Inventory Prediction Based on Backpropagation Neural Network

Lichuan Gu*, Yueyue Han, Chengji Wang, Guiyang Shu, Juanjuan Feng, Chao Wang

ABSTRACT

This paper aims to develop an effective way to predict the inventory demand of agricultural materials. Focusing on the demand of agricultural pesticide, the author introduced the backpropagation neural network (BPNN) and optimized the BPNN inventory prediction model by multiple interpolation method. In this way, a novel inventory prediction strategy was created, with the national macro policy, the pest and disease resistance, the market role and other factors as part of the BPNN. For the lack of input samples, the multiple interpolation method was adopted to restore the missing data. Then, the replacement values were combined in different ways to reveal the variation pattern of prediction error, making it possible to predict the exact pesticide demand. The research provides the decision support for inventory management in agricultural materials enterprises.

Key Words: Agricultural Materials, Inventory Prediction, Backpropagation Neural Network (BPNN), Multiple Interpolation

DOI Number: 10.14704/nq.2018.16.6.1608 **NeuroQuantology 2018; 16(6):664-673**

Introduction

The demand of agricultural materials varies obviously with the seasons, regions, policies and other factors, leading to oversupply or short supply of agricultural materials. To reduce warehousing cost, enhance enterprise profitability and guarantee normal production, it is necessary to find a scientific prediction method for the inventory demand of agricultural materials, and minimize the difference between the inventory and the actual demand.

The accuracy of inventory prediction relies heavily on the unified management of the circulation of agricultural materials. With the proliferation of Internet technology, agricultural enterprises are increasingly familiar with the management information system. The system helps to establish an information network that integrates external sales organizations with internal business units, laying the basis for the unified management of information and product circulation. In this way, agricultural enterprises can improve their operation efficiency with information technology (Wu, 2014).

Demand prediction is currently the focus of the research on rational inventory control. In 1915, Harris wrote the *Operations and Cost*, marking the first attempt to quantify the inventory. To solve the lack of historical data in medium-term sales forecast, Thomassey and Happiette (2007) proposed an assistant decision system using clustering and classification tools based on the neural network. In China, Jiang (2017) combined backpropagation neural network (BPNN) with particle swarm optimization (PSO) into a prediction model for the safety inventory in coal. Liu (2012) explored the new paradigm, organization and management mode of retail chain enterprises and created a neural network model that accurately forecasts the safety inventory.

Corresponding author: Lichuan Gu
Address: School of Computer and Information, Anhui Agricultural University No. 130 ChangJiang Road, Hefei 230036, China
e-mail ✉ glc@ahau.edu.cn
Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
Received: 4 March 2018; **Accepted:** 2 May 2018

Considering the set value, Zhou Hua developed a BPNN model to predict the inventory demand in that industry (Zhou *et al*., 2016). To predict the actual safety stock, Liu Tengda and Du Tiancang improved the BP neural network by using the learning rate adaptive algorithm and simulated the production data of the steel plant by using the improved neural network (Liu and Tiancang, 2017). However, the model fails to consider factors other than time and historical sales.

Targeted at the demand of agricultural pesticide, this paper introduces the BPNN to identify replacement values and optimize the prediction algorithm. Then, the replacement values were combined in different ways to reveal the variation pattern of prediction error, making it possible to predict the exact pesticide demand. For the lack of input samples, the idea of multiple interpolation was followed to generate a set of replacement values for each missing data. The research provides the decision support for inventory management in agricultural materials enterprises.

Terminology

With excellent nonlinear mapping ability, the BPNN has become one of the most popular neural network models since its birth in 1986 (Fan, 2011). The nonlinear mapping ability refers to the capacity to learn and store numerous mapping relationships between input and output without needing to know the relevant mathematical expressions in advance. In the context of learning, backpropagation is commonly used by the gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function, seeking to minimize the sum of squared errors of the network (Yuan and Yu, 2014). The practice has proved that a three-layer network supports any mapping of n dimensions to m dimensions. Therefore, the BPNN model consists of three layers: the input layer, the hidden layer, and the output layer. Among them, the input layer receives the input information and transfers it to the neurons in the hidden layer. The hidden layer, located in the middle of the model, is responsible for processing the received information. Depending on the information property, the model should be designed with a single hidden layer or multiple hidden layers. In the case of multiple hidden layers, the last hidden layer transfers the processed information to each neuron in the output layer. Finally, the output layer further processes the information, outputs the processing results, and completes the backpropagation learning process (Mai *et al*., 2015). Figure 1 illustrates a BPNN with multiple input layer neurons, multiple output layer neurons, and a single hidden layer.

Therefore, this model includes three layers: the Input Layer is responsible for receiving input information, which is transferred to the middle layer neurons in the hidden layer. Hidden Layer, the internal information processing Layer, is responsible for the information transformation. According to the demand for ability to adapt the change of information, the middle Layer can be designed as a single hidden layer or multi hidden layer. The information, transmitted to each neuron in output layer by the last hidden layer, after further processed, it has completed a forward propagation process of learning. The Output Layer outputs the information processing results (Mai *et al*., 2015). Figure 1 shows input layer with multiple, output layer with multiple neurons and single hidden layer BP neural network structure.

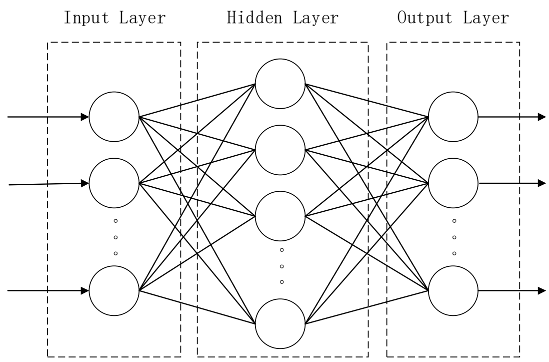


Figure 1. BPNN structure

The BPNN learning contains the following stages:

1) Suppose there are n, p and q neurons in the input layer, hidden layer and output layer, respectively. Then, denote the input vector as x, the hidden layer input vector as h_i, the hidden layer output vector as h_o, the output layer input vector as y_i, the output layer output vector as y_o, the expected output vector as d_o, the hidden layer neuron threshold as w_{ih}, the output layer neuron threshold as w_{ho}, the number of samples as k, the activation function as f, and the error function as $e = \frac{1}{2}\sum_{o=1}^{q}(d_o(k) - yO_o(k))^2$. Assign a random number to each connection weight and let ε and M be the calculation accuracy and the maximum number of learning cycles.

2) Randomly select the k^{th} input sample and the corresponding expected output:

$$x(k) = (x_1(k), x_2(k), \dots, x_n(k)) \quad (1)$$

$$d_o(k) = (d_1(k), d_2(k), \dots, d_n(k)) \quad (2)$$

3) Calculate the input and output of hidden layer neurons:

$$hi_h(k) = \sum_{i=1}^{n} w_{ih} x_i(k) - b_h \quad h = 1,2,\dots,p \quad (3)$$

$$ho_h(k) = f(hi_h(k)) \quad h = 1,2,\dots,p \quad (4)$$

$$\mathrm{ho_h(k) = f(hi_h(k))} \quad \mathrm{h = 1,2,\dots,p} \quad (5)$$

$$yo_o(k) = \sum_{h=1}^{p} w_{ho} ho_h(k) - b_o \; o = 1,2,\dots,q \quad (6)$$

$$yo_o(k) = f(yi_o(k)) \quad o = 1,2,\dots,q \quad (7)$$

4) Find the partial derivatives of the output layer neurons based on the expected output and actual output $\delta_o(k)$

$$\delta_o(k) = -\frac{\partial_e}{\partial_{yi_o}} - (d_o(k) - yo_o(k)) f'(yi_o(k)) \quad (8)$$

5) Derive $\delta_h(k)$, the partial derivative of the error function for each hidden layer neuron, in light of the connection weights from the hidden layer to the output layer, the $\delta_o(k)$ of the output layer, and the output of the hidden layer:

$$\delta_h(k) = -\frac{\partial_e}{\partial_{hi_h}(k)} - (\sum_{o=1}^{q} \delta_o(k) w_{ho}) f'(hi_h(\mathrm{k})) \quad (9)$$

6) Correct the connection weights $w_{ho}(k)$ based on the $\delta_o(k)$ of each output layer neuron and the output of each hidden layer neuron:

$$w_{ho}^{N+1} = w_{ho}^{N} + \Delta w_{ho}(k) \quad (10)$$

7) Correct the connection weights based on $\delta_h(k)$ of each hidden layer neuron and the input of input layer neuron:

$$w_{ih}^{N+1} = w_{ih}^{N} + \Delta w_{ih}(k) \quad (11)$$

8) Calculate the global errors:

$$E = \frac{1}{2m} \sum_{k=1}^{m} \sum_{o=1}^{q} (d_o(k) - y_o(k))^2 \quad (12)$$

9) Determine whether the network error meets the requirements. The algorithm ends when the error reaches the preset accuracy or the number of learning cycles exceeds the maximum number. Otherwise, select the input samples and expected output for next learning cycle, and continue with the learning process (Yang, 2011).

Model Building

Pesticide stands out of the various agricultural materials (e.g. pesticide, fertilizer and seeds) as the one with the most violent demand fluctuations and the most number of influencing factors. The main factors affecting the pesticide demand include the national macro policy, the crop acreage, the occurrence of pests and weeds, the pest and disease resistance, the market role, and the biotechnology (Shu *et al.*, 2010). In view of this, imidacloprid, a pesticide on wheat aphids, is taken as the object of the agricultural inventory prediction in this research. The prediction model was applied after readjusting the other agricultural materials like fertilizer and seeds.

For the purpose of preventing wheat aphids, the input layer neurons of the model were adjusted as per the prevention plan. The input layer is shown in Table 1.

$$T = \begin{bmatrix} x_{1,1}, & x_{1,2}, \cdots & x_{1,n}; \\ x_{2,1}, & x_{2,2}, \cdots & x_{2,n}; \\ & \cdots & \\ x_{19,1} & x_{19,2}, \cdots & x_{19,n}; \end{bmatrix} \quad (13)$$

Where T is the input sample matrix of the model; $x_{m,n}$ is the value of the m^{th} factor in the n^{th} year, with m being the number of influencing factors and n being the order of the year in the selected time series. Here, the BPNN model involves input layer with multiple neurons and output layer with a single neuron. As shown in Table 2, the expected output, i.e., the imidacloprid demand, varies with the trainings and simulations.

To establish the BPNN, the data in the first n-3 columns in Table 1 were adopted as the set of training samples *T1* for network learning, and the data in the last three rows were used as the set of test data. *T2* for network detection. The BPNN was modelled on Matlab through the following steps:

Data normalization: Map data to [0, 1], [-1, 1] or even smaller intervals. Here, the data are normalized to the interval of [0, 1] by the equation below:

$$xnew = \frac{x - min}{max - min} \quad (14)$$

where X is the input vector; *max* and *min* are the maximum and minimum values of *x*, respectively; *xnew* is the normalized output. The entire sample data matrix was processed in Matlab using the *mapminmax* function. The syntax is: *Tin* = *mapminmax* (*T1*, 0, 1) (*T1* is the sample data matrix before the normalization; *Tin* is the sample data matrix after the normalization). BPNN establishment: The hidden layer was activated by the sigmoid function *logsig*, the output layer was activated by the linear function *purelin*, the training was performed by the quasi-Newton algorithm *trainbfg*. The number of hidden layer nodes can be determined by the following empirical equations:

$$hidden = \sqrt{in + out} + a \qquad (15)$$

$$hidden = \sqrt{in * out} \qquad (16)$$

$$hidden = \log_2 in \qquad (17)$$

Where *hidden* is the number of hidden nodes; *in* is the number of input layer nodes; *out* is the number of output layer nodes; *a* is an integer from 1 to 10. In our model, *in* is set to 19, and *out* is set to 1. According to the above equations, the number of hidden layer nodes *hidden* should fall in the range of [4.25, 14.47]. The number of hidden layer nodes in our experiment is 9, which needs to be adjusted to find the optimal value. Hence, the *newff* function was adopted to create the network with the following syntax: net = *newff* (*minmax* (Tin), [hidden out], {'logsig', 'purelin'}, 'trainbfg'). Parameter setting: the maximum number of training cycles is set to 5,000, the training accuracy, 0.00001, the learning rate, 0.01, and the momentum factor, 0.9.

Simulation and multiple interpolation

Simulation verification

The network was trained by the function with the syntax of net=train (*net*, *Tin*, *Tout*). After the training, the derived network formed our model and can be analysed online. To verify the prediction accuracy, the last three items of T were combined with T1 into test data samples T2 and T2, the two samples were normalized into Tin2, and Tin2 was inputted to the network for simulation. The predicted value was contrasted with the actual value to determine the prediction accuracy. The simulation ran on the *Sim* function with the syntax of net_out=*sim* (net, Tin2).

Table 1. The input layer of the model

		Year 1	Year 2	Year 3	...	Year n
1	The planting area of wheat in Huaibei city	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	...	$x_{1,n}$
2	The price of wheat	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	...	$x_{2,n}$
3	The area of the wheat aphid	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	...	$x_{3,n}$
4	Class of wheat aphids	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	...	$x_{4,n}$
	Pesticide policy support					
5	Imidacloprid	$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	...	$x_{5,n}$
6	Acetamiprid	$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	...	$x_{6,n}$
7	24 percent against aphid	$x_{7,1}$	$x_{7,2}$	$x_{7,3}$	...	$x_{7,n}$
	Wheat aphid drug resistance					
8	Imidacloprid	$x_{8,1}$	$x_{8,2}$	$x_{8,3}$	...	$x_{8,n}$
9	Acetamiprid	$x_{9,1}$	$x_{9,2}$	$x_{9,3}$	...	$x_{9,n}$
10	24 percent against aphid	$x_{10,1}$	$x_{10,2}$	$x_{10,3}$	...	$x_{10,n}$
	Historical sales					
11	Imidacloprid	$x_{11,1}$	$x_{11,2}$	$x_{11,3}$	...	$x_{11,n}$
12	Acetamiprid	$x_{12,1}$	$x_{12,2}$	$x_{12,3}$	...	$x_{12,n}$
13	24 percent against aphid	$x_{13,1}$	$x_{13,2}$	$x_{13,3}$	...	$x_{13,n}$
	The sales price					
14	Imidacloprid	$x_{14,1}$	$x_{14,2}$	$x_{14,3}$	...	$x_{14,n}$
15	Acetamiprid	$x_{15,1}$	$x_{15,2}$	$x_{15,3}$	...	$x_{15,n}$
16	24 percent against aphid	$x_{16,1}$	$x_{16,2}$	$x_{16,3}$	...	$x_{16,n}$
	Sales profit					
17	Imidacloprid	$x_{17,1}$	$x_{17,2}$	$x_{17,3}$	...	$x_{17,n}$
18	Acetamiprid	$x_{18,1}$	$x_{18,2}$	$x_{18,3}$	...	$x_{18,n}$
19	24 percent against aphid	$x_{19,1}$	$x_{19,2}$	$x_{19,3}$	...	$x_{19,n}$

Table 2. The expected outputs

	Year 2	Year 3	...	Year n-1	Year n
The Demand of Imidacloprid	$y_{1,1}$	$y_{1,2}$	...	$y_{1,n-1}$	$y_{1,n}$
The Demand of Acetamiprid	$y_{2,1}$	$y_{2,2}$	...	$y_{2,n-1}$	$y_{2,n}$
The Demand of 24 percent against aphid	$y_{3,1}$	$y_{3,2}$	...	$y_{3,n-1}$	$y_{3,n}$

Table 3. Prediction error

	10 times forecast		20 times forecast	
The real value	Average forecast value	error rate	Average forecast value	error rate
0.583069828	0.603367478	3.48%	0.575325	1.33%
0.710934273	0.675491569	4.98%	0.691767	2.69%
0.937153093	0.871153362	7.04%	0.897164	4.27%

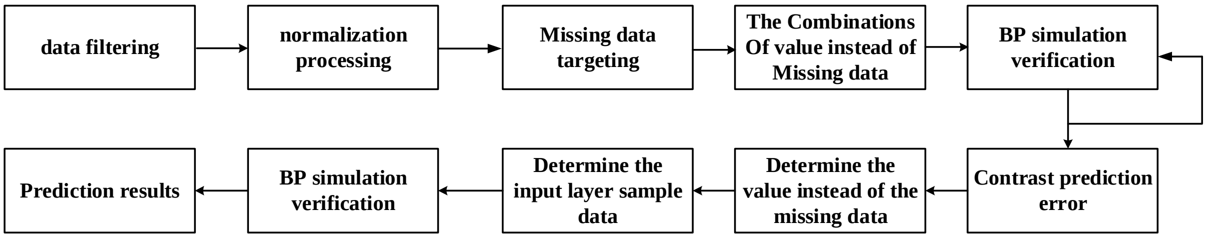


Figure 2. Three-phase prediction process

The results show that the BPNN method performed well in the demand prediction of imidacloprid inventory.

Multiple interpolation of the missing data

The numerical values of the sample data must be kept constant during the simulation of the BPNN model, so as to simplify the input layer and explore the sample data on prediction efficiency. Therefore, the missing data must be interpolated to ensure the completeness of the sample data in the input layer for the demand prediction. The interpolation divides the entire prediction process into three phases: data filtering, missing data interpolation and prediction phase (Figure 2).

(1) Data filtering

For some of the sample data, the vector remained the same throughout the sampling period. After mapping, these data tend to form a near-horizontal line segment in the coordinate system. In this research, these data are known as the horizontal data. For example, the policy support to imidacloprid may stay consistent for a long time, and the support level can be regarded as a type of horizontal data. According to the data normalization formula, the horizontal data had no impact on the prediction results. The normalized result was either 0 or 1 when the difference between the max. and min. was 0, thus rendering the data meaningless. To disclose the effect of the horizontal data on the prediction efficiency, different volumes of horizontal data were interpolated into Tin, and then inputted into the same model. The simulation results in each cycle, together with the run time, were recorded (Table 4).

Table 4. Simulation results and run time at different volumes of horizontal data

Horizontal data set	Time-consuming growth rate
0	0
1	9.51%
2	14.65%
5	18.44%

It is clear that the run time of the model increased with the volume of horizontal data, while the prediction efficiency decreased. For better efficiency, the data of T was filtered before being normalized into Tin, and the rows containing horizontal data were removed from the sample data matrix. The filtering was realized through the function *all* in Matlab. The syntax is: I = all (T, 2) | all (~ T, 2); T (I,:) = []. The filtering eliminated the horizontal data contained in T. The post-filtering T was normalized into Tin, and inputted to the BPNN model. In this way, the number of input layer neurons decreased and the prediction became more efficient.

(2) Missing data interpolation

In this phase, the missing data are replaced by the multiple interpolation method (Pang, 2012). To

*e*ISSN 1303-5150

validate the method, a random data point M1 (actual value: 0.7718) was selected from Tin as the location of the missing data, and the replacement value of M1 was denoted as M1'. Since Tin is a normalized matrix, the set of replacement values for the missing data was generated from the interval [0,1] by a step of 0.1. Hence, M1' was expressed as a set {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}. The values in the set were inputted to the model one by one. The prediction results are shown in Table 5.

Table 5. Prediction results after missing data interpolation

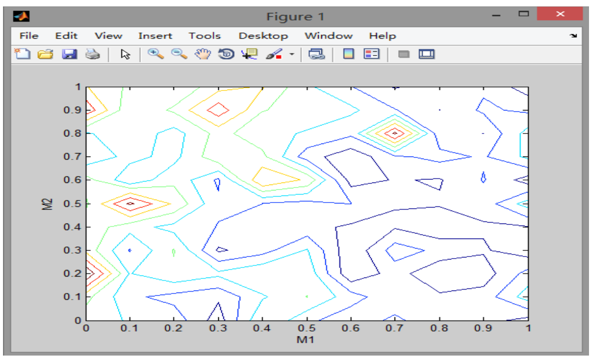
Substitute value	Forecast average error rate
0	2.77%
0.1	3.20%
0.2	2.39%
0.3	1.57%
0.4	2.37%
0.5	0.96%
0.6	0.92%
0.7	0.64%
0.8	0.70%
0.9	1.22%
1	0.77%

It can be seen that, as the replacement value M1' approached the actual value of 0.7718, the prediction error decreased. The inverse is also true.

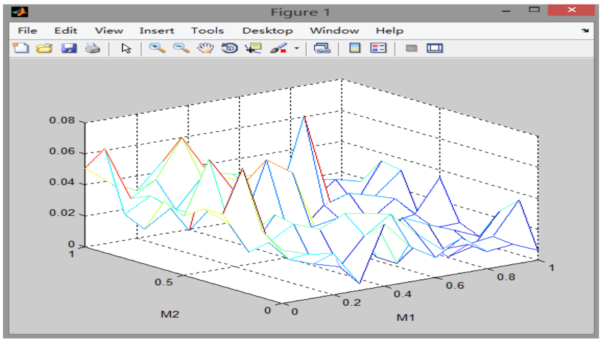
a) Determination of the set of replacement values

This sub-section aims to determine the set of replacement values for the missing data. The size of the dataset directly bears on the simulation time and the distribution density of the replacement values, which, in turn, determine the prediction efficiency and prediction accuracy (Spoorthy *et al*., 2016). To identify the size of the dataset, it is necessary to explore how different sets of replacement values affect the prediction efficiency and accuracy. Therefore, another data point M2 (actual value: 0.3557) was selected from the Tin as the second missing data point, and the replacement value of M2 was denoted as M2'.

Plan 1: When M1' and M2'∊ {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}, the two sets each has 11 replacement values. In this case, there are 121 combinations of replacement values. Taking the 121 combinations as the sample data for simulation, the author obtained a total of 121 prediction results. The error distribution of the prediction results was plotted by Matlab (Figure 3).



(a)

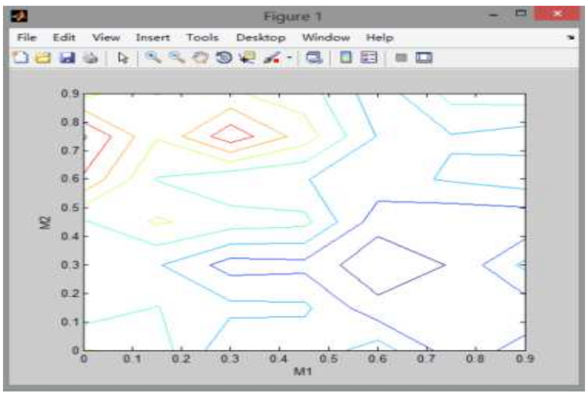


(b)

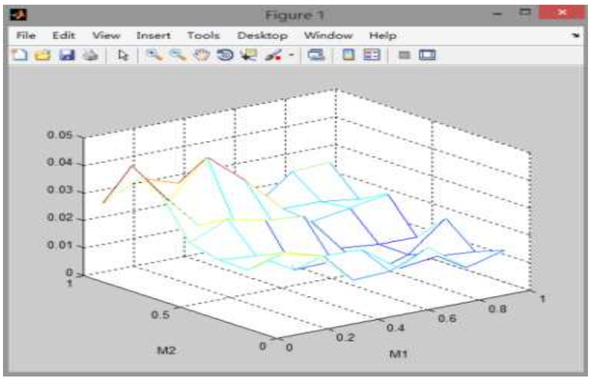
Figure 3. Error distribution of Plan 1.
(a)Contour map*; (b)Network surface
**The redder the colour, the greater the error*

According to Figure 3, the error plunged as (M1', M2') approached the actual values (0.7718, 0.3557).

Plan 2: When M1' and M2' ∈ {0, 0.15, 0.3, 0.45, 0.6, 0.75, 0.9}, the two sets each has 7 replacement values. In this case, there are 49 combinations of replacement values. Following the same process in Plan 1, the error distribution was drawn by Matlab.



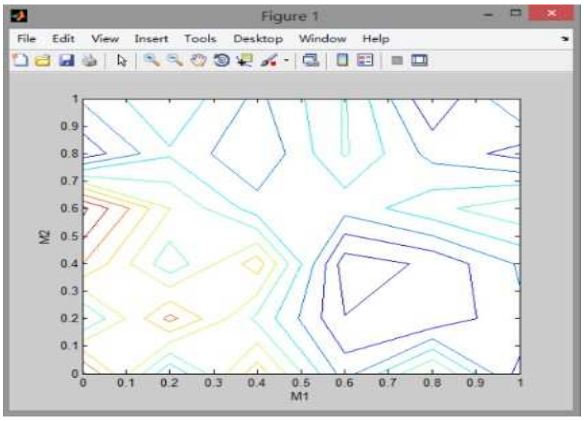
(a)



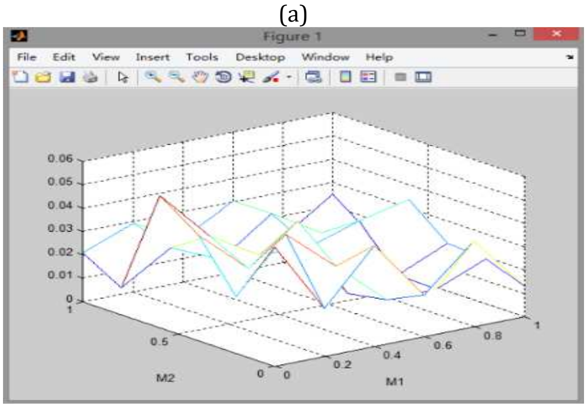
(b)

Figure 4. Error distribution of Plan 2.
(a)Contour map; (b) Network surface

Compared to Figures 4-2 (b) and 4-3 (b), Figure 5 (b) reflects a small error fluctuation, making it hard to determine the range of the actual values.



(a)



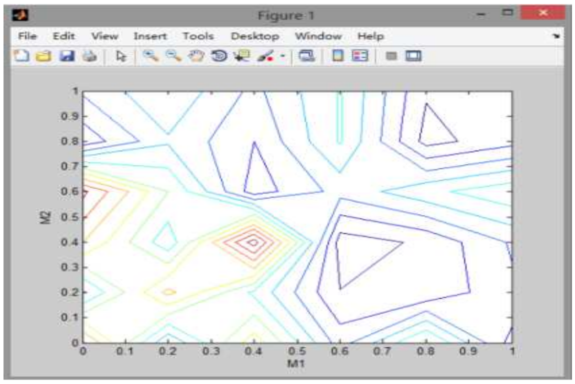
(b)

Figure 5. Error distribution of Plan 3.
(a)Contour map; (b) Network surface

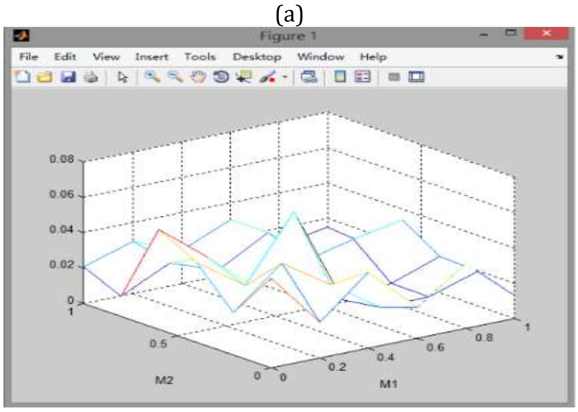
Plan 4: When M1’ and M2' ∈ {0, 0.25, 0.5, 0.75, 1}, the two sets each has 5 replacement values. In this case, there are 25 combinations of replacement values. Following the same process in Plan 1, the error distribution was drawn by Matlab.

The prediction error curve of Plan 4 is more gradual than that of the previous 3 plans. It cannot be used to determine the range of the actual values.

Plan 5: When M1’ and M2' ∈ {0, 0.3, 0.6, 0.9}, the two sets each has 4 replacement values. In this case, there are 16 combinations of replacement values. Following the same process in Plan 1, the error distribution was drawn by Matlab



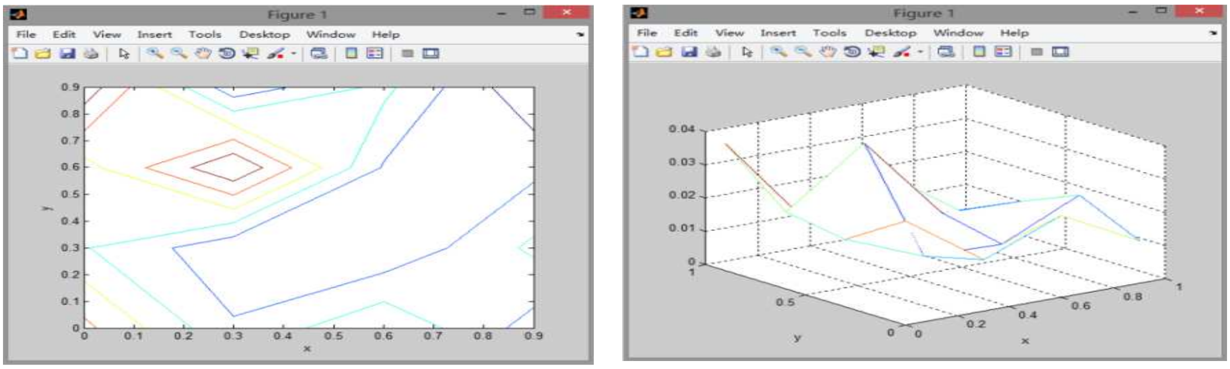
(a)



(b)

Figure 6. Error distribution of Plan 4.
(a)Contour map; (b) Network surface

*e*ISSN 1303-5150



(a) (b)

Figure 7. Error distribution of Plan 5. (a)Contour map; (b) Network surface

In this plan, the number of substituting values is too small to provide a reference value for the range of actual values.

Comparing the prediction results of the above five plans, the author discovered that the variation pattern of the prediction error can be found when {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1} (Plan 1) was taken as the set of the replacement values, and the variation pattern can also be identified intuitively when {0, 0.15, 0.3, 0.45, 0.6, 0.75, 0.9} (Plan 2) was taken as that set. In other plans, the prediction error was too stable to determine the range of the actual values.

Although Plans 1 and 2 did well in the fitting of the actual values, they may consume a long time in the simulation process. The simulation time of different plans is illustrated in Figure 8. As shown in the figure, Plan 1 took 2,357 seconds to complete the simulation, while Plan 2 consumed a 153% shorter time than Plan 1. Of course, the simulation time of Plan 2 is 26% longer than that of Plan 3. Taken together, Plan 2 {0, 0.15, 0.3, 0.45, 0.6, 0.75, 0.9}was adopted for the multiple interpolation, aiming to strike a balance between prediction efficiency and accuracy.

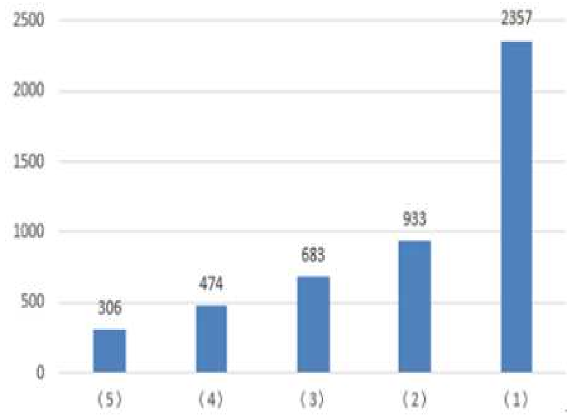


Figure 8. Simulation time of different plans

b) Determination of the optimal replacement values

The next step is to find the replacement value lying the closest to the actual value. In the above simulation on Plan 2, there are 49 combinations of replacement values, and thus 49 prediction results. Taking the results as the matrix of M1' and M2', the linear method and the rectangular method were adopted to identify the optimal value of M1' and M2'.

The linear method:

Let us denote the optimal value of M1’ as BestM1’. Then, the 7 combinations between BestM1’ and M2’ should have the smallest mean prediction error than the other 42 combinations of replacement values. Similarly, let us denote the optimal value of M2’ as BestM2’. Then, the 7 combinations between BestM2’ and M1’ should also have the smallest mean prediction error than the other 42 combinations of replacement values. Overall, (BestM1', BestM2') must be the optimal replacement values.

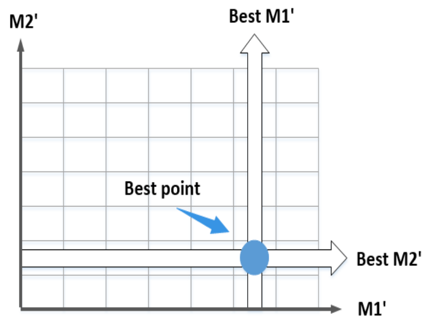


Figure 9. The linear method

By this method, the mean prediction error of each combination was obtained and recorded in the table below.

As shown in the table, the mean prediction error was the lowest at the M1’ of 0.6 and the M2’ of 0.3. The four combinations with

Table 6. Mean prediction error

Substitution value	0	0.15	0.3	0.45	0.6	0.75	0.9
M1'	0.03017	0.02775	0.02468	0.02253	0.01361	0.01576	0.01844
M2'	0.02035	0.01746	0.01522	0.01868	0.02380	0.02986	0.02759

Table 7. Replacement values corresponding to the central points

NO.	M1'	M2'	NO.	M1'	M2'	NO.	M1'	M2'
1	0.15	0.15	10	0.3	0.75	19	0.6	0.6
2	0.15	0.3	11	0.45	0.15	20	0.6	0.75
3	0.15	0.45	12	0.45	0.3	21	0.75	0.15
4	0.15	0.6	13	0.45	0.45	22	0.75	0.3
5	0.15	0.75	14	0.45	0.6	23	0.75	0.45
6	0.3	0.15	15	0.45	0.75	24	0.75	0.6
7	0.3	0.3	16	0.6	0.15	25	0.75	0.75
8	0.3	0.45	17	0.6	0.3			
9	0.3	0.6	18	0.6	0.45			

Table 8. Normalized mean prediction errors

M2'	0.15	0.30	0.45	0.60	0.75
0.15	0.46459	0.34567	0.17323	0.100548	0.09511
0.30	0.50933	0.47380	0.22823	0.086679	0
0.45	0.59908	0.46604	0.25819	0.155231	0.127612
0.60	0.93950	0.77959	0.58972	0.374142	0.22788
0.75	1	0.82645	0.63863	0.490116	0.426949

the smallest prediction error are (0.6, 0.3), (0.75,0.3), (0.6, 0.45) and (0.75, 0.45). Among them, (0.75,0.3) is the closest to the actual values (0.7718, 0.3557). This means the linear method is very reliable in finding the optimal replacement values.

The rectangular method:
This method looks for the data area with the smallest mean prediction error is ascertained, and then pinpoints the centre of the interval of the optimal replacement value. The data are grouped into 3x3 continuous units, with the central point being the level of the regional prediction error. In Plan 2, there are 25 valid central points. The corresponding replacement values are shown in Table 7. The mean error of each combination, together with its adjacent 8 combinations, was calculated and normalized. The normalized data are displayed in Table 8.

According to the normalized data in the table above, the regional prediction error reached the minimum when the replacement values of M1 and M2 converged 0.75 and 0.3, respectively. In other words, (0.75, 0.3) is the closest combination to the actual values (0.7718, 0.3557). It is proved that the rectangular method is also a useful tool for determining the optimal replacement values.

Comparatively speaking, the optimal replacement values found by the rectangular method are closer to the actual values than those determined by the linear method. Therefore, these values were taken as the optimal ones and used as the input layer sample data for the prediction of inventory demand.

Conclusions

With the aim to predict the inventory demand of agricultural materials, especially agricultural pesticide, this paper introduces the backpropagation neural network (BPNN) and optimizes the BPNN inventory prediction model by multiple interpolation method. In this way, a novel inventory prediction strategy was created, with the national macro policy, the pest and disease resistance, the market role and other factors as part of the BPNN. For the lack of input samples, the multiple interpolation method was adopted to restore the missing data. Then, the replacement values were combined in different ways to reveal the variation pattern of prediction error, making it possible to predict the exact pesticide demand. The research provides the decision support for inventory management in agricultural materials enterprises.

Acknowledgments

We thank the reviewers for their thoughtful comments and suggestions. This work was supported by the National Natural Science Foundation of China (Grant No.31771679, No.31671589, No.31371533), the Special Fund for Key Program of Science and Technology of Anhui Province of China (Grant No.15czz03131, 16030701092), Project supported by the Natural

Science Foundation of the Anhui Higher Education Institutions of China (Grant No. kJ2016A836).

References

Fan ZY. The Algorithm and Model of BP Neural Network. Software Guide 2011; 10(7): 66-68.

Jiang S. Application of PSO-BP Neural Network in Prediction of Safety Stock in Coal Mine Machinery Enterprise. Coal Technology 2017; 2017(10): 305-07.

Liu JJ. Study on safety stock forecasting for China retail enterprises based on BP neural network. Logistics Technology 2012; 31(21): 326-29.

Liu T, Tiancang DU. Research on Inventory Prediction of Welded Pipe Plant Based on BP Neural Network. Journal of Beijing Institute of Petrochemical Technology 2017; 25(3): 53-57.

Mai J, Zhu Q, Wu D, Xie Y, Wang L. Back propagation neural network dehazing. Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on IEEE, 2015(40): 1433-38.

Pang XS. A Comparative research on method of missing data interpolation processing. Statistics and Decision 2012; 2012(24): 18-22.

Shu F, Tang QY, Shao ZR, Shi S, Cheng JA. Analyze influence factors of pesticide application of China. Agrochemicals 2010; 49(4): 241-45.

Spoorthy S, Thaskani S, Sood A, Chandra MG, Balamuralidhar P. Missing data ınterpolation using compressive sensing: An application for sales data gathering. Machine Intelligence and Signal Processing 2016; 2016(390): 27-35.

Thomassey S, Happiette M. A neural clustering and classification system for sales forecasting of new apparel items. Applied Soft Computing 2007; 7(4): 1170-87.

Wu T. Design and research on e-commerce system for agricultural commodities. University of Science and Technology of China, 2014.

Yang JW. Prediction of Regional Logistics based on GM/BP neural network Prediction model. Central South University, 2011.

Yuan J, Yu S. Privacy preserving back-propagation neural network learning made practical with cloud computing. IEEE Transactions on Parallel and Distributed Systems 2014; 25(1): 212-21.

Zhou H, Ding DY, Yi-Lin LI, Meng-Jie WU. Research on inventory demand forecast based on BP neural network. Information Technology 2016; 11: 38-41.