, and with the support of building a convolutional neural network model, according to the measurement results of the vulnerability of the machine translation system, use the convolutional neural network algorithm to determine the system load scheduling amount. The robust controller is selected as the executive element to complete the robust design of the machine translation system. The experimental results show that the machine translation system designed by the optimization method has higher load robustness coefficient and recovery robustness coefficient under different scenarios and operating conditions, which confirms that the robustness design effect of the optimized machine translation system is better.

Keywords: Convolutional neural Network · Machine Translation System · Robust Design · Decoding

1 Introduction

Machine translation [1, 2] is the process of converting one natural language into another by using computers, also known as automatic translation. Translation is the main method of equivalent information transfer between different languages. The traditional manual translation is inefficient, so intelligent translation is used instead of manual translation. The incompleteness of the current translation results indicates that there is still much room for research in this field. However, deep learning has good advantages for speech recognition, so deep learning [3] is introduced to study translation.

At present, there are many kinds of machine translation systems, which have been well applied in teaching and international communication, and they have good development prospects. However, the system operation is unstable in the practical application

B. Wang et al. (Eds.): ICMTEL 2023, LNICST 532, pp. 100–113, 2024.
https://doi.org/10.1007/978-3-031-50571-3_8

of machine translation system due to the low robustness of the system. The robustness of the machine translation system is designed according to its working principle in order to improve the running robustness of the machine translation system and thus improve the running stability.

Robustness plays an important role in the system. At present, the research on system robustness has gradually become a hot topic. For example, literature [4] studied a method to quantify the robustness of chaotic systems, and proposes a scheme to determine the degree to which system parameters can be changed before the probability of destroying chaos exceeds 50%. The Monte Carlo method is used in the calculation and is applied to several common dissipative chaotic maps and flows with different number parameters. This method has a good effect in practical application, but its performance of restoring robustness coefficient is poor. Literature [5] studied the multi-stage robust scheduling method of this system, which proposed a new proposition. In view of the uncertainty of load and renewable output, unpredictability and robustness were taken into account, and the gradient scheduling model is established. However, the method has the problem of low load robustness coefficient. Therefore, convolutional neural network algorithm is introduced.

Due to the characteristics of convolutional neural networks, this paper applies them to optimize the robustness of machine translation systems to improve the robustness.

2 Robustness Design of Machine Translation System

2.1 Simulate the Running Process of the Machine Translation System

Machine translation is divided into three parts, and the overall structure is shown in Fig. 1.

Model training includes basic processing steps and training language models and sequencing models. The parameter feature tuning stage uses the existing bilingual corpus to continuously update the feature weights of the model to make it optimal. The translation and decoding module translates the source sentence through the model and feature weights learned during model training and parameter feature tuning. The operation process of the machine translation system is shown in Fig. 2.

The machine translation model can be directly mapped into the output sequence in an end-to-end manner. If the source language is given as $A(t)$ and the target language is set as $B(t)$, the conditional probability of the machine translation description is shown in Formula 1.

$$P(B|A) = \prod_{t=1}^{m} P(b_t|b < t, A) \tag{1}$$

Variable m in Eq. 1 is the number of source languages. The encoder encodes it into a fixed-length low-dimensional real number vector as the semantic representation of the sentence for each input source language target sentence, and the encoder encodes it under the condition of the given source language target sentence semantic representation, the translation results are word for word sentences in the target language.

Corpus Preprocessing

It is necessary to convert the corpus into a regular form that meets the model input

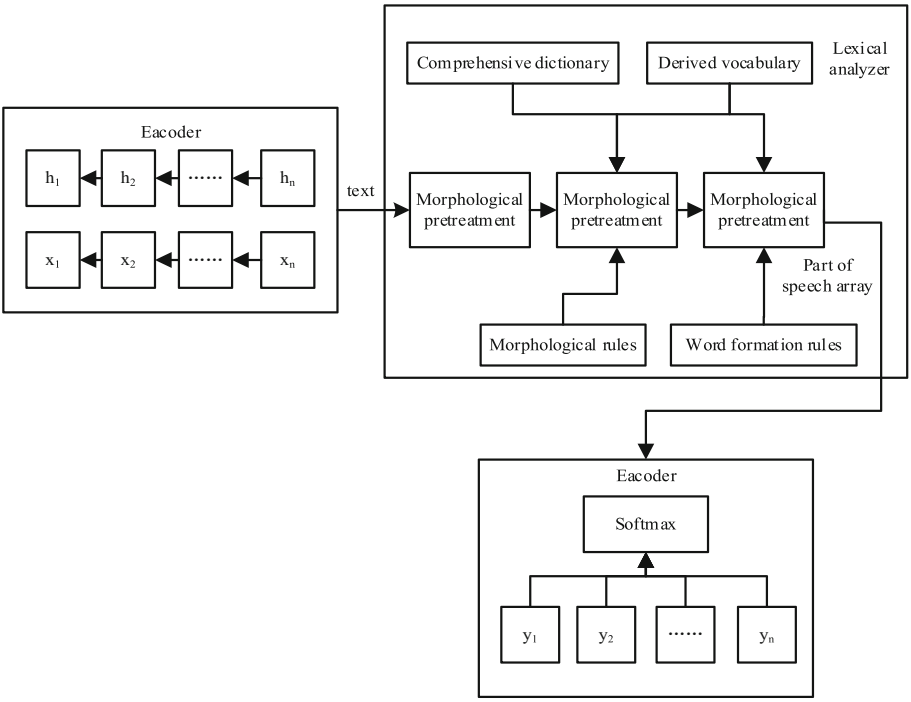

Fig. 1. Block diagram of the machine translation system

requirements before the machine translation system runs the translation program [6, 7]. The irregular corpus is preprocessed, including word segmentation and special words. The word segmentation process of the initial translated words is shown in Fig. 3.

The content of special vocabulary processing mainly includes: all uppercase letters are changed to lowercase letters; because the word segmentation of English data only needs to separate words from punctuation marks by spaces, so when processing, it is only necessary to separate the terminator at the end of the sentence on the original data. Yes; the named entity recognition part is in Chinese and English, so it also needs to be processed accordingly, that is generalization of special nouns.

Word Alignment Processing

Word alignment is a basic and important step in the process of machine translation. It is an essential link in many research tasks that use parallel corpora, such as question answering system, word sense disambiguation, lexicography, machine translation, etc. Generally speaking, alignment can be divided into several forms at different levels, such as text, paragraph, sentence, phrase and vocabulary. However, the goal of this step is to find language fragments that can be translated from parallel corpora [8]. The method of word alignment based on dictionary is based on the fact that the dictionary contains high quality information of word translation. The fully dictionary-based method has good accuracy in the case of non-empty word alignment, but the context is diverse in actual translation scenarios, so the translation is flexible and the coverage of dictionary

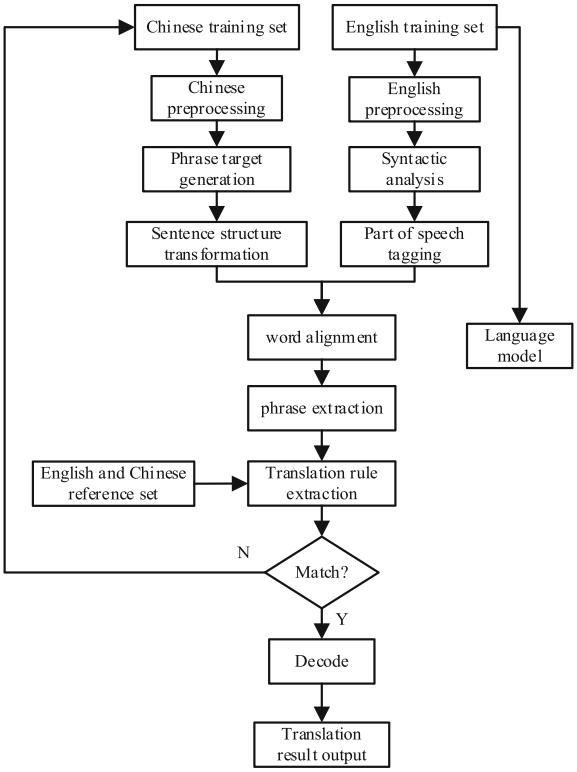

Fig. 2. Flowchart of the operation of the machine translation system

translations is relatively low. Therefore, if only dictionary-based word alignment cannot meet the requirements, other methods need to be introduced. Some words can achieve a certain degree of matching, but they cannot meet the matching requirements 100%. Therefore, a fuzzy mechanism is used to match words. The matching process of any two words can be expressed as:

$$\begin{cases} s(E, C) = \phi + \varphi \\ \phi = \max(s(d, c)) \\ \varphi = (Count(s(d, c) > \eta) - 1) \times 0.1 \end{cases} \tag{2}$$

where: $s(d, c)$ is the matching degree of words d and c, max() and $Count()$ are the maximum value calculation function and the number statistics function respectively, and η is the matching degree threshold of words. The specific calculation formula of variable $s(d, c)$ can be expressed as:

$$s(d, c) = \frac{2 * (d \cap c)}{|d| + |c|} \tag{3}$$

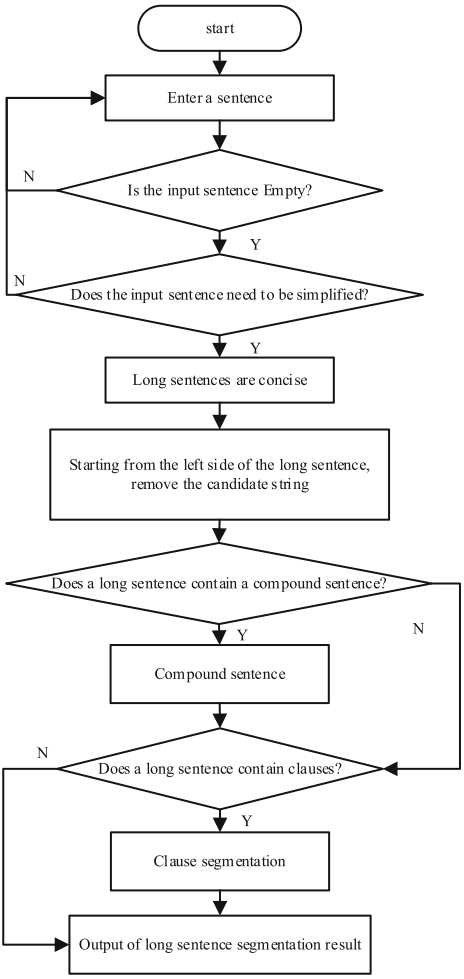

Fig. 3. Process flow chart of machine translation word segmentation

Substitute the calculation result of Eq. 3 into Eq. 2 to complete the alignment of the source translation words of the machine translation system.

Phrase Extraction

Phrase extraction is an important step. It is the difference between phrase based method and word based method. The decoding part can be successfully implemented only by extracting phrases according to corresponding rules and constructing translation tables. The target is to search each phrase in the source sentence and its corresponding target phrase based on the results of word alignment in the process of phrase extraction. First, the phrase is determined to be processed from the source language, then search for the corresponding relationship between each word that composes it and the target word, and finally determine whether the relationship is satisfied between the target phrase and

the source phrase, and expand the corresponding relationship. Empty word. Extracting phrases can only be regarded as a basic step. All phrase pairs cannot be treated equally, because the rationality of different phrase pairs is different. Therefore, the rationality of phrases needs to be scored. Phrase pair scores are calculated as follows:

$$F_r(f, e) = \frac{Count(F_r(e, f))}{\sum_{f_l} Count(F_r(e_l, f))} \tag{4}$$

where: e and f are phrases in the target language and in the source language respectively. When the training data is not sufficient, some low-frequency phrase pairs may appear although the phrase translation probability can represent the corresponding relationship score of the phrase level. If the frequency of occurrence of both is small and the possibility of co-occurrence is high, the phrase score is too high. Therefore, the combined decision of lexicalized weighted information is supplemented in order to deal with this situation better.

Machine Translation Result Output

Decoding is to traverse the set space of solutions, obtain suitable statements, and output them as translation results. Traversal is a search process that uses a heuristic depth-first search algorithm. The search behavior depends on the calculation results of the different modules, which determine the next direction until the translation terminator is generated, ending the search. Statistics-based machine translation calculates a probability. Given the source sentence to be input, the translation model will produce multiple translation results [9]. The result score is obtained through evaluation, and the translation result with high score is selected as the output result of the final model.

2.2 Obtaining Machine Translation System Load Data

Obtain the number of translation tasks executed by the machine translation system at any time as the system load data. When the server detects the load, it uses the load prediction method to determine whether the server is overloaded through load prediction, which can avoid errors caused by instantaneous high load. When the server agent detects that k times in the past n times exceed the load threshold, the load prediction algorithm is triggered. The algorithm is based on time-series forecasting, so the load values detected at the past t moments are calculated in time order. If the detected load value sequence is:

$$L(t) = \{l_1, l_2, \cdots, l_t\} \tag{5}$$

The n-order autoregressive model is used to predict the load value at time $t + 1$, then the load prediction value at time $t + 1$ is calculated as follows:

$$l_{t+1} = \tau_1 l_t + \tau_2 l_{t-1} + \cdots + \tau_n l_{t-n+1} + \delta \tag{6}$$

where: τ_1 is the time series parameter, and δ is the noise disturbance coefficient of the machine translation system. When a certain resource exceeds the load threshold for k consecutive times, the prediction model is triggered to predict the load at time $k + 1$. If

the load still exceeds the threshold, virtual machine migration is triggered. The server agent selects the virtual machine with a large proportion of the resource to be migrated according to the type of overloaded resource. When the agent manager detects that multiple resources exceed the load threshold at the same time, it needs to set the weight according to the proportion of various resources of the server, it is converted into a comprehensive load index, and obtain the real-time load data acquisition result of the machine translation system.

2.3 Building a Convolutional Neural Network Model

Neuron is the basic unit of convolutional neural network, and also the basic unit of information processing domain in the network. The neuron is equivalent to a many to one nonlinear mapping in the mathematical field, and the neuron model constituting the CNN is shown in Fig. 4.

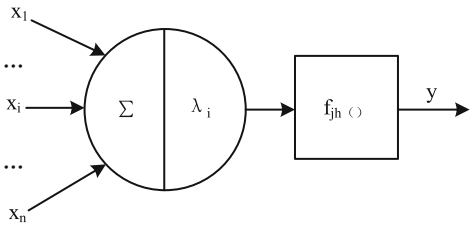

Fig. 4. Schematic diagram of neuron model

In Fig. 4, variable x_j is the input value, x_j is the threshold value, and $f_{jh}()$ and y_i represent the transfer function and the output value, respectively. A CNN model is formed through the connection of neuron models, and its topology is shown in Fig. 5.

As can be seen from Fig. 5, it is a three-layer network, and CNN is a feedforward network, which can process information cooperatively and store information distributed in parallel. The hidden layer is the middle layer of CNN, which contains convolution operation, its role is to extract features, input and output layers, while the hidden layer adjusts weights, which is a "self-organizing process". The output of any neuron i in the convolutional neural network can be expressed as:

$$y_i = \text{sgn}\left(\sum_{i=1}^{n_{net}} \omega_{ji} x_j - \lambda_i\right) \tag{7}$$

where: ω_{ji} is the weight value between neurons i and j, and sgn() is the step function or the sign function. The Sigmoid function is used as the activation function of the neuron, and its function expression is:

$$f_{jh}(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

Determine the number of neurons in each layer of the CNN and the weight between each layer, and complete the construction of the CNN model.

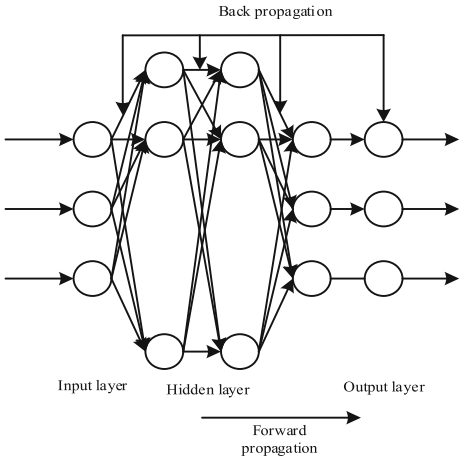

Fig. 5. Convolutional neural network topology diagram

2.4 Measuring Machine Translation System Vulnerability

The load instability coefficient of the machine translation system is set as the vulnerability measurement index to measure the vulnerability of the machine translation system. The calculation formula of the system load instability coefficient index is as follows:

$$\kappa_{\text{Instability}} = \frac{\sum_{i=1}^{n_{\text{thread}}} (L_i - L_{i-1})}{L_{avg}} \tag{9}$$

where: L_i and L_{i-1} are the load values of the i and $i-1$ threads in the system, n_{thread} is the total number of sites running in the system, and L_{avg} is the average value of the system load. The quantitative measurement results of the vulnerability of the machine translation system can be obtained through the acquisition of relevant market protectors and the calculation of Formula 9.

2.5 Use Convolutional Neural Network Algorithm to Determine System Load Scheduling Amount

Substitute the obtained machine translation system operating data and the system vulnerability result obtained from the measurement into the constructed CNN model, and determine the load scheduling amount of the machine translation system through the operation of the CNN algorithm. Figure 6 shows the execution principle of the CNN algorithm.

The output results of each level of the convolutional neural network model are obtained after the calculation from the front to the back, which as follows:

$$\begin{cases} x_{\text{Implied}} = f_{\text{jh}}\left(\omega_{\text{Implied}} \cdot x_{\text{in}}\right) \\ x_{out} = f_{\text{jh}}\left(\omega_{out} \cdot x_{\text{Implied}}\right) \end{cases} \tag{10}$$

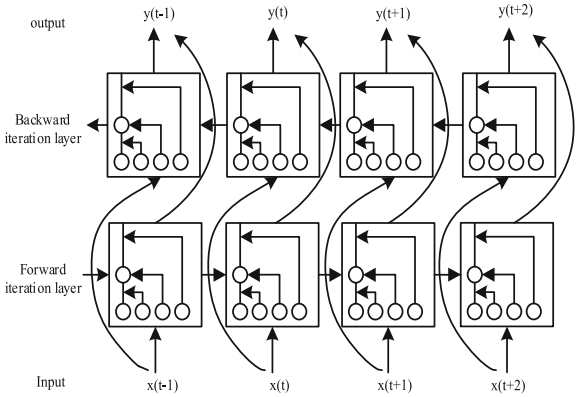

Fig. 6. Operating principle diagram of the convolutional neural network algorithm

where: x_{in}, x_{Implied} and x_{out} are the output results of the input layer, hidden layer and output layer of the convolutional neural network, respectively. The corrections of the weights of each layer can be expressed in the forward propagation process, which as follows:

$$\Delta\omega_{1j}(n) = \kappa_{\text{con}}\kappa_0 x_{out} + \kappa_{\text{inertia}}\Delta\omega_{1i}(n-1) \tag{11}$$

where: κ_{con}, κ_0 and κ_{inertia} are learning rate, convergence coefficient and inertia coefficient respectively. Then the weight correction process during the operation of the convolutional neural network algorithm can be expressed as:

$$\omega_{ij}(n+1) = \omega_{ij}(n) + \Delta\omega_{ij}(n) \tag{12}$$

The correction of the weight value can be completed by combining Formula 11 and Formula 12. Repeat the above work until the termination condition is met. The determined value of the load scheduling amount of the system is output through the loop iteration of the convolutional neural network algorithm, and the specific value can be expressed as:

$$\Delta L_i = \left|L_i - L_{avg}\right| \tag{13}$$

Thus, the load scheduling of MT system is solved.

2.6 Implement Robust Design of Machine Translation System

Systems generally have defects [10]. Therefore, a robust controller is installed in the machine translation system as an executive element for robust design in order to further improve the performance of machine translation systems. The structure of the robust controller is shown in Fig. 7.

Where: $r(t)$ is the input of the nonlinear system, $y_l(t)$ is the output of the system, and $U(t)$ is the pseudo-system input containing the u(t − 1) polynomial. The real control input $u(t-1)$ can be obtained by solving the polynomial method, such as the Newton

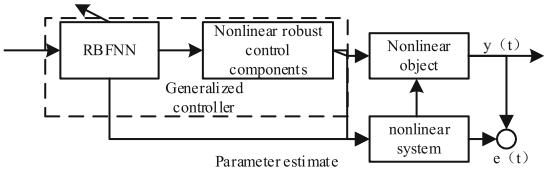

Fig. 7. Structure diagram of robust controller of machine translation system

iteration algorithm. Among them, RBFNN is defined as the local controller; the local controller and the Newton iterative Φ(t) solver are collectively defined as the generalized convolutional neural network controller [10]. Make full use of the framework of the U model to estimate unknown parameters online. The wide area controller adjusts the input $u(t-1)$ of the system with the obtained estimated parameters. This is a new type of control structure and identification structure based on the U model, and many similar convolutional neural networks use this standard control. The robustness of machine translation system is realized with its support.

3 Experiment Analysis of Robust Design Effect Test

A number of different types of systems are selected as the research objects in order to test the design effect of the robust design method of machine translation system based on convolutional neural network, and the stability of the analysis system is tested under different operating environments and operating conditions.

3.1 Configure the Research Object of Machine Translation System

The machine translation system based on phrase statistics in the experiment, the machine translation system based on microengine pipeline and the machine translation system based on paraphrase information are selected as the research objects of the experiment. The selected machine translation system uses Intel(R) Xeon(R) CPU E5-2660 V4 @2.00 GHZ as the processor, Tesla K80 as the graphics card, Python(3.5.3); pytorch(0.4.1) as the third-party library In the experimental environment, the development of the system was completed. Prepare the translation sample and input it into the developed system to obtain the corresponding machine translation result. The operating interface of the translation system is shown in Fig. 8.

Using Mteval_sbp as a test tool, data was collected via the website. The training corpus is used to collect 2 million sentences in a laboratory. The benchmark translation model takes the gated convolutional neural network LSTM as the base through the encoder and decoder, and the attention mechanism is used in the connection between the decoder and the encoder.

3.2 Input the Operating Parameters of the Convolutional Neural Network Algorithm

The running parameters of the CNN are set from the initial value of the network's learning rate, the number of neural nodes in each layer, the number of hidden layers in

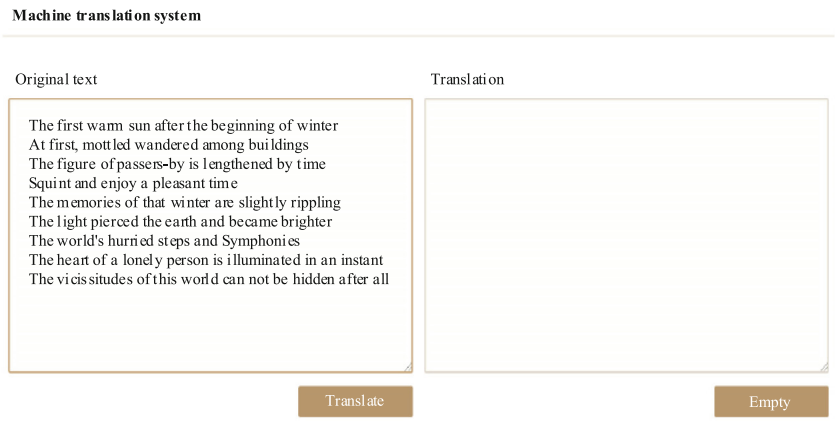

Fig. 8. Machine Translation System Operation Interface

the network, etc. The single-layer CNN has high error and low identification accuracy. The number of hidden layers can improve identification accuracy is increased, error is reduced and error accuracy is improved. However, while it has advantages, it also has disadvantages, which will make the network overly complicated, because it will increase the training time of the network connection weight coefficient. If you only increase the error accuracy, you can achieve this requirement by increasing, which will make it easier to observe and adjust the training effect of the convolutional neural network. Therefore, the first consideration should be to increase the number of nodes in the hidden layer in general, rather than the first consideration to increase the number of hidden layers. The number of nodes in the hidden layer in the CNN is expressed as:

$$n_y = \sqrt{0.43vg + 0.12g^2 + 2.54v + 0.77g + 0.35} \tag{14}$$

where: v and g are the number of nodes in the input layer and output layer respectively. An initial value is obtained through the above calculation, and then the step-by-step pruning method is used, that is starting from a relatively complex convolutional neural network, the hidden layer nodes are deleted step by step until the network reaches the best. If not, increase the number of nodes in the hidden layer. The learning rate can determine the change of the connection weight coefficient in each cycle. When the value of the learning rate is too large, the system will run erratically, and even the system will be paralyzed; if the value of the learning rate is too small, the training time of the system will definitely be longer, resulting in a slower convergence rate of the system. Although the learning rate is too small, it will have a certain negative impact on the system, but the learning rate can make the system avoid getting trapped in the local convergence of the error function, and the system eventually is made to tend to the minimum error. Therefore, a small learning rate is usually adopted in order to ensure the stability of the convolutional neural network.

3.3 Set the Test Index of Robust Design Effect

The load robustness coefficient and recovery robustness coefficient are respectively set as quantitative test indicators in the experiment, and the numerical results can be expressed as:

$$\begin{cases} \mu_{load} = \dfrac{L_{\max}}{(L_i - L_j)} \\ \mu_{\text{recovery}} = 1 - \left[\dfrac{(R_{\text{f}} - R_{\text{d}})}{R} \right] \end{cases} \tag{15}$$

where: L_i and L_j are the load values of the i and j threads in the machine translation system, respectively; $L_{\max}$ is the maximum thread load of the system; R_{f}, R_{d}, and R are the number of tasks interrupted by the system, and the number of tasks recovered after the interruption. The number of translation tasks and the total number of translation tasks entered in the system. The higher the load robustness factor and the recovery robustness factor of the system, the better the robustness of the system design.

3.4 Experimental Test Process and Result Analysis

The traditional method based on the direct state space theory and the robust design method based on the disturbance observer are set as the comparison methods of the experiment in the system test experiment. The robustness of the machine translation system is optimized by the design method, and the robustness design is completed. This experiment tests the robustness of different systems from two aspects: the running scenarios and operating conditions of the machine translation system, machine translation tasks is ran under different attack environments, counts the running data of the system, and obtains the evaluation test results of the system robustness under different operating scenarios through the calculation of Formula 15, as shown in Fig. 9.

It can be seen intuitively from Fig. 9 that the higher the attack intensity in the system operating environment, the lower the system robustness coefficient. The robust design method of the convolutional neural network-based machine translation system applying the optimal design has higher load robustness coefficients and recovery robustness coefficients is compared with the traditional robust design method, that is the robustness design effect is better. This is because the load instability coefficient of the machine translation system is set as the vulnerability measurement index to measure the vulnerability of the machine translation system. The final test results are shown in Fig. 10.

It can be seen from Fig. 10 that the two are negatively correlated. Through vertical comparison, it is found that the average value of the MT system load robustness coefficient of the comparison method is 0.37 and 0.28, the average value of the MT system load robustness coefficient of the optimization design method is 0.66, and the value of the optimization design is 0.29 and 0.38 higher than that of the comparison method, This proves that the robust design method of machine translation system based on convolutional neural network has better robustness design effect. This is because the neural network algorithm is introduced into the system, through which the load scheduling amount of the system is determined, and the robust controller is designed. The robustness is improved through the combination of various aspects.

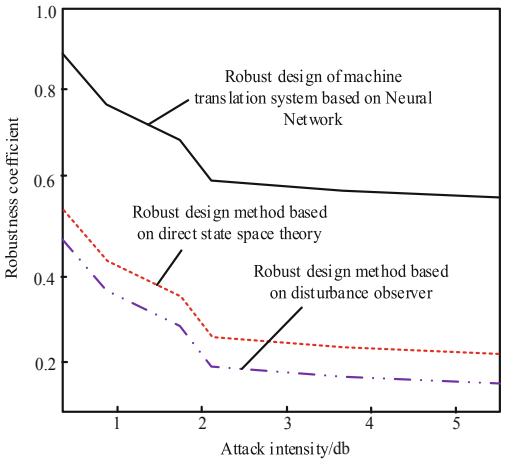

Fig. 9. System robustness test results under different scenarios

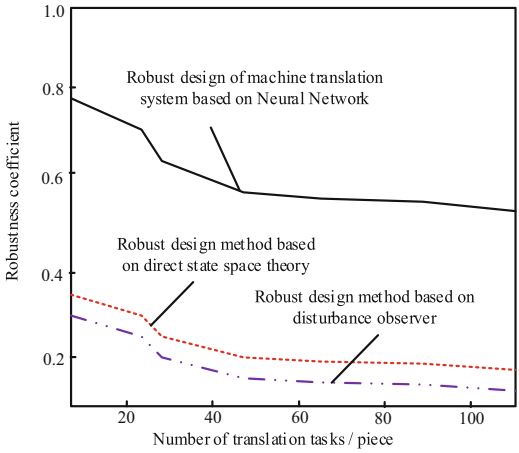

Fig. 10. System robustness test results under different working conditions

4 Conclusion

The machine translation system realizes the intelligent transformation of translation work. This technology is used in translation work can innovate the retrieval technology of cross-language information in my country, and it has industrial application value and social value. The operation stability of the machine translation system in different scenarios and operating conditions is effectively improved through the design and application of the robustness design method of the machine translation system based on the convolutional neural network, and the application value of the machine translation system in the translation work is improved.

Acknowledgement. Social Science Research Planning Project (JJKH20221258SK) of Jilin Provincial Department of Education: A study on improving foreign language skills of the language services industry in Jilin Province.

References

1. Zou, D.F., Hu, Q.B.: Construction of neural machine translation model based on tree-to-string model enhancement. Comput. Simulat. **38**(2), 344–347, 476 (2021)
2. Bayatli, S., Kurnaz, S., Ali, A., et al.: Unsupervised weighting of transfer rules in rule-based machine translation using maximum-entropy approach. J. Inf. Sci. Eng. **36**(2), 309–322 (2020)
3. Zhu, Z.Q., Wang, S.H., Zhang, Y.D.: ROENet: a ResNet-based output ensemble for malaria parasite classification. Electronics **11**(13), 2040 (2022)
4. Sprott, J.C.: Quantifying the robustness of a chaotic system. Chaos **32**(2), 1–7 (2022)
5. Zhou, Y.Z., Zhao, J.X., Zhai, Q.Z.: 100% renewable energy: a multi-stage robust scheduling approach for cascade hydropower system with wind and photovoltaic power. Appl. Energy **301**(1), 1–12 (2021)
6. Ternero, C., Pastor, G.C.: Bridging the 'gApp': improving neural machine translation systems for multiword expression detection. Yearbook Phraseol. **11**(1), 61–80 (2020)
7. Kim, S.D., Lee, S.K.: English-Korean machine translation system with the improved ability to resolve linguistic differences by pre- and post-processing. J. Linguist. Sci. **92**(3), 151–179 (2020)
8. Pandey, V., Padmavati, D., Kumar, R.: Hindi chhattisgarhi machine translation system using statistical approach. Webology **18**(2), 208–222 (2021)
9. Matheus, A., Adriano, P., Fabrício, B.: A comparative study of machine translation for multilingual sentence-level sentiment analysis. Inf. Sci. **512**(8), 1078–1102 (2020)
10. Dabre, R., Chu, C., Kunchukuttan, A.: A survey of multilingual neural machine translation. Assoc. Comput. Mach. Comput. Surv. **53**(5), 1–38 (2020)