

Integration of PSO and BP Neural Network for Building the Artillery Ballistic Model

Yi-Wei Chen, Yung-Lung Lee

National Defense University, Taiwan

0424

The Asian Conference on Society, Education and Technology 2013

Official Conference Proceedings 2013

Abstract

The artillery firing precision plays an important role in the war and it's hard to describe the projectile trajectory in a mathematical model. In this paper, the neural network is used to build the artillery ballistic model for range prediction and Particle Swarm Optimization (PSO) is applied to optimize the initial weight and bias to accelerate the training speed. Besides, some firing data from one middle-caliber artillery are utilized in orthogonal array to reduce the experimental runs. The result shows that the proposed method has the faster training speed and better precision of range prediction than the traditional neural networks and proves to build quickly a suitable artillery ballistic model in less firing data without the complicated mathematical equations.

Keywords: Particle swarm optimization, neural network, orthogonal arrays, ballistic

1. Introduction

The artillery firing precision influences the war result deeply. There are many complicated reasons to influence the projectile trajectory. The projectile weight, ammunition quantity, cannonball shape and so on, are set in the fabrication process and which are the controllable factors. The projectile after leaving the muzzle will be affected by the uncontrollable factors such as initial velocity, air temperature, air pressure, relative humidity, wind velocity, wind direction, coriolis force and so on, but it's very difficult to build the ballistic mathematical model. For many applications, buying the foreign artillery and using the ammunition made by themselves is the tendency but the ammunition is not suitable for the original factory range table in firing. Therefore, some of military researches are focused on building a suitable artillery ballistic model to predict the range precisely and quickly.

It is hard to explain the complicated relation between input and output by a mathematical model for the physical system. The neural network is developed gradually and used extensively without the complex mathematical model, and that combined with design of experiments can not only reduce the experimental runs but also build a model for fitting the true system effectively. Wang et al. proposed the experimental results of orthogonal array (O.A.) to be the training data of neural network for building a model which can make prediction, interpolation, extrapolation and optimization [1]. Chang et al. also suggested the experimental results of orthogonal array to be the training data of neural network to build the back-propagation neural network (BPN) which can simulate the feasible domain for the optimal filter design. The result not only decreased the experimental runs but also gained better results than the common scheme [2]. Besides, the neural network was often combined with Taguchi-design of experiment to decrease the experimental runs and carry out the parameter optimization, experiment validation, analysis of variance and so on [3].

Some studies used PSO to optimize the parameters of BPN to improve the local optimum produced from the application of neural network. The results showed the higher prediction precision and faster convergence speed than traditional BPN [4-6]. PSO and genetic algorithm (GA) to combine with BPN in reservoir parameter dynamic prediction are also proposed. The results indicated that PSO-BP neural network is superior to GA-BP neural network and the traditional neural network [7]. In this study, the artillery ballistic model for range prediction is built by integrating PSO and BPN, where the PSO is utilized to optimize the initial weight and bias of neural work and the orthogonal array is used to reduce the training samples of neural network. The proposed method can accelerate the training speed of neural network and improve the prediction precision.

Nomenclature	
t	iteration count of PSO
T_{\max}	the maximal iteration count of PSO
w	inertia weight of PSO
E	error function of neural network
d_k	actual or expected output in the neural network
y_k	output value of output layer neuron in the neural network
y_j	output value of hidden layer neuron in the neural network
w_{ji}	weight of neural network
Δw_{ji}	weight correction of neural network
b_j	bias of neural network
Δb_j	bias correction of neural network
η	learning rate of neural network
r_1, r_2	random numbers between 0 and 1 in PSO
c_1, c_2	learning factors which are positive constants in PSO
c_{1ini}, c_{2ini}	the minimal constants initially in PSO
c_{1end}, c_{2end}	the maximal constants at the end in PSO

2. Particle Swarm Optimization

PSO algorithm means that a group of solutions are produced randomly called population in the beginning and each individual is a particle which replaces a random solution for the optimum of problem. The basic concept is to imitate the social behavior of foraging from a flock of birds [8,9]. Each particle will search continuously and memorize the optimal solution called particle best value (pbest) in problem space during the evolutionary process. Furthermore, the optimal solution between particles during the evolutionary process is considered and called global best value (gbest). When the population is composed of m particles and each particle will search the optimal solution on D dimension space. The location of particle i is denoted $X_i=(x_{i1},x_{i2},\dots,x_{iD})$. and the velocity is denoted by $V_i=(v_{i1},v_{i2},\dots,v_{iD})$, $1 \leq i \leq m$, $1 \leq d \leq D$, and the location of optimal solution is denoted by $P_i=(p_{i1},p_{i2},\dots,p_{iD})$, the location of global optimal solution is denoted by $P_g=(p_{g1},p_{g2},\dots,p_{gD})$. The update location and velocity for each particle during the evolutionary process is as follows:

$$v_{id}(t+1) = wv_{id}(t) + c_1r_1(p_{id}(t) - x_{id}(t)) + c_2r_2(p_{gd}(t) - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

The linear decreasing inertia weight w is also derived from Eberhart and Kennedy in 1998 [10]. The larger w makes the particle owing the large exploration, and the smaller w makes the particle owing the large exploitation. The inertia weight w was usually set to decrease gradually from 0.9 to 0.4 in most literatures. Ratnaweera etc. thought that the local optimum would still produce even if the inertia weigh was used. That's because the learning factors were constants to restrict the algorithm. As a result, the iteration count t is used to change the learning factors dynamically [11]. All the

relative equations are as follows:

$$w = \frac{(w_{ini} - w_{end}) * (T_{max} - t)}{T_{max}} + w_{end} \quad (3)$$

$$c_1 = (c_{1end} - c_{1ini}) \frac{t}{T_{max}} + c_{1ini} \quad (4)$$

$$c_2 = (c_{2end} - c_{2ini}) \frac{t}{T_{max}} + c_{2ini} \quad (5)$$

3. Back-Propagation Neural Network

The basic concept of artificial neural network is to imitate the nervous system of organism. It's composed of numerous nonlinear operational units (nerve cells) and connection between operational units. The neural network can construct the model which can explain the correlation between input and output by the observation data without the mathematical model. The back-propagation network (BPN) is the most popular neural network algorithm. The scheme of BPN is shown in Fig. 1. BPN consists of input, hidden and output layers. The training process of neural network is composed of forward pass, error computation and error back-propagation. In the forward pass, a neuron driven by the input signal produces the output that differs from the actual or desired target output and leads to the error. The gradient steepest descent method is applied to minimize the error function. The error signals are then back propagated through the network from output layer to input layer as a sequence of corrective adjustment called weight modification. The amount of hidden layers, neurons, learning rate and transfer function are modified properly in the training process to minimize the error. The relative mathematical equations are described as follows:

$$y_j^n = f(net_j^n) \quad (6)$$

$$net_j^n = \sum_i w_{ji}^n y_i^{n-1} - b_j^n \quad (7)$$

$$E = \frac{1}{2} \sum_k (d_k - y_k)^2 \quad (8)$$

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \quad (9)$$

$$w_{ji} = w_{ji} + \Delta w_{ji} \quad (10)$$

$$\Delta b_j = -\eta \frac{\partial E}{\partial b_j} \quad (11)$$

$$b_j = b_j + \Delta b_j \quad (12)$$

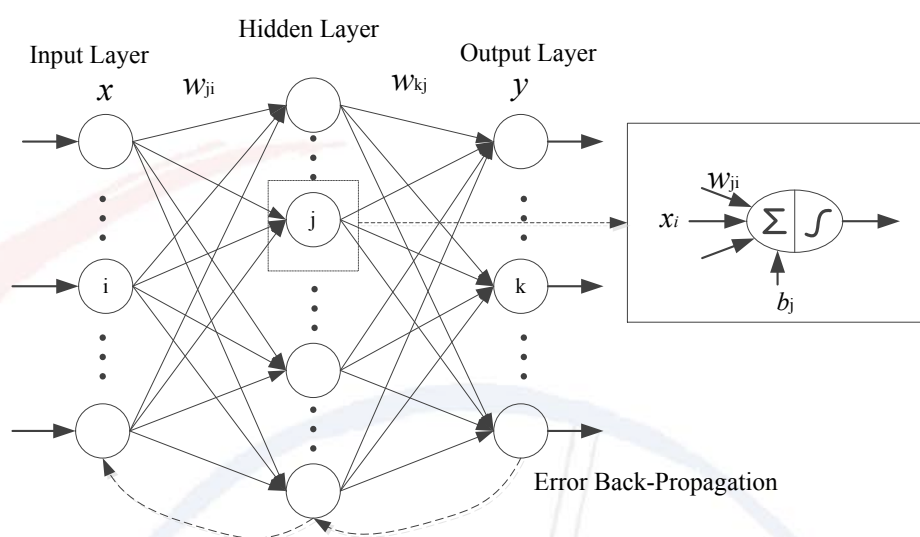


Fig. 1. Back-propagation neural network.

4. Orthogonal Array and Artillery Ballistic Data

The property of orthogonal arrays is to obtain the same effective information as the full factorial experiment, and the experimental runs will be reduced. The type of orthogonal arrays include two-level, three-level and mixed-level and which are usually shown in $L_A(B^C)$. A , B and C represent respectively the number of experimental runs, level and factor. The major purpose of this paper is to build the artillery ballistic model to predict precisely the range. Angle of departure, muzzle velocity, air temperature, air pressure, wind velocity, wind direction and relative humidity are the main factors to influence the artillery range, in which the wind velocity contains both of the following wind and the cross wind, and each of them includes the downwind direction and upwind direction. The downwind direction means the same direction with the motion trajectory of projectile and the upwind direction is on the contrary. Because the correlation between the artillery range (output) and the variables mentioned above (input) are very complex, the three-level design are used to each variable except that the wind direction is in two-level design.

In this paper, the firing data come from one middle-caliber artillery. Table 1 shows the data scope and $L_{36}(2^2 \times 3^7)$ orthogonal array is utilized to investigate the prediction effectiveness of ballistic model built by the PSO-BP neural network (Table 2).

Table 1. The artillery ballistic data scope

Variables	Minimum	Maximum
Angle of departure (°)	1	30
Muzzle velocity (m/s)	1000	1010
Air temperature (oC)	5	39
Relative humidity (%)	50%	100%
Air pressure (mb)	1002	1019
Wind velocity (m/s)	1.7	6.3
Wind direction	Downwind	Upwind

Note:

1. The standard muzzle velocity is 1005(m/s) and the data range is defined by the fabrication deviation of ammunition.
2. The range of atmosphere data is defined by Taiwan climate condition.

Table 2. $L_{36}(2^2 \times 3^7)$

Experimental Runs	Variables Column								
	A	B	C	D	E	F	G	H	I
1	1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2	2
3	1	1	3	3	3	3	3	3	3
4	1	1	1	1	1	1	2	2	2
5	1	1	2	2	2	2	3	3	3
6	1	1	3	3	3	3	1	1	1
7	1	1	1	1	2	3	1	2	3
8	1	1	2	2	3	1	2	3	1
9	1	1	3	3	1	2	3	1	2
10	1	2	1	1	3	2	1	3	2
11	1	2	2	2	1	3	2	1	3
12	1	2	3	3	2	1	3	2	1
13	1	2	1	2	3	1	3	2	1
14	1	2	2	3	1	2	1	3	2
15	1	2	3	1	2	3	2	1	3
16	1	2	1	2	3	2	1	1	3
17	1	2	2	3	1	3	2	2	1
18	1	2	3	1	2	1	3	3	2
19	2	1	1	2	1	3	3	3	1
20	2	1	2	3	2	1	1	1	2
21	2	1	3	1	3	2	2	2	3
22	2	1	1	2	2	3	3	1	2
23	2	1	2	3	3	1	1	2	3
24	2	1	3	1	1	2	2	3	1
25	2	1	1	3	2	1	2	3	3
26	2	1	2	1	3	2	3	1	1
27	2	1	3	2	1	3	1	2	2
28	2	2	1	3	2	2	2	1	1
29	2	2	2	1	3	3	3	2	2
30	2	2	3	2	1	1	1	3	3
31	2	2	1	3	3	3	2	3	2
32	2	2	2	1	1	1	3	1	3
33	2	2	3	2	2	2	1	2	1
34	2	2	1	3	1	2	3	2	3
35	2	2	2	1	2	3	1	3	1
36	2	2	3	2	3	1	2	1	2

5. Integration of Particle Swarm Optimization and Back-Propagation Neural Network

In this paper, PSO is utilized to search the optimal initial weight and bias of neural network to accelerate the training speed, and the gradient steepest descent method of BPN is applied to modify gradually the weight and bias for the minimization of error function. There are two key points about the optimization of neural network by PSO: Firstly, the particle's dimensions must be transferred into the initial weight and bias of neural network and each size is equal to the numbers of initial weight and bias. Secondly, we define the fitness function to compute the fitness of particle. The operation steps are explained as follows:

Step 1: Normalize the input data.

The 36 artillery ballistic data in orthogonal array are used to train the neural network, and we choose additionally 10 data for verification and evaluation. Because of the physical significance and dimensions of data are different and unsuitable for training the neural network. The normalization is used to transform these data into [0,1] domain and the equation is as follows:

$$X_{new} = \frac{X_{old} - X_{min}}{X_{max} - X_{min}} \times (D_{max} - D_{min}) + D_{min} \quad (13)$$

Where X_{old} , X_{new} are respectively the data of original and normalization, X_{min} , X_{max} are respectively the original data of minimum and maximum, D_{min} , D_{max} are respectively the minimum and maximum in [0,1] domain.

Step 2: Set the neural network parameters.

The hidden layers, neurons, learning rate and transfer function influence the neural network deeply such as the training speed and convergence condition. Try and error method and empirical formula from literatures are utilized to determine these parameters of neural network as follows: hidden layers: 2; hidden layer neurons: (7, 4); learning rate: 0.1; transfer function: sigmoid function. Mean absolute error (MAE) is used as the cost function and the goal error is 0.007.

Step 3: Initialize PSO parameters.

The location and velocity of D-dimensions particle are random in [0, 1] domain initially. The D-dimensions particle is transferred into the initial weight and bias of neural network and each size is equal to the numbers of initial weight and bias. The linear decreasing inertia weight is used and decreases gradually from 0.9 to 0.4 (Eq. 3). The learning factors are changed dynamically (Eq.4, 5) and c_{1ini} , c_{2ini} are 0.1, 0.6, c_{1end} , c_{2end} are 0.5, 1. Population size: 100; Iteration: 200.

Step 4: Input the training data.

Step 5: Compute the output value of hidden layers and output layer.

Step 6: Compute the fitness of particle and the fitness function is defined as mean absolute error.

$$F(i) = \frac{\sum_{q=1}^Q |(d_q - y_q)|}{Q}$$

(14)

d_q is the actual value, y_q is the output value of output layer, Q is the number of training samples.

Step 7: Compute and update the history optimal location of each particle, P_i which has the minimal fitness during the evolutionary process, and if the particle's fitness is the minimum compared with all, its location will be the global optimal location, P_g .

Step 8: Update the velocity and location.

Step 9: Repeat steps from step 4 to step 8 until the maximum iteration is satisfied.

Step 10: Input the optimal initial weight and bias obtained by PSO into the neural network for training.

Fig. 2 is the flow chart for the building of artillery ballistic model which combines the neural network with the orthogonal array. First of all, the actual range of orthogonal array design chosen outputs, 36D1 obtained from the ballistic data and the orthogonal array design chosen inputs, 36X1 are both normalized to train the BPN. The weight and bias will be modified gradually by error back-propagation if the value of cost function doesn't satisfy the goal error. Finally, we choose 10 ballistic data randomly to confirm the objectivity and efficiency of trained BPN besides the original 36 training ballistic data. The range outputs of trained BPN are denormalized and compared with the actual range that will produce an error. Under the premise that overfitting is avoided, the goal error will be modified properly according to the result to approach the purpose of error minimum and that the BPN range prediction will be close to the true range. As mentioned above, the neural network ballistic model is completed.

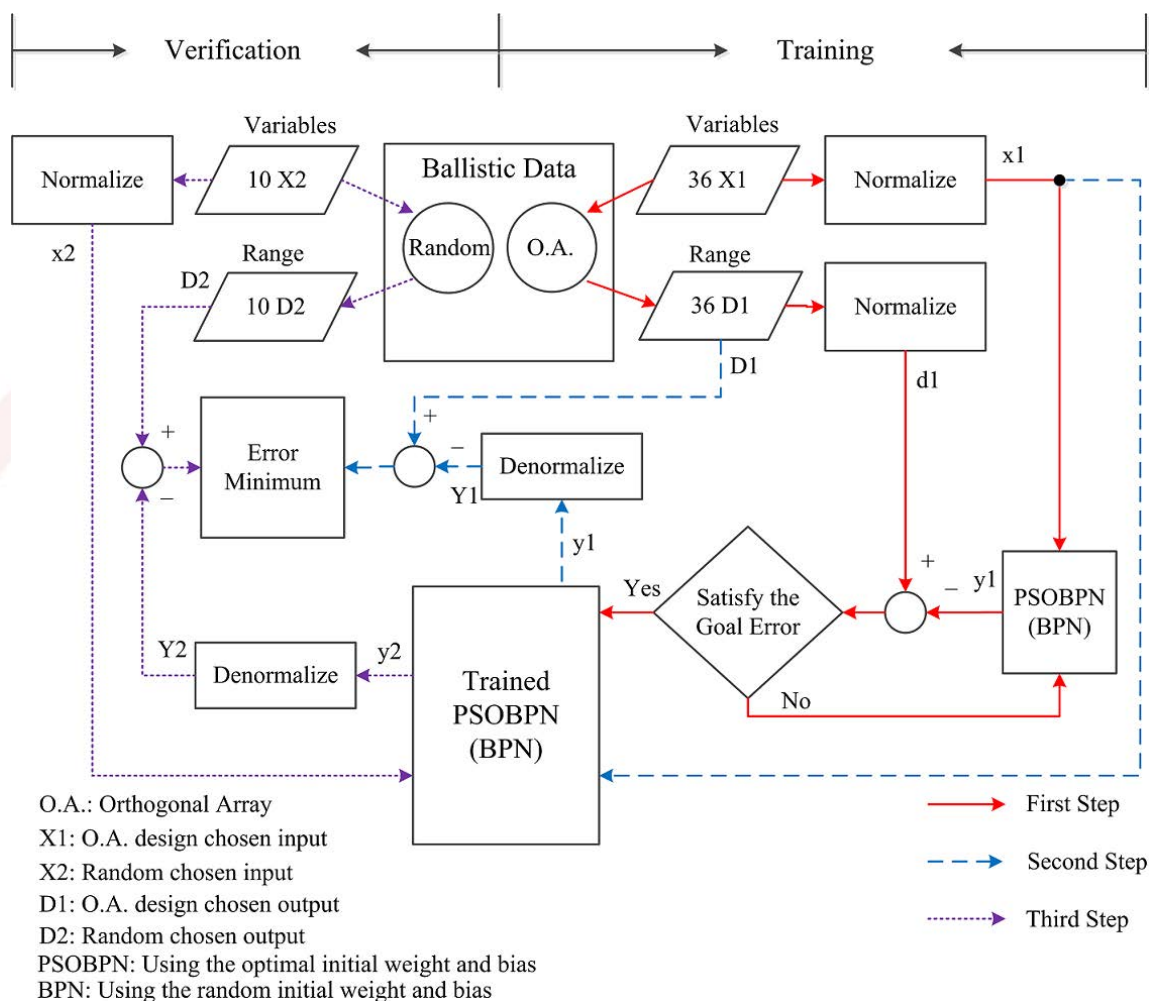


Fig. 2. Flow chart for the building of artillery ballistic model

6. Simulation Results

The 10 ballistic data chosen randomly are used to evaluate the performance of PSO-BP neural network combined with the orthogonal array (O.A. PSOBPN). Furthermore, there are two model built by the traditional BPN theory and which are utilized to compare with the proposed method. Non-orthogonal array BPN (Non-O.A. BPN): The ballistic model of BPN is built by 36 data without using the orthogonal array and these training data are widespread in the data scope to increase the precision of BPN. Orthogonal array BPN (O.A. BPN): The ballistic model of BPN is built by 36 data in orthogonal array. It's utilized to understand the performance of prediction precision when the neural network combines with the orthogonal array in less training samples.

Fig. 3-4 is the comparison of training time and prediction error for three neural network ballistic models. Each model is repeated for five times and the mean absolute percentage error (MAPE) between prediction and true is the index to evaluate the precision of neural network ballistic model. From the figure, Non-O.A. BPN needs the longest training time and obtains the worst prediction precision. These conditions will be improved greatly when the neural network combines with the orthogonal array shown in O.A. BPN and O.A. PSOBPN. In addition, the training speed of neural network will be accelerated obviously when the initial weight and bias is optimized by PSO. Fig. 5 is the range prediction error of three different BPN models, in which the prediction value of range is the mean of five repeats. For the same goal error in

the network training, the BPN model without combining with the orthogonal array has the worst prediction precision than O.A. BPN and O.A. PSOBPN. As a result, PSO-BP neural network which combines with the orthogonal array has the best performance for the artillery range prediction.

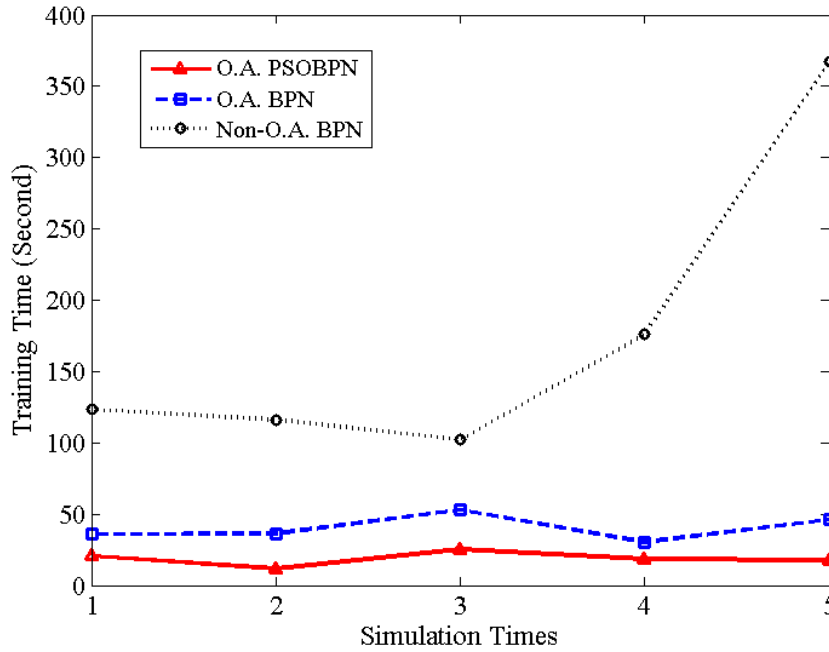


Fig. 3. Training time of three different BPN models

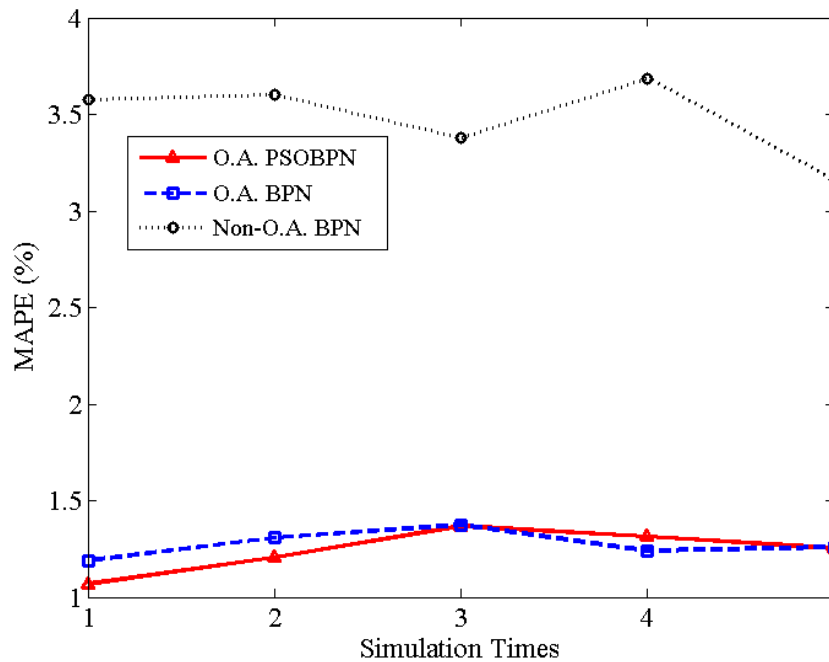


Fig. 4. Mean absolute percentage error of three different BPN models

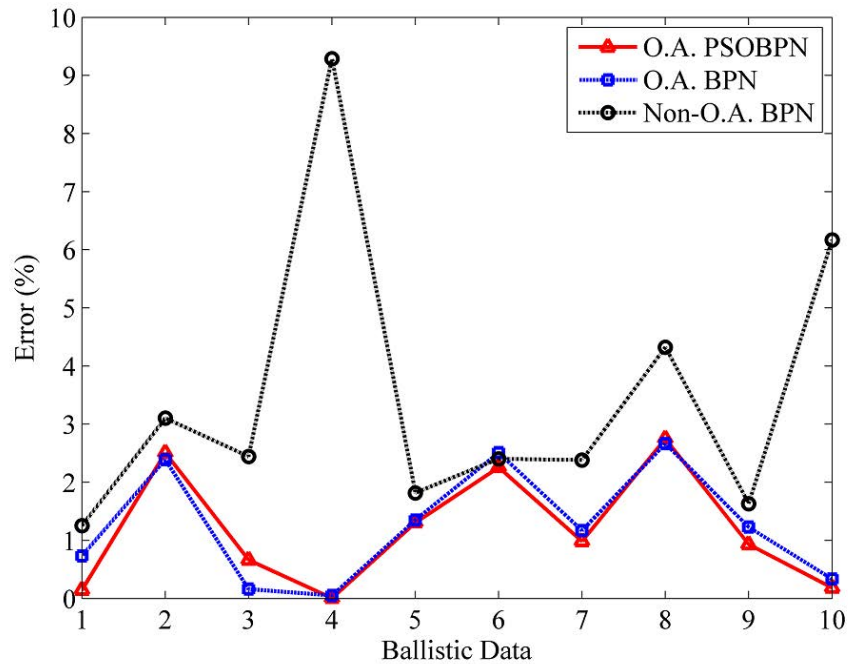


Fig. 5. Range error of three different BPN models

7. Conclusion

The firepower of artillery is one of main factors to decide the war result. The promotion of firing precision and accuracy for the artillery is always the research goal of nation defense industry. In this study, the intelligent artillery ballistic model based on PSO-BP neural network and orthogonal array is utilized for range prediction. The orthogonal array can improve both of the prediction precision and training time of neural network and the application of PSO can accelerate the training speed obviously. The proposed method has the better performance than the traditional BPN theory and which build the artillery ballistic model in less experimental data without the complicated mathematical model. The result can not only reduce the research cost and time but also suggest a method to build the artillery ballistic model for range prediction.

Acknowledgement

The authors would like to thank Prof. Yung-Lung Lee for valuable suggestions in this paper. Thanks are also due to anonymous reviewers for useful opinions to improve this article.

References

- [1] G. J. Wang, J. C. Tsai, P. C. Tseng and T. C. Chen, "Neural-taguchi method for robust design analysis", *Journal of the Chinese Society of Mechanical Engineers*, vol. 19, no. 2, pp. 223-230, 1998.
- [2] Ying-Pin Chang, Chinyao Low, Chi-Jui Wu, "Optimal design of discrete-value passive harmonic filter using sequential neural-network approximation and orthogonal array", *IEEE Transactions on Power Delivery*, vol. 22, no. 3, July 2007.
- [3] Dedy Purwanto, Herman Agustawan, Mohd Fakhizan Romlie, "The taguchi-neural network approach to forecast electricity consumption", *IEEE Electrical and Computer Engineering*, pp. 1941-1944, May 2008.
- [4] Pian Zhaoyu, Li Shengzhu, Zhang Hong, Zhang Nan, "The application of the PSO based BP network in short-term load forecasting", *Physics Procedia*, vol. 24, pp. 626-632, 2012.
- [5] Chun-Guang Chang, Ding-Wei Wang, Ya-Chen Liu, Bao-Ku Qi, "Application of particle swarm optimization based BP neural network on engineering project risk evaluating", Third International Conference on Natural Computation, 2007.
- [6] Jui-Fang Chang, Cheung-Wen Chang, Wen-Yan Tzeng, "Forecasting exchange rates using integration of particle swarm optimization and neural network", Fourth International Conference on Innovative Computing, Information and Control, 2009.
- [7] Liumei Zhang, Jianfeng Ma, Yichuan Wang, "PSO-BP neural network in reservoir parameter dynamic prediction", Seventh International Conference on Computational Intelligence and Security, 2011.
- [8] R. C. Eberhart, and J. Kennedy, "A new optimizer using particle swarm theory", Proc. Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39-43, 1995a.
- [9] J. Kennedy, and R. C. Eberhart, "Particle swarm optimization", Proc. IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, 1995b, pp. IV: 1942-1948.
- [10] J. Kennedy, and R. C. Eberhart, "Empirical study of particle swarm optimization", Proceedings of the 1999 Congress on Evolutionary Computation, vol. 3, 1999.
- [11] Asanga Ratnaweera, Saman K. Halgamuge, Member, IEEE, and Harry C. Waston, "Self-Organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, June 2004.

