

Article

Enhanced Neural Network for Rapid Identification of Crop Water and Nitrogen Content Using Multispectral Imaging

Yaoqi Peng^{1,2,3}, Mengzhu He^{1,2}, Zengwei Zheng^{1,2,*} and Yong He^{3,*} 

¹ School of Computer and Computing Science, Hangzhou City University, Hangzhou 310015, China; pengyaoqi@yeah.net (Y.P.); hemz@hzcu.edu.cn (M.H.)

² Zhejiang Provincial Engineering Research Center for Intelligent Plant Factory, Hangzhou 310015, China

³ College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou 310015, China

* Correspondence: zhengzw@hzcu.edu.cn (Z.Z.); yhe@zju.edu.cn (Y.H.)

Abstract: Precision irrigation and fertilization in agriculture are vital for sustainable crop production, relying on accurate determination of the crop's nutritional status. However, there are challenges in optimizing traditional neural networks to achieve this accurately. This paper aims to propose a rapid identification method for crop water and nitrogen content using optimized neural networks. This method addresses the difficulty in optimizing the traditional backpropagation neural network (BPNN) structure. It uses 179 multi-spectral images of crops (such as maize) as samples for the neural network model. Particle swarm optimization (PSO) is applied to optimize the hidden layer nodes. Additionally, this paper proposes a double-hidden-layer network structure to improve the model's prediction accuracy. The proposed double-hidden-layer PSO-BPNN model showed a 9.87% improvement in prediction accuracy compared with the traditional BPNN model. The correlation coefficient R^2 for predicted crop nitrogen and water content was 0.9045 and 0.8734, respectively. The experimental results demonstrate high training efficiency and accuracy. This method lays a strong foundation for developing precision irrigation and fertilization plans for modern agriculture and holds promising prospects.

Keywords: nitrogen content; water content; double hidden layer; BP neural network; particle swarm optimization



Citation: Peng, Y.; He, M.; Zheng, Z.; He, Y. Enhanced Neural Network for Rapid Identification of Crop Water and Nitrogen Content Using Multispectral Imaging. *Agronomy* **2023**, *13*, 2464. <https://doi.org/10.3390/agronomy13102464>

Academic Editor: Shubo Wang

Received: 9 August 2023

Revised: 17 September 2023

Accepted: 19 September 2023

Published: 23 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Irrigation and fertilization are essential key factors in the crop growth stage [1] To improve the current situation of overuse in traditional farmland production, it is necessary to obtain information on crop water and nitrogen in advance to realize precise irrigation and fertilization [2,3]. The demand for water and fertilizer of crops is influenced by several factors, including but not limited to sunshine, air temperature and humidity, soil temperature and humidity, and CO₂ concentration. These factors contribute to the complexity of the crop-growing environment, which can be characterized as a time-delay system with a multitude of parameters that exhibit nonlinearity and strong interdependence. An artificial neural network has powerful self-learning, self-organizing, and self-adapting ability, which makes it possible for the network to deal with uncertain or unknown complex nonlinear systems, and by optimizing the network structure it can fully approximate any complex nonlinear relations. In the development history of the artificial neural network, for a long time there was no effective algorithm to adjust the connection weight of hidden layers [4–6]. This lasted until the error backpropagation (BP) algorithm was proposed, the weight adjustment problem of a multilayer feedforward neural network for solving nonlinear continuous functions was successfully solved, and the BP neural network was used in many applications [7,8]. For example, the remotely sensed leaf area index (LAI) and vegetation temperature condition index (VTCI) are closely related to crop growth and

crop water stress as two key variables for indicating crop growth conditions and estimating crop yields in the Guanzhong Plain, and the BP neural network and the IPSO–BP neural network were used to calculate the weight coefficients and thresholds of the VTCI and LAI at the four growth stages and to establish an integrated index, I , during the main growth period [9]. Three simulated tidal flow systems and a full system of continuous vertical flow of synthetic wastewater were treated by effluent removal with the help of a BP neural network; by comparing the influent and effluent concentrations, the results show that the ability of the BP artificial neural network model to predict nutrient concentrations in the effluent was good; there were only small errors when correlating the predicted values and the actual values [10]. Based on the monitoring data of soil moisture, soil electrical conductivity, air temperature, and light intensity, a prediction model of crop water demand based on the 4–8–1–structure BP neural network was established to guide water–saving irrigation in the crop production process [11]. Many studies have proved that a BPNN has a good effect on data prediction.

Although BPNNs have been widely used, they have some defects and deficiencies, including the following: (1) Because the learning rate is fixed, the convergence speed of the network is slow, and the network requires a long training time [12]. For some complex problems, the training time of a BP algorithm may be very long, which is mainly due to the long learning duration [13]. It can be improved by using a variable learning rate or an adaptive learning rate [14]. (2) A BP algorithm can make the weight converge to a certain value, but it does not guarantee that the value is the global minimum of the error plane because the gradient descent method may produce a local minimum value [15]. The additional momentum method can be used to solve this problem [16]. (3) BPNN learning and memory are unstable. In other words, if learning samples are added, the trained network must be trained from the beginning, and there is no memory for the previous weights and thresholds [17]. However, it can save the better weights of the prediction, classification, or clustering [18]. (4) There is no theoretical guidance for the selection of the number of layers and units of the network's hidden layer; these are generally determined by experience or by repeated experiments [19]. Motivated by these problems, we focused on determining the number of hidden layer nodes of a BP neural network based on particle swarm optimization (PSO).

In this study, field crops were taken as the research object. The spectral information of crop water and nitrogen was extracted by a multi–spectral camera, and the information of crop leaf water and nitrogen content was measured by a hand–held sensor. Considering that artificial neural networks can process nonlinear adaptive information well, by analyzing the correlation between crop spectrum characteristics and moisture nitrogen, an improved BPNN model using multispectral crop images to rapidly identify crop nitrogen and water contents was constructed. Finally, this study provides a theoretical basis for precision irrigation and fertilization of field crops. We also considered the influence of the number of hidden layers. The main contributions of this study are as follows:

- PSO was used to optimize the number of hidden layer nodes in a BP neural network, which improved the training efficiency and reduced the time and tedium of determining the number of hidden layer nodes by experience.
- In addition to increasing the number of hidden layer nodes and improving the prediction accuracy of a BP neural network, we found that a double–hidden–layer structure can effectively reduce the performance errors of the network and improve its performance.
- A prediction model of crop nitrogen and water contents based on PSO–BPNN with a double–hidden–layer structure was established. Experiments showed that the model was highly efficient at predicting the nitrogen and water contents of the crop.

The primary objectives of this study are as follows:

Section 2: to provide a comprehensive description of the data acquisition scheme, model descriptions, and mathematical preliminaries utilized in the study.

Section 3: to present and explain the optimization principle and modeling scheme employed for the model, specifically focusing on the application of particle swarm optimization (PSO) and the proposed double–hidden–layer network structure.

Section 4: to present and discuss the comparative experimental results obtained from the study, analyzing the performance and accuracy of the PSO–BPNN model in comparison to the traditional BPNN model.

Section 5: to draw conclusions based on the findings and implications of the study, summarizing the key outcomes, and discussing the potential prospects and applications of the proposed method.

2. Materials and Methods

2.1. Data Acquisition

This study aimed to determine the water and nitrogen content in a crop canopy by employing a plant nutrient analyzer (device model: YLS–D, Hubei, China) in the field. The collection methodology involved dividing the field into multiple rectangular grid areas, from which random samples were extracted. Each sample was meticulously obtained from distinct sections of the plant canopy, namely, the upper, middle, and lower regions. Subsequently, the measurements of water and nitrogen content at each sampling location were diligently recorded. To obtain accurate results, the average value of the three canopy regions was deemed representative of the overall water and nitrogen content of the crop canopy. Concurrently, a handheld multi–spectral camera (device model: RedEdge–M, Seattle, Washington, US) was employed to capture vertical images of the canopy at the sampling positions. The real–time previewing of the multi–spectral image position was enabled through Wi–Fi connectivity with a mobile phone. By maintaining a camera–lens–to–canopy distance of approximately 30 cm and ensuring proper alignment with the sensor’s five channels, images were captured and saved in a 16–bit TIFF format. Notably, the comprehensive dataset encompassed 179 sets of crop canopy multispectral images, along with the corresponding water and nitrogen content data. A detailed depiction of the data acquisition process can be found in Figure 1.

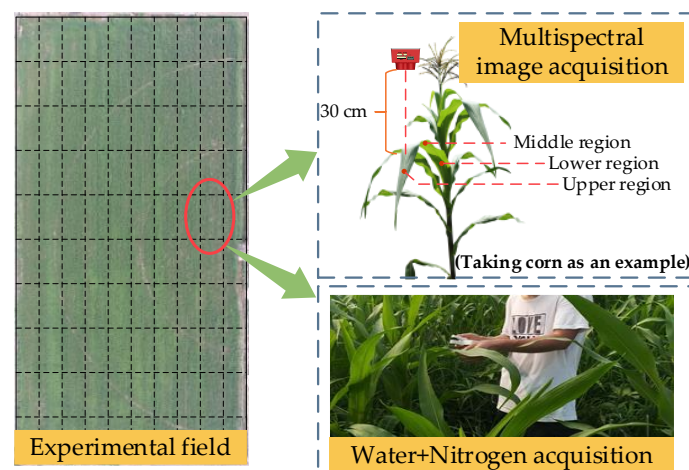


Figure 1. Crop data acquisition process.

2.2. BPNN Model Description

A BPNN is a kind of multilayer feedforward network trained by an error backpropagation algorithm [20]. It is one of the most widely used neural network models. The neurons in the input layer are responsible for receiving the input information from the outside and transmitting it to the neurons in the middle layer. The middle layer is the internal information processing layer, which is responsible for information transformation. Depending on the demands of information change, the middle layer can be designed as a single hidden layer or multiple hidden layers. The output hidden layer transmits the

information to each neuron in the output layer. After further processing, it completes a learning forward propagation process and outputs the information processing results to the outside world from the output layer. A double-hidden-layer neural network structure was used in this study. Network precision was improved by increasing the number of hidden layers [21]. The topological structure of a BP neural network is shown in Figure 2.

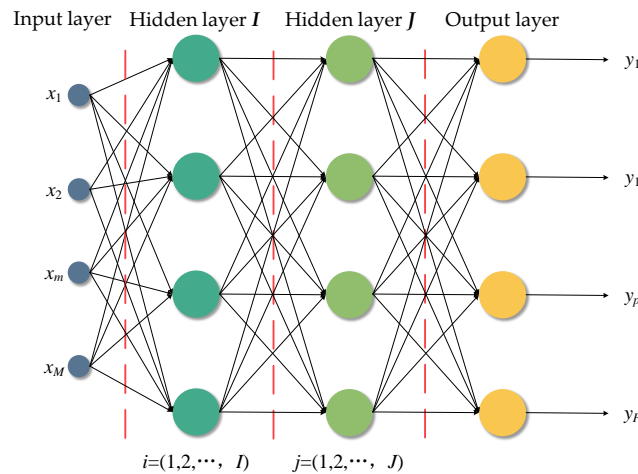


Figure 2. The topological structure of a BP neural network.

The learning process of a BP neural network comprises mainly the following parts:

- (i) Setting variables and parameters. $X_k = [x_{k1}, x_{k2}, \dots, x_{kM}]$, ($k = 1, 2, \dots, N$) is the input variable, also known as training samples, and N is the number of training samples.

$$W_{MI}(n) = \begin{bmatrix} w_{11}(n) & w_{12}(n) & \dots & w_{1M}(n) \\ w_{21}(n) & w_{22}(n) & \dots & w_{2M}(n) \\ \vdots & \vdots & \vdots & \vdots \\ w_{M1}(n) & w_{M2}(n) & \dots & w_{MI}(n) \end{bmatrix}$$

is the weight vector between the input layer and the hidden layer I in the n th iteration.

$$W_{IJ}(n) = \begin{bmatrix} w_{11}(n) & w_{12}(n) & \dots & w_{1J}(n) \\ w_{21}(n) & w_{22}(n) & \dots & w_{2J}(n) \\ \vdots & \vdots & \vdots & \vdots \\ w_{I1}(n) & w_{I2}(n) & \dots & w_{IJ}(n) \end{bmatrix}$$

is the weight vector between the hidden layer I and the hidden layer J in the n th iteration.

$$W_{JP}(n) = \begin{bmatrix} w_{11}(n) & w_{12}(n) & \dots & w_{1P}(n) \\ w_{21}(n) & w_{22}(n) & \dots & w_{2P}(n) \\ \vdots & \vdots & \vdots & \vdots \\ w_{J1}(n) & w_{J2}(n) & \dots & w_{JP}(n) \end{bmatrix}$$

is the weight vector between the hidden layer J and the output layer in the n th iteration.

$Y_k(n) = [y_{k1}(n), y_{k2}(n), \dots, y_{kP}(n)]$, ($k = 1, 2, \dots, N$) is the actual output of the network in the n th iteration. $d_k = [d_{k1}, d_{k2}, \dots, d_{kP}]$, ($k = 1, 2, \dots, N$) is the desired output.

η is the learning rate, and n is the number of iterations.

- (ii) Initialization. Assign a smaller random nonzero value to $W_{MI}(0)$, $W_{IJ}(0)$, $W_{JP}(0)$, and $n = 0$.
- (iii) Random input sample X_k .
- (iv) The input sample X_k , input signal u , and output signal v of each layer of the BPNN are calculated forward, where $v_p^P(n) = y_{kp}(n)$, $p = 1, 2, \dots, P$.
- (v) Calculate the error $E(n)$ from the expected output d_k and the actual output $Y_k(n)$ obtained in the previous step to judge whether it meets the requirements. If it meets the requirements, go to step viii; if not, go to step vi.

- (vi) Determine whether the $n + 1$ is greater than the maximum number of iterations. If it is greater, go to step viii. If it is not greater, the local gradient δ of each layer of neurons is inversely calculated for the input sample X_k . The equations are

$$\delta_p^P(n) = y_p(n)(1 - y_p(n))(d_p(n) - y_p(n)), p = 1, 2, \dots P \quad (1)$$

$$\delta_j^J(n) = f'(u_j^J(n)) \sum_{p=1}^P \delta_p^P(n) w_{jp}(n), j = 1, 2, \dots J \quad (2)$$

$$\delta_i^I(n) = f'(u_i^I(n)) \sum_{j=1}^J \delta_j^J(n) w_{ij}(n), i = 1, 2, \dots I \quad (3)$$

- (vii) Calculate the weight correction Δw and correct the weight; if $n = n + 1$, go to step iv.

$$\Delta w_{jp}(n) = \eta \delta_p^P(n) v_j^J(n) w_{jp}(n) \quad w_{jp}(n+1) = w_{jp}(n) + \Delta w_{jp}(n) \quad j = 1, 2, \dots J; p = 1, 2, \dots P \quad (4)$$

$$\Delta w_{ij}(n) = \eta \delta_j^J(n) v_i^I(n) w_{ij}(n) \quad w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n) \quad i = 1, 2, \dots I; j = 1, 2, \dots J \quad (5)$$

$$\Delta w_{mi}(n) = \eta \delta_i^I(n) x_{km}(n) w_{mi}(n) \quad w_{mi}(n+1) = w_{mi}(n) + \Delta w_{mi}(n) \quad m = 1, 2, \dots M; i = 1, 2, \dots I \quad (6)$$

- (viii) Judge whether all the training samples have been learned. If they have, the learning process is finished. If they have not, go to step iii.

2.3. Application Principle of the Particle Swarm Optimization Algorithm

PSO has the characteristics of evolutionary computation and swarm intelligence. Similar to other algorithms, PSO can search for the best solution in complex space through cooperation and competition among individuals [22].

In a PSO algorithm, the solution of each optimization problem is regarded as a “bird” or “particle” in the search space [23]. At the beginning of the algorithm, the initial solution is generated; that is, the population composed of m particles is randomly initialized in the feasible solution space, where the position $Z_i = \{z_{i1}, z_{i2}, \dots, z_{in}\}$ of each particle represents a solution to the problem, and a new solution is searched for according to the objective function calculation. In each iteration, the particle tracks two extrema to update itself; one is the best solution p_{id} found by the particle itself, and the other is the best solution p_{gd} found by the entire population, which is the global extremum. Also, each particle has a velocity of $V_i = \{v_{i1}, v_{i2}, \dots, v_{in}\}$. When the two optimal solutions are found, each particle updates its velocity according to Equation (7):

$$\begin{aligned} v_{id} &= wv_{id}(t) + \eta_1 \text{rand}() [p_{id} - z_{id}(t)] + \eta_2 \text{rand}() [p_{gd} - z_{id}(t)] \\ z_{id}(t+1) &= z_{id}(t) + v_{id}(t+1) \end{aligned} \quad (7)$$

where $Z_{id}(t+1)$ is the velocity of the i th particle in the d dimension in the $t+1$ iteration, w is the inertia weight, η_1 and η_2 are acceleration constants, and $\text{rand}()$ is a random number between 0 and 1. Also, the upper limit of the velocity can be set to prevent the particle velocity from being too great; that is, when $v_{id}(t+1) > v_{\max}$, $v_{id}(t+1) = v_{\max}$; when $v_{id}(t+1) < -v_{\max}$, $v_{id}(t+1) = -v_{\max}$.

From the updated equation of particles, we can see that the moving direction of particles is determined by three parts: their original velocity v_{id} ; the distance $p_{id} - z_{id}(t+1)$ from their best experience; and the distance $p_{gd} - z_{id}(t)$ from the best experience of the group, and their relative importance is determined by the weight coefficients w , η_1 , and η_2 . When the end condition of the algorithm is reached, that is, a sufficiently optimal solution is found or the maximum number of iterations is reached, the algorithm ends. The basic flow of PSO is shown in Figure 3.

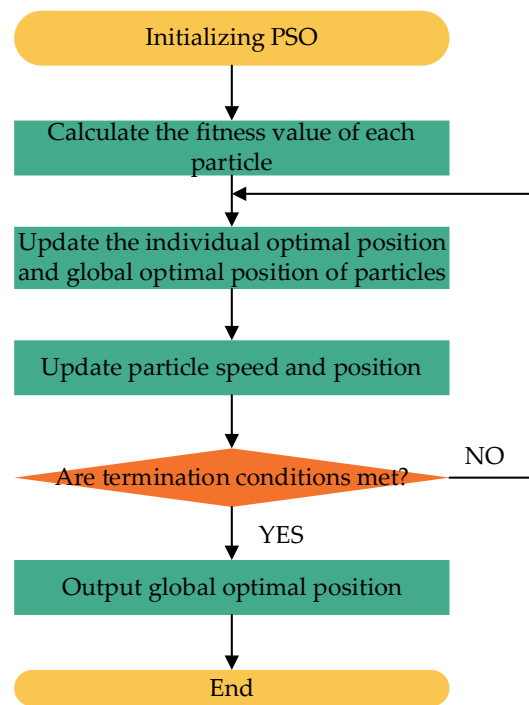


Figure 3. The basic flow of PSO.

3. Construction of BPNN Model Based on PSO Optimization

3.1. Optimization of BPNN by PSO Algorithm

To solve the defects of a BPNN, we combined a PSO algorithm with a BPNN algorithm and applied the PSO algorithm to optimize the initial weight and the threshold values of the neural network. The overall algorithm flow is shown in Figure 4.

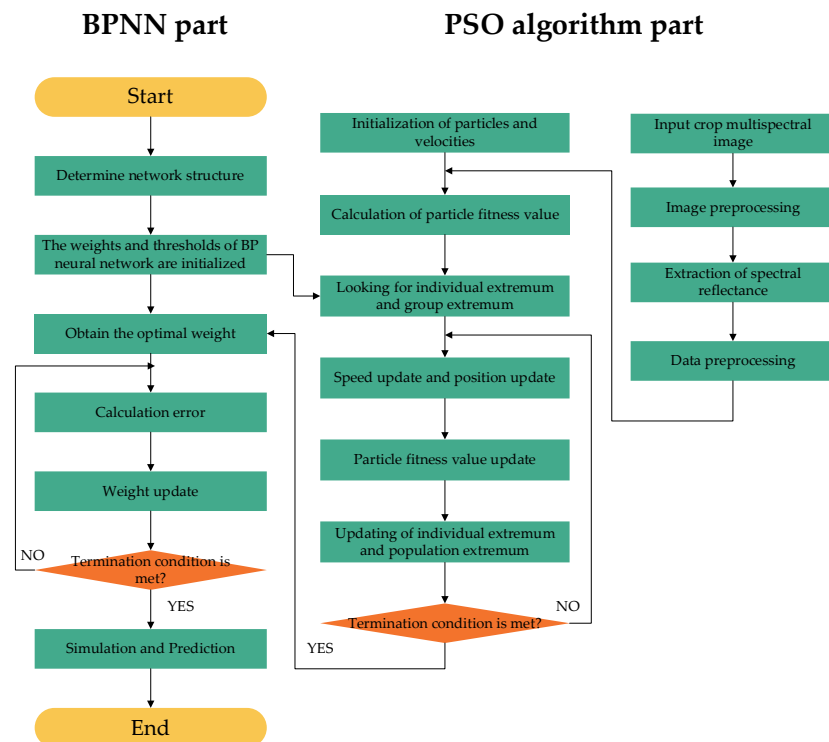


Figure 4. Combined PSO–BP algorithm flow chart.

The implementation steps of the PSO–BP algorithm are

Step 1. The structure and parameters of the BP neural network are initialized.

Step 2. Combined with the connection weights of the BP neural network, the network structure of the PSO is initialized. First, the weight vector $W = \{w_1, w_2, \dots, w_n\}$ of the BP network is constructed as the space particle in the PSO optimization algorithm, and then the parameters of the PSO network are set, namely, inertia weight, the acceleration constant, particle speed, and position in Equations (8) and (9).

Step 3. The speed and position of the weighted particles are updated. The particles start from an initial position X_{id} in space with a certain initial velocity V_{id} , where i is the number and d is the dimension. In the process of particle motion, the velocity and position will change constantly, and the update formulas are Equations (8) and (9), respectively:

$$V_{id}^{k+1} = \omega V_{id}^k + c_1 r_1 (P_{id}^k - X_{id}^k) + c_2 r_2 (P_{gd}^k - X_{id}^k) \tag{8}$$

$$X_{id}^{k+1} = X_{id}^k + V_{id}^{k+1} \tag{9}$$

where Equation (8) is the velocity of the particle and Equation (9) is the position of the particle.

Step 4. Find the global optimal extremum. First, the fitness function value of each particle in the space is calculated. When the particle is iterated many times, a new fitness function value is calculated. If the fitness function value of the new particle is better than the current value, the individual extreme value p_{best} and the population extreme value g_{best} are updated until the best extreme value is found. The mean square error of the BP neural network on the training set is taken as the fitness function, and the calculation method is Equation (10).

$$E(x_p) = \frac{1}{N} \sum_{p=1}^n \sum_{k=0}^m (y_{pk}(x_p) - t_{pk})^2 \tag{10}$$

where x_p is the input sample of group p , $p = 1, 2, \dots, n$; y_{pk} is the k th output of input x_p sample; and t_{pk} is the expected value of the k th output of input x_p sample, $k = 1, 2, \dots, m$.

Step 5. Weight optimization is achieved. Compare the best fitness function value obtained in **Step 4** with the preset objective, or judge whether the maximum iteration times have been reached. If the requirements are met, it indicates that the global best weight has been found and the operation is finished.

Step 6. In the PSO--BP network, the outputs of the hidden and output layers are calculated as follows:

$$x^l = f\left(\sum_{i=0}^2 w_{ij} - \theta_1\right) d = f\left(\sum_{j=0}^2 w_{jk} x^l - \theta_2\right) \tag{11}$$

$$f(u) = \frac{1}{1 + e^{-u}} = \frac{1}{1 + e^{-(\sum w_j x_j - \theta_j)}} \tag{12}$$

Step 7. Error judgment. Calculate whether the error function meets the expectation. If so, the network training ends and retains the trained weight; otherwise, the error of each layer is calculated layer by layer. The calculation of expectation and error is as follows:

$$e_p = \frac{1}{2} \sum_k (y_k - \hat{y}_k)^2 \tag{13}$$

$$\delta_{jk}^{p1} = (t_k^{p1} - d_k^{p1}) d_k^{p1} (1 - d_k^{p1}) \delta_{ij}^{p1} = \sum_{k=0}^{p1} \delta_{jk}^{p1} w_{jk} x^{p1} (1 - x^{p1}) \tag{14}$$

Step 8. Adjust the weight of each layer of the network; the specific calculation is

$$w_{jk}(n_0 + 1) = w_{jk}(n_0) + \eta \sum_{p_1=1}^p \delta_{jk}^{p_1} x^{p_1} w_{ij}(n_0 + 1) = w_{ij}(n_0) + \eta \sum_{p_1=1}^p \delta_{ij}^{p_1} x^{p_1} \quad (15)$$

Step 9. After the adjustment, continue to input the sample and repeat the calculation process in **Step 6** with the new weight. Once the error meets the requirements of Equation (14), the training is stopped.

Step 10. Save the trained neural network models and predict the crop nitrogen and moisture contents of the trained neural network.

3.2. Construction of Prediction Model of Crop Water and Nitrogen Contents Based on BPNN with Double Hidden Layers

The performance of the PSO algorithm is affected by various interaction parameters, such as group size N , inertia factor ω , learning factors c_1 and c_2 , maximum speed v_{\max} , and maximum iteration number G_k . The range of the population size is generally from 20 to 50. If $\omega = 0$, then the adjustment of particle velocity is related only to the current position and the historical best position, and the change in velocity size has nothing to do with it [24]. When the inertia factor ω is added, particles can effectively explore other parts of space, not only in the vicinity of the individual and global best positions but also in the function of global and local exploration. When ω is large, the global search ability is strong; when ω is small, the local search ability is strong [25]. Generally, $c_1 = c_2 = 2$. Ambroziak's research shows that if we want to obtain good results and simplify the operation, c_1 and c_1 are better as constants [26]. These parameters must be given artificially at the beginning of training. The specific parameter values of this study are shown in Table 1.

Table 1. Parameter selection of PSO algorithm.

N	ω	c_1	c_2	v_{\max}	G_k
50	0.1	2	2	0.5	200

In this study, 179 sets of multispectral images of crop leaves were collected to extract the reflectance value, and the nitrogen and water contents of the leaves were measured by a plant nutrient analyzer. Finally, the reflectance of the blue, green, red, near-infrared (NIR), and RedEdge bands, as well as crop nitrogen and water contents, were obtained—a total of seven variables. A correlation analysis of the seven variables is shown in Figure 5. The correlation between the reflectance of each spectral band and the nitrogen and water contents of the crops was not strong, showing a nonlinear mapping relation. It is therefore more suitable to use a neural network to build a prediction model.

Five neurons were in the input layer of the BPNN—that is, the reflectance of five spectral bands—and two neurons were in the output layer; namely, the nitrogen and water contents of the crops. Because the units of input data and output data were different, to facilitate data training the data had to be normalized into a dimensionless form before modeling. The training data in this study were normalized to between 0.001 and 0.999. The naturalization equation is given by Equation (16).

$$P = \frac{p - p_{\min}}{p_{\max} - p_{\min}} \times 0.998 + 0.001 \quad (16)$$

where p is the original data, p_{\min} is the minimum value of the data with the same dimension, p_{\max} is the maximum value of the data with the same dimension, and P is the normalized data.

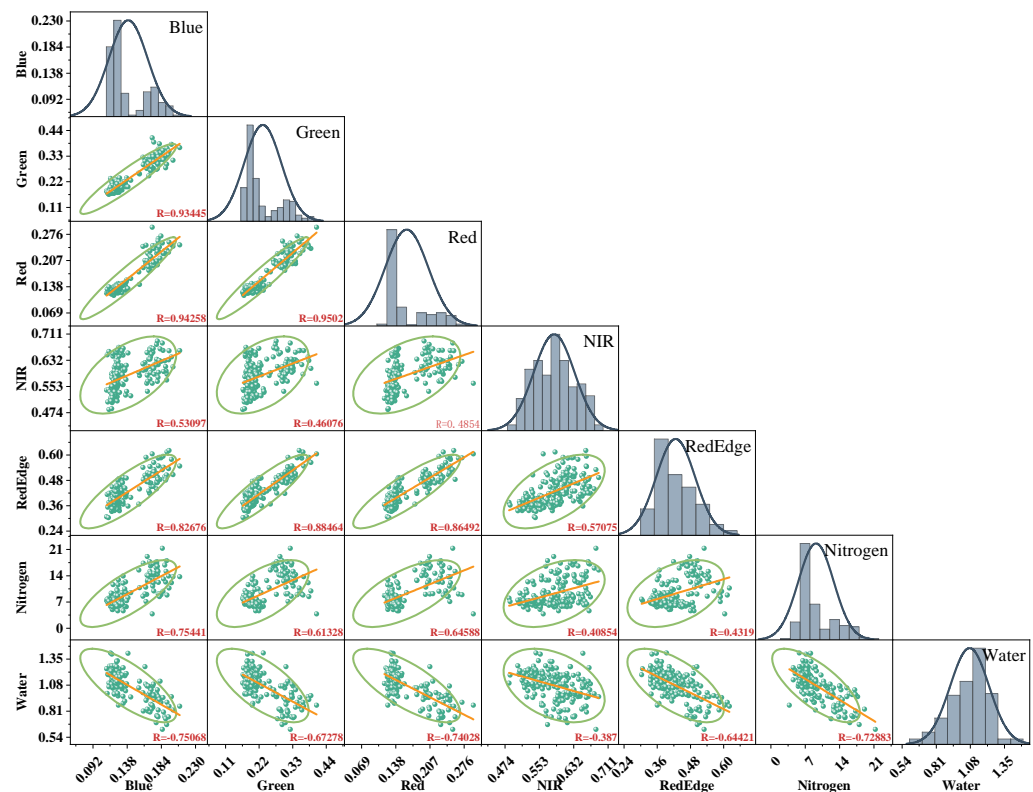


Figure 5. Correlation analysis of seven variables.

In the process of modeling with the BPNN, the number of hidden layer neurons in the network structure could not be determined. Generally, the greater the number of hidden layers in a neural network, the stronger its nonlinear mapping capability. Upon reaching an appropriate number of neurons, further increasing the number would not help much to improve the accuracy of the network but would increase the amount of calculation [27]. Currently, the number of neurons is determined mainly by trial calculation via an empirical equation; the general empirical formulas are given by Equations (17) and (18).

$$l < \sqrt{(m+n)} + a \quad (17)$$

$$l < 2m + 1 \quad (18)$$

l is the number of hidden neurons, m is the number of neurons in the input layer, n is the number of neurons in the output layer, and a is a constant from 0 to 10 [28].

According to the empirical formulas of Equations (17) and (18), it was determined that the number of neurons in the first hidden layer ranged from 2 to 13. The networks with different hidden layers were trained 10 times, and the numbers of epochs and mean squared error (MSE) values were recorded. The recorded results are shown in Figure 6. In Figure 6, the color of the 3D sphere represents the number of neurons in the hidden layer, and the size of the sphere represents the MSE. After the trial calculation, when the number of neurons in the hidden layer was 12, the correlation coefficient R -value of the trained BPNN was the largest, while the number of iterations and the MSE value were low.

This study aimed to establish a BPNN with double hidden layers. Through trial calculation, we determined the number of nodes in the first hidden layer to be 12. It was found that the best ratio of the number of nodes in the first hidden layer and the second hidden layer was 3:1 in the case of high-dimensional input [29]. Therefore, the number of nodes in the second hidden layer of the neural network was determined to be 4. Finally,

a BPNN with a 5–12–4–2 structure was established. The structure diagram is shown in Figure 7.

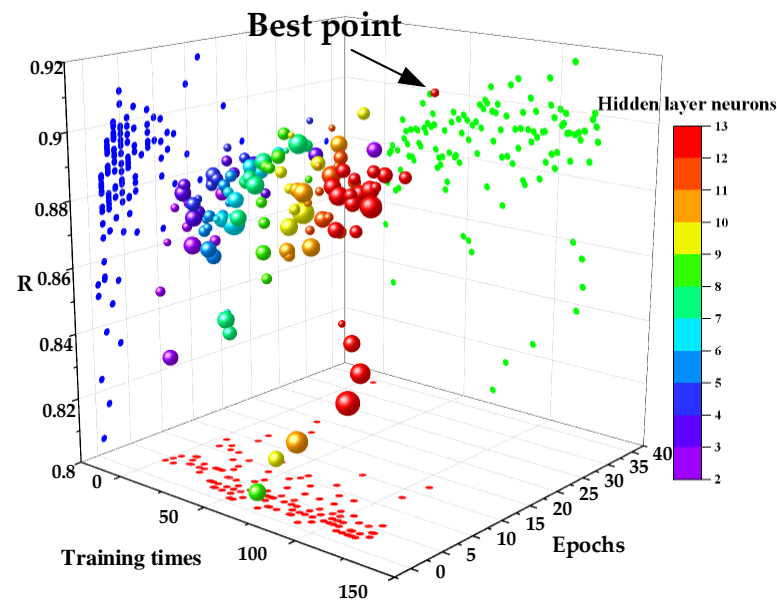


Figure 6. First hidden –layer trial results.

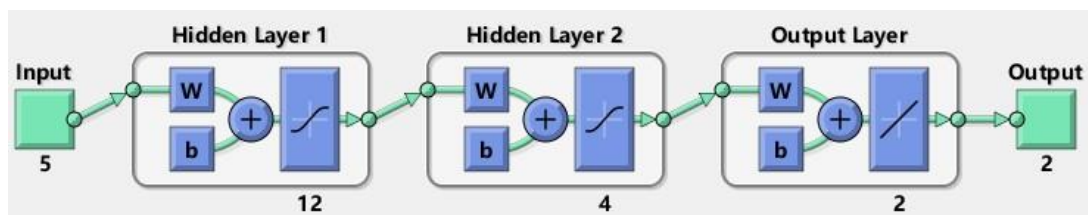


Figure 7. BPNN with a 5–12–4–2 structure (It contains 5 input neurons, with 12 neurons in the first hidden layer, 4 neurons in the second hidden layer, and 2 neurons in the output layer).

Below is the pseudo–code for optimizing a BP neural network using PSO:

1. Initialization:
 - Define the population size and the maximum number of iterations.
 - Randomly initialize the position and velocity of particles in the search space.
 - Initialize the global best position and fitness value.
2. Particle Movement and Update:
 - For each iteration, update the velocity and position of each particle using the PSO equations.
 - Apply velocity and position limits if necessary.
 - Evaluate the fitness of each particle using the BP neural network with the current position as the weights.
 - Update the personal and global best positions if a particle finds better solutions.
3. Termination Condition:
 - Stop the process when either the maximum number of iterations is reached, or a desired fitness value is achieved.
4. Return the Best Solution:
 - After the termination condition is met, return the position of the particle with the best fitness value as the optimized weights for the BP neural network.

4. Comparative Experimental Results and Analysis

Next, we trained 150 groups of samples, and the remaining 29 groups were used for verification. We compared and analyzed the performances of a traditional BPNN, a single-hidden-layer BP neural network optimized by PSO (PSO-1H-BPNN), and a double-hidden-layer BP neural network optimized by PSO (PSO-2H-BPNN), and also determined the prediction accuracies of the three neural networks.

4.1. Performance Analysis of the Neural Networks

Three kinds of neural networks were trained by 150 groups of samples. The convergence and network performances of the training are shown in Figures 8 and 9. In Figure 8, we can see that the MSE of the BPNN was the largest and that of PSO-2H-BPNN was the smallest. This shows that a BP neural network with a double-hidden-layer structure optimized by PSO can effectively reduce the number of training errors. Also, increasing the number of hidden layers increases the number of iterations, but the increase is not obvious. This also confirms the results of a previous study that increasing the number of hidden layers will increase the training time [30].

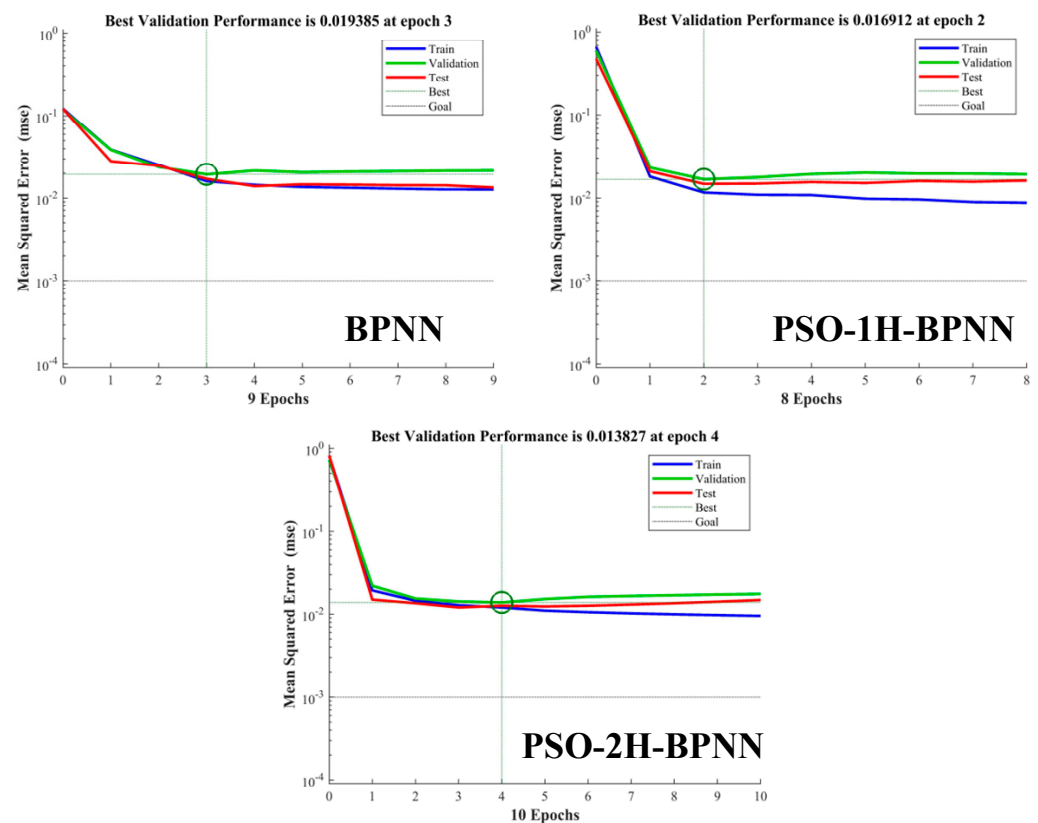


Figure 8. Convergence process of neural network training.

In Figure 9, we can see that the decision coefficient R of the three neural networks gradually increased after training, and the decision coefficient R of PSO-2H-BPNN reached 0.92978. Compared with a conventional BPNN, the network performances of PSO-1H-BPNN and PSO-2H-BPNN improved by 3.97% and 9.87%, respectively, which indicates that PSO-2H-BPNN had the best performance of the three kinds of neural networks.

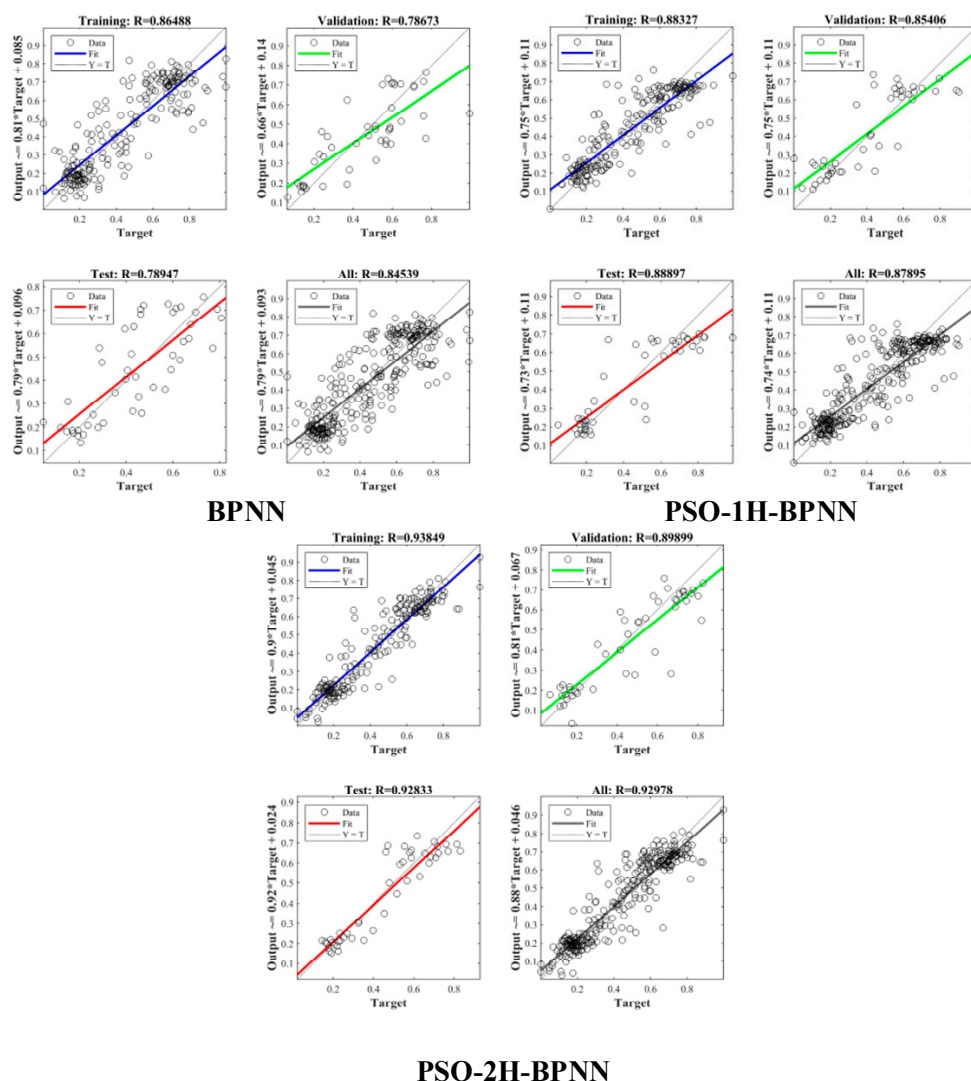
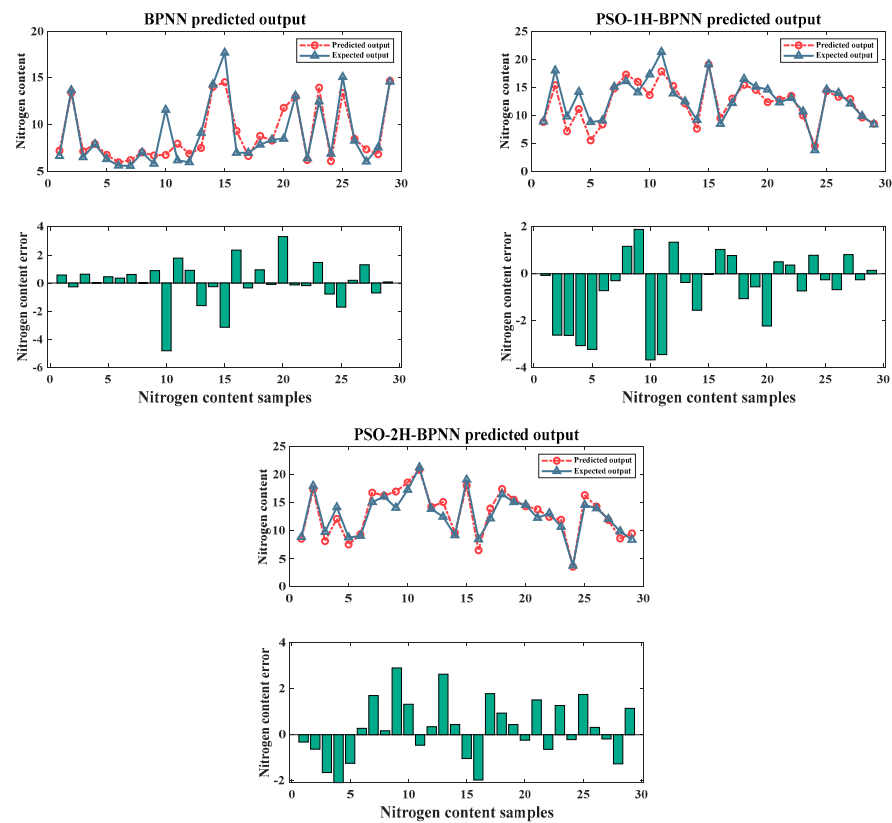


Figure 9. Neural network performance.

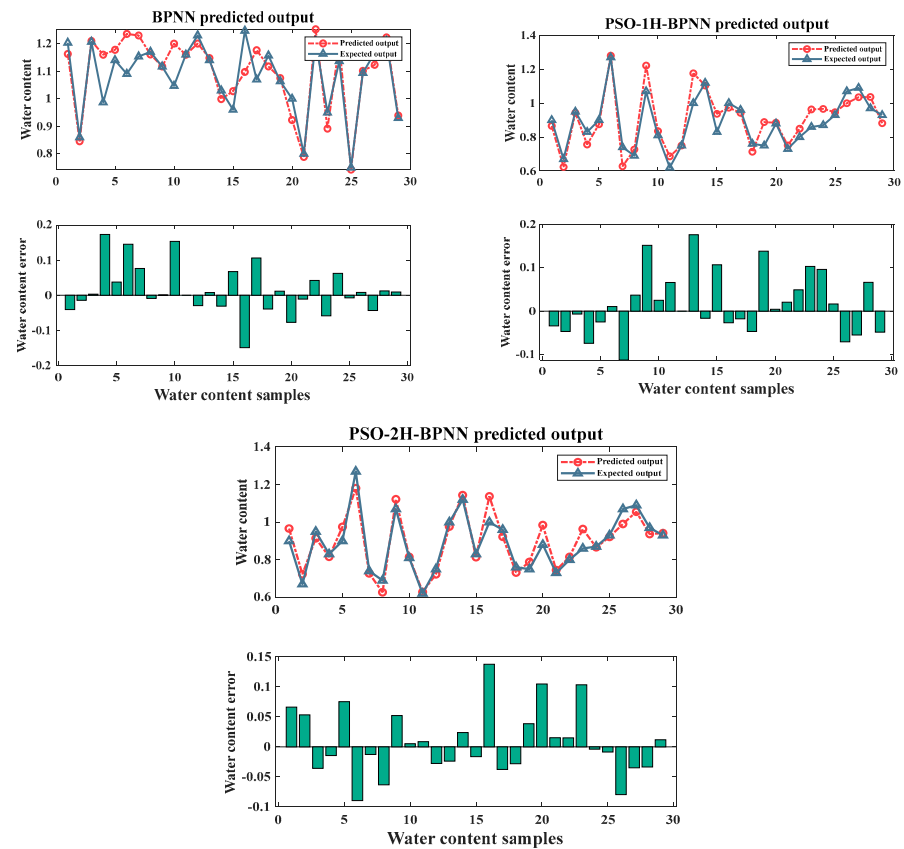
4.2. Prediction Accuracy Analysis of Three Kinds of Neural Networks

We used three kinds of neural networks to simulate the remaining 29 groups of samples. The results of the predicted and expected outputs are shown in Figure 10. From Figure 10, it can be seen that in the simulation of either nitrogen content or water content, the predicted output and the expected output had the strongest consistency in PSO-2H-BPNN. Also, the deviation degree of the PSO-2H-BPNN prediction error was the smallest of the three kinds of neural networks.

Finally, a linear correlation analysis was performed between the predicted output and the expected output of the sample, as shown in Figure 11. From Figure 11, it can be seen that for the predicted nitrogen content of the crop the correlation coefficient R^2 of the BPNN, PSO-1H-BPNN, and PSO-2H-BPNN reached 0.7995, 0.8352, and 0.9045, respectively, and the correlation coefficient R^2 of PSO-2H-BPNN was the highest. On the predicted water content of the crop, the correlation coefficient R^2 of the BPNN, PSO-1H-BPNN, and PSO-2H-BPNN reached 0.7533, 0.88099, and 0.8734, respectively, and the correlation coefficient R^2 of PSO-2H-BPNN was also the highest. In the same kind of neural network prediction, the accuracy of predicting crop nitrogen content was higher than that of predicting crop water content.

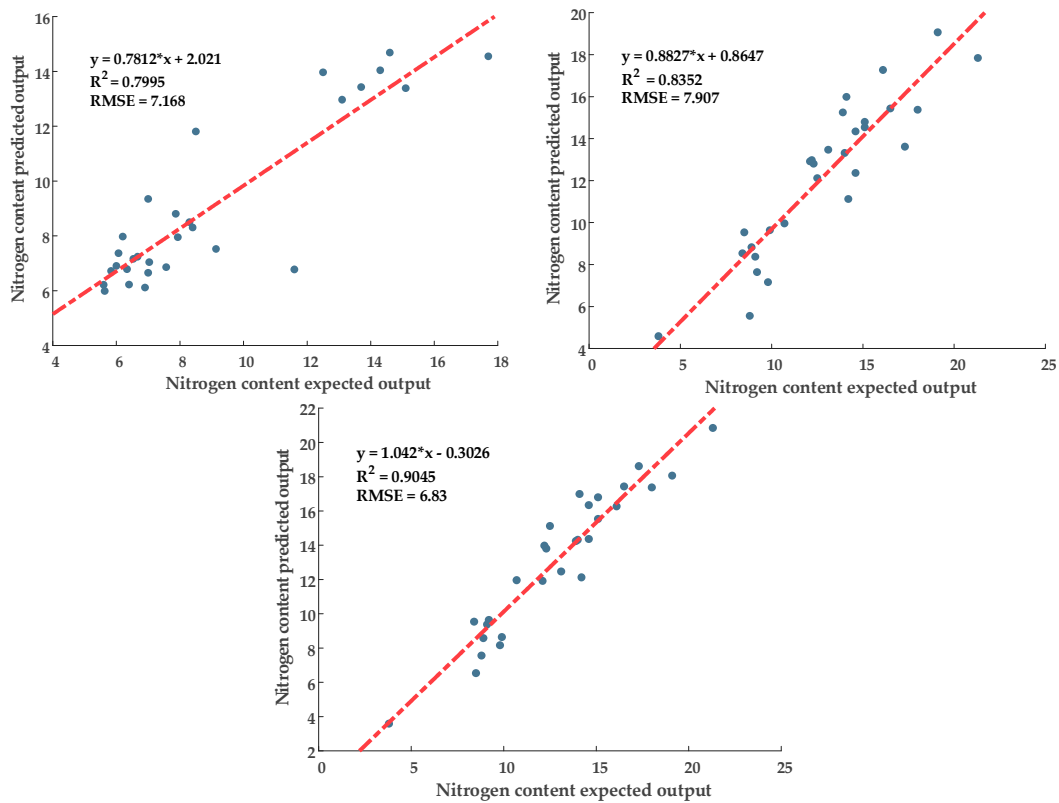


(a) Simulation verification of crop nitrogen content.

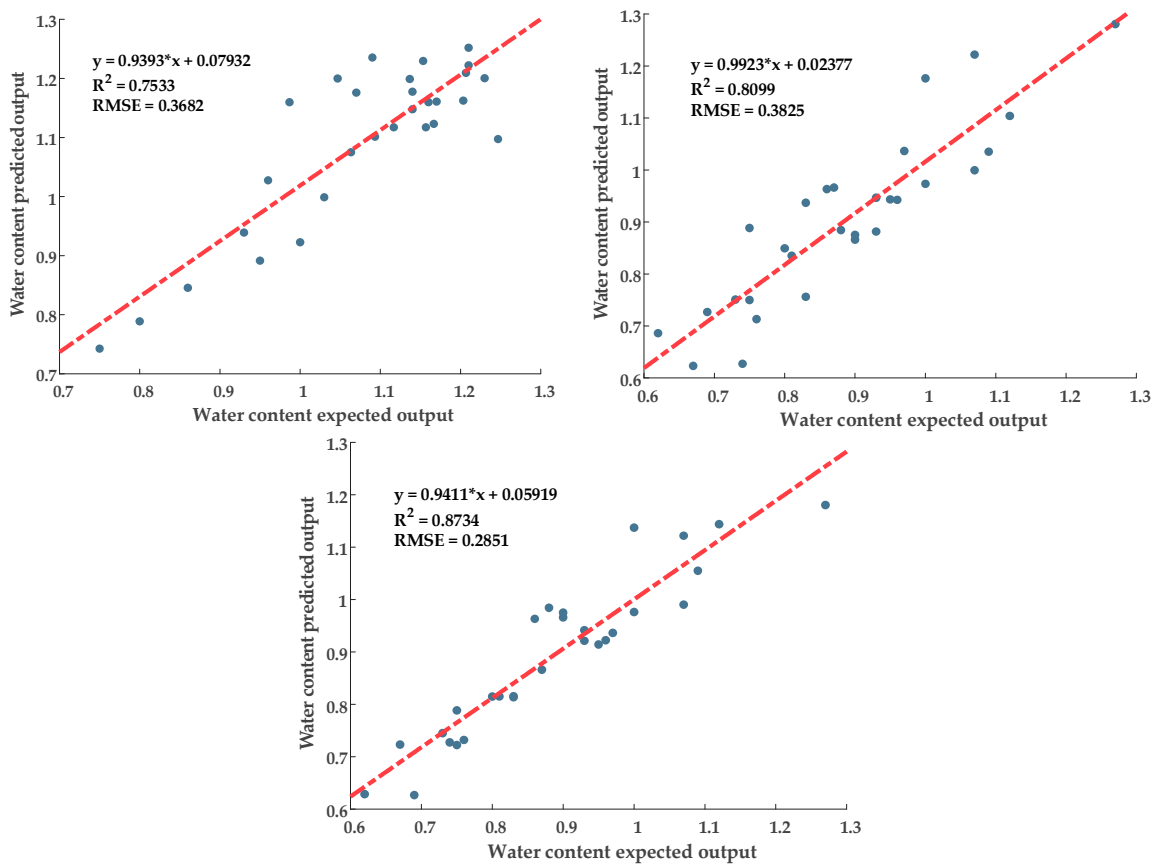


(b) Simulation verification of crop water content.

Figure 10. Simulation results of three kinds of neural networks.



(a) Linear correlation analysis of crop nitrogen content.



(b) Linear correlation analysis of crop water content.

Figure 11. Linear correlation analysis of three kinds of neural networks.

5. Conclusions

In this study, we designed a double–hidden–layer BP neural network optimized by a PSO algorithm. Compared with a conventional BPNN, the network performance of PSO–2H–BPNN was improved by 9.87%. On the predicted nitrogen content of the crop, the correlation coefficient R^2 of PSO–2H–BPNN reached 0.9045, and on the predicted water content of the crop, the correlation coefficient R^2 of PSO–2H–BPNN reached 0.8734. Both had the highest determination coefficient. For the same kind of neural network prediction, the accuracy of predicting the crop nitrogen content was higher than that of the crop water content.

Although the network can quickly identify crop moisture and nitrogen content, it has a high requirement for collecting multi–spectral images of crop canopy. For canopy multispectral image data collected under sufficient lighting conditions, high accuracy can be easily achieved, but canopy multispectral image data collected under cloudy or evening conditions may reduce recognition accuracy. In future studies, we will try using a different algorithm to replace PSO, continue to optimize BP neural networks, and continually improve the prediction accuracy of the network.

Author Contributions: Writing—original draft preparation, Y.P.; writing—review and editing, M.H.; project administration, Z.Z.; supervision, Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Zhejiang Provincial Natural Science Foundation of China (Grant No. LGN21F020002).

Data Availability Statement: Not applicable.

Acknowledgments: The authors are thankful for the Zhejiang Provincial Natural Science Foundation for funding this work through the Grant No. (LGN21F020002).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Alcaide Zaragoza, C.; González Perea, R.; Fernández García, I.; Camacho Poyato, E.; Rodríguez Díaz, J.A. Open source application for optimum irrigation and fertilization using reclaimed water in olive orchards. *Comput. Electron. Agric.* **2020**, *173*, 105407. [[CrossRef](#)]
- Perich, G.; Aasen, H.; Verrelst, J.; Argento, F.; Walter, A.; Liebisch, F. Crop Nitrogen Retrieval Methods for Simulated Sentinel–2 Data Using In–Field Spectrometer Data. *Remote Sens.* **2021**, *13*, 2404. [[CrossRef](#)] [[PubMed](#)]
- Rattalino Edreira, J.I.; Guilpart, N.; Sadras, V.; Cassman, K.G.; van Ittersum, M.K.; Schils, R.L.M.; Grassini, P. Water productivity of rainfed maize and wheat: A local to global perspective. *Agric. For. Meteorol.* **2018**, *259*, 364–373. [[CrossRef](#)]
- Heydarizad, M.; Gimeno, L.; Minaei, M.; Shahsavan Gharehghouini, M. Stable Isotope Signatures in Tehran’s Precipitation: Insights from Artificial Neural Networks, Stepwise Regression, Wavelet Coherence, and Ensemble Machine Learning Approaches. *Water* **2023**, *15*, 2357. [[CrossRef](#)]
- Kamal, S.; Prajapati, H.S.; Cahill, N.D.; Hailstone, R.K. Probe Aberration Correction in Scanning Electron Microscopy Using Artificial Neural Networks. *Microsc. Microanal.* **2023**, *29*, 739–740. [[CrossRef](#)]
- Li, B. A Productivity Prediction Method Based on Artificial Neural Networks and Particle Swarm Optimization for Shale–Gas Horizontal Wells. *Fluid. Dyn. Mater. Process.* **2023**, *19*, 2729–2748. [[CrossRef](#)]
- Wei, L.; Xv, S.; Li, B. Short–term wind power prediction using an improved grey wolf optimization algorithm with back–propagation neural network. *Clean. Energy* **2022**, *6*, 288–296. [[CrossRef](#)]
- Xu, S.; Wan, H.; Zhao, X.; Zhang, Y.; Yang, J.; Jin, W.; He, Y. Optimization of extraction and purification processes of six flavonoid components from Radix Astragali using BP neural network combined with particle swarm optimization and genetic algorithm. *Ind. Crops Prod.* **2022**, *178*, 114556. [[CrossRef](#)]
- Tian, H.; Wang, P.; Tansey, K.; Zhang, S.; Zhang, J.; Li, H. An IPSO–BP neural network for estimating wheat yield using two remotely sensed variables in the Guanzhong Plain, PR China. *Comput. Electron. Agric.* **2020**, *169*, 105180. [[CrossRef](#)]
- Li, W.; Cui, L.; Zhang, Y.; Cai, Z.; Zhang, M.; Xu, W.; Zhao, X.; Lei, Y.; Pan, X.; Li, J.; et al. Using a Backpropagation Artificial Neural Network to Predict Nutrient Removal in Tidal Flow Constructed Wetlands. *Water* **2018**, *10*, 83. [[CrossRef](#)]
- Peng, Y.; Xiao, Y.; Fu, Z.; Dong, Y.; Zheng, Y.; Yan, H.; Li, X. Precision irrigation perspectives on the sustainable water–saving of field crop production in China: Water demand prediction and irrigation scheme optimization. *J. Clean. Prod.* **2019**, *230*, 365–377. [[CrossRef](#)]

12. Wan, T.; Bai, Y.; Wang, T.; Wei, Z. BPNN–based optimal strategy for dynamic energy optimization with providing proper thermal comfort under the different outdoor air temperatures. *Appl. Energy* **2022**, *313*, 118899. [[CrossRef](#)]
13. Zhang, D.; Lou, S. The application research of neural network and BP algorithm in stock price pattern classification and prediction. *Future Gener. Comput. Syst.* **2021**, *115*, 872–879. [[CrossRef](#)]
14. Takase, T.; Oyama, S.; Kurihara, M. Effective neural network training with adaptive learning rate based on training loss. *Neural Netw.* **2018**, *101*, 68–78. [[CrossRef](#)] [[PubMed](#)]
15. Knoll, C.; Mehta, D.; Chen, T.; Pernkopf, F. Fixed Points of Belief Propagation—An Analysis via Polynomial Homotopy Continuation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2124–2136. [[CrossRef](#)] [[PubMed](#)]
16. Aprilia, B.; Marzuki; Taufiq, I. Performance of backpropagation artificial neural network to predict el nino southern oscillation using several indexes as onset indicators. *J. Phys. Conf. Ser.* **2021**, *1876*, 12004. [[CrossRef](#)]
17. Wei, M.; Hu, X.; Yuan, H. Residual displacement estimation of the bilinear SDOF systems under the near–fault ground motions using the BP neural network. *Adv. Struct. Eng.* **2022**, *25*, 552–571. [[CrossRef](#)]
18. Zhou, C.; Gui, S.; Liu, Y.; Ma, J.; Wang, H. Fault Location of Distribution Network Based on Back Propagation Neural Network Optimization Algorithm. *Processes* **2023**, *11*, 1947. [[CrossRef](#)]
19. Zhang, J.; Gao, P.; Fang, F. An ATPSO–BP neural network modeling and its application in mechanical property prediction. *Comput. Mater. Sci.* **2019**, *163*, 262–266. [[CrossRef](#)]
20. Zhu, H.; Liu, J.; Yu, J.; Yang, P. Artificial neural network–based predictive model for supersonic ejector in refrigeration system. *Case Stud. Therm. Eng.* **2023**, *49*, 103313. [[CrossRef](#)]
21. Mahadeva, R.; Kumar, M.; Patole, S.P.; Manik, G. Employing artificial neural network for accurate modeling, simulation and performance analysis of an RO–based desalination process. *Sustain. Comput. Inform. Syst.* **2022**, *35*, 100735. [[CrossRef](#)]
22. Mokarram, V.; Banan, M.R. A new PSO–based algorithm for multi–objective optimization with continuous and discrete design variables. *Struct. Multidiscip. Optim.* **2018**, *57*, 509–533. [[CrossRef](#)]
23. Meng, Z.; Zhong, Y.; Mao, G.; Liang, Y. PSO–sono: A novel PSO variant for single–objective numerical optimization. *Inf. Sci.* **2022**, *586*, 176–191. [[CrossRef](#)]
24. Merugumalla, M.K.; Navuri, P.K. Chaotic inertia weight and constriction factor–based PSO algorithm for BLDC motor drive control. *Int. J. Process Syst. Eng.* **2019**, *5*, 30–52. [[CrossRef](#)]
25. Fan, Y.; Zhang, Y.; Guo, B.; Luo, X.; Peng, Q.; Jin, Z. A hybrid sparrow search algorithm of the hyperparameter optimization in deep learning. *Mathematics* **2022**, *10*, 3019. [[CrossRef](#)]
26. Ambroziak, A.; Chojecki, A. The PID controller optimisation module using Fuzzy Self–Tuning PSO for Air Handling Unit in continuous operation. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105485. [[CrossRef](#)]
27. Wen, S.; Xiao, S.; Yang, Y.; Yan, Z.; Zeng, Z.; Huang, T. Adjusting learning rate of memristor–based multilayer neural networks via fuzzy method. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2018**, *38*, 1084–1094. [[CrossRef](#)]
28. Acheampong, A.O.; Boateng, E.B. Modelling carbon emission intensity: Application of artificial neural network. *J. Clean. Prod.* **2019**, *225*, 833–856. [[CrossRef](#)]
29. Majalca, R.; Acosta, P.R. Convex Hulls and the size and the Size of the Hidden Layer in a MLP Based Classifier. *IEEE Lat. Am. Trans.* **2019**, *17*, 991–999. [[CrossRef](#)]
30. Tian, J.; Liu, Y.; Zheng, W.; Yin, L. Smog prediction based on the deep belief–BP neural network model (DBN–BP). *Urban. Clim.* **2022**, *41*, 101078. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.