Projectile Impact Point Prediction Based on Genetic Algorithm BP Neural Network

Sen Wang a, Liangming Wang, Jian Fu
School of Energy and Power Engineering, Nanjing University of Science and Technology, Nanjing, China
a 742226731@qq.com

Abstract. Aiming at the problems of long calculating time and cumulative error in traditional projectile impact point prediction methods, a prediction method based on BP neural network is proposed in this paper. In order to avoid the local minimum and slow convergence speed, genetic algorithm is used to optimize the initial weights and thresholds of the BP neural network. The model of genetic algorithm BP neural network for predicting the impact point of projectile is obtained by training the model with a large number of flight state parameters and impact point information, and the simulation test is carried out. The simulation results show that the method can predict the impact point with high accuracy, and it is superior to the numerical integration method in calculating time. Therefore, it is reasonable and feasible to use this method to predict the impact point of projectile, which provides a reference for the practical application of projectile impact point prediction.

1. Introduction
In modern warfare, precise attack on targets has become a key factor affecting the outcome of the war. More and more countries begin to take precise attack as an important direction of the development of missile technology. In order to hit the target accurately and timely, it is necessary to predict the impact point of the projectile accurately and quickly, then adjust the actuator of the projectile according to the deviation between the predicted impact point and the target point, so as to correct the trajectory and reduce the miss distance.

At present, the numerical integration method is mainly used to predict the impact point of projectile [1]. Firstly, the external ballistic equation needs to be established according to the motion characteristics of different projectiles. Then, the flight state parameters of a certain point during the projectile flight are brought into the external ballistic equation as the integral initial value of the equation. Finally, the Runge-Kutta method is used to solve the ballistic equation until the projectile lands and the impact point is obtained. The accuracy of the numerical integration method is determined by the performance of the missile-borne computer and the integration step. In order to obtain higher accuracy, it is necessary to choose a smaller integration step, which will make the calculating time too long and require more computing resources. At the same time, the cumulative error in the integration process will also increase. On the basis of the numerical integration method, some scholars introduced Kalman filter methods into projectile impact point prediction [2-4]. These methods reduce the influence of random noise and cumulative error on prediction accuracy to a certain extent. The introduction of Kalman filter methods improves the fault tolerance of the numerical integration method, but it needs to establish a more accurate system state model and measurement model. Because the ballistic model is a nonlinear model, the amount of calculation will increase during the filtering process. Therefore, the calculating time will
increase. Some other scholars proposed polynomial fitting methods to predict the impact point [5-6]. These methods calculate the impact point by fitting the relationships between the state variables of the projectile and the flight time. The calculating time of the methods is faster, but the accuracy needs to be improved.

In recent years, more and more intelligent algorithms have been applied in system control, pattern recognition, performance prediction and other fields. Artificial neural network (ANN) is one of them. Inspired by the operation mechanism of biological neural network, ANN uses distributed parallel computing method to process the input information, and uses the nonlinear activation function in the transmission of the input information, which makes it have the ability of nonlinear mapping and fault tolerance. ANN is very suitable for dealing with nonlinear systems questions. In the studies of various ANNs, back propagation (BP) neural network is a widely used method. However, traditional BP neural network is easy to fall into local minimum and has slow convergence speed. These problems limit the performance of BP neural network. Genetic algorithm (GA) is a global random search optimization algorithm based on the biological evolution mechanism. It can search the optimal solution globally in complex nonlinear space, so it is widely used in various optimization problems. In order to get higher accuracy and faster convergence speed, this paper uses GA to optimize the initial weights and thresholds of BP neural network [7-9]. The optimized BP neural network is used to predict the impact point of projectile.

2. BP Neural Network Optimized by GA

2.1. Basic Principle of BP Neural Network

BP neural network is a multilayer feed-forward neural network with an input layer, a number of hidden layers and an output layer. The neurons between the layers are fully connected, and the neurons in each layer are not connected to each other. BP neural network adopts the supervised learning rule. The forward transmission of inputs which are from the input layer will produce the actual outputs of the network. By comparing the actual outputs with the expected outputs, the error of the network is obtained, and the error is propagated from the output layer to the network. The gradient descent method is used to modify the weights and thresholds of each layer of the network until the error is less than the set value [10]. A typical three-layer BP neural network structure is shown in Fig. 1.

![Figure 1. The structure of a three-layer BP neural network.](image)

As shown in the Fig. 1, there are \( n \) neurons in the input layer, \( h \) neurons in the hidden layer and \( m \) neurons in the output layer. \( w_{ij} \) is the connection weight between the \( j^{th} \) neuron of the input layer \( x_i \) and the \( i^{th} \) neuron of the hidden layer, \( b_i \) is the threshold of the \( i^{th} \) neuron of the hidden layer. \( w_{ki} \) is the connection weight between the \( i^{th} \) neuron of the hidden layer and the \( k^{th} \) neuron of the output layer \( y_k \), \( a_k \) is the threshold of \( y_k \). Besides, \( \varphi \) represents the activation function of the hidden layer and \( \psi \) represents the activation function of the output layer.

In the process of forward transmission of inputs, the relationships among the layers are as (1)-(4):
\begin{equation}
net_i = \sum_{j=1}^{n} w_{ij}x_j + b_i, \quad i = 1, 2, \cdots, h
\end{equation}

\begin{equation}
o_i = \phi\left(\sum_{j=1}^{n} w_{ij}x_j + b_i\right)
\end{equation}

\begin{equation}
nets_k = \sum_{j=1}^{k} w_{kj}\phi\left(\sum_{j=1}^{n} w_{ij}x_j + b_i\right) + a_k, \quad k = 1, 2, \cdots m
\end{equation}

\begin{equation}
y_k = \psi\left[\sum_{j=1}^{k} w_{kj}\phi\left(\sum_{j=1}^{n} w_{ij}x_j + b_i\right) + a_k\right]
\end{equation}

Where \( net_i \) and \( o_i \) are the input and output of the \( i \)-th neuron of the hidden layer respectively, \( nets_k \) and \( y_k \) are the input and output of the \( k \)-th neuron of the output layer respectively.

The error function of the network is defined as:

\begin{equation}
E_r = \frac{1}{2S} \sum_{i=1}^{S} \sum_{k=1}^{m} (T_k - y_k)^2
\end{equation}

Where \( S \) is the number of all samples, \( m \) is the number of output neurons, \( T_k \) and \( y_k \) are the expected output and actual output of the \( k \)-th neuron of the output layer.

After obtaining the error of the network, the parameters of each layer can be adjusted according to the negative gradient direction of the error function according to the gradient descent method.

The process of correcting the weights and thresholds of each layer is as (6)-(9):

\begin{equation}
\Delta w_{ij} = \eta \phi'\left(\sum_{j=1}^{n} w_{ij}x_j + b_i\right) \sum_{i=1}^{S} (T_i - y_i)\phi'(nets_k)
\end{equation}

\begin{equation}
\Delta a_i = \eta \sum_{i=1}^{S} (T_i - y_i)\phi'(nets_k)
\end{equation}

\begin{equation}
\Delta w_{kj} = \eta \phi'\left(\sum_{j=1}^{n} w_{ij}x_j + b_i\right) \sum_{i=1}^{S} (T_i - y_i)\phi'(nets_k)w_{ij}
\end{equation}

\begin{equation}
\Delta b_k = \eta \phi'\left(\sum_{i=1}^{n} w_{ij}x_j + b_i\right) \sum_{i=1}^{S} (T_i - y_i)\phi'(nets_k)w_{ij}
\end{equation}

Where \( \eta \) is the learning rate, \( \Delta w_{ij} \) and \( \Delta w_{kj} \) are the correction of weights, \( \Delta a_i \) and \( \Delta b_k \) are the correction of thresholds.

2.2. GA-BP Neural Network

GA is a kind of global optimization algorithm through simulating the process of nature selection and genetic mechanism \[11\]. It starts at a group which is a potential solution set of the problem and generates a new group which is better adapted to the environment after selection, crossover and mutation operation. This process continues until the optimal population is generated or the number of iterations is reached. After decoding, the optimal individual in the optimal population or the last generation population can be regarded as the approximate optimal solution of the problem.

Because of the strong searching ability to find the best solution in the global space, this study uses GA to optimize the initial weights and thresholds of BP neural network to enhance network performance. The optimization process is as follows:

2.2.1. Initialization of BP neural network: Determine the structure of BP neural network and initialize weights and thresholds.
2.2.2. **Initialization of population:** Set the population size of GA and code the weights and thresholds as an individual. The coding method is real coding in this paper. Length of the individual \( L \) is as:

\[
L = n \times h + h \times m + h + m
\]  

(10)

Where the number of input layer neurons is \( n \), the number of hidden layer neurons is \( h \) and the number of output layer neurons is \( m \).

2.2.3. **Fitness function evaluation:** Fitness function value is evaluated to find the best individual in the group. It is defined as:

\[
F = \frac{1}{\sum_{i=1}^{S} \sum_{k=1}^{m} |T_k - y_k|}
\]  

(11)

Where \( S \) is the number of all samples, \( m \) is the number of output neurons, \( T_k \) and \( y_k \) are the expected output and actual output of the \( k \)th neuron of the output layer.

2.2.4. **Selection operation:** The purpose of selection operation is to select the excellent individuals from current population to propagate descendants in next generation. The roulette wheel method is used in this paper. The selection probability of the \( i_{th} \) individual \( p_i \) is as:

\[
p_i = \frac{F_i}{\sum_{j=1}^{N} F_j}
\]  

(12)

Where \( F_i \) is the fitness value of the \( i_{th} \) individual, \( N \) is the population size.

2.2.5. **Crossover operation:** Crossover operation will create two new individuals by exchanging genes from two different individuals. It is as:

\[
C_1 = P_1 \alpha + P_2 (1 - \alpha)
\]  

\[
C_2 = P_1 (1 - \alpha) + P_2 \alpha
\]  

(13)  

(14)

Where \( C_1 \) and \( C_2 \) are new individuals after crossover operation, \( P_1 \) and \( P_2 \) are two crossover individuals, \( \alpha \) is a random number between 0 and 1.

2.2.6. **Mutation operation:** Mutation operation can change the value of a gene string structure data with a small probability and create a new individual. It is as:

\[
a_{ij} = \begin{cases} 
    a_{ij} + (a_{ij} - a_{\text{max}}) \ast f(g) & r \geq 0.5 \\
    a_{ij} + (a_{\text{max}} - a_{ij}) \ast f(g) & r < 0.5
\end{cases}
\]  

(15)

Where \( a_{ij} \) is the \( j_{th} \) gene of the \( i_{th} \) individual, \( a_{\text{max}} \) and \( a_{\text{min}} \) are the upper bound and the lower bound of \( a_{ij} \), \( f(g) = \frac{r_2 (1 - g / G_{\text{max}})}{r_2} \), \( r_2 \) is a random number, \( g \) is the number of current iterations, \( G_{\text{max}} \) is the maximum number of iterations, \( r \) is a random number between 0 and 1.

Through the optimization process of GA, BP neural network obtains better initial weights and thresholds, which improves its performance compared with random initialization.

3. **The Impact Point Prediction Method Based on GA-BP Neural Network**

3.1. **Determination of the Network Structure**
According to the exterior ballistic theory, there is a nonlinear relationship between the current state parameters and the impact point of the projectile. Since BP neural network has a good approximation relationship to the nonlinear system, a mapping relationship between the current state parameters and the impact point of the projectile can be obtained by means of BP neural network. Through the mapping relationship, the impact point corresponding to the current state parameters of the projectile can be obtained without the need for iteration calculation.

Considering the easy accessibility of the parameters during the flight, the component of current position and velocity of projectile in ground coordinate system are selected as the inputs of the input layer, and the impact point information (range and lateral deviation) is selected as the outputs of the output layer.

For any given continuous function, a three-layer neural network can achieve its mapping relationship [12]. Therefore, a three-layer BP neural network is used in this paper. There is no definite method for the selection of the number of neurons in the hidden layer at present. Empirical formula is used to determine the number:

\[ h = \sqrt{n + m + a} \]  

(16)

Where \( n \) is the number of input layer neurons, \( m \) is the number of output layer neurons, \( a \) is an integer between 1 and 10, \( h \) is the number of hidden layer neurons. By simulating the number of hidden layer neurons with different values, we select 13 hidden layer neurons at the end.

In summary, the structure of the neural network for impact point prediction is 6-13-2, as shown in Fig. 2.

In the figure, \( x \), \( y \) and \( z \) are the component of current position of the projectile in ground coordinate system, \( v_x \), \( v_y \) and \( v_z \) are the component of current velocity of the projectile in ground coordinate system, \( X \) is the range and \( Z \) is the lateral deviation of the projectile.

3.2. Training of the Network

The trajectory data for GA-BP neural network training are obtained by solving the six-degree-of-freedom external ballistic equation with the fourth-order Runge-Kutta method under different initial fire angles. The integration step of the fourth-order Runge-Kutta method is set to 0.005s. The range of fire angle is from 6° to 70° and the external ballistic equation is calculated every 2° to obtain the trajectory data. The acquisition time of the projectile state parameters is from 5s after firing to 5s before landing, and the acquisition interval is 0.5s [13]. 4646 sets of data are obtained by matching the collected state
parameters with the impact point information one by one. In order to avoid the situation of over-fitting and improve the generalization ability of the network, 80% of the data set is set as the training data set and 20% of the data set is set as the validation data set. Before starting the training of network, it is necessary to normalize the data to eliminate the influence of the difference in the order of magnitude between the data on the network training.

After the training data is preprocessed, the initial weights and thresholds of the BP neural network are optimized by GA, and then the optimized initial weights and thresholds are brought into the network to train it. After the error of the network converges, the GA-BP neural network for projectile impact point prediction is obtained.

4. Simulation and Analysis
Three sets of trajectory data outside the training data set (fire angles are 35°, 45° and 55° respectively) are selected and 30 trajectory state points are randomly selected from each set of trajectory data. The state parameters of the 30 trajectory state points are used as the inputs of GA-BP neural network which has been trained to predict the impact point. The absolute values of the range prediction error and the lateral deviation prediction error are shown in Fig. 3 and Fig. 4. The absolute values of the maximum, minimum and average values of the prediction errors for each set of data are shown in Table I.

![Figure 3. The prediction error of range.](image1)

![Figure 4. The prediction error of lateral deviation.](image2)
Table 1. Statistics of Prediction Error

| Error Statistics                                                                 | The first data set | The second data set | The third data set |
|---------------------------------------------------------------------------------|--------------------|---------------------|-------------------|
| The maximum of range error (BP) (m)                                            | 16.72              | 17.24               | 34.18             |
| The maximum of range error (GA-BP) (m)                                        | 11.31              | 10.20               | 11.36             |
| The minimum of range error (BP) (m)                                           | 2.15               | 0.02                | 0.08              |
| The minimum of range error (GA-BP) (m)                                        | 2.68               | 4.29                | 2.03              |
| The average of range error (BP) (m)                                           | 9.57               | 9.26                | 12.07             |
| The average of range error (GA-BP) (m)                                        | 6.86               | 7.35                | 7.31              |
| The maximum of lateral deviation error (BP) (m)                               | 2.23               | 2.14                | 2.19              |
| The maximum of lateral deviation error (GA-BP) (m)                            | 0.48               | 0.65                | 0.65              |
| The minimum of lateral deviation error (BP) (m)                               | 0.05               | 0.01                | 0.03              |
| The minimum of lateral deviation error (GA-BP) (m)                            | 0.02               | 0.01                | 0.02              |
| The average of lateral deviation error (BP) (m)                                | 1.10               | 1.08                | 0.96              |
| The average of lateral deviation error (GA-BP) (m)                            | 0.31               | 0.30                | 0.30              |

The simulation results show that both BP neural network and GA-BP neural network can achieve the mapping of the current state parameters of the projectile to the impact point with high accuracy. Compared with BP neural network, the overall prediction error of GA-BP neural network is smaller, and the prediction error value is more stable, indicating that the training process of GA-BP neural network is better than BP neural network and GA-BP neural network has the better generalization performance. In practical application, the amount of data in training data set can be enlarged to obtain more accurate prediction results.

At the same time, 10 trajectory state points are randomly selected from the total of 90 state points, and the calculating time of numerical integration method and GA-BP neural network method are counted respectively. The results are shown in Table II. From the statistical results of calculating time, the average time of numerical integration method is 243.01ms, and that of GA-BP neural network method is 24.45ms. GA-BP neural network method has an advantage in the calculating time of projectile impact point prediction. In practical application, the training process can be completed offline. Only the trained neural network model needs to be loaded into the projectile computer for real-time prediction of projectile impact point. Therefore, a lot of training time can be saved to meet the rapid requirement of projectile impact point prediction.

Table 2. Statistics of Calculating Time

| Number of Test Samples | Numerical integration (ms) | GA-BP neural network (ms) |
|------------------------|----------------------------|----------------------------|
| 1                      | 120.1                      | 22.3                       |
| 2                      | 265.2                      | 21.6                       |
| 3                      | 225.7                      | 19.1                       |
| 4                      | 170.2                      | 25.6                       |
| 5                      | 270.5                      | 27.7                       |
| 6                      | 322.1                      | 24.3                       |
| 7                      | 302.6                      | 28.2                       |
| 8                      | 276.4                      | 25.2                       |
| 9                      | 196.5                      | 22.6                       |
| 10                     | 280.8                      | 27.9                       |

5. Conclusion

In order to realize the rapidity and accuracy of projectile impact point prediction, this paper introduces BP neural network algorithm, and improves the standard BP neural network by genetic algorithm. Based on the genetic algorithm, a GA-BP neural network model for predicting the impact point of projectile is established, and then the model is trained by a large number of projectile state parameters and the impact point information. Finally, the simulation test of the neural network model for predicting the impact point is carried out. The simulation results show that GA-BP neural network has the better performance.
for prediction than standard BP neural network. Meanwhile, the calculating time of GA-BP neural network method is less than that of numerical integration method. Therefore, the method proposed in this paper can predict projectile impact point with high accuracy and less calculating time, which can provide reference for the practical application.

Acknowledgments
This study is supported by the National Nature Science Foundation of China (61603191).

References
[1] Z. P. Han, Exterior ballistics of Projectiles and Rockets. Beijing Institute of Technology Press, 2014.
[2] J. G. Shi, M. Liu, C. Z. Cao, and Z. Y. Wang, “Research on method of falling point prediction for trajectory correction projectile,” Journal of Ballistics, vol. 26, Jun. 2014, pp. 29-30.
[3] F. Kumru, T. Akca, and R. S. Acar, “Performance evaluation of nonlinear filters on impact point prediction of ballistic targets,” International Conference on Information Fusion, ISIF Press, Jul. 2017.
[4] C. W. Li, M. Gao, D. Fang, and Z. C. Lu, “Fading memory filter-based impact point prediction of trajectory correction projectile,” Journal of Projectiles, Rockets, Missiles and Guidance, vol. 35, Jun. 2015, pp. 134-137.
[5] G. L. He, B. H. Ma, K. S. Chen, and B. Q. Wang, “A speedy arithmetic of trajectory point of fall based on GPS Measures,” Journal of Detection and Control, vol. 25, Mar. 2003, pp. 33-36.
[6] J. X. Chen, Z. K. Liu, Z. Liu, and J. W. Xia, “Study on impact point prediction based on GPS,” Ship Electronic Engineering, vol. 37, Aug. 2017, pp. 48-52.
[7] J. W. Tian, and M. J. Gao, Study and Application of Artificial Neural Network. Beijing Institute of Technology Press, 2006.
[8] J. G. Huang, H. Luo, H. J. Wang, and B. Long, “Prediction of time sequence based on GA-BP neural net,” Journal of University of Electronic Science and Technology of China, vol. 38, Sep. 2009, pp. 687-692.
[9] W. H. Zhou, and S. Q. Xiong, “Optimization of BP neural network classifier using genetic algorithm,” International Conference of Intelligence Computation and Evolutionary Computation, Springer press, Jul. 2012, pp. 599-605.
[10] H. Li, C. X. Hu, and Y. Li, “Application of the purification of materials based on GA-BP,” International Conference on Future Electrical Power and Energy Systems, Elsevier Ltd Press, 2012, pp. 726-729.
[11] T. S. Yan, “Research on neural network training algorithm based on genetic algorithm,” Journal of Hunan Institute of Science and Technology (Natural Sciences), vol. 20, Mar. 2007, pp. 31-34.
[12] E. Ozgan, “Artificial neural network based modelling of the Marshall Stability of asphalt concrete,” Expert Systems with Applications, vol. 38, May. 2007, pp. 6025-6030.
[13] X. Huang, H. D. Zhao, and Z. P. Li, “A projectile impact-point prediction method based on interpolation radial basis function neural network,” Journal of Detection and Control, vol. 37, Aug. 2015, pp. 101-105.