Application and Research of Elman Neural Network Model Optimized Input by Improved Immune Genetic Algorithm in MBR Membrane Flux Prediction

Chunqing Li and Fan Xu

ABSTRACT

Membrane flux is an important indicator to show and prove the extent of membrane pollution. Our team adopted the Elman neural network to establish the MBR contamination simulation prediction model. By setting up the non-linear relationship of influencing MBR membrane pollution’s factors and membrane flux, to complete the membrane flux prediction. Research shows that the MBR membrane flux prediction model based on Elman neural network still has some problems. After reviewing a large number of literatures, we applies improved adaptive immune genetic algorithm based on antibody concentration to optimize this model. At the same time, we compares the optimization effect of this algorithm with the optimization effect of traditional genetic algorithm. By the contrast and analysis, the Elman neural network model optimized input by the improved immune genetic algorithm is superior to the Elman neural network model optimized input by the traditional genetic algorithm on the prediction accuracy and stability of MBR membrane flux.1

KEYWORDS

Membrane Bio-Reactor, MBR Membrane Contamination Prediction, Elman Neural Network, Immune Genetic Algorithm

1Chunqing Li, Fan Xu, College of Computer Science and Software, Tianjin Polytechnic University.
Introduction

Membrane contamination affects the effect of MBR technology, reducing the membrane flux can retard the membrane contamination. How to adjust the system operating conditions and parameters and build the simulation model of membrane flux becomes the research hotspot. Our team adopt the Elman neural network to set up the non-linear relationship of influencing MBR membrane pollution’s factors and membrane flux, to complete the prediction of membrane flux. Research shows that the MBR membrane flux prediction model based on Elman neural network still has some problems, such as the random initialization of network weight leads to the efficiency of network training is low and the number of network’s hidden nodes is determined by trial method, which makes the data is not accurate. After reviewing a large number of documents, we applies improved adaptive immune genetic algorithm based on antibody concentration to optimize the MBR membrane flux prediction model based on Elman neural network[1], and we compared the optimization effect of this algorithm with the optimization effect of traditional genetic algorithm.

PREDICTION MODEL OF MBR MEMBRANE POLLUTION BASED ON ELMAN NEURAL NETWORK

Establishment of MBR Membrane Pollution Prediction Model

In this paper, we use MATLAB to build Elman neural network[2-5]. Because the number of neurons in the input and output layers has been determined, therefore, it is necessary to determine the number of neurons in the hidden layer. Here we use the hidden layer which has a single layer and design the Elman neural network that the number of hidden neurons is variable to determine its number. The hidden layer function takes the default tansig function. The output layer uses the purelin function. The training function takes the default trainlm function. The performance evaluation function of the network adopts the mse function. We set the error range to 2e-3. The number of neurons with the smallest network error is optimal. When the network error is the smallest, we gets the best number of hidden neurons. Part of the training code is shown below.

```matlab
a=3:12; %the number of nodes in the hidden layer
for i = 1:10
    net=newelm(minmax(input_train),[a(i),1]),{'tansig','purelin'});
    sim_out(i,:)=sim(net,input_set);
    MSE(i)=mse(output_test-sim_out(i,:))
end
```

As shown in the following table 2-1, after operation, we can obtain the network error with different hidden layer nodes.
TABLE 2-1. CORRESPONDING NETWORK TRAINING ERRORS FOR DIFFERENT HIDDEN LAYER NODES.

| Nodes | 3   | 4   | 5   | 6   | 7   |
|-------|-----|-----|-----|-----|-----|
| Error | 0.0774 | 0.0762 | 0.0626 | 0.0679 | 0.0626 |

| Nodes | 8   | 9   | 10  | 11  | 12  |
|-------|-----|-----|-----|-----|-----|
| Error | 0.0665 | 0.0672 | 0.0509 | 0.0597 | 0.0558 |

From the above results, we can see that when the number of neurons in the hidden layer is 10, the value of the network objective function is the best, that is, the error is the smallest. The three-tier structure of the network has been identified. The amount of information in the neural network exists in the network connection weights, and the connection weights need to be teacher-trained for sample data. Through its continuous optimization and correction, we ultimately get the most appropriate network weights, and network training is completed. Finally, we need to use the test samples to predict and validate the trained network, to calculate the objective function value and analyze the network forecasting effect.

SIMULATION EXPERIMENT AND RESULT ANALYSIS

There are 80 groups of data used in the experiment. After normalization, we randomly selected 70 sets of data as training input samples and the remaining 10 sets of data are used for predictive verification. The effect of the network is analyzed by calculating the error between the network output and the desired output. The experiment is realized by MATLAB. As shown in figure 2-1, we use the Elman neural network structure diagram established by MATLAB, and the figure 2-2 shows the training performance of the network. We can see that the network tends to be smooth after more than 600 iterations. In the 669th iteration, the mean square error of the network is 0.0019997, reaching a predetermined 0.002. Figure 2-3 shows the Elman neural network prediction fit graph, we can see that the predicted value of the network is basically fitted to the desired output. It is shown that the MBR membrane pollution prediction model established by Elman neural network is basically successful, and able to complete the prediction of MBR membrane flux.

Figure 2-1. Elman neural network structure diagram.
Figure 2-2. Elman neural network training performance chart.

Figure 2-3. Elman neural network prediction fit graph.

| Sample ordinal | Expected value \( (L / m^2 h) \) | Elman Relative error | BP Relative error |
|----------------|-----------------------------------|----------------------|------------------|
| 1              | 45.5000                           | 0.0050               | 0.0135           |
| 2              | 28.9000                           | 0.0073               | 0.0457           |
| 3              | 51.8000                           | 0.0557               | 0.1002           |
| 4              | 9.4000                            | 0.1888               | 0.1934           |
| 5              | 42.2000                           | 0.0100               | 0.0294           |
| 6              | 42.3000                           | 0.0812               | 0.0776           |
| 7              | 21.7000                           | 0.1690               | 0.2111           |
| 8              | 21.6000                           | 0.0360               | 0.0526           |
| 9              | 43.4000                           | 0.0146               | 0.0133           |
| 10             | 32.5000                           | 0.0183               | 0.0034           |

Average relative error: 0.0586 0.0740
As the figure 2-4 said, Elman neural network has a good prediction effect on MBR membrane pollution prediction. It has a higher prediction accuracy, but the individual point of the forecast results is still be a large error. As can be seen from table 2-2, using the same test sample, the average relative error of the Elman neural network is 0.0586. While the BP neural network has a relative error of 0.0740, the error is reduced. Membrane flux prediction accuracy increases by 26.28%.

Through the experiment and the analysis of the experimental results, We can see that, Compared to the BP neural network, the Elman neural network is a typical dynamic recursive network, which makes the accuracy of network training and fault tolerance to further enhance because of joining the feedback of the hidden layer nodes, but the network also has some inherent defects:

(1) As the Elman neural network is still using the gradient descent method for network learning, its convergence is slow and it is easy convergence to local minimum, such as the feed forward neural networks;

(2) The model structure of Elman neural network, the weight range between the layers and the self-feedback gain factor are determined by experiment or trial. This will result in inefficient network training. Which ultimately affects the prediction accuracy and stability of MBR membrane fouling prediction.

In order to make the Elman Neural Networks be more accurate in the MBR membrane flux prediction and more intense in network training, the Elman neural network needs to be optimized.

OPTIMIZATION OF ELMAN NEURAL NETWORK PREDICTION MODEL THROUGH ADAPTIVE IMMUNE GENETIC ALGORITHM BASED ON ANTIBODY CONCENTRATION ALGORITHM IMPLEMENTATION

The Features And Process of Adaptive Immune Genetic Algorithm Based on Antibody Concentration Algorithm Implementation

For the inherent defects of the Elman neural network, this section proposes the use of immune genetic algorithm to solve Elman neural network modeling parameters selection and other issues[1], in order to improve the network training efficiency and forecasting ability of the model. Compared with the traditional genetic algorithm, the flow chart is shown in figure 3-1:
ALGORITHM IMPLEMENTATION

When immune genetic algorithm based on antibody concentration optimizes dynamic recurrent neural network, the mainly optimized parameters are: Elman neural network hidden layer nodes, the connection weights between the layers, the initial value of the contact unit, and the recursive factor of the Elman network. First of all, using the immune genetic algorithm combined with Elman neural network to search for, which is the best antibody. And then, using the best combination of parameters to initialize the Elman neural network and train the network, At last, using the test samples to predict and verify the network. The key steps are described below:

1. Antibody coding and initialization
   The encoding is encoded in real numbers. Assume the number of nodes of the input layer of the network model is $s_1$, the number of nodes of the hidden layer and carrying layer is $s_2$, The number of nodes of the output layer is $s_3$, the self-connection feedback factor of the network is $\alpha$. So the number of parameters to be optimized is $1 + s_1 \times s_2 + s_2 \times s_2 + s_2 \times s_3$, set the antibody to $X (G_1, \cdots, G_m)$ .Among them, $X$ represents antibody, $G$ represents the gene in the antibody. $m$ represents the number of genes. Each real range is $(-1,1)$.

2. Antibody affinity assessment
   Since the Elman network is finally optimized, the error squared sum between the network output and the expected output is set as the objective function. The reciprocal of the objective function is taken as a function of calculating the antibody affinity, which is:
   \[
   \text{fit}(i) = \frac{1}{\sum_{i=1}^{n} (y(i) - y_d(i))^2}
   \]
$y_d(i)$ is the specimen $i$’s the expects output of the network. $y(i)$ is the specimen $i$’s the actual output of the network

3. Evaluation of antibody similarity

Calculating the similarity between the two antibodies. For the antibody $X_w(G_{i}^w, \cdots, G_{m}^w)$ and the antibody $X_v(G_{i}^v, \cdots, G_{m}^v)$, the similarity between the two antibodies was defined as:

$$H_{wv} = \sqrt{\sum_{k=1}^{m} (G_{k}^w - G_{k}^v)^2}$$

In the formula, $G$ is the antibody gene. $m$ is the number of genes. The concept of similarity between two antibodies is extended to the whole population, and the overall similarity is obtained, it is $A(N)$, defined as: $A(N) = 1/1 + H(N)$, $A(N) \in (0,1)$, the smaller the value of $A(N)$, the lower the similarity of the population, that is to say that the population diversity is higher.

4. Population regulation based on antibody concentration

Define the antibody concentration as the ratio of the number of similar antibodies to the total number of antibodies, calculating and obtaining the antibody similarity, according to the definition of antibody concentration, the antibody concentration can be calculated, $C_i = t/N$. In the formula, $C_i$ is the concentration of antibody $i$, $t$ indicates the number of antibodies $(0.9 \leq \lambda \leq 1)$ which antibody similarity is greater than $\lambda$ compared with antibody $i$. $N$ indicates the total number of antibodies.

When the population is updated, it is necessary to consider the antibody affinity’s sum of the concentration, and introducing the polymerization fitness, defining it as follows: $fit' = fit \times \exp(k \times C_i)$, $k$ takes a negative number, in algorithm, taking $k = -0.8$, after the polymerization fitness is calculated and obtained, the population is suppressed and facilitated according to the polymerization fitness.

5. Select, cross, mutate operation

Select operation: the selection operation is based on the aggregation fitness to choose, using the roulette selection mechanism. The probability of each antibody being selected is proportional to the polymerization fitness of the antibody. The probability of the antibody being selected is:

$$P_i = \frac{fit'(i)}{\sum_{i=1}^{n} fit'(i)}$$
Cross operation: the antibody $X^w(G_1^w, \cdots, G_m^w)$ and the antibody $X^v(G_1^v, \cdots, G_m^v)$ performs an arithmetic cross in the position $(i, j)$. The next generation of antibodies is $X^w(G_1^w, \cdots, G_i^w, G_j^v, \cdots, G_m^w)$ and $X^v(G_1^v, \cdots, G_i^v, G_j^w, \cdots, G_m^v)$, among them:

$$G_i^w = \varepsilon G_i^w + (1-\varepsilon)G_i^v, \quad G_i^v = \varepsilon G_i^v + (1-\varepsilon)G_i^w.$$  

$\varepsilon$ is the constant between $(0, 1)$.

Variant operation: use the following formula for mutation operation. In the formula, $P_m^i$ is the $i$th generation individual mutation rate.

$$P_m^i = \begin{cases} 
P_m^{i-1} - \frac{(P_m^i - P_m^{i-1})(\text{fit}_{\text{max}} - \text{fit})}{\text{fit}_{\text{max}} - \text{fit}_{\text{avg}}}, & \text{fit} > \text{fit}_{\text{avg}} \\
P_m^{i-1}, & \text{fit} < \text{fit}_{\text{avg}}
\end{cases}$$

The above is an adaptive mutation operator, which can improve the efficiency of mutation operation and speed up the convergence rate.

CALCULATION RESULTS AND ANALYSIS

In this algorithm, the population size is 50, the threshold of the population similarity is 0.1, the crossover rate is 0.85, the mutation rate is 0.05, the maximum number of evolutionary iterations is 300, the antibody group is initialized, and then, do a teacher training and learning to the various parameters of the Elman neural network. After getting the best individual, the user initializes the Elman neural network. In order to verify the optimization effect of evolutionary algorithm, we use the same sample data to train the Elman neural network, and use the test samples to predict and verify the network. Figure 4-1 shows the improved data in the optimized network after the optimized network. We can see that the optimized network has greatly improved in the forecast effect.

Figure 4-2 compares the predicted data with the real data. It can reflect the effect of the prediction and the error of prediction more clearly.

![Fit graph of the forecast data.](image-url)
Figure 4-2. Comparison of forecast data with actual values.

From the figure 4-2, after the Elman neural network is optimized, the training effect of the network is better, which shows that the learning ability of the dynamic neural network optimized by immune genetic algorithm is better than that of the no optimized Elman neural network. It is found that the optimized model is more stable, the final error changes smaller. To illustrate the optimization of the immune genetic algorithm is better than the traditional genetic algorithm, using the same sample data to train the network, table 4-1 is for the two algorithms optimized network training results.

| Sample ordinal | Expected value (L/m²h) | SAIGA-Elman | Relative error | GA-Elman | Relative error |
|----------------|------------------------|-------------|---------------|----------|---------------|
| 1              | 45.5000                | 46.0020     | 0.0117        | 45.1959  | 0.0097        |
| 2              | 28.9000                | 28.7380     | 0.0658        | 29.3148  | 0.0144        |
| 3              | 51.8000                | 49.8164     | 0.0283        | 49.5254  | 0.0376        |
| 4              | 9.4000                 | 8.9836      | 0.0484        | 9.0807   | 0.0345        |
| 5              | 42.2000                | 43.1100     | 0.0216        | 43.4003  | 0.0294        |
| 6              | 45.3000                | 45.6940     | 0.0380        | 45.2678  | 0.0702        |
| 7              | 21.7000                | 22.8917     | 0.0549        | 23.1916  | 0.0687        |
| 8              | 21.6000                | 21.8978     | 0.0138        | 20.2706  | 0.0615        |
| 9              | 45.4000                | 45.7935     | 0.0540        | 45.3081  | 0.0094        |
| 10             | 32.5000                | 32.1951     | 0.0094        | 31.7723  | 0.0224        |

Average relative error: 0.0246% 0.0334%

It can be seen from the above table that after several calculations and experiments, it is found that the Elman neural network model optimized by the adaptive immune genetic algorithm based on the antibody concentration is superior to the Elman neural network model optimized by the traditional genetic algorithm in the prediction accuracy and stability of MBR membrane fouling, indicating that SAIGA-Elman model is more suitable for MBR flux prediction.
CONCLUSIONS

The prediction accuracy of the MBR membrane flux simulation prediction model based on Elman neural network has been heightened, the accuracy and fault tolerance of the network training and learning have been further enhanced. However, this model still has some problems such as low network training efficiency and inaccurate calculation data. After reviewing a large number of literatures, we applies improved adaptive immune genetic algorithm based on antibody concentration to optimize Elman neural network model, which realizes the automatic structure and the programming of the dynamic recursive neural network and improves the network training efficiency and prediction accuracy. At the same time, we compares the optimization effect of this algorithm with the optimization effect of traditional genetic algorithm, in order to verify the effect of the model, we use the same sample to train and test the simulation model. The experiment and calculate results show that, the Elman neural network model optimized input by the adaptive immune genetic algorithm based on the antibody concentration is superior to the Elman neural network model optimized input by the traditional genetic algorithm in the prediction accuracy and stability of MBR membrane fouling.

ACKNOWLEDGMENT

This research was supported by the National Natural Science Foundation of China(51378350) and the National Natural Science Foundation of China(50808130).

REFERENCES

1. Feng-Ping L. Research on Several Kinds of Artificial Immune Algorithm and Its Application [D]. Jilin University, 2009.
2. Jing-Tian Tang, Cao Yang, Jia-Ying Xiao, et al. Predication of plasma concentration of remifentanil based on Elman neural network [J]. Journal of Central South University, 2013, 20(11): 3187-3192.
3. Kao-She Z., Jian Yang. Short - term wind power prediction based on Elman neural network [J]. Grid and clean energy, 2012, 28(12): 87-91.
4. Dong-Wen Cui. Application of Improved Elman Neural Network in Runoff Forecast [J]. Journal of Water Resources and Hydropower Engineering, 2013(2): 71-77.
5. Ding Y. Ad Hoc network traffic prediction based on the Elman neural network [J], 2015.